

Képfeldolgozók és Alakfelismerők
IV. konferenciája

Miskolc-Tapolca
2004. január 28-30.



Szerkesztette:
Gácsi Zoltán, Barkóczy Péter, Sárközi Gábor



ITA/404

Képfeldolgozók és Alakfelismerők IV. konferenciája

Miskolc-Tapolca
2004. január 28-30.

Szerkesztette:
Gácsi Zoltán, Barkóczy Péter, Sárközi Gábor

A Konferencia a Neumann János Számítógéptudományi Társaság
Képfeldolgozók és Alakfelismerők Szakosztályának
hivatalos rendezvénye.

A Szakosztály vezetősége:

Elnök:

Kuba Attila

Titkár:

Ruskó László

Vezetőségi tagok:

Berke József

Csetverikov Dimitrij

Fazekas Attila

Gácsi Zoltán

Szabó József

Szirányi Tamás

A Konferencia támogatói:

Neumann János Számítógéptudományi Társaság

Miskolci Egyetem Fémtechnológiai Tanszéke

MTA Anyaginformatikai Albizottsága

Miskolci Akadémiai Bizottság Anyagtudományi Munkabizottsága

Nyomdai kivitelezés:

Duna-Print Nyomda, Dunaújváros

Előszó

Rendkívüli megtiszteltetés és nagy felelősség a Miskolci Egyetem Anyagtudományi Intézet Fémtani Tanszékének képfeldolgozással foglalkozó munkatársai számára, hogy 2004. január 28. és 30. között Miskolc-Tapolcán megrendezhetik a Neumann János Számítógép-tudományi Társaság (NJSZT) Képfeldolgozók és Alakfelismerők Szakosztályának (KÉPAF) 4. Konferenciáját.

Az összetett anyagok térbeli szerkezetének modellezésével (sztereológia) és a mikroszkópos szövetképek számítógépes feldolgozásával és elemzésével kapcsolatos ismeretek oktatása és kutatása a Miskolci Egyetemen hosszú múltra tekinthet vissza. Egy viszonylag fiatal, interdiszciplináris tudományterületről van szó, amelynek jelentősége az 1980-as évektől kezdve fokozatosan növekedett, s napjainkra – véleményünk szerint – az anyagok szerkezetével kapcsolatos kutatások egyik kulcsterületévé vált. Ez a tudományos módszer az oktatásban is egyre nagyobb szerepet kap, amit az is jelez, hogy a Miskolci Egyetemen létrejött az Anyaginformatikai Laboratórium, megszületett az Anyaginformatikai Ágazat, amelynek keretében évente 8-15 mérnökhallgató ismerkedik a számítógépes képelemzéssel. Évente 2-4 tudományos diákköri dolgozat és diplomaterv is születik a témában. Külön öröm számunkra, hogy jelenleg három fiatal oktató, egy kutató és két doktorjelölt kutatási témája kapcsolatos a képelemzéssel.

A konferencia megszervezésével a Képfeldolgozók és Alakfelismerők Szakosztályának az a célja, hogy két évente méltó keretet biztosítson a képfeldolgozással és képelemzéssel foglalkozó kutatók tudományos eszmecesterjének. Emellett a vezetőség keresi a kapcsolatot a képfeldolgozás területén működő vállalkozásokkal és gyakorlati szakemberekkel is. Ugyanis meggyőződéssel valljuk, hogy a tudomány igazi próbája a gyakorlat! Nem titkolt célja a konferenciának a képfeldolgozás egyetemi, főiskolai oktatási helyzetének áttekintése sem, hiszen az oktatás megfelelő színvonala biztosítja a tudományterület szakmai utánpótlását.

A szervezők remélik, hogy megfelelő körülményeket tudnak biztosítani, s a résztvevők maradandó élményekkel távoznak majd Miskolc-Tapolcáról.

A 4. KÉPAF konferencia megszervezését jelentősen támogatta a Neumann János Számítógép-tudományi Társaság. Szakmai segítséget nyújtott a Miskolci Egyetem Fémtani Tanszéke, a Magyar Tudományos Akadémia Anyaginformatikai Albizottsága, és az MTA Miskolci Akadémia Bizottság Anyagtudományi Munkabizottsága.

Miskolc, 2004. január 17.

Gácsi Zoltán, Barkóczy Péter, Sárközi Gábor
a Szervezők



Tartalomjegyzék

Neurális hálók alkalmazása elektronikus beléptető rendszerekben	1
<i>Balázsfalvi Gábor</i>	
Visualization and Control Interface for MRI Prostate Robot	7
<i>E. Balogh, A. Tanacs, A. Deguet, R.C. Susil, A. Krieger, A. Viswanathan, L.L. Whitcomb, E. Atalar, G. Fichtinger</i>	
Gyors szomszédság kereső algoritmus véletlen rácsú cella automatához ...	8
<i>Barkóczy Péter</i>	
Személyek követése a járásminta számítógépes analízisével	14
<i>Benedek Csaba, Szirányi Tamás</i>	
Fraktáldimenzió szerepe a képfeldolgozásban	20
<i>Berke József</i>	
Képtömörítő eljárások pszichovizuális összehasonlítása laboratóriumi körülmények között - MAMIKA	21
<i>Busznyák János, et al.</i>	
Visszacsatolás alapú képi indexelés	29
<i>Császár Gergely, et al.</i>	
Importance Sampling for X-Ray Volume Rendering	35
<i>Balázs Csébfalvi, László Szirmay-Kalos</i>	
3D modell készítése ortopédiai műtétek szimulálásához	43
<i>Csernátony Zoltán, Hajdu András, Manó Sándor, Zörgő Zoltán</i>	
Spatial Image Analysis in the Field of Metallography	50
<i>Csikós Tímea</i>	
Depth-based Indexing and Retrieval of Photographic Images	58
<i>László Czúni, et al.</i>	
Műtési tervek előkészítése véges elemes analízishez a MedEdit orvosi képfeldolgozó rendszer segítségével	65
<i>Erdőhelyi Balázs, Ollé Krisztián, Varga Endre, Halmai Csongor, Kuba Attila</i>	
Recognition of Facial Gestures	71
<i>Attila Fazekas</i>	
Összetett anyagok szövetszerkezetének rendezettsége	76
<i>Gácsi Zoltán, et al.</i>	

Robust Structure from Motion by Outlier Rejection	83
<i>Levente Hajder, Dmitry Chetverikov, István Vajk</i>	
Aktív kontúrok és Fast Marching eljárás alkalmazása az orvosi képfeldolgozásban	90
<i>Hajder Levente, Kardos István, Csetverikov Dmitrij, Renner Gábor</i>	
Analytical and approximation properties of neighborhood sequences	97
<i>András Hajdu, Lajos Hajdu</i>	
Distance functions in multidimensional image processing applications	106
<i>András Hajdu, Tamás Tóth, Krisztián Veréb, Zoltán Zörgő</i>	
Gépi látáson alapuló rendszerek humán felhasználói környezetben – A Török 2	112
<i>Hajdu András, Hodnics István Márton, Kovács Kornél, Ungvári Antal</i>	
Symmetry axis extraction	118
<i>László Havasi, et al.</i>	
Optical Flow meghatározása energiainimalizációs módszerrel	125
<i>Horváth Péter, Kató Zoltán</i>	
Szín, textúra és mozgás alapú szegmentálás Gradient Vector Flow segítségével	131
<i>Horváth Péter, Kató Zoltán</i>	
Precise Registration based on Photo-Consistency	137
<i>Zsolt Jankó, Dmitry Chetverikov</i>	
Reversible Jump Markov Chain Monte Carlo for Unsupervised MRF Color Image Segmentation	144
<i>Zoltan Kato</i>	
Color Textured Image Segmentation Using a Multi-Layer Markovian Model	152
<i>Zoltan Kato, Ting-Chuen Pong, Song Guo Qiang</i>	
Előfeldolgozás és diszkrét tomográfiai rekonstrukció a neutron radiográfiában	159
<i>Kiss Zoltán, et al.</i>	
Automatic Centerline Extraction from Tubular Volumetric Structures ...	165
<i>József Koloszar, László Szirmay-Kalos</i>	
Rendering and Coding of Still and Motion Picture by Stochastic Painting	171
<i>Levente Kovács, Tamás Szirányi</i>	
Automatic Stabilization of Vibration on Archive Films	178
<i>Attila Licsár, et al.</i>	

Wide Baseline Stereo: From Pixels to 3D Model	185
<i>D. Chetverikov, Z. Megyesi and Z. Jankó</i>	
Számítógépes képfeldolgozás oktatása a Szegedi Tudományegyetemen	191
<i>Nagy A., Balogh E., Dudásné Nagy M., Kuba A., Máté E., Palágyi K., Katona E., Kató Z., Nyúl L., Tanács A.</i>	
Szomszédási szekvenciák a háromszögrácson	197
<i>Nagy Benedek</i>	
National Instruments, Hungary	206
<i>National Instruments</i>	
Building 3D Models Using Calibrated Stereo and Structured Light	213
<i>Novák Norbert, Megyesi Zoltán</i>	
MedEdit: Műtéti tervezést segítő orvosi képfeldolgozó rendszer	221
<i>Ollé Krisztián, Erdőhelyi Balázs, Kuba Attila, Varga Endre, Halmai Csongor</i>	
Multimodalitású orvosi képfeldolgozás céljaira kidolgozott absztrakt szoftverkönyvtár	227
<i>Opposits Gábor, Valastyán Iván, Emri Miklós, Trón Lajos</i>	
Légutak vizsgálata 3D CT-képeken	232
<i>Palágyi Kálmán</i>	
Valódi és szimulált dendrites struktúrák morfológiai jellemzése	237
<i>Pólliska Csaba, et al.</i>	
Volume and surface models from CT/MR images	244
<i>Ferenc Pongrác, Gábor Renner</i>	
Körmetszeteket tartalmazó tárgyak rekonstrukciója néhány vetületből ...	252
<i>Rodek Lajos, Kiss Zoltán, Kuba Attila</i>	
Lézeres felületötvözési technológia ellenőrzése sztereológiai vizsgálatok segítségével	257
<i>Svéda Mária, et al.</i>	
Szemelvények a hazai képfeldolgozás (h)őskorából	263
<i>Szabó József</i>	
Képfeldolgozás alkalmazása gömbgrafitos öntöttvas szövetszerkezetének vizsgálatára	271
<i>Szalai Ibolya, et al.</i>	
Stochastic Radiosity on the Graphics Hardware	277
<i>László Szirmay-Kalos and Gábor Szijártó</i>	

Finding matching points based on motion-statistics from two views	284
<i>Zoltán Szlávik, et al.</i>	
Kerámiarészecskék bevonatának jellemzése elektronmikroszkópos kép feldolgozásával	291
<i>Tomolya Kinga, et al.</i>	
Vizuális objektumkövetés részecskeszűrőkkel lokális, likelihood alapú mintavételezéssel	297
<i>Torma Péter, Szepesvári Csaba</i>	
Validált kvantitatív PET szimulátor használata az orvosi képfeldolgozásban és diagnosztikai kutatásokban	303
<i>Valastyán Iván, Balkay László, Emri Miklós, Trón Lajos</i>	

Neurális hálók alkalmazása elektronikus beléptető rendszerekben

Balázsfalvi Gábor

University of Debrecen
Institute of Informatics

1. Előszó

A neurális hálózatokat sikerrel alkalmazzák az arcfelismerésben [1][6]. A gondolat, mely ezt a munkát elindította, a következő volt: A különböző, regisztrációt igénylő adattárakhoz, szolgáltatásokhoz világszerte hozzá lehet férni, és adott esetben több tízezer regisztrált felhasználó van. Az ilyen helyekre történő regisztrálás általában bonyolult felhasználónevek és jelszavak megjegyzésével jár. Ehelyett lehetne alkalmazni egy emberközelibb megoldást: a webkamera által rögzített kép alapján történő arcfelismerést.

Kutatásaink azt mutatják, hogy a neurális hálók (bevonva bizonyos fuzzy rendszereket) 95-100%-os biztonsággal azonosítanak egy arcot. Sok aktivációs függvényel, sokféle topológiával, valamint tanítási algoritmussal foglalkoztunk. A back propagation tanítási algoritmus a háromrétegű hálókon adta meg a legjobb összhangot a sebesség és a biztonság ellentétes érvei közt. Ezzel a módszerrel a háló percek alatt felkészül tíz-húsz arc befogadására, és 95-98%-os biztonsággal ismeri fel a megtanult arcokat.

2. Neurális Hálók

2.1. Modell

Egy neurális háló sok-sok **processzáló egységből** áll, melyek kapcsolatban vannak egymással. Ezeket a processzáló egységeket nevezzük **neuronnak**. Egy neuronnak bemeneti és kimeneti kapcsolatai vannak, melyeken keresztül „kommunikál” a többi neuronnal. Ez a „kommunikációs csatorna” egyirányú. Minden kapcsolatnak van súlya. Az i . neuron felé a j . neuron felől mutató kapcsolat súlyát $W_{j,i} \in R$ jelöli. Ezentúl a súly maga is képviselheti a kapcsolatot. A kapcsolatokon keresztül érkeznek az aktivációs értékek: $a_j \in R$. A neuron számítása két lépésből áll:

1. Kiszámítja a $b_i \doteq t_i + \sum_j W_{j,i} \times a_j$ (súlyozott bemenet) értéket. A $t_i \in R$ -t **torzításnak** nevezzük. A t_i minden neuronra különböző lehet, így pótolni tudjuk egy új súly bevezetésével. Ez a későbbiekben nagyon kényelmes megoldás lesz. Itt a szummázás azon j -kre történik, melyekhez tartozó neuronokból van kapcsolat az i . neuron felé.
2. Kiszámítja az $a_i \doteq g(b_i)$ értéket. Ezt nevezzük a **neuron számításának**. Itt g egy ún. aktivációs függvény, melyről a következő pont részletesebben szól. Majd mindezek után továbbküldi a kiszámított aktivációs értéket a vele összeköttetésben lévő neuronoknak.

Az aktivációs függvény. Az **aktivációs függvény** $g : \mathcal{R} \rightarrow \mathcal{A}$, ahol $\mathcal{A} = (0, 1)$ vagy $\mathcal{A} = (-1, 1)$. Egyes függvények felvehetik szélsőértékeiket is, de az ilyen függvények nem jellemzők. A taníthatóság miatt fontos, hogy ez folytonosan differenciálható legyen. Általában a $g(x) = \frac{1}{1+\exp(-x)}$ függvényt használják, mivel ez esetben a függvényérték ismeretében a derivált értéke könnyedén kiszámítható. Gyakran használt aktivációs függvények még a $g(x) = \tanh(x)$ és a $g(x) = \frac{2}{1+\exp(-x)} - 1$. A g deriváltjára vonatkozó összefüggések:

$$g'(x) = \begin{cases} g(x) * (1 - g(x)) & | \quad g(x) = \frac{1}{1+\exp(-x)} \\ 1 - g(x)^2 & | \quad g(x) = \tanh(x) \\ \frac{1}{2}(1 - g(x)^2) & | \quad g(x) = \frac{2}{1+\exp(-x)} - 1 \end{cases}$$

Az első függvény **unipoláris sigmoid**, a másik kettő **bipoláris sigmoid**. Általában az elsőre mondjuk, hogy **sigmoid**.

Bizonyítás: [4]-ben megtalálható.

Többrétegű előrecsatolt háló. A neurális hálózatokat nem csak az aktivációs függvényük és súlyaik határozzák meg alapvetően, de a háló struktúrájának is fontos szerepe van. Az **előrecsatolt hálókbán** az információ mindig egy irányban terjed. Egy hálón belül a neuronok egy csoportját **rétegeknek** nevezzük, ha a csoporton belül a neuronok nincsenek kapcsolatban egymással. **Rétegelt** a háló, ha diszjunkt rétegekre osztható oly módon, hogy a rétegek beszámozhatók, és az i . réteg összes neuronja összeköttetésben áll az $(i + 1)$. réteg összes neuronjával, és más neuronokkal nem. A legelső réteget **bemeneti**, a legutolsót **kimeneti** rétegek hívjuk. Akkor **többrétegű** a háló, ha e kettőn kívül más rétege is van. Egy ilyen köztes réteget **rejtett rétegeknek** nevezünk, a benne található neuronok neve: **rejtett neuron**. Azon adatok vektorát, melyet a bemeneti rétegek továbbítunk, a háló **bemenetének** nevezzük. Egy bemenet esetén a kimeneti réteg neuronjainak számításaiból álló vektort a háló **kimenetének** nevezzük.

[2][4] szerint a többrétegű előrecsatolt neurális hálóak a bemenet bármely folytonos függvényét képesek approximálni tetszőleges pontossággal:

2.2. Tanítás

Most az előrecsatolt, rétegekbe szervezett hálóak „tanításáról” lesz szó. Jelölje L_i az i . réteget, ahol, ha M réteg van, $i \in \{0 \dots M - 1\}$; az i . rétegben található neuronok számát n_i . Jelölje továbbá $W_{j,k}^i$ azt a súlyt, amely az $(i - 1)$. réteg j . neuronját az i . réteg k . neuronjával összekötő kapcsolathoz tartozik (Itt $M > i \geq 1$, $n_i \geq k \geq 1$ és $n_{i-1} \geq j \geq 1$). Legyen továbbá $W_{0,k}^i$ az i . rétegbeli k . neuron **torzítása**. Tegyük még hozzá, hogy minden i . rétegre x_0^i legyen 1. Látni fogjuk, hogy a jelölés helyes, mert ugyanúgy kell vele számolni, mint a súlyokkal.

A tanításon ún. felügyelt tanítást értünk: adottak az $\langle \bar{v}^n, \bar{y}^n \rangle$ minta párok, melyek azt mondják meg, hogy ha a bemenet \bar{v}^n , akkor a kimenet „közel” \bar{y}^n . Itt az \bar{v}^n -ek n_0 dimenziós vektorok, az \bar{y}^n -ek pedig n_{M-1} dimenziósak. Legyen N darab ilyen párunk. Minimalizálni kell az $E(\mathbf{W}) \doteq \frac{1}{2 * N} \sum_{k=1}^{n_{M-1}} \sum_{n=1}^N (y_k^n - O_k(n))^2$ hibafüggvényt, ahol $O_k(n)$ a háló kimenetének a k . eleme az \bar{v}^n bemenetre, \mathbf{W} pedig a súlyokból álló mátrix (ez explicit soha sem jelenik meg, hanem úgy kell érteni, hogy E minden súlytól

függ). Ehhez egy iterációs módszert használunk, a gradiens módszert. A szabály az, hogy ha létezik egy adott helyen a függvénynek gradiens vektora, akkor ezzel ellentétes irányban kell keresnünk a minimumhelyet. $\Delta W_{j,k}^i = -\eta \frac{\partial E}{\partial W_{j,k}^i}$ Az η a gradiens módszerben konstans, értéke egy pozitív, egynél kisebb valós szám. A parciálisok kibontása után kapjuk az ún. **delta szabályt**: $\Delta W_{j,k}^i = \eta \delta_k^i x_j^{i-1}$. Az n. példa esetén:

$$\delta_k^i \doteq \begin{cases} g'(b_k^i)(y_k^n - x_k^i), & \text{a kimeneti rétegre,} \\ g'(b_k^i) \sum_{n=1}^{n_{i+1}} \delta_n^{i+1} W_{k,n}^{i+1}, & \text{különben.} \end{cases}$$

Itt az y_k^n a tanítóhalmazbeli y_k^n , b_k^i az i . réteg k . neuronjának bemenete, x_k^i pedig az i .réteg k . neuronjának a kimenete (számítása). A g az aktivációs függvény.

Bizonyítás: [4]-ben megtalálható.

A delta szabályból következik az ún. **batched back propagation** módszer. A módszer szerint ciklusban változtatjuk az összes súlyt a delta szabály szerint. Ez nagy tanítóhalmazra nem jó, mivel az összhibát csökkenti. Helyette a **standard back propagation** ajánlott, ekkor a tanítóhalmaz elemeire külön-külön használjuk az eljárást.

A gradiens módszer könnyen belefuthat egy lokális minimumba, melyből a standard eljárás nem is tud kijutni. Az η segítségével tudunk ezen segíteni; ha az η -t nem konstansnak vesszük, hanem változtatjuk a következők szerint:

1. Mivel az η a sebességet is befolyásolja, legyen η arányos az E^2 -tel. Így, ha távol van a rendszer a minimumhelyétől, onnan gyorsan fog távolodni.
2. Ha $\eta > 1$ akkor $\eta := 1$.
3. Ha $\eta < \epsilon$ és a rendszer még nem jól azonosít, $\eta := 1$. Így teszünk egy (vagy több) nagy lépést kifelé a lokális minimumból.

2.3. A háló tárolása

A háló tárolása (szerializálás) valamilyen perzisztens tárolón, tulajdonképpen egyértelmű. Ha a hálót Java nyelven valósítjuk meg, használhatjuk a nyelv által kínált szerializációs módszert, a **java.io.Serializable** interfészt [3] szerint, melyet azonban az igényeinknek megfelelően kell kialakítani, mert az egész háló-objektum „elég” nagy. Az újratartás érdekében a normált képeket is el kell tárolni a háló mellett.¹

2.4. Implementáció

Az implementáció, hogy a kellő sebességet elérjük, és a webes előnyök is megmaradjanak, C++ és Java² nyelven javasolt. A súlyokat rétegenként egy-egy tömb reprezentálja, az aktivációs függvényt bele kell építeni a programba. **Ne használjuk ki az OO programozás természetesen adta előnyeit, bármennyire is csábító.** Ezzel jelentős sebességjavulást érünk el.

¹ Az RBF hálókat nem kell újratartani, de gyengébb teljesítményt mutatnak [6][9]

² Esetleg a Microsoft által javasolt .NET architektúrán alapuló C#, J#,C++,.NET és társai is szóba jöhetnek.

3. Képek kezelése

3.1. Beolvasás

Amikor egy felhasználó szeretne bejelentkezni egy olyan web-helyre, mely támogatja az arckép alapján történő bejelentkezést, a böngészőben elindul egy webes program. Ez a program elindítja a felhasználó web-kameráját (engedély kell hozzá), és a kamera által felvett minden 10. képet továbbítja a szerver felé. Így a rendszernek több lehetősége is lesz az azonosításra.

3.2. Normálás

A normálás az a művelet, amely egy képből a háló számára olvasható bemenetet ad. Tegyük fel, hogy a képet a normálást elvégző alrendszer $N * M$ -es felbontásban kapja, szűrkeskálás reprezentációban. Ebből pl. a Java nyelv PixelGrabber osztályával [3] pixelenkénti információkat kaphatunk. Ezután megkeresük az arc kontúrjait. Ezt segíthetjük pl. hisztogram kiegyenlítéssel. A kontúrkövetés homogén háttér esetén elég egyszerű eljárás, néhány feltételezésre van szükség: mintát kell venni az arc közepéből, olyan feltevéssel, hogy ez egyben a kép közepe is; ez lesz a referencia pixel. Majd, ha pl. a bal szélét keressük, elindulunk a kép bal szélétől oszloponként, és ha egy pixel „sokat” változik a tőle balra lévőhöz képest, és összevetve a referencia pixellel „nem túl” nagy a különbség, valószínűleg megtaláltuk (a kép bal szélét). Ehhez fuzzy logikát használhatunk ([4] leírja, miként lehet modellezni olyan fogalmakat, mint „sok” és „nem túl nagy”). (A módszer nem tökéletes, és inhomogén háttérre nem működik.) Egy másik ötlet a kontúr meghatározására, hogy neurális hálóval végezzük el. Ez a háló az eredeti kép alapján adja vissza a határoló szakaszokat.

Miután megtaláltuk az arc kontúrjait, csak a keletkező $M_1 * N_1$ -es képpel dolgozunk. Ebből egy $32 * 32$ -es képet állítunk elő.

Ezt a $32 * 32$ db. valós számot már közvetlenül is átadhatjuk egy olyan neurális hálónak, melynek a bemeneti rétegében $32 * 32$ db. neuron van, de később leírok egy technikát, mellyel ez 32 -re redukálható, így a háló mérete is jelentősen csökkenthető.

4. Arcfelismerés

4.1. Elmélet

Neurális hálókkal a probléma megoldottnak tekinthető, [6] egy több ezer arcot is 99.5%-ban felismerő rendszerről ír, mely az RBF-háló (Radial Basis Function Network) és az LVQ-háló (Learning Vector Quantization) előnyeit egyesíti.

Az azonosítás valós időben működik. A tanítási idő azonban kritikus pontja a rendszernek, és a háttérben kell folynia.

A neurális hálóval történő felismerés a háló osztályozó képességén alapszik. A kimeneti rétegben, K arc esetén $K+1$ neuron kell, hogy legyen. Az utolsó neuron ³akkor aktiválódik, ha az arcot nem ismeri fel a rendszer. A többi neuron egyértelműen van hozzárendelve egy-egy archoz. A felismerés kritériuma, hogy az archoz tartozó neuron aktivációs értéke „magas” legyen, az összes többié pedig „alacsony”. E két fuzzy halmaz a rendszer megbízhatóságát jelenti. Minél szűkebbek ezek a halmazok, annál biztonságosabb a rendszer. Gyakran használt a $40-20-40$ szabályt, ami az alacsonyt a $[0, 0.4]$, a magast pedig a $[0.6, 1]$ intervallumokkal azonosítja.

³ Ennek a neve általában **trash**

4.2. Receptív mezők helyének megtalálása

Az arcfelismerés, mint általában a képfelismerés alapja abban rejlik, hogy a kép elmozdulása során a kép identitása nem változik. Tegyük fel, hogy minden pixelnek a képen egy bemeneti neuront feleltetünk meg. Az eredmény nagyon rossz lesz, ha a kép elmozdul. Ezen próbáltunk segíteni a kép-normálással is, de az nem járt elég sikerrel. Egy megoldás erre, ha egy neuron egyszerre nem egy pixel adatait kapja meg, hanem egy kis mezőtét. Ha pl. a mező mérete 8×8 pixel, akkor egy bemeneti neuron egyszerre 64 pixelről tartalmaz információt, és invariáns lesz a kismértékű eltolásra vagy elforgatásra. Továbbá vannak „érdektelen” pixelek (pl. a homlok). Tehát nem kell minden 8×8 -as mezőt lefedni. A „jó” helyre elhelyezett mezőket **receptív mezőknek** nevezzük, értékük az alattuk lévő pixel-értékek számtani közepe.

Ezek helyének megtalálására a genetikus algoritmust[2][5] használjuk. Tegyük fel, hogy van néhány (4-5 db.) kiinduló megoldás, melyet véletlenszám-generátor állított elő.⁴ Megmérjük, hogy az adott populáció egyedeivel mennyire hatékony a felismerés (és a tanítás). Minél jobb értéket kapunk, annál nagyobb eséllyel lesz benne az egyed egy köztes populációban.

A köztes populáció elemei közül párokat választunk ki, majd pl. átlagolással új egyedeket készítünk. Az új egyed mezőit „mutálás” címén pl. egy kicsit eltoljuk vagy elforgatjuk. Bár igaz, hogy a populációkkal történő tanulásban minden egyedet ki kell próbálni, és ez nagy tanítási időt jelent, de a receptív mezőkből sokkal kevesebb kell, mint ahány pixel van a képen. Pl. 32×32 pixeles képhez a 100%-os felismeréshez 32 db. mező elég, de 16 db. mezővel is elég jó teljesítményeket lehet elérni.

5. Beléptető rendszerek

5.1. Gondok a jelenleg működő rendszerekkel

Mint azt az előszóban leírtam, a jelenlegi beléptető rendszerek egyetlen karaktersisten azonosítanak egy felhasználót. Elméletileg sokkal több ilyen „elfogadható” karaktersisten van, mint ahány felhasználó. Az emberek mégis szeretnek valamilyen egyszerű szóra asszociálni, mikor egy rendszerbe szeretnének belépni. Úgy értem, sokkal egyszerűbb megjegyezni, hogy az én „accountom” mindenhova pl. „gabor”, mint hogy az egyik helyre „gabor1983”, a másikra „gabor@12#52”. Ráadásul ezeket „el lehet lopni”. Ezért a jelszavakra is egyre nagyobb biztonsági megszorítások vannak. Tehát manapság már nem lehet mindenhol pl. a kutyám nevét jelszónak adni, mert abban nincs sem számjegy, sem speciális karakter.

5.2. Egy megoldási lehetőség

Bontsuk a rendszert tartományokra. Minden tartomány egyedi, és csak kevés (1-2) felhasználó esik bele egy tartományba. A tartományoknak már lehet ilyen azonosítójuk, hogy „jozsef@12”, stb. Mikor egy felhasználó regisztrálja magát, létrejön egy új, saját tartomány, melyben már használhatja a kedvenc becenevét felhasználónévként. A tartomány nevét csak olyan helyen kéri a rendszer, ahol nincs arckép alapján történő beléptető kliens. Ha van, a kliens elküldi a képeket a felhasználóról, a szerver azonosítja a tartományt, majd a belépés megy a megszokott módon. Ha nem tudta azonosítani a rendszert a tartományt, szintén bekéri a tartomány nevét.

⁴ Esetleg itt is használhatunk neurális hálókat, bemenetként az összes képet véve.

5.3. Alkalmazás a gyakorlatban

Szinte minden munka a szervert terheli. A regisztráció során egy webes program küldi a képeket a szervernek; miután az összes megérkezett (kb. 5-6 db.), a többi adattal együtt bekerülnek egy gyűjtőbe. Míg a szerver nem tanulta meg az új adatokat, a felhasználó csak a tartománynév segítségével tud belépni. Mondjuk naponta egyszer vagy kétszer a szerver kibővíti a kimeneti réteget, szükség esetén a rejtett réteget is, és „megtanulja” az új adatokat is. Addig a régi háló egy másolata dolgozik. Mikor egy felhasználó be szeretne jelentkezni, szintén az előbbi program küldi a képeket, addig, míg nem sikerül a beléptetés, és legfeljebb egy bizonyos képszámig. Ha a tartományt továbbra sem tudja azonosítani, bekéri a tartománynevet. Miután sikerült belépni, az új képekkel helyettesíti a régieket. Ha a rendszer hibázik, és rossz tartományba sorol, akkor szintén ugyanez az eljárás. (A rossz tartományba nyilván nem tud belépni a felhasználó.)

A kliensoldalon engedélyezni kell bizonyos dolgokat, pl. hogy a program használhassa a kamerát. A rendszer megvalósítója pl. felteszi, hogy a kliens gépek elég erőteljesek, és a kép normalizálását a programra bízta.⁵

Köszönetnyilvánítás

Ezúton is szeretném megköszönni Dr. Várterész Magdának azt a rengeteg segítséget és törődést, mellyel munkámat segítette.

Hivatkozások

1. Cornelius T. Leondes: Image Processing and Pattern Recognition (vol. 5 of Neural network systems techniques and applications), Academic Press, 1998.
2. Futó Iván (szerk.): Mesterséges Intelligencia, Aula kiadó, 1999.
3. Nyékyné G. Judit: Java 2 utikalauz programozóknak 1.3, ELTE-TTK Hallgatói Alapítvány, 2001.
4. Robert Fullér: Introduction to Neuro-Fuzzy Systems (Advances in Soft Computing), kiadó: Physica-Verlag Heidelberg, 2000.
5. Stuart J. Russel - Peter Norvig: Mesterséges Intelligencia (modern megközelítésben), Panem-Prentice-Hall, 2000.
6. A. S. Tolba - A. N. Abu-Rezq: Combined Classifiers for Invariant Face Recognition, Springer-Verlag, 2000.
7. E. Pessa - P. L. Bandinelli - M. P. Penna: Simulating prosopagnosia through a lesion of lateral connections in a feed-forward neural network, Springer-Verlag, 1999.
8. H. C. Sim - R. I. Damper: A Neural Network Approach to Planar-Object Recognition in 3D Space, Springer-Verlag, 1999.
9. Sushmita Mitra - Jayanta Basak: A Fuzzy Radial Basis Function Network, Springer-Verlag, 2001.

⁵ Míg a kliens esetén ez négy-öt képet jelent, szerver esetén ez egyszerre több száz képet is jelenthet, miközben a processzoridőt esetleg a háló tanítására is használni kellene.

Visualization and Control Interface for MRI Prostate Robot

E. Balogh^{1,2}, A. Tanacs¹, A. Deguet², R.C. Susil³, A. Krieger³,
A. Viswanathan², L.L. Whitcomb⁴, E. Atalar³, and G. Fichtinger^{2,3}

¹ Institute of Informatics, University of Szeged, Szeged, Hungary

² Engineering Research Center, Johns Hopkins University, Baltimore, USA

³ Department of Radiology, Johns Hopkins University, Baltimore, USA

⁴ Department of Mechanical Engineering, Johns Hopkins University, Baltimore, USA

Abstract

We present a software system to visualize and control a surgical robot working inside a conventional closed MRI scanner. The robot translates and rotates inside the rectum and executes precise transrectal needle placement into the prostate for diagnostic, analytical, and therapeutic purposes, under active real-time MRI guidance and monitoring.

The overall system concept was presented in [1], the design of the robot in [2], while the first clinical study on canines in [3]. Here we present the design and implementation of the software that is integrated in 3D Slicer (www.slicer.org), a medical visualization system that provides a unified environment for image guided surgery. The physician selects the target point for the needle insertion and the software calculates the kinematic sequence to bring the needle to the selected position. In 2D, the axial, coronal and sagittal views of the MRI volume are shown, while a 3D view presents the actual position of the robot and the projection of the needle after insertion is presented. The system provides the 3D visualization of the slices and the robot, giving a good appreciation of the current position of the robot with respect to the MRI volume.

References

1. G. Fichtinger, A. Krieger, R.C. Susil, A. Tanacs, L.L. Whitcomb, and E. Atalar. Transrectal prostate biopsy inside closed MRI scanner with remote actuation, under real-time image guidance. In *Fifth International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 2488 of *Lecture Notes in Computer Science*, pages 91–98. Springer Verlag, 2002.
2. A. Krieger, R. C. Susil, A. Tanacs, G. Fichtinger, L. L. Whitcomb, and E. Atalar. An MRI compatible device for MRI guided transrectal prostate biopsy. In *Annual Conference of International Society for Magnetic Resonance in Medicine, 18-24 May 2002 Honolulu, HI, USA*, 2002.
3. R.C. Susil, A. Krieger, J.A. Derbyshire, A. Tanacs, L.L. Whitcomb, E.R. McVeigh, G. Fichtinger, and E. Atalar. A system for MRI guided diagnostic and therapeutic prostate interventions. *Journal of Radiology*, 2003.

Gyors szomszédság kereső algoritmus véletlen rácsú cella automatához

Barkóczy Péter¹

Miskolci Egyetem, Fémtechnológiai Tanszék

¹fembark@uni-miskolc.hu

Kivonat A cella automata egy lehetséges morfológiai modellje az *Avrami* kinetikával jellemezhető szilárd fázisú átalakulási folyamatok eredményezte mikroszerkezeteknek. A véletlenszerű rács alkalmazásával cella automata hatékonysága növelhető. A szomszédság kérdése a véletlen rácsú cella automata módszerben nem egyértelmű. A következőkben az irodalomban használatos szomszédság definíciók közül kiválasztjuk a feladatnak legjobban megfelelő szomszédságot, és egy algoritmust készítünk a kiválasztott szomszédság gyors meghatározására.

1. Bevezetés

1.1. Az *Avrami* kinetikával leírható folyamatok

A szilárd fázisú átalakulások közül a csíráképződéssel és csíranövekedéssel végbemenő folyamatok leírhatók az *Avrami* kinetikával. Legegyszerűbben úgy tekinthetünk ezekre a folyamatokra, mintha a vizsgált síkbeli illetve térbeli tartományban folyamatosan pontok jelennek meg, amelyek növekvő objektumoknak lesznek a középpontjai. Az objektumok a sík illetve a tér egymástól független irányokban adott sebességgel növekednek. Természetesen ha két objektum összeér akkor a növekedésük abban az irányban megáll. Tekintsük azt az esetet amikor a növekedés addig tart, amíg objektumok el nem foglalják a teljes vizsgált tartományt.

1.2. A cella automata

A cella automata módszerben a síkot illetve a teret térkitöltő elemekre, cellákra bontjuk. Meghatározzuk a cellák állapotjelölőit, és az állapotjelölők lehetséges értékei alapján meghatározzuk a cellák lehetséges állapotait. Definiálunk egy f függvényt, amely függő változója a lehetséges állapotok egyike, független

változója a vizsgált cella és a szomszédjainak állapota. Ehhez meg kell állapítanunk a szomszédságot, amely megadja azokat a cellákat amelyek közvetlen hatással vannak a vizsgált cellára. A síkbeli illetve térbeli tartomány határán lévő cellák szomszédsága nem teljes, ezt a problémát kezeli a határfeltétel. Az automata működése során n cellára alkalmazza az f függvényt, hogy meghatározza a következő időlépésben érvényes szimulációs síkot illetve teret [1].

Az *Avrami* kinetikával leírható folyamatok szimulációja egyszerűen megvalósítható cella automata módszerrel [3]. Ebben a megoldásban egy négyzetes rácson működő teljes szabályrendszerű (f függvény) szinkron automatát használnak. A szerzők kimutatják, hogy a növekvő objektumok alakja függ a választott szomszédságtól. Ebből következik, hogy a folyamat végén kialakuló morfológiának is függenie kell a választott szomszédságtól.

Több írásban számolnak be arról az erőfeszítésről, hogy a szomszédságtól függetlenítsék a növekvő objektumok alakját és a kialakuló morfológiát. A módszerek között találjuk a valószínűségi elveket figyelembe vevő szabályrendszert [4], az aszinkron automatát [1] és a véletlenrácsú automatát [5].

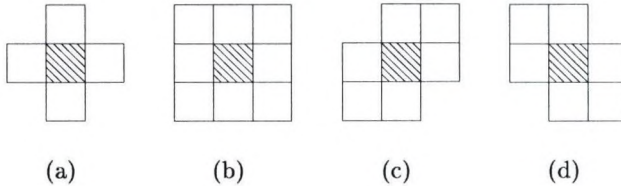
2. A szomszédság meghatározása

A véletlen rácsú cella automatában a rács elkészítésénél a sík illetve a tér vizsgált tartományában véletlenszerűen pontokat helyezünk el. A cellák a pontok által meghatározott *Voronoi* diagram poligonjai illetve poliéderei. A véletlenrácsú cella automatákban a szomszédság meghatározásánál két elvet követnek. Az egyik szerint a szomszédság a vizsgált ponthoz egy adott távolságnál közelebb lévő pontok halmaza [5]. A másik módszer szerint az adott pont szomszédságába csak a *Voronoi* diagramban érintkező poliéderek pontjai tartoznak [2]. A két módszer közül a második alkalmasabb az *Avrami* kinetikával leírható folyamatok során kialakuló morfológia modellezésére.

A *Voronoi* diagram megszerkesztése időigényes algoritmusokkal lehetséges. A véletlenrácsú cella automatában az egyedüli cél a szomszédság meghatározása, más geometria jellemzőre nincs szükség az automata működéséhez, azaz elegendő a *Voronoi* diagram közelítése. Ennek a feladatnak a megoldására már lehet készíteni egyszerű, gyors, algoritmust, akár cella automata alapon is. Nevezzük el ezt a módszert szomszéd kereső automatának.

Legyen a P a pontok halmaza. Mivel semmilyen előfeltétel nem vonatkozik a cellák koordinátaira, legyenek P pontjainak koordinátái egész számok. Ezzel P leképezhetővé válik egy négyszöges cella automata rácsra. A lehetséges állapotok legyenek a 0 és a pontok tárolás szerinti sorszámai. A kiindulási állapotban azon cellák állapota legyen 0 ahová nem esik pont. Ahová pont esik, azon cella állapota egyezzen meg a pont sorszámával. Az f függvényt fogalmazzuk meg úgy, hogy ha egy 0 állapotú cella szomszédságában legalább egy nem 0 állapotú cella van, akkor f független változója legyen a nem 0 állapotú szomszéd állapota. Az f függvényt szinkron módon alkalmazva a cellákra, P pontjaiból kiindul "növekedés" figyelhető meg. Ha megfelelő szomszédságot és határfeltelt választunk, és az automatát addig működtetjük, amíg minden cella állapota

0-tól különböző nem lesz, akkor a vizsgált sík- illetve térfogatrészben az állapot átmenetek kirajzolják a *Voronoi* diagram közelítését. Sorra véve mindegyik átmenetet megmondható P összes pontjának szomszédsága.



1. ábra.

A leggyakrabban használt szomszédságok négyzetes cellageometria esetén: (a) *Neumann* féle, (b) *Moore* féle, (c) és (d) alternáló 7-es.

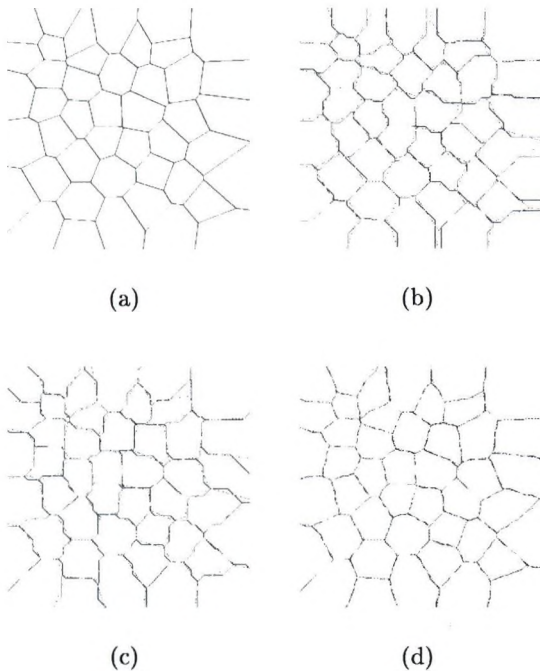
Négyszöges rács esetén három szomszédságot használnak a leggyakrabban (1. ábra) [6]. Egy példa P esetén semleges határfeltétellel [6] elkészítve a *Voronoi* diagram közelítését a 2 ábra eredményeit kapjuk. Látható, hogy az alternáló 7-es szomszédság adja a legjobb közelítést.

A semleges határfeltétel az esetünkben azt jelenti, hogy egy adott pontból induló "növekedés" a vizsgált tartomány határán megáll. Használható még a periodikus határfeltétel, amelyik például síkban egy tóruszá alakítja át a vizsgált tartományt. Ezzel elérhető, hogy az adott pontból kiinduló "növekedés" nem áll meg a tartomány határán, ellenben itt a közelítés elromlik.

A szomszéd kereső algoritmuskak egyik sajátossága, hogy annál gyorsabban megtalálja a szomszédokat, minél több pontot tartalmaz P . Ennek oka, hogy, adott tartományba minél több középpontot helyezünk el, annál kisebb az átlagos távolság két pont között, azaz annál kevesebb lépésben kell "növeszteni" a pontokat. Igaz, hogy minél több pontot helyezünk el az adott tartományon belül annál többféle átmenetet kapunk, ellenben az átmenetek regisztrálásához minden cellát meg kell vizsgálni, így az átmenet keresés és vizsgálat ideje a pontok számától független, csak a vizsgált tartomány méretének függvénye.

3. A modell leírása

Készítsünk el véletlen rácsú cella automatát az *Avrami* kinetikával leírható folyamatok során kialakuló morfológia modellezésére. Nevezzük el ezt az automatát morfológiai automatának. A modellezéshez Ehhez vegyük a P pontalmazt, amelyben a pontok száma az objektumok növekedésének sebességétől függ. A szomszéd kereső automatával meghatározzuk mindegyik pont szomszédságát. Azt hogy a morfológiai automata periodikus vagy semleges határfeltétellel működik, a szomszédkeresés során alkalmazott határfeltétel adja meg. Ha a szomszéd keresés során semleges határfeltételt alkalmazunk, akkor csak tényleg az



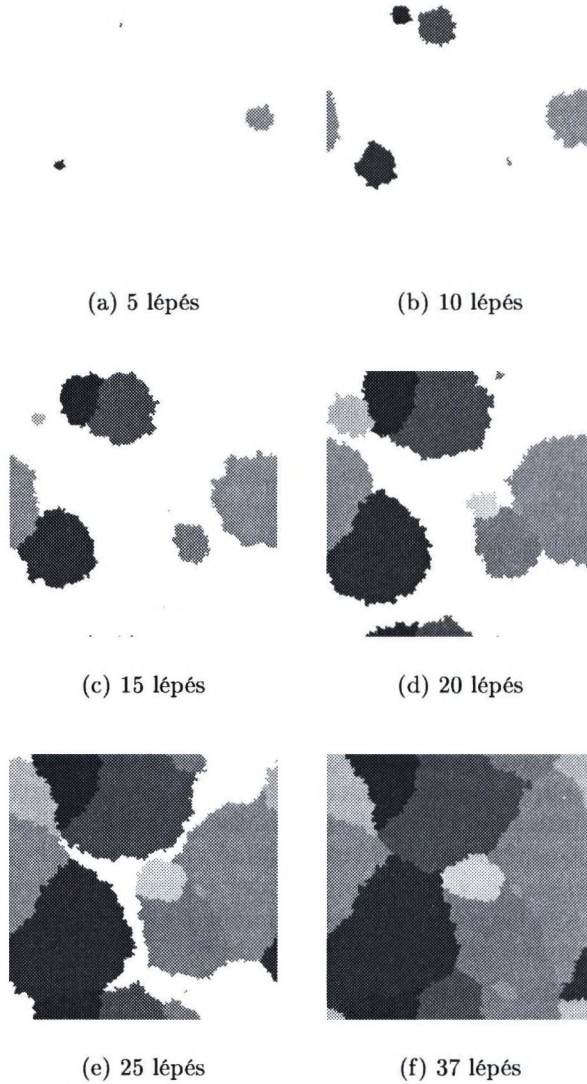
2. ábra.

A Voronoi diagram (a) közelítése Neumann (b), Moore (c), és alternáló 7-es (d) szomszédságok figyelembevételével.

Érintkező poligonok illetve poliéderek alapján állapítja meg a szomszédkereső automata a szomszédokat. Ha periodikus határfeltételt alkalmazunk a szomszédok keresése során akkor a szomszédok között szerepelnek majd azon pontok is, amelyek a vizsgált tartomány átellenes oldalához vannak közelebb. A morfológiai automata futása során maga a szomszédság gondoskodik a határfeltételről, külön kezelést nem igényel. A lehetséges állapotok halmaza legyen a természetes számok halmaza, és a tartomány minden cellája kiinduláskor rendelkezzen a 0 állapottal. Két függvényt kell definiálnunk f helyére, egyet a középpontok megjelenésére, egyet az objektumok növekedésére.

A középpontok megjelenésének vizsgálatakor a középpontok megjelenésének üteméből meghatározzuk, hogy az adott lépés alatt hány középpontot lehet elhelyezni a tartományban. A kapott értéknek megfelelő számú pontot kiválasztunk P -ből. Ha a kiválasztott pontnak és minden szomszédjának az állapota 0, akkor az állapota az addig megjelent középpontok számával lesz egyenlő (az aktuálisan vizsgált cellát is beleértve). Ha szomszédságában található nem 0 állapotú, vagy a cella maga nem 0 állapotú, akkor a tartományban változás nem történik.

Az objektumok növekedésénél alkalmazott függvény megegyezik a szomszédkereső automatában használttal. Sorra vesszük P minden elemét. Ha a vizsgált pont által megadott cella 0 állapotú, és van legalább egy nem 0 állapotú szomszédja, akkor a vizsgált cella állapota felveszi a jelölt szomszéd állapotát.



3. ábra. A morfológiai automata működésének eredménye

4. Eredmények

A morfológiai automatát egy 500x500 cella méretű síkbeli tartományon futtatuk, úgy hogy P 10000 pontot tartalmazott, és minden lépésben egy pont válhatott középponttá. A szomszéd kereső automatában periodikus határfeltételt használtunk. Az eredményül kapott képek a 3. ábrán láthatók.

Hivatkozások

1. B. Schönfisch, BioSystems, Vol. 41, pp. 29-41, 1997
2. A. Flache - R. Hegselmann, Journal of Artificial Societies and Social Simulation (www.soc.surrey.ac.uk/JASSS/4/4/6.html), Vol 4, No. 4, 2001
3. H. W. Hesselbarth - I. R. Göbel, Acta Metall., Vol. 39, No. 9, pp 2135-2143, 1991.
4. C. H. J. Davies, SRIPTA Mater., Vol. 36, No. 1, pp. 35-40, 1997.
5. K. G. F. Janssens, Modelling Simul. Sci. Eng. Vol. 11, pp. 157-171, 2003
6. R. J. Gaylord - K. Nishidate: Modelling Nature, Cellular Automata Simulations-with Mathematica, Springer, 1989

Személyek követése a járásminta számítógépes analízisével

Benedek Csaba, Szirányi Tamás

MTA SZTAKI Analogikai és Neurális Számítógépek Laboratórium, Budapest
bcsaba@sztaki.hu

Kivonat Munkánk célja a járás, mint biometrikus jellemző különféle paramétereinek mérése és összehasonlítása volt digitális videófelveteleken. Először egy klasszikus módszert alakítottunk át, hogy alkalmas legyen valódi felvételeken megfigyelhető rövid járásciklusok összevetésére is. Később módszert fejlesztettünk ki az emberi testen található jellegzetes izületi pontok megtalálására és követésére.

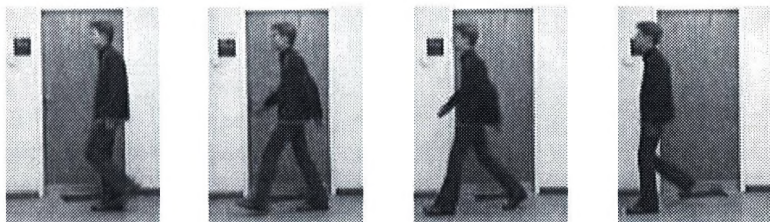
1. Bevezetés

Az intelligens felügyeleti rendszerek egyik központi feladata mozgó személyek észlelése és követése biztonsági kamerák felvételein. A követő algoritmusok nehéz problémája, hogy a célszemély gyakran egy időre a kamerák által láthatatlan helyre kerül, például bemegy egy ajtón, vagy elvegyül egy tömegben, és ismételt felbukkanása esetén fel kell ismerni, hogy az eltűntek közül melyik alakzat jelent meg újra. Célunk ennek a részfeladatnak a kutatása volt.

Az azonosításhoz a személyekről egyedi jellemzőket kell gyűjteni, esetünkben azonban ez egyben többet és kevesebbet is jelent a hagyományos biometrikánál. Többet, mert a személyeket természetes cselekvésük során, általában rövid idő alatt kell megfigyelni, nincs mód beállítani és a kamerához közel hozni őket; kevesebbet, mert nem célunk, hogy módszereink nagy nyilvántartási adatbázisokon hibátlan felismerési eredményekkel működjenek, csupán azt szeretnénk, hogy néhány egyszerre takarásban lévő alakzat esetén biztonsággal elvégezzék az össze-rendelést. A biometrikus jellemzők közül az emberi járás tanulmányozását választottuk.

2. A járásanalízis motivációi és nehézségei

A járásanalízis kezdeti kutatásait orvosi és pszichológiai tudományos eredmények motiválták. [1] felhívta a figyelmet, hogy a járás igen erős azonosító jegy, ami valamennyi paraméterét figyelembe véve minden emberre egyedi. Előnye, hogy olyan helyzetekben is lehetővé teszi a felismerést, amikor más, hagyományos biometrikus jegyek, mint az arc, vagy fül takart helyzetben vannak, vagy éppen az alacsony felbontás miatt elmosódtak. A megfigyelhetősége mellett a járás tanulmányozása mellett szól, hogy szakértők szerint az ember nehezen leplezi mozgásának jellegzetességét.



1. ábra. Négy pillanatkép a járáscikusból

A számítógépes járásanalízis sikeres felhasználásához azonban több nehézséggel is szembe kell néznünk. Az egyediséget biztosító tényezők jelentős része, például medence és a mellkas elfordulásának mértéke és dinamizmusa nagyon nehezen és pontatlanul mérhető, mint ahogy az orvosi kísérletek alapjait jelentő világító markereket sem helyezhetünk el a megfigyelt személyeken. A tudományterületen számos műtermi eredmény született a témával kapcsolatban, ahol egységes háttér előtt, jól megvilágított helyen jó minőségű felvételeken színészi mozgással sétálnak emberek. Ezek az eredmények gyakran nem skálázhatók, valódi környezetben használhatatlanok.

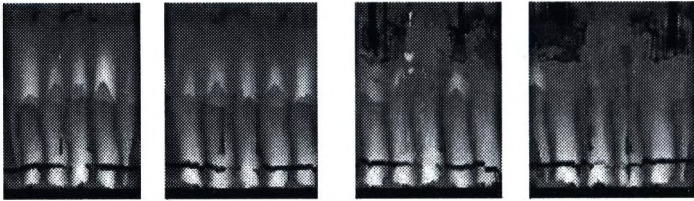
Kísérleteinkben ezért több egyszerűsítő feltevést tettünk. Ilyen feltevés, hogy létezik egy közel állandó, vagy lassan változó, de nem ideálisan egyszínű háttér. Az emberek mellmagasságból, oldalról figyelhetők meg, és a képen nincsenek egymást takaró alakzatok. Nem számíthatunk azonban hosszú periodikus mozgás megfigyelésére, hanem a valóságnak jobban megfelelő rövid, maximum három félcikusból álló járásmintát vizsgáltunk. Ezek a feltevések az általános megoldás valódi megkötései, azonban több gyakorlati szempontból is érdekes helyzetben, például egy bankban teljesülhetnek ezek a feltételek.

A bemutatásra kerülő módszerek első lépése a személyek észlelése után bináris sziluettjük meghatározása. Az apriori feltevések miatt egy adaptív háttérmodellel működő eljárással megfelelő, de korántsem tökéletes árnyképet nyertünk. Elsőként [2] sziluett szélességének változását felhasználó módszerét igazítottunk a saját feltételrendszerünkbe illő környezetbe.

3. Járásminták párosítása a sziluett-lenyomat segítségével

A sziluett-lenyomat névre keresztelt algoritmus alapja az az észrevétel, hogy a sziluett szélességének változása eltérő magasságban eltérő dinamikát mutat. A módszer működése közben minden mintavételi ciklus során eltávolítjuk az árnykép szélességét különböző magasságokban. Ha ennek a vektornak az időbeni változását vizsgáljuk, azt tapasztalhatjuk, hogy a lábak magasságában álló elemek változása jelentős amplitúdójú, közel periodikus mozgást ír le, a fejhez tartozó elemek közel állandóak, míg a kéz környéki szélességváltozás jellege erősen egyénfüggő dinamikát mutat.

Az így nyert jellemzőfolyam szemléletesen ábrázolható grafikusán. A 2. ábra képein az első két, illetve az utolsó két lenyomat tartozik azonos személyhez. Az

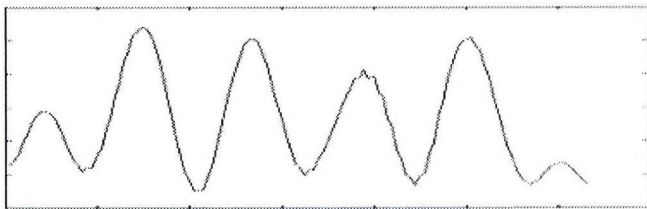


2. ábra.

A sziluett-lenyomatok vizuális megjelenítésben. A vízszintes tengely az idő, a függőleges a sziluetten belüli magasság. Egy pixel értéke annál világosabb, minél szélesebb volt a sziluett a megfelelő koordinátákkal jelzett időben és magasságban.

ábrák magassága megegyezik, mert a sziluettek magassága normált, szélességük azonban különböző, hiszen eltérő mintavételi ciklus alatt tették meg a célszemélyek az előírt utat.

A járás lenyomata tehát egy vektorfolyam, következő lépésként ezeket a folyamatokat kell összehasonlítani. Összehasonlítási módszerként a Dynamic Time Warping algoritmusra esett a választás. Az algoritmus két jelnek úgy határozza meg a távolságát, hogy előbb egy nemlineáris transzformációval megpróbálja optimálisan egymásra illeszteni azokat, majd a transzformált jeleknek kiszámolja az euklideszi távolságát. Működéséből adódóan a DTW képes különböző fázisban lévő periodikus jeleket automatikusan szinkronizálni, azonban az összeillesztés elején illetve végén jelentős hiba jelenik meg. Ez a hiba a vizsgált jelek rövidege miatt megengedhetetlen, ezért külön figyelmet kellett fordítanunk a lenyomatok szinkronizációjára. Az előfeldolgozó lépés célja tehát a lenyomatokból szinkronban lévő és azonos számú lépésciklust tartalmazó részek kivágása volt.



3. ábra. A lábak területét körülfogó téglalap szélességének változása.

A mozgás ritmusát elsősorban a lépés határozza meg, ezért külön vizsgáltuk a test alsó részét, eltárolva itt az árnykép szélességét. A szélesség változását a 3. ábra szemlélteti. Az első, illetve utolsó periódus amplitúdója lényegesen kisebb, ez abból adódik, hogy az alak megjelenésekor és eltűnésekor épp valahol a járás-ciklus közepén tartott. Első feladat tehát az első periódus levágása, mivel ez nem

mozgást, hanem a felbukkanáskor látható járásállapotot jellemzi. Ettől a ponttól számítva kivágható három ciklus. A járás lenyomata ezután a vektorfolyam megfelelő ablakba eső része lett.

A sziluett-nyomat módszer tesztelését három lépcsőben végeztük. Elsőként azt vizsgáltuk, hogy az azonos, illetve különböző személyektől származó járáslenyomatok DTW összehasonlításával milyen eredmények születnek. A kísérlet egy lépése során két személy két-két járáslenyomatát állítottuk párba a lehetséges 6-féle módon, és meghatároztuk a minták DTW távolságát. 9 kísérletéből 6 biztonsággal kimutatta, melyek az összetartozó minták, a többi esetben bizonytalan eredményt adott. Azt a tanulságot vontuk le, hogy a járáslenyomatok összehasonlítása hordoz információt a személyek összerendeléséhez, ennek az információnak a kinyerésére azonban a zajra robusztus és a környezetből származó járulékos információkat kihasználó módszert kell alkalmazni.

A második kísérlet során feltételeztük, hogy a rendszer két ember mozgásmintáját rögzítette, akik a folyosón egyik irányban elhaladtak, majd ellenkező irányban visszatértek. Az előbbi kísérlethez hasonlóan itt is négy mozgásmintát vetettünk össze, jelölje ezeket ODA1, ODA2, VISSZA1, VISSZA2! Kihasználtuk azonban, hogy az azonos személyekhez tartozó minták különböző irányú mozgásból származnak, tehát a 3 megfeleltetésből csak 2 lehetséges: ODA1-VISSZA1 / ODA2-VISSZA2, valamint ODA1-VISSZA2 / ODA2-VISSZA1. Egy összerendelés távolságának a benne szereplő két mintapár távolságának összegét tekintettük, és a két összerendelés közül a kisebb távolságút választottuk. Az előző kísérletsorozat 9 párjára elvégeztük az összerendelő kísérletet, a két párosítás távolságára a 4. ábra táblázatában bemutatott értékek adódtak. Az első párosítások a valóságban összetartozók, a program ezt valamennyi esetben láthatóan jelentős biztonsági tartalékkal határozta meg. Megállapítható, hogy két ember

1. mf:	22.3	18.9	17.9	15.6	16.4	18.7	18.5	16.4	21.1
2. mf:	30.1	30.7	32.2	19.8	27.1	34.1	21.1	23.3	29.6

4. ábra.

Az oszlopok jelölik az egyes kísérleteket, a sorok a kísérleten belüli összerendelések távolságait. Valamennyi kísérletet úgy állítottuk össze, hogy az első megfeleltetés legyen a valódi, a táblázat oszlopaiban valóban az első érték a kisebb.

esetén a megfeleltetés megbízhatóan működik, ha már rendelkezünk a négy járásmintával. A módszer gyakorlati alkalmazása során azonban egyszerre több személy megfigyelése elvárható, valamint fontos, hogy megbízható összerendelést ne csak valamennyi befelé és kifelé irányuló mozgás összegyűjtése után kapjunk, hanem lehetőség szerint valós időben, az egyes emberek vissza irányú mozgásának megfigyelésekor.

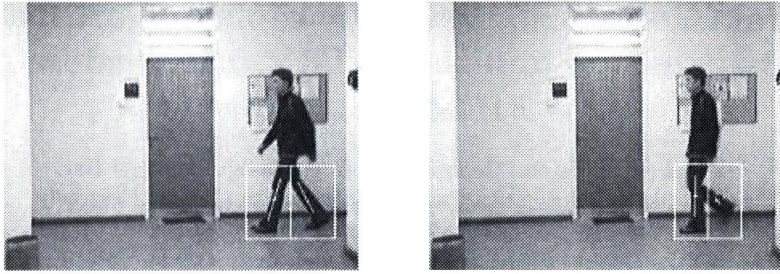
A harmadik kísérletsorozatban implementált megfigyelőrendszer feltételezi, hogy az újonnan megjelenő személyek egy kitüntetett irányból jönnek, és előbb-

utóbb ellenkező irányban ismét elhaladnak a kamera előtt. Ez a helyzet áll elő, amikor a lépcsőházból érkezők a folyosón keresztül bemennek egy terembe, amelynek nincs több kijárata, így várhatóan később kijönnek. Az algoritmus tárolja a teremben tartózkodók járás-lenyomatait, és amikor valaki kijön a teremből, a jellemzőit összeveti a memóriában lévőekkel. Ha a hasonlóság egy tárolt mintával a többihez képest arányaiban jelentős, akkor az első kísérletsorozatra támaszkodva nagy valószínűséggel a kijövő személy párját sikerült megtalálni, így párosítás jelzésével a minta törölhető a teremben lévőek listájából. Ha nem sikerül egyértelműen megtalálni a termet elhagyó személy párját, mert több található nagyságren-dileg azonos távolságra tőle (legyenek ezek az esélyes jelöltek), akkor szintén az első kísérletsorozat alapján kockázatos a legközelebbire dönteni. Az algoritmus ezért ekkor inkább tárolja az összehasonlítások nyert adatokat, bízva abban, hogy az esélyes jelöltek a valódi pár kivételével később kiesnek. Ha kiürül a terem, vagy bizonyos idő eltelik, a függőben lévő bemenő is kimenő párokat az algoritmus megpróbálja optimálisan egymáshoz rendelni. Az implementált prototípus rendszert négy személy együttes e megfigyelésével teszteltük. A párosításokat minden esetben helyesen elvégezte, általában 1 vagy 2 személy párját azonnal megtalálta, a többit csak valamennyi járásminta begyűjtése után.

4. Modell illesztése az sziluettre

A kutatás második szakaszában modell alapú módszerekkel kísérleteztünk. A követett személyekre illesztett modell hat jellegzetes pontból állt: fej, súlypont, a két térd, illetve a két lábfej középpontja. A pontokat illesztő eljárás célja pontok elfogadható elhelyezése lett rosszabb minőségű sziluetten is, és itt nem feltételeztük, hogy akár rövid periódikus mozgást is meg tudnánk figyelni. A súlypont a teljes sziluett súlypontja, míg a fej helyzete normál testtartásban jól közelíthető a sziluett felső nyolcadának súlypontjával. Méréseink alapján ezek stabilnak, tehát járás közben közel egyenes vonalú egyenletes mozgást végzőnek tekinthetők.

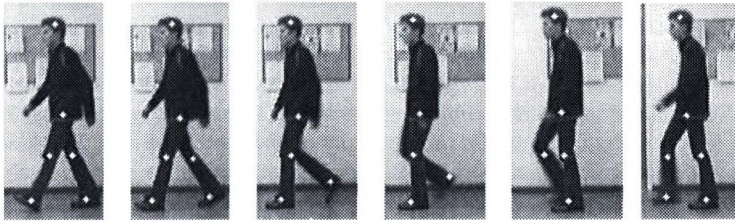
A térdek és a lábfejek helyzetének meghatározásához mindenképp egy-egy pillanatképből meg kell határozni, melyik állapotánál tart a járásperiódus. Ezt segíti PCA-s lábegyenes-detektáló, ami kihasználja, hogy önmagában a láb hosszú és keskeny alakzat, tehát a hozzá tartozó kétdimenziós pontok főkomponens iránya közelíti a láb irányát. Ha a lábhoz tartozó pontok nem határozhatók meg tisztán a felvételtől, mert a lábak takarják egymást, a módszer akkor is hasznos információt szerez a járásállapot meghatározása szempontjából. Az algoritmus a test súlypont alatti részét, tehát a lábak tartományát vizsgálja. A tartományt egy függőleges egyenessel a 5. ábrán látható két téglalap által határolt egyenlő nagyságú részre osztottuk, és a sziluett téglalapba eső részén vizsgáltuk a főkomponens-irányt. Ezt az irányt az adott téglalapba eső sziluett-pontok autokovariancia mátrixának nagyobb sajátértékéhez tartozó sajátvektora jelöli ki. A 5. ábrán két egyenes szakasz jelzi a megtalált láb-irányt, a szakaszok hossza a sajátértékekkel arányos. Látható módon az eljárás nyitott helyzetben jól meghatározza a lábak egyenesét. A második képen látható jellegzetes szituáció, ahol az első láb támaszkodik, a második épp előre lendül, a sajátértékek



5. ábra. A lábak egyeneseinek meghatározása.

arányának figyelembevételével jelezhető. Kísérleteink megmutatták, hogy a fenti módszer megfelelő hangolással valóban alkalmas a két előbbi mozgásállapot megkülönböztetésére.

A járás pillanatnyi állapotának becslésével tesztfelvételeinken jól működő módszer készítettünk a jellegzetes pontok meghatározására. A 6. ábrán a pontok követése látható, további kutatásunk célja, hogy a pontok segítségével újabb egyéni jellemzőket találjunk.



6. ábra. Jellegzetes pontok követése a járás során.

Köszönetnyilvánítás

A szerzők köszönetet mondanak a SZTAKI Analogikai és Neurális Számítógépek Laboratórium munkatársainak, akik készséggel közreműködtek saját járás-adatbázisunk elkészítésében.

Hivatkozások

1. M. P. Murray: Gait as a total pattern of movement, American Journal of Pattern Medicine, Vol 46, No. 1, pp. 290-332, 1967
2. A. Kale, N. Cuntoor, B. Yegnanarayana, A.N. Rajagopalan and R. Chellappa: Gait Analysis for Human Identification, Audio, Video and Biometric Person Authentication, 2003.

Fraktáldimenzió szerepe a képfeldolgozásban

Dr. Berke József

Veszprémi Egyetem, Georgikon, Mezőgazdaságtudományi Kar,
Statisztika és Informatika Tanszék, Informatika Csoport, Keszthely
berke@georgikon.hu

Kivonat A 80-as években nagy reményéseket fűztek olyan matematikai eljárások gyakorlati alkalmazásához, amelyek elsősorban törtdimenziós matematikai konstrukciókra épültek. Kezdetben számos területen (fizika - anyagszerkezet vizsgálat, kaotikus jelenségek szimulációja (földrengés, tornádó), számítástechnika, valós folyamatok modellezése, geológia, folyók, partszakaszok hosszának meghatározása, stb.) születtek eredmények. Később a számítógépes grafika egyes területei, az adattömörítés végül a számítógépes osztályozás során értek el jelentősnek mondható eredményeket az informatikai jellegű gyakorlati alkalmazásokban.

Az előadás során megismerkedhetünk az eddig jól ismert, egyszerű módon fraktáldimenziót számító algoritmus szerzők által kifejlesztett gyakorlati felhasználásával, valamint a fogalom matematikai és gyakorlati kiterjesztésével.

Képtömörítő eljárások pszichovizuális összehasonlítása laboratóriumi körülmények között - MAMIKA

Busznyák János¹, Dr. Berke József¹,

Veszprémi Egyetem, Georgikon, Mezőgazdaságtudományi Kar,
Statisztika és Informatika Tanszék, Informatika Csoport,

¹ Keszthely

8360 Keszthely, Deák F. u. 57.

Tel: 06-83-312-330

berke@georgikon.hu

bjs@georgikon.hu

Internet: www.georgikon.hu/digkep.htm

Kivonat Napjaink interaktív multimédia alapú oktatási és ismeretátadási rendszerei a célcsoportok és az alkalmazott felhasználói felület alapján jelentősen eltérő fejlesztési követelményeket igényelnek, annak ellenére, hogy a megjelenési forma akár teljesen azonos (HTML, DHTML alap) is lehet. A kliens oldali lehetőségek - a felhasználó szemszögéből megítélve - közel egységesítettnek vagy kezelhetőnek tekinthetők. Egy interaktív multimédia alapú CD-ROM vagy DVD-ROM személyi számítógépen, egy internetes hálózati közeg multimédia alapú kliens oldali eszközzel, egy kézi számítógép vagy egy mobil kép- és hangátvitelre alkalmas eszköz napjainkban teljesen eltérő módon, de alkalmas interaktív (multimédia alapú) ismeretátadásra. A tananyagokat fejlesztők célja azonban, hogy ugyanazon ismereteket adják át eszközöktől függetlenül. Előadásunkban, egy ilyen jellegű multifunkcionális (a szó eszköz oldaláról értelmezve) fejlesztés kutatási feladatait és gyakorlati megvalósításának alapjait kívánjuk bemutatni. Ismertetjük - a MAMIKA kutatás-fejlesztési programhoz kapcsolódva - napjaink jelentősebb képtömörítő eljárásainak pszichovizuális összehasonlításának eredményeit.

1. Multimédia alapú, multifunkcionális informatikai oktatási és kutatási anyagok fejlesztése - "MAMIKA" - ITEM-28/2002 - kutatási program céljai

A multifunkcionális informatikai fejlesztések teljes körű elemeinek, szabványainak, leírásainak fejlesztéshez köthető meghatározása és kezelése. Multifunkcionális felépítésű, interaktív (oktatási) anyagok kidolgozása. A kidolgozott tananyagok laboratóriumi körülmények között történő ellenőrzése. Integrált oktatási és ismeretátadási anyagok kivitelezése egyszerű fejlesztői eszközök alkalmazásával. MediaServer alapú információ átadás fejlesztése és gyakorlati alkalmazása. A kutatási program fő támogatói:

- INFORMATIKAI ÉS HÍRKÖZLÉSI MINISZTERIUM
- OKTATÁSI MINISZTERIUM
- SGI MAGYARORSZÁG

1.1. Kutatási területek

Egy interaktív multimédia alapú CD-ROM vagy DVD-ROM, egy Internetes hálózati közeg multimédia alapú kliens oldali eszközzel (asztali gép, laptop), egy kézi számítógép és egy mobil hangátvitelre alkalmas eszköz (mobil telefonok, kommunikátorok) napjainkban teljesen eltérő módon, de alkalmasak interaktív (multimédia alapú) ismeretátadásra. A tananyagokat fejlesztők célja azonban, hogy ugyanazon ismereteket adják át eszköztől függetlenül. Ismereteink szerint egy ilyen jellegű multifunkcionális (a szó eszköz oldaláról értelmezve) fejlesztés gyakorlati megvalósítása a fejlesztők szempontjából számos problémát vet fel. Munkánk e problémák megoldására irányul.

Az eddigi kutatásaink során áttekintettük a szabványok, leírások alkalmazhatóságát mobil, kézi és asztali eszközökre. Elvégeztük napjaink leghatékonyabb tömörítési eljárásaival tömörített képek pszichovizuális összehasonlítását, multifunkcionális fejlesztés elemeinek meghatározását gyakorlati fejlesztői és alkalmazói oldalról. Teszteltük a korábban már általunk elkészített, részben multifunkcionális anyagokat.

1.2. Fejlesztési területek

A kutatási programhoz közvetlenül kapcsolódóan, az alábbi témakörökben multifunkcionális anyagokat fejlesztünk:

- e-Europa
- GPS helymeghatározás, navigáció és adatgyűjtés
- Mobil-kommunikáció
- Webszolgáltatások
- Vizuális informatika napjainkban.

Olyan gyakorlatban kipróbált digitális csomagot kívánunk végezetül létrehozni, amely alkalmas napjaink lehetőségeit, valamint a fejlesztők gyakorlati tudását bővítve, az integrált oktatási és ismeretátadási anyagok kivitelezésére. Tervezett megjelenési formák

- CD-ROM vagy DVD-ROM
- Internetes hálózati közeg multimédia alapú kliens eszközzel (asztali gép, laptop)
- Kézi számítógép (PALM, Pocket PC, ...)
- Mobil telefon

1.3. Multifunkcionális információk szolgáltatása - a MédiaSzerver

A média szerver alapú modern tartalommenedzsment rendszerek korszerű informatikai technológián alapulnak. A médiaszerver a számítógép-hálózat "központjában ül", és több tíz vagy száz felhasználó egyidejű kiszolgálására alkalmas. A legkorszerűbb médiaszerverek a videótartalmat a kódolástól függetlenül olyan digitális adatként kezelik, amely adatbázisban tárolható, kereshető, szokásos számítógép hálózatokon átvihető, de videóként nézhető, vágható és szerkeszthető ("distribute data, view video").

Az ilyen alapon működő archívumban a tartalombevitel (digitalizálás) a videó katalogizálással egyidőben, automatikusan történik. A digitalizálás során a forrásanyag a későbbi felhasználástól függően számos különböző formátumban, párhuzamosan kerülhet fel a médiaszerverre. A formátumok a RealVideo-tól és a QuickTime streaming-től, az MPEG-1 és MPEG-2-n át a ma egyre népszerűbb DVCPRO-ig szinte tetszőlegesen lehetnek, hiszen adatként kezeli őket a rendszer. A katalogizálás eredményeként egy olyan metaadatbázis jön létre, amelyben a felhasználói oldalon hatékonyan lehet keresni, és amelynek segítségével a kiválasztott videórészlet azonnal, streaming video formában, online módon megtekinthető. Az ilyen preview minőségű anyag (kb. VHS minőség) alkalmas a felvételek tartalmi értékelésére, vagy akár vágólista készítésére is.

Sok-felhasználós rendszerben fontos követelmény, hogy a felhasználói oldalon olcsó eszközök legyenek, pl. személyi számítógépek, webes kezelői felülettel. Ez a médiaszerverrel szemben különleges műszaki követelményeket támaszt. A médiaszerverek legfontosabb műszaki jellemzője a nagy belső I/O sávszélesség, a file-rendszerből származó garantált bitsebesség a videostream-ek számára és az igényeknek megfelelő skálázhatóság. Tekintettel arra, hogy egyórányi, "jóminőségű"; videóanyag helyfoglalása 1-25+ GB is lehet (a minőségtől függően), a tartalom tárolása adatbázisban történik, akár többlépcsős, hierarchikus háttértároló rendszeren.

A médiaarchívumok, a televíziók hírszerkesztőségei és archívumai az elsőszámú hasznélvezői az ilyen rendszereknek. A médiaszerver alapú modern tartalommenedzsment rendszer fő tulajdonságai:

- videó katalogizálás (automatikus, on-line, real-time digitalizálással)
- digitalizálás (többféle formátumban, a katalogizálással egyidőben)
- tárolás (digitálisan, médiaszerveren, hierarchikus háttértárolón)
- keresés (PC-n, webes kezelői felületen)
- megtekintés (a hálózaton keresztül, streaming video formában)
- kiválasztás (a találati listából, preview alapján)
- felhasználás (vágólista készítés a preview alapján).

Media Streaming vállalati és Internet alkalmazások számára kliens/szerver:

- Windows95/98/2000, Windows NT, MacOS és Unix kliensek
- Streaming formátumok: MPEG-1, MPEG-2, MP3, RealMedia és QuickTime
- Beépített Informix Online Dynamic Server metaadatok tárolására
- Web-alapú monitoring és menedzsment rendszer

- IP és ATM támogatás
- 28.8Kb/s-től 15Mb/s-ig tetszőleges streaming sávszélesség
- skálázható architektúra akár több tízezer független videó kijátszására.

1.4. A teszt során alkalmazott mobil eszközök

Nokia 3650

A teszt szempontjából legfontosabb funkciók:

Symbian operációs rendszer

Nagyfelbontású, színes kijelző (4096 szín), 176 x 208 képpont

RealOne player; formátumok: H.263 és MPEG-4 video (3GP formátumban);
RealAudio és RealVideo RealMedia formátumban

Compaq iPAQ Pocket PC H3870

A teszt szempontjából legfontosabb funkciók:

Microsoft Pocket PC 2002 op. rendszer

64K TFT LCD, 240 x 320 képpont

Windows Media Player 8.5

2. Pszichovizuális felmérés eredményei

2.1. Pszichovizuális teszt előzményei

1997 és 2001 között végfelhasználói szempontokat is alapul véve, pszichovizuális összehasonlító vizsgálatot végeztünk két veszteséges /JPEG és FIF/ tömörítő eljárás egymás közötti és tömörítetlen képekkel történő összehasonlítására. A méréseket laboratóriumi körülmények között és Internetes kérdőívek alapján végeztük. Az ezzel kapcsolatos főbb eredmények az [2] és [3] publikációkban kerültek közlésre.

A laborbeli vizsgálatok több mint 100 fővel, az Internet vizsgálatok közel 200 fő eredményeinek értékelésével történt. Az Internetes kérdőív jelenleg is elérhető és kitölthető a <http://www.georgikon.hu/visual.htm> webcímen. A laborbeli méréseknél a résztvevők közel 80

A tömörítetlen és a JPEG tömörített képek laboratóriumi körülmények között történő összehasonlításának eredményei alapján megállapítható, hogy szinte minden feltett kérdés esetén a felhasználók jelentősnek, azaz zavarónak ítélték a JPEG 30:1 arányban tömörített képek és a tömörítetlen képek közötti eltérést.

Különösen zavaró / idegen / volt szinte mindenki számára, a képek nagytá-
sakor jelentkező "digitális" hatás. Ugyanezen kérdésekre adott válaszok során a felhasználók laboratóriumi körülmények között a fraktál tömörített és a tömörítetlen képek esetén észrevehetőnek, néhány esetben zavarónak ítélték az eltérést. Az emberi agy számára idegen "digitális" hatás nem volt érezhető a felhasználók között.

A két tömörítő eljárás összehasonlításakor egyértelmű (szignifikáns) különbség mutatkozott a fraktál tömörített képek javára. Ez elsősorban a színárnyalatok visszaadása során volt feltűnően érezhető.

2.2. Vizsgálat célja, tárgya, eszközei

Az 1997-2001 során elvégzett és kiértékelt, illetve a mai napig élő, kitölthető Internetes felmérés eredményeit figyelembe véve készítettük el újabb kérdőívünket. Különböző tömörítési eljárásokkal (Fraktál, MrSID Wavelet, JPEG Wavelet) tömörített képeket hasonlítottunk össze ezúttal is egymással és tömörítetlen TIFF képekkel. A felhasználásra került képek Canon EOS D30 típusú digitális fényképezőgéppel készült, 3*10 bites, 2160x1440 pixel2 méretű, veszteségmentes (RAW) felvételek voltak. A felvételek előfeldolgozás /látványhoz igazítás/ után kerültek felhasználásra. A felmérés technikai részletei:

A megjelentésre Photoshop 6 szoftvert használtunk. A képeket automatikusan töltöttük be standard elrendezésben. 24 bites TIFF, Fraktál Lizardtech: GF Print Pro 2.5, MrSID DLL, JPG2000 Algo Vision LuratTech 7.4.262.0 Plug-In-ek segítségével.

A teszt kitöltői minden egyes alkalommal / előképzettségtől függetlenül/ ugyanazt az ismertetést / szóban és írásban is/ kapták a képek nagyítási lehetőségeiről, a kérdések értelmezéséről, illetve ötleteket kaptak, hogy a képek mely területeit érdemes a nagyítással kapcsolatos kérdések megválaszolásához vizsgálni. A felmérést egy húsz /napjaink technikai színvonalát képviselő/ számítógépet tartalmazó oktatóteremben végeztük. A terem kellő mértékű elsötétítésére is lehetőségünk nyílt.

Hardver: IBM PC, 1.6 GHz INTEL processzor, 256MB memória,
NVIDIA Vanta 16MB videokártya,
IBM 6632-67 N/C 17"-os monitor

Szoftver: Windows XP Professional
Adobe Photoshop 6.0.1 CE angol verziója

Beállítások:

Képernyő: 1152x864 pixel2 képernyőfelbontás, 32bit színmélység, 75Hz frissítési frekvencia. A teszt időtartama 60 perc volt. Ez kérdésenként 6 perc, kérdéscsoportonként 20 perc időtartamot biztosított a válaszadásra.

A teszt kitöltése folyamatos. 2004. január 16-ig 116 fő /hallgató/ töltötte ki. A mintát mindenképpen tovább kívánjuk emelni úgy, hogy minden korosztály valamint informatikai tudásszint képviselve legyen. Vizualis teszt kérdései:

1. Mekkora az etalon képhez (1. kép) viszonyítva a színárnyalatok eltérése? (nincs) (alig van) (minimális) (jelentős) (nagy)
2. Milyen mértékben változtak meg a kép színárnyalatai az etalon képhez viszonyítva 4x nagyítás hatására?
3. Mekkora az etalon képhez viszonyítva a folytonos színárnyalatoktól való eltérés mértéke?

4. Mekkora az etalon képhez viszonyítva az eltérés a képek apró részleteinek láthatósága között ?
5. Mekkora az etalon képhez viszonyítva a geometriai alakzatok torzulása?
6. Rakja sorba a képeket az alapján, hogy milyen a képek apró részleteinek láthatósága a 8x nagyítás hatására? Kezdje a legjobb képpel!
7. Rakja sorba a képeket az alapján, hogy milyen a képek apró részleteinek láthatósága 16x nagyítás hatására? Kezdje a legjobb képpel!
8. Rakja sorba a képeket az alapján, hogy milyen a képek alakzatainak felismerhetősége! Kezdje a legjobb képpel!
9. Mekkora az etalon képhez viszonyítva a szintelitettségek változása?
10. Milyen mértékben változott meg az etalon képhez viszonyítva a kép szintelitettsége 8x nagyítás hatására?

3. Eredmények, következtetések

Eddigi mérési eredményeink alátámasztják előzetes feltételezésünket, hogy nagyon nehéz lesz markáns különbségeket találni az eljárások között, akár egymáshoz, akár az etalon TIFF képhez viszonyítva. A tömörítő eljárások nagyon gyors fejlődése látványosan megmutatkozik az eredményekben.

A kettes kérdéscsoportra (4.-7. kérdések) adott válaszoknál nincs szignifikáns különbség, bár a legjobbnak a JPEG2000 tűnik, különösen nagyítva. Érdekeségeket találhatunk a diagramm 2., 3., és 5.-ként ábrázolt kérdéseinél. Ezeknél nem a TIFF képhez hasonlítottuk a többit, hanem a négy képet egymáshoz. A TIFF kapott 2.1érték (az 1 helyett) jelzi, hogy néhány esetben még itt sem tudtak a teszt kitöltésében részt vevők egyértelmű minőségi különbséget felfedezni. Abban az esetben, ha csak azokat a teszteket vesszük figyelembe, ahol mindhárom esetben a TIFF képet választották legjobbnak /alakzatok ábra/, akkor is az előzőekben ismertetett eredményt kapjuk.

A színárnyalatok tekintetében a normál méretnél a fraktál a legjobb, a nagyítást viszont az MrSID viseli a legjobban. A szintelitettségek vizsgálatánál kijelenthető, hogy a MrSID eljárás jobbnak bizonyult a többinél /színábrázolás ábra/. Végezetül az alábbiakban megadunk néhány - a kutatással kapcsolatos - Internetes oldalt:

A MAMIKA kutatási projekt hivatalos oldala:

<http://www.georgikon.hu/digkep/mamika.htm>

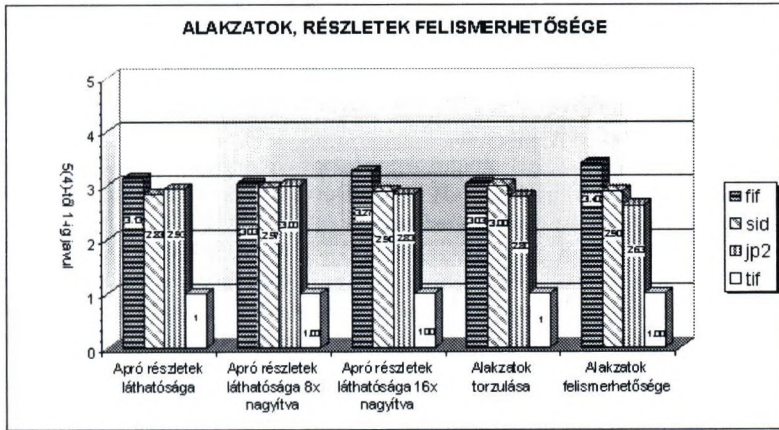
Georgikon mobil médiaszerver /html változat/:

<http://real.silicon.hu/georgikon/index.html>

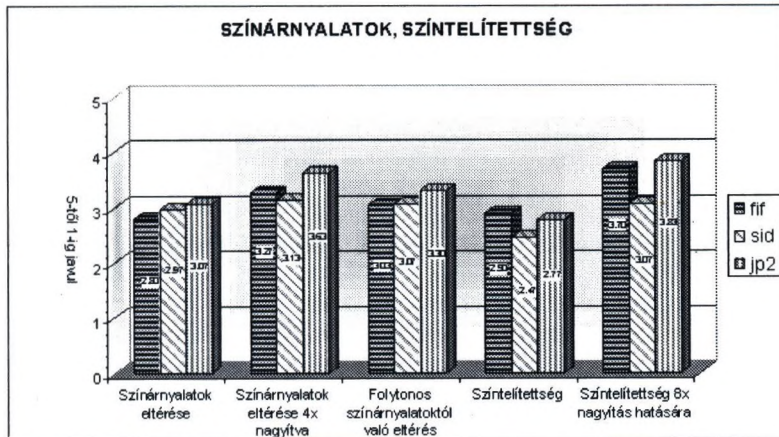
Georgikon mobil médiaszerver /wap változat/:

<http://real.silicon.hu/georgikon/wap/index.wml>

Korábbi pszichovizuális vizsgálatok eredményei:



1. ábra. Alakzatok



2. ábra. Színábrázolás

<http://www.georgikon.hu/digkep/publikaciok.htm>

Korábbi pszichovizuális tesztoldal:

<http://www.georgikon.hu/visual.htm>

Hivatkozások

1. TEMESI, T. (2002): Multimedia knowledge base in the agriculture support systems, Mobile Information Systems in Agriculture'2002, Keszthely, ISBN 963 9495 02 6.

2. BERKE, J. - KOCSIS, P. - KOVÁCS, J. (1997): DCT és fraktál alapú képtömörítő eljárások pszichovizuális összehasonlítása (Psychovisual Comparison of DCT and Fractal Based Image Compressing Methods), Magyar Képfeldolgozók és Alakfelismerők Országos konferenciája (KEPAF Conference on Image Analysis and Pattern Recognition), Keszthely, 131-136.
3. BERKE, J. (1999): Comparison and Application Possibilities of JPEG and Fractal-based Image Compressing Methods in the Development of Multimedia Based Material, IEEE Data Compression Conference - DCC '99, USA.
4. BUSZNYÁK, J. - CSÁK, M. - HEGEDŰS, G. - NAGY, S. - KOVÁCS, E. - BERKE, J. (2002): The integration of research results of Mobile Information Systems into Information Technology instruction at the University of Veszprém Georgikon Faculty of Agriculture, Mobile Information Systems in Agriculture'2002, Keszthely, ISBN 963 9495 02 6.
5. BERKE, J. (2002): Information Technology based Research at University of Veszprém, Georgikon Faculty of Agriculture. IST5 workshop, Pozsony, 2002. January, 30.
6. BUSZNYÁK, J. - CSÁK, M. - HEGEDŰS, G. - NAGY, S. - KOVÁCS, E. - BERKE, J. (2003): The integration of Research Results of Mobile Information Systems into Information Technology Instruction at the University of Veszprém Georgikon Faculty of Agriculture, 9th Conference Information Systems in Agriculture and Forestry, SEC, 2003. March, 11-12.
7. TEMESI, T. - BERKE, J. - NAGY, S. (2003): Médiaszerver alapú multimédia szolgáltatás, "MAMIKA" - ITEM-28/2002, MMO'2003, Pécs.
8. BERKE, J. - BUSZNYÁK, J. (2003): Multimédia alapú, multifunkcionális informatikai oktatási és kutatási anyagok fejlesztése "MAMIKA" - ITEM-28/2002, MMO' 2003, Pécs
9. Lizardtech honlap: <http://www.lizardtech.com>
10. Pszichovizuális tesztoldal: <http://www.georgikon.hu/visual.htm>
11. A MAMIKA kutatási projekt hivatalos oldala:
<http://www.georgikon.hu/digkep/mamika.htm>
12. Médiaszerver leírások: <http://www.silicon.hu-megoldások-média>.

Visszacsatolás alapú képi indexelés

Császár Gergely¹, Czúni László¹,

Veszprémi Egyetem, Képfeldolgozás és Neuroszámítógépek Tanszék

¹ H-8200 Veszprém, Egyetem u. 10

czuni@almos.vein.hu

Kivonat Az cikkben bemutatott képviszakereső rendszer ún. relevancia visszacsatolással próbálkozik a színes képek tartalom alapú keresésével, ahol a képek jellemző színeit egy egyszerűsített EMD (Earth Mover's Distance) algoritmus hasonlítja össze.

1. Bevezetés

Munkánk célja egy a felhasználókhöz igazodó képviszakereső rendszer kifejlesztése, amit egyrészt jól lehet használni otthoni vagy irodai környezetben általános témájú képek visszakeresésére, másrészt felhasználhatjuk színészlelettel kapcsolatos további kísérletekben. A rendszer főbb építőelemei a következők: lekérdező felület, kereső motor, magas dimenziójú indexstruktúra, képi tulajdonságot kivonó alrendszer. A képek indexelésére reprezentatív színeket használunk, amelyek kivonását dinamikus szegmentálással végezzük el a HSV színtérben. Az így kapott színhisztogramok adják a keresés során végzett összehasonlítások alapját: a színvektorokon az ún. EMD (Earth Mover's Distance) algoritmus egy egyszerűsített verzióját használjuk összehasonlító metrikának. Az így kapott indexelő és visszakereső rendszer hatékonyságát fontossági visszacsatolás mechanizmusával tesszük adaptívvá a felhasználó oldalt tekintve.

2. Alkalmazott képi tulajdonság

J. P. Eakins és M. E. Graham [2] osztályozása szerint eljárásunk alacsony szintű képi tulajdonságokat dolgoz fel, egyfajta absztrahálási képességeit visszacsatolási mechanizmusának köszönheti. A tartalom alapú képviszakereső rendszerek szakirodalma a szín, mintázat és forma különböző reprezentációjának ezer és ezer féle kombinációját dolgozza fel, mégis munkákban egy egyszerű tulajdonság, a szín alkalmazása mellett döntöttünk. A színeloszlás kétségtelenül egyik fő előnye robusztussága a különböző torzításokkal szemben (pl.: átméretezés, tükrözés, forgatás) ugyanakkor általánossága miatt célszerű más módszerekkel kombinálni. Dolgozatunkban nem törekszünk ilyen fajta kombinációk vizsgálatára, csupán egy képkocát legjobban leíró színek alapján végezzük el a képi indexelést és visszakeresést.

2.1. Reprezentatív színek kivonása

A képek színeit leggyakrabban a képpontok RGB színtérbeli koordinátájával, ill. az ezekből képzett hisztogramokkal fejezik ki. Azonban ha egy kép színeit próbáljuk felidézni, általában olyan kifejezéseket használunk, mint a sárga, az élénkvrös vagy a sötétkék, nem pedig az RGB modell szerinti vörös, zöld és kék intenzitások mérőszámait. Indokolt tehát a képi index előállításánál egy olyan színteret választani, amely közelebb áll az emberi színészlelet leírásához. Ilyen például a HSV színtér, amelyben a színingert három érték jellemzi: a színezet (hue, H), a telítettség (saturation, S) és a világosság (value, V). Mivel a rendszer által nyilvántartott képek képpontjai - a betöltést végző részrendszer kimeneteként - RGB koordinátákkal vannak megadva, a feldolgozás előtt a színi eloszlást a HSV térbe kell transzformálni. Az átalakítás a következő alakban történik:

$$H_1 = \cos^{-1} \left(\frac{\frac{1}{2} [(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - G)(R - B)}} \right) \quad (1)$$

majd ebből $H = H_1$, ha $B \leq G$, különben $H = 360^\circ - H_1$.

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} \quad (2)$$

ha $\max(R, G, B) \neq 0$, egyébként $S = 0$, valamint:

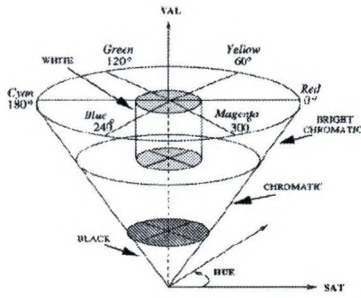
$$V = \frac{\max(R, G, B)}{255} \quad (3)$$

A HSV színtér dinamikus felosztására az Androutsos és mások [1] által javasolt módszert vettük alapul, de a számítások felgyorsítása érdekében az eredeti megoldástól néhány részletben eltértünk. A színtér felosztását a következők szerint végezzük el (lásd 1. ábra):

- Ha a világosság értéke 25
- Ha a világosság értéke 75
- Ha a világosság értéke 75
- Ha a telítettség 15
- Ha a fentiek egyike sem teljesül, akkor a szín a kromatikus tartományba esik.

A fekete, fehér és a szürke tartomány határa előre definiált, képtartalomtól független, ezeket a térrészeket egyszerűen a hozzájuk tartozó képpontok RGB átlagával és a képpontok számának a teljes képmérethez viszonyított arányával (relatív gyakoriságukkal) jellemezzük. A kromatikus rész és a világos kromatikus rész tartalmazza a színi információ nagy részét. Ennek a két térrésznek a felosztását a színezet értékekből készített hisztogramok alapján készítjük el. A hisztogram csúcsai jelölik a jellemző színárnyalatokat, a következő lépés tehát ezek megkeresése.

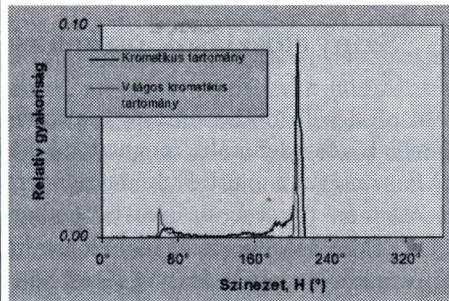
Az alábbi ábra (2. Ábra) egy tipikus képet és a kromatikus régióiból készített színhisztogramokat ábrázolja. Jól láthatók a csúcsok által kijelölt jellemző színárnyalatok: a világos kromatikus részen a sárga és a világoskék, a kromatikus részen pedig a mélykék.



1. ábra. 1. Ábra A HSV tér tartományai [1] szerint



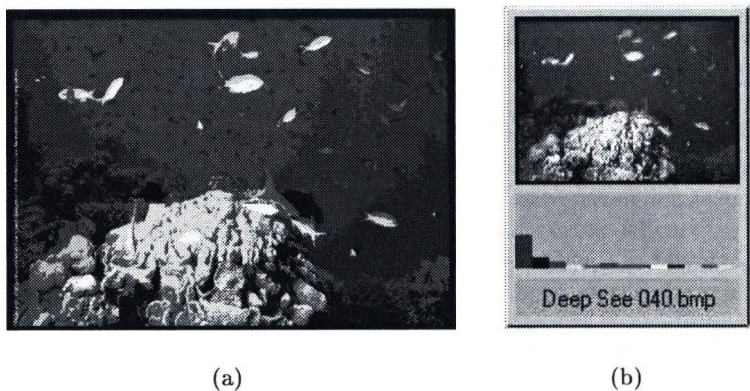
(a)



(b)

2. ábra. Színes kép (a) és a hozzá tartozó színezet-hisztogramok (b)

Eljárásunk egy 30° szélességű átlagoló konvolúciós ablakkal simítja a színezet-hisztogramokat, majd megkeresi a maximumot és annak 30° -os szélességű környezetét elvonja a histograból. Ezt a lépést addig folytatja, amíg elfogy a histogram, miközben sorban feljegyzi az egyes csúcsok jelzett környezetének átlagos színét és a benne előforduló pixelek számát. A 30° -os szélesség elég szűk ahhoz, hogy lényegesen eltérő színárnyalatok semmiképpen ne kerülhessenek egy színcsoportba. Ezzel a módszerrel megkapjuk az HSV színtér két kromatikus régiójának reprezentatív színeit, míg a többi régiót az őket jellemző átlagos színnel és pixelszámmal jellemezzük. Kísérleteink szerint az első 13 legnagyobb elemszámmal bíró, így kiválasztott reprezentatív szín már jól jellemzi a képet és azok területének min. 90-95



3. ábra. Az átfestett kép (a) és a hozzá konstruált színi aláírás (b)

3. Színi eloszlások összehasonlítása

A színi eloszlások hasonlóságának megfogalmazásához szükség van az eloszlás elemei között definiált, megbízható hasonlósági mértékre. Erre a célra a CIE-LAB színtérben mért Euklideszi normát választottuk, mivel e színtér pontjai között mért távolság arányos az emberi színészlelet szerinti különbséggel.

Az eloszlások hasonlóságának meghatározásához még egy egyszerű hisztogram-reprezentáció esetén sem elégséges, ha csak az egymásnak pontosan megfelelő színek arányát vizsgáljuk, hiszen két színi eloszlás akkor is lehet lényegében hasonló, ha egyetlen közös színük sincs. Különösen igaz ez a jelen rendszer alapját képező színi aláírások esetében, ahol a képet jellemző színpontokat a színtér dinamikus, tartalom által vezérelt felosztásával kapjuk. A fenti feltétel figyelembevételével az eloszlások közti távolság mérésére az EMD technikát ([3]) választottuk, amely szerint két eloszlás távolsága arányos azzal az energiával, amellyel az egyik eloszlás a másikba transzformálható. A módszer elnevezése egy szemléletes példából ered: a transzformáció felfogható egy olyan munkavégzésként, amely során különböző nagyságú földhalmokat (azaz: az első eloszlást) kell különböző nagyságú gödrökbe (a második eloszlásba) átvinni a lehető legkevesebb munkával; ahol az átvitelhez szükséges munka az átvitt tömegek és távolságok szorzatának összege. A tömegpontok mérete arányos a színpontokban összpontosuló lefedett képterülettel, távolságuk pedig a színek CIE-LAB térben mért távolsága. Az EMD így egy olyan lineáris programozási feladat megoldásaként áll elő, amely során az eloszlások közti optimális átvitelt keressük. A probléma szállítási feladat néven ismert, megoldására a simplex módszer speciális változatait használják. Az eljárás az optimális megoldást iteratív módon közelíti, ezért nagyon időigényes. Ezt elkerülendő a módszert egy egyszerűbb, mohó algoritmussal helyettesítettük, amely minden lépésben a pillanatnyilag lehetséges, legkisebb távolságon történő átvitelt választja.

4. Fontossági visszacsatolás

Bár a globális színinformáció sok esetben önmagában is elegendő képek, képosztályok azonosítására és visszakeresésére, gyakran előfordul, hogy a képen ábrázolt tartalomnak csak egy része lényeges, így a színi aláírásban felsorolt színek is különböző fontossággal bírnak. Alkalmazásunkban a tulajdonságvektor elemeihez tartozó fontossági arányok megadását egy visszacsatoló rendszer végzi.

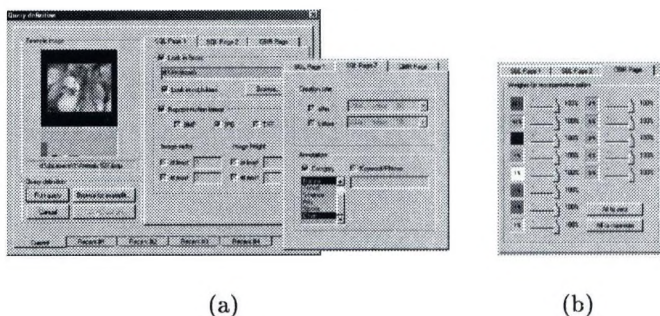
Sajnos az ismert fontossági visszacsatolást megvalósító rendszerek jelenős része az ún. vektor modellen alapszik, ahol a képi indexek "folytonos", magas dimenziójú vektorteret feszítenek ki (tulajdonságtér). Az általunk bemutatott rendszer színi aláírásai nem követik ezt a modellt, hiszen a képi index komponensei nincsenek előre rögzítve, és a távolság kiszámítása sem vektornormával történik. Ezért bevezettük a komponens EMD fogalmát, amely a már ismert módon kiszámított optimális átvitelekhez tartozó energiát nem az egész eloszlásra összegezi, hanem a mintakép aláírásában felsorolt minden színhez külön adja meg.

A visszacsatolás során a felhasználó által relevánsnak, a lekérdezés szempontjából fontosnak ítélt képek halmazán és a teljes képgyűjteményen egyaránt kiszámítjuk a mintaeloszlás színeihez tartozó átlagos komponens EMD értékeket, és az egyes színek fontosságát e két érték hányadosával arányosan változtatjuk. Így azok a színek kapnak nagyobb figyelmet, amelyek a pozitívan elbírált halmazban kicsi, de a teljes képgyűjteményre nézve nagyobb ingadozást mutatnak. Ellenkező esetben a fontosságot csökkentjük, hiszen a szín a pozitívan elbírált halmazban is nagy változatosságot mutat, ezért a keresett képosztályra nem jellemző a jelenléte. Ha a két átlagos komponens EMD hasonló értékű, nem változik a színhez rendelt relevancia-érték. A lekérdezés ismételt végrehajtásánál a teljes eloszlásokra vonatkozó EMD-t a komponens EMD értékek fontosságokkal súlyozott összegeként határozzuk meg, így az irreleváns színekre nagyobb változatosságot engedünk meg, míg fontos színek esetén minél pontosabb megfeleltetést kényszerítünk ki.

5. A kísérleti alkalmazás

Alkalmazásunk Microsoft Windows operációs rendszer alatt futtatható, a képi indexeket tartalmazó adatbázist a Microsoft JET relációs adatbázismotor vezérli, ODBC felületen keresztül. A program két fő komponense a fájlböngésző, amellyel egyrészt a számítógépen tárolt képi állományokat lehet megtekinteni, másrészt az adatbázis karbantartásával kapcsolatos feladatok végezhetőek el; a másik pedig a lekérdező felület, ami a tartalmi keresések eredményének megjelenítését végzi, továbbá a visszacsatoláshoz biztosít felhasználói felületet.

Ez utóbbi egy keresés tartalmi és nem tartalmi specifikációját is meg tudja adni, előbbire a 4/a Ábra mutat példát, míg az utóbbi alapvetően vázlat vagy példakép segítségével lehetséges. A 4/b Ábra azt a felhasználói felületet mutatja be, ahol manuálisan módosíthatjuk egy-egy szín fontosságát, illetve ezen a felületen keresztül szerezhetünk tudomást arról is, hogy egy visszacsatolás alapú lekérdezés hogyan befolyásolta keresésünk paramétereit.



4. ábra.

(a) Egy lekérdezést nem csak a képi tartalom alapján, hanem egyéb metaadatok és fájl adatok alapján is megadhatunk (b) A reprezentatív színek és azok fontosságát beállító dialógus ablak

6. Összegzés, tapasztalatok, további feladatok

Dolgozatunkban egy jellemző színek kivonatolásával működő képviszakereső alkalmazást mutattunk be. Megközelítésünk kicsi számításgéni alkalmazás kifejlesztését tette lehetővé, tapasztalatunk szerint egy kb. 1024x768-as kép indexelése és egy 500 képből álló adatbázisban való keresés is kb. 1-1 másodpercet vesz igénybe egy 450MHz-es PC-n. A visszakeresés hatékonyságának numerikus kiértékelését a jövőben tervezzük, a visszacsatolás alapú lekérdezések nagyjából elvárásaink szerint javítják a hatékonyságot. Szintén a jövő feladatai közé tartozik a területi eloszlás szerinti színi indexelés, ahol terveink szerint az itt ismertetett módszert a kép területi szegmentálásával kombináljuk.

Hivatkozások

1. D. Androustos, K.N. Plataniotis, A. N. Venetsanopoulos (1999). A Novel Vector-Based Approach to Color Image Retrieval Using a Vector Angular Based Distance Measure. *Computer Vision and Image Understanding*, vol. 75, 1-2:46-58
2. J. P. Eakins, M. E. Graham (1999). Content-based Image Retrieval: A report to the JISC Technology Application Programme. <http://www.unn.ac.uk/iidr/research/cbir/report.html>
3. Y. Rubner, L. Guibas, C. Tomasi, "The earth mover's distance, multidimensional scaling, and color-based image retrieval", *ARPA Image Understanding Workshop*, 1997, pp 661-668.

Importance Sampling for X-Ray Volume Rendering

Balázs Csébfalvi and László Szirmay-Kalos

Department of Control Engineering and Information Technology
Technical University of Budapest
{cseb, szirmay}@iit.bme.hu

Abstract. In this paper a novel volume-rendering technique based on Monte Carlo integration is presented. As a result of a preprocessing, a point cloud of random samples is generated using a normalized continuous reconstruction of the volume as a probability density function. This point cloud is projected onto the image plane, and to each pixel an intensity value is assigned which is proportional to the number of samples projected onto the corresponding pixel area. In such a way a simulated X-ray image of the volume can be obtained. Theoretically, for a fixed image resolution, there exists an M number of samples such that the average standard deviation of the estimated pixel intensities is under the level of quantization error regardless of the number of voxels. Therefore Monte Carlo Volume Rendering (MCVR) is mainly proposed to efficiently visualize large volume data sets. Furthermore, network applications are also supported, since the trade-off between image quality and interactivity can be adapted to the bandwidth of the client/server connection by using progressive refinement.

1 Introduction

Due to the dynamic development of modern 3D scanning technologies, like CT or MRI, the size of the acquired data drastically increases. Therefore efficient and meaningful visualization of large data sets is still one of the most important problems in volume rendering research. In a high resolution volume, one single low-opacity voxel, like a drop in the bucket, has a minimal visual contribution. The question is, whether it is really necessary to process every single voxel in the data to compute high precision pixel values, which are finally quantized anyway. In this paper, an alternative solution is searched for, where instead of a regular resampling an importance-sampling strategy is followed in order to explore the relevant information contained in the data.

Exploitation of coherence: Applying the classical volume rendering methods, like ray casting [10, 5], or splatting [16], a volume of size N^3 can be rendered in $O(N^3)$ time, since all the voxels have to be traversed. Many acceleration techniques exploit the data coherence by efficiently skipping the empty regions along the viewing rays [4, 8, 2]. However, this approach is data dependent and does not improve the worst case time complexity. From practical point of view, interactive

rendering of large data sets can be ensured by applying progressive refinement, which is usually based on a hierarchical or multi-resolution decomposition of the volume [11, 9].

Improved time complexity: Another research direction aims at a lower time complexity exploiting a simplified visualization model rather than relying on the data coherence. For example, based on the Fourier projection-slice theorem, a simulated X-ray image of a volume of size N^3 can be rendered in $O(N^2 \log N)$ time [12]. Although the extended version of Fourier Volume Rendering (FVR) supports also shading and depth cueing [14], it is limited to additive compositing and parallel projection. Using FVR for rendering large data sets, it might also be a problem that its memory complexity is $O(N^3)$. Therefore, if the Fourier transform of the volume does not fit into the main memory, the inverse transform of a slice with an arbitrary orientation might not be calculated efficiently enough to achieve interactive frame rates.

Hardware acceleration: Although the time complexity of brute-force volume rendering methods is $O(N^3)$, their hardware supported implementation can significantly reduce the constants in the “ O notation”. One possibility is to exploit the 3D texture mapping capability of the conventional graphics hardware [1, 15]. The main drawback of this technique is that large volumes (more than 256^3 voxels) currently cannot be loaded into the limited texture memory. Recently, a hierarchical wavelet decomposition of the volume was proposed, where only the levels of detail necessary for display are decompressed and sent to the texturing hardware [6]. However, it might happen that the hierarchical representation does not fit even into the main memory, so the performance is drastically reduced because of inefficient swapping.

Taking the above aspects into account, in this paper the strategy of reducing the time complexity as well as the memory complexity is followed. Therefore, similarly to FVR, a simplified optical model is applied.

2 Monte Carlo Volume Rendering

Assume that the input volume is available as a finite number of density samples $f(\mathbf{x}_{i,j,k})$ located at regular grid points. The convolution of this discrete representation with an appropriate reconstruction kernel $h(\mathbf{x})$ results in a continuous 3D density function:

$$f(\mathbf{x}) = \sum_{i,j,k} f(\mathbf{x}_{i,j,k}) \cdot h(\mathbf{x} - \mathbf{x}_{i,j,k}). \quad (1)$$

In classical X-ray volume rendering, a viewing ray is cast through the center of each pixel and the line integral of $f(\mathbf{x})$ is evaluated along the given ray. According to our approach, a pixel intensity $I_{i,j}$ is calculated as a volumetric integral of $f(\mathbf{x})$ over a pyramidal domain $V_{i,j}$ which is projected onto the given pixel (see Figure 1). The calculation of such a volumetric integral can be interpreted as an evaluation of infinitely many rays piercing through the rectangular pixel area.

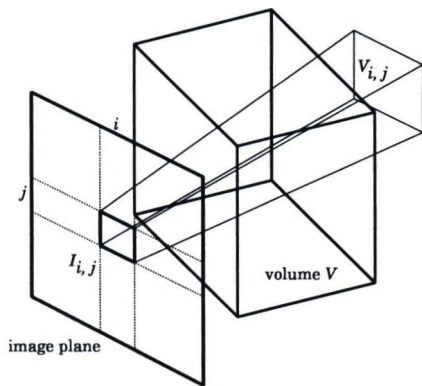


Fig. 1. Calculation of a pixel intensity as a volumetric integral over the corresponding pyramidal domain.

Applying the Monte Carlo method [13], a volumetric integral is formulated as an expected value computation:

$$I = \int g(\mathbf{x}) d\mathbf{x} = \int \frac{g(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x}) d\mathbf{x} = E \left[\frac{g(\mathbf{x})}{p(\mathbf{x})} \right], \quad (2)$$

where $p(\mathbf{x})$ is a probability density function of samples \mathbf{x} . Taking a large number of \mathbf{x}_k samples obtained with density $p(\mathbf{x})$, the expected value is estimated by the following average:

$$I = E \left[\frac{g(\mathbf{x})}{p(\mathbf{x})} \right] \approx \frac{1}{M} \sum_{k=1}^M \frac{g(\mathbf{x}_k)}{p(\mathbf{x}_k)}. \quad (3)$$

The ideal probability density function is proportional to the integrand, and makes the variance of the estimation zero. The computation of this ideal density, however, would require integral I , thus it is not a feasible alternative. Therefore, in a usual application of Monte Carlo integration, the importance-sampling probability density only mimics the integrand, and is not exactly proportional to it. However, there is a class of problems where the application of the "ideal" sampling probability makes sense. Suppose that the integrals of a function need to be computed over many subdomains and the integral is known over the entire domain, that is the union of these subdomains. Thus we are able to construct an "ideal" sampling probability for the entire domain. If this probability density is used for the integrals in the subdomains, the error caused by the variation of the function can be eliminated, only the variation of the subdomain integrals remains.

Using this scenario for volume rendering, a pixel intensity $I_{i,j}$ is calculated as follows:

$$I_{i,j} = \int_{V_{i,j}} f(\mathbf{x}) d\mathbf{x} = \int_V f(\mathbf{x}) v_{i,j}(\mathbf{x}) d\mathbf{x} = E \left[\frac{f(\mathbf{x}) v_{i,j}(\mathbf{x})}{p(\mathbf{x})} \right], \quad (4)$$

where the visibility function $v_{i,j}(\mathbf{x})$ is defined as:

$$v_{i,j}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in V_{i,j} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The visibility function $v_{i,j}(\mathbf{x})$ takes only those \mathbf{x} samples into account which fall into the domain $V_{i,j}$. This function is introduced in order to formulate integral $I_{i,j}$ over the entire volume V rather than over the domain $V_{i,j}$. Since we want to use the same set of random samples \mathbf{x}_k to generate different views of the volume, the probability density function $p(\mathbf{x})$ is defined proportional only to the view-independent term of the integrand, that is the reconstructed density function itself:

$$p(\mathbf{x}) = \frac{f(\mathbf{x})}{\int_V f(\mathbf{x})d\mathbf{x}}. \quad (6)$$

The probability that a random sample \mathbf{x}_k falls into the domain $V_{i,j}$ is expressed as:

$$p_{i,j} = \int_{V_{i,j}} p(\mathbf{x})d\mathbf{x}. \quad (7)$$

Due to the importance sampling, Equation 4 is simplified as follows:

$$I_{i,j} = E \left[\frac{f(\mathbf{x})v_{i,j}(\mathbf{x})}{p(\mathbf{x})} \right] = \int_V f(\mathbf{x})d\mathbf{x} \cdot E[v_{i,j}(\mathbf{x})]. \quad (8)$$

When the integrals are mapped onto pixel intensities, their relative differences are important and not their absolute values. Therefore a normalized intensity $I'_{i,j}$ is assigned to the corresponding pixel rather than the absolute value of integral $I_{i,j}$:

$$I'_{i,j} = \frac{I_{i,j}}{\int_V f(\mathbf{x})d\mathbf{x}} = p_{i,j} = E[v_{i,j}(\mathbf{x})]. \quad (9)$$

Taking M number of \mathbf{x}_k random point samples according to the probability density function $p(\mathbf{x})$, the expected value $E[v_{i,j}(\mathbf{x})]$ can be estimated by an average calculation:

$$E[v_{i,j}(\mathbf{x})] \approx \frac{1}{M} \sum_{k=1}^M v_{i,j}(\mathbf{x}_k) = \frac{M_{i,j}}{M}, \quad (10)$$

where $M_{i,j}$ is the number of those point samples which fall into the domain $V_{i,j}$. In fact, the random variable $M_{i,j}$ takes its value according to a binomial distribution:

$$P[M_{i,j}] = \binom{M}{M_{i,j}} p_{i,j}^{M_{i,j}} (1 - p_{i,j})^{M - M_{i,j}}. \quad (11)$$

Therefore the variance $\text{Var}[M_{i,j}/M]$ of the random variable $M_{i,j}/M$ is derived as follows:

$$\text{Var} \left[\frac{M_{i,j}}{M} \right] = \frac{\text{Var}[M_{i,j}]}{M^2} = \frac{M p_{i,j} (1 - p_{i,j})}{M^2} = \frac{p_{i,j} (1 - p_{i,j})}{M}. \quad (12)$$

Thus, taking M random point samples, the random variable $M_{i,j}/M$ gives an unbiased estimation of the normalized pixel intensity $I'_{i,j}$ with standard deviation $O(M^{-1/2})$.

3 X-Ray Volume Rendering

Applying the results presented in the previous section, an X-ray like volume-rendering algorithm can be implemented in a very simple way. First the input volume is preprocessed and a point cloud of random samples is generated according to the probability density function $p(\mathbf{x})$. Afterwards the point samples are projected onto the image plane. The normalized intensity of each pixel is estimated as the number of samples projected onto the given pixel divided by the number of all the samples in the point cloud.

In order to generate random point samples according to the probability density function $p(\mathbf{x})$, a uniform random distribution has to be transformed into the required distribution [13]. Let us denote the i th voxel in the sequence of voxels represented by the input file by \mathbf{v}_i . Assume that r_1, r_2, \dots, r_M is a sequence of uniformly distributed and independent random numbers in the interval $[0, 1]$. Each random sample \mathbf{x}_k in the point cloud is generated in two steps from the corresponding random number r_k in the following way. First a voxel location $\mathbf{v}_{i(k)}$ is randomly selected if $r_k > F(\mathbf{v}_{i-1})$ and $r_k \leq F(\mathbf{v}_i)$, where the distribution function $F(\mathbf{v}_i)$ is defined as:

$$F(\mathbf{v}_i) = \frac{1}{\sum_n f(\mathbf{v}_n)} \sum_{j=1}^i f(\mathbf{v}_j). \quad (13)$$

In the second step, the reconstruction kernel $h(\mathbf{x})$ (see Equation 1) is used as a local probability density function for generating a random translation vector \mathbf{t}_k , which is added to the voxel location $\mathbf{v}_{i(k)}$ selected in the first step (such an irregular sampling is similar to interleaved sampling [7], which was proposed to reduce inter-pixel aliasing). Thus the k th random sample in the point cloud is calculated as $\mathbf{x}_k = \mathbf{v}_{i(k)} + \mathbf{t}_k$.

In practice, the number of samples in the point cloud is a compromise between image quality and rendering speed, therefore it can be adapted to the available hardware resources. On the one hand, the point cloud has to fit into the main memory to avoid swapping during the rendering. On the other hand, the rendering speed depends on the performance of the processor. The trade-off between image quality and speed, however, can be controlled by progressive refinement. A subset of the point samples can be interactively rotated, and after having the appropriate viewing direction fixed, the rest of the samples are projected onto the image plane progressively refining the estimation.

4 Quantization

A simulated X-ray image of the volume is rendered by quantizing the estimated normalized intensities onto L gray levels provided by the available display device.

For each pixel, the quotient $M_{i,j}/M$ is mapped onto a quantized pixel intensity $Q_{i,j}$ using the following formula:

$$Q_{i,j} = \begin{cases} L - 1 & \text{if } \lfloor L \cdot WH \cdot B \cdot M_{i,j}/M \rfloor \geq L \\ \lfloor L \cdot WH \cdot B \cdot M_{i,j}/M \rfloor & \text{otherwise,} \end{cases} \quad (14)$$

where W and H are the width and height of the image in pixels respectively, while B is the average luminance taken from the interval $[0, 1]$. Such a mapping of the estimated normalized intensities maintains the average pixel brightness in images of different resolutions generated from the same number of random samples.

It is easy to see that the average pixel variance of the generated image is maximum if the projection of the volume is uniformly distributed. In this worst case $p_{i,j} = 1/WH$, therefore the variance $\text{Var}[M_{i,j}/M]$ is derived from Equation 12 as follows:

$$\text{Var} \left[\frac{M_{i,j}}{M} \right] = \frac{1}{M} \left(\frac{1}{WH} - \frac{1}{W^2H^2} \right). \quad (15)$$

The probabilistic error of the image is characterized by the average standard deviation of the quantized pixel intensities $Q_{i,j}$, which can be reduced under the level of quantization error if the following condition is fulfilled:

$$M > \left(\frac{1}{WH} - \frac{1}{W^2H^2} \right) \cdot W^2H^2B^2L^2 \approx WHB^2L^2. \quad (16)$$

5 Implementation

MCVR was implemented in C++ and tested on a 2GHz AMD Athlon XP 2600 PC with 1GB of RAM. The test CT data sets containing a lobster, an engine block, and an Xmastree, represent volumes of small ($120 \times 120 \times 34$), medium ($256 \times 256 \times 110$), and large size ($512 \times 512 \times 999$) respectively. Figure 2 shows the X-ray images generated by MCVR using 1M (a - c) and 16M (d - f) samples. In order to demonstrate that our method does not necessarily require an expensive graphics card to achieve high performance, we implemented it purely in software without exploiting the 3D hardware acceleration. Density fields approximated by 1M and 16M samples were rendered with 21.28 and 1.39 frames/second respectively.

6 Conclusion

In this paper a novel volume-rendering method based on Monte Carlo integration has been presented. This techniques is mainly proposed to interactively visualize large data sets, which do not necessarily fit into the main memory. It has been shown, that the time complexity as well as the memory complexity of MCVR is proportional to the number of pixels and does not depend on the number of voxels. Assuming a fixed image resolution, there exists an M number of samples

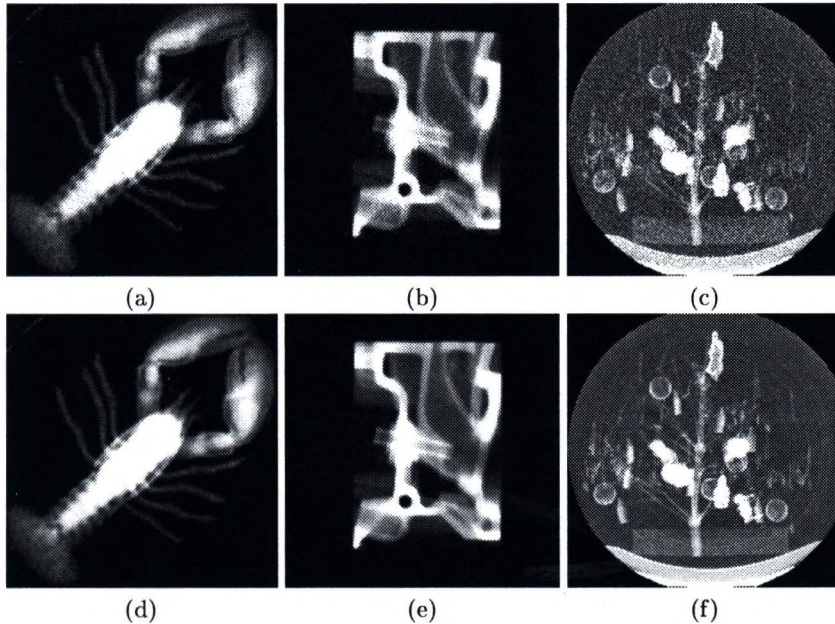


Fig. 2. X-ray images of the test data sets generated by Monte Carlo volume rendering using 1M (a - c) and 16M (d - f) samples.

such that the average standard deviation of the estimated pixel intensities is under the level of quantization error regardless of the size and the content of the data. In this sense, accurate X-ray like images of arbitrarily large volumes can be produced in constant time after the preprocessing. Moreover, our method supports network applications based on progressive refinement.

Acknowledgements

This work has been supported by OTKA 42735, IKTA 00159/2002, and the Slovenian-Hungarian Action Fund.

References

1. B. Cabral, N. Cam, and J. Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *Proceedings of IEEE Symposium on Volume Visualization*, pages 91-98, 1994.
2. D. Cohen and Z. Sheffer. Proximity clouds - an acceleration technique for 3D grid traversal. *The Visual Computer*, Vol.11, No.1, pages 27-38, 1994.
3. B. Csébfalvi and L. Szirmay-Kalos. Monte Carlo volume rendering. In *Proceedings of IEEE Visualization*, pages 449-456, 2003.

4. J. Danskin and P. Hanrahan. Fast algorithms for volume ray tracing. In *Proceedings of Workshop on Volume Visualization*, pages 91–98, 1992.
5. R.A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *Computer Graphics (Proceedings of SIGGRAPH '88)*, pages 65–74, 1988.
6. S. Guthe, M. Wand, J. Gonser, and W. Straßer. Interactive rendering of large volume data sets. In *Proceedings of IEEE Visualization 2002*, pages 45–52, 2002.
7. A. Keller and W. Heidrich. Interleaved sampling. In *Proceedings of EUROGRAPH-ICS Workshop on Rendering*, pages 269–276, 1998.
8. P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Computer Graphics (Proceedings of SIGGRAPH '94)*, pages 451–457, 1994. <http://www-graphics.stanford.edu/papers/shear/>.
9. D. Laur and P. Hanrahan. Hierarchical splatting: A progressive refinement algorithm for volume rendering. *Computer Graphics (Proceedings of SIGGRAPH '91)*, pages 285–288, 1991.
10. M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, Vol.8, No.3, pages 29–37, 1988.
11. M. Levoy. Volume rendering by adaptive refinement. *The Visual Computer*, Vol.6, No.1, pages 2–7, 1990.
12. T. Malzbender. Fourier volume rendering. *ACM Transactions on Graphics*, Vol.12, No.3, pages 233–250, 1993.
13. I. Sobol. *A Primer for the Monte Carlo Method*. CRC Press, 1994.
14. T. Totsuka and M. Levoy. Frequency domain volume rendering. *Computer Graphics (Proceedings of SIGGRAPH '93)*, pages 271–278, 1993. <http://www-graphics.stanford.edu/papers/fvr/>.
15. R. Westermann and T. Ertl. Efficiently using graphics hardware in volume rendering applications. *Computer Graphics (Proceedings of SIGGRAPH '98)*, pages 169–176, 1998.
16. L. Westover. Footprint evaluation for volume rendering. *Computer Graphics (Proceedings of SIGGRAPH '90)*, pages 144–153, 1990.

3D modell készítése ortopédiai műtétek szimulálásához

Csernátony Zoltán¹, Hajdu András², Manó Sándor³, Zörgő Zoltán²

¹ Debreceni Egyetem - Ortopédiai Intézet
H-4012, Debrecen Pf. 16
csz@jaguar.dote.hu

² Debreceni Egyetem - Informatikai Intézet
H-4010 Debrecen Pf. 12
{hajdua|zorgoz}@inf.unideb.hu

³ Debreceni Egyetem - Műszaki Főiskolai Kar
H-4028 Debrecen Ótemető u. 2-4.
molnarsz@jaguar.dote.hu

Kivonat Az orvosi informatika egyik egyre nagyobb kihívásokkal szembeesülő ágazata a műtéti tervezés, a beavatkozások számítógépes szimulációja. Az új műtéti eljárások kidolgozásánál ez a megközelítés különösen nagy előnyt jelent, hiszen az olcsó tesztelés mellett a várható eredmények is megbecsülhetők. A klinikus így egy egyszerű röntgen vagy akár CT felvétel által nyújtottnál sokkal konkrétabb adatokkal rendelkezhet még mielőtt egyetlen szikevágás is történne. Ez a cikk a 2002-es KÉ-PAF3 konferencián bemutatott problémafelvetés és előzetes vizsgálatok óta elért eredményeket tárgyalja. Egyfelől a műtéti szimulációs keretrendszerhez kötődő megvalósításokat – a megjelenítés, kezelőfelület és a végeselemes elemző rendszer összekapcsolásában történt előrelépéseket – másfelől esettanulmányként egy új csontnyújtási eljárás végeselemes vizsgálatát közöljük.

1. Bevezetés

Az orvosi képkalkotást szolgáló technológiák és berendezések, mint a CT, MRI, PET vagy SPECT szelet- vagy vetületi képeket szolgáltatnak. Ezek felbontása, képminősége különböző. Egyesek az emberi test anatómiai aspektusait, mások a test működéséből származó adatokat rögzítik. Ezek a berendezések maguk is biztosítanak némi megjelenítési lehetőséget, de ezek távol vannak attól, hogy további feldolgozásra közvetlenül alkalmasak legyenek. Nem utolsó sorban majdnem minden ilyen berendezés alkalmas képi adatokat és egyéb információkat tartalmazó közismert orvosi fájlformátumokban exportálni. Ilyen formátum például a DICOM [13].

Ezeket az állományokat más gyártók által készített számítógépes programokkal lehet feldolgozni. Sok kereskedelmi és nyílt kódú vagy ingyenes projekt létezik ebben a témában, és ezek kiváló eredményeket értek el a megjelenítésben, egyes szervek vagy szövetstruktúrák felületének rekonstrukciójában, nagyban megkönnyítve ezzel a diagnózist felállító orvosok munkáját (lásd [6,7,11,15]).

Néhány alkalmazás speciális műtéti beavatkozásokhoz nyújt segítséget. Ígéretes eredmények mutatkoznak még a távgyógyászat, az egészségügyi felsőoktatásban alkalmazható virtuális környezetek létrehozása, és a számítógéppel támogatott prototípusgyártás területein is, lásd például [9,14,16].

Ha összekapcsoljuk a felület- és testrekonstrukciós technikákat a mérnöki szoftverek (pl. CAD rendszerek [18]) nyújtotta testmodell-szerkesztő lehetőségekkel és a végelelemes elemzést lehetővé tevő (FEM/FEA) szoftverekkel [17,19], olyan gyakorlati alkalmazásokat hozhatunk létre amelyek alkalmasak például az egyes műtéti beavatkozások hatékonyságát becsülni és értékelni.

Kutatásaink célja az ortopédia területén használható szoftveralkalmazások létrehozása, különös tekintettel a csontműtétekre. Kiinduló adatként a CT berendezés által szolgáltatott képfájlokat használjuk. Egyfelől egy új beavatkozás elméleti vizsgálatát végezzük, másfelől egy eset-orientált, interaktív alkalmazást fejlesztünk a csípőízület hibáit javító műtéti beavatkozások szimulálására és elemzésére. Ez a két irányvonal sok részfeladat tekintetében átfedi egymást. Ilyen például a tulajdonképpeni képadatok kinyerése az egyes orvosi fájlformátumokból, a képek javítása, a megfelelő szegmentációs technika kiválasztása, CAD-orientált testfelépítés, adatcsere a FEM/FEA eszközzel, a virtuális beavatkozás geometriájának megfelelő leírása.

Egy igen fontos, mégis sokszor mellőzött szempont ilyen alkalmazásoknál, annak a figyelembe vétele, hogy nagyon kevés orvos képzett informatikai vagy mérnöki területen. Ezért javasolt a szoftver feladatainak leszűkítése egy jól meghatározott célterületre, sőt egy általános eszközzel szemben előnyösebb egy könnyen bővíthető, de előre rögzített útvonalakat alkalmazó program. Ezért úgy véljük, hogy a végelelemes szoftver felületének közvetlen használata nem lenne megfelelő a virtuális műtétek tervezésére. Tehát vagy létrehozunk speciális munkafelületeket a végelelem rendszeren belül, vagy olyan különálló grafikus alkalmazást készítünk, amely képes kétirányú kommunikációra a FEM/FEA szoftverrel.

2. Modellépítés

2.1. A CT adatfájlok

Ahogy azt fentebb említettük, az első lépés a tulajdonképpeni képadatok kinyerése az orvosi képfájlból. A CT képek képpontjai azt reprezentálják, hogy egy anatómiai keresztmetszet térfogatrészei (voxel) milyen mértékben nyelik el a röntgensugarakat. Így a képpontok a sugárelnyelés mértékének 12 biten ábrázolt egészértékű megfelelőjét tartalmazzák, és nem színinformációkat. Hogy a sugárterhelést a lehető legnagyobb mértékben csökkentsék, élő szervezetek szeknelésénél a szeletek távolságát a lehető legnagyobbra veszik. A legtöbb esetben a szeletvastagság sokkal kisebb, mint a szeletek közötti távolság, így hasznos részletek veszhetnek el, főként az emberi test összetettebb részeinél. Megjelentési célokra a hiányosságokat kijavíthatjuk különböző interpolációs technikák segítségével, de az elvesztett anatómiai részletek nem pótolhatók.

2.2. Képjavitás és szegmentálás

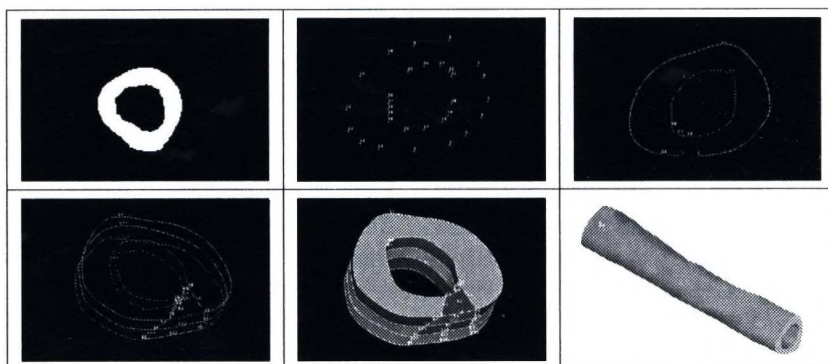
Mivel az egyes szövettípusoknak rájuk jellemző mértékben nyelnek el a röntgensugarakat, az intenzitás-ablak alkalmazása csak egyszerű megjelenítéshez elegendő. Mindazonáltal különböző szövettípusok azonos mértékben nyelhetnek el sugarakat, így az ablak alkalmazása után az általunk választott céltartományon (VOI, Volume Of Interest) kívül más szövetekből is maradhatnak részek. Néhány hagyományos eljárást lehet itt alkalmazni, mint a morfológiai vagy a frekvenciatarományban végzett szűrés [3,8], de léteznek a CT képek szegmentációjára kifejlesztett technikák is, mint például az, amelyről [12] oldalon olvashatunk. Megvizsgáltunk más módszereket is, mint a tartománynövelés és a gradiens alapú szegmentálás [3,8]. Mivel a projektünk a csontszövetekre koncentrálódik, ezek a megközelítések kielégítő eredményekhez vezettek.

2.3. Testmodell építése

A projekt végső célja olyan alkalmazás kifejlesztése, amely napi használatú segéd-eszköze lehet a műtéttervezésben a CAD rendszerekben nem jártas orvosoknak is. Mivel az alkalmazásnak struktúra jellegű adatokat kell átadni egy végeselemes modellező és elemző programnak, a testadatokat geometriailag megfelelő módon kell reprezentálni. Jelentős különbség van az egyszerű felületmegjelenítés és a CAD szemléletű testrekonstrukció között. Az térfogati adatokat merevtest-primitívekkel kell leírni, és az itt használatos elvek alapján összerakni: kulcspontok, parametrikus görbék, vonalsorok, görbült és síkfelületek, térfogati és merev testek.

Amikor a vizsgált területek (ROI, Region Of Interest) kontúrpontjaiból görbét készítünk, csak azokat a pontokat választjuk ki alappontnak, amelyek környezete gyorsan változik. Így átlagosan 10-15%-ra csökkentjük a pontok számát. Az eljárás alapötlete a [3,8] forrásanyagokban található. A görbékből felületeket állítunk össze, a felületekből testeket [5]. A legnagyobb gondot az okozza, hogy a CAD rendszerek szabályos alakú objektumok kezelésére lettek kialakítva, az emberi csontok pedig nem ilyenek. Ahhoz azonban, hogy a végeselemes rendszer lehetőségeit használjuk, ez a megközelítés elkerülhetetlen. Továbbmenvő, a virtuális műtéti beavatkozást is ezen a reprezentáción kell végrehajtani, vagy oly módon, hogy a módosult geometriát könnyű legyen előállítani. Mivel a különböző CAD fájlformátumok közötti átjárás sokszor hibákat eredményez, úgy döntöttünk, hogy a testmodel felépítését a FEM szoftveren belül végezzük el a szoftver által nyújtotta technológiákat felhasználva. Több FEM szoftver tesztelése után mi az ANSYS-t választottuk a feladatra [17]. A 1. ábrán egy 20cm-es combcsont darab (femur diaphysis) geometriája felépítésének lépései láthatók az ANSYS-ban. Az eljárás során egy ANSYS szkript fájlt generálunk, melyet a szoftver végrehajt.

Ezt a módszert könnyen alkalmassá lehet tenni a több darabból álló, vagy több üreget tartalmazó csontszerkezetek felépítésére is. A mi esetünkben az ANSYS testek közötti Boole műveletei igen hatékony eszköznek bizonyultak. A változó topológiájú részek (elágazások a csontban) kezelése komoly kihívást jelent, jelenleg is vizsgáljuk a lehetőségeket.



1. ábra. A combcsont testmodell felépítésének lépései.

3. A virtuális műtét megvalósítása

3.1. Síkbeli megközelítés

Néhány esetben a beavatkozás geometriája lehetővé teszi, hogy azt a kétdimenziós szeleteken hajtsuk végre a testek helyett. A csavarvonalú vágást használó combcsontnyújtás jó lehetőségeket biztosít ilyen kísérletekhez. A javasolt műtéti eljárás elméleti leírását [2] tartalmazza. Tekintve az elméleti vágóeszköz időbeli mozgását egyetlen 3D-s objektumként (2. ábra, bal oldal), majd véve ennek metszetét a CT szeletek síkjával (2. ábra, jobb oldal, sötét rész), a két szintre vágott képeken a metszetnek megfelelő részeket a háttér színére festjük (3. ábra bal oldal).



2. ábra. Az eszköz útjának egy része és annak egy síkmetszete.

Ezt a technikát alkalmazva az előbb vázolt testmodell építés után már a módosított csontstruktúrát kapjuk minden további beavatkozás nélkül. A 3. ábrán az eredményül kapott 3D-s objektumban látszik a vágószerszám síkmetszeteiből összeállított testtel együtt, amelyet egy köztes megtözelítésben használtunk. Minden esetben egy ANSYS szkriptet generáltunk, így azt végrehajtva eredményül a kívánt testet kaptuk.



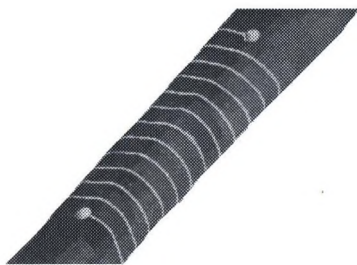
3. ábra. A módosított szeletkép és az eredményül kapott objektum.

A csípőízület-korrektíós műtét esetén is modellezhetők 2D-ben egyes beavatkozások, de a ROI-k kijelölése a szeletsorozaton nehezebb.

Egy jó köztes megoldás lehet a test pontjainak rendezetlen halmazként való kezelése is.

3.2. Háromdimenziós megközelítés

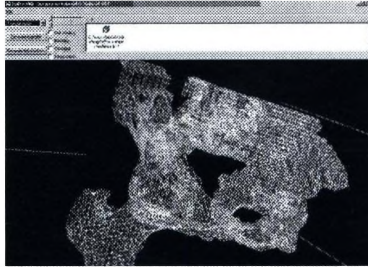
A csontnyújtás szimulálásánál az általunk alkalmazott 3D-s megközelítés a 2.3 alfejezetben leírt modellépítés és közönséges CAD testszerkesztés kombinációja. Először felépítettük a combcsont középső – diafizisnak nevezett – részét, majd ugyanabban a koordináta-rendszerben, a csont tengelyét felhasználva, egy olyan merevtestet készítettünk egy CAD rendszerben, amely a vágószerszám időbeli útjának felel meg. Ez utóbbit úgy ábrázoltuk elő, hogy a megadott tengely mentén csavarvonalú kiterjesztést (helical extrusion) alkalmaztunk egy téglalpra. A kiterjesztés paramétereinek a tervezett csontműtét paramétereit választottuk (vastagság, menetemelkedés). A két merevtest egymásból történt kivonása után a kívánt testet kaptuk (4. ábra).



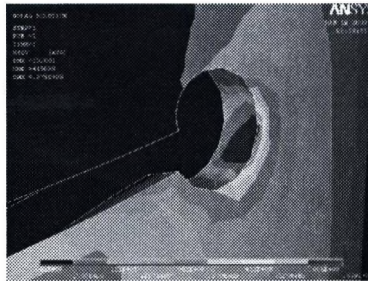
4. ábra. A combcsont modellje a beavatkozás után.

A csípőízület esetén a terület 3D-s rekonstrukciója sokkal bonyolultabb, főleg a CT szeletek távolságából adódó információvesztés miatt. Sok esetben az egymást követő CT szeletek nagyon különböznek egymástól. Itt az interpoláció nem segít, ezért az általános módszereket kell specializálni annak érdekében, hogy a felhasználói beavatkozás szükségessége a lehető legjobban csökkenthető legyen.

A virtuális beavatkozások elvégzésére készített felület GLUT-alapú, és tartalmaz létező térhálók importálására, és a FEM/FEA szoftver felé történő exportálásra alkalmas eszközöket. A már említett szkriptkészítés mellett megpróbáljuk kihasználni az ANSYS által nyújtott térháló kezelési eszközöket is, az adatszererét a térháló állományokra koncentrálva. Így lehetőségünk van például az OpenGL-ben használt tetraéder-hálót beolvasni ANSYS-ba végeelem elemzés céljából és visszafelé a megjelenítés számára (5. ábra).



5. ábra. A virtuális műtétek számára fejlesztett OpenGL felület.



6. ábra. A furat környékén kialakult feszültség.

4. Eredmények

A fentebb bemutatott tesmodell építést használva csupán a belső reprezentáció szkript nyelvre való fordítását kell elvégeznünk. A test anyagparamétereinek megadása szintén beágyazható a szkriptbe. Mi lineáris izotropikus [1] illetve ortotropikus [4] anyagmodelleket használtunk. A kényszerfeltételek és a húzóelmozdulások alkalmazása után megvizsgáltuk a testben fellépő egyenlő feszültséget (von Mises). Azt tapasztaltuk, hogy a legnagyobb feszültség a vágás kezdeti és

végpontjai mellett alakul ki (6. ábra), ezért további vizsgálatokat folytattunk a csatornát záró furatok formája és iránya tekintetében. Jelenleg folynak a nem egyenletes menetemelkedésű csavarpályák vizsgálatai, melyektől a nyújthatóság további javulását várjuk. A további vizsgálatok megkönnyítése érdekében megfelelő vastagfalú csövekkel is végeztünk számításokat és megállapítottuk, hogy a kiszámított értékek nagy mértékben korrelálnak. Az eredményekről részletesebben a [10]-ben olvashatunk.

Hivatkozások

1. H. Abé, K. Hayashi, and M. Sato (Eds.): Data Book on Mechanical Properties of Living Cells, Tissues, and Organs, Springer-Verlag, Tokyo, 1996.
2. Z. Csernátóny, L. Kiss, S. Manó, L. Gáspár and K. Szepesi: Multilevel callus distraction. A novel idea to shorten the lengthening time, Medical Hypotheses, 2002
3. R. C. Gonzalez, and R. E. Woods: Digital image processing, Addison-Wesley, Reading, MA, 1992.
4. R. Klabunde, E. Roland, W. R. Taylor, S. E. Clift, L. Rakotomanana: Experimental validation of an orthotropic FE-model using modal analysis, 12th Conference of the European Society of Biomechanics, Dublin, 2000
5. A.L. Marsan: Solid model construction from 3D iamges,(PDF, PhD dissertation) The University of Michigan, 1999
6. K. Radermacher, C. V. Pichler, S. Fischer, G. Rau: 3D-Visualisation in Surgery, Helmholtz-Institute Aachen, 1998
7. L. A. Ritter, M. A. Liévin, R. B. Sader, H-F. B. Zeilhofer, E. A. Keeve: Fast Generation of 3D Bone Models for Craniofacial Surgical Planning: An Interactive Approach, CARS/Springer, 2002
8. M. Sonka, V. Hlavac, and R. Boyle: Image processing, analysis, and machine vision, Brooks/Cole Publishing Company, Pacific Grove, CA, 1999.
9. Tsai Ming-Dar, Shyan-Bin Jou, Ming-Shium Hsieh: An Orthopedic Virtual Reality Surgical Simulator (PDF), ICAT 2000
10. Zoltán Zörgő, András Hajdu, Sándor Manó, Zoltán Csernátóny and Szabolcs Molnár: Analyzis of a new femur lengthening surgery, IEEE IASTED International Conference on Biomechanics (BioMech 2003) (2003), Rhodes, Greece, Biomechanics/34-38.
11. <http://graphics.stanford.edu/data/3Dscanrep/>
12. <http://image.soongsil.ac.kr/software.html>
13. <http://medical.nema.org>
14. <http://www.ablesw.com/3d-doctor/>
15. <http://www.kanazawa-it.ac.jp/ael/imaging/synapse>
16. <http://www.materialise.com>
17. Ansys Online Manual
18. CADKey Online Manual
19. MSC Patran/Nastran Online Manual

Spatial Image Analysis in the Field of Metallography

Csikós Tímea

GRIMAS Kft., Budapest

csikos@grimas.hu

1 Introduction

Physical imaging methods used for the examination of biological, medical, and materials science probes produce more and more spatial images of structures. The use of computer tomography for non-destructive quality control is just one example.

Besides visualisation and processing (transforming) of these images, there is a growing interest in an analysis of spatial image data. The goal of this analysis is to determine geometric and topological characteristics of microstructures. Examples are volume density and surface density. Moreover, spatial objects are isolated and their characteristics determined.

1.1 Mathematical Background

A new method is presented which is strictly based on integral geometric formulae such as Crofton's intersection formulae and Hadwiger's recursive definition of the Euler number. The corresponding algorithms have clear advantages over other techniques.

Consider a component of a microstructure which is modeled as a macroscopically homogeneous random set i.e. the distribution of χ is assumed to be invariant with respect to translations. The homogeneity of χ allows us to introduce so called basic geometric characteristics the volume density the surface density the specific integral of mean curvature and the specific integral of total curvature. These quantities play a central role in the quantitative characterization of structures components. Up to multiplicative constants the geometric characteristics are the densities of the random Minkowski measures defined for a homogeneous random set and the list of the four geometric characteristics is complete in some sense. Procedures for estimating the geometric characteristics are based on Crofton formulae, as well as a modification Hadwiger's recursive definition of the Euler number.

In recent years a generalized geometric characterization based on the local volume density, local surface density and local curvature densities was suggested for the geometric treatment of porous and heterogeneous media in physics. These

geometric characteristics can be readily incorporated into the mean field approximation for the microscopic boundary value problems describing transport phenomena in these media. It seems that the resulting parameter free predictions are in good agreement with experiment.

We assume that the α component is observed in a cuboidal lattice of points i.e. we consider spatial digital images of the microstructure to be investigated. The discrete version of a random set forms a random binary digital image. Depending on whether a lattice point is in α or in its complementary set this point is assigned the Boolean values 1 or 0, respectively. The selection can be performed by thresholding brightness values to separate the α -phase from the background.

For the purpose of application in image analysis the integrals that occur in the Crofton formulae and Hadwiger's recursive definition are discretized in such a way that measurement of the geometric characteristics can be performed by simple counting of elements in a digital image where the elements are voxels or neighborhood configurations of voxels. In other words the observation of the structure in a point lattice implies a corresponding discretization of the integral geometric formulae. The method which will be used is very close to ordinary repeated trapezoidal quadrature rule known from numerical mathematics.

A very powerful technique of image processing is the filtering of digital images. However, filtering can also be applied in the image analysis. The statistical estimation of the geometric characteristics suggesting in the following includes linear filtering as a basic tool. It consists of three steps:

- filtering of the binary image which yields a grey-tone image
- generating the vector of absolute frequencies of neighborhood configurations
- estimating the geometric characteristics from the absolute frequencies of configurations

By means of filtering each neighborhood configuration in a binary image is assigned an integer. Thus the result of the filtering is an image of integer valued voxels also referred to as a grey-tone image. The generation of the absolute frequencies of configurations can be understood as a discretized analog of the integration occurring in the integral geometric formulae and the vector of absolute frequencies carries the complete information of the image about the geometric characteristics it can be used as the data base of statistical estimation. Since the neighborhood configurations are represented by grey-tones, the vector of absolute frequencies are the vector of absolute frequencies of grey-tones in the filtered image.

Yields efficient algorithms for measuring basic geometric characteristics

- Volume density (volume fraction)
- Surface density (specific surface area)
- Density of integral of mean curvature
- Density of integral of total curvature (up to a constant the density of the Euler number).

Based on these results, application specific characteristics can be obtained. Examples are

- Mean fibre length per volume for fibre materials
- Mean number of sinter necks per sinter particle e.g. for sintered copper
- Mean cell size for open foams.

Moreover, texture analysis is possible. For example, preferred orientations of faces or fibres can be determined.

1.2 Spatial image analysis

Based on 3d or 2d images, microscopic heterogeneous but macroscopic homogeneous materials are modelled.

The model of the microscopic structure is constructed starting from microscopic or tomographic images. First, a suitable model is chosen. Then, geometric characteristics like porosity and specific surface density of the material are determined using (3D) image analysis. The model is fitted accordingly.

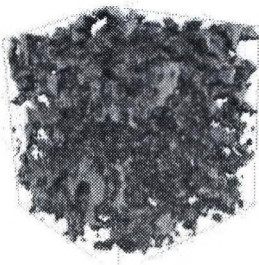
In simulated realisations of the model, parameters of the material like relative permeability or acoustic absorption can be computed by fluid dynamics. Using a model

- is the only possibility if 3d images are not available
- allows computations in a high number of realisations thus taking into account the spatial variability of the material
- makes virtual experiments possible: checking the effects of slight changes of the geometry of the material without actually creating the material

The basis are models from stochastic geometry like Boolean models, line processes, and tessellations. The macroscopic homogeneity of the materials is reflected by stationarity (invariance w.r.t. translations) of the models.

2 EXAMPLES

2.1 From the material to the model using 3d images



Reconstructed tomographic image of a Berea sandstone probe. The visualisation shows the pores.

A suitable model is a Boolean model: the union set of balls centered at the points of a homogeneous Poisson point field.

In the 3d image data, volume and surface density were determined:

$$V_v = 82.2\% \text{ and } S_v = 13.9 \text{ mm}^{-1}.$$

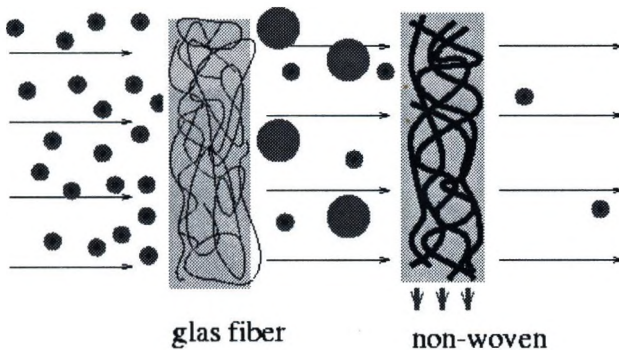


In the simplest case, there are just two parameters to choose - intensity of the point field and radius R of the balls.

Miles' formulas yield the relation between volume and surface density and the parameters of the Boolean model:

$$V_v = 1 - \exp^{-\lambda \frac{4}{3} \pi R^3}, S_v = 4\pi R^2 \exp^{-\lambda \frac{4}{3} \pi R^3} \text{ and thus } \lambda = 1403 \text{mm}^{-1}, R = 0.066 \text{mm}.$$

2.2 Two phase flow of air and oil in filters



Oil filters contain various layers of glass fiber papers and non-woven materials. The properties of the used materials are important for the efficiency of the filters. Aim is the development of a filter with minimal pressure decrease and minimal rest oil at the end of the filter.

We simulate the flow of two phases (e.g. air and oil) in a geometry model of fiber materials. For this purpose the Lattice Boltzmann method is especially suitable because this method can handle boundary conditions in any geometric structure without additional efforts.

We calculate

- the oil distribution.
- the relative Permeability with the Darcy law for two phases from the average velocity.
- the pressure distribution and the capillary pressure.

The results depend on:

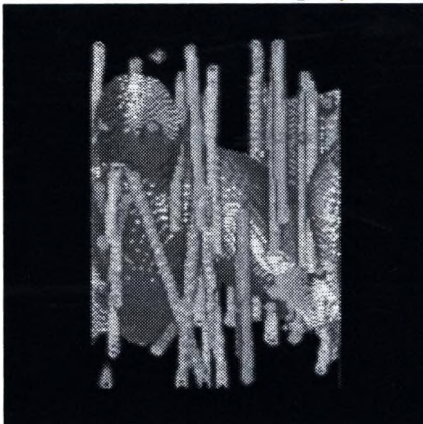
- The oil saturation.
- The properties of the geometry model (e.g. porosity, radii distribution).
- Material properties of the two phases (oil-air, water-air).
- Wetting properties



Oil non wetting

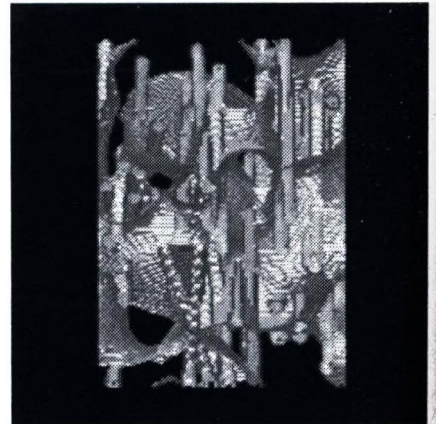


Oil non wetting



Oil wetting

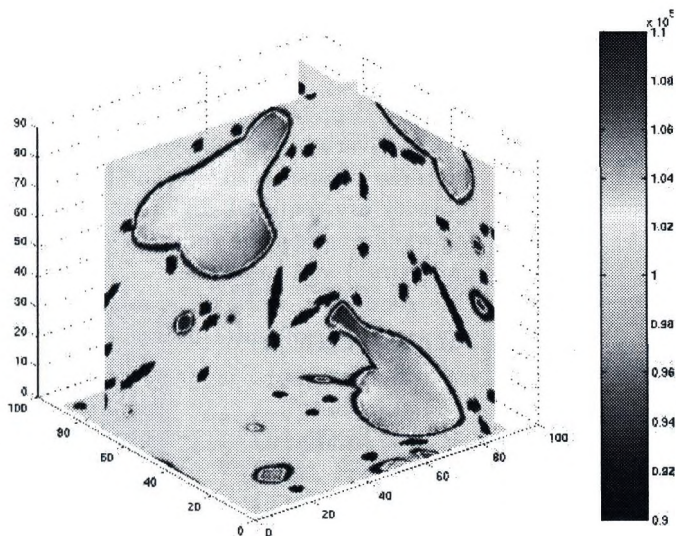
Saturation 0.22



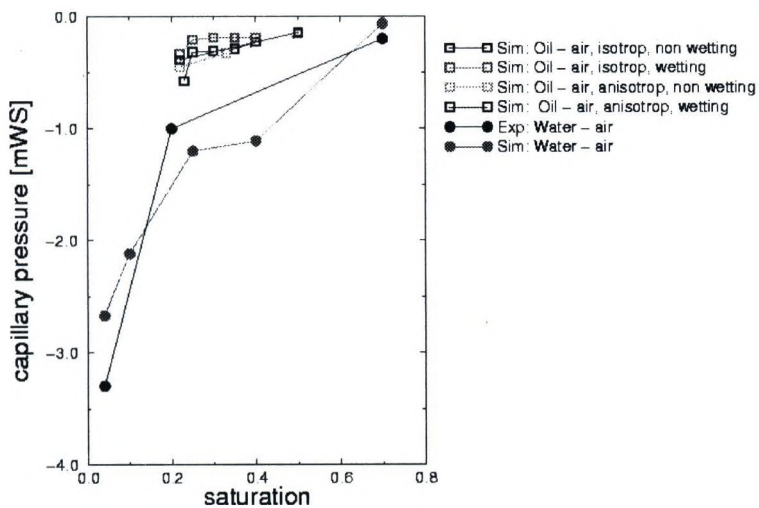
Oil wetting

Saturation 0.33

Oil distribution for various saturations and wetting properties for an anisotropic fiber material

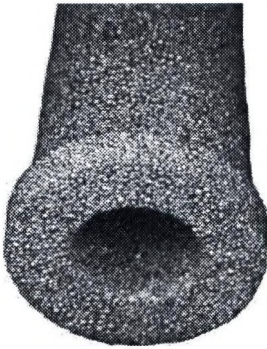


Pressure distribution in an isotropic fiber material: In the droplets the pressure is higher.



Capillary pressure curves

2.3 Analysis of sintered copper

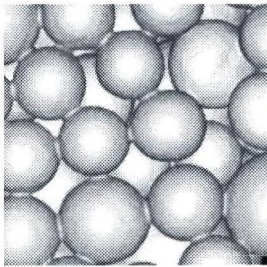


Sintered copper is a porous material, which is used for filters in various technological processes.

Of these filters, good mechanical properties and high porosity are required. In order to meet these conflicting requirements, the sintering process has to be better understood and controlled.

One geometric characteristic for the quality of a filter is the number of sinter necks per sinter particle.

Measured basic geometric characteristics:

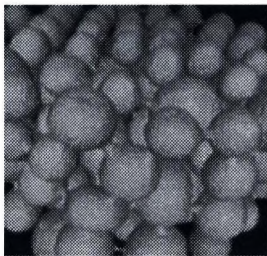


- volume density $V_v = 57.8 \%$
- surface density $S_v = 1.466 \text{mm}^{-1}$
- density of integral of mean curvature $M_v = -2.931 \text{mm}^{-2}$
- density of integral of total curvature $K_v = -0.447 \text{mm}^{-3}$.

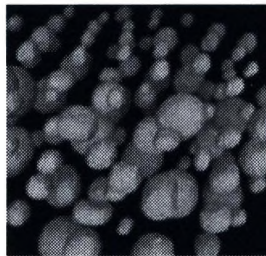
Euler number: $\chi = \frac{1}{4\pi} V K_v = -121$.

Reconstructed tomographic image of the original material.

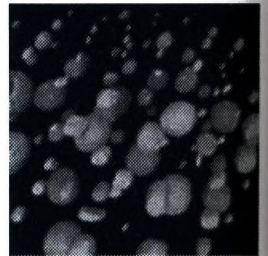
The sinter particles are separated by successive erosions. Then the Euler number is the number of particles.



Erosion not yet sufficient. There are still sinter necks visible.



Right step of erosion. Euler number: $\chi = 133$.



Erosion too far. Some particles have disappeared completely.

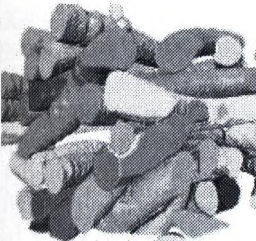
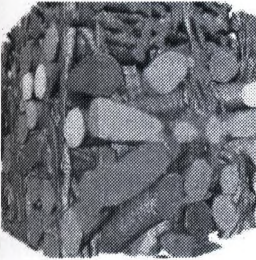
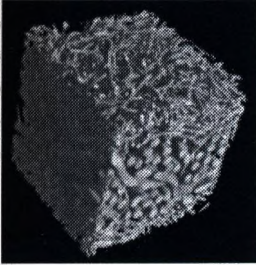
The mean number of sinter necks per particle is $2^{\frac{133-(-121)}{133}} = 3.82$. This result agrees with the number deduced from what is known about sinter material and process.

2.4 Fiber systems

Material: Fiber material

Application:

- Separation of thick and thin fibers
- Measurement of the directional distribution of thin and thick fibers
- Determination of porosity, density of total fiber length and mean area of cross sections
- Labeling of the thick fibers (in a window of site 2563 pixel)



Description:

1. Binarisation
2. Morphological opening yields the thick fibers
3. Black-Top-Hat transform yields the thin fibers
4. Measurement of the Minkowski functionals from the binary image, computing from the Minkowski functionals the following quantities: for the thick fibers: porosity 77%, density of total fiber length 5.2 mm² for the thin fibers : porosity 76%, density of total fiber length 125.8 mm², mean area of cross sections 0.002 mm²
5. Measurement of the mean number of cross sections with respect to 13 space directions, computing the directional distribution of the fibers
6. Erosion, watershed algorithm, intersection of the watershed image with the original image yields the labelled fibers

Depth-based Indexing and Retrieval of Photographic Images

László Czúni¹, Dezső Csordás¹, and

Veszprémi Egyetem, Képfeldolgozás és Neuroszámítógépek Tanszék

¹ H-8200 Veszprém, Egyetem u. 10

czuni@almos.vein.hu

Abstract. This paper proposes a new technique for image indexing and retrieval to implement a content-based image retrieval (CBIR) system [3, 4]. The introduced technique uses range from focus technique to gather 3D information of a scene and the obtained depth-map is segmented and stored together with each individual image. During retrieval the user can describe the query image not only in a conventional way but also with a layered representation where a few (typically 3) depth layers define the distance from the camera. The application of this technology could be implemented in future's digital still cameras.

1 Introduction

The main problem in CBIR systems is that humans think differently than computers [11]. For non-specialists it is difficult to explain what textures, adjacency maps, edge density, color saturation, etc. means. That is why query by example and relevance feedback techniques are so popular in general retrieval systems [11]. However, it might happen that the investigation of psycho-visual experiments may give us new ideas that can be used to increase the efficiency of today's image retrieval methods. A good example is to find out what is the role of the eye movement when seeing, understanding and memorizing an image [10]. Another interesting question is how humans remember the 3D structure of real-world scenes. The application of depth layers in CBIR would be just another new feature - one might say. But possibly this is very close to the model of the real world in our imagination. We don't have to explain to non-specialists what distance means but unfortunately its role in image retrieval has not been discovered yet. At the current stage we are only at the beginning of our work but we found the exposed preliminary results very encouraging. All proposed algorithms are simple and could be combined with more sophisticated methods for further enhancement.

2 Capture of Image and Depth Information, Segmentation

Commercial digital cameras can capture high quality color images and different techniques are used for setting the focus automatically. While there are a

few methods that use sound, laser or infrared active sensors for measuring the distance of objects from the camera the vast majority applies passive sensors. Measuring image sharpness is a very common way of setting the focus and consequently range from focus/defocus is a natural extension for gathering depth information. Range from zoom, structure from motion, and stereovision can also be applied in several cases.

In our paper we apply the range from focus technique to get depth information related to the captured 2D image. In general, depth from focus algorithms try to estimate the distance of one image region by taking many images with better and better focus. In contrast we try to estimate distance on the whole field of view. Range from defocus [9] is an alternative approach to get a detailed depth map but not discussed in this paper. The well-know formula describing the relation between the focal length of a lens (f), the object distance (u) and the lens position (v) is:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \quad (1)$$

That is if we work with fixed f (fixed zoom) and find the best lens position (v) with minimal blur we can give an estimation of u ; the only question is how to find the best focused position. Since we want to get a depth-map of the whole image the focus is measured on the whole image area and at every image location the lens' position (v) with the smallest blur effect is stored. However, the image structure can be quite complex and in some cases it would require a long range with dozens of position to sample to find all areas focused. For this reason and to minimize computational load in our experiments we made 8 shots with different lens positions. Although focus measuring does not require sophisticated algorithms, as explained below, to have the computations done in the camera requires special camera hardware that is not available in our case so the 8 shots were downloaded to the host PC then processed off-line.

Several techniques exist to measure the focus. The optimal method for a given configuration depends on the OTF (optical transfer function), the noise behavior, the camera parameters, and even the object that is observed [7]. Unfortunately, these pieces of information are not available in our case and practical considerations (such as the size of the area where focus is measured or the computational complexity of a focus measure function) can also be crucial when implementing the focus measure in a commercial camera. Although physical parameters of the camera (such as iris, focal-length) have a great impact on the estimation process either, our uncalibrated cameras gave satisfactory results. This was due to the fact that we didn't need precise depth maps; rather the relative position of image regions was important. (In our experiments we applied some zoom to decrease the depth-of-field to get more steep focus measure functions). We tested several focus measure functions: gradient, variance, entropy, and found the Laplacian operator

$$L(i, j) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2)$$

to be the most reliable for dozens of different test images. (Our observations are similar to the conclusions described in [7].) Since we don't need high-resolution depth-maps and want to decrease uncertainty we averaged the focus measure $|L|$ in blocks of size of app. 30×30 . In our experiments the focal length of the camera was set between 16 and 24mm and the 8 images were taken focusing at 0.7, 1, 2, 3, 5, 7, 10; and at (meters object distance. Figure 1 a) and b) illustrates a color image and the relevant gray-scale depth-map.

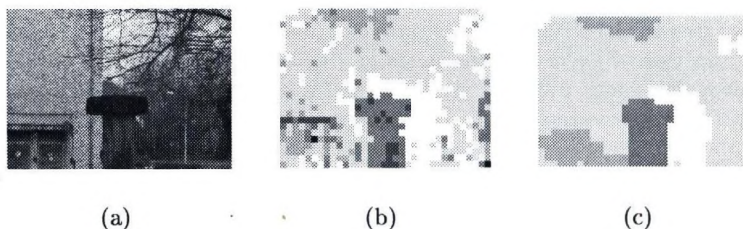


Fig. 1. Input image (a), related depth-map (b), and segmented depth map (c). Closer objects appear darker (doors belong to small building in front of the wall)

Segmentation

There are three main purposes of segmentation of depth-maps: it helps to remove noise (originating from measurement uncertainty), it decreases the required storage and indexing capacity, and it gives a better representation for symbolic description. We are investigating and testing different depth-map segmentation algorithms:

1. Quad tree segmentation: simple and fast but not very effective for noisy data.
2. MRF-based segmentation ([8]): more complex; can handle any arbitrary shapes and can be applied in the presence of strong noise.
3. Joint estimation and segmentation of focus information: in this method we would like to use MRF techniques to simultaneously evaluate and segment the focus measure function. Instead of simply searching for the maximum of the focus measure function we also consider the focus measure of neighboring regions.
4. Joint segmentation of depth-map and color: multimodal segmentation techniques are to be investigated to get more reliable results.

In this paper results of the 2nd type are presented. There are several MRF-based techniques used for the segmentation of noisy images. We have chosen [8] since it is a simple and fast technique without any a-priori model information and the applied Modified Metropolis Dynamics' (MMD) fast convergence

is also proven. The segmentation problem is solved with a MAP (Maximum A Posteriori) estimation of gray-scale pixel value classes (w) based on the initial observation (f and its smoothed version S) and on the rule that neighboring pixels are probably taking the same value on the segmented image. This is implemented in an energy optimization algorithm where the energy at a pixel location (p) to be minimized consists of two terms added:

$$E_p(w) = frac(\omega_p - \mu_p)^2 2\sigma_p^2 + \sum_{\{p,r\} \in C_p} V(\omega_p, \omega_r) \tag{3}$$

where $\mu_p = \frac{f_p + S_p}{2}$ and $\sigma_p = \frac{|f_p - S_p|}{2}$, and $V(\omega_p, \omega_r) = \begin{cases} -\beta, & \text{if } \omega_p = \omega_r \\ +\beta, & \text{if } \omega_p \neq \omega_r \end{cases}$.

In our implementation w was selected from the 8 possible range classes and β (the value controlling homogeneity) was typically around 1. The first term in Eq. 3 is responsible for getting a result that is close to our original observations while the second term gives homogeneity of neighboring regions (β denotes that pixels p and r form a "clique" [8]). The relaxation algorithm is controlled with MMD. Figure 1 c) shows an MRF segmented depth-map.

3 Image and Depth-Map Indexing and Retrieval

In our first experimental system we are to investigate the usefulness of pure depth-map search then we are to combine it with conventional methods in later experiments. At the moment of writing this proposal the number of image and depth-map pairs is app. 100 (but it is to be increased to get more reliable evaluation).

In our first experiments unsegmented depth-maps had a small resolution of 16x12 blocks. Although this could be larger we found that other factors influence results more significantly as explained below. We made several experiments with query by example and query by sketch methods. Rated results of the query by example searches according to the average distance with the l2 norm are in Figure 2/Table 1 and Figure 3/Table 2. The first elements of the tables are the query images and all values are normalized between 0 and 10 within each search.

Table 1. Results of the query of Figure 2

0	2.14	2.43	2.84	3.13	3.30
3.33	3.58	3.82	3.91	4.04	4.09
4.07	4.07	4.16	4.19	4.21	4.33

Table 2. Results of the query of Figure 3

0	0.55	0.58	0.61	0.62	0.63
0.74	0.74	0.75	0.85	0.92	0.98
1	1	1.12	1.13	1.13	1.15

As we can see in the first example (Figure 2) the first few results are close to expectations with very closed objects on the very left edge. In the second example (Figure 3) the background was dominant over the relatively small foreground

traffic sign. The results of this query did not meet users' expectations precisely, no matter that the quantitative results in Table 2 were quite close. Naturally, by refining and bounding the query to a specific region of the image leads to better results but in that case the exact position of the specified region becomes the key point of the query. Besides, there are some observations we experienced during our tests that should be considered:

- Our depth-maps had a range of 10 meters. Over this distance all objects are considered at inf.
- In many examples the foreground level (usually grass, pavement, floor, etc.) is also visible in the bottom of the images. People defining a query do not take this into consideration still this can significantly modify the results.
- In some cases relative distance measure (i.e. relative position along the depth axis) gave better results.

According to our preliminary test with humans, specifying the shape of blobs in the depth-map seems to be less reliable than describing their relative positions. It is also obvious that photographic images can be easily characterized by the focus position of the final (sharp) image. All these observations lead to a solution of symbolic description of segmented depth information. Segmentation techniques listed above could be used for this purpose. We propose to use a graph based representation similar to [5,6] or other symbolic-based methods [1]. Graph-based description of depth-map's blobs with information of their size, relative position could be also combined with color features.



Figure 2. Results of query by example and depth maps; query is the first element



Figure 3. Results of query by example and depth maps; query is the first element. Large background area (white) matches suppress the area of the traffic sign

4 Conclusions, Future Work and Possible Applications

The proposed technique is based on range from focus estimation that is available in many digital cameras today. We have shown preliminary results for image search and described some observations that lead to the need of implementing a symbolic description of depth information. There are several questions to be answered by future work: what is the minimum necessary resolution of the depth-map for general cases; how to get a more precise map; what is the optimal segmentation method for depth-maps; how to minimize the number of necessary focus measurements; what is an effective way of symbolic description of 3D structures?

This technique to be implemented in a digital camera requires no special hardware but the focus measuring parts should be extended to large areas of the image. With the combination of color indexing algorithms [2] the proposed method can be very useful in the indexing and retrieval of home or SOHO (small office/home office) photographic images within a Windows based software framework.

Authors would like to thank the help of Gergő Császár in programming work.

References

1. S. K. Chang and Erland Jungert: Symbolic Projection for Image Information Retrieval and Spatial Reasoning. Academic Press, London (1996)
2. G. Császár: Az emberi beavatkozás szerepe képek tartalom alapú visszakeresésében, M.Sc. Thesis, University of Veszprém (2003)

3. L. Czúni, D. Csordás: Depth-Based Indexing and Retrieval of Photographic Images, VLBV 2003, Madrid, Lecture Notes in Computer Science: Visual Content Processing and Representation, pp. 76-83, 2003.
4. L. Czúni, A. Licsár: Method of fixing, storing and retrieving images and the associated distance indexes. P0204432 Hungarian Patent Office, Budapest (2003)
5. L. Garrido, P. Salembier and A. Oliveras: Anti-extensive connected operators with application to image sequences. *IEEE Trans. on Image Processing*, 7(4) (April 1998) 555-570
6. P. Salembier and J. Serra: Flat Zones Filtering, Connected Operators, and Filters by Reconstruction. *IEEE Trans. on Image Processing*, Vol.4, No.8 (August 1995)
7. M. Subbarao and J.-K. Tyan: Selecting the Optimal Focus Measure for Autofocusing and Depth-From-Focus. *IEEE Trans. on PAMI*, Vol.20, No.8. (August 1998)
8. T. Szirányi, J. Zerubia: Markov Random Field Image Segmentation using Cellular Neural Network. *IEEE Trans. on Circuits and Systems I*, V.44 (January 1997) 86-89
9. Y. Xiong and S. Shafer: Depth from Focusing and Defocusing. Tech. report CMU-ROI-TR-93-07, Robotics Institute, Carnegie Mellon University (March 1993)
10. Yarbus, A. L.: Eye movements and vision, N.Y.: Plenum Press (1967)
11. R. C. Veltkamp, M. Tanase: Content-Based Image Retrieval Systems: A Survey. Tech. report, Utrecht University (October 2000)

Műtési tervek előkészítése véges elemes analízishez a MedEdit orvosi képfeldolgozó rendszer segítségével

Erdőhelyi Balázs¹, Ollé Krisztián¹, Varga Endre², Halmai Csongor¹, Kuba Attila¹

¹ Képfeldolgozás és Számítógépes Grafika Tanszék, Árpád tér 2,

² Traumatológiai Klinika, Semmelweis u. 6.

Szegedi Tudományegyetem, H-6720 Szeged

{ber, ollek, halmai, kuba}@inf.u-szeged.hu

varga@trauma.szote.u-szeged.hu

Kivonat ¹

A csontrendszer sérülésekor végzett műtétek gyakran nagy bonyolultságúak, ezért gondos, precíz tervezést igényelnek. Ebben segít a MedEdit rendszer a traumatológus sebésznek azzal, hogy lehetővé teszi a tényleges beavatkozás előtt akár több műtési terv kipróbálását, és azok hatásának, fizikai viselkedésének vizsgálatát. A megjelenítéshez szükséges geometriai modell mellett szükség van egy mechanikai modellre is, melynek alapján a fizikai analízis elvégezhető egy véges elemes szoftverrel.

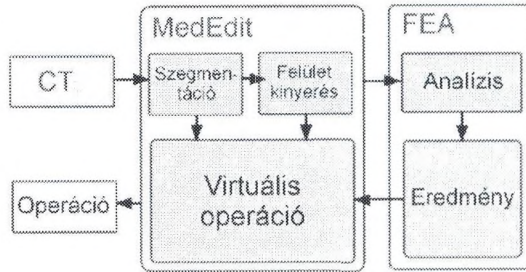
A MedEdit rendszerhez több modult is készítettünk, melyek alkalmasak mechanikai modellek előállítására, így a csont héjként és tömör testként is modellezhető különféle véges elemes hálókkal. Bemutatunk továbbá egy olyan modellt is, mely kombinálja a héj és a belső szivacsos állomány mechanikai modelljét. Az ismertetett modelleket megvizsgáljuk olyan szempontból is, hogy hogyan lehet elhelyezni bennük a csonttól eltérő anyagi tulajdonságokkal rendelkező implantátumokat. Dolgozatunkban ezen módszereket tárgyaljuk részletesen, számos példán keresztül.

1. Bevezetés

Az a lehetőség, hogy a traumatológus orvos még a sebészi beavatkozás előtt kipróbálhasson többféle műtési tervet, és ezek közül kiválaszthassa a beteg számára legmegfelelőbbet, felbecsülhetetlen értékű lehet és akár életet is menthet. Ezért a Szegedi Tudományegyetemen egy olyan projekt kezdődött melynek célja, hogy olyan eljárásokat ill. szoftvert fejlesszen ki, mely képes balesetben csontsérülést szenvedett betegeken biomechanikai tesztek elvégzésére, anélkül, hogy sebészi beavatkozásra kerülne sor. Az eljárás 3D képeket használ, és matematikai modell segítségével definiál egy műtétet, mely optimális esetben javítja a sérült csonton végezhető stabilizációs műtét típusát.

¹ Ezt a kutatást az OTKA T37840 támogatta.

Már léteznek kereskedelmi forgalomban is hasonló problémákat kezelő rendszerek [4] [5]. Ezek általában 2D ill. 3D leképező eszközöket használnak az adatgyűjtéshez és a beavatkozás megtervezéshez. A mi rendszerünk CT képekből indul ki, melyből felépíti a geometriai modellt majd a mechanikai modellt. Ezt követően végezhető el a véges elemes analízis. Dolgozatunkban ezen rendszert, és a véges elemes modell elkészítéséhez szükséges lépéseket mutatjuk be.



1. ábra.

A rendszer vázlatja a fő komponensekkel, és a köztük lévő kapcsolatokkal.

2. A MedEdit rendszer

Az 1. ábra mutatja a MedEdit rendszer főbb komponenseit. A CT képek DICOM (Digital Imaging and Communications in Medicine) formátumban állnak rendelkezésre illetve a csont szegmentációja a voxelek szűrkeségi értékein alapul [2].

A következő lépés a felületkinyerés. Itt az ún. Marching Cubes [6] algoritmust használjuk, mely a szegmentált 3D adatokból egy háromszög-hálós felületi modellt állít elő, amelynek méretét egy felületegyszerűsítő eljárással [1] csökkentünk. Az így nyert geometriai modellen a sebész virtuális operációt végezhet, lyukakat fúrhat, implantátumokat helyezhet a csontba.

Kiegészítve a geometriai modellt anyagi tulajdonságokkal kapjuk a mechanikai modellt, amin már végrehajtható a véges elemes analízis (FEA). Feszültséganalízis vizsgálatokhoz a mechanikai modellt el kell látni terhelési- és elmozdulási vektorokkal. A véges elemes analízishez általában speciális mérnöki tudás szükséges. Rendszerünk az orvos számára kényelmes felhasználói felületen keresztül biztosítja ezt a tudást, mert a FEA programmal a kommunikáció az orvostól elrejtve egy szkript felületen keresztül valósul meg.

A terheléses feszültség-analízis a véges elemes programon belül történik, és az analízis eredményei is ott láthatóak. Az eredményektől függően az orvos ellenőrizheti műtéti tervét, esetleg visszatérve a virtuális operációhoz, módosíthat az

ott megadott paramétereken. Így lehetséges a műtétet megelőzően több beállítás tesztelése, és azok hatásainak virtuális ellenőrzése.

3. Véges elemes modellek

A mechanikai modell elkészítéséhez szükség van egyrészt a geometriára, másrészt olyan mechanikai paraméterekre, mint az anyagi minőség és egyéb fizikai állandók.

A geometriát a FEA programok egy úgynevezett véges elemes hálóval közelítik. Ennek a hálónak a pontjai adják meg a modell alakját, míg az úgynevezett véges elemek azt mondják meg, hogy az egyes pontok milyen kapcsolatban vannak egymással.

A véges elemes háló előállításához FEA rendszer számára két különböző forrásból is származhatnak az adataink: a felületet leíró háromszögháló melyet a megjelenítéshez használunk, valamint a szegmentált 3D kép. Az előbbit a FEA rendszer héjként, az utóbbit pedig tömör testként kezeli.

3.1. Héj modell

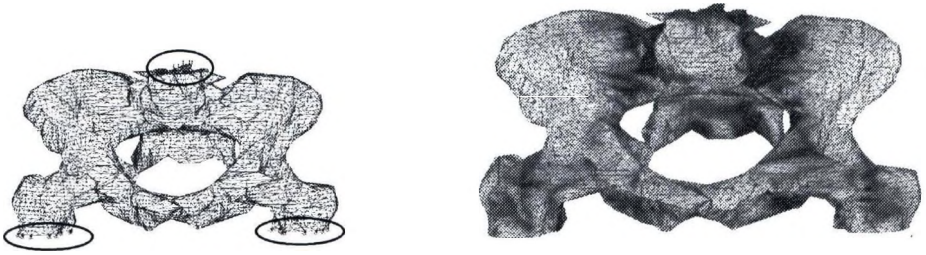
A héj exportálása esetén a megjelenítéshez használt háromszög-felületet alakítjuk át véges elemes hálóvá. Minden felületi háromszöghöz hozzárendelünk egy három pontból álló héj elemet. A héj elemek megadható vastagságú síkbeli lemezek modellezésére használhatók (2. ábra bal).

Az így nyert háló valójában nem más, mint egy apró lemezekkel határolt modell, melynek üres a belseje. Hogy ez mégis jól közelíti a valóságot, arra adhat magyarázatot, ha megvizsgáljuk az emberi csontok szerkezetét. Ugyanis a csont felületén található kb. 1 mm vastag kéregállomány 100-szor keményebb a belső szivacsos állománynál [3].

A héj modell esetében a kéregállomány anyagi tulajdonságait használja a rendszer, mely esetében az elasztikus modulus 1100 MPa valamint a Poisson érték 0.3 [3]. Az elemzés még nagy elemszám esetén is viszonylag gyors, és az eredmények (2. ábra jobb) is megfelelnek a klinikai elvárásoknak.

Implantátum. Sajnos a modell kevésbé alkalmazható, ha implantátumokat is be szeretnénk vonni az elemzésbe. Azáltal, hogy a modell belseje üres, az összes feszültség a csont felületen jelenik meg, illetve az implantátum olyan irányban is elmozdulhat, amit egy belső szerkezettel rendelkező modell már nem engedne meg.

Példa. A 2. ábrán egy medencecsont héj modellje látható, melynek legalsó pontjai rögzítettek, míg felső pontjait föntről lefelé irányuló, 600N erőhatás éri.



2. ábra.

Balról a medencecsont felületének mechanikai modellje. Tartalmazza a véges elemes hálót, az anyagi tulajdonságokat, a terhelés erővektorait, valamint a nulla elmozdulású pontokat (olyan pontok, melyek nem mozognak el). A véges elemes háló 3-pontú elemeket tartalmaz. Az ábra tetején ellipszisszel jelölt területen láthatóak az erővektorok, míg a két combcsontnál a nulla elmozdulású pontok. Jobbról medencecsont terheléses feszültség-analízisének eredménye. A sötétebb színek jelentik a nagyobb anyagi feszültséget.

3.2. Tömör modell

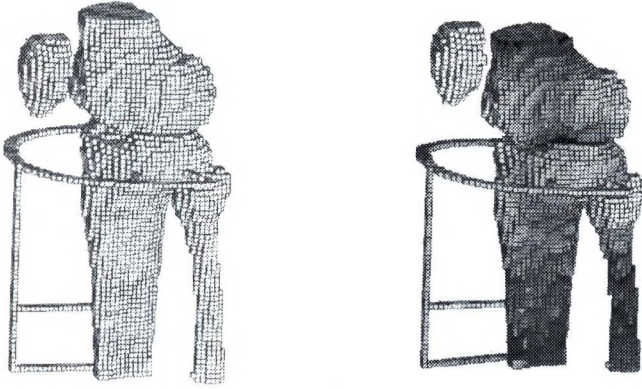
Az implantátumok elhelyezésével kapcsolatos problémát tették szükségessé, hogy olyan véges elemes hálót tudjunk generálni, mely belső pontokat is tartalmaz. A MedEdit rendszer a szegmentált 3D képet használja fel egy ilyen modell előállítására. Ekkor a szegmentált 3D kép minden egyes voxelét (mely csontot kódol) megfeleltetjük egy 8-csúcú, kocka alakú véges elemnek. Az így nyert háló (3. ábra bal) belső pontokat is tartalmaz, részletessége pedig a rendelkezésre álló CT kép felbontásától függ.

A felbontás növelésével a véges elemek száma köbösen növekszik, amivel együtt jár az analízis erőforrás igényének (memória, CPU) növekedése. Ha a háló túl nagyméretű, a rendszer egyszerű alul-mintavételezéssel csökkenti a voxelek számát.

Az anyagi tulajdonságok megadásakor itt nem a csont kéregállományát vesszük alapul, hanem átlagos értékekkel dolgozunk: elasztikus modulusnak 300 MPa-t, Poisson értékéhez 0.2-t használtunk [3].

Implantátum. A tömör modellben az implantátum elemeket is hasonlóan lehet generálni mint a csont elemeket, ha rendelkezésünkre áll a szegmentált 3D kép. Ezért a MedEdit rendszer az implantátumokat, melyeket a virtuális műtét során helyezzünk el, beilleszti a 3D voxelhalmazba is.

Példa. A 3. ábrán egy térd ízületet mutatunk be, melynek az alsó pontjait rögzítettük, míg felső pontjaira föntről lefelé irányuló, 600N nagyságú erőhatást fejtettünk ki. A modellt kiegészítettük egy külső merevítő gyűrűvel, melyet gyakran használnak törött csontok rögzítésére.



3. ábra.

Bal oldalon a térd szegmentált 3D képéből előállított mechanikai modellje melyet egy külső rögzítéssel (Hybrid Ring Fixator) láttak el. Az alkalmazott véges elemes háló csupa 8-csúcsú (kocka) elemeket tartalmaz. Jobbról a terheléses feszültséganalízis eredménye, ha a térdet föntről lefelé irányuló erőhatásnak tesszük ki. A sötétebb színek jelentik a nagyobb anyagi feszültséget.

3.3. Vegyes modell

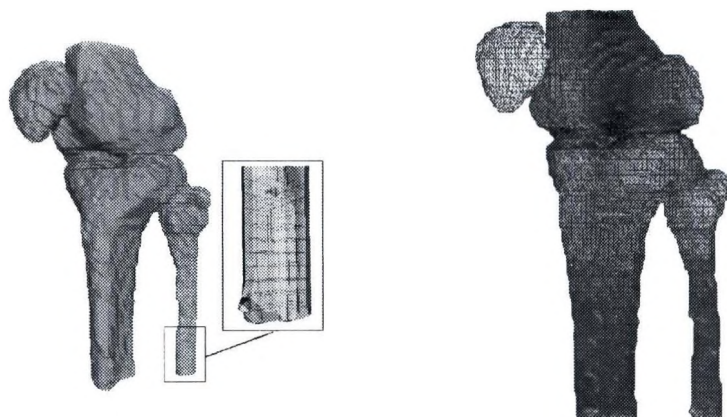
Harmadik lehetőségként generálhatunk a MedEdittel egy olyan véges elemes hálót, mely egyszerre tartalmazza a korábbi két modell előnyös tulajdonságait: kellőképpen kis méretű, a modell belsejét is kitölti elemekkel, és képes megkülönböztetni a csont kéregállományának és szivacsos állományának eltérő anyagi tulajdonságait.

A szegmentált 3D képet felhasználva, a Marching Cubes algoritmussal [6] generált háromszöghálót a 3.1 fejezetben leírt módon feleltetjük meg 3-pontú véges elemeknek. Ezek az elemek alkotják a modell felszínét. Ezt kiegészítjük olyan 2-pontú elemekkel, melyeket a modell belsejében helyezünk el a következő módon. A felület pontjaiból a koordináta tengelyekkel párhuzamos sugarakat bocsátunk ki, és megkeressük hol metszi a sugár a csont (vagy implantátum) felszínét (4. ábra bal). Az így nyert két pontot használjuk a véges elemek létrehozására, azzal a megszorítással, hogy ha egy elem túlságosan hosszú lenne, akkor azt feldaraboljuk közbenső pontokkal.

A külső elemek a csont kéregállományát modellezik, és anyagi tulajdonságukat is e szerint állítja be a rendszer. A belső elemekre a szivacsos állomány fizikai állandóit használjuk: elasztikus modulus 10 MPa, Poisson érték 0.3.

Ez a modell szintén alkalmas implantátumok kezelésére, hiszen a belső szerkezet ugyanúgy kapcsolódik csonthoz, mint más anyaghoz.

Példa. A példában szintén egy térd ízület látható, melyet a 3.2 fejezethez hasonlóan terheltünk meg (4. ábra).



4. ábra.

Bal oldalon a térd mechanikai modellje látható, mely kívül 3-pontú héj elemekből (áttetsző az ábrán) ill. 2 pontú belső elemekből épül fel. A jobb oldalon a feszültséganalízis eredménye látható, ha a csontot az 3. ábrához hasonló terhelésnek tesszük ki.

3.4. Összefoglalás

A mechanikai modell közvetlen adatokat biztosít a beteg sérülésének súlyosságáról, ami által számos orvosi komplikáció elkerülhető. Dolgozatunkban bemutatunk 3 módszert a mechanikai modell előállítására, valamint példákon keresztül szemléltettük a keletkező modell tulajdonságait. Ismertettünk egy olyan modellt is mely elegendően kis elemszámú, képes megkülönböztetni a különböző csontok anyagi tulajdonságait valamint implantátumok kezelésére is alkalmas.

Hivatkozások

1. Garland M, Heckbert PS. Surface simplification using quadric error metrics. ACM SIGGRAPH' Computer Graphics; 1997.p. 209–16.
2. Nyúl LG, Falcao AX, Udupa JK. Fuzzy- Connected 3D Image Segmentation at Interactive Speeds. SPIE Medical Imaging 2000; 3979 p.212- 23.
3. Beer FB, Johntson ER. Mechanics of Materials. McGraw-Hill Inc 1980; 42.
4. Computer Assisted Orthopaedic Surgery home page, <http://www.caosinternational.org/> [03/14/2003]
5. Robb, R. A., D. P. Hanson, J. J. Camp: Computer-aided surgery planning and rehearsal at Mayo Clinic. Computer, 29(1):39-47, 1996.
6. Lorensen WE, Cline HE. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM SIGGRAPH Computer Graphics; 1987 21(4)

Recognition of Facial Gestures

Attila Fazekas

University of Debrecen, Institute of Informatics
fattila@inf.unideb.hu

Abstract. In this paper we report a new application of the Support Vector Machines for facial gestures recognition.

1 Introduction

The general problem of pattern classification is as follows: Let us suppose that different types of measurements describing an unknown set of l patterns are observed. These measurements are designated as x_1, \dots, x_m , and can be represented by an m -dimensional vector \mathbf{x} in the (*input*) space X . Let the \mathcal{K} possible *pattern classes* be $\omega_1, \dots, \omega_{\mathcal{K}}$. The function of a *pattern classifier* is to assign (or to make a decision about) the correct class membership to a given *pattern* $\mathbf{x} \in X$ [1, 2].

We will focus on a *two-class pattern recognition problem* (or *binary decision*) ($\mathcal{K} = 2$). One of the classes is the class of *positive examples* and the other is the class of *negative examples*. The decision function of the classifier can be seen as an *indicator function* which admits the values $+1$ or -1 , if the corresponding feature vector is a positive example or a negative one, respectively.

One of the methods to solve a two-class pattern recognition problem is to resort to example-based techniques (or *learning systems*), such as *neural networks* or *support vector machines* (SVM) [4–7].

SVM implements the following idea [4]: By mapping the input pattern vectors, which are the elements of the training set, into a high-dimensional feature space through an a priori suitably chosen mapping, we expect that the elements of the training set will be linearly separable in the feature space. We construct the optimal separating hyperplane in the feature space to get a binary decision whether the input vector belongs to a given class or not. For example, in the case of the application studied in the paper, facial gesture recognition, the input vector comprises gray levels of pixels from a rectangular region of the digital image and the result of the binary decision is the answer whether this region, for example, is a smiling face or not.

2 Support Vector Machines

SVMs are learning algorithms based on the statistical learning theory [6]. In this section we briefly describe the foundation of SVMs by [6]. Statistical learning from examples aims at selecting from a given set of functions $\{f_{\alpha}(\mathbf{x}) \mid \alpha \in \Lambda\}$, the

one which predicts best the correct response (i.e. the response of a supervisor). This selection is based on the observation of l pairs that build the *training set*:

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), \quad \mathbf{x}_i \in \mathbb{R}^m, y_i \in \{+1, -1\} \quad (1)$$

which contains input vectors \mathbf{x}_i and the associated ground "truth" y_i given by an external supervisor.

Let the response of the learning machine $f_\alpha(\mathbf{x})$ belongs to a set of indicator functions $\{f_\alpha(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^m, \alpha \in \Lambda\}$. If we define the *loss-function*:

$$L(y, f_\alpha(\mathbf{x})) = \begin{cases} 0, & \text{if } y = f_\alpha(\mathbf{x}), \\ 1, & \text{if } y \neq f_\alpha(\mathbf{x}) \end{cases} \quad (2)$$

that measures the error between the ground truth y to a given input \mathbf{x} and the response $f_\alpha(\mathbf{x})$ provided by the learning machine, the expected value of the loss is given by:

$$R(\alpha) = \int L(y, f_\alpha(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy \quad (3)$$

where $p(\mathbf{x}, y)$ is the joint probability density function of random variables \mathbf{x} and y . $R(\alpha)$ is called the *expected risk*. We would like to find the function $f_{\alpha_0}(\mathbf{x})$ which minimizes the risk functional $R(\alpha)$. The selection of the function is based on the training set of l random independent identically distributed observations (1). In order to minimize the risk functional $R(\alpha)$ the *empirical risk minimization* (ERM) *induction principle* is usually employed by replacing the expected risk functional $R(\alpha)$ by the *empirical risk* functional, which is measured on the training set:

$$R_{\text{emp}}(\alpha) = \frac{1}{l} \sum_{i=1}^l L(y_i, f_\alpha(\mathbf{x}_i)). \quad (4)$$

The idea underlying the ERM induction principle is to approximate the function $f_{\alpha_0}(\mathbf{x})$ which minimizes $R(\alpha)$ by the function $f_{\alpha_l}(\mathbf{x})$ which minimizes the empirical risk. This approach may be valid for training sets having large size (ideally infinite). It is known that for some η such that $0 \leq \eta \leq 1$, the expected risk is bounded for arbitrary $\alpha \in \Lambda$ with probability $1 - \eta$ [4]:

$$R(\alpha) \leq R_{\text{emp}}(\alpha) + \sqrt{\frac{h(\log(\frac{2l}{h}) + 1) - \log(\frac{\eta}{4})}{l}}, \quad (5)$$

where h is a non-negative integer called the *Vapnik-Chervonenkis* (VC) *dimension*, and is a measure of capacity of SVMs. In the following, we shall call the right hand side of inequality (5) the *risk bound*. The second term in the risk bound is called the *VC-confidence*. Inequality (5) reveals the necessity to minimize both the empirical risk and the VC-confidence. This is the aim of the

so-called *structural risk minimization* (SRM) principle. The *support vector machines* (SVMs) are learning machines that implement the SRM principle in their training.

In order to introduce the basic idea of SVMs, let us consider the construction of the optimal separating hyperplane. Suppose the training data (1) can be separated by a hyperplane, that is $\exists \mathbf{v} \in \mathbb{R}^m$:

$$(\mathbf{v}^T \mathbf{x}_i) + b \geq 1, \quad \text{if } y_i = +1 \quad (6)$$

$$(\mathbf{v}^T \mathbf{x}_i) + b \leq -1, \quad \text{if } y_i = -1, \quad (7)$$

where \mathbf{v} is a normal to the hyperplane, $\frac{|b|}{\|\mathbf{v}\|}$ is the perpendicular distance from the hyperplane to the origin, and $\|\mathbf{v}\|$ is the Euclidean norm of \mathbf{v} . A compact notation for inequalities (6) and (7) is:

$$y_i ((\mathbf{v}^T \mathbf{x}_i) + b) \geq 1, \quad i = 1, \dots, l. \quad (8)$$

Let d_+ (d_-) be the Euclidean distance from the separating hyperplane to the closest positive (negative) example. Define the *margin* of the separating hyperplane to be $d_+ + d_-$. For the linearly separable case, SVM simply seeks for the separating hyperplane with the largest margin. The optimal hyperplane minimizes

$$\frac{1}{2} \|\mathbf{v}\|^2 \quad \text{subject to the inequalities (8)}. \quad (9)$$

In the case of inseparable data, the so-called "soft-margin" approach is used in the literature. For more details the interested reader may consult [5].

After training a support vector machine, one simply determines the side of the decision boundary where a given test pattern \mathbf{x} lies on and assigns the corresponding class label, i.e., $\theta(\mathbf{v}^T \mathbf{x} + b)$.

Let us investigate the generalization to the case where the decision function is not a linear function of the input vector. Now suppose we have first mapped the data to some other Euclidean space \mathcal{H} , using a mapping Φ :

$$\Phi : X \rightarrow \mathcal{H}. \quad (10)$$

Then the training algorithm would only depend on the data through inner products in \mathcal{H} , i.e., on functions of the form $\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$. If there were a *kernel function* K such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$, we would only need to use $K(\mathbf{x}_i, \mathbf{x}_j)$ in the training algorithm without necessarily explicitly knowing $\Phi(\mathbf{x})$.

How can we use this machine? After all, we need \mathbf{v} that will be in \mathcal{H} as well. The solution of the optimization problem (9) yields a coefficient vector \mathbf{v} that is expressed as a linear combination of a subset of training vectors, whose associated Lagrange multipliers are non-zero. These training vectors are called *support vectors*. In our case we will denote the images of the support vectors \mathbf{s}_i , $\Phi(\mathbf{s}_i)$. In the test phase, an SVM computes the sign of:

$$f(\mathbf{x}) = \theta \left(\sum_{i=1}^{N_S} \lambda_i y_i \Phi(\mathbf{s}_i)^T \Phi(\mathbf{x}) + b \right) \quad (11)$$

where λ_i are the Lagrange multipliers that are associated to $\Phi(\mathbf{x}_i)$, and N_s is the number of the support vectors. Again, we can avoid computing $\Phi(\mathbf{s}_i)^T \Phi(\mathbf{x})$ explicitly and use $K(\mathbf{s}_i, \mathbf{x}) = \Phi(\mathbf{s}_i)^T \Phi(\mathbf{x})$.

3 Experimental Results

For the facial gesture application, we used our face database of 600 images. All images in this database are recorded in 256 gray levels and are of dimensions 640×480 . A training data set of 40 images, 20 images containing a face pattern with one of the given gestures (angry, happy, serial, sad, surprised) and the others containing face patterns with gestures different from the given one.

The procedure for collecting face patterns is as follows: from each image, a bounding rectangle of dimensions 256×320 pixels has been manually determined that includes the actual face. This area has been subsampled four times. At each subsampling, non-overlapping regions of 2×2 pixels are replaced by their average. Accordingly, training patterns of dimensions 16×20 are built. The dimensionality of the input space is 320. The groundtruth, that is the class label +1, has been appended to each pattern containing the given gesture. In other cases the label has been -1.

The test set containing faces with different gestures. Thus, a set of 90 test images of size 16×20 each (50 faces with given facial gestures) was built from our database images not included in the training set.

For every facial gesture we trained 5 SVMs by the corresponding training set. The classification results of all SVMs in the case of 5 facial gestures are given in Table 1.

Table 1. Experimental results on Facial Gesture Database.

	Linear	Polynomial $d = 2$	Polynomial $d = 3$	Polynomial $d = 4$	Gaussian RBF
Angry	28.89	30.00	28.89	30.00	22.22
Happy	15.00	16.67	17.78	23.33	18.89
Sad	20.00	22.22	23.33	22.22	18.89
Serial	18.89	18.89	18.89	18.89	20.00
Suprised	26.67	26.67	26.67	26.67	27.78

Comparing these results, we can see that from the point of view of the classification error, the linear kernel outperforms all the other kernels. It can be seen the level of the classification error is acceptable. To improve the recognition algorithm we can combine these 5 SVMs trained for recognition one of the facial gesturers to one network. It will be the next step in our research work.



Fig. 1. Surprising face (a), smiling face (b), sad face (c), angry face (d) in the original resolution.

References

1. K.-S. Fu, "Learning control systems – review and outlook," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 3, pp. 327–342, May 1986.
2. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *Journal of the Association for Computing Machinery*, vol. 36, no. 4, pp. 929–965, October 1989.
3. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, 1990.
4. V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.
5. V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.
6. V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 988–999, September 1999.
7. E. E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," *CBCL Technical Report*, pp. 1–41, March 1997.

Összetett anyagok szövetszerkezetének rendezettsége

Gácsi Zoltán¹, Barkóczy Péter¹, Kovács Jenő²,

¹Miskolci Egyetem Fémteni Tanszék

²MTA Anyagtudományi Kutatócsoport
femtangz@uni-miskolc.hu

Kivonat Az összetett anyagok szövetszerkezetének jellemzésére általában egyszerű skaláris mennyiségeket használunk: szemcseátmérőt, részecskék közötti átlagos távolságot, térfogatarányt. Ezekkel a paraméterekkel nagyon megbízhatóan lehet leírni számos anyagfajta viselkedését, de vannak olyan anyagok is, amelyeknél ezek az adatok nem elegendők. Tipikusan ilyenek a részecske- és a szálerősítésű kompozitok, amelyek szakadásakor, valamint kifáradásakor és törési viselkedésekor az erősítő fázis eloszlása [1], [2] játszik kulcsszerepet [3]. A részecskék eloszlásának jellemzésére több eljárás is használatos [4], [5]. A dolgozatban a különböző módszerek közül a kovariancia alkalmazását mutatjuk be.

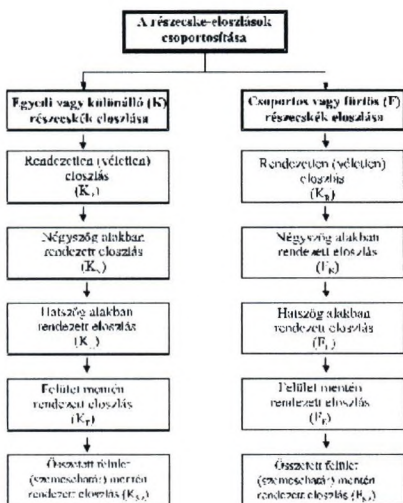
1. A részecske-eloszlások típusai

A részecske-eloszlások csoportosítása és az egyes típusok jellegzetességeinek összehasonlítása nélkülözhetetlen annak érdekében, hogy mélyebben tudjuk elemezni a felhasználás szempontjából lényeges anyagi tulajdonságok (pl. folyáshatár, kopásállóság, törési szívósság) és a szövetszerkezet közötti kapcsolatot. Mindezek mellett a csoportosítás megkönnyíti (megkönnyítheti) az egyes típusok jellemzésére leginkább alkalmas vizsgáló módszerek kiválasztását is.

A csoportosítás alkalmával a térbeli ($3D$) eloszlások statisztikai jellemzőit megtestesítő és véletlenszerűen kiválasztott $2D$ metszeteket elemzünk. Az irodalmi eredményeket felhasználva és a többfázisú szerkezetek csoportosítására általunk alkalmazott módszert továbbfejlesztve a részecske-eloszlások két nagy csoportját különböztetjük meg, úgymint:

1. Egyedi, különálló (K) részecskék szóródása a térben vagy a síkon.
2. Csoportos, fürtös (F) részecske-eloszlás a térben vagy a síkon.

Az első esetben az anyagi részecskék nem képeznek csoportokat, s nem hoznak létre izotróp konglomerátumokat, hanem különállóan helyezkednek el a síkon (1. ábra). Az ilyen részecskék tömegközéppontjai a síkon vagy *Poisson-mintázatot* rajzolnak (ez az ún. rendezetlen, véletlen eloszlás: K_R), vagy négyszögalakban (K_N), hatszögalakban (K_H) rendeződnek, esetleg felületi mintázatot (K_F), netalán szemcsehatárt (K_{SZ}) követnek. A másik nagy csoportba tartozó részecske-eloszlásnál a részecskék izotróp fürtökbe csoportosulnak, és az így létrejött részecskefürtök tömegközéppontjai lehetnek a térben véletlenszerűek (F_R), ekkor



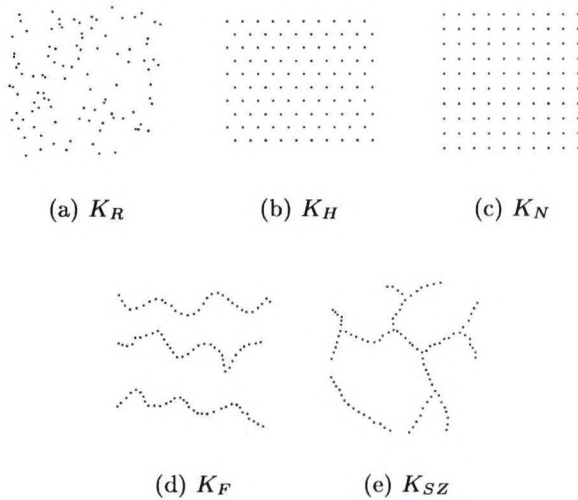
1. ábra. Részecske-eloszlások csoportosítása

rendezetlen fürtös eloszlásról beszélünk. Ha a részecskefürtök tömegközéppontjai négyszög (F_N), hatszög (F_H) mintázatot követnek, akkor rendezett fürtös eloszlásról van szó. Olyan eset is előfordulhat, hogy a fürtök felület mentén (F_F), illetve összetett szemcsehatár (F_{SZ}) mentén sorakoznak föl. A fenti csoportosítást és az alkalmazott jelöléseket a 1. ábra foglalja össze.

A különálló részecske-eloszlások egyes típusainak jellemzésére, valamint az alkalmazott módszerek összehasonlítására számítógéppel generált szövetképeket használtunk. A tesztképek mindegyike $480 \times 480 \text{ pixel}$ volt. A képek fehér háttérben $100 \text{ db } 7 \text{ pixel}$ átmérőjű kör alakú fekete részecskéket tartalmaztak (2. ábra). A véletlen részecske-eloszlást (K_R) úgy állítottuk elő, hogy a részecskék középpontjainak mindkét koordinátáját egy-egy véletlenszám-generátorral meghatároztuk, majd az így nyert pontokra helyeztünk a már említett részecskéket. Hasonló módszerrel hoztuk létre 400 db részecskét tartalmazó tesztalakzatokat is [6]. Számítógépes program segítségével a 7 pixel átmérőjű 100 db részecskét négyszög alakban (K_N) és hatszög alakban (K_H) is elrendeztük. Ezután a felület mentén rendezett eloszlás kétdimenziós képét modelleztük úgy, hogy a felület síkmetszetének megfelelő vonalak mentén helyeztük el a részecskéket (K_F). Végül a szemcsehatár mentén rendezett eloszlás (K_{SZ}) szimulálásakor szemcsehatár mentén sorakoztattuk fel az alkotóelemeket. Az így nyert tesztalakzatokat mutatja a 2. ábra.

2. A kovariancia értelmezése

A zárt halmazok eloszlásának jellemzésére használatos a kovariancia. Amennyiben a szövetszerkezetéről készült mikroszkópos felvételt bináris halmaznak tekintjük, úgy könnyen értelmezhetjük a kovarianciát. Legyen a mikroszkópos felvétel



2. ábra.

Különálló részecskék (a) véletlen eloszlása (b), valasmint hatszög alakban (c) négyszög alakban rendeződve, illetve (d) felület mentén, (e) szemcsehatáron felsorakozva.

bináris halmaza, vagy másképpen bináris képe: \bar{B} . Ha a kérdéses képpont - a mikroszkópos felvételen - a vizsgálat szempontjából érdektelen háttérterülethez tartozik, akkor értéke 0. A bináris képelem abban az esetben 1, ha a képpont az elemezni kívánt objektumra (pl. részecskére, szemcsére, határfelületre) esik. Toljuk el a \bar{B} halmazt \mathbf{h} translációs vektorral. Az eredeti halmaz (\bar{B}) és az eltolts halmaz ($\bar{B} + \mathbf{h}$) szorzatának (metszetének) segítségével a kovarianciához $KOV(\bar{B}, \mathbf{h})$ jutunk [7, 8]:

$$KOV(\bar{B}, \mathbf{h}) = E \left\{ Mes \left[(\bar{B}) \cap (\bar{B} + \mathbf{h}) \right] \right\} \quad (1)$$

Az összefüggésben:

\bar{B} a tanulmányozni kívánt szövetszerkezeti elemek bináris halmaza

\mathbf{h} translációs vektor

Mes halmaz "mértéke", pl. az 1 értékű képpontok száma

E várható érték.

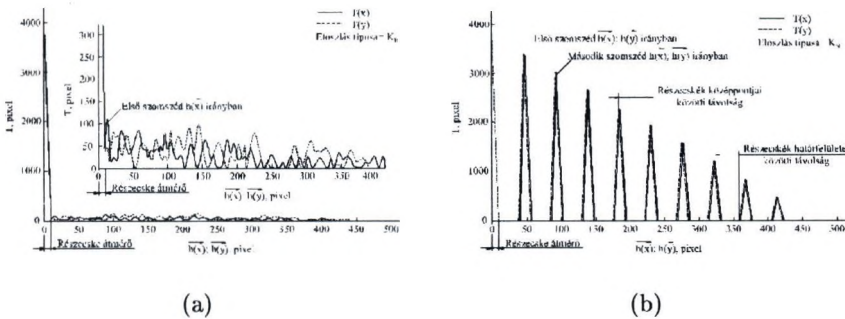
A kovariancia azt jelzi, hogy milyen összefüggés van az eredeti bináris kép, valamint a \mathbf{h} vektorral eltolts bináris kép között. Más szavakkal: a \mathbf{h} vektor irányában van-e valamilyen periodicitása, vagy anizotrópiája a bináris halmaznak. Végeredményben nem más, mint átlagos valószínűsége annak, hogy a \mathbf{h} vektorral eltolts bináris kép és az eredeti kép együttesen tartalmazza a vizsgálandó objektumokat.

A kovariancia nagyon hatékony eszköz a különböző egymásba rétegzett szövetszerkezeti elemek közötti kapcsolat kifejtésére. A függvényt a következő esetekben használhatjuk:

1. Pontok vagy részecskék csoportosulásának jellemzésére.
2. Olyan szövetszerkezet leírására, amelyben különböző méreteloszlású szemcsék vagy részecskék együttesen vannak jelen.
3. Periodikusság vagy pszeudo-periodikusság kimutatására.
4. Anizotrópia jellemzésére. De csak akkor, ha az (anizotróp) objektumok orientáltak, s a szövetszerkezetben textúra figyelhető meg. Nem alkalmas a kovariancia az individuális részecskék vagy szemcsék anizotrópiájának jellemzésére.

3. Különálló tesztalakzatok kovariancia diagramjai

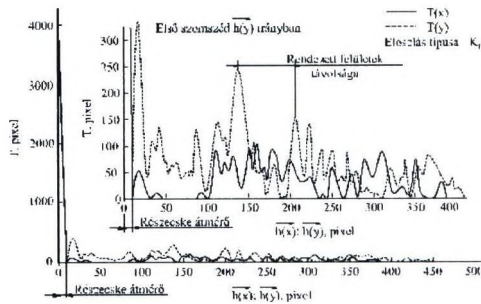
Kovariancia-diagram segítségével a tesztalakzatok jellemezhetők. Saját fejlesztésű programmal a részecskéket tartalmazó bináris képet egy-egy pixelrel x irányban majd y irányban eltolva meghatároztuk az eltolás és az eredeti kép metszetét. Az így kapott területeket a $h(x)$, $h(y)$ transzlációs vektor függvényében ábrázoltuk. A terület első minimuma a részecskeátmérőt mutatja (3. ábra). A véletlen eloszlású részecskénél (K_R) más jellegzetes szélsőértéket (maximumot vagy minimumot) nem figyelhetünk meg, éppen ez jelzi az eloszlás rendezetlen voltát (3. ábra (a) részlete).



3. ábra.

Kovariancia diagram (a) véletlen eloszlású, valamint (b) négyzet alakban rendeződött különálló részecskék esetén.

Teljesen eltérő képet mutat viszont a rendezett, mégpedig a négyzet alakban rendezett részecske-eloszlás (K_N) kovariancia-diagramja (3. ábra (b) részlete). A részecskeátmérő mellett a diagramról jól leolvasható az első, második, harmadik, negyedik szomszéd helyzete $h(x)$ és $h(y)$ irányban, valamint az is, hogy



4. ábra. Kovariancia diagram felület mentén rendezett különálló részecskéknél.

mekkora ezeknek, illetőleg határfelületeiknek a távolsága. Hasonlóan jellegzetes a hatszöges elrendezésű részecskék kovariancia-diagramja (K_H). Ebben az esetben a hatszöges csoportosítás jellegzetességeinek megfelelően a kérdéses részecskétől nem azonos távolságban van az első szomszéd, $h(x)$, illetve $h(y)$ irányban. Természetesen a részecskék középpontjai és határfelületei közötti távolság itt is leolvasható.

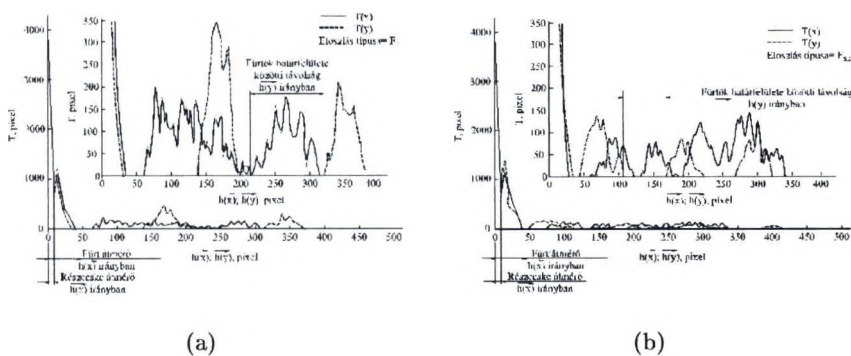
Rendkívül érdekes a felület mentén felsorakozott részecskék kovariancia-diagramja (K_F). Ekkor a $h(x)$, translációs vektor segítségével meghatározott kovariancia-diagram ingadozásán jól felismerhető, hogy ebben az irányban a részecskék véletlenszerűen helyezkednek el, egyszersmind a diagramon a másik (y) irányban jellegzetes csúcsok jelennek meg, a felület mentén rendeződött részecskéknél megfelelően. Ezek a helyi maximumok a véletlen eloszlásra jellemző csúcsoktól lényegesen magasabbak és megegyeznek a felületek középvonalainak távolságával (4. ábra).

3.1. Fürtös részecskék kovariancia-diagramjai

A kovariancia-diagram segítségével a fürtös tesztalakzatok jól identifikálhatók. Az ilyen diagramokon az első lokális minimumhoz tartozó érték nem más, mint a részecskeátmérő, ezután egy viszonylag nagyobb helyi maximum majd minimum következik. Ez a nagyobb csúcs nem jelentkezik a különálló részecskék véletlen eloszlására vonatkozó kovariancia-diagramjánál (3. ábra), vagyis ez a részecskefürtökkel van kapcsolatban. Nyilvánvaló, hogy a második helyi maximum $h(x)$ és $h(y)$ irányban a részecskefürt átmérőjét mutatja. A diagramot fölnagyítva azonban a fürtös részecske-eloszlásra jellemző helyi maximumok és minimumok váltakozása tűnik szemünk elé. A diagramból leolvasható, hogy egyetlen fürtbe átlagosan hány darab részecske tartozik, valamint $h(x)$, $h(y)$ irányban megbecsülhető a fürtök határfelületének távolsága is.

A felületek mentén rendezett részecskefürtökre vonatkozó kovariancia-diagram (5. ábra (a) részlete) jól tükrözi a fürtös részecske-eloszlásra jellemző első helyi minimumot követő erőteljes helyi maximumot, majd utána a nullához tartó görbealakot. Ezután a kovariancia-diagram $h(x)$ és $h(y)$ irányú translációra vonatkozó részlete szignifikánsan különbözik. Míg $h(x)$ irányban a részecskefürtök

véletlenszerűen helyezkednek el, s így a diagram is a véletlen eloszlás képét mutatja, addig $h(y)$ irányban jellegzetes helyi maximumok jelentkeznek. A helyi maximumok közötti távolságból megbecsülhető azon felületek – a síkban vonalak – távolsága, amelyek mentén a fürtök elrendeződtek. Azt is mondhatjuk, hogy a részecskefürtök $h(x)$ irányban véletlenszerűen, $h(y)$ irányban pedig vonal mentén rendezetten helyezkednek el, s ez a jelleg a kovariancia-diagramból egyértelműen kiolvasható.



5. ábra.

Kovariancia diagram (a) felület mentén, valamint (b) szemcsehatáron rendeződött fürtös részecskéknél.

A szemcsehatár mentén rendezett részecskefürtök kovariancia-diagramja (5. ábra (b) részlet) szintén magán viseli a fürtös eloszlás jellemző jegeit, nevezetesen az első helyi minimumot közvetlenül követő helyi maximumot, majd a nullához tartó minimumot. A részecskefürtök elhelyezkedésére vonatkozó további helyi maximumok és minimumok alapján azt mondhatjuk, hogy ez a diagram a határfelületek mentén rendezett részecske-eloszlásokhoz hasonlít, csak itt a részecskefürtök nemcsak egy irányban, hanem mind $h(x)$, mind $h(y)$ irányban rendezettséget mutatnak. A diagramról a fürtök határfelületei közötti távolság $h(x)$ és $h(y)$ irányban egyaránt leolvasható. Ugyanakkor csak kevésbé különbözik a részecskefürtök véletlen eloszlására vonatkozó diagramjától, s az attól való megkülönböztetése nem egyszerű feladat.

4. Következtetések

Számítógéppel generált teszt alakzatok kovariancia diagramjai alapján megállapítottuk, hogy a kovariancia sokoldalúan jellemzi a különálló részecskék egyes eloszlástípusait. Bebizonyítottuk, hogy a véletlen eloszlású részecskék diagramja szignifikánsan különbözik az összes többitől.

Rámutatunk, hogy teljesen eltérő képet mutat a négyszög alakban rendezett részecskék kovariancia diagramja, amelyről a részecske átmérő mellett az első, a második, a harmadik szomszéd helyzete is leolvasható $h(x)$, $h(y)$ irányban.

A felület és a szemcsehatár mentén rendezett különálló részecskék kovariancia diagramja eltér a véletlen elrendeződésű részecskék diagramjától. A diagramok alapján becsülhető azon felületek távolsága, amelyek mentén a részecskék rendeződtek. A diagramok alapján azt is megállapítható, hogy a felület és a szemcsehatár mentén rendeződött részecskéket tartalmazó szövetszerkezetek kovariancia diagramjai és a véletlen elhelyezkedésre vonatkozó diagramok között csak kisebb mértékű a különbség.

A kovariancia diagram segítségével a különböző típusú fürtös részecske eloszlások jól identifikálhatók. A fürtös részecskék diagramja szignifikánsan különbözik a különálló részecskék diagramjától: az első minimumot követő helyi maximum megjelenése, s a maximumok, minimumok váltakozása jellemzi az ilyen diagramokat. A diagramokról a részecskefürtök átmérője, a fürtök határfelülete közötti távolság, sőt a fürtök belsejében lévő átlagos részecske darabszám becsülhető.

Hivatkozások

1. Vugt L., Froyen L.: Gravity and temperature effects on particle distribution in Al-Si/SiC composites. *Journal of Materials Processing Technology*. 104 (2000) p. 133.
2. Olszówka-Myalska A., Szala J., Cwajna J.: Characterization of reinforcement distribution in Al/(Al₂O₃)_p composites obtained from composite powder. *Materials Characterization*. 46 (2001) p. 189.
3. Spowart J. E., Maruyama B., and Miracle D. B.: Multi-scale characterization of spatially heterogeneous systems: implications for discontinuously reinforced metal-matrix composite microstructures. *Materials Science and Engineering*. A307 (2001) p. 51.
4. Yotte S., Breyse D., Riss J., Ghosh S.: Cluster characterization in a metal matrix composite. *Materials Characterization*. 46 (2001) p. 211.
5. He D. Ekere N. N., Cai L.: New statistic techniques for structure evaluation of particle packing. *Materials Science and Engineering*. A298 (2001) p. 209.
6. Gácsi Z., Kovács J., Pieczonka T.: Characterization of Particle Arrangement Using the Radial Distribution Function. 3rd International Powder Metallurgy Conference. Gazi University, Ankara, Turkey. September 4-8, 2002. Turkish Powder Metallurgy Association. p. 542. (CD ROM)
7. Soille P.: *Morphologische Bildverarbeitung. Grundlagen, Methoden, Anwendungen*. Berlin. Springer. (1989) p. 75.
8. Susagna F., Yotte S., Riss J., Breyse D., Ghosh S.: Covariance and Spatial Distribution of Particles in a Metal Matrix Composite. *STERMAT'2000*. Krakow, Poland. Proceedings. (2000) p. 397.

Robust Structure from Motion by Outlier Rejection

Levente Hajder^{1,2}, Dmitry Chetverikov^{1,3}, and István Vajk²

¹ Computer and Automation Research Institute, HAS
hajder@sztaki.hu

² Department of Automation and Applied Informatics,
Budapest University of Technology and Economics

³ Department of Informatics, Eötvös Loránd University

Abstract. It is widely known that, for the affine camera model, both shape and motion can be factorized directly from the measurement matrix constructed from 2D image point coordinates. However, the classical algorithm for Structure from Motion (SfM) proposed by Tomasi and Kanade [1] is not robust: outliers, that is, incorrectly detected or matched feature points can destroy the result.

In this paper, we present a novel robust procedure for affine SfM recovery from a video sequence. The theory of the approach is described, its advantages and limits are discussed. The outlier rejection scheme is tested on synthesized video sequences.

1 Introduction

The structure from motion problem (recovering scene geometry and camera motion from a video sequence) has attracted attention of the computer vision community over last years. The original factorization procedure of Tomasi and Kanade [1] can generate three-dimensional coordinates of an object from a sequence of tracked feature points of the object. The input of the method is the 2D coordinates of the features, while the output is 3D coordinates of the points and the base vectors of the camera planes in all frames except the first one. In the literature, the three-dimensional data is called the **structure data**, the base vectors are called the **motion information**.

The method of Tomasi and Kanade [1] can only compute structural information in the case of a single rigid object. The method was published in 1992. It assumes orthographic projection. Recent studies [5–7] try to extend the theory to the nonrigid case as well, but in this article we only discuss the rigid case. There are also papers [13], [14] that extend the theory to the paraperspective and real perspective cases. The original method contains a nonlinear optimization. It was simplified to a linear LMS method by Morita and Kanade [4].

A key problem of the factorization is its **outlier sensitivity**. The input of the method is typically hundreds of tracked 2D feature points. Approximately, at least 5% of the points tracked by a standard feature tracker (e.g., [11]) are outliers. There are ways [2, 3, 8–10] to remove outliers, but these methods do

not work in all cases. The weakness of the existing methods is that the rank of several matrices must be calculated. Without noise (or with a very small noise), the calculation is easy, but with real noise, it is impossible in many of cases.

For this reason we developed a novell, general method that can filter the input data and can remove outliers.

2 Basic idea

SfM methods provide structural and motion information. The goal is to compute the three dimensional coordinates of an object from tracked outlier-free features. Our idea is to utilize the motion information, because outlier detection can be done based on the camera motion.

The input data to our procedure is the 2D coordinates of the points. Assume that P points are available. If we randomly select k points from these P points and run the Tomasi-Kanade factorization on this data, we obtain the base vectors of the camera in all frames. If we repeat the test selecting other k points, the motion information obtained is the same if the input data does not contain noise and outliers, that is, data comes from the same object.

If the input data contains noise, and we select the points randomly, the motion information is not the same but just similar. If the data contains outliers, the motion data is **inconsistent**.

3 Analysis of base vectors by simulation

We have analysed synthetic data to demonstrate the correctness of our idea. A single 3D object was considered. The points of this object were generated by a Gaussian random number generator. The mean was zero, the standard deviation σ was a parameter. The object was rotated to 30 different rotations. The angles of the first frame were $\alpha = 0$, $\beta = 0$, $\gamma = 0$, while the angles of the 30th frame were $\alpha = 1.0$, $\beta = 1.0$, $\gamma = 1.0$ (given in radian). The angles of the inner frames were calculated by linear interpolation.

After generating the 3D coordinate the following algorithm was applied:

1. Add Gaussian noise to the rotated coordinates (mean= 0, variance= r)
2. Select four points from the set.
3. Calculate motion and structure by the Tomasi-Kanade factorization.
4. Create a **concatenated vector** by concatenating the base vectors of the camera planes.
5. Repeat steps 1...4. Steps 1-5 yield two concatenated vectors.
6. Calculate the norm of the difference between the two concatenated vectors.
7. Repeat steps 2-6 200 times.
8. Calculate the error as the average of the 200 norms obtained.

This test was repeated with different levels of noise. Figure 1 shows the average square error versus the noise. (The horizontal axis is $\frac{100r}{\sigma}$.) For low noise, the error is significantly lower than for high noise.

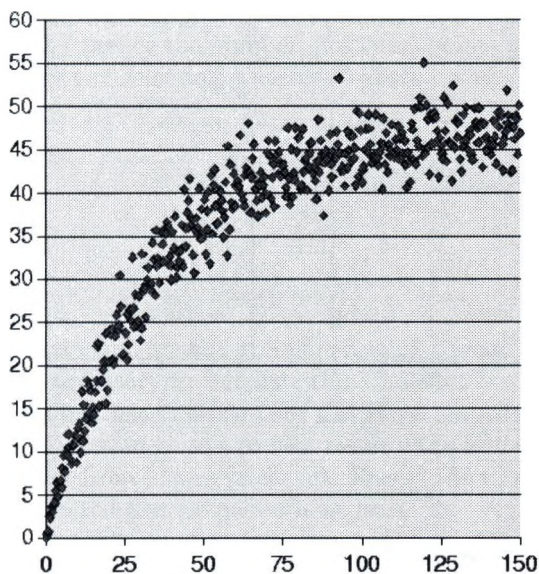


Fig. 1. Errors of motion estimation versus 2D noise

The following conclusion has been drawn : After factorization, the motion data can be used to determine whether 4 points belong to the same object.

4 Algorithm for outlier detection

The proposed outlier detector operates as follows.

Input: Two-dimensional points tracked by a feature tracker in a sequence of images. If a feature was not tracked through all frames of the sequence, it is discarded. (We used the KLT tracker [11].)

Steps of the algorithm:

1. Execute the following iteration for all feature points:
 - (a) Pick the next point, select three other points randomly. Do Tomasi-Kanade factorization. Concatenate the obtained motion vectors.
 - (b) Repeat step (a) m times ($m = 50$ was used).
 - (c) Pair m concatenated vectors in all possible ways. $n = \frac{m \times (m-1)}{2}$ pairs exist. (For $m = 50$, $n = 1225$.) For each pair, calculate the error as the norm of the difference between the two vectors.
 - (d) Sort the errors in the ascending order. (See figure 2 for sample sequences of sorted errors for a good point and an outlier.)
 - (e) Select the b^{th} element of the sequence. This number represents the **inconsistency** of the feature.
2. Sort the inconsistency values of feature points in the ascending order. (See figure 3 for an example.)

3. If the inconsistency of a feature point is high, discard this point.

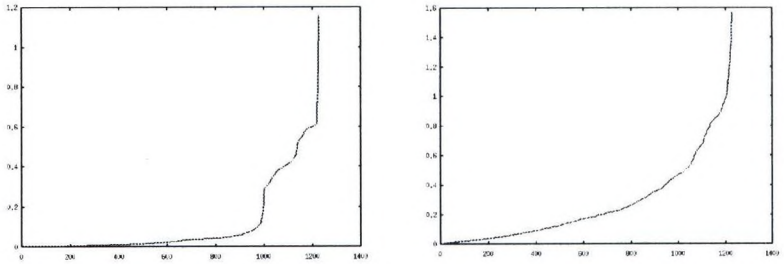


Fig. 2. Ordered errors of a good feature point and an outlier

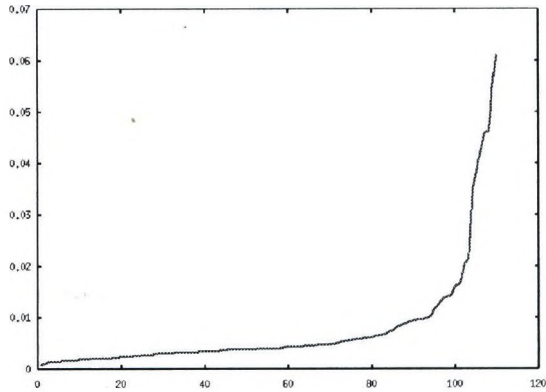


Fig. 3. Ordered inconsistency values of feature points

5 Examination of the algorithm

We have to address two important issues: (1) The rate of outliers that can be handled by the method. (2) The number of iterations needed.

These issues are examined below.

5.1 What is the rate of outliers that can still be handled?

To answer this question, we have to calculate the probability of outlier-free selection of points.

If k feature points were selected, the number of all feature points is p , and the number of outliers is l (hence the number of correct points is $(p-l)$). Then the estimated probability v of selecting k correct feature points can be calculated as

$$v = \frac{\binom{p-l}{k}}{\binom{p}{k}} \quad (1)$$

If $k \ll p$ we have

$$v \approx (p-l)^k \quad (2)$$

Note: The optimal value for k is 4. Smaller value cannot be selected because at least 4 points are necessary to calculate the 3D structure. On the other hand, increasing k reduces v .

Let $c = \frac{b}{n}$. The algorithm gives a correct result if the b^{th} element of the sorted sequence is calculated from inliers (step 1c). This holds when the proportion of inliers among the selected feature points is at least \sqrt{c} .

Calculating \sqrt{c} is a binomial distribution problem. The distribution shows the probability of selecting a given number of inliers. The probability of selecting \sqrt{c} or less tracked points is determined by the Chebyshev's inequality:

$$P\left(\|\xi - nv\| \geq \lambda\sqrt{nv(1-v)}\right) \leq \frac{1}{\lambda^2}, \quad (3)$$

where λ is an arbitrary value, P the probability. (The parameters of the binomial distribution are inserted.)

This means that

$$\lambda \leq \sqrt{\frac{1}{P}} \quad (4)$$

A limit for λ can also be determined by the inner part of the Chebyshev's inequality ($\xi = \sqrt{c}$), if we would like to calculate a good value for the b^{th} element:

$$\lambda_{\text{limit}} = \frac{nv - \sqrt{c}}{\sqrt{nv(1-v)}} \quad (5)$$

From (4) and (5) we obtain:

$$\sqrt{\frac{1}{P}} \geq \frac{nv - \sqrt{c}}{\sqrt{nv(1-v)}} \quad (6)$$

This can be re-written as

$$\left(n^2 + \frac{n}{P}\right)v^2 - \left(2n\sqrt{c} + \frac{n}{P}\right)v + c \leq 0 \quad (7)$$

If this inequation is not valid, there are too many outliers in the data, and a correct result cannot be obtained with the probability P .

5.2 How many iterations are needed?

By re-writing the inequation (7), we obtain

$$v^2 n^2 + \left(\frac{v^2}{P} - 2v\sqrt{c} - \frac{v}{P} \right) n + c \leq 0 \quad (8)$$

From this inequality, one can determine the minimal number of iteration by calculating the limit for n .

6 Experimental results

We used the 'hotel' sequence from the CMU VASC Image Database [12]. The KLT feature tracker [11] tracked 110 features. The proposed algorithm successfully removed the outliers. Due to a low threshold value used, the method discarded a small number of correct features as well. This is illustrated in figure 4, where 'x' marks a detected outlier, light spot an inlier. All the preserved points are correct.

7 Conclusion

In this paper we presented a novell method that robustifies a standard factorization method for the Structure from Motion problem. It was shown that in the case of orthographic projection the separation of correctly tracked features from outliers can be achieved by analysis of the motion component of the input data. An example was shown to demonstrate the operation of our algorithm.

Acknowledgement

This work was supported by the Hungarian Scientific Research Fund (OTKA) grants T038355 and M28078.

References

1. C. Tomasi, T. Kanade. Shape and motion from image streams under ortography: A factorization approach. *International journal of Computer Vision*, 9(2): 137-154, 1992.
2. J.P. Costeira, T. Kanade: A multibody factorization method for independently moving-objects. *International journal of Computer Vision* 29(3): 159-179, 1998.
3. M. Trajkovic, M. Hedley: An Algorithm for Recursive Structure and Motion Recovery under Affine Projection. *Third Asian Convergence on Computer Vision* (2) 376-383 1998.
4. T. Morita, T. Kanade: A Sequential Factorization Method for recovering Shape and Motion from Image Streams. *ARPA Image Understanding Workshop* 1177-1188 1994.



Fig. 4. Results for 'hotel' sequence. 'x' is a detected outlier.

5. M. Brand, R. Bhotika : Flexible Flow for 3D Nonrigid Tracking and Shape Recovery CVPR, 2001.
6. L. Torresani, D.B. Yang, E.J. Alexander, C. Begler : Tracking and Modeling Non-Rigid Objects with Rank Constraints. CVPR, 2001.
7. M. Brand Morphable 3D Models from Video. CVPR, 2001.
8. J. Costeira, T.Kanade : A Multi-Body Factorization Method for Motion Analysis. ICCV, 1995.
9. Y. Weiss : Segmentation using eigenvectors: a unifying view. ICCV, 1999.
10. L. Zelner-Manor, M. Machline, M. Irani : Multi-body Segmentation: Revisiting Motion Consistency. ECCV, 2002.
11. J. Shi, C. Tomasi: Good features to track. IEEE CVPR Seattle, 1994.
12. CMU VASC Image Database <http://vasc.ri.cmu.edu/idb/>
13. C. J. Poelman, T. Kanade : A Paraperspective Factorization Method for Shape and Motion Recovery. IEEE PAMI, 1993.
14. P. Sturm, B. Triggs : A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. ECCV, 1996.

Aktív kontúrok és Fast Marching eljárás alkalmazása az orvosi képfeldolgozásban

Hajder Levente¹, Kardos István², Csetverikov Dmitrij^{1,2}, Renner Gábor¹

¹ MTA Számítástechnikai és Automatizálási Kutatóintézete
hajder@sztaki.hu

² Eötvös Lóránd Tudományegyetem

Kivonat Az MTA SZTAKI Geometriai Modellezés és Számítógépes Látás Laboratóriuma képfeldolgozással, számítógépes látással, alakfelismeréssel, mérnöki visszafejtéssel, görbék és felületek számításával foglalkozik.

Egy új feladat kapcsán kezdtünk el foglalkozni orvosi alkalmazások fejlesztésével. Feladatunk volt, hogy emberi térdről készült MR felvétel alapján a csont kontúrját (és később a porc kontúrját is) állítsuk elő. Ez a cikk a projekt aktuális állását mutatja be. Ismertetünk két módszert, amelyek alkalmasak a porc határfelületének előállítására: az aktív kontúrok alkalmazását és az ún. Fast Marching eljárást. A módszerek gyenge minőségű, zajos MR felvételek alapján képesek a csont kontúrjának meghatározására.

1. Bevezetés

A térdízület mozgásformáit elsősorban az egymáson elmozduló porcfelületek geometriai alakja határozza meg. Ezen alak tanulmányozásához a legmegbízhatóbb információkat a térdízületről különböző orvosi képalkotási módszerekkel (MR, CT) nyert képek illetve azok sorozata szolgáltatja. E képek az ízületi struktúra kétdimenziós síkmetszeteit ábrázolják, az eltérő struktúráknak megfelelően különböző szürkeségi szint árnyalatokban. Következésképpen a csontfelület és a porcfelület egymástól és a környezetüktől eltérő szürkeségű kontúrok formájában mutatkoznak.

Kontúrdetektáló eljárásokat fejlesztettünk ki a kontúrgörbék sorozatának MR felvételsorozatokból történő meghatározására. Az eljárás ugyan metszetenként dolgozik, de felhasználja a szomszédos metszetek kontúrjaiból nyerhető információkat. Jelenlegi formájában a csontkontúrok meghatározására alkalmas: a porckontúrok detektálásához (a bonyolultabb struktúrának megfelelően) további kiegészítésére van szükség. A kontúrgörbék meghatározása a teljes képsorozaton lényegében automatikusan történik: mindössze arra van szükség, hogy az első képen néhány kontúrponthoz interaktív módon lokalizáljunk.

2. Feladat

A csontkontúr detektálás célja, hogy az eredeti, DICOM formátumban meglévő MR felvételekből a csont kontúrjának háromdimenziós modelljét előállítsuk.

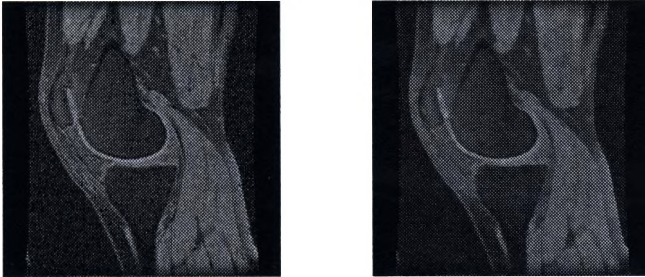
A feladat bemenete (tetszőleges szögben behajlított) térdről készült MR felvétel, lehetőleg axiális vagy szagitális irányokban. A felvételek DICOM formátumú fájlokban állnak rendelkezésre. Egy-egy fájl egy-egy metszetet tartalmaz, amelyeken az információt - kétdimenziós képeknek megfelelően - $0 \dots 4095$ tartományba eső szürkeségi értékek reprezentálják. A feladat ezekből a kétdimenziós képekből a csont határát felismerni, majd a határfelületet alkotó pontokat szöveges állományként elmenteni.

3. A képek szűrése

A képek eredetileg $0 \dots 4095$ intenzitásértéket vehetnek fel a DICOM formátumának megfelelően. Ezt az ImageJ [1] programhoz letölthető DICOM olvasó plug-in segítségével $0 \dots 255$ intenzitásértékű szürke képpé konvertáltuk.

A képek sajnálatos módon nagyon zajosak, ezért mindenképpen muszáj megszűrni a konvertált képeket. A szűrésnél fontos szempont, hogy a határokat ne mossa el, helyüket ne változtassa meg, mivel a feladat ezeknek a kontúroknak a minél pontosabb meghatározása.

Ezen szempontok alapján az adaptív szimmetrikus átlagszűrőt választottuk. A szűrés eredményét a 1 ábrán tekinthetjük meg.



1. ábra. Eredeti és szűrt felvétel

4. Detektálás aktív kontúrok segítségével

A aktív kontúros csontdetektáló algoritmus [4] minden egyes kétdimenziós szeleten meghatározza a csontok kontúrját, a háromdimenziós csontkontúrt a kétdimenziós kontúrok pontjaiból számítja ki.

A csontkontúr detektálásához a meglévő információt az a tény szolgáltatja, hogy a csont és a körülötte levő szövetek (pl. porc, izom, ín) intenzitása az MR felvételeken különböző. A csont sötétebb árnyalatú, míg a környező szövetek valamivel világosabbak (ahogyan az a 1 ábra bal oldali képén látszik).

Az egyes szeletek síkjában a kontúr meghatározása az alábbi három lépésben történik:

1. A kezdeti képen a kontúr kijelölése interaktív módon történik.
2. Második lépés a kontúr finomítása számítógépes algoritmus segítségével.
3. Az előző lépésben pontosított kontúrt alkotó pontok szűrése, majd azokra görbe illesztése.

A 2. és 3. lépést ismétljük oly módon, hogy amíg az interaktív kijelölést finomítottuk a kezdeti szeleten, a további szeleteken manuális kijelölés helyett az előző szeleten megkapott kontúrt vesszük alapul, majd azt a második pont szerint finomítjuk, és a harmadik lépés szerint görbét illesztünk rá. Az eljárást a kiinduló szelettől fogva a szomszédos szeleteken keresztül az összes szeletre iteráljuk. (Mivel a kiinduló szeletnek két szomszédja lehet, az eljárást csökkenő és növekvő szeletszám mellett is el kell végezni.)

4.1. Csontkontúr kijelölése interaktív módon



2. ábra. Kézzel kijelölt kontúr

A csontkontúr detektálásának első lépése a kézzel való kijelölés. A felhasználó megadhat néhány pontot, amelyek a kontúrt jelölik ki, és azt a számítógép összeköti. A módszer lehetőséget ad arra, hogy a pontok kijelölése közben a felhasználó ellenőrizni tudja az aktuális görbét. Ez úgy történik, hogy az eljárás a kurzor pozícióját és az előző pontot összeköti (az algoritmust lefuttatja), és az eredményt kirajzolja. A kurzor mozgatására az előző eredményt letörli, az új pozíciónak megfelelő görbét újra kiszámolja, és azt megjeleníti.

Az összekötés során a program az egyes pontokat külön-külön összeköti egymással. Az összekötés úgy történik, hogy elindul az egyik kontúrt alkotó pontból, majd megnézi, hogy a szomszédos pontok közül melyik az, amelyik leginkább határfelülethez hasonlít. A szomszédos pixelek közül az összehasonlítás során csak azokat veszi figyelembe, amelyek a másik, manuálisan kijelölt ponthoz közelebb vannak.

Ezt az eljárást addig iterálja, amíg el nem jut az egyik kijelölt ponttól a következőig.

Annak meghatározásához, hogy egy pont mennyire minősül határfelületnek, mértéket határoztunk meg:

Egy pixel 8 közvetlen szomszédját két összefüggő részre osztjuk (azaz 4-4 szomszédos pixelre), ezen a két összefüggő részen kiszámoljuk az intenzitásatlagozt. A két összefüggő részre bontást az összes lehetséges módon elvégezzük (összesen négyféleképpen lehet), és vesszük a különbségek közül a maximális értékűt.

Az így kapott érték adja meg, hogy az adott pixel mennyire határfelület. Minél nagyobb ez az érték, az adott pont annál inkább határfelületnek minősül.

Ennek az eljárásnak a segítségével kétdimenziós kontúrt kapunk. A kontúrt szomszédos pixelek sorozataként írja le az algoritmus.

A 2 ábrán egy kézzel kijelölt kontúrgörbe látszik. Körülbelül tíz pont manuális kijelölése alapján az algoritmus helyesen ki tudja jelölni a csont kontúrját. Ha két pontot rosszul köt össze, azt a felhasználó látja, és ki tudja javítani az eredményt azzal, hogy a következő pontot közelebb jelöli ki.

4.2. Csontkontúr finomítása

Minden egyes szeleten egy már létező kontúrból indulunk ki (amelyet vagy az előző szeletről hoztunk át, vagy kézzel jelöltünk ki). A következő feladat az adott szeleten lévő intenzitás-információ alapján ezt a kontúr úgy finomítani, hogy az minél jobban illeszkedjen a csont határára.

Azt a kontúrt, amely a finomítás alapját képezi, pontok határozzák meg.

A csont kontúrjának meghatározására iteratív eljárást alkalmazunk, amely az alábbi algoritmus alapján történik: Az algoritmus első lépésben a kontúr al-kotta pontokat megritkítja, ill. besűríti úgy, hogy a pontok közötti távolság egy megadott tartományon belül legyen (a tartomány mérete beállítható).

Ezek után az összes így kapott ponton végigszalad a módszer, és megpróbálja őket egyesével, egyszerre egy pixel távolságra mozgatni. A mozgatás úgy történik meg, hogy az alábbi feltételekből (súlyozottan) összeálló jóságfüggvény a mozgatás következményeként a leginkább csökkenjen:

- határfelületen mozogjon a pont (a határfelület mértékét a kézi kontúrkijelölésnél említett eljárás adja)
 - a görbület legyen minél kisebb (a kontúrgörbe sima legyen)
- A görbületet az alábbi, közelítő összefüggés segítségével számoljuk:

$$g = \frac{x_{norm}^2 + y_{norm}^2}{(x_{prev} - x_{norm})^2 + (x_{next} - x_{norm})^2 + (y_{prev} - y_{norm})^2 + (y_{next} - y_{norm})^2}$$

ahol g a görbületet,

- x_{prev} és y_{prev} az adott kontúrpoint előtti kontúrpoint két koordinátáját,
- x_{next} és y_{next} az adott kontúrpoint utáni kontúrpoint két koordinátáját jelöli,
- x_{norm} és y_{norm} pedig a következőképpen számítható ki: $x_{norm} = x_{prev} - 2x + x_{next}$, illetve $y_{norm} = y_{prev} - 2y + y_{next}$

Ez utóbbi összefüggésben x és y az aktuális kontúrpoint két koordinátáját jelöli.

- A szomszédos pontokon keresztül húzott egyenesen az intenzitásváltozás minél kisebb legyen.
- A pontok ne legyenek túl közel egymáshoz (ezzel biztosítjuk, hogy a pontok ne tömörödjének össze egy helyre).

A jóságfüggvény kritériumainak súlyait tapasztalati úton határoztuk meg.

Ezeknek a kritériumoknak megfelelően a módszer az összes pontot megpróbálja jobb helyzetbe mozgatni. Egyszerre mindig csak egy pontot mozgat, a

többbit változatlanul hagyja. Egy pont egy mozgatósi lépésben csak szomszédos pixelre kerülhet.

Az eljárás az összes ponton végigmegy (egymás után haladva), majd a sor végétől újra az elejére megy, és újra végigmegy az összes ponton. Az iteráció számát be lehet állítani (az eddigi próbák tapasztalatai alapján célszerű legalább 50-es értékkel dolgozni, hogy a kontúr eleget változhasson, és megtalálja ideális helyét).

4.3. Görbe illesztése

Miután a kontúrfinomító eljárás a kezdeti, durva kontúrból egy jóval pontosabb kontúrt határoz meg, ez a kontúr még tartalmazhat hibákat: előfordulhat, hogy a képen egy bizonyos szakaszon nincsen a kontúrra vonatkozó információ. Az eddigiekben ismertetett algoritmus feltételezi, hogy a kontúr megszakítás nélkül látszik minden egyes szeleten, ez pedig nem minden esetben felel meg a valóságnak. Előfordulhat, hogy egy rövidebb-hosszabb részen nem látszik kontúr, noha valójában ott van, csak ez az információ az MR felvételen nem jelenik meg. Ebben az esetben az algoritmus az adott részen is ki fog jelölni pontokat, ezeket azonban el kell dobnunk.

Mindezek miatt utószűrést kell végeznünk. Ennek során az összes pontra megvizsgáljuk, hogy az adott pont határfelületen helyezkedik-e el. Ennek érdekében a korábbi határfelületet adó mértékünkre egy beállítható küszöbszámot határozunk meg, és amelyik pont a küszöb felett van, azt megtartjuk, a többbit eldobjuk. A küszöbértéket tapasztalati úton határozhatjuk meg.

A csontkontúrok geometria tulajdonságainak kiértékeléséhez folytonos görbére van szükség, ezt a megszürt pontsorozatra alkalmazott spline-approximációs módszerrel állítjuk elő. Az általunk felhasznált eljárás az irodalomban fellelhető approximációs módszereknek ([5]) a feladat sajátosságainak megfelelően kialakított változata ([6]).

A kontúrdetektáló működését (a végleges kontúrt) a 3 ábra mutatja meg.



3. ábra. A végleges, sima kontúr

5. Fast Marching eljárás

A Fast Marching eljárás ([2] és [3]) alapvetően különbözik az aktív kontúros megközelítéstől. Az ötlet, hogy egy kezdeti kontúrból - amely általában egyetlen

ponttá zsugorított alakzat szokott lenni - elkezdünk távolodni. A távolodás függ az adott pont gradiensétől. Minél nagyobb a gradiens, azaz minél nagyobb a változás a kép adott pontjában, annál lassabban terjed az alakzat. Az eljárás maga egy iteráció, vagyis minden egyes lépésben egy-egy újabb kontúrt kapunk. Az összes pixelhez feljegyezzük, hogy mennyi idő múlva ér a hullám oda (ld. 4 ábra). Végeredményként azoknak a pixeleknek a halmazát kapjuk, amelyeket véges időn belül (pontosabban egy bizonyos küszöbidőnél hamarabb) elér a hullám.

5.1. Algoritmus

Az eljárás lelke a kontúr távolodási sebességének meghatározása. Minden egyes pontra ki kell számolni a kontúr bővülésének sebességét. Ezt a sebességet az alábbi összefüggés segítségével számoljuk ki:

$$g_I(x, y) = (1 + |\nabla(I(x, y))|)^{-1}$$

ahol $g_I(x, y)$ az adott (x, y) pontban a kontúr növekedésének sebessége.

Mindez azt jelenti, hogy minél nagyobb egy pontban a gradiens, annál kisebb a sebesség. Azaz ott, ahol a képen az intenzitás nagyon változik, a terjeszkedés lelassul (esetleg megáll), míg azokon a pontokon, ahol kicsi a változás (az intenzitásfüggvény vízszintes síkhoz közelít), ott nagyon gyors a terjeszkedés.

Az algoritmus lépésről lépésre az alábbiak szerint működik:

1. Inicializálás

A pontokat három kategóriákra osztjuk

- Belső pontok: ezek azok a pontok, amelyek a kiindulási kontúron belül, vagy azon rajta vannak. A mi implementálásunkban kiinduláskor a kontúrt egyetlen pont reprezentálja.
- Szomszédos pontok: Azok a pontok, amelyek a belső pont közvetlen szomszédai (4-szomszedságot figyelve).
- Távoli pontok: Azok, amelyek a belső pontokkal nincsenek közvetlen kapcsolatban.

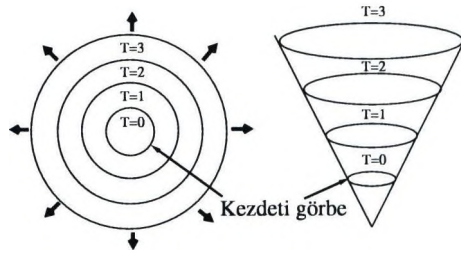
2. Terjeszkedés

- (a) Az összes lehetséges szomszédos pontra ki kell számítani a kontúr érkezési idejét.
 - (b) A legkisebb értékű szomszédot át kell tenni a belső pontok közé.
 - (c) Az új belső pont szomszédjait átesszük a szomszédos pontok közé.
3. Az előző lépést addig ismételjük, amíg az új belső ponthoz tartozó időérték egy megadott határértéket meghalad.

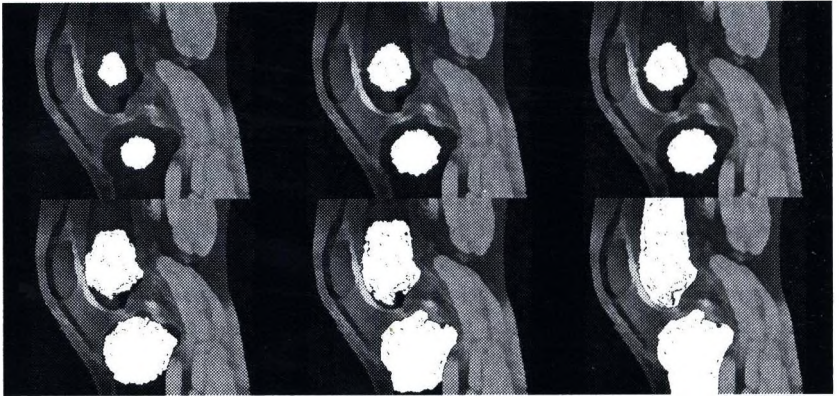
5.2. Példa

A 5. ábrán láthatunk a Fast Marching metódusra példát. Az ábra a belső pontok terjeszkedését mutatja különböző pillanatokban. Az utolsó állapot a végeredményt jeleníti meg.

A példán két csontnak a kontúráját határoztuk meg. Ez az eredeti eljáráshoz képest annyi változást jelent, hogy az inicializáláskor két különböző kiindulási pontot adtunk meg (ebből kifolyólag két különböző területet kapuk eredményként).



4. ábra. Alakzat terjedése az időben



5. ábra. A detektált csontterület terjeszkedése

Köszönetnyilvánítás

Ez a munka az Országos Tudományos Kutatási Alapprogramok (OTKA T038355 és M28078) támogatásával jöhetett létre.

Hivatkozások

1. ImageJ honlap: <http://rsb.info.nih.gov/ij/>
2. Fast Marching ismertető: <http://w3.impa.br/~apeixoto/fastmarching.htm>
3. J. A. Sethian :Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science, Cambridge University Press, 1999.
4. A. Blake, M. Isard :Active Contours, Springer, 1998.
5. J. Hoschek and D. Lasser, Fundamentals of Computer Aided Geometric Design, A.K. Peters, Wellesley, Massachusetts, 1993.
6. Renner G.: Interpolation with G2 cubic curves. In: The Mathematics of Surfaces V., Edinburgh, 1992. Ed.: R.B. Fisher, Oxford, Clarendon Press, pp. 385-391. 1994.

Analytical and approximation properties of neighborhood sequences

András Hajdu¹ and Lajos Hajdu²

¹ Institute of Informatics, University of Debrecen, hajdua@inf.unideb.hu

² Institute of Mathematics, University of Debrecen, hajdul@math.klte.hu

Abstract. Distance functions are used in several parts of digital geometry, including digital image processing. They are usually introduced via digital motions. That is, we can move in the digital space from one point to another, if they are neighbors in some sense. By allowing arbitrary mixture of motions based on the well-known 4- and 8-neighborhood relations, we obtain the concept of neighborhood sequences. Such sequences and the distance functions generated by them have important applications in image processing, e.g. in distance transforms. In this survey we briefly summarize our recent results concerning the analytical and approximation properties of neighborhood sequences (see [13], [14]). We show how a velocity (a norm-like concept) can be introduced on the set of such sequences in a way, which fits its algebraic structure. We also show how these sequences (or the associated distance functions) can be used to approximate the Euclidean distance.

1 Introduction

In many parts of digital geometry, including digital image processing, distance functions play a very important role. Such functions naturally arise as soon as some distance measurement is needed. As illustrative examples we mention [15] and [16] for multidimensional image segmentation, [19] for medial axis transformation, and those many areas which use distance transformations (see [1, 3] for the general theory of distance transformations, and e.g. [2, 12] and the references given there for characteristic applications).

Distance functions are usually introduced via digital motions. That is, we can move in the digital space from one point to another, if they are neighbors in some sense. Rosenfeld and Pfaltz [24] defined two types of motions in \mathbb{Z}^2 , the cityblock and chessboard motions. The first one allows only horizontal and vertical steps, while the second one diagonal movements, as well. By counting the number of steps needed to get from one point to another, these motions give rise to the distances d_4 and d_8 , respectively. To obtain a better approximation of

¹ Research supported by the OTKA grants T032361 and F043090.

² Research supported in part by the Netherlands Organization for Scientific Research (NWO), by the OTKA grants T042985, F034981 and F043090, and by the FKFP grant 3272-13/066/2001.

the Euclidean distance they recommended the alternate use of the cityblock and chessboard motions. The sequences obtained by an arbitrary mixture of these motion are called octagonal neighborhood sequences.

The literature of octagonal neighborhood sequences is very rich. Here we only mention a few papers which are relevant for our purposes. Das et al. [7, 8] studied periodic octagonal neighborhood sequences, and generalized them to arbitrary finite dimension. Later, Fazekas et al. [11] extended the investigations to general octagonal neighborhood sequences, in any finite dimension, and also in the infinite dimensional case. To any octagonal neighborhood sequence A one can attach a distance function $d(A)$, just as in case of the cityblock and chessboard motions. Some general properties of the functions $d(A)$ were studied by Das et al. [7, 8] in the periodic case, and by Fazekas et al. [11] and Nagy [22] in the general case. For periodic octagonal sequences, in [8] the authors provided a criterion to decide when $d(A)$ is a metric. This result was extended by Nagy [22] to the general case.

There is an important natural ordering relation on the set of octagonal neighborhood sequences, which can be introduced in the following way. Given two sequences A and B , we say that A is faster than B , if for every pair of points p, q we have $d(p, q; A) \leq d(p, q; B)$. In other words, A spreads faster in the digital space than B . The properties of this ordering relation were studied in the periodic case by Das [5] in 2D and by Fazekas [10] in 3D, and in the general case by Fazekas et al. [11], in arbitrary dimension. Summarizing these investigations, it turns out that in general "faster" is only a partial ordering. However, in certain applications it can be useful to compare any two neighborhood sequences, i.e. to decide which one spreads faster. For this purpose, in [13] we introduced a norm-like concept, called velocity, on the set of octagonal neighborhood sequences, in any finite dimension, and investigated its properties. This concept is introduced in a way to fit the relation "faster". In Section 3 we summarize our results in this direction.

Another important problem related to the theory of neighborhood sequences is to find a (digital) distance function on \mathbb{Z}^2 , which is appropriate for a certain application. Thus the investigation of digital distance functions and metrics becomes more and more important. See e.g. the survey paper [18] of Melter for an account, and the papers [4, 6–9, 22, 25] and the references given there for earlier results and the present state. One of the most essential tasks is to give a convenient digital metric, which approximates the Euclidean metric L_2 on \mathbb{Z}^2 well. Das [6] and Mukherjee et al. [20] determined distance functions that provide good approximations of the Euclidean distance in a certain sense. However, in these papers only periodic neighborhood sequences were used. In [14] we performed an approximation of L_2 by distance functions $d(A)$ based on general octagonal neighborhood sequences A . In contrast with the results obtained using periodic sequences (see e.g. [6, 9, 20] and the references given there), in this way we can give the actually best approximating sequence, instead of a finite part of it. Interestingly, the best approximating sequences we obtained in [14] are

(mostly) Beatty sequences, thus they can be constructed very easily. We outline our results in this direction in Section 4.

In this paper we deal with neighborhood sequences defined on \mathbb{Z}^n . However, there can be applications, where the grid points form another (e.g. triangular or hexagonal) structure. For a survey on planar grids, see [23]. The concept of neighborhood sequences can be easily generalized to these grids, see [21] for the cases of triangular and hexagonal grids. The investigations and concepts of the present paper can be extended to these structures, too.

2 Notation and basic concepts

In this section we introduce the notation and concepts which are needed to formulate our results. Fix a positive integer n for the whole paper.

Definition 1 *Let k be an integer with $0 \leq k \leq n$. The points $p = (p_1, \dots, p_n)$, $q = (q_1, \dots, q_n)$ in \mathbb{Z}^n are called k -neighbors, if $|p_i - q_i| \leq 1$ ($1 \leq i \leq n$), and $\sum_{i=1}^n |p_i - q_i| \leq k$.*

The sequence $A = (a(i))_{i=1}^\infty$, where $1 \leq a(i) \leq n$ for all $i \in \mathbb{N}$, is called an n -dimensional (shortly nD) neighborhood sequence. A is periodic, if for some $l \in \mathbb{N}$, $a(i + l) = a(i)$ ($i \in \mathbb{N}$). The set of the nD -neighborhood sequences will be denoted by S_n .

Let $p, q \in \mathbb{Z}^n$, and $A \in S_n$. The point sequence $p = p_0, p_1, \dots, p_m = q$, where p_{i-1} and p_i are $a(i)$ -neighbors in \mathbb{Z}^n ($1 \leq i \leq m$), is called an A -path from p to q of length m . The A -distance $d(p, q; A)$ of p and q is defined as the length of a shortest A -path between them.

A natural partial ordering relation on S_n can be introduced in the following way (see [8] and [11]). For $A, B \in S_n$ we define the relation \supseteq^* by

$$A \supseteq^* B \iff d(p, q; A) \leq d(p, q; B) \text{ for all } p, q \in \mathbb{Z}^n.$$

In case of $A \supseteq^* B$ we say that A is faster than B .

3 Assigning velocity to neighborhood sequences

In this section we summarize some of our results from [13], without proofs. As we mentioned, "faster" is only a partial ordering in general. However, it can be useful to compare any two neighborhood sequences. This is the reason why we introduced a norm-like concept, called velocity, on the set of octagonal neighborhood sequences. By defining velocity, we assign a positive real number to every neighborhood sequence. First we give some natural conditions, which should be met by this concept.

- (I) *Velocity must reflect the behavior of the sequences in different dimensions.*
It can happen that a sequence spreads "faster" than another one in higher dimensions, but they have the same "speed" in lower dimensional subspaces. This should affect the definition of velocity.

(II) *Sequence elements must be weighted with a suitable weight function.*

It is natural to consider the initial elements of the sequences to be more important than the elements which occur later.

(III) *Velocity must be defined such that it fits the natural ordering.*

This condition is clear: our main purpose is to define a velocity which preserves the ordering \sqsupseteq^* .

According to (II), first we give the concept of a weight system.

Definition 2 *The set of functions $\delta_j : \mathbb{N} \rightarrow \mathbb{R}$ ($1 \leq j \leq n$) is called a weight system, if δ_j is a monotone decreasing, non-negative function with $\sum_{i=1}^{\infty} \delta_j(i) < \infty$ ($1 \leq j \leq n$).*

In order to meet (I), we introduce the concept of velocity in two steps. First, we assign an n -tuple to every neighborhood sequence. The elements of this n -tuple reflect the "velocity" of the given neighborhood sequence in the subspaces of \mathbb{Z}^n of dimensions from 1 to n . Then, we define one descriptive velocity value.

Definition 3 *Let $A \in S_n$, $A = (a_i)_{i=1}^{\infty}$, and δ_j ($1 \leq j \leq n$) be a weight system. The j -dimensional velocity of A is defined as $v_j^A = \sum_{i=1}^{\infty} \max(a(i), j) \delta_j(i)$. Then the velocity of A is given by $v^A = \frac{1}{n} \sum_{j=1}^n v_j^A$.*

We show how conditions (I), (II) and (III) are met by this velocity concept.

Because of using the expressions $\max(a(i), j)$ in the definition, the velocity vector $(v_1^A, v_2^A, \dots, v_n^A)$, thus also v^A is sensitive for the behavior of the sequence A in subspaces of \mathbb{Z}^n of dimensions from 1 to n . Thus, (I) is satisfied. As we use a weight system to define $(v_1^A, v_2^A, \dots, v_n^A)$ and v^A , the requirements of (II) are also met. The following theorem verifies that our velocity concept satisfies condition (III), too.

Theorem 1 *Let $A, B \in S_n$ with $A \sqsupseteq^* B$, and let δ_j ($j = 1, \dots, n$) be a weight system. Then, $v_j^A \geq v_j^B$ for every $j = 1, \dots, n$.*

Remark 1 *By the definition of the velocity, the above theorem implies that if $A \sqsupseteq^* B$, then $v^A \geq v^B$. Note that the monotonicity of δ_j is necessary to have Theorem 1.*

As one can easily see, it can happen that with some weight system δ_j , $v_j^A \geq v_j^B$ for every $j = 1, \dots, n$, but $A \sqsupseteq^* B$ does not hold. However, in some sense we can reverse Theorem 1, even in case of positive weight systems, i.e. when $\delta_j(i) > 0$ for all $i \in \mathbb{N}$, $j = 1, \dots, n$. More precisely, we have

Theorem 2 *Let $A, B \in S_n$. If for any positive weight system δ_j ($1 \leq j \leq n$), $v_j^A \geq v_j^B$ holds for all $j = 1, \dots, n$, then $A \sqsupseteq^* B$.*

Remark 2 *It can be easily verified that the condition $v_j^A \geq v_j^B$ for all $j = 1, \dots, n$ cannot be replaced by $v^A \geq v^B$.*

4 Approximating the Euclidean distance

In this section we summarize without proofs some of our results from [14], about the approximation of the Euclidean distance L_2 by distance functions, corresponding to general octagonal neighborhood sequences. To measure the error of approximation, we compare the disks of radii $k \in \mathbb{N}$ of L_2 and of the distance functions. A similar but slightly different error function was used by Das and Chatterji [9], and Mukherjee et al. [20] for periodic neighborhood sequences. We start by introducing some notation. Let $A \in S_2$, and for every $k \in \mathbb{N}$ let

$$A_k = \{q \in \mathbb{Z}^2 : d(\mathbf{0}, q; A) \leq k\}$$

denote the region occupied by A after k steps. Write $H(A_k)$ for the convex hull of A_k in \mathbb{R}^2 . Observe that $H(A_k)$ in general is an octagon which is symmetric to the coordinate axes and to the lines $y = x$ and $y = -x$ in the $[x, y]$ plane. Let

$$O_k = \{q \in \mathbb{Z}^2 : L_2(\mathbf{0}, q) \leq k\} \quad \text{and} \quad G_k = \{q \in \mathbb{R}^2 : L_2(\mathbf{0}, q) \leq k\}$$

be the disks of radius k in \mathbb{Z}^2 and \mathbb{R}^2 , respectively. The sets A_k and O_k will be called the k -disks of the distances $d(A)$ and L_2 , respectively. We will use the number of 2 values occurring among the first k elements of a neighborhood sequence $A = (a(i))_{i=1}^\infty$. So for every $k \in \mathbb{N}$ put

$$\mathbf{2}_A(k) = |\{a(i) : a(i) = 2, 1 \leq i \leq k\}|.$$

For convenience, write $\mathbf{2}_A(0) = 0$. For any $x \in \mathbb{R}$, let $\lfloor x \rfloor$ denote the largest integer which is less than or equal to x , and $\lceil x \rceil$ the smallest integer which is greater than or equal to x . Let $\alpha \in \mathbb{R}$ with $0 \leq \alpha \leq 1$, and let $A = (a(i))_{i=1}^\infty$, $B = (b(i))_{i=1}^\infty$ be sequences of 1-s and 2-s, defined by

$$a(i) = \lfloor i\alpha \rfloor - \lfloor (i-1)\alpha \rfloor + 1, \quad b(i) = \lceil i\alpha \rceil - \lceil (i-1)\alpha \rceil + 1 \quad (i \in \mathbb{N}).$$

The sequences A, B are called Beatty sequences on the letters 1, 2. We refer to [17] for the basic properties of such sequences and their generalizations.

To decide how a digital distance $d(A)$ approximates the Euclidean distance L_2 on \mathbb{Z}^2 , we compare the k -disks A_k and O_k . More precisely, we investigate the function

$$TE_A(k) = \text{Area}(H(A_k) \triangle G_k),$$

called the total error of the approximation at the k -th step. We also use the relative error at the k -th step, and the limit relative error (if it exists) defined as

$$RE_A(k) = \frac{TE_A(k)}{k^2\pi} \quad \text{and} \quad RE_A = \lim_{k \rightarrow \infty} RE_A(k),$$

respectively. We note that the authors in [9] and [20] simply took the difference $|\text{Area}(H(A_k)) - \text{Area}(G_k)|$ as an error function. Clearly, our approach is more sensitive for the matching of $H(A_k)$ and G_k . We consider the following three approximation problems:

Problem 1 Find a sequence $A^{(1)} \in S_2$ (if exists) such that for every $B \in S_2$ and $k \in \mathbb{N}$

$$\text{Area}(H(A_k^{(1)}) \triangle G_k) \leq \text{Area}(H(B_k) \triangle G_k).$$

We consider separately the case when the octagons $H(A_k)$ cover G_k for every $k \in \mathbb{N}$, that is the corresponding function $d(A)$ minorates L_2 .

Problem 2 Find a sequence $A^{(2)} \in S_2$ (if exists) such that $H(A_k^{(2)}) \supseteq G_k$ for every $k \in \mathbb{N}$, and for every $B \in S_2$, $H(B_k) \supseteq G_k$ implies that

$$\text{Area}(H(A_k^{(2)}) \setminus G_k) \leq \text{Area}(H(B_k) \setminus G_k).$$

Observe that it does not make sense to consider a problem with $H(A_k) \subseteq G_k$. We also investigate the following discrete version of Problem 2. Note that Problem 1 does not have a similar variant.

Problem 3 Find a sequence $A^{(3)} \in S_2$ (if exists) such that $A_k^{(3)} \supseteq O_k$ for every $k \in \mathbb{N}$, and if $B \in S_2$ with $B_k \supseteq O_k$, then $B_k \supseteq A_k^{(3)}$.

The following results give answers to the above problems.

Theorem 3 Let the neighborhood sequence $A^{(1)} = (a^{(1)}(i))_{i=1}^\infty$ be defined by

$$a^{(1)}(i) = \begin{cases} 1, & \text{if } E\left(\frac{2_{A^{(1)}}(i-1)}{i}\right) < E\left(\frac{2_{A^{(1)}}(i-1)+1}{i}\right), \\ 2, & \text{otherwise,} \end{cases}$$

where

$$E(y) = \begin{cases} 2y - y^2, & \text{if } y \geq \sqrt{2} - 1, \\ \frac{2 \arccos(y(y+2)) - 2(y+1)\sqrt{1-2y-y^2} - y^2 + 2y}{2}, & \text{otherwise.} \end{cases}$$

Then for any $B \in S_2$ and $k \in \mathbb{N}$,

$$\text{Area}(H(A_k^{(1)}) \triangle G_k) \leq \text{Area}(H(B_k) \triangle G_k).$$

Remark 3 We have $A^{(1)} = (2, 1, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, \dots)$. By a simple calculation, for the relative error of $A^{(1)}$ we obtain

$$RE_{A^{(1)}} = \frac{2 - \pi}{\pi} + \frac{2}{\pi} \left(2 \arccos\left(\frac{3 + 8\sqrt{6}}{25}\right) + \frac{4\sqrt{6} - 11}{5} \right) = 0.046525347 \dots$$

Unfortunately, $d(A^{(1)})$ is not a metric on \mathbb{Z}^2 . Another unpleasant feature of $A^{(1)}$ is that it is not easy to generate: to obtain its k -th element, we have to calculate the first $k - 1$ elements previously. Now we give two sequences which are easy to construct, and for every $k \in \mathbb{N}$, one of them is also the best to approximate G_k .

Corollary 1 For $j = 1, 2$ and $i \in \mathbb{N}$ put

$$c^{(j)}(i) = \begin{cases} j, & \text{if } i = 1, \\ \lfloor i \frac{2\sqrt{6}-3}{5} \rfloor - \lfloor (i-1) \frac{2\sqrt{6}-3}{5} \rfloor + 1, & \text{if } i > 1, \end{cases}$$

and write $C^{(1)} = (c^{(1)}(i))_{i=1}^\infty$ and $C^{(2)} = (c^{(2)}(i))_{i=1}^\infty$. Then for every $B \in S_2$ and $k \in \mathbb{N}$,

$$\min \{TE_{C^{(1)}}(k), TE_{C^{(2)}}(k)\} \leq TE_B(k).$$

Remark 4 As $C^{(1)}$ is a Beatty sequence, it is easy to prove that $d(C^{(1)})$ is a metric on \mathbb{Z}^2 . Thus in a sense $d(C^{(1)})$ can be considered to be the best metric (coming from a neighborhood sequence) to approximate the Euclidean distance on \mathbb{Z}^2 . Note that $RE_{C^{(1)}} = RE_{A^{(1)}}$.

Theorem 4 Let $A^{(2)} = (a^{(2)}(i))_{i=1}^\infty$ be defined by

$$a^{(2)}(i) = \lceil i(\sqrt{2} - 1) \rceil - \lceil (i-1)(\sqrt{2} - 1) \rceil + 1 \quad (i \in \mathbb{N}).$$

Then $H(A_k^{(2)}) \supseteq G_k$ for any $k \in \mathbb{N}$, and $B \in S_2$, $H(B_k) \supseteq G_k$ implies that

$$\text{Area}(H(A_k^{(2)}) \setminus G_k) \leq \text{Area}(H(B_k) \setminus G_k).$$

Remark 5 The octagons $H(A_k^{(2)})$ are almost regular. (For regularity we should have $2_A(k) = k(\sqrt{2} - 1)$, which is impossible.) Obviously, the ratio of the inclined and horizontal (or vertical) sides of $H(A_k^{(2)})$ tends to 1 as $k \rightarrow \infty$. It was already noted by Rosenfeld and Pfaltz [24] that the "best approximating" sequence should have such property. A detailed study in this direction with periodic sequences under slightly different circumstances was performed by Das and Chatterji [9].

Remark 6 By a simple calculation for the relative error of $A^{(2)}$ we obtain

$$RE_{A^{(2)}} = \frac{8(\sqrt{2} - 1) - \pi}{\pi} = 0.054786175\dots$$

By the following theorem we solve Problem 3.

Theorem 5 Let $A^{(3)} = (a^{(3)}(i))_{i=1}^\infty$ be defined by

$$a^{(3)}(i) = \lfloor i(\sqrt{2} - 1) \rfloor - \lfloor (i-1)(\sqrt{2} - 1) \rfloor + 1 \quad (i \in \mathbb{N}).$$

Then for every $k \in \mathbb{N}$, $O_k \subseteq A_k^{(3)}$. Moreover, if $B \in S_2$ such that $O_k \subseteq B_k$ for some $k \in \mathbb{N}$, then $A_k^{(3)} \subseteq B_k$.

Remark 7 As $A^{(3)}$ is a Beatty sequence, one can easily see that $d(A^{(3)})$ is a metric on \mathbb{Z}^2 . Moreover, for all $x, y \in \mathbb{Z}^2$ we have $d(x, y; A^{(3)}) \leq L_2(x, y)$, and if $d(A)$ is any metric corresponding to some $A \in S_2$ and

$$d(x, y; A) \leq L_2(x, y) \quad \text{for all } x, y \in \mathbb{Z}^2,$$

then

$$d(x, y; A) \leq d(x, y; A^{(3)}) \text{ for all } x, y \in \mathbb{Z}^2.$$

That is, $d(A^{(3)})$ is "uniformly" the best digital metric corresponding to an octagonal neighborhood sequence to approximate L_2 from below.

Finally, we compare our approximation results with those of Das in [6], and Mukherjee et al. in [20]. As in each of the papers [6, 20, 14] different error functions were used, we introduce a new one for this purpose. We define the k -th discrete total error

$$DTE_A(k) = |A_k \triangle O_k|,$$

being the number of grid points in the symmetric difference of A_k and O_k , where $A \in S_2$, $k \in \mathbb{N}$. The k -th discrete relative error of the approximation is defined as

$$DRE_A(k) = \frac{DTE_A(k)}{|O_k|}.$$

The periodic neighborhood sequence $S = (1, 1, 2, 1, 2)$ (which generates a metric) proposed by Das [6] to approximate L_2 is a refinement of the sequence $(1, 1, 2)$ suggested by the authors in [20]. Thus we compare the behavior of S and $C^{(1)}$ (which also generates a metric) defined in Corollary 1. The following table shows the discrete relative errors of these sequences with respect to L_2 . From the table we can see that $C^{(1)}$ behaves better in this "digital" sense.

Table 1. Discrete relative errors of distance functions generated by S and $C^{(1)}$.

k	$DRE(k)$	
	$S = (1, 1, 2, 1, 2)$	$C^{(1)} = \left(\lfloor i \frac{2\sqrt{6}-3}{5} \rfloor - \lfloor (i-1) \frac{2\sqrt{6}-3}{5} \rfloor + 1 \right)_{i=1}^{\infty}$
10	0.12618297...	0.11356467...
50	0.06322498...	0.05863607...
100	0.05551135...	0.05169176...
200	0.05218540...	0.04909694...
500	0.05023499...	0.04755593...
1000	0.04955390...	0.04704176...

References

1. G. Borgefors: Distance transformations in arbitrary dimensions. *Comput. Vision Graphics Image Process* **27** (1984) 321–345
2. Borgefors, G.: Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10** (1988) 849–865

3. Cuisenaire, O.: Distance transformations: fast algorithms and applications to medical image processing. Ph.D. Thesis, UCL, Louvain-la-Neuve, Belgium, October 1999
4. Danielsson, P.E.: 3D octagonal metrics. Eighth Scandinavian Conf. Image Process. 1993, pp. 727–736
5. Das, P.P.: Lattice of octagonal distances in digital geometry. *Pattern Recognition Lett.* **11** (1990), 663–667
6. Das, P.P.: Best simple octagonal distances in digital geometry, *J. Approx. Theory* **68** (1992) 155–174
7. Das, P.P., Chakrabarti P.P., Chatterji, B.N.: Generalised distances in digital geometry. *Inform. Sci.* **42** (1987) 51–67
8. Das, P.P., Chakrabarti, P.P., Chatterji, B.N.: Distance functions in digital geometry. *Inform. Sci.* **42** (1987) 113–136
9. Das P.P., Chatterji, B.N.: Octagonal distances for digital pictures. *Inform. Sci.* **50** (1990) 123–150
10. Fazekas, A.: Lattice of distances based on 3D-neighbourhood sequences. *Acta Math. Acad. Paed. Nyfregyháziensis* **15** (1999) 55–60
11. Fazekas, A., Hajdu, A., Hajdu, L.: Lattice of generalized neighbourhood sequences in nD and ∞D . *Publ. Math. Debrecen* **60** (2002) 405–427
12. Y. Ge and J.M. Fitzpatrick: On the generation of skeletons from discrete euclidean distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18** (1996) 1055–1066
13. Hajdu, A., Hajdu, L.: Velocity and distance of neighbourhood sequences. *Acta Cybernet.* **16** (2003) 133–145
14. Hajdu, A., Hajdu, L.: Approximating the Euclidean distance by digital metrics. *Discrete Math.*, to appear
15. Hajdu, A., Nagy, B., Zörgő, Z.: Indexing and segmenting colour images using neighbourhood sequences. *IEEE ICIP 2003*, Barcelona, Spain, I/957–960
16. Hajdu, A., Tóth, T., Veréb, K., Zörgő, Z.: Distance functions in multidimensional image processing applications. *KÉPAF 4* (2003), Miskolc-Tapolca
17. Lothaire, M.: *Combinatorics on words* (Addison-Wesley Publishing Co., Reading, Mass., 1983) xix+238 pp.
18. Melter, R.A.: A survey on digital metrics. *Contemporary Mathematics* **119** (1991) 95–106
19. Mukherjee, J., Aswatha Kumar, M., Das, P.P., Chatterji, B.N.: Use of medial axis transforms for computing normals at boundary points. *Pattern Recognition Lett.* **23** (2002) 1649–1656
20. Mukherjee, J., Das, P.P., Aswatha Kumar, M., Chatterji, B.N.: On approximating Euclidean metrics by digital distances in 2D and 3D. *Pattern Recognition Lett.* **21** (2000) 573–582
21. Nagy, B.: Finding shortest path with neighbourhood sequences on triangular grids. *ITI – ISPA 2001*, Pula, Croatia (2001) 55–60
22. Nagy, B.: Distance functions based on neighbourhood sequences. *Publ. Math. Debrecen* **63** (2003) 483–493
23. Radványi, A.: On the rectangular grid representation of general CNN networks. *Int. J. Circ. Theor. Appl.* **30** (2002) 181–193
24. Rosenfeld, A., Pfaltz, J.L.: Distance functions on digital pictures. *Pattern Recognition* **1** (1968) 33–61
25. Yamashita, M. and Ibaraki, T.: Distances defined by neighbourhood sequences. *Pattern Recognition* **19** (1986) 237–246

Distance functions in multidimensional image processing applications

András Hajdu¹, Tamás Tóth², Krisztián Veréb³, and Zoltán Zörgő⁴

¹ Institute of Informatics, University of Debrecen, H-4010 Debrecen PO Box 12.
hajdua@inf.unideb.hu

² Institute of Informatics, University of Debrecen, H-4010 Debrecen PO Box 12.
tothtam@delfin.klte.hu

³ Institute of Informatics, University of Debrecen, H-4010 Debrecen PO Box 12.
sparrow@inf.unideb.hu

⁴ Institute of Informatics, University of Debrecen, H-4010 Debrecen PO Box 12.
zorgoz@inf.unideb.hu

Abstract. In this paper we show how we can take advantage of using different distance functions in image processing applications. The proposed methods are based on well-known algorithms that use distance measurement. We focus on multidimensional image indexing and segmentation procedures, and also show examples for the extraction of such feature vectors that can be used in image retrieval. As a special family, we perform a detailed analysis using distance functions generated by neighbourhood sequences. The application of such distance functions is quite natural and descriptive for images, since e.g. the colour coordinates of the pixels are non-negative integers. An additional interesting property of neighbourhood sequences is that they do not generate metrics in general, so we can obtain many distance functions in this way. Our final purpose is to find distance functions that provide the "best" results for a given problem, so we present some tools that help with finding them.

1 Introduction

Indexing and segmenting colour images is a very important field in digital image processing with growing interest. Many of these applications are based on some distance function to calculate the difference between the colours of the pixels. Usually some classic metrics are used, though a special distance function might be more suitable according to the nature of the problem. For example, several image representations use integer colour coordinate values, thus digital distance functions also might be considered.

This observation has led to the investigation of the applicability of a family of digital distance functions in [11]. In that paper the authors showed how colour image indexing and segmentation can be achieved by applying distance functions generated by neighbourhood sequences in simple colour distance measurement, region growing, and clustering. Now, we continue the work started in [11] by introducing some analytical tools that help with choosing the optimal distance

functions for the given task. However, for the sake of the reader, first we briefly summarize the main points presented in [11].

2 Neighbourhood sequences

To get familiar with neighbourhood sequences the reader should overview the results presented in [5, 8, 10, 15]. Now we recall those concepts of this theory which are essential in our investigations. Though we will consider the RGB image representation in details, we give the notions for arbitrary dimension, since our procedures can be applied to arbitrary dimensional integer image representations. From now on, let n be an arbitrary positive integer. Let q and r be two points in \mathbb{Z}^n . The i -th coordinate of the point q is indicated by $\text{Pr}_i(q)$. Let m be an integer with $1 \leq m \leq n$. The points q and r are m -neighbours, if the following two conditions hold:

- $|\text{Pr}_i(q) - \text{Pr}_i(r)| \leq 1 \quad (1 \leq i \leq n)$,
- $\sum_{i=1}^n |\text{Pr}_i(p) - \text{Pr}_i(q)| \leq m$.

The sequence $A = (A(i))_{i=1}^{\infty}$, where $A(i) \in \{1, \dots, n\}$ for all $i \in \mathbb{N}$, is called an n -dimensional (shortly nD) neighbourhood sequence. If for some $l \in \mathbb{N}$, $A(i+l) = A(i)$ ($i \in \mathbb{N}$), then A is periodic with period l . In this case we briefly write $A = \{A(1)A(2)\dots A(l)\}$. For example, we write $\{12\}$ for the neighbourhood sequence $1, 2, 1, 2, 1, 2, \dots$. The point sequence $q = q_0, q_1, \dots, q_m = r$, where q_{i-1} and q_i are $A(i)$ -neighbours in \mathbb{Z}^n ($1 \leq i \leq m$), is called an A -path from q to r of length m . The A -distance $d(q, r; A)$ of q and r is defined as the length of the shortest A -path(s) between them. The distance functions generated by neighbourhood sequences are not metrics in general and the existence of this property can be checked by a simple criterion [14]. Nevertheless, in some cases non-metric distance functions also may provide nice results, thus it is not recommended to exclude them from analysis. Moreover, this way we have a lot more distance functions to choose from to refine our results.

3 Applications based on measuring distance in the RGB cube by neighbourhood sequences

Colour image indexing and segmentation procedures are based on the comparison of the colour of the pixels. We use the 24-bit RGB cube (that is the domain is between black= $(0, 0, 0)$ and white= $(255, 255, 255)$) to illustrate the descriptive behaviour of neighbourhood sequences in measuring the distance of two colours. As it is shown in [11], reasonable differences may occur according to the chosen neighbourhood sequence, thus it must be selected carefully to achieve the desired result. Now we present some applications. We start with the generalisation of the "fuzziness" method referring to Adobe Photoshop terminology, then consider colour image segmentation with region growing, finally present an indexing method using cluster analysis.

Fuzziness. This indexing procedure [9] selects those pixels which are within a given distance to one or more initially fixed seed colours. The implementation of this method for a fixed distance function also can be found in Adobe Photoshop, where it is referred to as the "Fuzziness" option [1]. Figure 1 shows that the results of the fuzziness procedure highly depend on the chosen seeds, threshold and neighbourhood sequence.

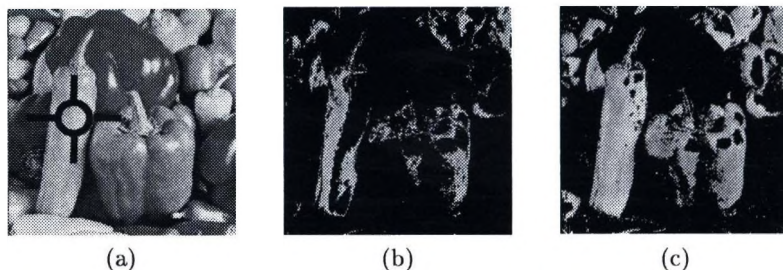


Fig. 1. Fuzziness from the indicated seed colour with threshold=45, using neighbourhood sequences; (a) original image, (b) {1}, (c) {2}.

Region growing. To obtain a connected region, we can insert the distance functions used at fuzziness into a region growing algorithm (see [9, 16]). In Figure 2 we show some results.

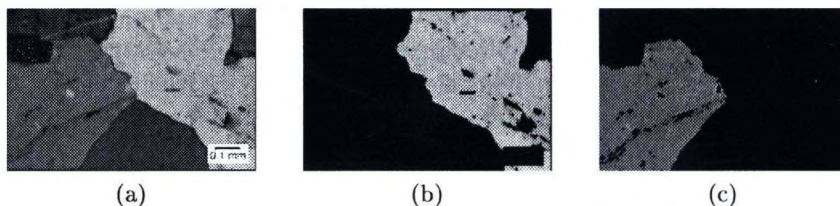


Fig. 2. Region growing of stone parts; (a) original image, (b) using one seed colour, (c) using two seed colours.

Clustering. We recall an algorithm for indexing colour images based on cluster analysis [9]. In this procedure, the elements of the RGB cube are classified into clusters using a suitable metric. In our experiments, we used digital distance functions generated by neighbourhood sequences. In Figure 3 we show results for our clustering method.

A quantitative analysis of the proposed clustering method can be obtained by considering a suitable measure, like the specialisation of the uniformity measure of Levine and Nazif [12].

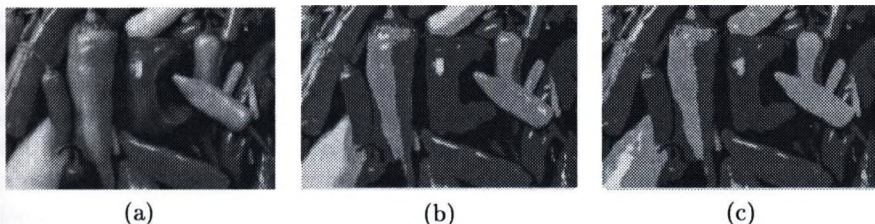


Fig. 3. Classifying colours into six clusters by neighbourhood sequences; (a) original image, (b) {12}, (c) {23}.

4 Tools for finding optimal distance measurement

Now we turn to present some tools and give guidelines to help with finding optimal distance functions and threshold values for the above procedures. Our approach is based on histogram analysis, and we propose two types of histograms which may be helpful.

Fuzziness histogram. This histogram type can be assigned to the "Fuzziness" procedure, and might be useful especially in region growing. We choose one or more seed colour(s) and for every non-negative integer K we calculate the number of image pixels that have the colour at the distance K from the seed(s). From this data we compose a histogram. Naturally, the shape of the histogram highly depends on the chosen distance function. For example, a "faster" distance function results a shorter histogram, but reasonable differences also may occur with respect to modality. These properties can be nicely observed in Figure 4.

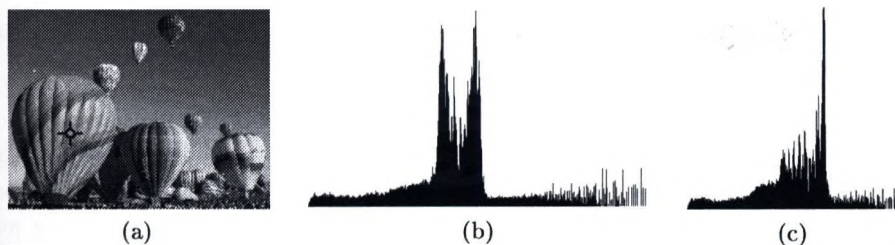


Fig. 4. Fuzziness histograms for the indicated seed colour, using different metrics; (a) original image, (b) L_1 , (c) L_2 .

The difference between two such histograms can be measured by suitable histogram distance measures [3] which reflect the different behaviour of different distance functions. The first mode of the histogram may play very important role in region growing, as we often do not take much care about colours far from the seed(s). The separation of this mode can be performed e.g. by density estimation

[17] to find the first local minimum. Distance functions which provide nicely separable first mode are expected to give good results in this case. However, note that a mode may include more than one colour classes (and thus more objects). By repeating the procedure only to the subimage defined by the investigated mode these colour classes can be further separated.

Global histogram. We propose another type of histogram which depends on only the distance function and the image. This case the K -th column of the histogram corresponds to the number of those pixel pairs in the image, whose colours have the distance K . Since this way no more additional parameters (e.g. seeds) are expected to be given, we refer to this descriptor as a global histogram. Global and fuzziness histograms can be processed similarly, but this time the first mode has no particular importance. These histograms nicely show if distance functions behave differently on a given image, and similarly to fuzziness histograms, histogram distances also can be calculated. Moreover, we gain a nice illustrative representation if the distance function is based on a neighbourhood sequence. For example, in Figure 5 the obtained global histograms nicely reflect the values in the period of the neighbourhood sequences. Finding local minima by den-

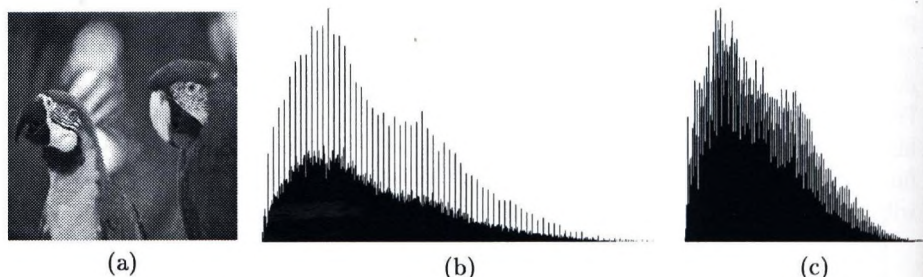


Fig. 5. Global histograms based on neighbourhood sequences; (a) original image, (b) {1111113}, (c) {123}.

sity estimation, one can use global histograms to find colour classes (however, this time we should mind that the different modes of the histogram may contain same image pixels). Global and fuzziness histograms also seem to be well applicable in image retrieval as feature vectors extracted from the images. As these histograms correspond mainly to the colour distribution of the images, this approach is expected to be useful in the pre-filtering phase of image retrieval.

5 Conclusions

The most image indexing and segmentation techniques are based on classical (e.g. Euclidean) metrics. In some cases other metrics may provide better results [7], but very few suggestions can be found in the literature how to choose

them. The authors in [11] showed that distance functions based on neighbourhood sequences also might be worth taking into consideration in some digital image processing problems, since they are valid alternatives for integer domains. Especially, classic metrics also can be approximated by distance functions generated by neighbourhood sequences; see e.g. [4, 6, 10, 13]. In the present paper we propose some tools that may be helpful with choosing the appropriate distance functions, and nicely reflect the behaviour of neighbourhood sequences, as well. Our future aim is to make the choice of a suitable distance function as automatic as possible based on the tools presented here.

References

- [1] Adobe Photoshop User's Guide
- [2] M.R. Anderberg, *Cluster Analysis for Application*, Academic Press, New York, 1973.
- [3] S.H. Cha and S.N. Srihari, On measuring the distance between histograms, *Pattern Recognition* 35 (2002), 1355–1370.
- [4] P.E. Danielsson, 3D octagonal metrics, *Eighth Scandinavian Conf. Image Process.* (1993), 727–736.
- [5] P.P. Das, P.P. Chakrabarti and B.N. Chatterji, Distance functions in digital geometry, *Inform. Sci.* 42 (1987), 113–136.
- [6] P.P. Das and B.N. Chatterji, Octagonal distances for digital pictures, *Inform. Sci.* 50 (1990), 123–150.
- [7] B. Everitt, *Cluster Analysis*, Heinemann Educational Books Ltd, London, 1973.
- [8] A. Fazekas, A. Hajdu and L. Hajdu, Lattice of generalized neighbourhood sequences in nD and ∞D , *Publ. Math. Debrecen* 60 (2002), 405–427.
- [9] R.C. Gonzalez and R.E. Woods, *Digital image processing*, Addison-Wesley, Reading, MA, 1992.
- [10] A. Hajdu and L. Hajdu, Analytical and approximation properties of neighborhood sequences, *KÉPAF* 4 (2003), Miskolc-Tapolca.
- [11] A. Hajdu, B. Nagy and Z. Zörgő, Indexing and segmenting colour images using neighbourhood sequences, *IEEE ICIP 2003*, Barcelona, Spain, I/957–960.
- [12] M.D. Levine and A.M. Nazif, Dynamic measurement of computer generated image segmentations, *IEEE trans. PAMI* 7 (1985), 155–164.
- [13] J. Mukherjee, P.P. Das, M. Aswatha Kumar and B.N. Chatterji, On approximating Euclidean metrics by digital distances in 2D and 3D, *Pattern Recognition Lett.* 21 (2000), 573–582.
- [14] B. Nagy, Distance functions based on neighbourhood sequences, *Publ. Math. Debrecen* 63 (2003), 483–493.
- [15] A. Rosenfeld and J.L. Pfaltz, Distance functions on digital pictures, *Pattern Recognition* 1 (1968), 33–61.
- [16] M. Sonka, V. Hlavac and R. Boyle, *Image processing, analysis, and machine vision*. Brooks/Cole Publishing Company, Pacific Grove, CA, 1999.
- [17] www.stat.sc.edu/rsrch/gasp/density/

Gépi látáson alapuló rendszerek humán felhasználói környezetben – A Török 2

Hajdu András¹, Hodnics István Márton², Kovács Kornél³, Ungvári Antal⁴

¹ Debreceni Egyetem, Informatikai Intézet, 4010 Debrecen, Pf. 12.
hajdua@inf.unideb.hu

² Debreceni Egyetem, Informatikai Intézet, 4010 Debrecen, Pf. 12.
hodnics@delfin.klte.hu

³ Debreceni Egyetem, Informatikai Intézet, 4010 Debrecen, Pf. 12.
thekoko@hotmail.com

⁴ Debreceni Egyetem, Informatikai Intézet, 4010 Debrecen, Pf. 12.
uantal@tigris.klte.hu

Kivonat A kutatás/fejlesztés fő célkitűzése olyan gépi látáson alapuló technikák kidolgozása, amelyek segítségével egy számítógépes rendszer igen magas szinten úgy tájékozik egy emberi felhasználásra szánt környezetben, hogy ez az emberi felhasználót nem zavarja. A megfigyelésen kívül a rendszer beavatkozást is végezhet, amelyhez megfelelő robottechnikára van szükség, melynek jellege nyilvánvalóan a célfeladathoz idomul. A rendszer kialakítása során pszichológiai szempontból nagyon fontos szerepet kívánunk szánni a humán felhasználói tényezőnek, azaz a rendszert úgy kell kialakítani, hogy a gépi látás megbízhatóságát biztosító tényezők (például könnyen szeparálható színek használata, éles kontúrok, stb.), illetve a robottechnikai megoldások az emberi résztvevő számára a lehető legkevésbé legyenek zavaróak. Egy ilyen rendszer demonstrálásaként konkrét példaként bemutatjuk egy olyan robotsakkozó tervét, amely érzékeli az emberi játékos által megtett lépéseket, és egy robotkar segítségével megteszi a megfelelő válaszlépéseket.

1. Bevezetés – gépi látás használata humán környezetben

Jelen leírásunkban az absztraktban felvázolt rendszer gépi látáson alapuló részére összpontosítunk, ám rögtön megjegyezzük, hogy mindenképpen figyelmet érdemel az egyéb érzékeléseken (hang, érint(kez)és, stb.) alapuló technikákkal való kombináció is. A fő cél az embert úgy visszahelyezni megszokott környezetébe, hogy közben a háttérben – láthatatlanul – rendelkezésre álljon napjaink számítógépes technológiája. A koncepció egy nagyon fontos eleme, hogy nem elég pusztán a gépi rendszer hatékonyságára tekintettel lenni, hanem ügyelni kell az emberi felhasználó rendszerrel szembeni toleranciájára is. Így nem válogathatunk össze például csak aszerint színeket egy irodai alkalmazásnál, hogy a gépi rendszer minél könnyebben különbséget tudjon tenni az objektumok között, mert a felhasználó esetleg pszichikailag nem képes elviselni.

Egy ilyen rendszer demonstrálásaként konkrét példaként bemutatunk egy robotsakkozót. Az emberek sokáig csak egymással sakkoztak, ami kialakította

a sakkjáték megszokott rendjét: ketten ülnek egy sakktábla mellett és felváltva tesznek lépéseket. A számítógép megjelenésével a játék más jelleget öltött. A játékosok elhagyták a megszokott, természetes közeget, és a számítógép elé ültek. A lépések megtételéhez az egerrel kell kattintani a megfelelő helyre, a játék felgyorsult és személytelenné vált. A következő lépés az lehet, ha az ember úgy ilhet vissza a a sakkasztalhoz, hogy közben nem veszíti el a számítástechnikai vívmányokat.

2. A Török 2

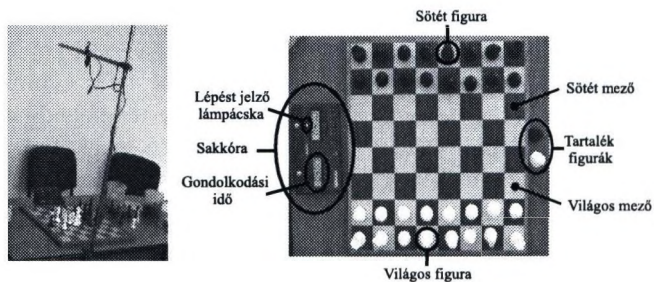
Jól ismert, hogy Kempelen Farkas sakkozógépében, a Törökben, egy törpe tevékenykedett. Célunk egy olyan rendszer létrehozása, amely valóban képes arra, hogy hagyományos módon sakkjátszmát váltson az emberrel. A gépi sakkozónk terve három fő pilléren nyugszik: gépi látás segítségével érzékelni az emberi játékos által megtett lépéseket; sakkprogram segítségével megkeresni a megfelelő válaszlépést; robotkar segítségével megtenni a lépést a sakktáblán. Az alábbiakban a rendszer azon részét tekintjük át, melynek feladata az emberi játékosok által megtett lépések detektálása. Az ebben használt eljárások a következő szempontok köré csoportosíthatók: kamera elhelyezése; játék elemeinek kalibrálása; lépések detektálása változásfigyeléssel.

2.1. Kamera elhelyezése

Egy természetese lehetőség a játékot figyelő kamera elhelyezésére, ha azt a sakktábla közepe fölött helyezük el olyan magasságban, ahol az már a játékosokat nem zavarja. Ennek előnyei: minimálisra csökkennek a takarásokból származó problémák, így nincs szükség további kamerákra; a sakktábla és az egyes mezők viszonylag könnyen megtalálhatók különböző helyszíneken is; a modern sakkórák gondolkodó felet jelző lámpácskái, illetve digitális számjegyei jól leolvashatók. A módszer hátránya, hogy a felülnézeti képen a figurák típusának (gyalog, huszár, futó, bástya, vezér, király) megkülönböztetése problematikus, ezért a figuradetektáláshoz kiegészítő technikák is szükségesek.

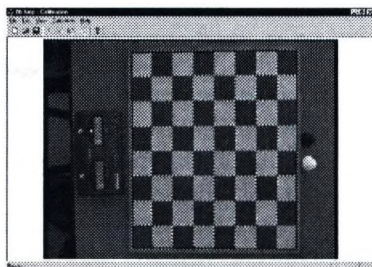
2.2. A játék elemeinek kalibrálása

A gyorsabb feldolgozás érdekében feltesszük, hogy a kamera, a sakktábla és a sakkóra helyzete nem változik meg a játék folyamán. Inicializálásakor a következő elemek kalibrálására kerül sor: sakktábla mezőinek helye (automatikus); sakkóra számjegyeinek helye (kézi); sakkóra lépést jelző lámpácskájának helye (automatikus); tartalékfigurák helye (kézi); világos/sötét mező (átlagos) színe (automatikus); világos/sötét figurák (átlagos) színe (automatikus). A rendszer megvalósítása és elemei az 1. ábrán láthatók.



1. ábra. A játékot figyelő rendszer és a játék elemei.

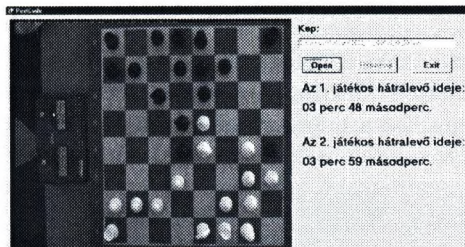
Sakktábla mezőinek helye.. A sakk szabályai szerint a figurák nem lóhatnak le az egyes mezőkről, így a lépések megtalálásának természetes alapja a mezők figyelése. Bizonyos fokú figuraátlógással a módszerek kidolgozásánál mégis célszerű kalkulálni. Rendszerünkben a könnyebb feldolgozáshoz minden mezőt téglalappal reprezentálunk. Az általunk használt algoritmus a mezőket tartománynövelessel határozza meg, a mezők színének átlagos értékéből, mint magpontokból kiindulva [2,3]. A magpontokat adó színnek meghatározásához a tábla közepének bizonyos méretű környezetén belül lévő tartomány két domináns színosztálynak (világos és sötét mezők) centroidját használhatjuk. A mezők automatikus kalibrációjának eredményét a 2. ábrán láthatjuk.



2. ábra. A mezők automatikus detektálása és téglalappal való reprezentálása.

A sakkóra információinak leolvasása.. A sakkóra mutatja a két játékos még meglévő gondolkodási idejét, illetve hogy ki következik lépésen. Rendszerünkben digitális sakkórákat használtunk, amelyeken egy riktó lámpácska jelzi azt, hogy ki következik lépésen, a gondolkodási időt pedig szabványos digitális kijelző mutatja. A gépi sakkozó számára mindkét adat fontos, mivel egyrészt elegendő a lámpácskát ellenőriznie ahhoz, hogy megtudja, ha ő következik, másrészt mindig tudja, hogy mennyi a még rendelkezésre álló idő. A lámpácskát annak környezetől élesen elütő színe miatt automatikusan is könnyű detektálni. A számjegyek automatikus megkeresése már lényegesen nehezebb feladat, ezért rendszerünkben jelenleg az adminisztrátor feladata az egyes digitális számjegyek befoglaló

téglalapjának megadása. Azt a feltevést használva, hogy a digitális kijelzés két elütő színt használ, az aktuálisan mutatott idő már automatikusan kerül leolvasásra. A számjegyek felismerését nem mintaillesztéssel, hanem a digitális számok szabványos megjelenítését kihasználva oldottuk meg. A 3. ábra a jelzett idő automatikus leolvasását mutatja.



3. ábra. A játékban használt digitális sakkóra és az idő automatikus leolvasása.

Figuratípusok megkülönböztetése, tartalék figurák helye.. A figurák típusának megkülönböztetésére helyes lépéskövetés esetén csak akkor van szükség, amikor az emberi játékos gyalogátalakulásnál új figurát vesz fel. A felvett figura típusának meghatározását jelenleg azt segíti, hogy a tábla mellé előkészített, tartalék figurákat kell használni. Olyan sakkészletek esetén, ahol az egyes figuratípusok eltérő színűek (például a szírványsakknál), ez a probléma nem lép fel. A figuratípusok megkülönböztetésére jelenleg olyan eljárásokat teszünk, amelyek infravörös/ ultrabolya megvilágítás esetén más-más színt adnak vissza a digitális képeken. Az infravörös képek használata azért is ígéretes, mivel az az emberi résztvevőt nem zavarja. Ezen technikák használatával akár az is lehetővé válik, hogy a figurák és a mezők színe megegyezzen.

Világos/sötét mezők és figurák színe.. Ha a látható fénytartományban dolgozunk, akkor a rendszerünk megbízható működésének elengedhetetlen feltétele, hogy a mezők és a figurák színe elütő legyen. A sakk esetében ez négy fő szín csoportot jelent a két-két mező/figuraszínnek megfelelően. A kalibrációs szakaszban a későbbi változáskereső módszerekhez célszerű meghatározni ennek a négy fő csoportnak az átlagos színét is.

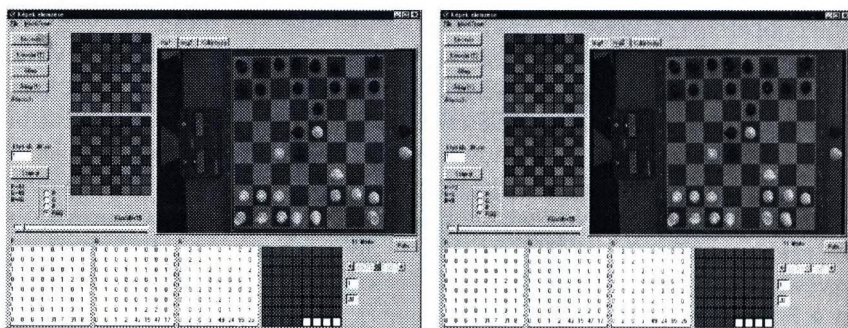
3. Lépések detektálása változáskereséssel

A megtett lépések meghatározását a lépés során megváltozott mezők detektálásával végzi a rendszer. Szabályos lépés esetén ez a sakkban 2, 3 vagy 4 mező változását jelenti. Rendszerünkben három eljárást kombináltunk, amelyből kettő a mezőváltozások mértékét figyeli, míg a harmadik kifejezetten a figurák helyének meghatározására irányul. A módszerek optimalizálás nélküli, lépésenkénti együttes lefutási ideje körülbelül 150ms (Pentium, 2GHz). Tapasztalataink azt

mutatták, hogy ha a három módszerből mindig azt választjuk ki, amely a legmegbízhatóbban detektál az adott pillanatban, a tévesztés gyakorlatilag kizárt. Az alábbiakban a változások érzékelését figyelő módszereket mutatjuk be.

3.1. Változásfigyelés színátlag számolásával

Egy egyszerűen implementálható, gyors és betörésfigyelő rendszerekben már bizonyított módszer az intenzitásváltozások vizsgálata. Rendszerünkben a mezők átlagszínének meghatározása után egyszerű küszöböléssel döntjük el az egyes lépések után, hogy történt-e változás a mezőn. Kísérleteink során a módszer igen hatékonynak bizonyult, ám a nem egyenletes megvilágítás okozta árnyékok, továbbá a nem kellően távoli figura és mezőszínek rontják a hatékonyságot. Ennek javítására, illetve a rugalmasabb kezelhetőség érdekében a színátlagot az összehozott RGB átlag mellett az R, G és B csatornán külön-külön is lehet tekinteni. A 4. ábrán a színátlag megváltozása alapján végrehajtott küszöbölés eredményét látjuk két egymást követő játékaláson (világos sánc). Jól látható, hogy a nagy mértékű változások valóban a megtett lépést mutatják.



4. ábra.

Lépés (világos sánc) detektálása a mezők színátlagának változása alapján.

3.2. Változásfigyelés különbségkép alapján

A digitális képfeldolgozás irodalmának jól ismert, a változásfigyelést különbségkép elkészítésével végző technikái [1] is sikeresen alkalmazhatók rendszerünkben. A két egymást követő képet kivonva egymásból, ideális esetben csak a megváltozott mezőkön maradnak releváns információk. Gyakorlati tapasztalataink azt mutatták, hogy a különbségkép használatát az árnyékolás teheti kérdésessé, amint azt az 5. ábra mutatja.

3.3. Változásfigyelés a figuraszínek keresésével

Láthattuk, hogy az előző két eljárás árnyékolás esetén kevésbé megbízható. Ezért kidolgoztunk egy olyan eljárást is, ami a figurák helyének megkeresésére irányul.



5. ábra. A különbségkép ideális és árnyékos esetében.

Ebben az eljárásban azokat a pontokat tartjuk meg, amelyek színének távolsága a magyszíntől adott küszöb alatt van [3]. Az előző példák esetére az eljárás eredményét a 6. ábra mutatja. A megtett lépés természetesen könnyen megha-



6. ábra. A világos figurák helyzete két egymást követő játékállásban.

tározható a figurák helyének megváltozásából. A módszer akkor nem hatékony, ha a figurák és mezők színe közel van.

4. Tovább lépés

Vizsgálunk olyan technikákat, amelyek a fenti eljárásokat tovább javíthatják, például az infravörös/ultraibolya tartományban való feldolgozást. Ha a feladat két emberi játékos játszmányának követése, akkor előfordulhat hogy pár lépés "kiesik". Ez olyan algoritmusokkal korrigálható, amelyek megkeresik két játékállás között lehetséges lépéssorozatokat. Dolgozunk továbbá egy olyan pszichológiai tesztsorozat összeállításán, amelynek segítségével kiválasztható az emberi felhasználó és a gépi látást használó rendszer számára egyaránt optimális színösszeállítás. Folyik a sakkozógép hiányzó részeinek megvalósítása is, így a teszteléshez rendelkezésünkre áll egy helyi fejlesztésű sakkprogram, a lépések megtételéhez szükséges robotkar pedig tervezési fázisban tart.

Hivatkozások

- [1] R.C. Gonzalez and R.E. Woods, Digital image processing, Addison-Wesley, Reading, MA, 1992.
- [2] A. Hajdu, B. Nagy and Z. Zörgő, Indexing and segmenting colour images using neighbourhood sequences, IEEE ICIP 2003, Barcelona, Spain, I/957-960.
- [3] A. Hajdu, T. Tóth, K. Veréb and Z. Zörgő, Distance functions in multidimensional image processing applications, KÉPAF 4 (2003), Miskolc-Tapolca.

Symmetry axis extraction

László Havasi¹, Zoltán Szlavík², Tamás Szirányi², and

Department of Information Technology, PPCU, Budapest

¹havasi@digitus.itk.ppke.hu

Analogical and Neural Computing Laboratory, MTA-SzTAKI, Budapest

²[szlavik;sziranyi]@sztaki.hu

Abstract. We introduce step-by-step a fast method to extract symmetry of human shapes primarily. We show that the algorithm with restrictions is able to present useable features. Symmetry extraction in complex environment isn't a common process especially when the aspect is the fast computation time. On the other hand the speed is very important factor because the symmetry can be considered as a low level feature of the image and it needs some complex processing method to be interpreted. The presented algorithm will be demonstrated with human body analysis on real images. As it will be seen on the results the simplifications won't cause serious damage on the symmetry maps. We process the symmetry information with modified regression formula.

1 Introduction

Symmetry is an elemental attribute and most objects have typical symmetry maps. These unique and invariant properties lead the applicability of symmetries in wide range [1]. From our point of view the biometrical applying is of interest in surveillance systems. In real cases the people cannot be followed for long time and pure attributes can't be extracted. To solve the recognition and identification problem all information is needed. The symmetry of a human body can be detected from each viewpoint. The biometric attributes are unique for the person and allow the recognition [2]. The symmetry is also useable for recognition or discrimination [3].

The Generalized Symmetry Operator (GSO) is used to extract symmetry maps. This operator consists of distance weight function, phase weight function and a logarithmic mapping. An extension can be found in [4] to handle local and global symmetries. The calculation with GSO is expensive and cannot handle the nearest edges. A fast method was published in [5] to estimate the axis of skew.

Our method introduces a new approach to extract symmetries. The main goal is the short computation time by using over-simple operations. The final algorithm does not use larger kernels than 3x3. Beside linear operators we use rank-order filters in the preprocessing state. These operators can be implemented in image processor hardware e.g. [6]. The symmetry extraction algorithm can be run in binary mode and gray mode too.

2 Estimable parts detection

The symmetry operator uses edge map. All approaches analyze the shape of the objects. This step is very difficult because the precise contour or silhouette can't be extracted from real images in most cases. The threshold of edge map is also a difficult task and the GSO tries to solve this by using the logarithmic intensity mapping. Of course it's increasing the computation time. The explanation of results is also difficult when the full edge map is processed. We don't want to separate the object from the background precisely. That's why we are using only a simple running average for motion detection. The main part is to leave the edge fragments of the objects.

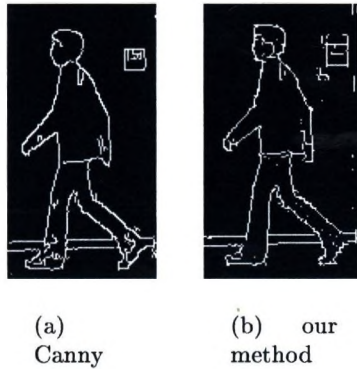


Fig. 1. Result of edge detection compared to Canny

In this chapter we introduce a fast and simple edge detection method that is near to the Canny edge detector [7]. By extracting the local maximums of edge maps we can reduce the input data thus the speed is increasing.

The algorithm is described in Figure 2. To leave the local maximums from the edge map we have to delete all smaller neighboring pixels. For this we are using rank-order filters and image masking. The division of vertical and horizontal edges aids to hold the continuity of edges. Figure 1 shows the difference between the Canny edge detector [8] and our implementation.

3 Symmetry extraction

The described algorithm below approximates the GSO. It can be done because the human body principally contains nearby vertical symmetry parts. Our conception is to calculate symmetry maps by morphological iterations. The illustration of it can be seen in Figure 4. The binary input image is the (a), (b) and (c) contains the results of dilatations. The extinction front of waves can be seen in

(d). We can imagine the dilatation like a wave that started from the edges. These waves can extinguish each other and show the points of the same distances. In the simplified symmetry detection these points mean real symmetry points. The first limitation of the algorithm is the dilatation with 3x3 kernel can't handle the perfect waves that are fully parallel to edges. This limitation insignificant matter when we assume the essentially vertical symmetries.

The major task of the algorithm is to detect the above

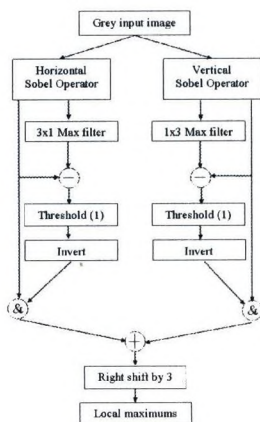


Fig. 2. Detecting local maximum of edges

reviewed extinctions with elementary operators. Figure 3 shows the implemented solution. The morphological operators are defined by 3x1 kernel size for vertical symmetries. The figure demonstrates one cycle so for calculating the full symmetry map

it has to run several times and has to summarize the symmetry points. The input of the next cycle is the output of "Copy back" item that is the dilated input. "Image buffer" represents a one-cycle delay for the differential operator. Thus every cycle are using the difference of the previous cycle in the addition operator. When the input image is a binary edge map (Figure 5a after threshold) the result can be seen in Figure 5b. The mechanism of the waves and the limitations of the binary mode are observable. The main symmetry axis of the two winger lines is missing or rather it is fragmented.

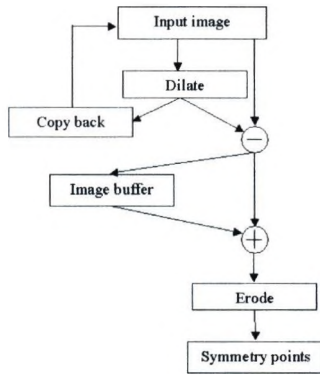


Fig. 3. Iteration cycle of symmetry detection

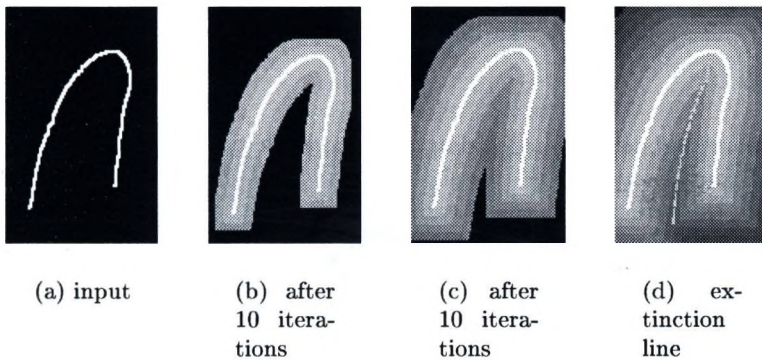


Fig. 4. Waves of dilatation

An extension of the algorithm can be created by assigning weight to the edges. We call this mode gray mode because it needs gray level dilatation (max filter). The weights

can be calculated from the length (or vertical range) of the edges. This mode requires more computation time with e.g. 8 neighboring flood-fill. Figure 5c shows the result after applying the weighting function. Now the higher weighted

waves extinguished the lower waves and the inner symmetry axis was evolved. The advantage of this solution is it's able to return the waves weights in the extinction points. The iteration count determines the symmetry width between the edge points. In gray mode the small edge fragments are unable to cause serious error in the symmetry axis.

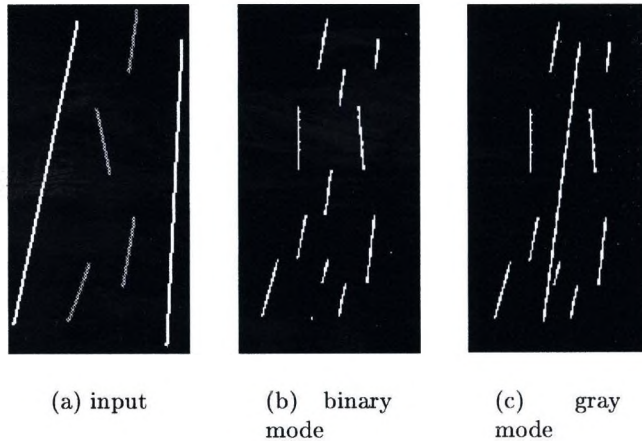


Fig. 5. Horizontal symmetries of artificial image

In binary mode the imprecision can be completed with handling the reflections in the symmetry map. We should keep the fast computation therefore we introduce the symmetry levels: Level 1, Level 2 and Level 3. Level 1 means the above calculated maps and the Level 2 means the symmetry map of Level 1 and so on. The symmetry levels aid to repair noises and detect reflections in adding extra weights to points. These levels can be seen in Figure 6. The symmetry points calculated by waves consider the structure of objects so the local and global symmetries cannot be mixed. With using symmetry levels we can collect local and global symmetries into one map without reflections.

There are some logical rules of levels:

If the Level 3 exists than the Level 1 and Level 2 should be deleted.

If the Level 2 exist and Level 3 misses than the Level 1 should be deleted.

Applying these rules we reach Figure 6d. Necessarily these levels can cover each other and in this case we can increase the weight of symmetry point because the global and local symmetries have a collective axis. One example can be found in the next chapter about using these weighted symmetry points.

4 Estimating symmetry axis

The extracted symmetry attributes allow the main symmetry axis of the human body to be estimated. It can be obtained by applying linear regression to symmetry points. These points have a weighted value about the strength of the symmetry. A more accurate estimation can be calculated if we modify the error function of optimization:

$$Q = \sum_{i=1}^n f(x_i, y_i)(y_i - \hat{y}_i)^2 = \min \quad (1)$$

In this formula the $f()$ means the weight of points, x_i and y_i means the coordinates of extracted symmetry points. With this completion the strong points will be more fitted to regression line. The final equations are the followings:

$$\hat{y}_i = a + bx_i \quad (2)$$

$$b = \frac{\sum_{i=1}^n f(x_i, y_i)(x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sum_{i=1}^n f(x_i, y_i)(x_i - \bar{x}_i)^2}, a = \bar{y} - b\bar{x} \quad (3)$$

$$\bar{x} = \frac{\sum_{i=1}^n f(x_i, y_i)x_i}{\sum_{i=1}^n f(x_i, y_i)}, \bar{y} = \frac{\sum_{i=1}^n f(x_i, y_i)y_i}{\sum_{i=1}^n f(x_i, y_i)} \quad (4)$$

The calculated regression line is the main symmetry axis of the body. In the real computation the meaning of x and y coordinates must be changed otherwise some cases cause division by zero. This axis can be seen in Figure 6.

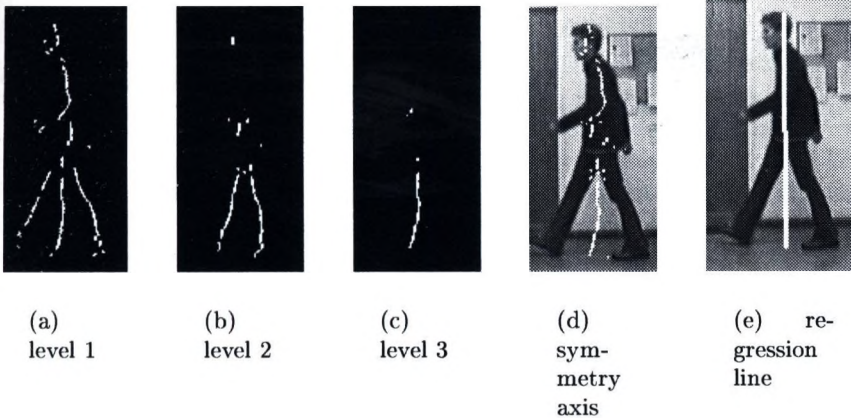


Fig. 6. Symmetry levels and filtered points

5 Conclusions

The described algorithm is an approximate implementation of Generalized Symmetry Operator. We build up the full operator from two parts: edge detection and symmetry map extraction. Both parts are consisting of linear convolution kernels and rank-order filters. This approach enables the efficient implementations for real-time applications. Our symmetry extraction method results more easily interpretable output. Other advantage is it can handle and detect local symmetries without mixing them to global symmetries. By defining symmetry levels we showed the easy way to detect reflections and global symmetries. Finally one of the applications of symmetry was shown detecting symmetry axis of moving human body.

References

1. D. Reissfeld, H. Wolfson, and Y. Yeshurun, "Context-free attentional operators- The generalized symmetry transform", *Int. J. Computer Vision*, vol. 17, 1995, pp. 119-130.
2. James B. Hayfron-Acquah, Mark S. Nixon and John N. Carter, "Human identification by spatio-temporal symmetry", *IEEE ICPR 2002*, Canada, 2002.
3. James B. Hayfron-Acquah, Mark S. Nixon and John N. Carter, "Recognizing human and animal movement by symmetry", *Proc. IEEE ICIP 2001*, Greece, 2001, pp. 290-293.
4. C. J. Parson and M. S. Nixon, "Introducing focus in the generalized symmetry operator", *IEEE Sig. Proc. Letter*, 3.1999, pp. 49-51.
5. T. J. Cham and R. Cipolla, "Symmetry Detection through local skewed symmetries", *Image Vis. Comput.*, vol. 13, 1995, pp. 439-450.
6. Bi-I: Bio-inspired Real-Time Very High Speed Stereo or Mono Image Processing System, www.analogic-computers.com/cgi-bin/sub-pages/products/bi-i.php
7. J. Canny, "A computational approach to edge detection", *IEEE Trans. on Pattern An. and Mach. Intell.*, 8(6) 1986, pp. 679-698.
8. Intel Open Source Computer Vision Library, www.intel.com/research/mlr/research/opencv

Optical Flow meghatározása energiaminimalizációs módszerrel

Horváth Péter, Kató Zoltán

Szegedi Tudományegyetem, Informatikai Tanszékcsoport,
Pf. 652, 6701 Szeged, Fax: 62/546 397

Horvath.Peter.3@stud.u-szeged.hu, kato@inf.u-szeged.hu

Kivonat A cikkünkben *optical flow* becslésére mutatunk be algoritmust melynek legfontosabb rész a Mumford-Shah által bevezetett energia függvény megoldása, melyet kiterjesztünk mozgások detektálására. Az inicializálásra használt vektor mezőre bemutatunk egy *multi-scale* technikát, melyet egy *k-means* klaszterező algoritmus követ. Ezen kezdeti *optical flow*-t, mint inicializálást használjuk az energiaminimalizálás során, melyet *simulated annealing* algoritmussal optimalizálunk.

1. Bevezetés

A mozgás nagyságának és irányának meghatározása nehéz de igen sok gyakorlati alkalmazásban fontos probléma, melyhez gyors algoritmusokra van szükségünk. P. Anandan 1989-ben megjelent cikkében [1] találkozhatunk *multi-scale* technikával, ezen alapötlet továbbfejlesztésével készítettük a cikk második részében bemutatásra kerülő eljárást, mely a mozgás nagyságára és irányára próbál becslést adni. Először bemutatjuk a triviális algoritmust melynek futásideje $\Theta(N^4)$. Könnyebb dolgunk van ha feltesszük, hogy nem lehetnek a képen egy pixelnél nagyobb mozgások, erre is bemutatunk egy technikát, majd megnézzük hogyan tudjuk elérni azt, hogy ne legyenek egy pixelnél nagyobb mozgások. P. J. Burt és E. H. Adelson 1983-ban megjelent cikkében [2] bemutat egy *Gauss-piramis* nevű képkicsinyítési eljárást, mely segítségével felére kicsinyítjük az adott képet, de nem egyszerű átlagolási technikával hanem egy Gauss konvolúciós maszkot használva, ennek többszöri alkalmazásával egy piramist állítunk elő, melynek megfelelően magas szintjén már nincsenek egy pixelnél nagyobb mozgások. Itt meghatározzunk a kis mozgásokat, majd a piramis egyel alacsonyabb szintjére ugrunk és itt finomítjuk a becslésünket és így folytatjuk az eljárást a legalsó szintig. Mivel minden szinten minden egyes pixelnek csak egy nagyságú környezetét kell vizsgálnunk így csak konstans időt töltünk minden pixel vizsgálatával amely azt jelenti, hogy a futásidő $\Theta(N^2 \log^2 N^2)$ -re csökken. Az így kapott becslésünk bár jól közelíti a valóságot de sok hibaforrás lehetséges, zajos képen előfordul, hogy egy-egy vektor nem mutat korrekt eredményt azaz egészen más irányba mutat mint a környezetében lévő szomszédok ezt a hibát simítással lehet javítani, a másik hibája az algoritmusnak, melyen még a simítás sem segít: a

homogén mozgó területek illetve a háttérbe beleolvadó mozgó objektumok detektálása, mivel szín alapú a folyamat, nem érzékelheti ezeket. Mindenzen hibák miatt döntöttünk energiainimalizációs módszerek használata mellett. Bemutatjuk Mumford-Shah által kidolgozott szegmentálási eljárást [6], melyet átdolgoztunk optical flow meghatározására. F. Gibou és R. Fedkiw [5] publikáltak egy módszert melyben *k-means* algoritmus alkalmazása után végezték a szegmentálást, módszerünkben mi is *k-means* algoritmussal inicializáljuk az energiainimalizáló függvény klasztereit. Miután megtörtént a kezdeti osztályok beállítása a függvény minimalizálására több lehetőségünk is van, cikkünkben *Simulated Annealing* felhasználásával minimalizálunk, de T. F. Chan és L. A. Vese [3] cikkében található *Level-Set* eljárás segítségével is megoldható a probléma. N. Paragios [7] 2000-ben publikált PhD dolgozatában foglalkozik aktív kontúrok energiájának minimalizációjával, *Level-Set* függvények optimalizálásával, mozgás detektálásával. D. Cremers [4] PhD dolgozatában az aktív kontúrokat további jellemzők hozzáadásával bővíti, így a priori tudást visz a rendszerbe, mely működését mozgás detektálására is bemutatja.

2. Optical flow becslése

Az optical flow egy képsorozat két időben egymást követő képkockáján végbemenő mozgásokat mutató vektormező. Jelölje I_1 és I_2 a képsorozat első és második szekvenciáját, melyek között eltelt Δt idő alatti mozgások vektoraira fogunk becslést adni. $\overline{OF}[i, j]$ jelölje azt, hogy az I_1 kép $[i, j]$. pixelén lévő objektum Δt idő alatt milyen irányban és mennyit mozgott, azaz hol található az I_2 képszekvencián ($[i, j] + \overline{OF}[i, j]$).

2.1. Triviális algoritmus a mozgás meghatározására

A triviális algoritmus az első képkocka minden pixelére megvizsgálja a második képen levő összes pixelt és közülük a leghasonlóbbat választja ki:

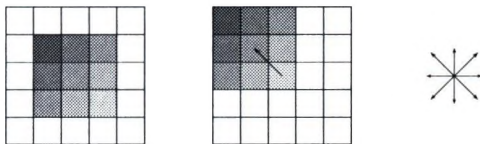
$$\overline{OF}[i, j] = \arg \min_{\substack{k=1 \dots N-1 \\ l=1 \dots N-1}} |I_1[i, j] - I_2[k, l]| - [i, j]. \quad (1)$$

$I_1[i, j]$ és $I_2[i, j]$ az első és második képszekvencia pixeljei. A triviális algoritmus futásideje a kép pixelszámának másodfokú polinomiális függvénye, $\Theta((M \times N)^2)$, ahol $M \times N$ a kép mérete. Ez gyakorlatban nem használható, ezért a probléma megoldására most bemutatunk egy $\Theta(N^2 \log^2 N^2)$ futásidejű algoritmust.

2.2. Maximálisan egy pixel nagyságú mozgások meghatározása

Ha feltételezzük, hogy a képen egy pixel nagyságúnál nagyobb mozgások nem lehetnek, akkor könnyebb dolgunk van. Ebben az esetben az előző részben bemutatott módszert használva számíthatjuk a mozgásvektorokat, de csak egy pixel nagyságú környezetet vizsgálva, így két dimenziós mozgásvektorokat kapunk, melyek értékei $(-1 \dots 1)$ közöttiek lesznek. Ez a módszer nagyon gyors eredményt

szolgáltat, de nem minden esetben korrekt. Ennek javított változata amikor a keresett pixel valamilyen környezetét vizsgáljuk (1. ábra), célszerű ezt súlyozni egy Gauss normális eloszlású mátrixszal, mely mátrix elemeinek összege 1. A



1. ábra. Eljárás maximálisan egy pixel nagyságú mozgás meghatározására.

normál eloszlás sűrűségfüggvénye segítségével számoljuk ki a Gauss mátrixot a következőképpen:

$$G[x, y] = f_{m, \sigma}(x, y) \tag{2}$$

$$G'[x, y] = G[x, y] / \sum_{k, l = -\lfloor R/2 \rfloor}^{\lfloor R/2 \rfloor} G[x, y], \tag{3}$$

ahol R a környezet nagysága, így a mozgásvektorok keresése a következőképp zajlik. $OF[i, j] = [x, y]$, ahol $[x, y]$:

$$\arg \min_{\substack{-1 \leq x \leq 1 \\ -1 \leq y \leq 1}} \sqrt{\sum_u \sum_v G'[u, v] (I_1[i + u, j + v] - I_2[i + u + x, j + v + y])^2}. \tag{4}$$

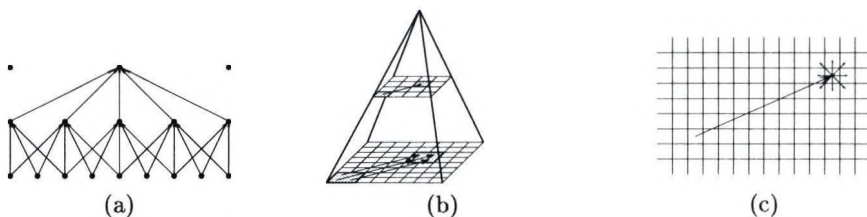
Az eddigiek során megismertük hogyan lehet maximum egy pixel nagyságú mozgásokat meghatározni, most megnézzünk egy eljárást, mely segítségével a nagyobb mozgásokat egy pixel nagyságúra redukálhatjuk.

2.3. Piramis nagyságának meghatározása és felépítése

Tekintsünk egy képet, amelyen egy x nagyságú mozgást találhatunk, ha ezt a képet n -szeresére nagyítjuk, akkor a mozgás is n -szeres lesz. Ezt az állítást felhasználva ha felére csökkentjük a képet akkor a mozgások is felére csökkenek. Így ha a képet x -edére csökkentjük, akkor a mozgás pont 1 egység nagyságú lesz. Itt bemutatunk egy eljárást, amely segítségével igen gyorsan felére csökkenthetjük a kép méretét, az eljárás a Gauss piramis [2] felépítésének része.

Az elkészítéshez egy normál eloszlású $N(0, 1)$ paraméterű fentebb bemutatott Gauss mátrixot használunk. Egy $2n + 1 \times 2n + 1$ nagyságú képből indulunk és egy $n + 1 \times n + 1$ mértűt készítünk. Konvolúciót alkalmazunk, a (2 a). ábrán látható az 1 dimenziós megvalósítás.

Az eljárást alakalmazva elértük, hogy a képet lineáris futásidejű algoritmussal felére csökkentettük. Itt jegyezzük meg, hogy több más lehetőségünk is van a kép átméretezni, például a lefedett pixelek átlaga esetleg súlyozott átlaga. Így



2. ábra.

A Gauss piramis kiszámításának menete (a) ábra, a visszafelé dolgozó stratégia (b) ábra, a becslés finomítása (c) ábra.

elértük azt, hogy a mozgások a képen a felére csökkentek. A gondolatmenetet tovább folytatva a kapott képet ismét felér csökkentjük és így a mozgások negyedére, nyolcadára, ... csökkennek. Így az x nagyságú mozgás a k . lekicsinyítés után $x/2^k$ nagyságú lesz. Már megismertük a maximum 1 pixel nagyságú mozgások detektálását. Ha tudjuk, hogy maximum k nagyságú mozgások vannak a képen akkor $\log_2 k$ darab képből kell állnia a piramisnak, így a $\log_2 k$. képen már csak maximum 1 pixel nagyságú mozgások lesznek.

2.4. Visszafelé dolgozó stratégia

A piramis felépítése során az utolsó képen meghatározzuk a maximum 1 pixel nagyságú mozgásokat, ezután átugrunk az utolsó előtti szintre, itt az imént megállapított mozgások kétszer akkora lesznek ám ez nyilván csak egy durva becslés. Tehát az $i + 1$. szinten lévő Optical Flow-ból úgy készítjük a nagyobb i . szinten lévő, hogy az $OF_i(2k, 2l)$, $OF_i(2k + 1, 2l)$, $OF_i(2k, 2l + 1)$ és $OF_i(2k + 1, 2l + 1)$ elemei megkapják az $OF_{i+1}(k, l)$ vektor hosszának kétszeresét (2 b. ábra). Ezek után jön az algoritmus egyik kulcs lépése, ez a finomítás, amely során a vektorokról lévő durva becslésünket finomítjuk. A finomítás a durva becslés 3×3 -as környezetében megkeresi pontosan melyik pixel is lehet a második képen a mozgás korrekt helye. Ezt pedig a 2.2 fejezetben ismertetett maximálisan egy pixel nagyságú mozgások detektálására szolgáló módszer segítségével végezzük (2 c. ábra).

3. Az Optical Flow simítása a Mumford-Shah energiaminimalizációs eljárás segítségével

Mumford és Shah [6] bemutatott egy általános szegmentálási módszert, a szegmentálást egy függvény segítségével írták fel:

$$E(f, \Gamma) = \mu^2 \int \int_R (f - g)^2 dx dy + \int \int_{R-\Gamma} \|\nabla f\|^2 dx dy + \nu |\Gamma|, \quad (5)$$

ahol f a szegmentálás, g az eredeti kép, Γ a határ a régiók között és R a régiók összessége. Az első rész kicsi, ha a g hasonlít az f -re, a második a simaságot

biztosítja és a harmadik a kontúr hosszának minimalizálásáról gondoskodik. Itt bemutatjuk az M-S speciális esetét optical flowra:

$$e = \alpha \int \int_{\Omega} (I_2(x, y) - I_{shynt}(x, y))^2 dx dy + \beta \int \int_{\Omega \setminus \Gamma} E_{OF}(x, y) dx dy + |\Gamma|, \quad (6)$$

ahol α és β súlyparaméterek, I_{shynt} egy olyan kép, melyet az első képszekvencia és az optical flow segítségével készítünk el a következőképpen $\forall i, j$:

$$I_{shynt}(i + OF(i, j).x, j + OF(i, j).y) = I_1(i, j). \quad (7)$$

Legyen E_{OF} az optical flow parciális deriváltja, ez a homogén területeken kis értékeket vesz fel viszont a határokon az értéke magas. Közelítsük a következőképpen:

$$E_{OF} = \sqrt{S_1(OF.x)^2 + S_2(OF.x)^2 + S_1(OF.y)^2 + S_2(OF.y)^2}, \quad (8)$$

ahol S_1 és S_2 a Sobel gradiens operátorok. Az egyenlet harmadik része $|\Gamma|$, ahol Γ a határpontok halmaza és $|\Gamma|$ ezen pontok száma, azaz a kontúr hossza.

4. Az Optical Flow, k-Means és a Mumford-Shah algoritmusok kapcsolata

A 2 fejezetben bemutattuk hogyan tudjuk az optical flow-t becsülni. A 3 fejezetben láthattuk a Mumford-Shah energiafüggvényt és optimalizálását. Láthattuk, hogy az M-S eljárásához a kezdeti kép klaszterezésére van szükségünk. Erre a célra használunk az u.n. *k-means* klaszterező eljárást [5], mely segítségével az optical flow vektorait osztályokba soroljuk, majd ezen osztályokkal mint az M-S inicializáció lépése oldjuk a meg a problémát. Az algoritmus:

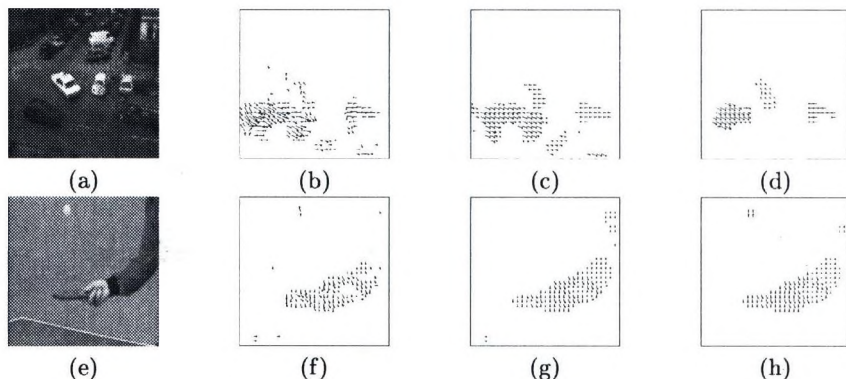
```
Optical Flow(Image 1, Image 2): OF
  Estimation(Image 1, Image 2) =: OF;
  K-means(OF, int Clusters);
  Mumford-Shah(Image 1, Image 2, OF) =: OF;
    a. Fast Marching;
    b. Simulated Annealing;
    c. ...
return OF;
```

A *k-means* algoritmusban a *Clusters* a keresett osztályokra ad felső korlátot. Az M-S megoldásánál több módszer közül válszthatunk [7].

5. Kísérleti eredmények

Az energiafüggvény inicializálását a 2 fejezetben bemutatott algoritmussal végezzük, majd *k-means* algoritmussal klasztereztük a vektorokat. Az energiafüggvény minimalizálását szimulált hűtéssel végeztük. Két tesztet mutatunk be az egyik a jól ismert hamburgi taxi sorozatból való a másikon egy asztalitenniszész játékossal.

Hamburg taxi sorozat: Az eredeti kép a (3 a). ábrán látható. A sorozat 15. és 18. szekvenciája között számoltuk az optical flow-t. A becslést a (b) ábrán láthatjuk, esetek túlnyomó többségében helyesek a vektorok, de mivel a kép eléggé zajos találunk hibákat. A (c) ábrán a k-means algoritmus eredménye, kezdetben 10 osztállyal, és a (d) ábrán a szimulált hűtés eredménye.



3. ábra.

A hamburgi taxi sorozat 15. szekvenciája (a), a becsült optical flow (b), a k-means klaszterezés eredménye (c) és az M-S megoldása szimulált hűtéssel (d).

Az asztaliteniszező kép (e), az optical flow becslése (f), a k-means eredménye (g) és az M-S enegiafüggvény szimulált hűtéssel optimalizálva (h).

Asztaliteniszező játékos: A (3 e, f, g és h) ábrákon láthatjuk a képeket, a minősége jobb mint a taxi sorozaté, így a legtöbb alkalmazásban már a (3 f) ábrán látható becslés is jól használható, a (g) ábrán a k-means utáni klaszterezés és a (h)-n a szimulált hűtés eredménye.

Hivatkozások

1. P. Anandan: A Comutational Framework and an Algorithm for the Measurement of Visual Motion, *Int. J. Computer Vision*, Vol. 2, No. 3, pp. 283-310
2. P. J. Burt and E. H. Adelson: The Laplacian Pyramid as a Compact Image Code, *IEEE Transactions on Communications*, Vol. COM-31, no. 4, april 1983 pp. 532-540
3. T. F. Chan, B. Y. Sandberg, L.A. Vese: Active Contours without Edges for Vector-Valued Image, N00014-96-1-0277, DMS-9626755.
4. D. Cremers: Statistical Shape Knowledge in Variational Image Segmentation, PhD Thesis 2002, Mannheim.
5. F. Gibou, R. Fedkiw: A Fast Hybrid k-Means Level Set Algorithm for Segmentation
6. D. Mumford and J. Shah: Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems, *Communications on Pure and Applied Mathematics*, Vol. XLII pp. 577-685, 1989.
7. N. Paraigos: Geodesic Active Regionas and Level Set methods, PhD Thesis 2000

Szín, textúra és mozgás alapú szegmentálás Gradient Vector Flow segítségével

Horváth Péter, Kató Zoltán

Szegedi Tudományegyetem, Informatikai Tanszékcsoport,
Pf. 652, 6701 Szeged, Fax: 62/546 397

Horvath.Peter.3@stud.u-szeged.hu, kato@inf.u-szeged.hu

Kivonat Az *aktív kontúr* azaz *snake* közkezdvelt eszközök a képfeldolgozásban és szegmentálásban. Az aktív kontúr olyan görbe amelyekhez egy energiát tudunk rendelni, mely energia a görbe mozgatásával változik, így az energia minimalizálásával alkalmas élek detektálására, képek szegmentálására, felületek modellezésére (3D-ben). Két alapvetően különböző aktív kontúr létezik, paraméteres aktív kontúr (*parametric active contour*) és geometriai aktív kontúr (*geometric active contour*). A Snake energiáját két komponens alkotja, a belső energia (*internal force*), amelyet a kontúr alakja befolyásol és a külső energia (*external force*), mely a kontúr környezetétől függ. Cikkünkben a *Gradient Vector Flow (GVF)* nevű eljárás kiterjesztéseit mutatjuk be ($\mathbb{R}^2 \rightarrow \mathbb{R}^n$). Az eredeti módszer szürkeárnyalatos képek szegmentálására alkalmas, bemutatunk egy lehetőséget mely segítségével több dimenziós esetekben is használhatjuk a *GVF*-et, majd néhány példát *CIE L*u*v** színtérben, *Optical Flow* szegmentálására és textúrázott képek esetére. Bemutatunk egy technikai megoldást amely segítségével gyorsíthatunk az algoritmuson.

1. Bevezetés

Az Aktív kontúrok olyan görbék, melyek a kép területén mozognak abban az irányban, amerre az őket mozgató energiák mutatnak. Kass, Witkin és Terzopoulos 1987-ben megjelent cikkében elsőként [3] mutatják be a paraméteres aktív kontúr modellt. Terzopoulos további cikkeiben [6] [7] foglalkozott alakmodellezéssel és a jelen cikk egyik fő részét képező mozgás követéssel. A tradicionális *snake* modellek problémája, hogy vannak olyan konkáv alakzatok, melyekre körülményes a használatuk. 1998-ban C. Xu és J. L. Prince publikálták a Gradient Vector Flow [8] előállításáról szóló cikküket. A módszer segítségével tetszőleges konkáv alakzatokra ráhúzódo *snake* készíthető. A *GVF*-et szürke árnyalatos képek szegmentálására használhatjuk, itt bemutatunk egy módszert melyet olyan általános esetekben tudunk használni, amikor több jellemző áll rendelkezésünkre (pl.: RGB, CIE L*u*v*, Optical Flow,...).

2. Aktív kontúrok

Ebben a fejezetben bemutatjuk a tradicionális paraméteres *snake* modellt és annak hibáit, ismertetjük a Gradient Vector Flow-t majd a több dimenziós ki-

terjesztését. Végül egy olyan snake struktúrát mutatunk be, mely könnyen megvalósítható és gyors.

2.1. Paraméteres snake modell

A snake egy olyan paraméteres görbe, amely a következőképpen írható fel $\overline{x(s)} = [x(s), y(s)]$, $s \in [0, 1]$, ezen függvény energiáját a következőképpen definiáljuk:

$$\int_0^1 \frac{1}{2} (\alpha |x'(s)|^2 + \beta |x''(s)|^2) + E_{ext}(x(s)) ds, \quad (1)$$

ahol α és β súlyparaméterek, az $x'(s)$ és $x''(s)$ az első és második derivált a kontúr feszességét és merevségét jellemzi. A külső energiát a következőképpen definiálták:

$$E_{ext}(x, y) = -|\nabla I(x, y)|^2, \quad (2)$$

azaz a kép gradienseinek értékei adják az energiát. Az energiafüggvény minimalizálásáról bővebben [8]-ben olvashatunk.

Problémák a klasszikus aktív kontúrral. A snake egy vektormező mentén mozog, ezt a vektormezőt kell megkonstruálnunk. Erre a klasszikus megoldás egy távolság definícióval minden pontból egy alacsonyabb energiájú rész felé mutató vektor, a távolság lehet Euklideszi vagy Chamfer definíciója alapján [1]. A probléma ezekkel a módszerekkel, hogy az alakzatok egyes részein nem megfelelően használható a vektormező. Erre a problémára található megoldást a [8]-es cikkben.

2.2. Gradient Vector Flow

Egy iteratív módszer, mely a konkáv problémák esetében segít a jó megoldást megtalálni. Egy adott (x, y) pontban a vektor irányát a következőképpen kaphatjuk meg:

$$\vec{x}_t(s, t) = \alpha x''(s, t) - \beta x''''(s, t) + \vec{v}, \quad (3)$$

ahol $v(x, y) = E_{ext}$ a külső energia.

Gradient Vector Flow. C. Xu és J. L. Prince cikkükben [8] definiáltak egy éltérképet, legyen $f(x, y)$ az $I(x, y)$ éle, amelyet például a Sobel operátorok segítségével könnyedén kiszámolhatunk

$$f(x, y) = \sqrt{s_1(i, j)^2 + s_2(i, j)^2}. \quad (4)$$

A GVF egy $\vec{v}(x, y) = [u(x, y), v(x, y)]$, amely vektorok minimalizálják a következő egyenletet:

$$\varepsilon = \int \int \mu (u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\vec{v} - \nabla f|^2 dx dy \quad (5)$$

A (1 a, b és c) ábrán GVF segítségével előállított képek láthatók, de az a ábrán látható körvonal a 2.3. fejezetben ismertetett módszerrel készült, azaz új pontok beszúrásával és felesleges pontok elhagyásával.

2.3. Pontok felvétele és elhagyása a GFV snake-ből

Az itt bemutatott gyakorlati megvalósítás inicializálásakor felveszünk pontokat, melyeknek nem kell szomszédos pixelen lenniük (ezek alkotják), de egy listában vannak tárolva így a sorrendjük szigorúan rögzített. Ezeket a pontokat fogjuk a GVF által kreált vektormezőn mozgatni.

Új pontok felvétele. Amikor inicializáltuk a snake-et és elkezdjük iteratív lépésekkel a GVF által meghatározott irányába mozgatni előfordulhat, hogy két pont nagyon eltávolodik egymástól és ekkor a közöttük lévő területre nem jut olyan pont amely ráhúzódhatna a korrekt pozícióra. Ez főleg "U" alakú részen és szakadások közelében figyelhető meg (1 e). Ekkor beszúrunk egy újabb pontot a kontúrba, úgy, hogy az a két eltávolodott pontot összekötő egyenesen a pontok között középen legyen. Ezzel nem rontunk a snake állapotán, és az új pont elindulhat a konkáv részek irányába. A módszer egy lehetséges implementálása, ha megkeressük a k legnagyobb távolságot a snake mentén és oda beszúrunk egy-egy pontot. Tapasztalataink szerint egy n pontot tartalmazó kontúrnál $k = n/10 \dots n/100$ választása a legjobb. A (1 e) ábrán látható a beszúrás nélküli eredmény. Megfigyelhető, hogy a szakadásba nem jutott pont és a konkáv, de "U" alakú területekre nem konvergált be a snake.

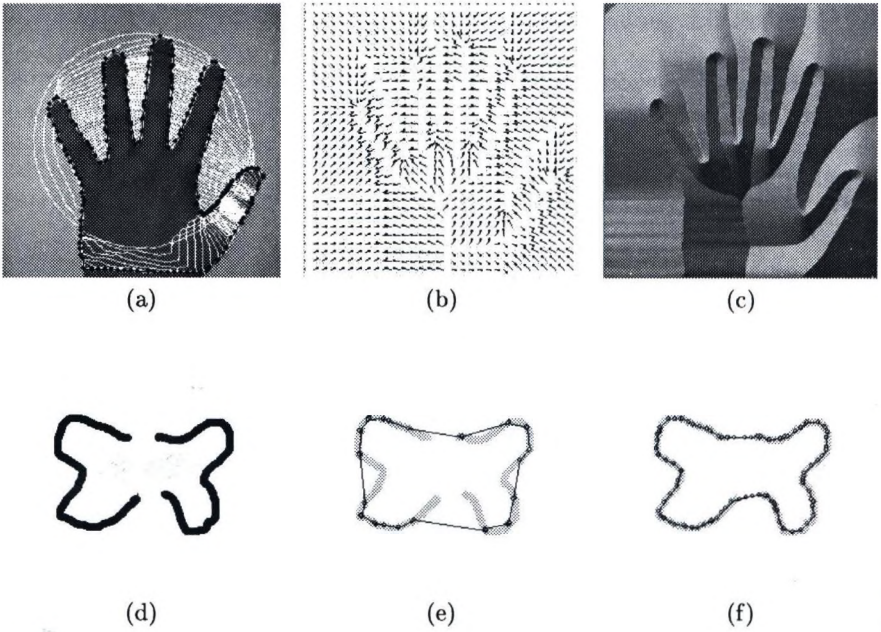
Pontok törlése. Ha minden iterációban beszúrunk k pontot, akkor az i . iterációban már $n + ik$ db pontunk lesz, mely nem rontja el a snake jellemzőit, de a számítási időt jelentősen befolyásolhatja. Ezért érdemes minden iterációban kitörölni a felesleges pontokat. Ezek olyan pontok amelyek túl közel esnek egymáshoz. Azaz minden (p_{i-1}, p_{i+1}) párosra vizsgáljuk meg milyen közel esnek egymáshoz, és a k legkisebbre töröljük ki a p_i pontot.

2.4. Gradient Vector Flow kiterjesztése több dimenziós jellemzővektorokra

Az eddig ismertetett módszer $\mathcal{R}^2 \rightarrow \mathcal{R}$ -be történő leképezésekre, azaz szürkeárnyalatos képekre lett definiálva. Az optical flow például $\mathcal{R}^2 \rightarrow \mathcal{R}^2$ képező függvény, de az RGB vagy a CIE $L^*u^*v^*$ $\mathcal{R}^2 \rightarrow \mathcal{R}^3$, azaz a GVF-et nem tudjuk közvetlenül fehasználni, ezért általánosítottuk több dimenzióra. Feladatunk a következő: adott $f(x, y) : \mathcal{R}^2 \rightarrow \mathcal{R}^n$ és ehhez készítsük el a GVF-et. A szürkeárnyalatos kép ¹ parciális deriváltjévésszük, ez több dimenziós függvények esetében ez egy Jacobbi mátrix, és a következőképpen írható fel:

$$Jac(f(x, y)) = \begin{pmatrix} \frac{\partial f_1(x, y)}{\partial x} & \dots & \frac{\partial f_n(x, y)}{\partial x} \\ \frac{\partial f_1(x, y)}{\partial y} & \dots & \frac{\partial f_n(x, y)}{\partial y} \end{pmatrix}_{2 \times n}, \quad (6)$$

¹ A szürkeárnyalatos képet most tekintjük egy $f : \mathcal{R}^2 \rightarrow \mathcal{R}$ függvénynek.



1. ábra.

Gradient Vector Flow segítségével előállított képek. Az a ábrán egy kézfejre ráhúzott snake, b-n a vektormező és a c-n a vektorok színskálán ábrázolva. Konkáv alakzat szakadással látható a d ábrán, e ábra a GVF snake beszúrás és törlés nélkül ($n = 200$), f ábrán beszúrás és törléses GVF snake ($n = 200, k = 2$).

legyen $M \in \mathbb{R}^{(2 \times n)}$, és $M = Jac(f(x, y))$. Bontsuk fel az M -et 2 sorvektorára, ezek legyenek $M_x, M_y \in \mathbb{R}^n$:

$$M_x = (M_{1,1}, \dots, M_{1,n}), M_y = (M_{2,1}, \dots, M_{2,n}), \quad (7)$$

így a parciális deriváltak koordinátáinként a két vektorban vannak. Ezután mivel lehetséges, hogy bizonyos jellemzőket nem ugyanakkora súllyal akarunk figyelembe venni, súlyozzuk egy α súlyvektorral ezeket, legyen $\alpha \in \mathbb{R}^n$. Ebből a GVF-nél használt $f_x(x, y), f_y(x, y)$ iránykomponenseket úgy kapjuk meg, hogy vesszük az M'_x, M'_y vektorok normáit, azaz:

$$f_x(x, y) = \|M_x\|_k, f_y(x, y) = \|M_y\|_k, k = 1, 2, \infty, max. \quad (8)$$

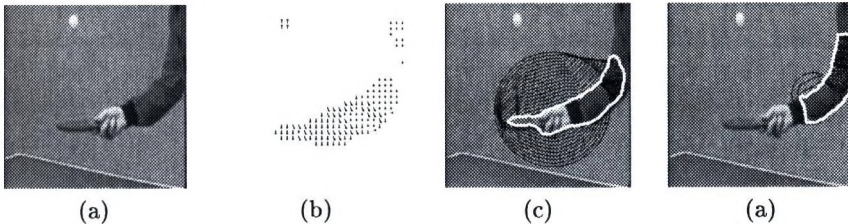
3. Kísérleti eredmények

A kísérletek során különböző több dimenziós jellemzőket vizsgáltuk, ezek:

- optical flow

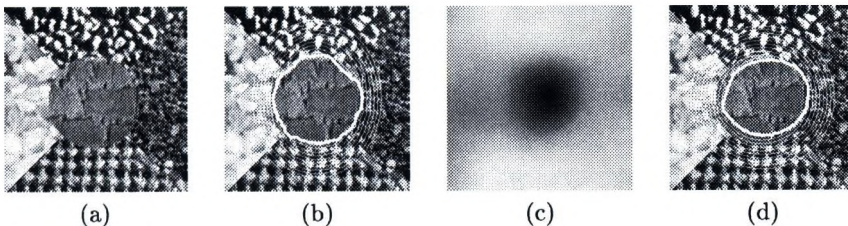
- szín (CIE $L^*u^*v^*$ színtérben [5])
- textúra *Gabor-filter* [2] és *MrSAR* (Multiresolution Autoregressive modell) [4] szűrők felhasználásával

Szegmentálás mozgás alapján. A (2 a). ábrán egy asztaliteniszező játékos látható, mely egy képsorozat egy szekvenciája, a (2 b). ábrán a kép és a rákövetkező kép közötti mozgások vektorai láthatók. Végül a szegmentálás eredménye a (2 c). ábrán. A Gradient Vector Flow paraméterei: $n = 200, k = 3$. A kontúrt inicializálásakor kör alakú volt, és a játékos kezét akartuk szegmentálni az optical flow segítségével, így a két jellemző az optical flow vektorainak x és y koordinátái az α súlyvektor komponensei egyenlők voltak. A (2 c) ábrán látható a snake mozgása, a fekete kontúrokat minden 5. iteráció után rajzoltuk ki. A snake kb. 100 iteráció után bekonvergált.



2. ábra.

Kép szegmentálása mozgás alapján, az a ábrán az eredeti kép látható a b-n az optical flow, a c-n pedig a szegmentálás eredménye, és a kontúr mozgása, a d ábrán a játékos karjának szegmentálása szín alapján.



3. ábra.

Kép szegmentálása textúra alapján, az a ábrán láthatjuk az eredeti képet, a b-n az MrSAR algoritmussal kinyert jellemzők alapján szegmentált képet, a c ábrán a Gabor-filter által kinyert jellemzők a d-n pedig az ezek segítségével készített szegmentálás látható.

Textúra alapú szegmentálás. A (3 a és b). ábrákon textúra alapú szegmentálást láthatunk. Az eredeti kép különböző rendezett és végtelen textúrák felhasználásával szintetikusán lett előállítva. A textúrázott képből MrSAR algoritmus segítségével nyerték ki a textúrajellemzőket. Az MrSAR algoritmus főleg a véletlen textúrák feldolgozásakor használható eredményesen, ha a képen rendezett textúrák vannak akkor Gabor-filter használata előnyösebb. A (3 a) képen a közepén található kör alapú textúra részt akartuk szegmentálni, a (3 b) ábrán láthatjuk a szegmentálás eredményét és a kontúr mozgását, a snake alakja inicializáláskor ellipszis volt. A (3 c) ábrán Gabor-filter segítségével kinyert jellemzőket láthatjuk, a (3 d) ábrán pedig a Gabor-filterek felhasználásával szegmentált képet láthatjuk. Mivel szintetizált képet szegmentálunk a hibát könnyedén tudjuk mérni, a következő táblázat tartalmazza a kontúr átlagos átmérőjét, az pontok szórását és a futásidőt.

Név	Sugár (r)	Szórás (σ)	Futásidő (ms)	Lefedett terület (százalék)
Eredeti	64	-	-	-
MrSAR	59.8377	2.1756	6054	89.4
Gabor-filter	55.1819	3.5	6008	86.1

Szegmentálás szín alapján. A (2 d) ábrán az asztaniteniszező karjának szegmentálását találjuk meg, látható, hogy nem sikerült az egész karra ráhúzódnia a kontúrnak, mert felül az alakzat konvex, azaz szélesedik az átmérője. A színnek CIE L*u*v* színtérbe lettek konvertálva. A kontúr inicializálása egy körrel történt, a kör a játékos karján helyezkedik el.

Hivatkozások

1. H. Borgeforst: Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10. No. 6., November 1988.
2. A. K. Jain and F. Farrokhnia, „Unsupervised texture segmentation using Gabor filters,” *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.
3. M. Kass, A. Witkin, D. Terzopoulos: Snakes: Active contour models, *Int. J. Comput. Vis.*, vol. 1, pp. 321–331, 1987.
4. J. Mao and A. K. Jain, „Texture classification and segmentation using multiresolution simultaneous autoregressive models,” *Pattern Recognition*, vol. 25, no. 2, pp. 173–188, 1992.
5. S. J. Sangwine and R. E. N. Horne, editors. *The Colour Image Processing Handbook*. Chapman & Hall, 1998.
6. D. Terzopoulos, K. Fleischer: "Deformable models," *Visual Comput.*, vol. 4, pp. 306–331, 1988.
7. D. Terzopoulos, R. Szeliski: "Tracking with Kalman Snakes," in *Active Vision*, Cambridge, MA: MIT Press, 1992, pp. 3–20.
8. C. Xu and J. L. Prince: Snakes, Shapes, Gradient Vector Flow, IEEE Transactions on Image Processing, Vol. 7, no. 3, march 1998 pp. 359–369

Precise Registration based on Photo-Consistency

Zsolt Jankó and Dmitry Chetverikov

Computer and Automation Research Institute and
Eötvös Loránd University
Budapest, Hungary
{janko,csetverikov}@sztaki.hu

Abstract. The following data fusion problem is considered: Given a 3D geometric model of an object and two uncalibrated images of the same object, and assuming that the object surface is textured and Lambertian, precisely register the images to the model. Solving this problem is necessary for building a geometrically accurate, photorealistic model from laser-scanned 3D data and high quality images. We generalise the photo-consistency approach by Clarkson et al. [1] to the case of uncalibrated cameras, when both intrinsic and extrinsic parameters are unknown. This gives a user the freedom of taking the pictures by a conventional digital camera, from arbitrary positions and with varying zoom. We propose a method based on manual pre-registration followed by a genetic optimisation algorithm. The issues of speed and robustness are addressed. Results for real data are shown.

1 Introduction

This paper is devoted to a specific type of data fusion of different modalities. The current study is a part of a project to build rich photorealistic 3D models of real-world objects based on multimodal sensor data. In our view, a photorealistic model has three major components, geometry, appearance and dynamics, which must meet the following requirements: precision, continuity, high-level description (*geometry*), texture, realistic surface models, presentation at varying level of detail (*appearance*), motion, deformable shapes (*dynamics*). In this paper, we address the problem of combining precise geometry with high quality images.

A possible **scenario** we consider is the following: Use an accurate 3D scanner to obtain a dense set of surface points covering the whole surface. Triangulate the point set to obtain a 3D model as a triangular mesh with the surface normals assigned. Separately and independently, use an uncalibrated digital camera to acquire a collection of high-quality colour images showing the whole object. Finally, precisely register the images to the 3D model to obtain a geometrically accurate, photorealistic model.

In this study, we consider the special case of a 3D model and two images, but we formulate and approach the underlying mathematical problem in a very general way. The solution we propose can be applied to a 3D model of any origin, measured or synthetic, and to the situation when *all* parameters of the camera are unknown and may vary between the two views. If desired, two different uncalibrated cameras can be used. This degree of freedom is an essential

generalisation of the photo-consistency based registration method by Clarkson et al. [1] which needs a **calibrated setup**. (Due to the paper size limitations, the reader is referred to this paper for a survey of relevant papers.)

The **contributions** of this paper are as follows: (1) The registration problem [1] is generalised to uncalibrated images, which leads to photo-consistency optimisation in 22-parameter space. (2) A genetic optimisation algorithm is selected and successfully applied. (3) Solutions are proposed to robustify the photo-consistency based cost function and speed up its calculation.

2 Problem formulation

In this section we give a mathematical formulation of the registration problem. The input data consists of two colour images, I_1 and I_2 , and a 3D model. The images and the model represent the same object; an example of input data is shown in figure 1. We assume that the images are taken under fixed lighting conditions and the cameras have identical sensitivity. All other camera parameters, the intrinsic and the extrinsic, may differ. (In practice, the images will normally be taken with the same camera from different viewpoints and distances, and with different zooms.) Furthermore, it is assumed that the surface of the object is textured and Lambertian. The 3D model consists of a 3D point set \mathcal{P} and a set of normal vectors. We obtain \mathcal{P} by a hand-held 3D scanner. The point set is then triangulated by the robust algorithm [4] which provides the normal vectors.



Fig. 1. The Frog dataset. Center: 3D model. Sides: Image pair.

The pinhole camera model is used to represent the projection of the object surface to the image plane. If \mathbf{X} is a surface point and \mathbf{u} an image point, the projection can be written as $\mathbf{u} \simeq P\mathbf{X}$, where P is the 3×4 **projection matrix** [3]. (\simeq means that the projection is defined up to an unknown scale.)

The task of the registration is to determine the precise projection matrices, P_1 and P_2 , for both images. By these matrices the images can be back-projected onto the surface resulting in a textured 3D model.

The projection matrix P has 12 elements but only 11 degrees of freedom (DOFs), since it is up to a scale factor. Decomposing P as $P = K[R] - Rt$ shows the meaning of these DOFs [3]: K is the 3×3 upper triangular camera matrix, R the 3×3 rotation matrix and t the 3×1 translation vector. The elements of K are the intrinsic camera parameters, while R and t are the extrinsic camera

parameters, namely the orientation and the position of the camera. We have 5 intrinsic and 6 extrinsic parameters which are unknown. Let us denote by p the collection of these parameters. p represents the projection matrix P as an 11-dimensional **parameter vector**.

We search for values of p_1 and p_2 such that the images are *consistent*. When the surface is Lambertian, consistency means that the corresponding points – different projections of the same 3D point – have the same colour value. Formally, we say that images I_1 and I_2 are consistent by P_1 and P_2 (or p_1 and p_2) if for each $X \in \mathcal{P}$: $\mathbf{u}_1 = P_1\mathbf{X}$, $\mathbf{u}_2 = P_2\mathbf{X}$ and $I_1(\mathbf{u}_1) = I_2(\mathbf{u}_2)$. (Here $I_i(\mathbf{u}_i)$ is the colour value in point \mathbf{u}_i of image I_i .) This type of consistency is called **photo-consistency** [5, 1].

The photo-consistency holds for accurate estimates for p_1 and p_2 . Inversely, when the projection matrices are misregistered, the images are much less photo-consistent. A suitable cost function is the following measure of *photo-inconsistency* (abbreviated as phi, hence ϕ):

$$C_\phi(p_1, p_2) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{X} \in \mathcal{P}} \|I_1(P_1\mathbf{X}) - I_2(P_2\mathbf{X})\|^2 \quad (1)$$

Here $|\mathcal{P}|$ is the number of points in \mathcal{P} . Given two colour values, $I_1 = (R_1, G_1, B_1)$ and $I_2 = (R_2, G_2, B_2)$, the elementary inconsistency is calculated as $\|I_1 - I_2\| = w_R|R_1 - R_2| + w_G|G_1 - G_2| + w_B|B_1 - B_2|$. The weights are selected as in [6]: $w_R = 0.299$, $w_G = 0.587$, $w_B = 0.114$.

Finding the minimum of the cost function (1) over p_1 and p_2 yields estimates for the projection matrices.

3 The proposed method

3.1 Selecting the optimisation strategy

Even though the cost function $C_\phi(p_1, p_2)$ is not overly complicated, finding the minimum is a difficult task. One faces nonlinear minimisation in a 22-dimensional parameter space. In addition, the cost function is not smooth. Due to tricky interactions between the parameters, $C_\phi(p_1, p_2)$ has unpredictable shape with possible multiple minima. We had tested a number of standard techniques, including the Levenberg-Marquardt algorithm [3], before we finally decided to apply a global search strategy, namely a *genetic algorithm*.

To reduce the running time to an acceptable level, **manual pre-registration** was done prior to running the selected genetic algorithm. Using the human assistance to initialise the search is reasonable because this operation is simple and fast compared to the 3D scanning, which is also done manually. On the other hand, the manual pre-registration itself cannot provide the precision required: it is only used to start and restrict the registration process.

Figure 2 summarises the proposed two-step method. The manual pre-registration provides the rough estimates P_1^0 and P_2^0 of the projection matrices P_1 and P_2 , respectively. Then P_1^0 and P_2^0 are refined by minimising the photo-consistency based cost function (1). Note that each image is pre-registered to the

3D model *separately*, while the final precise registration involves *simultaneous* registration of both images to the model.

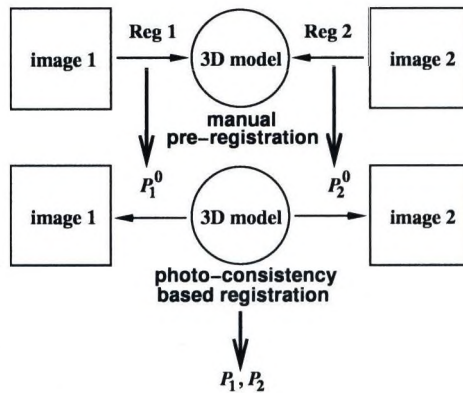


Fig. 2. Block-diagram of proposed method.

3.2 Genetic algorithm and issues of robustness

A selection of genetic algorithms is available [2]. We use the GALib genetic algorithm package [8] written by Matthew Wall at the MIT. The **steady-state algorithm** with the default elitist option yielded the best results. The algorithm applies uniform crossover and Gaussian mutation operator. (The simple genetic algorithm [2] was also tested, but its results were worse.) The optimised parameters are real numbers taking their values from pre-defined intervals. The intervals are set to the pre-registered values plus a margin of $\pm 5\%$.

In registration and correspondence, **robustness** is a critical issue. The cost function (1) is not robust, since some of the inconsistencies may be produced by outliers, typically, by occluded points. Mutually visible points should only be used. In [1], the visibility is checked by ray tracing. This method is the best in the sense that it eliminates occluded points perfectly. However, it is very costly. We have implemented a less accurate but much faster algorithm that uses surface normals. The scalar product of the normal vector and the unit vector pointing towards the camera is a simple measure of visibility. When this product falls below a threshold, the point is discarded. The product is the cosine of the angle between the two vectors. Typically, the threshold was set at 0.5, which discards some mutually visible points, but still leaves enough points for reliable registration.

This method is quite efficient, but it cannot guarantee that all outliers will be filtered out. In particular, false large inconsistencies may still remain due to texture periodicity, or at the object boundaries. To suppress the remaining outliers, the cost function (1) was modified in a robust manner. Two variants of modification were considered: the Trimmed Squares (TS) and the α -trimmed mean [7]. Both techniques have a single parameter, α . In TS, α is the rate of the

largest squares which are discarded. In the α -trimmed mean, both smallest and largest values are rejected: when α is close to 0.5, the median is used. In our pilot experiments, we used $\alpha = 0.2$. The α -trimmed mean performed slightly better. Rejecting the most photo-consistent points appears to decrease the influence of the less informative part of the data.

4 Tests

Careful testing of the method is currently in progress. In this section, we present two pilot results of the ongoing experimental study. The two input datasets were obtained for two very different shapes. Figure 1 shows the first of them, called the Frog dataset. The second one, called the Shell dataset, is shown in figure 5.

The result of our algorithm for the Frog is displayed in figure 3. After computing the projection matrices the texture was back-projected onto the surface. (In figures 3 and 6, the textureless 3D model is on the right, the textured model is on the left.) Although the texture is not distinct, some details still show the accuracy of the registration. Such details are the mouth, the tie and the feet.



Fig. 3. Frog registration.

To check the **precision of the registration**, we put markers on the registered 3D surface and projected them onto the images, as shown in figure 4. The corresponding points are marked by the same colour. One can see that most of the points are located precisely.



Fig. 4. Frog precision test. Center: 3D markers. Sides: projections onto images.

A registration result for the Shell is displayed in figure 6. Note that the stripes of the shell texture are in the proper positions. In this case, the surface texture is distinct and periodic. As the shape is also periodic, precise registration of the texture to the wavy shape is crucial for the photorealism.

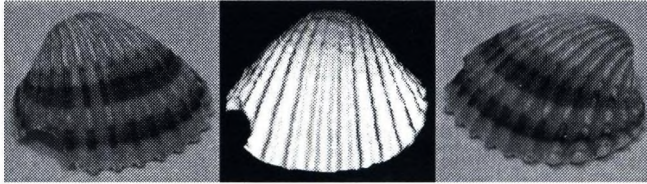


Fig. 5. The Shell dataset.

The genetic optimisation in the 22-dimensional space is time-consuming despite the restrictions due to the manual pre-registration. In our experiments the population of the genetic algorithm consists of 300 individuals and the number of the generations was also 300. The 3D model contained about 1000 points. Under these conditions the program ran for about 10 minutes on a 1.6 GHz PC with 512MB memory.



Fig. 6. Shell registration.

5 Discussion and outlook

We have presented a method for precise registration of an image pair to a 3D model for uncalibrated cameras. The method uses a manual pre-registration followed by a genetic algorithm that minimises a photo-consistency based cost function and finds estimates for the 22 parameters of the two projection matrices.

By nature, the genetic algorithm is not deterministic. Each time we run the algorithm we obtain, strictly speaking, a different result. For high-precision registration, the **reproducibility** of the results is an important question. We have carried out some pilot experiments to study the reproducibility. Figure 7 displays three results obtained by running the registration algorithm with the pre-registration shown in the leftmost block of the figure. The genetic algorithm moves and corrects the initial markers. Some of the final positions slightly differ between the three runs. As discussed below, there may be several good solutions in the multidimensional search space. Still, the solutions are convincingly **consistent in registration**, demonstrating a reasonable degree of reproducibility.

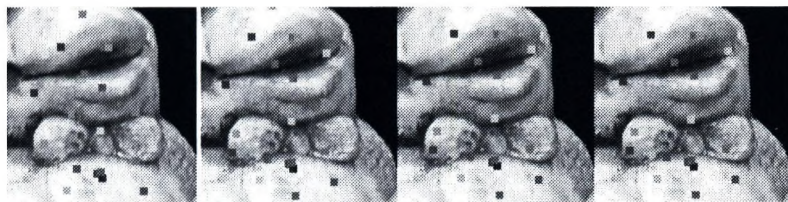


Fig. 7. Reproducibility of genetic registration. Leftmost image is pre-registration.

Our method cannot serve as a camera calibration technique. The goal is not to precisely determine the physical parameters: what we look for are the projection matrices that register the images to the model as photo-consistently as possible. Different solutions may be acceptable and valid. Partially, this results from the inaccuracy of the correspondences: a small error in correspondence can yield large changes in some parameters. The degree to which the parameters may vary is an open question to be studied. Our pilot tests show that the camera orientation parameters are the most stable, with a typical variation of about 0.5%. The focal length and the principal point varied most strongly, up to 3%.

The proposed method is still being developed. Until now, we only worked with image pairs. Further research will show if the approach can and should be extended to more images. Using more images allows one to impose several constraints on the intrinsic parameters. On the other hand, the number of the extrinsic parameters would grow. Thus the advantages of using more images are not obvious. Another open problem is robustness against changes of lighting conditions and shadows.

Acknowledgement. This work was supported by the Hungarian Scientific Research Fund (OTKA) grants T038355 and M28078. The authors thank Anikó Ekárt for useful comments on the genetic algorithm, and Evgeny Lomonosov for providing the measurement data.

References

1. M. J. Clarkson, D. Rueckert, D. L. Hill, and D. J. Hawkes. Using photo-consistency to register 2D optical images of the human face to a 3D surface model. *IEEE Tr. on Patt. Anal. and Machine Intell.*, 23:1266–1280, 2001.
2. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
3. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
4. G. Kós. An algorithm to triangulate surfaces in 3D using unorganised point clouds. *Computing Suppl.*, 14:219–232, 2001.
5. K. Kutulakos and S. Seitz. *A Theory of Shape by Space Carving*. Prentice Hall, 1993.
6. M. Lhuillier. Efficient dense matching for textured scenes using region growing. *The Ninth British Machine Vision Conference*, pages 700–709, 1998.
7. I. Pitas. *Digital Image Processing Algorithms*. Prentice Hall, 1993.
8. M. Wall. The GALib genetic algorithm package. URL: <http://lancet.mit.edu/ga>.

Reversible Jump Markov Chain Monte Carlo for Unsupervised MRF Color Image Segmentation *

Zoltan Kato

University of Szeged, Dept. of Informatics, P.O.Box 652, H-6701 Szeged, Hungary
kato@inf.u-szeged.hu

Abstract. This paper deals with unsupervised image segmentation. We use a first order Markov random field (MRF) model where the singleton energies derive from a multivariate Gaussian distribution and second order potentials favor similar classes in neighboring pixels. The second order potential is fixed, all other parameters are estimated using a reversible jump Markov chain Monte Carlo method. The algorithm finds the most likely number of classes, their associated model parameters and generates a segmentation of the image by classifying the pixels into these classes.

1 Introduction

Image segmentation is an important early vision task where pixels with similar features are grouped into homogeneous regions. In a probabilistic framework, unsupervised segmentation can be treated as an *incomplete data problem* where the color values are observed, the label field is missing and the associated class model parameters need to be estimated. When the number of pixel classes is also being estimated, the unsupervised segmentation problem may be regarded as a model selection problem over a combined model space. Basically, there are two approaches in the literature. One of them is an exhaustive search of the combined parameter space [11, 8]: Segmentation and parameter estimates are obtained via an iterative algorithm by alternately sampling the label field based on the current estimates of the parameters. Then the maximum likelihood estimates of the parameter values are computed using the current labeling. The resulting estimates are then applied to a model fitting criteria to select the optimum number of classes. Another approach consists of a two step approximation technique [6, 9]: the first step is a coarse segmentation of the image into the most likely number of regions. Then the parameter values are estimated from the resulting segmentation and the final result is obtained via a supervised segmentation.

Our approach consists of building a Bayesian color image model using a first order MRF. The observed image is represented by a mixture of multivariate Gaussian distributions while inter-pixel interaction favors similar labels at

* This research was partially supported by an ERCIM postdoctoral fellowship during the author's stay at CWI, Amsterdam, The Netherlands; and the Janos Bolyai Research Fellowship of the Hungarian Academy of Sciences.

neighboring sites. Then a MCMC algorithm is used to sample from the whole posterior distribution in order to obtain a MAP estimate via simulated annealing [3]. However, classical MCMC methods are restricted to problems where the dimensionality of the parameter vector is fixed. Therefore, the estimation of the number of mixture components is not possible. Recently, a novel method, called Reversible Jump MCMC (RJMCMC), has been proposed [4] which makes it possible to construct reversible Markov chain samplers that jump between parameter subspaces of different dimensionality. It has also been applied to univariate Gaussian mixture identification [10], intensity based image segmentation [1], and computing medial axes of 2D shapes [12]. Although we present the model in the case of 3 dimensional observations, it is straightforward to extend it to higher dimensions.

2 Color Image Segmentation Model

Let us suppose that the observed image $\mathcal{F} = \{\mathbf{f}_s | s \in \mathcal{S}, \forall i : 0 < f_s^i < 1\}$ consists of three spectral component values at each pixel s denoted by the vector \mathbf{f}_s . The segmentation is done by assigning a label $\omega_s \in \Lambda = \{1, 2, \dots, L\}$ to each site s . $\omega \in \Omega$ denotes a labeling (or segmentation), Ω is the set of all possible labeling.

Basically, we regard our image as a sample drawn from an unknown Gaussian mixture distribution. The goal of our analysis is inference about the number L of components, the component parameters $\Theta = \{\forall \lambda \in \Lambda \Theta_\lambda = (\boldsymbol{\mu}_\lambda, \boldsymbol{\Sigma}_\lambda)\}$, the components weights p_λ , summing to 1, the clique potential (or inter-pixel interaction strength) β , and the segmentation ω . The joint distribution of the variables $L, p, \beta, \omega, \Theta, \mathcal{F}$ is given by: $P(\omega, \mathcal{F} | \Theta, \beta, p, L)P(\Theta, \beta, p, L)$ In our context, it is natural to impose conditional independences on (Θ, β, p, L) so that their joint probability reduces to the product of priors. Let us concentrate now on the posterior distribution of (\mathcal{F}, ω) which may be expressed as: $P(\mathcal{F} | \omega, \Theta, \beta, p, L)P(\omega | \Theta, \beta, p, L)$. Before further proceeding, let us examine the above factorization. As we declared earlier, pixel classes are represented by a three-variate Gaussian distribution and the underlying MRF label process follows a Gibbs distribution defined over a first order neighborhood system [3]. Thus, we can impose further conditional independences yielding:

$$P(\mathcal{F} | \omega, \Theta, \beta, p, L) = P(\mathcal{F} | \omega, \Theta) =$$

$$\prod_{s \in \mathcal{S}} \frac{1}{\sqrt{(2\pi)^3 |\boldsymbol{\Sigma}_{\omega_s}|}} \exp\left(-\frac{1}{2}(\mathbf{f}_s - \boldsymbol{\mu}_{\omega_s})\boldsymbol{\Sigma}_{\omega_s}^{-1}(\mathbf{f}_s - \boldsymbol{\mu}_{\omega_s})^T\right) \quad (1)$$

$$P(\omega | \Theta, \beta, p, L) = P(\omega | \beta, p, L) = \frac{1}{Z(\beta, p, L)} \exp(-U(\omega | \beta, p, L)), \text{ where } (2)$$

$$U(\omega | \beta, p, L) = \sum_{s \in \mathcal{S}} -\log(p_{\omega_s}) + \beta \sum_{\{s, r\} \in \mathcal{C}} \delta(\omega_s, \omega_r)$$

$U(\omega \mid \beta, p, L)$ is called the *energy function*. $\delta(\omega_s, \omega_r) = 1$ if ω_s and ω_r are different and -1 otherwise. $Z(\beta, p, L) = \sum_{\omega \in \Omega} \exp(-U(\omega \mid \beta, p, L))$ denotes the normalizing constant (or *partition function*). Furthermore, \mathcal{C} denotes the set of cliques and $\{s, r\}$ is a doubleton containing the neighboring pixel sites s and r . Since the partition function $Z(\beta, p, L)$ is not tractable [7], the comparison of the likelihood of two differing MRF realizations from Eq. (2) is unfeasible. However, we can compare their Pseudo-Likelihood [7]. Therefore, using Eq. (1) and the fact that $P(\mathcal{F})$ is constant for a given image, we can now easily approximate the posterior distribution $P(L, p, \beta, \omega, \Theta \mid \mathcal{F})$ by:

$$P(\mathcal{F} \mid \omega, \Theta) \prod_{s \in \mathcal{S}} \frac{p_{\omega_s} \exp \left(-\beta \sum_{\forall r: \{s, r\} \in \mathcal{C}} \delta(\omega_s, \omega_r) \right)}{\sum_{\lambda \in \Lambda} p_{\lambda} \exp \left(-\beta \sum_{\forall r: \{s, r\} \in \mathcal{C}} \delta(\lambda, \omega_r) \right)} \times P(\beta) P(L) \prod_{\lambda \in \Lambda} P(\mu_{\lambda}) P(\Sigma_{\lambda}) P(p_{\lambda}) \quad (3)$$

Concerning the priors, we will follow [10] and chose uniform reference priors for $L, \mu_{\lambda}, \Sigma_{\lambda}, p_{\lambda}$ ($\lambda \in \Lambda$).

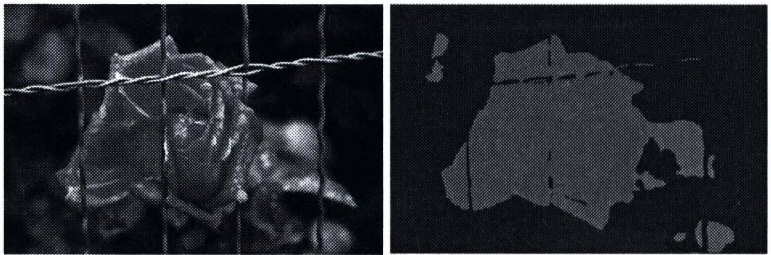


Fig. 1. Segmentation of image *rose41* (3 classes).

3 Sampling the Posterior Distribution

A broadly used tool to sample from the posterior distribution in Eq. (3) is the Metropolis-Hastings method [5]. Classical methods, however, can not be used due to the changing dimensionality of the parameter space. To overcome this limitation, the promising RJMCMC approach has been proposed in [4]. When we have multiple parameter subspaces of different dimensionality, it is necessary to devise different *move types* between the subspaces [4]. These will be combined in a so called *hybrid sampler*. For our image segmentation model, we shall make use

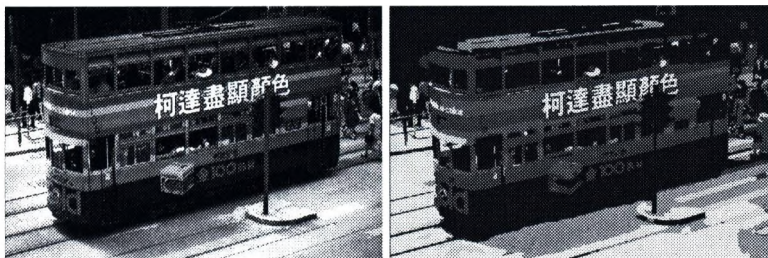


Fig. 2. Segmentation of image *kodakBus93* (10 classes).

of the following move types: (1) sampling the labels ω (ie. re-segment the image); (2) sampling Gaussian parameters $\Theta = \{(\mu_\lambda, \Sigma_\lambda)\}$; (3) sampling the mixture weights $p_\lambda (\lambda \in \Lambda)$; (4) sampling the MRF hyperparameter β ; (5) sampling the number of classes L (splitting one mixture component into two, or combining two into one). We note that the only randomness in scanning these move types is the random choice between splitting and merging in move (5). One iteration of the hybrid sampler, also called a *sweep*, consists of a complete pass over these moves. The first four move types are conventional in the sense that they do not alter the dimension of the parameter space. In each of these move types, the posterior distribution can be easily derived from Eq. (3) by setting unaffected parameters to their current estimate.

Hereafter, we will focus on move (5), which requires the use of the reversible jump mechanism. This move type involves changing L by 1 and making necessary corresponding changes to ω , Θ and p . We remark that at this move type, basically the whole posterior distribution in Eq. (3) is sampled. Only β can be set to its estimate $\hat{\beta}$.

3.1 Splitting One Class into Two

The split proposal begins by choosing a class λ *at random* with a uniform probability $P_{select}^{split}(\lambda) = 1/L$. Then L is increased by 1 and λ is split into λ_1 and λ_2 . In doing so, a new set of parameters need to be generated. Altering L changes the dimensionality of the variables Θ and p . Thus we shall define a deterministic function ψ as a function of these Gaussian mixture parameters: $(\Theta^+, p^+) = \psi(\Theta, p, u)$ where the superscript $+$ denotes parameter vectors after increasing L . u is a set of random variables having as many elements as the degree of freedom of joint variation of the current parameters (Θ, p) and the proposal (Θ^+, p^+) . Note that this definition satisfies the *dimension matching* constraint [4], which guarantees that one can jump back and forth between different parameter sub-spaces. The new parameters of λ_1 and λ_2 are assigned by matching the 0^{th} , 1^{th} , 2^{th} moments of the component being split to those of a combination of the two new components (eg. $p_\lambda = p_{\lambda_1}^+ + p_{\lambda_2}^+$, etc. . .) [10]: There are 10 degrees of freedom in splitting λ since covariance matrices are symmetric. Therefore, we need to generate a random variable u_1 , a random vector u_2 and a symmetric random

matrix $\mathbf{u3}$. We can now compute the new parameter values as follows:

$$p_{\lambda_1}^+ = p_{\lambda} u1, \quad p_{\lambda_2}^+ = p_{\lambda} (1 - u1) \quad (4)$$

$$\mu_{\lambda_1, i}^+ = \mu_{\lambda, i} + u2_i \sqrt{\Sigma_{\lambda, i, i} \frac{1 - u1}{u1}} \quad (5)$$

$$\mu_{\lambda_2, i}^+ = \mu_{\lambda, i} - u2_i \sqrt{\Sigma_{\lambda, i, i} \frac{u1}{1 - u1}} \quad (6)$$

$$\Sigma_{\lambda_1, i, j}^+ = \begin{cases} u3_{i, i} (1 - u2_i^2) \Sigma_{\lambda, i, i} \frac{1}{u1} & \text{if } i = j \\ \frac{u3_{i, j} \Sigma_{\lambda, i, j} \sqrt{(1 - u2_i^2)}}{\times \sqrt{(1 - u2_j^2)} u3_{i, i} u3_{j, j}} & \text{if } i \neq j \end{cases} \quad (7)$$

$$\Sigma_{\lambda_2, i, j}^+ = \begin{cases} (1 - u3_{i, i}) (1 - u2_i^2) \Sigma_{\lambda, i, i} \frac{1}{u1} & \text{if } i = j \\ \frac{(1 - u3_{i, j}) \Sigma_{\lambda, i, j}}{\times \sqrt{(1 - u2_i^2)} (1 - u2_j^2)} \\ \times \sqrt{(1 - u3_{i, i}) (1 - u3_{j, j})} & \text{if } i \neq j \end{cases} \quad (8)$$

The random variables u are chosen from the interval $(0, 1]$. In order to favor splitting the class into roughly equal portions, beta(1.1, 1.1) distributions are used.

The next step is the reallocation of those sites $s \in \mathcal{S}_{\lambda}$ where $\omega_s = \lambda$. This reallocation is based on the new parameters and has to be completed in such a way as to ensure the resulting labeling ω^+ is drawn from the posterior distribution with $\Theta = \Theta^+$, $p = p^+$ and $L = L + 1$. At the moment of splitting, the neighborhood configuration at a given site $s \in \mathcal{S}_{\lambda}$ is unknown thus the calculation of the term $P(\omega^+ | \hat{\beta}, p^+, L + 1)$ is not possible. First, we have to provide a tentative labeling of the sites in \mathcal{S}_{λ} . Then we can sample the posterior distribution using a Gibbs sampler. Of course, a tentative labeling might be obtained by allocating λ_1 and λ_2 at random. In practice, however, we need a labeling ω^+ which has a relatively high posterior probability in order to maintain a reasonable acceptance probability. To achieve this goal, we use a few step (around 5 iterations) of ICM [2] algorithm to obtain a suboptimal segmentation of \mathcal{S}_{λ} . The resulting label map can be used to draw a sample from the posterior distribution using a one step Gibbs sampler [3]. The obtained ω^+ has a relatively high posterior probability since the tentative labeling was close to the optimal labeling.

3.2 Merging Two Classes

A pair (λ_1, λ_2) is chosen with a probability inversely proportional to their Mahalanobis distance. In this way, we favor merging classes that are close to each other. The merge proposal is deterministic, once the choices of λ_1 and λ_2 have been made. These two components are merged, reducing L by 1. As in the case of splitting, altering L changes the dimensionality of the variables Θ and p . The

new parameter values (Θ^-, p^-) are obtained from the moment equations. The reallocation is simply done by setting the label at sites $s \in \mathcal{S}_{\{\lambda_1, \lambda_2\}}$ to the new label λ . The random variables u are obtained by back-substitution into Eq. (4)–(8).

3.3 Acceptance Probability

In the Metropolis-Hastings algorithm [5], the split or merge proposal is accepted with a probability relative to the probability ratio of the current and the proposed states. Let us first consider the acceptance probability \mathcal{A}_{split} for the split move. For the corresponding merge move, the acceptance probability is obtained as the inverse of the same expression with some obvious differences in the substitutions. $\mathcal{A}_{split}(L, \hat{p}, \hat{\beta}, \hat{\omega}, \hat{\Theta}; L+1, p^+, \hat{\beta}, \omega^+, \Theta^+) = \min(1, A)$, where A is given by:

$$\frac{P(L+1, p^+, \hat{\beta}, \omega^+, \Theta^+ | \mathcal{F}) P_{merge}(L+1) P_{select}^{merge}(\lambda_1, \lambda_2)}{P(L, \hat{p}, \hat{\beta}, \hat{\omega}, \hat{\Theta} | \mathcal{F}) P_{split}(L) P_{select}^{split}(\lambda) P_{realloc}} \times \frac{1}{P(u1) \prod_{i=1}^3 \left(P(u2_i) \prod_{j=i}^3 P(u3_{i,j}) \right)} \left| \frac{\partial \psi}{\partial (\Theta_\lambda, p_\lambda, u)} \right|$$

$P_{realloc}$ denotes the probability of reallocating pixels labeled by λ into regions labeled by λ_1 and λ_2 . $P_{realloc}$ can be derived by restricting the set of labels Λ^+ to the subset $\{\lambda_1, \lambda_2\}$ and taking into account only those sites s for which $\omega_s^+ \in \{\lambda_1, \lambda_2\}$. The last factor is the Jacobian determinant of the transformation ψ .

4 Optimization According to the MAP

The MAP estimator $(\hat{\omega}, \hat{L}, \hat{p}, \hat{\beta}, \hat{\Theta}) = \arg \max_{L, p, \beta, \omega, \Theta} P(L, p, \beta, \omega, \Theta | \mathcal{F})$ is used to obtain a segmentation and model parameters. The optimization problem can be solved via simulated annealing [3]:

Algorithm 1 (RJMCMC Segmentation)

- ① Set $k = 0$, $T_0 = 6.0$, and initialize $\hat{\beta}^0, \hat{L}^0, \hat{p}^0, \hat{\Theta}^0$.
- ② A sample $(\hat{\omega}^k, \hat{L}^k, \hat{p}^k, \hat{\beta}^k, \hat{\Theta}^k)$ is drawn from the posterior distribution using the hybrid sampler. Each sub-chain is sampled via the corresponding move-type while all the other parameter values are set to their current estimate.
- ③ Goto Step ② with $k = k + 1$ and $T_{k+1} = 0.98T_k$ until $k < \mathcal{K}$.

In our experiments, the algorithm was stopped after 200 iterations ($T_{200} \approx 0.1$). We note that small temperatures may cause an overflow in the computation of the posterior probability even when using double precision numbers and working with logarithms.

Although our discussion includes the estimation of the MRF hyperparameter β , we haven't estimated it in our experiments. Since the likelihood is approximated by the pseudo-likelihood, the posterior density of β may not be proper under particular label configurations. This problem has been reported in [1] and our findings were similar. Fortunately, the mixture weights p_λ are able to maintain a balance between the external and internal field strength, which makes it possible to set β a priori.

5 Experimental Results

The proposed algorithm has been tested on a variety of real color images. Herein, we present a few examples of these results. First, the original images were converted from RGB to LHS color space in which chroma and intensity informations are separated. The dynamic range of all color components was normalized such that they take their values from $(0, 1)$. Independently of the input image, we start the algorithm with two classes, each of them having equal weights ($\hat{p}_0^0 = \hat{p}_1^0 = 0.5$). The mean vectors were set to $[0.2, 0.2, 0.2]$ and $[0.7, 0.7, 0.7]$. The diagonal elements of the covariance matrices were set to 0.05, while other elements were set to 0.00001. The hyperparameter $\beta = 2.5$ was fixed a priori. Furthermore, the number of classes L was restricted to the interval $[1, 50]$.

On Fig. 1, 3 classes have been identified and an accurate segmentation is obtained. A more difficult scene is segmented in Fig. 2, where 10 classes have been detected. The test images can be found at the Kodak Digital Image Offering WWW site:

<http://www.kodak.com/digitalImaging/samples/imageIntro.shtml>.

Overall, the algorithm performed well but the sequential implementation is quite CPU intensive (45 minutes for a 256×256 image with 5 classes). Fortunately, the Markov property of our segmentation model and the construction of the hybrid sampler allows massive parallel computation of conditionally independent pixel labels and simultaneous sampling of different sub-chains.

6 Conclusion

In this paper, we have proposed an unsupervised color image segmentation algorithm. We have established a Bayesian segmentation model using MRF modeling of the underlying label field. Pixel classes are represented by multivariate Gaussian distributions. The number of classes, class model parameters, and pixel labels are all directly sampled from the posterior distribution using an RJMCMC sampler. Experimental results show that an accurate segmentation can be obtained on a variety of real images.

References

1. S. A. Barker and P. J. W. Rayner. Unsupervised image segmentation using Markov random field models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 165–178. Springer, 1997.

2. J. Besag. On the statistical analysis of dirty pictures. *J. R. Statist. Soc., ser. B*, 48(3):259–302, 1986.
3. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
4. P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–731, 1995.
5. W. K. Hastings. Monte Carlo sampling methods using Markov chains and their application. *Biometrika*, 57:97–109, 1970.
6. C. L. Huang, T. Y. Cheng, and C. C. Chen. Color images segmentation using scale space filter and Markov random field. *Pattern Recognition*, 25(10):1217–1229, 1992.
7. S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(8):799–813, Aug. 1989.
8. D. A. Langan, J. W. Modestino, and J. Zhang. Cluster validation for unsupervised stochastic model-based image segmentation. *IEEE Trans. on Image Processing*, 7(2):180–195, Feb. 1998.
9. J. Liu and Y. H. Yang. Multiresolution Color Image Segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(7):689–700, July 1994.
10. S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *J. R. Statist. Soc., ser. B*, 59(4):731–792, 1997.
11. C. S. Won and H. Derin. Unsupervised segmentation of noisy and textured images using Markov random fields. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 54(4):308–328, July 1992.
12. S. C. Zhu. Stochastic jump-diffusion process for computing medial axes in Markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(11):1158–1169, Nov. 1999.

Color Textured Image Segmentation Using a Multi-Layer Markovian Model*

Zoltan Kato¹, Ting-Chuen Pong², and Song Guo Qiang³

¹ University of Szeged, Dept. of Informatics, P.O.Box 652, H-6701 Szeged, Hungary
`kato@inf.u-szeged.hu`

² Hong Kong University of Science and Technology, Computer Science Dept., Clear Water Bay, Kowloon, Hong Kong
`tcpong@cs.ust.hk`

³ National University of Singapore, School of Computing, 3 Science Drive 2, Singapore 117543
`songguoq@comp.nus.edu.sg`

Abstract. Herein, we propose a new Markov random field (MRF) image segmentation model which aims at combining color and texture features. The model has a multi-layer structure: Each feature has its own layer, called *feature layer*, where an MRF model is defined using only the corresponding feature. A special layer is assigned to the combined MRF model. This layer interacts with each feature layer and provides the segmentation based on the combination of different features. The uniqueness of our algorithm is that it provides both color only and texture only segmentations as well as a segmentation based on combined color *and* texture features. The number of classes on feature layers is given by the user but it is estimated on the combined layer.

1 Introduction

Image segmentation is an important early vision task where pixels with similar features are grouped into homogeneous regions. There are many features that one can take into account during the segmentation process: gray-level, color, motion, texture features, etc. However, most of the segmentation algorithms presented in the literature are based on only one of the above features. Recently, the segmentation of color images received more attention [2, 5, 11, 10, 12, 9]. In this paper (see also [8]), we are interested in the segmentation of color textured images. Basically, there are two approaches to this problem: One approach deals directly with *color textures* [11, 10]. In [11], an unsupervised segmentation algorithm is proposed which uses Gaussian MRF models for *color textures*. These models are defined in each color plane with interactions between different color planes. The segmentation algorithm is based on agglomerative hierarchical clustering.

* partially supported by grants HKUST6072/97E - Hong Kong; NUS R-252-000-085-112 - Singapore; and the Janos Bolyai Research Fellowship of the Hungarian Academy of Sciences.

A different approach is presented in [10] which uses a multiband smoothing algorithm to generate a multiscale representation of an image. The smoothing is based on human psychophysical measurements of color appearance. First the coarsest level is clustered to isolate core clusters. Other pixels are then reassigned to these core clusters using a probabilistic assignment. Another frequently used approach tries to combine traditional gray level texture features together with pure color features [12, 9]. Our approach falls into this category.

The novelty of our model can be summarized as follows: First, we use *different* features at different layers. This allows us to work with different models or to have varying number of regions at different layers, choosing the one which describes the best our feature data at a given layer. In addition, we have a special layer, called *combined layer*, which does not correspond to any feature but provides a way to combine different features. Second, the layers are fully connected: each pixel interacts with the corresponding pixel at other layers. A similar, fully connected pyramidal model is used in [7]. Multiscale pyramids have also been successfully applied for image segmentation [4]. In these models, each layer usually contains the *same* image data at different resolutions. However, we use *different* data at different layers and we do not perform subsampling, therefore our model is not a pyramid. Each layer is of the same size. In this respect, our model is similar to [11, 10].

2 Multi-Layer Segmentation Model

We use perceptually uniform CIE-L*u*v* color values and texture features derived from the Gabor filtered gray-level image. Segmentation requires simultaneous measurements in both spatial and frequency domain. However, spatial localization of boundaries requires larger bandwidths whereas smaller bandwidths give better texture measurements. Fortunately, Gabor filters have optimal joint localization in both domains [6]. In addition, when we are combining texture features with color, the spatial resolution is considerably increased.

Our model consists of 3 layers. At each layer, we use a first order neighborhood system and extra inter-layer cliques (Fig. 1). The image features are represented by multi-variate Gaussian distributions. Herein, we do not address parameter estimation but we note that the task can be solved using an adaptive segmentation technique [13].

Let us denote the color layer by \mathcal{S}^c , the texture layer by \mathcal{S}^t and the combined layer by \mathcal{S}^x . All layers are of the same size. Our MRF model is defined over the lattice $\mathcal{S} = \mathcal{S}^c \cup \mathcal{S}^x \cup \mathcal{S}^t$. For each site s , the region-type (or class) that the site belongs to is specified by a class label, ω_s , which is modeled as a discrete random variable taking values in $\Lambda = \{1, 2, \dots, L\}$. The set of these labels $\omega = \{\omega_s, s \in \mathcal{S}\}$ is a random field, called the *label process*. Furthermore, the observed image features (color and texture) are supposed to be a realization $\mathcal{F} = \{f_s | s \in \mathcal{S}^c \cup \mathcal{S}^t\}$ from another random field, which is a function of the label process ω . Basically, the *image process* \mathcal{F} represents the deviation from the underlying label process. Thus, the overall segmentation model is com-

posed of the hidden label process ω and the observable noisy image process \mathcal{F} . Our goal is to find an optimal labeling $\hat{\omega}$ which maximizes the a posteriori probability $P(\omega | \mathcal{F})$, that is the *maximum a posteriori* (MAP) estimate [3]: $\arg \max_{\omega \in \Omega} P(\omega | \mathcal{F}) = \arg \max_{\omega \in \Omega} \prod_{s \in \mathcal{S}} P(\mathbf{f}_s | \omega_s) P(\omega)$, where Ω denotes the set of all possible labelings. We use the ICM algorithm [1] to obtain a suboptimal MAP estimate. According to the *Hammersley-Clifford theorem* [3], $P(\omega | \mathcal{F})$ follows a Gibbs distribution:

$$P(\omega | \mathcal{F}) = \frac{\exp(-U(\omega))}{Z(\beta)} = \frac{\prod_{C \in \mathcal{C}} \exp(-V_C(\omega_C))}{Z(\beta)} \tag{1}$$

where $U(\omega)$ is called an *energy function*, $Z(\beta) = \sum_{\omega \in \Omega} \exp(-U(\omega))$ is the normalizing constant and V_C denotes the *clique potential* of clique $C \in \mathcal{C}$ having the label configuration ω_C . Note that the energies of *singletons* (ie. cliques of single sites $s \in \mathcal{S}$) directly reflect the probabilistic modeling of labels without context, while higher order clique potentials express relationship between neighboring pixel labels. In the remaining part of this section, we will define these clique potentials for each layer.

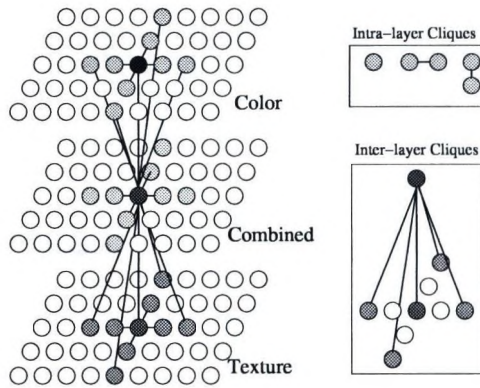


Fig. 1. Multi-layer MRF model.

2.1 Color Layer

On the color layer, the observed image $\mathcal{F}^c = \{\mathbf{f}_s^c | s \in \mathcal{S}^c\}$ consists of three spectral component values ($L^* \mathbf{u}^* \mathbf{v}^*$) at each pixel s denoted by the vector \mathbf{f}_s^c . We assume that $P(\mathbf{f}_s^c | \omega_s)$ follows a Gaussian distribution, the classes $\lambda \in \Lambda^c = \{1, 2, \dots, L^c\}$ are represented by the mean vectors $\boldsymbol{\mu}^c_\lambda$ and the covariance matrices $\boldsymbol{\Sigma}^c_\lambda$. The class label assigned to a site s on the color layer is denoted by ψ_s . The energy function $U(\psi, \mathcal{F}^c)$ of the so defined MRF layer has the following form:

$$\sum_{s \in \mathcal{S}^c} \mathcal{G}^c(\mathbf{f}_s^c, \psi_s) + \beta \sum_{\{s,r\} \in \mathcal{C}} \delta(\psi_s, \psi_r) + \sum_{s \in \mathcal{S}^c} V^c(\psi_s, \eta_s^c)$$

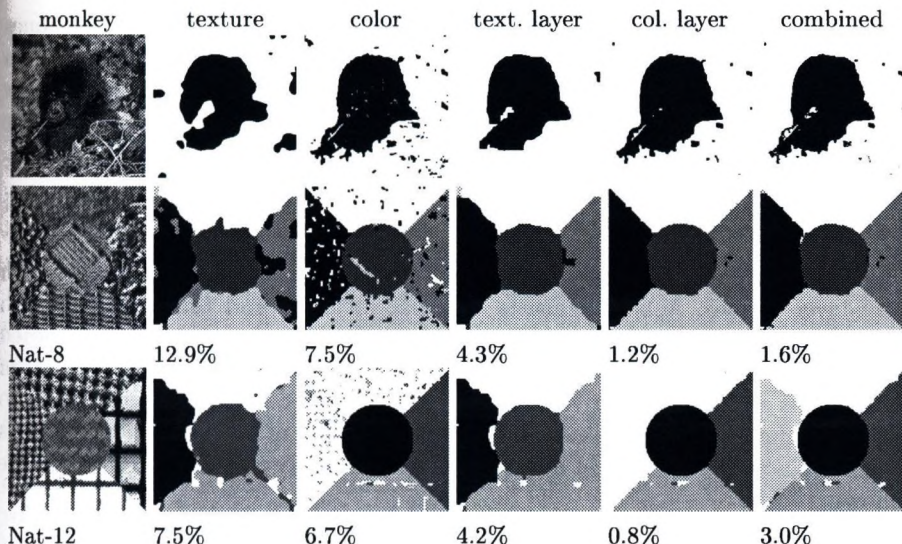


Fig. 2. Results and misclassification rate of color only, texture only, and combined models

where $\mathcal{G}^c(\mathbf{f}_s^c, \psi_s)$ denotes the following log Gaussian:

$$\ln(\sqrt{(2\pi)^3 |\Sigma^c_{\psi_s}|}) + \frac{1}{2}(\mathbf{f}_s^c - \boldsymbol{\mu}^c_{\psi_s})\Sigma^c_{\psi_s}{}^{-1}(\mathbf{f}_s^c - \boldsymbol{\mu}^c_{\psi_s})^T \quad (2)$$

$\delta(\psi_s, \psi_r) = 1$ if ψ_s and ψ_r are different and -1 otherwise. $\beta > 0$ is a parameter controlling the homogeneity of the regions. As β increases, the resulting regions become more homogeneous. The last term ($V^c(\psi_s, \eta_s^c)$) is the inter-layer clique potential which will be defined later.

2.2 Texture Layer

On the texture layer, the observation consists of a set of Gabor image features. To obtain these features, we use the multi-channel filtering approach [6]: The channels are represented by a bank of real-valued, even-symmetric Gabor filters. In our tests, we used four orientations: $0^\circ, 45^\circ, 90^\circ, 135^\circ$ and the radial frequencies u_0 were 1 octave apart: $\sqrt{2}, \sqrt{2}/2, \sqrt{2}/4, \sqrt{2}/8, \dots$ We automatically select 6 of these filtered images such that they cover more than 99% of the image's intensity variations (see [6] for more details). From each filtered image g , we compute a feature image using the nonlinear transformation $|\tanh(\alpha g_s)|$, $s \in \mathcal{S}^t$ followed by a Gaussian blurring with a deviation proportional to the center frequency of the Gabor filter: $\sigma = k/u_0$. In our experiments, the Gabor filtered images are scaled to the interval $[-1, 1]$ and we set $\alpha = 40$ and $k = 1$.

The MRF model itself is quite similar to the one outlined in the previous section. The only difference is that the observation consists of 6 dimensional

texture feature vectors $\mathcal{F}^t = \{\mathbf{f}_s^t | s \in \mathcal{S}^t\}$. The energy of higher order cliques is $\xi \sum_{\{s,r\} \in \mathcal{C}} \delta(\phi_s, \phi_r) + \sum_{s \in \mathcal{S}^t} V^t(\phi_s, \eta_s^t)$, where ξ has the same role as β in the color layer. Furthermore, ϕ_s denotes the label assigned to a site s . As in the previous case, classes are represented by the mean vectors $\boldsymbol{\mu}_{\lambda}^t$ and the covariance matrices $\boldsymbol{\Sigma}_{\lambda}^t$. The inter-layer clique potential ($V^t(\phi_s, \eta_s^t)$) will be defined later.

2.3 Combined Layer

The combined layer only uses the texture and color features indirectly, through inter-layer cliques. A label consists of a pair of color and texture labels such that $\eta = \langle \eta^c, \eta^t \rangle$, where $\eta^c \in \mathcal{A}^c$ and $\eta^t \in \mathcal{A}^t$. The set of labels is denoted by $\mathcal{A}^x = \mathcal{A}^c \times \mathcal{A}^t$ and the number of classes $L^x = L^c L^t$. Obviously, not all of these labels are valid for a given image. Therefore the combined layer model also estimates the number of classes and chose those pairs of texture and color labels which are actually present in a given image. The energy function $U(\eta)$ is of the following form:

$$\sum_{s \in \mathcal{S}^x} (V_s(\eta_s) + V^c(\psi_s, \eta_s^c) + V^t(\phi_s, \eta_s^t)) + \alpha \sum_{\{s,r\} \in \mathcal{C}} \delta(\eta_s, \eta_r)$$

where $V_s(\eta_s)$ denotes singleton energies, $V^c(\psi_s, \eta_s^c)$ (resp. $V^t(\phi_s, \eta_s^t)$) denotes inter-layer clique potentials. The last term corresponds to second order intra-layer cliques which ensures homogeneity of the combined layer. α has the same role as β in the color layer model and $\delta(\eta_s, \eta_r) = -1$ if $\eta_s = \eta_r$, 0 if $\eta_s \neq \eta_r$ and 1 if $\eta_s^c = \eta_r^c$ and $\eta_s^t \neq \eta_r^t$ or $\eta_s^c \neq \eta_r^c$ and $\eta_s^t = \eta_r^t$. The idea is that region boundaries present at both color and texture layers are preferred over edges that are found only at one of the feature layers. Inter-layer interactions are as follows:

$$V^c(\psi_s, \eta_s^c) = \gamma^c \sum_{\{s,r\} \in \mathcal{C}_5} W_r D^c(\psi_r, \eta_r^c)$$

where $D^c(\psi_r, \eta_r^c) = | \mathcal{G}^c(\mathbf{f}_r^c, \psi_r) - \mathcal{G}^c(\mathbf{f}_s^c, \eta_s^c) |$ (see Eq. (2)). $V^t(\phi_s, \eta_s^t)$ and $D^t(\phi_r, \eta_r^t)$ are defined in a similar way using texture features and γ^t as weight. At any site s , we have 5 inter-layer interactions (denoted by \mathcal{C}_5) between two layers: Site s interacts with the corresponding site on the other layer as well as with the 4 neighboring sites two steps away (see Fig. 1). W_r is the weight of the clique $\{s, r\} \in \mathcal{C}_5$. We assign higher weight (0.6) to the corresponding site whereas smaller weights (0.1 each) to the other 4 neighboring sites. The latter 4 sites help to ensure homogeneity on the combined layer. Note that D^c and D^t corresponds to the difference of the first order potentials at the corresponding feature layer. Clearly, the difference is 0 if and only if both the feature layer and the combined layer has the same label. If the labels are different then it is proportional to the energy difference between the two labels. γ^c (resp. γ^t) controls the influence of the inter-layer cliques. A higher value will increase the importance of the information coming from the other layer.

The singleton energy is defined as follows:

$$V_s(\eta_s) = R((10N_{\eta_s})^{-3} + \mathcal{P}(L))$$

$$+\rho^c \sum_{\{s,r\} \in C_5} W_r D^c(\psi_r, \eta_s^c) + \rho^t \sum_{\{s,r\} \in C_5} W_r D^t(\phi_r, \eta_s^t)$$

The importance of the above potential function is twofold: First of all, it controls the number of classes at the combined layer. $(10N_{\eta_s})^{-3}$ penalizes small classes (N_{η_s} is the percentage of the sites assigned to class η_s), while $\mathcal{P}(L)$ includes some prior knowledge about the number of classes. Currently this is expressed by a log Gaussian term (similar to the one in Eq. (2)) with mean value \hat{L} (basically an initial guess) and variance σ (confidence in the initial guess). R is simply a weight of this term, we set it to 0.5 in our tests. Second, it balances the influence of the feature layers to the combined layer. This is achieved by including D^c and D^t in the potential function. ρ^t (resp. ρ^c) controls this balancing: When the clique potentials are summed for a given site *on the combined layer*, the final weight of D^t equals to $\rho^t + \gamma^t$. However, the weight of D^t at a site *on the texture layer* is always γ^t . Therefore, depending on the value of ρ^t (resp. ρ^c), we can increase or decrease the influence of a feature layer to the combined layer without changing the influence of the combined layer to a feature layer. We found this an important issue in the case of the texture layer.

3 Experiments

The proposed algorithm has been tested on a variety of synthetic and real images. The computing time was 2-4 minutes on a Pentium III 933. We also compare the results to texture only and color only segmentation (basically a monogrid model similar to the one defined for the feature layers but without inter-layer cliques). The mean vectors and covariance matrices were computed over representative regions selected by the user. The number of texture and color classes is known a priori but classes on the combined layer are estimated during the segmentation process. Hyper-parameters have been trained on a small subset of images: $\alpha = 1.0$, $\beta = \xi = 10.0$, $\gamma^c = \gamma^t = 1.5$, $\rho^c = 0.5$, and $\rho^t = -0.3$. These values have been found to provide satisfactory results on *all* test images. The value of β and ξ is not crucial, basically any value between 2 and 15 provides good segmentations. γ and ρ values need slightly higher accuracy. Note that by setting $\rho^t < 0$ and $\rho^c > 0$, we decrease the influence of the texture layer and increase the influence of the color layer on the combined layer. This is necessary because texture features (due to filtering and blurring) have weaker spatial localization. Hence, we give a higher weight to the color layer so that edges will be localized correctly while region homogeneity (where color layer is slightly weaker, especially in textured regions) is still maintained. Fig. 2 shows some segmentation results together with the measured misclassification rate. Clearly, the multi-layer model provides significantly better results compared to color only and texture only segmentations. *Nat-12* shows an image with 4 different textures and 4 different colors. We can see, that our method provides accurate segmentations on both feature layers and it is also able to detect 5 classes on the combined layer. Note that the combined layer produces slightly higher misclassification rates ($\approx 0.5\%$) than the color layer. This is due to sharper boundaries on the color

layer (texture has weaker spatial resolution and the combined layer is directly influenced by the texture layer). We have also compared our results to those reported in [10] and found them equally good. One example is the *monkey* image but more results are available on our website (*removed for blind review*).

4 Conclusion

We have proposed a new multi-layer MRF segmentation model which successfully combines color and texture features. However, the model is not restricted to these features. It can be applied to multicue segmentation in general.

References

1. J. Besag. On the statistical analysis of dirty pictures. *J. Roy. Statist. Soc., ser. B*, 1986.
2. D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–755, San Juan, Puerto Rico, June 1997.
3. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
4. F. Heitz, P. Perez, and P. Bouthemy. Multiscale Minimization of Global Energy Functions in Some Visual Recovery Problems. *CVGIP:IU*, 59(1):125–134, 1994.
5. C. L. Huang, T. Y. Cheng, and C. C. Chen. Color images segmentation using scale space filter and Markov random field. *Pattern Recognition*, 25(10):1217–1229, 1992.
6. A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
7. Z. Kato, M. Berthod, and J. Zerubia. A hierarchical Markov random field model and multi-temperature annealing for parallel image classification. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 58(1):18–37, Jan. 1996.
8. Z. Kato, T. C. Pong, and S. G. Qiang. Multicue mrf image segmentation: Combining texture and color. In *Proceedings of the International Conference on Pattern Recognition*, Quebec, Canada, Aug. 2002. IEEE-IAPR.
9. W. Y. Ma and B. S. Manjunath. Edge flow: A framework of boundary detection and image segmentation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 744–749, San Juan, Puerto Rico, June 1997.
10. M. Mirmehdi and M. Petrou. Segmentation of color textures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(2):142–159, Feb. 2000.
11. D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of textured color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(10):939–954, Oct. 1995.
12. S. C. Tan and J. Kittler. Colour texture classification using features from color histogram. In *Proc. 8th Scandinavian Conference on Image Processing*, 1993.
13. C. S. Won and H. Derin. Unsupervised segmnetation of noisy and textured images using Markov random fields. *Computer Graphics and Image Processing: Graphical Models and Image Processing*, 54(4):208–328, July 1992.

Előfeldolgozás és diszkrét tomográfiai rekonstrukció a neutron radiográfiában

Kiss Zoltán¹, Ruskó László¹, Rodek Lajos¹, Kuba Attila¹, Balaskó Márton²

Szegedi Tudományegyetem, Képfeldolgozás és Számítógépes Grafika Tanszék

¹{kissz, rusko, rodek, kuba}@inf.u-szeged.hu

6720 Szeged, Árpád tér 2.

+36 (62) 544-000/6399

KFKI, AEKI

²balasko@sunserv.kfki.hu

Budapest, 1114, PO49

+36 (1) 395-9159

Kivonat A neutron tomográfia mára már széles körben alkalmazott technika nemroncsoló anyagvizsgálatra. A módszer fizikai természetéből adódóan a vetületek jelentős zajjal terheltek, ami azonban előfeldolgozással mérsékelhető. Mivel a vetületek előállítása időigényes és költséges művelet, a cél az, hogy a rekonstrukció minél kevesebb vetületből történjen. Kevés vetületből azonban nem rekonstruálható minden függvényosztály elfogadható eredménnyel. A diszkrét tomográfia (DT) alkalmas eszköznek tűnik a probléma megoldására azon feltételezés mellett, hogy az átvilágított tárgy csak néhány anyagfajtát tartalmaz.

Előadásunkban valós vetületi adatokon végzett előfeldolgozási lépéseket mutatunk be (pl. zajhatást csökkentő, háttérkorrigáló, uniformizáló). Továbbá megadunk egy olyan diszkrét tomográfiai rekonstrukciós módszert, amely a problémát kombinatorikai optimalizálási feladatként tekintve oldja meg. Ennek hatékonyságát fantomokon végzett kísérletekkel és a rendelkezésre álló valós adatokon is demonstráljuk.

1. Bevezetés

A KFKI Atomenergetikai Kutató Intézetével (AEKI) együttműködve kezdtük meg kísérletsorozatunkat, melyben az AEKI kísérleti kutató atomreaktora által előállított neutron sugárzással leképezett tárgyak vetületi képeiből igyekszünk minél jobb minőségű keresztmetszeti képeket rekonstruálni.

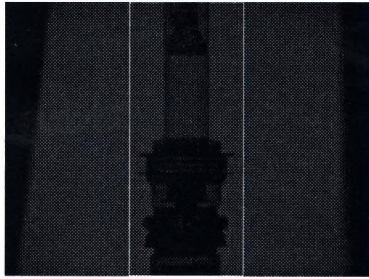
Neutron sugárral átvilágítva egy tárgyat hasonló képet kapunk, mint röntgen sugárzást használva, a lényegi különbség az, hogy a neutron sugárzást bizonyos fémek szinte akadály nélkül átengedik, míg például a víz a sugárzás nagy részét elnyelik. Ez a fizikai tulajdonság jól kihasználható nemroncsoló ipari anyagvizsgálatok végzésére.

A leképező rendszer és sugárforrás sajátosságai miatt a tomográfiai rekonstrukciók során több probléma is felmerült: a vetületi képek intenzitás értékei

Ezt a kutatást az OTKA T/12 032241 támogatta.

alacsonyak voltak, a kép szinte fekete volt; a sorozatbeli képeken megfigyelhető az összintenzitás indokolatlanul nagymértékű ingadozása; a vetületek nem voltak uniformak és jelentős zajjal voltak terhelve. Ezek csökkentésére ill. megszüntetésére alkalmazott előfeldolgozási lépéseket ismertetjük a 2. fejezetben.

Mivel egy neutron sugárzással készített vetületi kép elkészítése igen időigényes és költséges művelet, ezért törekedni kell arra, hogy a rekonstrukciót minél kevesebb vetületből végezzük el. A 3. fejezetben bemutatásra kerül egy diszkrét tomográfiai rekonstrukciós módszer, amely kevés (8-30) vetületből rekonstruál keresztmetszeti képeket.

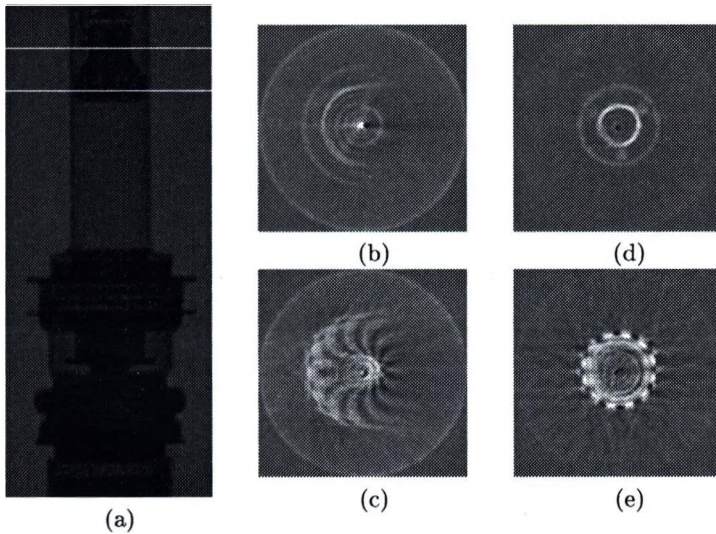


1. ábra. A vizsgált tárgy 0°-os vetülete és a tárgyat tartalmazó részlet vágása

2. Előfeldolgozási lépések

Az előfeldolgozás lépéseit és a rekonstrukcióra gyakorolt jótékony hatásait egy (neutron sugárzással átvilágított) vidicon csőről készült vetületi sorozaton (180 vetület, 1°-os lépésként) fogjuk szemléltetni.

1. Az első lépés a képek intenzitásának növelése volt. (Konstanssal való szorzás.)
2. A tárgyat tartalmazó releváns terület kivágása. (Ld. 1. ábra.)
3. A vetületi képek szorzása úgy, hogy az összintenzitás a képsorozaton konstans legyen.
4. A sorozathoz rendelkezésünkre állt egy objektum nélküli (referencia) felvétel is, amelyen látható volt, hogy felvételt készítő berendezés által készített kép nem homogén, így a sorozat minden képén homogenitás korrekciót is végeztünk:
 - (a) A sorozathoz tartozó referencia kép minden pontjára (legyen a képpontok száma N) meghatároztuk az $M_j = 1/H_j$ hányadost, ahol H_j a háttérkép j -edik pixele ($1 \geq j \geq N$)
 - (b) Minden i kép minden j pontjának intenzitását beszoroztuk a hozzá tartozó M_j konstanssal.



2. ábra.

(a) A tárgy egy vetülete; (b) jelölt felső metszet vágás utáni rekonstrukciója; (c) alsó metszet vágás utáni rekonstrukciója; (d) a felső metszet rekonstrukciója a korrekciós lépések után; (e) az alsó metszet rekonstrukciója a korrekciós lépések után

5. A vetületek zajszintjének csökkentése, valamint az egyes vetületi képeken ugyanazon a helyeken megfigyelhető magas intenzitású (fehér) foltok megszüntetése érdekében küszöbölt medián szűrést alkalmaztunk.

Az előfeldolgozási lépések által eredményezett javulás a 2. ábra (d) és (e) képen látható. Viszont maradtak hiányosságok, hiszen nem tudta megszüntetni az (e) képen lévő szaggatott körgyűrű bal alsó negyedének elmosódását, illetve helyenként a fekete rések erőteljesen megjelenő „árnyékát”.

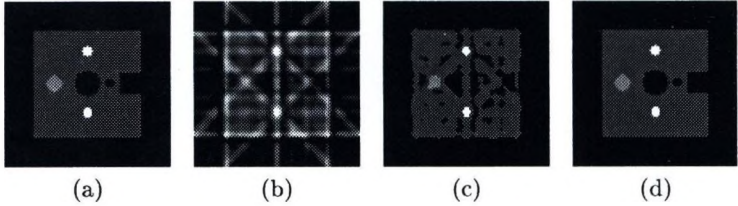
3. Rekonstrukció

A bevezetésben leírtak alapján célunk, hogy minél kevesebb projekció felhasználásával (8-30) kapjunk elfogadható rekonstrukciós eredményt. A kevés felhasznált vetület miatt azonban az általánosság megszorításával meg kellett határozni a rekonstruálandó tárgyak azon osztályát, amelyet várhatólag jó minőségben tudunk rekonstruálni.

Mivel a vetületek számának drasztikus csökkentése folytonos kép előállítását szinte lehetetlenné teszi, így csupán néhány, ismert számú és ismert elnyelődési együtthatójú anyagfajtát tartalmazó objektumok rekonstrukciójára hagyatkozunk. A rekonstrukciós problémát tehát diszkrét tomográfiai feladatként tűztük ki.

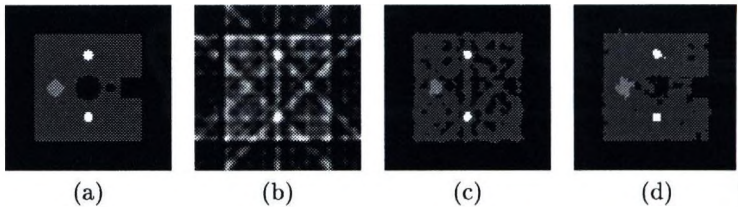
A rekonstrukciós módszer kombinatorikus optimalizálási eszközökkel oldja meg a problémát, és a következő lépésekből áll:

1. Egy folytonos iniciális kép előállítása a SNARK93 programrendszer segítségével [2], amely kezdőértékként szolgál a diszkrét képet előállító iteratív rekonstrukcióhoz.
2. Diszkrétizálás. Az iniciális képen lévő intenzitások küszöbölése az ismert elnyelődési együtthatókhoz. [1]
3. Iteratív rekonstrukció. A módszer szimulált hűtéssel [4] keresi a célfüggvény globális minimumát, ahol a célfüggvény az aktuális iterációhoz tartozó és az eredeti vetületek közötti négyzetes eltéréseként áll elő. Az egyes iterációs lépésekben előállított közelítő képfüggvény az előző lépés képfüggvényének $n * n$ -es monokrom blokkonkénti módosításával áll elő. (Részletesebben ld. [3].)



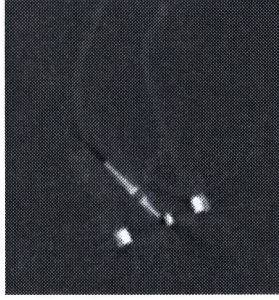
3. ábra.

A rekonstrukció hatékonysága zajmentes esetben 12 vetületből. (a) fantom; (b) a SNARK93 által adott folytonos rekonstrukció eredménye; (c) a folytonos rekonstrukció diszkrétizált változata, az optimalizálás iniciális képe; (d) a rekonstrukció eredménye



4. ábra.

A rekonstrukció hatékonysága 12 vetületből 10%-os zaj hozzáadásával. (a) fantom; (b) a SNARK93 által adott folytonos rekonstrukció eredménye; (c) a folytonos rekonstrukció diszkrétizált változata, az optimalizálás iniciális képe; (d) a rekonstrukció eredménye



5. ábra. Egy cső és szelepjének rekonstruált metszete

Az előbbi rekonstrukciós eljárás hatékonyságát először fantomokon vizsgáltuk [5], és zajmentes esetben az eljárás igen jó eredményt adott (3. ábra). Zajjal terhelte adatok esetén romlik a rekonstrukció minősége (4. ábra), tapasztalataink alapján, azonban az eljárás hatékonysága valamelyest javítható, ha a szimulált hűtés hatékonyság-csökkenésének megfelelően monoton csökkentettük a módosított blokkok méretét.

További gondot jelent a rekonstruálandó alakzat magas geometriai bonyolultsága, ami szintén az rekonstrukció instabilitásához vezet.

4. Tesztelés fizikai mérésű adatokon - Előzetes eredmények

Az algoritmus működését fizikai mérésű adatokon is teszteltük. Az egyik kísérletben az alábbi tárgyat rekonstruáltuk (ld. 5. ábra).

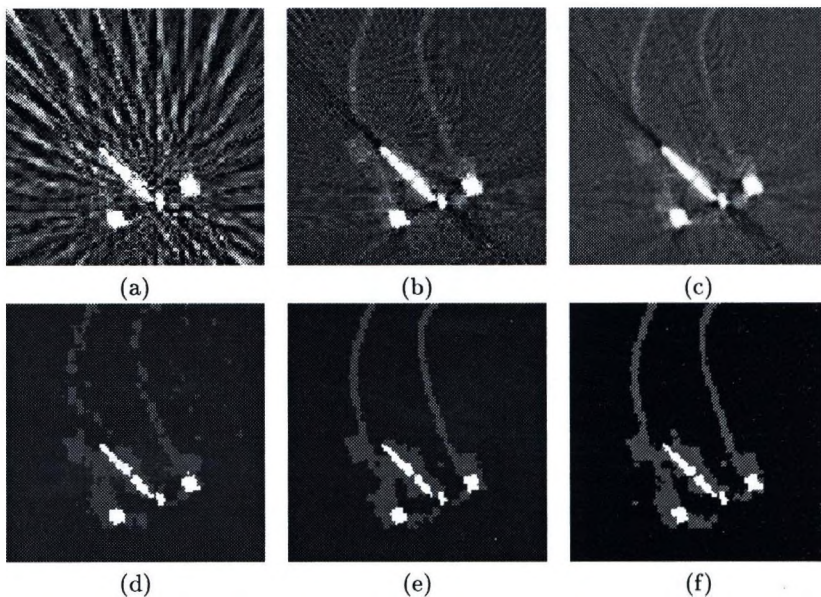
A vetületszám növelésének hatása követhető nyomon a klasszikus és diszkrét tomográfiai eljárás esetében (ld. 6. ábra).

5. Távlati tervek

A korábban bemutatott korrekciós módszereken kívül szükség lehet olyan eljárásra is, amely a vetületi képek alkalmas mozgatásával képes azt az esetleges rossz beállítást korrigálni, amikor a tárgy forgási tengelye nem vetül a kép függőleges középvonalára. Ekkor a vetületi sorozatokon a forgási tengely pontjainak a vetülete szinuszos hullámot ír le.

További problémaként merült fel, a DT rekonstrukciós módszer geometriai bonyolultságra és zajra való érzékenysége. Ezekre jelenleg a priori információk (pl. alakzatok összefüggősége) felhasználásában látjuk a megoldást.

További igény mutatkozott a módszerek egységes keretrendszerbe foglalására és továbbfejlesztésére is (DIRECT [5]).



6. ábra.

Rekonstrukció klasszikus módszerrel (a) 20, (b) 80 és (c) 120 vetületből, ill. a rekonstrukció eredménye az ismertetett módszerrel (d) 20, (e) 80 és (f) 120 vetületből

Hivatkozások

1. A. Kuba, L. Ruskó, L. Rodek, Z. Kiss, Preliminary Studies of Discrete Tomography in Neutron Imaging, submitted to IEEE Trans. on Nuclear Sciences.
2. A SNARK93 programrendszer honlapja: <http://mipgsun.mipg.upenn.edu/snark>
3. N. Robert, F. Peyrin, M. J. Yaffe, Binary vascular reconstruction from a limited number of cone beam projections. Medical Physics 21(12), pages 1839-1851, 1994
4. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, J. Chem. Phys. 21, 1087 (1953)
5. A DIRECT rekonstrukciós programrendszer honlapja: <http://www.inf.u-szeged.hu/~direct>

Automatic Centerline Extraction from Tubular Volumetric Structures

József Koloszár and László Szirmay-Kalos¹

TU Budapest, Dept. of Control Engineering and Information Technology
kolijoco@yahoo.com

Abstract. Skeleton extraction, specifically centerline extraction is an important issue in medical imaging, especially virtual colonoscopy. This paper presents a technique that uses a graph based approach originally developed to complement a distance field algorithm to improve and automate topological peeling. Unlike traditional topological peeling implementations, the algorithm is tolerant to topological irregularities often introduced by fecal residue, partial collapses etc.

1 Introduction

One of the issues in virtual endoscopy is to locate and extract points of the centerline of the biological structure being explored. The information obtained can be utilized in camera navigation, refined segmentation and various other associated techniques. This document focuses on applications, where the data processed is stored as volumetric array of scalars, usually representing density values sampled along points of a 3D rectangular grid.

One of the first stages of the examination is to locate the regions of interest representing the organ to be subjected to examination. This is referred to as segmentation. This process, depending on the precision required can be very complicated. For purposes of virtual colonoscopy, the visualization of the rectal and colonic surfaces from inside the colon, simple semi- or fully automated segmentation can be implemented without serious difficulties. As the colon is inflated prior to scanning, voxels inside the colon have very low density values compared to voxels representing the soft tissue of the colonic wall. To extract the colonic structure a 3D flood-fill constrained to low density values is initiated at an inside point of the colon. The algorithm marks all inside points in a binary volumetric mirror buffer.

The two most prominent approaches to the centerline or more generally the skeleton extraction problem are the use of force or potential fields [1], and topological thinning, sometimes referred to as onion-peeling [3]. Most techniques require one or more of the following conditions to hold:

1. The colon is single connected,
2. There are no holes or tunnels through the colon,
3. The colon is long and narrow.

Topological thinning removes voxels from the surface of the structure layer by layer until only a single thread of voxels remains, which represents a good estimate of the geometric 1D skeleton. Peeling must be performed very “carefully”, however as the original topology of the structure must be maintained. In terms of implementations, this usually means that one or more conditions must be evaluated before a candidate voxel is removed, or peeled away. Though this led to conclusions deeming the technique to be computationally too expensive in the past, even mainstream hardware today facilitates high performance implementations. The real issues to be addressed are practical ones.

Topological thinning is very sensitive to the second condition mentioned above. Though the human colon is devoid of holes or tunnels, fecal residue inside the colon, partial inflation and other circumstances may, and in most cases do result in inadequate colon topology. As the thinning process maintains the topology even if it is incorrect, the resulting centerline may show unexpected properties that may invalidate its use for further processing. Unless the start- and endpoints are explicitly fed the algorithm, either by having the user specify them or using some heuristics to automatically locate them in the volume, high performance topological thinning algorithms yield “hairy skeletons”. Efficient implementations evaluate only the local neighborhood of the voxel candidate for peeling to verify topological consistency. The algorithm preserves generic endpoints — usually defined as points having only a single neighbor in their n -neighborhood — as topological features. If this were not the case, the desired skeletal thread would be peeled away. If the actual endpoints of the desired centerline are specified a priori, the algorithm is free to peel away all other generic endpoints it encounters. Unwanted generic endpoints are usually encountered as the algorithm peels away voxels from spike-like features quite common at irregular regions of the surface. In order to implement a fully automated centerline extraction with local candidate evaluation to maintain high-performance, this issue must be addressed in some way.

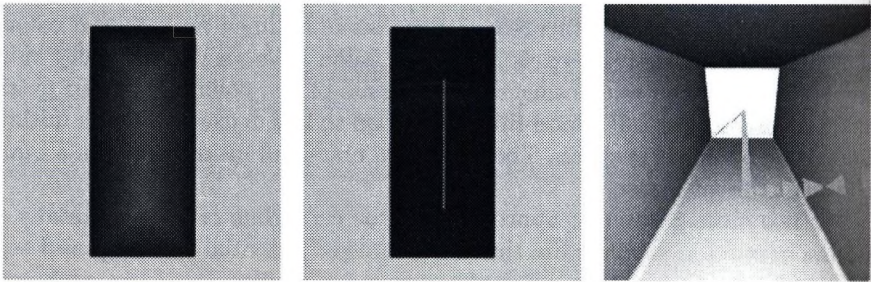


Fig. 1. Thinning a stick-like phantom. Simple topological peeling is sufficient, because of the regular and smooth surface.

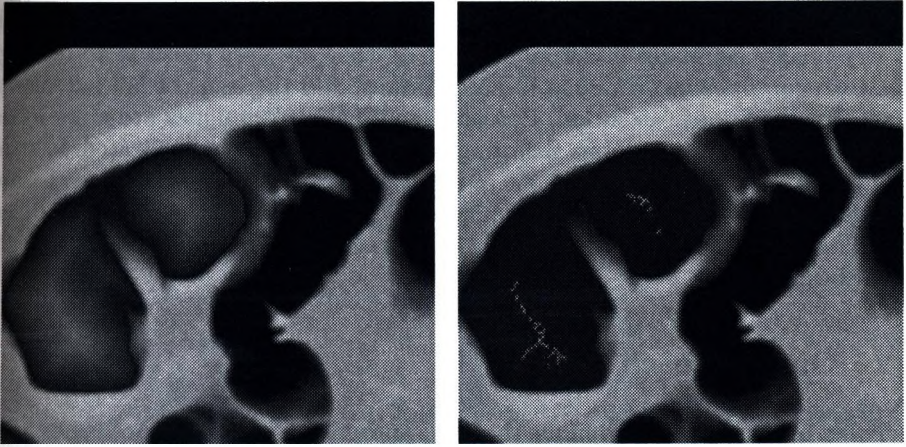


Fig. 2. Thinning a real dataset. As seen in the second image, a large number of hairs line the centerline candidate.

Potential field based methods compute one or more distance fields and evaluate their properties to estimate a centerline. Almost all algorithms compute a field storing the shortest distance to the colonic wall for all internal points, as the centerline is expected to run through local maximum regions in this field. Finding an appropriate skeleton that runs close to the centerline and traverses the entire colon is non-trivial. Additional fields, in most cases a flooded-distance from the endpoint are sometimes also required.

Techniques usually discard larger groups of points at certain stages to reduce the number of calculations in successive ones. One approach yielding high-quality results constructs a weighted graph in its final stage deriving weights from the distance fields, and performs two searches for the longest path in the graph to find the centerline. The algorithm exploits the third condition mentioned above, which is satisfied in almost all cases except those showing collapsed regions due to insufficient colon inflation. Practical advantages of the technique are that it facilitates full automation of the process as it does not require the explicit specification of endpoints, and it is not as sensitive to the violation of the second condition. Abnormal topological features caused by fecal residue or partial collapses are "local" features, meaning that though they result in some unexpected local extremes in the distance fields, but they do not significantly hinder the extraction of the final centerline candidate.

Because of the condition that the colon is long and narrow, one of the actual endpoint candidates can be found by finding the longest shortest-walk initiated from any arbitrary point. The second identical search initiated from the endpoint candidate finds the other endpoint. While force fields can be computed at acceptable speeds on current mainstream hardware, their storage may raise memory issues. The only real disadvantage of these techniques however lies in the complexity of their implementation.

In summary it can be stated that while topological thinning methods yield very precise estimates of actual centerlines, they are very sensitive to topological irregularities too common in actual scans, and the issue of full automation requires additional attention. Potential field based techniques are more robust and easier to automate, but their implementation is more complicated. In terms of speed various implementations of the two approaches perform at comparable levels as most computational techniques involved are of linear complexity. Centerlines for virtual colonoscopy can be extracted in just a few seconds on PC hardware.

2 Our approach

This paper presents a technique that uses automated topological peeling to find a branched and hairy skeleton and searches for the longest shortest-walk to locate the actual endpoints and the centerline. The algorithm takes a binary volume as its input. Voxels representing points of the structure to be processed are black, or ones, while points of the background are white, or zeros. Points of the surface of the structure are stored in a list, which is updated dynamically as layers are peeled away. The list is initiated to contain all black points that have any white ones in their neighborhood. In one cycle all points in the list are checked to see if their removal violates topology. This condition is evaluated locally: the 26-neighborhood of the candidate voxel is encoded bitwise into an integer, which is used to index a Boolean look-up-table (LUT) to see if removal is safe. Any points "uncovered" when a candidate is removed are added to the head of the list. All generic endpoints are preserved. The process is iterated until no more voxels can be removed without violating topology. For a detailed description of the algorithm, refer to [4, 5].

To extract the points approximating a valid centerline from the hairy skeleton thus obtained, a graph is constructed from the list in a single linear pass. One node is generated for each voxel of the skeleton. Two nodes are connected with a weighed, undirected edge, if they are 26-neighbors. Edges are simply weighed by the Euclidean distance between neighbors. The shortest walks to all nodes of the graph are evaluated from an arbitrary node by performing a breadth-first traversal of the graph from the node of origin. The first point among those of maximal walking distance is selected to be one of the endpoints of the centerline.

A second shortest walk search is initiated from this endpoint to find the other endpoint. While traversing the graph to evaluate shortest walk distances, for every node visited a pointer is stored to point to the previous node in the walk. Again the first point among those of maximum walking distance is selected, and the walk leading to it reconstructed from the stored pointers. Nodes of the walk are chosen as the final estimate of the centerline.

The algorithm yields a correct approximation if the structure processed satisfies the third condition. Because the colon is a tubular, single connected structure, peeling away points while maintaining topology results in a skeletal structure that features a thread of voxels corresponding to a centerline estimate as

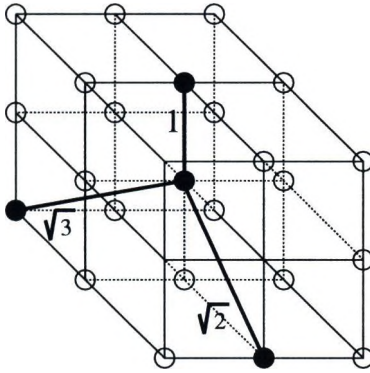


Fig. 3. Graph derived from skeleton. Edges in the 26-neighborhood of a node are weighed by their Euclidean distance. The diagram assumes a homogeneous grid.

its axis, or backbone, with hairs branching off toward the corresponding colonic surface. Because topology is maintained throughout peeling, hairs never have more than one common point.

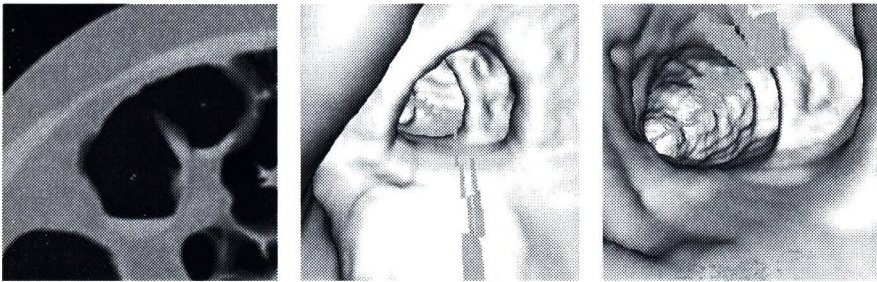


Fig. 4. Axial and endoscopic views of extracted centerline in a real dataset.

3 Implementation results and conclusions

The algorithm has been implemented as part of the virtual colonoscopy application ColVis developed and maintained by the authors. Performance is satisfactory: peeling an average colonic structure takes a few seconds on a 2GHz PC system, while the time it takes to construct and evaluate the corresponding graph to obtain the centerline is negligible. The algorithm yields a very good estimate of the centerline as characteristic to topological peeling while being tolerant to minor topological defects and is fully automated just like prominent potential field based techniques.

4 Acknowledgement

This work has been supported by the National Scientific Research Fund (OTKA ref. No.: T042735), IKTA ref. No.: 00159/2002, the Bolyai Scholarship and the Slovene-Hungarian Fund.

References

1. I. Bitter, M. Sato, M. Bender, K.T. McDonnell, A. Kaufman, and M. Wan. CEASAR: A smooth, accurate and robust centerline extraction algorithm. In *IEEE Visualization 2000, Conference Proceedings*, 2000.
2. B. Csébfalvi. *Interactive Volume-Rendering Techniques for Medical Data Visualization*. PhD thesis, Technische Universität Wien, Institut für Computergraphik und Algorithmen, 2001. <http://www.cg.tuwien.ac.at/research/theses/>.
3. N. Gagvani. Skeletons and volume thinning in visualization. Technical report, 1997. M.Sc. thesis. Brunswick, Rutgers, The State University of New Jersey.
4. K. Palágyi. 3D thinning algorithms (skeletonization) and its medical applications. Technical report, 2003. <http://www.inf.u-szeged.hu/~palagyi/skel/skel.html>.
5. K. Palágyi, E. Sorantin, E. Balogh, A. Kuba, Cs. Halmai, B. Erdőhelyi, and K. Hausegger. A sequential 3D thinning algorithm and its medical applications. In *Proc. 17th Int. Conf. Information Processing in Medical Imaging*, 2001.
6. A. Vilanova. *Visualization Techniques for Virtual Endoscopy*. PhD thesis, Technische Universität Wien, Institut für Computergraphik und Algorithmen, 2001. <http://www.cg.tuwien.ac.at/research/theses/>.

Rendering and Coding of Still and Motion Picture by Stochastic Painting

Levente Kovács¹ and Tamás Szirányi²

¹ University of Veszprém, Dept. of Image Processing and Neurocomputing
kla@vision.vein.hu

² Hungarian Academy of Sciences, Analogical Comp. Lab. Comp. and Aut. Research
Inst.
sziranyi@osztaki.hu

Abstract. We know of many impressive non-photorealistic stroke-based rendering (NPR/SBR) techniques that are able to generate artistic, abstract images or mimic real-world painting styles. We present our enhanced Stochastic Paintbrush Transformation (eSPT) technique for painterly rendering still images and video (Motion-eSPT or MSPT), for creating stroke-coded painting-like images and animation/cartoon-like videos. We introduce several new features over our Stochastic Painting technique that enrich painting abilities (styles and features), reduce transformation times and improve compression of rendered images (like multiscale gradient following edge-placement, stroke templates, area filling morphology). We demonstrate that coding painted sequences as stroke sequences can be lossless, fast and compact (with up to 10:1 compression ratio) and better suited for storing these kinds of sequences than lossless raster coders.

1 Introduction

The Stochastic Paintbrush Transformation (SPT) [13], [14] was developed modeling a primitive painting technique when transforming an input image into a series of rectangular brush-strokes, in a coarse-to-fine way. Then Monte Carlo Markov Chain (MCMC) optimization was added at the stroke acceptance step [15] seriously reducing the code-length of transformed images. Later work has shown that this transformation method is suitable not only for producing painting-like images, but also for other areas e.g. image classification and indexing [16].

We are to show that a properly designed video-painting method can generate painted sequences that can be effectively coded losslessly, far better than pixel-based lossless coders would do. We present a lossless compression scheme for eSPT-generated videos which can achieve around 10:1 compression ratios, coding stroke-sequences. It contains motion detection, cut detection, region of interest selection and uses the eSPT algorithm for keyframe coding.

2 Paintbrush Transformation

Our original [14] and present still-image painting technique is similar and in many ways different than other known painting techniques, like: [3], [10], [11], [4], [5], [6], [2], [12], [7].

The present work, which is an extension of [8], has its roots in [10], [13], [5]. We introduce some novel features in our automatic painting technique, which make it differ from other similar methods in the field, like the following: unrestricted stroke templates, no stroke grid, stroke positions are stochastic, region of interest areas, optionally following only the edges of the frames after a specified painting step, multiscale gradient-based edge orientation process, area filling, storing the output as a compressed stream of stroke-series.

2.1 Enhanced Stochastic Painting (eSPT)

The transformation starts with larger strokes then refines the actual image with smaller ones. During the coarse-to-fine refinement process a stroke with given size (iterated through stroke-sets), orientation (eight possible, rotated by $p/8$) and color (interpolated from samples drawn from the model image) is placed on the canvas, if its placement makes the painting converge to the model (its placement does not induce such errors that the painting achieved will be less like the original than the painting at the previous step). The resulting image is described/represented by the stroke-series used (for an example see Figure 1). The original algorithm works with about 10 sizes of rectangular strokes (grouped



Fig. 1. Full painting with 10 layers, rectangular strokes. First, fifth and final step.

in stroke-sets), paints each layer until there is no significant improvement with the actual stroke, then switches to a smaller set. Using rectangular strokes can be good (can cover almost any shape, emulate long lines and contours, etc.) but can also be a drawback (e.g. there are many round shapes to cover). The first extension is the ability to use different stroke-templates.

When the painting process starts, one can choose a brush template. Here templates are generated from 60x60 pixel grayscale images. The intensity values of a grayscale template are used for weighting the color values sampled from the model image. For an example image and sample templates see Figure 2. A single



Fig. 2. Sample templates and painted image.

image could also be painted using multiple different strokes, e.g. starting with a large round stroke then switching to a long rectangular one, etc.

In [14] strokes are placed at random positions with random orientation. This can lead to situations when the placement of a stroke makes some improvement to the image in general, but it causes such local errors, which can only be corrected with multiple smaller strokes at a later step. This led to the conclusion that the random orientation is not always an acceptable solution. What we did was trying to place strokes with the orientation of nearby edges. The idea behind this approach was that if the orientation of larger strokes follows the orientation of edges on the image, than smaller strokes need only to cover the remaining areas well. This approach would also eliminate such induced errors when a stroke gets placed normal to an image edge.

We use an implementation of Lindeberg's [1] multiscale edge detection method. This method tries to find main edges by searching for edge positions which produce an accentuate curve in scale-space. We use the edge definition when at a given point, if it is an edge-point then the second order derivate is zero and the third order derivative is negative (for each respective direction). Using 10 scales we calculate scale-space derivatives. Then we search for all curves connected on the same (maximal) scale and calculate the edge strength, which is a line integral of weighted derivatives along the actual curve. We store all connected curves on the image and sort them in descending order of their summed weights. After all this we display the first N curves on an edge map.

At an earlier stage we used multiple (10) layers of strokes. Switching to a smaller stroke-set occurred when painting the present step did not cause enough improvement. This method can be a quite slow one, considering all the other necessary steps as well. A next almost evident step was the following: we start with a stroke and make the algorithm to stop painting the actual layer when it gets covered beyond a given percent (relative to the whole canvas area). When this stage is achieved we use color morphology to cover the remaining areas. This is what we call here area filling. (example on Figure 3 Right image.)



Fig. 3. Left: Layer of strokes covering 50% of the canvas, following the edge orientation map; then one layer of strokes, empty areas filled in with weighted averaging; then the final image. Right: Area covered partially then morphed.

If we want to take better care for a special part of an image (or a special part of frames) then we define a region of interest, which will only be processed (painted/covered) at the last step, with the smallest available stroke. The error of the ROI relative to the model image will be smaller than that of the rest of the image.

3 Transforming Video

First a cut detection is performed on the video source to identify scene changes. Then we start with fully painting the first frame with eSPT (which will be done for all keyframes). Between keyframes motion detection is performed to identify the motion areas which will be partially repainted. Output is stored as a lossless compression of stroke series and motion data, on a frame-by-frame basis. Generation of painted areas and frames can be done with all the possible features of the eSPT algorithm. When repainting the motion areas and do the post-processing which eliminates over-painted and therefore hidden strokes the stroke density becomes generally uniform.

When doing the painting, we take into account the specified ROIs, the optionally checked option to follow the edges on the frames (for achieving better contours) and the position of the cuts. ROIs can be defined for a single frame, for a series of frames or for the whole sequence, or even different ROIs for all the frames.

3.1 Transformation Details (Motion-eSPT)

The main steps of the video painting algorithm are as follows: 1. Setting the parameters (key-frame frequency, type of motion detection to use and other parameters like block radiuses, search ranges, thresholds, etc.); 2. Cut detection; 3. Edge detection and Paintbrush transformation of frame $F(i)$ to $F^*(i)$, Huffman compressing $F^*(i)$ and writing of $F^*(i)$ into the intermediate format; 4. Edge detection and optical flow calculation between $F(i)$ and $F(i+1)$; 5. Transformation

of the areas where motion occurred using the optical flow field data calculated at the previous step, and the edge data if specified; 6. Writing the partially transformed frame data onto $F^*(i)$, generating $F^*(i+1)$ and writing it into the intermediate format along with the motion data; 7. If the next frame will be a key-frame then increase i and jump to step 3, otherwise to step 4.

Frame data gets Huffman-encoded before storing into the output file and the motion data is run-length encoded. Using the data stored in the output file the painted video can be reconstructed stroke-by-stroke, frame-by-frame. The average compression ratios of the output is about 5-10:1. When using %-coverage settings and area filling we get a very good feature for the video coding part of the algorithm: the frame sizes tend to be flat and uniform which gives us better control over the size of the output.

We tried multiple motion detection algorithms to calculate optical flow (block matching, gradient-based and hierarchical gradient-based methods) to obtain the areas where motion occurs. And we stuck to an implementation of the latter hierarchical detection method because of its speed and robustness.

We use a simple, but fast cut-detection algorithm to detect cuts in the input video, and treat the first frame after the cut as key frame. The cut-detection algorithm we developed works with the edge map-differences and pixel-differences of the frames to get possible scene change locations. Some artifacts can remain on the image because bad cut detection, but they can be eliminated by increasing the keyframe density at uncertain scene change places.

Compression A placed stroke can be described with its parameters (identification, color, position). This representation stays the same for every type of stroke (let that be rectangular, or user-defined). The stroke-stream is Huffman-coded and the motion data is run-length encoded. If we use area filling no extra storing is necessary, only a status byte which signals the decoder, that it has to perform itself the area filling on the layers of the decompressed frames.

We always stated and stuck to the approach that storing painted sequences as stroke-series data can lead to better lossless compression ratios than coding them with other pixel-based lossless coders (like huffyuv or jpeg-ls which we used in our tests). The next section is to demonstrate that if this was true for the past [8] it is even truer for today.

4 Results

In the video transformation process we have keyframes at given intervals and sometimes denser because of the detected scene changes. So it is important to check how the coding behaves when changing the keyframe density. Figure 4 shows just that. The coding scheme used for these checks was: rectangular strokes, 10 layers, full painting, PSNR-driven. The Left graph on Figure 5 shows frame size variations of normal and eSPT generated video frames. An important thing is that video frames transformed using eSPT features (fewer layers, %-coverage, edge-orientations and area filling) are more stable in size over time.

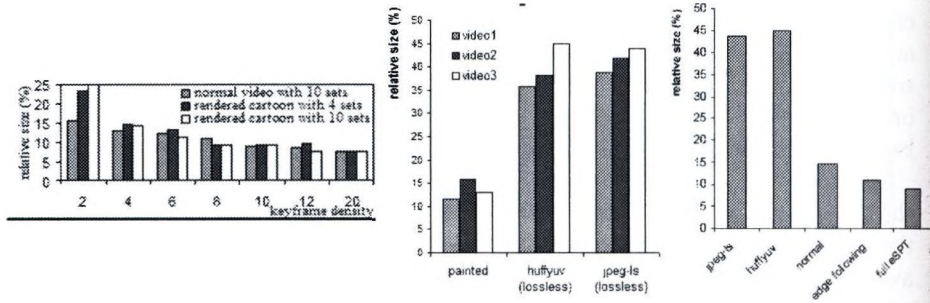


Fig. 4. Left: Relative compression sizes (related to uncompressed video source) for 3 sample videos with changing keyframe density (smaller means denser). Middle: Lossless coding of stroke series is more efficient than lossless pixel-based coding of painted sequences. Right: Coding with eSPT features gives even better compression ratios.

The Right graph on Figure 5 is to show that besides a). giving stable frame sizes, b). less vibration, c). smaller frame sizes and d). still being a lossless coder, coding with eSPT is in addition even faster than normal stochastic painting (SPT). Giving the previous numbers the transformation times can be as much as 10 times faster.

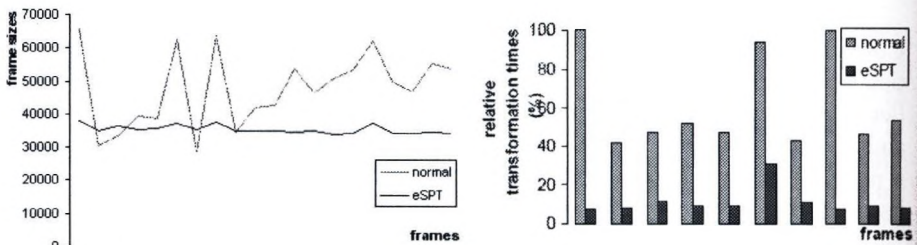


Fig. 5. Left: Coding with eSPT features gives stable frame-sizes. Right: Coding with eSPT is more faster than with SPT stochastic painting.

5 Conclusion

We presented our new eSPT technique for painterly rendering video sources and a lossless compression scheme for coding and storing painted sequences stored as stroke-sequences instead of pixel data with raster lossless image coders. We have shown that its abilities for image representation, coding and feature extraction make it a worthy partner of other existing painterly rendering techniques. The benefits of its lossless coding are becoming clear when we compare its compression ratios with existing lossless raster image coders. We also demonstrated that

the new features introduced (multiscale edge orientation stroke placement, covering layers till a given percent and then do area filling with color morphology) do not reduce painted quality but make the coder more stable and faster. By using the color morphology-based area filling and grayscale templates styles can be emulated, different effects can be produced for still and motion picture.

References

1. Almansa, A. And Lindeberg, T. 2000. Fingerprint Enhancement by Shape Adaptation of Scale-Space Operators With Automatic Scale Selection. *IEEE Trans. Image Processing*, vol. 9., 2027-2042.
2. Curtis, C. J., Anderson, S. E., Seims, J. E., Fleischer, K. W., And Salesin, D. H. 1997. Computer-generated watercolor. In *Proc. of ACM SIGGRAPH 97*, 421-430.
3. Haerberli, P. 1990. Paint by numbers: Abstract image representations. In *Proc. of ACM SIGGRAPH 90*, 207-214.
4. Hertzmann, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *Proc. of ACM SIGGRAPH 98*, 453-460.
5. Hertzmann, A., and Perlin, K. 2000. Painterly Rendering for Video and Interaction, In *Proc. of NPAR 2000*, 7-12, 121.
6. Kalnins, R.D., Markosian, L., Meier, B.J., Kowalski, M.A., Lee, J.C., Davidson, P.L., Webb, M., Hughes, J.F., And Finkelstein, A. 2002. WYSIWYG NPR: Drawing Strokes Directly on 3D Models. In *ACM Transactions on Graphics*, 21.
7. Kaplan, M., Gooch, B., And Cohen, F. 2000. Interactive Artistic Rendering. In *Proceedings of NPAR 2000*, 67-74.
8. Kovacs, L., And Sziranyi, T. 2002. Creating Animations Combining Stochastic Paintbrush Transformation and Motion Detection, In *Proceedings of 16th ICPR, IAPRIEEE, Quebec, Canada*, vol. 2, 1090-1093.
9. Kowalski, M.A., Markosian, L., Northrup, J.D., Bourdeu, L., Holden, S., And Hughes, J. 1999. Art-Based Rednering of Fur, Grass and Trees. In *Proc. of ACM SIGGRAPH 2001, ACM Press / ACM SIGGRAPH, New York*. E. Fiume, Ed., *Computer Graphics Proc., Annual Conference Series, ACM*, 433-438.
10. Litwinowicz, P. 1997. Processing images and video for an impressionist effect. In *Proc. of ACM SIGGRAPH 97*, 407-414.
11. Meier, B.J. 1996. Painterly Rendering for Animation. In *Proc. of ACM SIGGRAPH 2001, ACM Press / ACM SIGGRAPH, New York*. E. Fiume, Ed., *Computer Graphics Proc., Annual Conference Series, ACM*, 477-484.
12. Santella, A., And DeCarlo, D. 2002. Abstracted Painterly Renderings Using Eye-Tracking Data. In *Proc. of NPAR 2002*, 75-82.
13. Sziranyi, T., Toth, Z., And Kopilovic, I. 1999. Paintbrush Image Transformation. Fundamental Structural Properties in Image and Pattern Analysis, In *Proceedings of FSPIPA99, IAPR*, 157-168.
14. Sziranyi, T., And Toth, Z. 2000. Random Paintbrush Transformation. In *Proc. of 15th ICPR, Barcelona, IAPRIEEE, V.3*, 155-158.
15. Sziranyi, T., And Toth, Z. 2001. Optimization of Paintbrush Rendering of Images by Dynamic MCMC Methods. *Lecture Notes on Computer Science, Springer Verlag, Vol.LNCS 2134*, 201-215.
16. Sziranyi, T., Kato, Z., Ji, X., Toth, Z., And Czuni, L. 2002. Content-Based Image Retrieval Using Stochastic Paintbrush Transformation. In *Proc. of ICIP 2002, Multimedia Retrieval and Applications, IEEE, Rochester*.

Automatic Stabilization of Vibration on Archive Films

Attila Licsár¹, László Czúni¹, Tamás Szirányi^{1,2}, and

University of Veszprém, Department of Image Processing and Neurocomputing

¹H-8200 Veszprém, Egyetem u. 10, Hungary

licsara@almos.vein.hu, czuni@almos.vein.hu

Analogical and Neural Computing Laboratory, Computer and Automation Research

Institute, Hungarian Academy of Sciences

² H-1111 Budapest, Kende u. 13-17, Hungary

sziranyi@osztaki.hu

Abstract. Image vibration is a typical type of degradation that is difficult to restore in an automatic film restoration system. It is usually caused by improper film transportation during the copying or the digitization process. We have developed a method for automatic image stabilization consisting of two main steps: estimating vibration then correction by drifting the whole frame. Earlier stabilization algorithms are unsuccessful in cases of multiple motions and human interaction is necessary to achieve satisfactory results. The proposed technique is automatic and avoids false results for most difficult situations as shown in example. The paper describes the technique used for motion estimation of regions; the selection of image regions for adequate vibration estimation; and the method of stabilization.

1 Introduction

Archive films suffer from several degradations such as blotches, scratches, flickering, image vibration, fading, etc. [1]. Frame vibration can be defined as a rapid random spatial drift and/or rotation between consecutive frames. It is usually caused by unsteady film transportation in film cameras, telecine and copying machines or other instruments. The observed vibration of films can be very complex since camera ego-motion can also cause vibration having serious effects on the 2D projection of 3D sceneries. It is important to note that our vibration model does not deal with the ego-motion of the recording camera itself (neither rotation), however our solution has side effects on camera vibration as will be explained and demonstrated later.

The paper is organized as follows: in Section 2 we briefly overview earlier stabilization methods emphasizing differences found in comparison to our model. In Section 3 the model of the new stabilization algorithm is described. We explain motion estimation, vibration estimation and filtering. Then we present the result of automatic stabilization in Section 4. Conclusion and Future Work comes finally in Section 5.

2 Overview of Stabilization Methods

Basically, there are three main sources of image vibration: camera motion (ego-motion: translation, rotation, zoom), film vibration (in our article the global drift due to unsteady film transfer in a copy or in a telecine machine is called film vibration), object vibration (shaking objects such as a moving car on a rough road).

The removal of the random vibration from motion can be related to a wide area of image processing applications. Taking a closer look, stabilization algorithms can be grouped into three main classes:

1. Methods to remove vibration originating from camera ego-motion (for example [2][3]);
2. Methods to remove film vibration by global motion estimation and filtering (for example [4][5][6]);
3. Methods to remove object vibration (this problem often arises as object tracking).

Methods of the second-class implement stabilization process based on the global motion of a frame, while in the third case local object motion gives a support for stabilization. In some papers local feature points are used to estimate motion vectors to remove film vibration [7]. In contrast, our proposed model combines classes 2 and 3 by automatic ROF (Region Of Fixation) selection where stabilization will be done based on this specific image region. The ROF can be the whole frame (initial assumption) or a specific fixed region (i.e. an object) of the image; everything depends on the local structure of the motion on the film. In case of manual restoration an operator should mark some areas as ROFs on the image. The purpose of our method is the stabilization of film sequences by automatic ROF selection.

Please, notice here that camera ego-motion filtering is not the target of our method, but our model will not differentiate camera vibration from film vibration at the selected ROF. As found in our experiments, and shown in some examples, this side effect is not disturbing, while differentiating between these two kinds of motion would require 2D/3D parametric models. It results in a considerable increasing of computational costs. Talking about any of the above classes, almost all algorithms contain the following three main steps: 1. estimating motion (of camera, object, or feature points); 2. filtering motion trajectory to estimate the ideal noise-free motion; 3. modify image motion (i.e. warp the image) to get smooth trajectories.

As for the estimation of motion, several techniques can be applied. One can set up 2D or 3D parametric models for ego-motion estimation [8]; can make block matching [9][10][11], feature point tracking [12], image derivatives for measuring local motion; can use phase correlation [13][14] to estimate global motion. Once motion has been measured it is smoothed by different techniques. Simple methods include low-pass filtering, median filtering [15][16] while more complicated solutions apply motion statistics or error propagation control [17], polygonal approximation [18], Kalman filtering, etc. Finally, the input sequence is processed

to remove vibration by applying a drift or other warping techniques. Methods adequate for the stabilization of archive films suffer from the following problems: 2D/3D parametric models are too time consuming, global motion estimation fails in case of complex scenes, models based on local features are sensitive to noise (e.g. flickering, blotches), in most cases robust and fast tracking algorithms require human interaction.

3 Unsupervised Image Stabilization

The main task of film restoration is to differentiate between the three sources of vibration (in an implicit or explicit way) and to eliminate film vibration effects only. The restoration method should be computationally feasible for high-resolution image sequences, robust for flickering and blotches, automatic in most cases.

In our model, image vibration affects all image pixels the same way (i.e. we excluded non homogenous shrinking and film rotation from our model at this stage): the same unwanted random motion (separable in x and y directions) is considered to be added to the motion of all image pixels. Image stabilization is the process of removing this noise from the motion of all image points by repositioning the whole frame with a global drift. This can lead to image loss on the borders of a frame but this can be eliminated by applying zooming, with a magnitude proportional to the amplitude of the vibration, or by inpainting from a previous/successive frame that has adequate border information.

The main point of estimating and compensating vibration is to find the global drift and reposition the whole frame in the opposite direction. We can estimate global vibration with the help of the vibration of local regions. This process is as follows:

1. Find suitable velocity functions (i.e. find characteristic local motion e.g. the global motion of the selected ROF).
2. Smooth these functions to estimate a stabilized image.
3. Measure temporal deviations of these functions from the smoothed version.
4. Estimate global vibration with the help of these local deviations.

Local deviations will contain the superposition of film vibration and camera vibration. The effect of camera vibration can differ from region to region depending on the 3D structure of the scene, while film vibration is constant over the whole frame. Since it would be very time-consuming to separate camera vibration from film vibration the proposed technique compensates the superposition instead. As shown in the examples in most cases this is not a limitation of our algorithm and even sequences with camera vibration are reconstructed adequately.

3.1 Motion Estimation with Phase Correlation

We have chosen the phase correlation technique for estimating motion of an image region [19]. This method is relatively insensitive to fluctuations in image

intensity (flicker and blotches are very typical for archive films) due to two reasons: it uses normalized Fourier coefficients and input images are high-pass filtered first, since edges and textures are less affected by intensity variation (flicker). The phase correlation technique evaluates the phase of the Cross Power Spectrum (CPS) of consecutive images. If one image (f_1) is a shifted version of the other (f_2) then the CPS is as follows:

$$\frac{F_1(\zeta, \eta)F_2^*(\zeta, \eta)}{|F_1(\zeta, \eta)F_2^*(\zeta, \eta)|} = e^{j2\pi(\zeta x_0 + \eta y_0)} \quad (1)$$

where F_1 and F_2 are the Fourier transforms of the images and F_2^* is the complex conjugate of F_2 . The inverse Fourier transform of the CPS has an impulse with coordinates located at the required shift (x_0, y_0).

In case of real video sequences successive frames are never perfectly shifted of each other and instead of an impulse a so-called correlation surface can be obtained, containing peaks indicating the motion components in the block. (See later examples.) The phase correlation technique offers a high degree of accuracy and robustness [20][21].

3.2 Automatic ROF Selection

Most stabilization algorithms require the user to select an object (or say an image region) that has a suitable motion trajectory. In our algorithm an image region can describe vibration adequately when the following conditions are met:

- The image area can be characterized with one typical (global) motion.
- The image area should have relatively smooth motion.

The first question is how to find those image areas that represent image vibration suitably: in our algorithm we apply a top-bottom search for adequate ROFs. The motion estimation is often affected by film defects (e.g. blotches, scratches) and non-rigid object motion. For the better estimation accuracy we need the largest regions on the image with unambiguous motion. Accordingly, we use a quad-tree based image splitting method (see Figure 1.). Automatic ROF selection starts with processing the first frame of an image sequence and the selected ROFs are valid for a predefined length of app. 0.5-1.0 seconds (depending on the constancy of the film). The main steps of automatic stabilization follow:

1. The whole image is selected as one ROF to be used for vibration estimation.
2. The motion of the ROF is estimated by the phase correlation technique.
3. Peaks of the inverse Fourier transform of the CPS are analyzed: if a unique peak is found then we suppose that a global motion vector can characterize the ROF. Uniqueness satisfies two constraints: 1. the height of the peak is over a predefined threshold (here 0.023); 2. no other local maximum can reach 30
4. The speed, in both directions of x and y , of ROFs at the leaves of the tree are calculated. Velocity functions are plot for each ROF as a function of time and straight lines are fitted with linear regression. We choose the function with the smallest square error. This function determines the ROF used for further vibration estimation for the whole frame within the short sequence.

If the tree has maximum depth (here 3) and the motion is ambiguous (but the tree cannot be divided into sub-regions) then the motion function can be estimated from different blocks of similar motions. Then we calculated the median of the motion vectors of several ROFs with the smallest square error.

If automatic ROF selection is not satisfactory for some reason, user interaction is still allowed to choose the appropriate region for manual vibration compensation. In this case the system indexes the frame and its features, then at the next occurrence of a similar scene chooses the user selected ROF [22]. In our experiments no such case with necessary manual interaction occurred.

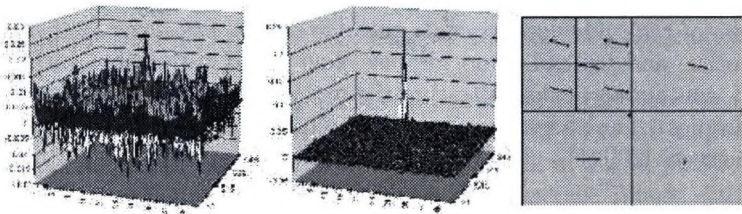


Fig. 1. Ambiguous and unambiguous peaks of the inverse Fourier transform of the CPS and an example of quad-tree with motion vectors.

4 Example Sequence "Chevy"

In this Section we illustrate the performance of the proposed stabilization. In the image sequence called "Chevy" we measured the horizontal motion of the car manually. The input sequence contains camera motion, strong local motion and has definite vibration either. The global stabilization technique fails while adaptive ROF selection stabilizes the sequence (see right of Figure 2) In the left of Figure 2 two frames (upper line frame: # 13, bottom line: frame # 24) can be compared visually. The horizontal position of the right lamp of the car is rapidly altering when the non-adaptive technique is applied. Automatic ROF selection works well.

5 Conclusions and Future Work

We introduced a new adaptive model for film stabilization. It enables automatic operation with minimal human interaction reducing costs in film restoration. The proposed technique is robust for noise and can handle complex scenes. There are two main areas where improvements could be achieved: 1. to introduce rotation and zoom into the vibration model. 2. to increase computation speed by multi-scale processing in case of very high-resolution sequences (2-6K). This

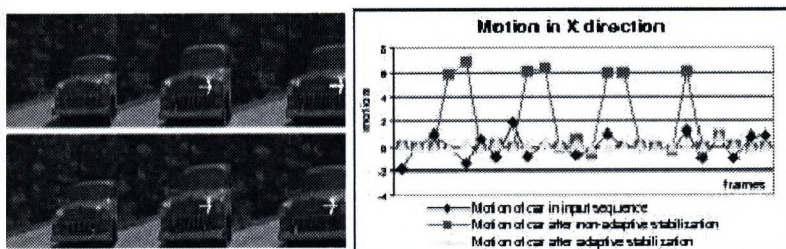


Fig. 2. Left: in columns are input, adaptive and non-adaptive filtered frames # 13 and # 24 (in rows) of sequence "Chevy". Right: the motion of the car measured manually in direction X.

means that the first few levels of the tree could be processed at lower resolution with satisfactory results. For better visual presentation of results please go to: <http://www.knt.vein.hu/~dimorf/demo>

References

1. A. Censi, A. Fusiello and V. Roberto, "Image Stabilization by Features Tracking," In 10th International Conference on Image Analysis and Processing, Venice, Italy, 1999.
2. T. Chen, "Video Stabilization Algorithm Using a Block-Based Parametric Motion Model," Technical Report, Department of Electrical Engineering, Stanford University, 2000.
3. Z. Duric and A. Rosenfeld, "Shooting a Smooth Video with a Shaky Camera", 2002.
4. M. Irani, B. Rousso, and S. Peleg, "Recovery of Ego-Motion Using Region Alignment," IEEE Trans. on PAMI, pp. 268-272, 1997.
5. Attila Hanis, Tamás Szirányi, "Measuring the motion similarity in video indexing",
6. L. Hill and T. Vlachos, "On the Estimation of Global Motion using Phase Correlation for Broadcast Applications," in Proceedings of IPA, pp. 721-725, Manchester, 1999.
7. C. D. Kuglin and D. C. Hines, "The phase correlation image alignment method", in Proceedings of International Conference on Cybernetics and Society, pp. 163-165, IEEE, 1975.
8. Paul Read and Mark-Paul Meyer, "Restoration of Motion Picture Film", Butterworth-Heinemann, 2000.
9. T. Saito, T. Komatsu, T. Hoshi, and T. Ohuchi, "Image Processing for Restoration of Old Film Sequences," 10th International Conference on Image Analysis and Processing (ICIAP), 1999.
10. P. Schallauer, A. Pinz and W. Haas, "Automatic restoration algorithms for 35mm film," VIDERE, MIT press, 1(3), pp. 60-85, 1999.
11. L. Tenze, G. Ramponi, and S. Carrato, "A complete system for old motion picture restoration," in Proceedings of EVA 2002, Florence, Italy, pp. 110-114, Mar. 2002.
12. T. Vlachos, "Simple Method for Estimation of Global Motion Parameters Using Sparse Translational Motion Vector Fields", Electronics Letters, pp. 60-62, 1998.

13. D. Wang and L. Wang, "Global Motion Parameters Estimation Using a Fast and Robust Algorithm", IEEE Trans. Cets. And Sys. For Video Technology, pp. 823-826, 1997.
14. Y. Wu and D. Suter, "Noisy Image Sequence Registration and Segmentation," in Proceedings of Second Asian Conference on Computer Vision, ACCV'95, Singapore, pp. 1533-1537, December 1995.
15. <http://www.dv99.com>
16. <http://steadyhand.dynapel.com/>

Wide Baseline Stereo: From Pixels to 3D Model

Zoltán Megyesi, Dmitry Chetverikov, and Zsolt Jankó

Computer and Automation Research Institute and Eötvös Loránd University
Budapest, Kende u.13-17, H-1111
megyesi@sztaki.hu

Abstract. The Image and Pattern Analysis (IPAN) group of Geometric Modeling and Computer Vision Laboratory at MTA SZTAKI, Budapest, is doing research in the high-level interpretation of image data for building rich and precise models of real-world objects and scenes. This paper gives a brief overview of our recent results in wide baseline stereo, when the two views of a scene are widely separated and differ significantly.

1 Introduction

This paper summarises recent results of IPAN research in the area of **wide baseline stereo**, when the two views of a scene are widely separated. The corresponding features of the images are far away and are subject to affine distortion. This basic problem is often further complicated by occlusions. For these reasons, a critical step of the reconstruction process is establishing the initial (sparse) correspondences and building the epipolar geometry.

In our approach, we automatically detect and identify (match) the corresponding high-level structural features of a wide-baseline stereo pair. The structural features considered are the dominant, compact periodic structures, called periodic distinguished regions, PDRs.

The initial correspondences between the PDRs are not precise enough for building the accurate epipolar geometry. However, they can be used to obtain a rough affine alignment of the images, in which the corresponding local features, such as Harris corners, are much closer to each other than in the initial pair. Once this has been done, a conventional, close-range feature matching procedure is used to find a sufficient number of precise correspondences.

Based on the epipolar geometry, we rectify the two images and apply a recently developed dense matching procedure based on affine region growing. The procedure accounts for affine distortion of the local features that is typical for wide-baseline stereo images. Comparison to conventional dense matching techniques shows visible improvement in the quality of the reconstruction. 3D models reconstructed from various wide baseline stereo pairs are shown.

2 PDR correspondences

Consider a wide baseline stereo pair. Assume that a large *part* of the scene has the following property: relation between the two images of this part can

be *roughly* approximated by a 2D affine transformation in the image plane. The transformation has 6 parameters, which means that 3 correspondences are required to estimate it. A key idea is that once an *approximate affine alignment* has been found and applied, the corresponding image features get much closer and a conventional matching algorithm can be used.

Assume the pair contains a few compact and distinct, periodic texture regions. Finding and identifying the regions in the two images, one can establish three rough correspondences between the centroids of the regions. Then one can estimate the 2D affine transformation that approximately aligns the images. The following **framework algorithm** summarises the proposed method.

- (1) Detect periodic distinguished regions in both images and at varying resolution levels.
- (2) Describe PDRs by their affine-invariant regularity features and periodicity vectors.
- (3) Identify PDRs in the two images and establish tentative region correspondences.
- (4) Test each triple of region correspondences by computing rough affine alignment and applying a conventional matching algorithm to the roughly aligned pair.
- (5) Select alignment that produces the highest matching score and apply inverse transformation to obtain point matches for original pair.
- (6) Estimate fundamental matrix and rectify the images.
- (7) Refine point matches by local affine matching in rectified images.
- (8) Re-estimate fundamental matrix from refined point matches.

Steps 1–3 of the algorithm are described in our previous paper [4]. Step 1 involves computation of our regularity features [3] in a sliding window, which is called regularity filtering. PDRs are detected in both images of a wide baseline stereo pair and at different resolution levels. Relatively large, dominant regions are normally detected. Their number is small.

Search for correspondences involves two different but related operations: PDR matching and feature point matching. First, tentative PDR correspondences are established (step 3) as discussed in [4]. Then each triple of tentative correspondences is tested and the best one is selected. Testing a triple includes computing the rough affine alignment for this triple, then applying the modified KLT Tracker [2] to the roughly aligned pair. To speed up the procedure, the KLT matching is done at a reduced image resolution. The best alignment is the one that produces the highest matching score, that is, the largest number of successfully matched feature points. Finally, the inverse affine transformation is applied to obtain point matches for the original pair.

The process of PDR matching is illustrated in figure 1 where the tentative correspondences obtained at both resolution levels are shown by lines connecting the centroids of the regions. The same region may appear at different levels. Correspondences forming the best matching triangle are indicated by small circles marking the centroids.

Given three pairs of corresponding points, a rough affine alignment can be determined by solving a simple linear system. A modified version of the KLT Tracker [2] is applied to the roughly aligned pair. When an alignment is invalid, only a few spurious matches appear. When it is valid, the number of matches

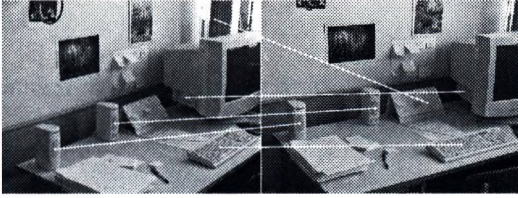


Fig. 1. PDR matching in JANKO pair.

increases drastically, indicating that the tested triangle is a valid candidate. The valid candidate with the highest matching score is selected.

Figure 2 shows the matches obtained for the best candidate triangle marked in the two images. The successfully matched points lie close to the plane of the best triangle. The regions that are definitely off the plane, such as the pictures on the wall, were not matched because of the large parallax.

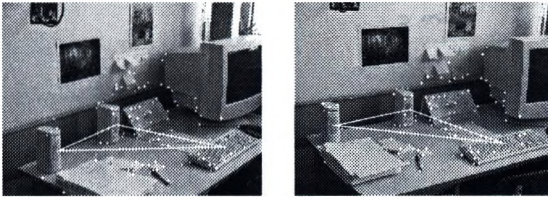


Fig. 2. Points matched in JANKO pair.

When a sufficient number of point matches has been obtained, the fundamental matrix is estimated using a robust method such as RANSAC. A selection of efficient methods are available [5]. An example of the epipolar geometry built for the JANKO pair is demonstrated in figure 4.

Based on the epipolar geometry, the images are rectified using a standard technique [5]. Then the point matches are *refined* in the rectified images in order to make the epipolar geometry more stable and precise. The task is twofold: to remove the outliers that may still be present, and to refine the locations of the matches. This is achieved by local affine pattern matching constrained by the epipolar geometry. Finally, the fundamental matrix is re-estimated from the refined point matches.

3 Affine dense matching

Our affine dense matching algorithm [1] operates on a gray-scale **rectified image pair**. It assumes that the surfaces are piecewise-smooth, Lambertian and textured. It is also assumed that: (1) projective distortion between the views can locally be approximated by an affine transformation; (2) on a smooth surface, this affine transformation and the disparity change gradually.

The algorithm is based on propagating the affine distortion parameters of seed correspondences. This is similar to **region growing** [8] used in image segmentation. The idea of combining affine matching with region growing was proposed by Otto and Chau [7]. However, our propagation and stopping criteria are different: they are surface-oriented. The affine propagation procedure applies the following main rules. **Propagation rule:** Best affine transformation and best disparity are propagated to the neighbour as an initial guess, then refined. **Stopping rule:** If a significantly different affine transformation performs better, the propagation is halted. The procedure tries to match smooth areas around the seed points, eventually segmenting them into connected regions.

Efficient region growing needs properly located, reliable *seeds*. In our case, these seeds should be well-discriminable, matchable points located on all relevant surfaces. We use the KLT corner detector [2] to automatically select the seeds. Examples of seeds are shown in figure 3. The seeds are matched by exhaustive search with left-right consistency check.

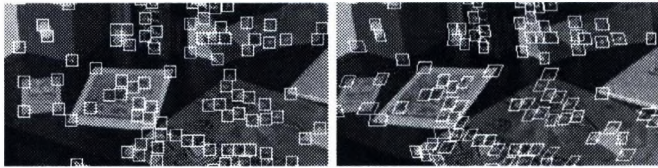


Fig. 3. Best matching affinely distorted windows for seed points.

4 Wide baseline stereo tests

In this section we show sample wide baseline stereo results obtained by the proposed method. Figure 4 shows sample epipolar lines for the already mentioned JANKO pair.

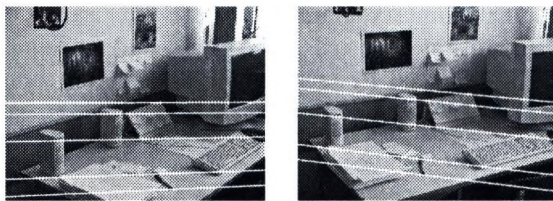


Fig. 4. Sample epipolar lines in JANKO pair.

The well-known VALBONNE pair (figure 5) was also successfully processed. Note that two of the PDRs are essentially the same brick wall texture. To find the proper correspondences, one has to test two alternative variants.

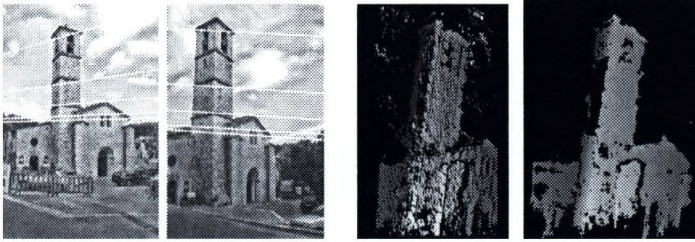


Fig. 5. Epipolar lines in VALBONNE pair and disparity maps obtained by a conventional stereo matching method [6] and by our method.

The disparity maps were calculated with the proposed affine matching method. No post-processing was done; in particular, no effort was made to close the gaps, where the depth measurement was impossible because of rejection or lack of texture. For visualisation of the projective reconstruction, disparities were interpreted as 3D depths.

Figure 6 the disparity map obtained for the JANKO data. The projectively reconstructed surface is shown from different viewpoints.

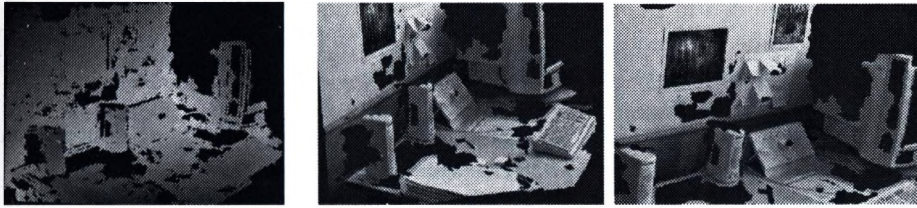


Fig. 6. Disparity map and reconstructed surface for JANKO data.

Finally, figures 7 and 8 show disparity map and reconstructed surfaces obtained with the proposed method for outdoor images with numerous occlusions, shadows and varying zoom.

5 Conclusion

It is obvious that the PDR method can only be applied in the presence of distinct periodic structures, and that PDRs provide constraints rather than precise correspondences. The success of the method depends on whether the assumptions of partial affine alignment are met; if there are few feature points in the roughly aligned area, the method might fail.

The affine propagation algorithm for dense matching preserves depth discontinuities and works well for scenes with large textured, piecewise-smooth surfaces in presence of fine details. It is less efficient for natural scenes with trees or other

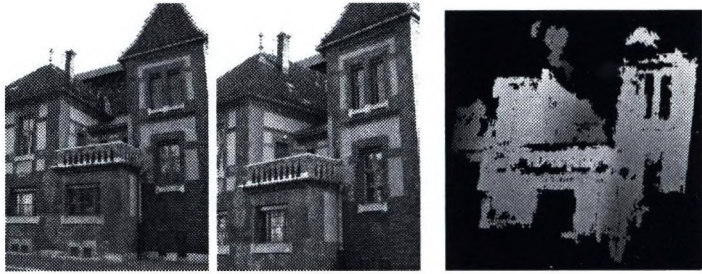


Fig. 7. Stereo pair WORKSHOP and disparity map for rectified images.

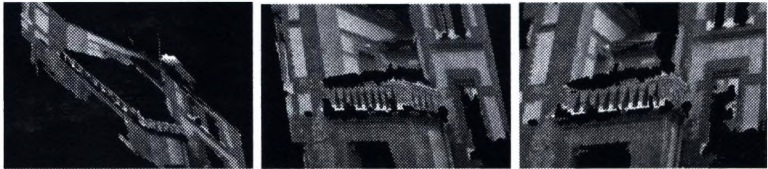


Fig. 8. Reconstructed surface of figure 7 shown from different viewpoints.

non-smooth surfaces. Currently, no treatment of occlusions and non-textured regions is included in the method. Indexing techniques will be applied to implement fast affine parameter search. Future work will also include integration of methods for handling occlusion.

Acknowledgement. This work was supported by the Hungarian Scientific Research Fund (OTKA) grants T038355 and M28078.

References

1. Z. Megyesi and D. Chetverikov. Affine dense matching for wide baseline stereo. In *Proc. Grafika 2003*, pages 109–114, 2003.
2. S. Birchfield. *KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker*. Stanford University, <http://robotics.stanford.edu/~birch/klt/>.
3. D. Chetverikov. Pattern regularity as a visual key. *Image and Vision Computing*, 18:975–985, 2000.
4. D. Chetverikov and J. Matas. Periodic Textures as Distinguished Regions for Wide-Baseline Stereo Correspondence. In *Proc. 2nd Int. workshop on texture analysis and synthesis Texture'2002*, pages 25–30, 2002. Available at <http://www.cse.hk.ac.uk/texture2002/>.
5. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
6. H. Hirschmüller. Improvements in real-time correlation-based stereo vision. In *Proc. IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 141–148, 2001.
7. G.P. Otto and T.K. Chau. A region-growing algorithm for matching of terrain images. *Image and Vision Computing*, 7:83–94, 1989.
8. I. Pitas. *Digital Image Processing Algorithms*. Prentice Hall, 1993.

Számítógépes képfeldolgozás oktatása a Szegedi Tudományegyetemen

Nagy Antal, Balogh Emese, Dudásné Nagy Marianna,
Kuba Attila, Máté Eörs, Palágyi Kálmán, Katona Endre,
Kató Zoltán, Nyúl László, Tanács Attila

SZTE TTK, Képfeldolgozás és Számítógépes Grafika Tanszék
{nagya, bmse, marcsi, kuba, mate, palagyi,
katona, kato, nyul, tanacs}@inf.u-szeged.hu

Kivonat Az SZTE Informatikai Tanszékcsoportja által gondozott szakok tanterveiben 1993 óta szerepel a képfeldolgozás és alkalmazásainak oktatása. A kreditrendszer bevezetésével a Képfeldolgozás I. tárgy kötelező az ötéves képzésben részt vevő informatikus hallgatóknak. Ezen felül a választható szakirányok között szintén szerepel a *Képfeldolgozás szakirány*. A szakirányon belül különböző képfeldolgozási területeket tárgyaló kurzusok épülnek egymásra. Az elméleti megalapozás mellett a képfeldolgozás alkalmazásaira is nagy hangsúlyt fektetünk. A kutatások illetve az orvosi alkalmazások fejlesztése során szerzett eredményeket a kötelező jellegű tárgyak mellett speciálkollégiumok keretében építjük be az oktatási anyagba. Számos hallgatónk választ a képfeldolgozás területéről témát a diplomamunkájához, dolgozataikkal rendszeresen és sikerrel szerepelnek az OTDK-n. Hallgatóink évente több hónapot tölthetnek külföldi partneregyetemeken, ahol a kutató- és fejlesztőmunka mellett nálunk is elfogadott kurzusokat teljesíthetnek. A képfeldolgozás témakörön belül „ipari” projekt munkákban is egyre több hallgató vesz részt. A doktori programon belül is meghirdetünk képfeldolgozáshoz kapcsolódó kutatási irányokat. Az évente megrendezésre kerülő, 11-éves múlttra visszatekintő Képfeldolgozó Nyári Iskolának (SSIP) eddig hatszor adott otthont Szeged. A rendezvénysorozat kiemelkedő fontosságú nemzetközi fórum hallgatóink és oktatóink számára is.

1. Főiskolai alapképzés

A főiskolai 3 éves képzésen a *Programozó matematikus* hallgatók vesznek részt. Ebben a képzési módban a *Számítógépes grafika* tantárgyat oktatjuk a hallgatók számára. A számítógépes grafika tematikája J.D. Foley: Computer Graphics, Principles and Practice [1] könyv fejezeteire épül. A gyakorlatokon a Linux-os környezetben OpenGL [6] felhasználásával sajátíthatják el hallgatók az elméleti anyagot. Az adott félév második felében pedig a Postscript [8] nyelvvel ismerkedhetnek meg. Mindkét témakörhöz oktatási segédanyag (példatár) kapcsolódik, amit a hallgatók interneten keresztül érhetnek el.

2. Egyetemi alapképzés

Az 5 éves egyetemi szintű képzésbe az *Informatika tanár, Közgazdasági programozó matematikus, Műszaki informatikus, Programtervező matematikus* szakok tartoznak.

A hallgatók az alábbi *törzstárgyakkal* találkoznak a tanulmányaik során: *Számítógépes grafika*, *Képfeldolgozás I.*, *Multimédia*. A *Számítógépes grafika* tantárgy mindenkinek kötelező. Az Informatika tanárszakon a *Multimédia* a többi szakon pedig a *Képfeldolgozás I.* kurzust kell kötelező jelleggel teljesíteniük.

A *Képfeldolgozás I.* tárgyon (Gonzalez-Woods könyv alapján [2]) belül a következő témakörökkel ismerkedhetnek meg a hallgatók: a képfeldolgozás tárgya és története, képek reprezentációja, digitalizálás, pontoperációk, képek helyreállítása, szűrése, kódolása, szegmentálása. A gyakorlatokon a Khoros [7] rendszerben valósítanak meg különböző feladatokat a hallgatók.

A *Multimédia* tantárgy keretein belül ismertetésre kerülnek a multimédiás eszközök, a digitalizálás műveletei, az állókép, hang, animáció és videó formátumok, adathordozók, WEB-es lehetőségek valamint az interaktív anyagok elkészítésének technikái. A multimédia tantárgyhoz tartozó gyakorlaton Flash [3] programozással foglalkoznak a hallgatók.

A *Képfeldolgozás szakirányt* vállalasztó hallgatók *Multimédia, Képfeldolgozás II., Orvosi képfeldolgozás, Térinformatika* és a *Grafikai rendszerek* tantárgyak közül kell választaniuk a képzési idejük alatt.

A *Képfeldolgozás II.* tárgy a *Képfeldolgozás I.*-re épül (csak azok a hallgatók vehetik fel, akik teljesítették a *Képfeldolgozás I.-et*).

Az előadás tematikája tág témaköröket ölel fel, melyek mindegyikében igyekszünk a legfejlettebb módszereket bemutatni. A tematika: éldetektálás (beleértve a Canny-féle éldetektort is), szegmentálás (benne a watershed módszert és a fuzzy objektumokat is), alakjellemezés (különös tekintettel a vázkijelölés különféle módszereire), textúraelemzés, matematikai morfológia (bináris és többszintű képekre egyaránt), képregisztrációs technikák (különös tekintettel a 3D orvosi képek illesztésére és fúziójára), leképezések interpolációs technikái, warping és morphing. A gyakorlatokon a hallgatók számítógépes kabinetekben, Linux-os PC-ken képfeldolgozó eljárásokat implementálnak, a gyakorlatot összetett alkalmazási feladat megoldásával teljesíthetik.

Az *Orvosi képfeldolgozás* tantárgy a *Képfeldolgozás I.*-re alapozva a különböző orvosi képalkotó eljárásokat (röntgen, CT, nukleáris medicina, SPECT, PET, ultrahang, doppler UH, mágneses rezonancia) tárgyalja. Ismerteti a különböző leképezési módok fizikai alapjait. Foglalkozik a képalkotás minőség ellenőrzésével és minőség biztosításával (kalibrációk és korrekciók). Bemutatja a klinikumban leggyakrabban alkalmazott kép és kép sorozat feldolgozási technikákat. Gyakorlatokon az előadás anyagához kapcsolódó részfeladatokat oldanak meg a hallgatók C vagy C++ nyelven.

A *Térinformatika* tantárgyat programtervező matematikus szakon a Képfeldolgozás és a Szoftverfejlesztés szakirányok keretében, valamint geográfus szakon a Geoinformatika szakirány keretében oktatjuk. A hallgatók megismerkednek a földrajzi információs rendszerek (Geographic Information Systems) alapfogalmával, majd a két alapvető terület, a raszteres és vektoros térinformatika részletes tárgyalása következik. Gyakorlaton a hallgatók a CAD rendszerként és térinformatikai szoftverként egyaránt széles körben alkalmazott MicroStation [10] rendszer használatával ismerkednek meg. A hallgatóknak egy alkalmazást kell készíteniük, amely grafikus adatokat (térkép, műszaki rajz, stb.) és relációs adatbázist együtt, integráltan képes kezelni.

A *Grafikai rendszerek* tantárgy négy éve került bevezetésre. A cél az volt, hogy a olyan ismereteket adjunk át, amelyeket gyakorlati segítséget adnak a későbbi grafikai alkalmazások fejlesztésekor. A tantárgy első részében az VRML-lel [4] ismerkednek meg a hallgatók. A gyakorlatokon, az eladáson előre kiadott kötelező feladatok megoldásakor felmerülő problémákkal foglalkozunk. Ezek a kötelező feladatok egy-egy komplex virtuális világ megvalósítása (pl. szegedi közlekedési csomópontok). A kurzus második felében főleg a Tcl/Tk-val [5] foglalkozunk. A kurzus végén a PDF-et [9] mutatjuk be a hallgatóknak. Az adott tárgy speciálkollégiumként is felvehető azok számára, akik nem a Képfeldolgozás szakirányt választották.

3. Speciálkollégiumok

A kötelező tárgyakat rendszerint jó néhány speciálkollégium egészíti ki, amelyeken friss kutatási területeket és új fejlesztési eszközöket igyekszünk bemutatni.

A *Digitális topológia és matematikai morfológia* kurzus első része a leggyakoribb 2D és 3D diszkrét képrácson értelmezett bináris képekkel és képműveletekkel foglalkozik. Tárgyaljuk a szekvenciális és párhuzamos képműveletekre a topológia-megőrzését. A matematikai morfológia eszköztára egyaránt hasznos a bináris és többszintű képek analízise során. Tárgyaljuk a legfontosabb morfológiai alpműveleteket (pl. erózió, dilatáció, hit-or-miss transzformáció) és számos összetett műveletet is (morfológiai vékonyítás, vázkijelölés, konvex burok, váztisztítás, ...).

A *Vázkijelölés diszkrét tereken* kurzus tárgyalja a váz különféle meghatározásait, tulajdonságait és foglalkozik a legfontosabb vázkijelölő módszerekkel.

Az *Orvosi képfeldolgozás az izotópdiaosztikában* tárgy célja az volt, hogy a hallgatók az elméleti alapok elsajátítása után találkozhassanak a képfeldolgozás konkrét alkalmazásával is egy adott képmodalitáshoz kapcsolva. Klinikai intézetlátogatás keretén belül láthassák az eszközök működését felvétel és kiértékelés közben. Orvosi megközelítésből is halljanak a témáról, ehhez az izotópiintézetből kértünk fel orvost előadónak.

Az *MRI képek feldolgozása* speciálkollégiumon belül a mágneses rezonanciás (MR) képalkotás fizikai hátterének rövid ismeretete után átismételjük az alapvető képfeldolgozási műveleteket, és konkrét alkalmazásait speciálisan MR

képekre. Az előfeldolgozó műveletek közül a képszegmentálási módszereket osztályozzuk, példáulk illusztráljuk és a fuzzy összefüggőségen alapuló képszegmentálási módszert tárgyaljuk részletesen.

A *Képregisztráció és orvosi alkalmazásai* speciálkollégium a digitális képfeldolgozás napjainkban felmerült problémájával, mely kettő, egymástól függetlenül felvett kép közötti geometriai viszony meghatározásával foglalkozik. Ezen képek felvétele különböző időpontokban, különböző nézőpontokból vagy akár különböző képképző berendezésekkel is történhet. A regisztrációs problémával számos alkalmazási területen találkozunk, közülük az orvosi képek illesztése az egyik legjelentősebb. A betegekről különböző időpontokban, irányokból vagy módokon készült felvételek illesztése megbízhatóbb diagnosztikai információkat biztosít, segíti a képmegértést és javítja a műtét- és a terápia tervezését, valamint kiértékelését.

A *Digitális képek szegmentálása* a képfeldolgozás egyik legfontosabb alapproblémája, mely a hasonló tulajdonságú pixelek homogén területekbe történő csoportosításával foglalkozik. Sok képfeldolgozási feladat (például alakfelismerés, felület leírás, sztereó látás, stb...) kiindulási adata ilyen feldolgozott, szegmentált képből áll. Az alkalmazott eljárások sokfélék: megtaláljuk köztük a digitális szűrők, matematikai morfológia, statisztika, valószínűség számítás, döntésmélt eszközeit használó megoldásokat. A probléma igazi nehézsége azonban abban rejlik, hogy maga a szegmentálási folyamat (mint az emberi látás része) szubjektív és függ a képről meglévő előzetes ismereteinktől is.

Az *OpenGL II.* tantárgy a *Számítógépes grafika* kurzus OpenGL részének a folytatása. A kurzus keretein belül foglalkozunk a textúrázásával, display listákkal, vertex tömbökkel, különböző vizuális hatásokkal, pufferekkel is. Külön kitérünk az OpenGL képfeldolgozás kiterjesztésre, ami kapcsolódik a *Képfeldolgozás I.* tantárgyhoz.

4. Oktatási eszközök

Az oktatás során különböző eszközöket használunk. Ezeket úgy választottuk ki, hogy a hallgatók minnél gyorsabban érthessenek el eredményeket a gyakorlatokon.

A *Khoros* rendszer több komponensből épül fel. A vizuális programozási felület segítségével nagyon könnyen lehet bemutatni az alap képfeldolgozási műveleteket. Továbbá alkalmas arra is, hogy a fejlesztői környezet segítségével a felhasználó által elkészített új rutinokat is be lehet illeszteni az adott folyamatba. Nagy előnye, hogy az általunk használt verziója ingyenes.

A *Postscript és PDF*: a *Postscript* lepleíró nyelv, melynek segítségével 2D-s képek állíthatók elő. A nyelv postfix alakú parancsai segítségével különböző primitíveket lehet elhelyezni az adott lapon, amelyek később vághatóak, kitölthetők stb. A nyelv bemutatására a ghostscript programot (interpretert) használjuk. A *PDF* egy jól struktúrált dokumentum formátum, amely segítségével dokumentumokat lehet különböző rendszerek között továbbítani.

Az *OpenGL* a legjobban elterjedt függvénykönyvtár 3D-s modell-tér létrehozására. Segítségével könnyedén elő lehet állítani egy 3D-s objektumot, megfelelő megvilágítással, textúrázással. A gyakorlatokon C-s környezetben használjuk a függvényeket.

A *Flash* egy multimédiás grafikus program, amit speciálisan WEB-es környezetben lehet használni. Segítségével interaktív „mozikat” lehet készíteni. A *Flash* vektor grafikát használ ezért könnyedén skálázhatóak. Nem igényel nagy programozási ismereteket. Ezzel az eszközzel a hallgatók, csoportmunkában játékokat programoznak le a gyakorlatok keretein belül.

A *VRML* (Virtual Reality Modeling Language) segítségével programozási ismeretek nélkül lehet a megfelelő csomópontok ismeretével, előállítani egy 3D-s világot. Ebben a virtuális világban különböző események hatására folyamatok indulhatnak el és állhatnak meg. Emellett hanghatásokat is elhelyezhetünk és videót is levetíthetünk. Ez a világ színesíthető Java-s utasításokkal is. Különböző megjelenítőket használunk a gyakorlatokon, amelyek segítségével navigálhatunk ezekben a virtuális világokban.

Az Informatikai Tanszékcsoport saját *Oktatási kabinetekkel* rendelkezik. A kabinetek kialakításakor figyelembe vették a különböző igényeket. A windows-os és a linux-os termek rendelkeznek a szükséges hardware-s és software-s környezettel.

5. Vizsgadolgozatok, diplomamunkák és Tudományos Diákköri Konferenciák

A hallgatók közül évről évre többen választják a grafika és képfeldolgozás témakörébe eső szakdolgozat, vizsgadolgozat, diplomamunka témákat.

Tehetséges hallgatónk rendszeresen eredményesen vesznek részt a helyi és országos Tudományos Diákköri Konferenciákon is.

6. Doktori (Ph.D.) program

A *Matematika és Számítástudományok Doktori Iskola, Informatika Doktori Program* beindulásával a képfeldolgozásban is meghirdettünk kutatási irányokat. Így évente 1-2 olyan Ph.D. hallgató nyer a doktori programunkba felvételt, akik a képfeldolgozás területén szeretnének új eredményeket elérni.

7. Képfeldolgozó Nyári Iskola

Rendszeresen megrendezzük a *nemzetközi Képfeldolgozó Nyári Iskolát* (SSIP), ahol nem csak Magyarországról, hanem a környező országokból fogadunk hallgatókat és oktatókat. Ezen a rendezvényen délelőttönként angol előadások hangzanak el a képfeldolgozást oktató hazai és külföldi vendégektől. Délutánonként

nemzetközi csapatokban kell a hallgatóknak az általuk kiválasztott feladatot megoldani. A csapatok munkáját a nyári iskola végén a zsűri értékeli és díjazza. Az iskola végén a hallgatók tesztet írnak, melynek eredményétől függően oklevelet adományozunk. Az SSIP-t adott egyetemek doktori programjai Ph.D. kurzusként fogadják el.

8. Kapcsolat az iparral

Az utóbbi időben sikerült a képfeldolgozásban érdekelt ipari cégekkel felvenni a kapcsolatot. Ezek révén évente 6-8 olyan hallgatónk vesz részt ilyen projekteken. Ezen projektek eredményeit közvetlenül az iparban használják fel.

9. Összefoglalás

A számítógépes képfeldolgozás az oktatásban egyre fontosabb szerepet tölt be. Ezt mutatja a *Képfeldolgozás szakirány*-ra, speciálkollégiumokra illetve a Képfeldolgozó Nyári Iskolára jelentkező hallgatók érdeklődése, a végzős hallgatók által elkészített szakdolgozatok száma, a Ph.D.-s hallgatók munkája is. Természetesen ehhez a fejlődéshez hozzájárult a megfelelő oktatási eszközök kiválasztása illetve beszerzése. Terveink között szerepel az oktatás korszerűsítése (tematika és hardver) illetve szorosabb kapcsolat ipari partnerekkel.

Hivatkozások

1. James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes: Computer Graphics Principles and Practice Addison-Wesley Publishing Company (1991)
2. Rafael C. Gonzalez, Richard E. Woods: Digital Image Processing Prentice Hall (2002)
3. <http://www.macromedia.com>
4. Andrea L. Ames, David R. Nadeau, John L. Moreland: VRML 2.0 Sourcebook John Wiley & Sons, Inc. (1997)
5. Brent B. Welch: Practical Programing in Tcl & Tk Prentice Hall PTR (1997)
6. Richard s. Wright, Jr., Michael Sweet: OpenGL[®] SuperBible Waite Group Press (1999)
7. <http://www.khoral.com>
8. <http://www.adobe.com/products/postscript/main.html>
9. <http://www.adobe.com/products/acrobat/main.html>
10. <http://www.bentley.com/>

Szomszédsági szekvenciák a háromszögrácson

Nagy Benedek

Debreceni Egyetem
nbenedek@inf.unideb.hu

Kivonat A háromszögrácsot, a szimmetriát megőrizve három koordinátával írjuk le. A bevezetett koordináta-rendszer és a három lehetséges szomszédsági viszony összeillik, vagyis a koordináták segítségével ugyanazokat a szomszédsági relációkat definiálhatjuk. Bevezetjük a szomszédsági szekvenciákat (B), értelmezzük a segítségükkel kapott utakat és távolságokat. Algoritmust mutatunk egy legrövidebb B-út előállítására bármely adott ponttól egy adott másik pontig. A háromszögrácson előfordulnak nem-szimmetrikus B-távolságok is (a háromszögegyenlőtlenség sérülésétől függetlenül). Szükséges és elégséges feltételt mutatunk a B szomszédsági szekvenciákra ahhoz, hogy a B-távolság metrika legyen, valamint képletet adunk, amellyel bármely pontok B-távolságai meghatározhatóak. A szomszédsági szekvenciákkal terjedő jelek hullámfrontjait is vizsgáljuk, összehasonlítva a négyzet- és a háromszögrács digitális köreit. A háromszögrácson nyolcféle alakú digiáltlis kör alakulhat ki, amiket lefrunk.

1. Bevezetés

A háromszögrácsokat az utóbbi időben egyre több helyen alkalmazzák a gyakorlatban (geometriai modellezés, 3D szkennelés). A háromszögrácson a három alapvető szomszédsági viszony már [2]-ben megtalálható (Fig. 1, bal oldal). A szomszédsági sorozatokkal definiált távolság a négyzetrácsokra már a 80-as évektől ismert [1,3,9]. Háromszögrácsra a [4] és [5] cikkeinkben vezettük be. Eredményeinket itt is összefoglaljuk: algoritmust adunk tetszőleges két pont közti legrövidebb út meghatározására, illetve megadjuk a szükséges és elégséges feltételét annak, hogy egy tetszőleges szomszédsági sorozattal definiált távolságfüggvény mikor határoz meg metrikus teret. A háromszögrácson több olyan érdekes jelenség is megfigyelhető amely a négyzetes rácsokon nem tapasztalható. Ezeket a jelenségeket, mint a nem szimmetrikus távolságfüggvény vagy az egymást 'megelőző' sorozatok külön is megvizsgáljuk. Az utolsó fejezet a digitális körökről szól.

A cikk a [8] disszertáció alapján készült, annak a háromszögrácsra vonatkozó fejezetét foglalja össze.

2. A digitális háromszögrács - jelölések, definíciók

A dolgozatban a távolság fogalma központi szerepet játszik. Általában a metrikus tulajdonságokat teljesítő függvényeket szokás "j0" távolságoknak tekinteni. Egy

d távolságfüggvényről akkor mondjuk, hogy 'metrika' tulajdonságú, ha bármely p, q, r pontok esetén teljesülnek a következő feltételek:

1. $d(p, q) = 0$, akkor és csak akkor, ha $p = q$
2. $d(p, q) = d(q, p)$ (szimmetria),
3. $d(p, q) + d(q, r) \geq d(p, r)$ (háromszög-egyenlőtlenség).

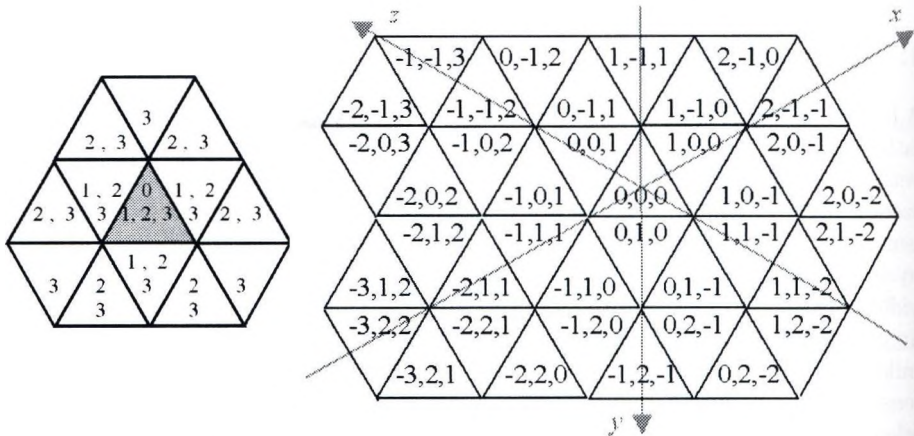
A koordinátákat az 1. ábrán látható módon vezetjük be. Ekkor matematikailag a következő definíció alapján definiálhatjuk a szomszédsági relációkat.

1. definíció. Legyen $p(p(1), p(2), p(3))$ és $q(q(1), q(2), q(3))$ két adott pont a háromszögrácson és $k \in \mathbb{N}, k \leq 3$. A p és a q pontok ' k -szomszédok', ha teljesülnek a következők:

1. $|p(i) - q(i)| \leq 1$ minden $1 \leq i \leq 3$,
2. $\sum_{i=1}^3 |p(i) - q(i)| \leq k$.

Ha a második feltételben egyenlőség áll fenn, akkor azt mondjuk, hogy a p és q pontok 'szigorúan' k -szomszédok.

(Az 1. ábrán a bal oldalon minden mezőben a beleírt legkisebb érték jelzi a szigorú szomszédság fokát.)



1. ábra. Szomszédsági típusok és koordináták a háromszögrácson

2. definíció. A $B = (b(i) : i \in \mathbb{N})$ sorozatot, ahol $b(i) \in \{1, 2, 3\}, \forall i$ 'szomszédsági sorozatnak' hívjuk. Ha van olyan $l \in \mathbb{N}$, hogy $b(i) = b(i + l)$ fennáll minden $i \in \mathbb{N}$ esetén, akkor B -t periódikusnak mondjuk, l periódussal. Ebben az esetben egy periódusnyi elemmel jellemezhetjük a sorozatot: $B = (b(1), \dots, b(l))$. Legyen p és q két pont és $B = (b(i) : i \in \mathbb{N})$ egy szomszédsági sorozat. Egy véges

$\Pi(p, q; B)$ pontsorozatot – ami $p = p_0, p_1, \dots, p_m = q$ formába írható, ahol p_{i-1} és p_i $b(i)$ -szomszédok minden $1 \leq i \leq m$ esetén – egy a B által meghatározott p -ből q -ba vezető útnak (B -út) hívunk. A $\Pi(p, q; B)$ út hossza $|\Pi(p, q; B)| = m$. A p -ből q -ba vezető legrövidebb utat (vagy ezek egyikét) jelölje $\Pi^*(p, q; B)$. Ekkor p és q B -távolságát definiáljuk ezen út hosszával: $d(p, q; B) = |\Pi^*(p, q; B)|$.

Rögzítsünk le egy koordinátát. Azon pontok, amelyeknek ez a koordinátája megegyezik ezzel a fix értékkel egy 'sávot' alkotnak.

A háromszögrács pontjai egy-egyértelműen azonosíthatók az olyan koordinátaérték-hármasokkal, amelyekre az értékek összege 0 vagy 1. Így a háromszögrács pontjainak megfelelő koordináták a kockarácsonban két szomszédos ferdesíkot határoznak meg.

A 0 koordinátaösszegű pontok 'paritása páros'. Azokat a pontokat pedig, amelyeknek a koordináta-összege 1 'páratlannak' hívjuk.

Ha két pont 1-szomszéd, akkor paritásuk különböző, és két különböző sáv is tartalmazza mindkettőjüket. Két szigorúan 2-szomszédos paritása megegyezik, és pontosan egy sávra igaz, hogy mindkettőt tartalmazza. A szigorúan 3-szomszédos paritása különbözik, és nincs olyan sáv, amely mindkettőt tartalmazza.

3. definíció. Legyen p és q két pont. A $w_{p,q}$ vektort az p és a q pont 'különbségvektorának' hívjuk ha $w(i) = q(i) - p(i)$. A $w_{p,q}$ -nak mint multihalmaznak az elemeit abszolútértékük nagysága szerint nem csökkenő sorrendbe szedve kapjuk a $v_{p,q}$ vektort. Ekkor a $v_{p,q}$ multihalmazként tekintve megegyezik a $w_{p,q}$ -val, ráadásul $|v(i)| \geq |v(j)|$ ha $i < j$ is fennáll ($1 \leq i, j \leq 3$). A $v_{p,q}$ vektort 'rendezett különbségvektornak' nevezzük. Amikor az egyértelműséget nem veszélyeztetni el fogjuk hagyni a p, q indexpárt.

3. Legrövidebb út

1. algoritmus

Input: a kezdő- és a végpont (p, q) , valamint egy B szomszédási sorozat.

- 1. lépés: Legyen w a 2 pont különbségvektora, valamint $\Pi = (p)$, $x_0 = p$ és $j = 0$.
- 2. lépés: Ha minden $w(i) = 0$ akkor kész és menjünk a 11. lépésre.
- 3. lépés: Legyen $j = j + 1$. Legyen h_i ($i = 1, 2, 3$) a $(1, 2, 3)$ értékek egy olyan permutációja, amire $|w(h_1)| \geq |w(h_2)| \geq |w(h_3)|$, és $\text{sgn}(w(h_1)) \neq \text{sgn}(w(h_2))$.
- 4. lépés: Ha $b(j) = 1$ akkor attól függően, hogy x_{j-1} páros vagy páratlan, csökkentsük abszolút-értékben eggyel a pozitívát, illetve a negatívát a $w(h_1)$ és a $w(h_2)$ közül. Ezután a 8. lépés következik.
- 5. lépés: Ha $b(j) = 2$ akkor csökkentsük abszolút-értékben eggyel a $w(h_1)$ és a $w(h_2)$ közül az(oka)t, amely(ek) nem 0(-k). Ezután a 8. lépés következik.
- 6. lépés: Ha x_{j-1} páros és w -ben két pozitív érték szerepel, akkor legyen $w(i) = \text{sgn}(w(i))|w(i) - 1|$ ($i = 1, 2, 3$), különben pedig (ha x_{j-1} páros és w -ben nincs két pozitív érték) legyen $w(h_1) = \text{sgn}(w(h_1))|w(h_1) - 1|$ és $w(h_2) = \text{sgn}(w(h_2))|w(h_2) - 1|$.

- 7. lépés: Ha x_{j-1} páratlan: ha w -ben két negatív érték szerepel, akkor legyen $w(i) = \operatorname{sgn}(w(i))|w(i) - 1|$ ($i = 1, 2, 3$), különben pedig (ha x_{j-1} páratlan, de w -ben nincs két negatív érték) legyen $w(h_1) = \operatorname{sgn}(w(h_1))|w(h_1) - 1|$ és $w(h_2) = \operatorname{sgn}(w(h_2))|w(h_2) - 1|$.
- 8. lépés: Az x_j pont koordinátái legyenek: $x_j(i) = q(i) - w(i)$, ($1 \leq i \leq 3$).
- 9. lépés: Fűzzük a Π minimális úthoz az x_j pontot.
- 10. lépés: Vissza a 2. lépésre.
- 11. lépés: Output: Π , egy legrövidebb lépésszámú út p -ből q -ba, valamint j az út hossza.

Az 1. algoritmus helyes: mindig megáll és egy legrövidebb utat állít elő. Az algoritmus mohó algoritmus, konstans tár-bonyolultságú, és a pontok koordinátakülönbségösszegével lineáris időbonyolultságú.

Bármely két pont B -távolsága függ a pontok különbségvektorától és paritásától, valamint a szomszédsági sorozattól.

4. A metrikus B -távolság feltétele

A háromszögrácson (a négyzetrácscsal ellentétben) nem csak a háromszögegyenlőtlenséggel, hanem a szimmetriával is gond lehet, így egy új, érdekes helyzet előtt állunk.

1. lemma. *Tetszőleges p és q pontok (1)-távolsága:* $d(p, q; (1)) = |w_{p,q}(1)| + |w_{p,q}(2)| + |w_{p,q}(3)|$.

4. definíció. *Egy B szomszédsági sorozat 'minimális ekvivalens sorozatán' a következő tulajdonságú B' szomszédsági sorozatot értjük.*

1. $d(p, q; B) = d(p, q; B')$ bármely p, q pontpárra; és
2. minden olyan B_1 szomszédsági sorozatra, amire $d(p, q; B) = d(p, q; B_1)$ minden p, q pontpárra, teljesül az, hogy $b'(i) \leq b_1(i)$ minden i -re.

2. lemma. *Egy B szomszédsági sorozat B' minimális ekvivalens sorozata egyértelműen meghatározott:*

- $b'(i) = b(i)$, ha $b(i) < 3$,
- $b'(i) = 3$, ha $b(i) = 3$ és nincs olyan $j < i$ amire $b'(j) = 3$,
- $b'(i) = 3$, ha $b(i) = 3$ és van olyan $b'(l) = 3$, hogy $l < i$, de $\sum_{k=j+1}^{i-1} b'(k)$ páros,
ahol $j = \max \{l | l < i, b'(l) = 3\}$,
- $b'(i) = 2$, különben.

A minimális ekvivalens sorozat intuitíve kb. éppen azt "felügyeli" (erre még a következő részben visszatérünk), hogy a legrövidebb út megkonstruálásakor ne lehessen a kockarác távolabbi pontjait előnyösebben felhasználni annál, mintha a 0- és 1-összegű koordinátaértékekkel rendelkező pontok két ferdesíkjára korlátoznánk a lehetséges mozgást.

3. lemma. Egy B -távolság pontosan akkor nem teljesíti a szimmetria tulajdonságot, ha van olyan $i \in \mathbb{N}$, hogy $b(i) = 3$, és fennáll legalább a következő esetek egyike az $i = \min \{l \mid b(l) = 3\}$ értékkel:

- $\sum_{k=1}^{i-1} b(k)$ páratlan; vagy
- van olyan j , amire $b(j) = 1$ és $i < j$.

4. lemma. Legyen a B szomszédsági sorozat olyan, hogy nem tartalmazza a hármast. A háromszög-egyenlőtlenség pontosan akkor nem teljesül a B -távolságra ha van olyan i és j , hogy $\sum_{k=1}^i b(k) > \sum_{k=j+1}^{j+i} b(k)$.

Ha a B szomszédsági sorozat tartalmazza a 3-at, akkor általában nem egyszerű megmondani mikor teljesíti a B -távolság a háromszög-egyenlőtlenséget. Abban az esetben viszont, ha a B -távolság szimmetrikus a megfelelő B' -ben csak legfeljebb egy 3 szerepelhet. Ekkor könnyű ellenőrizni, hogy az előző lemmában elég az összegzéseket addig vizsgálni, amíg a $j + i$ értéke el nem éri ennek a 3-nak a helyét. Ha a lemma feltétele eddig a pontig fennáll, akkor a generált távolságfüggvényre teljesül a háromszög-egyenlőtlenség. Az előzőek alapján kimondjuk a tételt:

1. tétel. Legyen B egy szomszédsági sorozat. A B -távolság pontosan akkor metrikus, ha teljesülnek a következők:

- ha $b(j) = 3$ és $b(i) = 1$ akkor $i < j$,
- ha a B -ben szerepel a 3, akkor $\sum_{b(k)=1} b(k)$ páros,
- $\sum_{k=1}^i b(k) \leq \sum_{k=j+1}^{j+i} b(k)$, ahol $i + j < l$, arra az l -re ami a B -ben levő első 3 helye, (ha a 3 nincs a B -ben akkor a feltételnek minden $i, j \in \mathbb{N}$ párra fenn kell állnia).

5. Képlet a távolságszámításra

Felhasználva a háromszögrács kockarácsba való beágyazhatóságát a kockarácsra levezetett képlet ([8]) alapján számoljuk ki a távolságot a háromszögrácson.

Mivel - a háromszögrács sajátosságai miatt - egy $b(i) = 3$ -assal nem mindig tudunk "közelebb" kerülni a célponthoz, mint ha csak $b(i) = 2$ lenne, bevezetjük a csökkentett minimális ekvivalens sorozat fogalmát is.

5. definíció. A B'' szomszédsági sorozatot a B csökkentett minimális ekvivalens sorozatának hívjuk, ha

- $b''(k) = 2$, ahol k az első B -ben előforduló 3 helye;
- $b''(i) = b'(i)$, minden más i -re, ahol a $b'(i)$ -k a B minimális ekvivalens sorozatának az elemei.

Egy B szomszédsági sorozat 2-korlátozott sorozatának nevezzük azt a sorozatot, amelyet a B -ből a 3-asok 2-esre cserélésével kapunk: $B^{(2)} = (b^{(2)}(i) : i \in \mathbb{N})$.

2. tétel. Legyen p, q két pont és B egy szomszédsági sorozat. Jelölje d'_2, d'_3, d''_2 és d''_3 azokat az értékeket, amiket rendre a B minimális ekvivalens sorozatának 2-korlátozott sorozatával, a B minimális ekvivalens sorozatával, a csökkentett minimális ekvivalens sorozatának 2-korlátozott sorozatával, illetve a csökkentett minimális ekvivalens sorozatával számolunk a kockarácsban:

$$d'_2 = \max \left(i \left| \sum_{l=1}^2 |v(l)| > \sum_{j=1}^{i-1} b^{(2)}(j) \right. \right), d'_3 = \max \left(i \left| \sum_{l=1}^3 |v(l)| > \sum_{j=1}^{i-1} b'(j) \right. \right),$$

$$d''_2 = \max \left(i \left| \sum_{l=1}^2 |v(l)| > \sum_{j=1}^{i-1} b''(j) \right. \right), d''_3 = \max \left(i \left| \sum_{l=1}^3 |v(l)| > \sum_{j=1}^{i-1} b''(j) \right. \right).$$

Ekkor legyen

$$d' = \max(|v(1)|, d'_2, d'_3) \text{ és } d'' = \max(|v(1)|, d''_2, d''_3).$$

$d(p, q; B) = d''$, ha $d' \geq k$ és az alábbi esetek egyike fennáll:

- p páros, $\sum_{i=1}^{k-1} b(i)$ páros, és $w_{p,q}$ -ben két negatív és egy pozitív érték van;
- p páratlan, $\sum_{i=1}^{k-1} b(i)$ páratlan és $w_{p,q}$ -ben két negatív és egy pozitív érték van;
- p páratlan, $\sum_{i=1}^{k-1} b(i)$ páros és $w_{p,q}$ -ben egy negatív és két pozitív érték van;
- p páros, $\sum_{i=1}^{k-1} b(i)$ páratlan és $w_{p,q}$ -ben egy negatív és két pozitív érték van.

Különben pedig $d(p, q; B) = d'$.

3. tétel. Egy B -távolság függ a B felhasznált elemeinek sorrendjétől, ha a felhasznált elemeknek van olyan permutációja, amellyel a távolság nem szimmetrikus.

6. Digitális körök

6. definíció. Legyen B egy szomszédsági sorozat, és $k \in \mathbb{N}$, továbbá

$$C_k^B = \{p \mid d(o, p; B) \leq k\}.$$

Ekkor C_k^B az o középpontú k sugarú digitális B -kör.

Négyzetrácson az $O_k^B = \{p \mid d(o, r; B) \leq k\}$ jelölést használjuk. A négyzetrácson a különböző szomszéd-sági sorozattal generált, de egyező sugarú digitális körök egy jól rendezett halmazt alkotnak. Ez háromszögrácson nem teljesül. A négyzetrácson O_k^B nem függ a B első k elemének sorrendjétől. Ezzel szemben a háromszögrácson pl. $C_2^{(1,3)}$ és $C_2^{(3,1)}$ két egymással nem összemérhető pontthalmazt jelöl. Ahogyan a négyzetrácson, úgy a háromszögrácson is igaz viszont az, hogy a k sugár növekedtével az ugyanazzal a szomszéd-sági sorozattal generált B -körök szigorúan monoton nőnek, vagyis: ha $k > l$ akkor C_k^B -nek valódi része a C_l^B . A négyzetrácson nem fordulhat elő, hogy két szomszéd-sági sorozatra a különböző sugarú körök megegyezzenek. Ezzel szemben pl. $C_2^{(1)} = C_1^{(2)}$.

Itt jegyezzük meg, hogy adott B szomszéd-sági sorozat és minimális ekvivalens sorozata B' ugyanazt a digitális körsorozatot generálja, vagyis $(C_k^{B'} = C_k^B)$ bármely $k \in \mathbb{N}$ esetén. (Ezzel szemben a négyzetrácson ha egy B_1 sorozat elemenként nem kisebb egy másik B_2 sorozatnál és van olyan $i \in \mathbb{N}$, hogy $b_1(i) > b_2(i)$ akkor minden legalább i sugarú köreikre ($k \geq i$) igaz, hogy $O_k^{B_1}$ nagyobb, mint $O_k^{B_2}$.)

Tanulmányozva az egy pontból induló, szomszéd-sági szekvenciával generált hullámfrontokat láthatjuk, hogy a háromszögrácson a digitális köröknek három fajta oldala és hét-féle csúcsa lehet. (lásd 1. és 2. táblázat).

5. lemma. (az oldalak változásainak összegzése)

- Egy $b(i) = 3$ lépés után a 'sima' és a 'fűrészfog' oldalakból 'sima' oldal lesz.
- A $b(i) = 2$ -es lépések nem változtatnak az oldalak típusán.
- A 'dombos' oldal nem változik meg egyik lépésben sem.

6. lemma. (a csúcsok változásainak összegzése)

- Két 'dombos' él között kétféle csúcs is kialakulhat (a 6-os típus lapos, míg a 7-es csúcsos).
- A következő esetekben egy csúcsból kettő (azonos típusú) lesz, vagyis egy új él is kialakul: két 'fűrészfog' közti csúcsból egy $b(i) = 3$ lépéssel, egy új 'dombos' oldal alakul ki két 'sima' között; két 'dombos' között (a 6-os csúcs-típus esetén) $b(i) = 2$, illetve $b(i) = 3$ esetén egy 'sima' oldal jelenik meg; végül két 'dombos' oldal között (a 7-es esetben) attól függően, hogy $b(i) = 1$ vagy $b(i) = 2$ egy 'sima' vagy egy 'fűrészfog' oldal keletkezik. (A 2. táblázatban ezeket az eseteket jeleztük dupla értékkel).

eredeti oldal	$b(i) = 1$ lépés után	$b(i) = 2$ lépés után	$b(i) = 3$ lépés után
'fűrészfog'	'sima'	'fűrészfog'	'sima'
'dombos'	'dombos'	'dombos'	'dombos'
'sima'	'fűrészfog'	'sima'	'sima'

1. táblázat. A lehetséges oldaltípusok és lépésenkénti változásaik

eredeti csúcs	csúcsok egy $b(i) = 1$ lépés után	csúcsok egy $b(i) = 2$ lépés után	csúcsok egy $b(i) = 3$ lépés után
'sima'-'sima' (1)	'fűrészfog'- 'fűrészfog' (3)	'sima'-'sima' (1)	'sima'-'sima' (1)
'sima'-'fűrészfog' (2)	'sima'-'fűrészfog' (2)	'sima'-'fűrészfog' (2)	'sima'-'sima' (1)
'fűrészfog'- 'fűrészfog' (3)	'sima'-'sima' (1)	'fűrészfog'- 'fűrészfog' (3)	'sima'-'dombos'- 'sima' (4,4)
'sima'-'dombos' (4)	'fűrészfog'-'dombos' (5)	'sima'-'dombos' (4)	'sima'-'dombos' (4)
'fűrészfog'- 'dombos' (5)	'sima'-'dombos' (4)	'fűrészfog'-'dombos' (5)	'sima'-'dombos' (4)
'dombos'-'dombos' (6-os típus)	'dombos'-'dombos' (7-es típus)	'sima'-'dombos'- 'sima' (4,4)	'sima'-'dombos'- 'sima' (4,4)
'dombos'-'dombos' (7-típus)	'sima'-'dombos'- 'sima' (4,4)	'dombos'-'fűrészfog'- 'dombos' (5,5)	'dombos'-'dombos' (6-típus)

2. táblázat. A lehetséges csúcsok és lépésenkénti változásaik

A) háromszög	kiinduló: 0 (maga az eredeti háromszög) vagy az abból egyetlen $b(1) = 1$ -lépéssel kapott kör
B) hatszög – hat 'sima' oldallal	egyetlen felhasznált $b(i) = 3$ -lépés, a többi lehet $b(j) = 1$, illetve $b(j) = 2$ úgy, hogy a $b(i) = 3$ utáni lépésekre a B elemeinek összege (a sugárig) páros
C) hatszög – három 'sima' és három 'fűrészfog' oldallal	csak $b(j) = 1$, illetve $b(j) = 2$ lépések (nem volt $b(j) = 3$ -lépés)
D) hatszög – hat 'fűrészfog' oldallal	egyetlen $b(j) = 3$ -lépés a a többi $b(j) = 1$, illetve $b(j) = 2$, úgy, hogy a $b(i) = 3$ utáni lépésekre a B elemeinek összege (a sugárig) páratlan
E) kilencszög – hat 'dombos' és három 'sima' oldallal	csak $b(i) = 1$ -lépések és $b(j) = 3$ -lépések, felváltva (legalább 2 lépés: mindkét-féle lépés előfordul), és $b(k) = 3$ -lépés az utolsó
F) kilencszög – hat 'dombos' és három 'fűrészfog' oldallal	csak $b(i) = 1$ -lépések és $b(j) = 3$ -lépések, felváltva (legalább 2 lépés: mindkét-féle lépés előfordul), és $b(k) = 1$ -lépés az utolsó
G) tizenkétszög – hat 'dombos' és hat 'sima' oldallal	legalább egy $b(i) = 2$ -lépés vagy két egymás utáni $b(j) = 1$ és $b(j+1) = 1$ -lépések, valamint legalább két $b(h_a) = 3$ -lépés és az utolsó $b(h_a) = 3$ -lépés utáni lépésekre a B elemeinek összege (a sugárig) páros
H) tizenkétszög – hat 'dombos' és hat 'fűrészfog' oldallal	legalább egy $b(i) = 2$ -lépés vagy két egymás utáni $b(j) = 1$ és $b(j+1) = 1$ -lépések, valamint legalább két $b(h_a) = 3$ -lépés és az utolsó $b(h_a) = 3$ -lépés utáni lépésekre a B elemeinek összege (a sugárig) páratlan

3. táblázat. A B -körök osztályozása

A 3. táblázatban nyolc osztályba soroltuk a háromszögrács B -köreit az alakjuk alapján.

7. lemma. Ha egy $p = (p_1, p_2, p_3)$ pont benne van a $C_k^B(0, 0, 0)$ -középpontú körben, akkor a $(p_{i_1}, p_{i_2}, p_{i_3})$ pontok is a kör pontjai, ahol (i_1, i_2, i_3) az $(1, 2, 3)$ tetszőleges permutációja.

4. tétel. A háromszögrácson egy B -kör akkor és csak akkor konvex, ha A vagy B típusú (a 3. táblázatban).

7. Összefoglalás

A digitális háromszögrácsot 3 egészértékű koordinátával írtuk le, ahol a koordinátaösszeg 0 vagy 1 lehet. Megadtunk egy legrövidebb B -utat, illetve a B -távolságo(ka)t két tetszőleges pont között, tetszőleges szomszédsági sorozatot felhasználva. A metrikusság feltételét is megmutattuk. Több érdekes tulajdonság lép fel a háromszögrácson a B -távolságokat használva, ami a négyzetrácsonál nincs jelen. Ezekre is mutattunk példákat, valamint a B -köröket kategorizáltuk.

Köszönetnyilvánítás

Ez a cikk nem jött volna létre, az OTKA 043090 és kutatótársaim segítségével nélkül.

Hivatkozások

1. P. P. Das, P. P. Chakrabarti és B. N. Chatterji: Distance functions in digital geometry, *Inform. Sci.* **42** (1987), 113-136.
2. E. S. Deutsch: Thinning algorithms on rectangular, hexagonal and triangular arrays, *Communications of the ACM*, **15** No.3 (1972) 827-837.
3. A. Fazekas, A. Hajdu és L. Hajdu: Lattice of generalized neighbourhood sequences in nD and ∞D , *Publ. Math. Debrecen* **60** (2002), 405-427.
4. B. Nagy: Shortest Path in Triangular Grids with Neighbourhood Sequences, *Journal of Comp. and Inf. Techn.*, **11** (2003), 111-122.
5. B. Nagy: Metrics Based on Neighbourhood Sequences in Triangular Grids, *Pure Math. Appl.*, **13** (2002), 259-274.
6. B. Nagy: Characterization of Digital Circles in Triangular Grid, *Pattern Recognition Lett.*, közlésre elfogadva.
7. B. Nagy: A Family of Triangular Grids in Digital Geometry, *ISPA'03, 3rd International Symposium on Image and Signal Processing and Analysis*, Rome, Italy, (2003), 101-106.
8. B. Nagy: Neighbourhood sequences in different grids, PhD disszertáció (2003).
9. M. Yamashita és N. Honda: Distance functions defined by variable neighbourhood sequences, *Pattern Recognition* **17** (1984), 509-513.

National Instruments, Hungary

National Instruments

1. About National Instruments

National Instruments (<http://www.ni.com>) is a technology pioneer and leader in virtual instrumentation - a revolutionary concept that has changed the way engineers and scientists approach measurement and automation. Leveraging the PC and its related technologies, virtual instrumentation increases productivity and lowers costs for customers worldwide through easy-to-integrate software, such as the NI LabVIEW graphical development environment, and modular hardware, such as PXI modules for data acquisition, instrument control and machine vision. Headquartered in Austin, Texas, NI has more than 3,000 employees and direct operations in 40 countries. In 2002, the company sold products to more than 25,000 different companies in more than 80 countries around the world. For the past five consecutive years, FORTUNE magazine has named NI one of the 100 best companies to work for in America.

2. Virtual instrumentation

A virtual instrument consists of an industry-standard computer or workstation equipped with powerful application software, cost-effective hardware such as plug-in boards, and driver software, which together perform the functions of traditional instruments. Virtual instruments represent a fundamental shift from traditional hardware-centered instrumentation systems to software-centered systems that exploit the computing power, productivity, display, and connectivity capabilities of popular desktop computers and workstations. Although the PC and integrated circuit technology have experienced significant advances in the last two decades, it is software that truly provides the leverage to build on this powerful hardware foundation to create virtual instruments, providing better ways to innovate and significantly reduce cost. With virtual instruments, engineers and scientists build measurement and automation systems that suit their needs exactly (user-defined) instead of being limited by traditional fixed-function instruments (vendor-defined).

This document describes powerful programming tools, flexible acquisition hardware, and the personal computer, which are the essential components for virtual instrumentation. The synergy between them offers advantages that cannot be matched by traditional instrumentation.

3. Virtual Instruments versus Traditional Instruments

Stand-alone traditional instruments such as oscilloscopes and waveform generators are very powerful, expensive, and designed to perform one or more specific

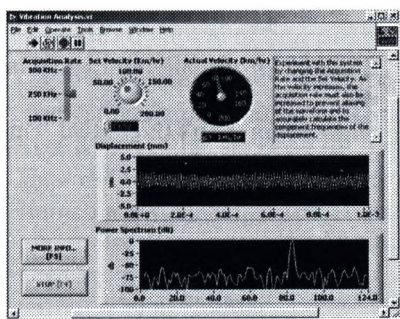
tasks defined by the vendor. However, the user generally cannot extend or customize them. The knobs and buttons on the instrument, the built-in circuitry, and the functions available to the user, are all specific to the nature of the instrument. In addition, special technology and costly components must be developed to build these instruments, making them very expensive and slow to adapt.

Virtual instruments, by virtue of being PC-based, inherently take advantage of the benefits from the latest technology incorporated into off-the-shelf PCs. These advances in technology and performance, which are quickly closing the gap between stand-alone instruments and PCs, include powerful processors such as the Pentium 4 and operating systems and technologies such as Microsoft Windows XP, .NET, and Apple Mac OS X. In addition to incorporating powerful features, these platforms also offer easy access to powerful tools such as the Internet. Traditional instruments also frequently lack portability, whereas virtual instruments running on notebooks automatically incorporate their portable nature.

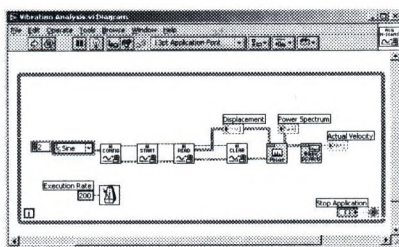
Engineers and scientists whose needs, applications, and requirements change very quickly, need flexibility to create their own solutions. You can adapt a virtual instrument to your particular needs without having to replace the entire device because of the application software installed on the PC and the wide range of available plug-in hardware.

4. Graphical Programming

One of the most powerful features that LabVIEW offers engineers and scientists is its graphical programming environment. With LabVIEW, you can design custom virtual instruments by creating a graphical user interface on the computer screen.



(a)



(b)

You determine the behavior of the virtual instruments by connecting icons together to create block diagrams, which are natural design notations for sci-

entists and engineers. With graphical programming, you can develop systems more rapidly than with conventional programming languages, while retaining the power and flexibility needed to create a variety of applications.

5. Virtual Instruments beyond the Personal Computer

Recently, commercial PC technologies have begun migrating into embedded systems. Examples include Windows CE, Intel x86-based processors, PCI and CompactPCI buses, and Ethernet for embedded development. Because virtual instrumentation relies so heavily on commercial technologies for cost and performance advantages, it also has expanded to encompass more embedded and real-time capabilities. For example, LabVIEW runs on Linux as well as the embedded ETS real-time operating system from VenturCom on specific embedded targets. The option of using virtual instrumentation as a scalable framework that extends from the desktop to embedded devices should be considered a tool in the complete toolbox of an embedded systems developer.

LabVIEW Vision Development Module

NI IMAQ Vision

- High-level machine vision, image processing functions, and display tools
- Grayscale, color, and binary image processing and analysis
- High-speed grayscale and color pattern matching for locating objects of various size and orientation, even in poor lighting
- Embedded vision capabilities
- Area, perimeter, location, and up to 49 other parameters calculated with blob analysis
- Lens distortion and camera angle correction
- Support for LabVIEW Real-Time

Vision Assistant

- Interactive vision prototyping environment that generates LabVIEW diagrams
- Offline inspection with automated scripting and batch processing
- Multiple images displayed with zooming, panning, extracting, and scrolling
- Measurements such as distances, areas, and locations returned
- Browser for visually managing images
- Support for LabVIEW Real-Time

6. Conclusion

The virtual instrumentation concepts of integrated software and hardware, flexible modular tools, and the use of commercial technologies combine to create a framework upon which you can rapidly complete your systems development and also maintain them for the long term. Because virtual instrumentation offers so many options and capabilities in embedded development, it makes sense for embedded developers to understand and review these tools. This concept not

only ensures that your work will be usable in the future but also provides the flexibility to adapt and extend as needs change. LabVIEW was designed with scientists and engineers in mind, providing powerful tools and a familiar development environment created specifically for the design of virtual instruments

7. Overview

The National Instruments LabVIEW Vision Development Module is for scientists, engineers, and technicians who are developing LabVIEW machine vision and scientific imaging applications. The module includes Vision Assistant, an interactive environment for developers who need to quickly prototype LabVIEW Vision applications without programming, and NI IMAQ Vision, a library of powerful functions for image processing.

Vision Assistant and IMAQVision work together to simplify vision software development. Vision Builder can automatically generate a LabVIEW block diagram that contains the same functionality as the series of operations you prototyped in Vision Assistant. You can integrate the diagram into your automation or production test application, which may include motion control, instrument control, or data acquisition. In addition, the Vision Development Module contains embedded vision capabilities under LabVIEW Real-Time for ease of use with embedded systems and greater reliability.

For more information about the Vision Development Module and other NI Vision products, ni.com/vision

8. Explore the Possibilities Quickly with Vision Assistant

Building a vision application often involves time-consuming experimentation. Here are some of the benefits of using Vision Assistant to prototype your application before development:

- Get ideas about how to solve your application using the solution wizard
- Test different processing strategies
- Test a particular strategy on a variety of images
- Explore „what-if” conditions quickly and easily
- Immediately visualize the effects of changing an image processing control parameter Benchmark performance of your solution
- Develop a script that you can run in batch mode on hundreds of images within Vision Builder

9. Develop Your Vision Solution with IMAQ Vision

IMAQ Vision is a high-level programming library that includes an extensive set of MMX-optimized functions for machine vision and scientific imaging. Here are just some of the tasks you can perform with IMAQ Vision:

- Calibrate images to take accurate, real-world measurements regardless of camera perspective or lens distortion
- Filter images to improve their quality before inspection
- Search grayscale and color images for instances of a predefined template
- Measure features of a part regardless of its orientation within the image

10. Vision Hardware

- Analog and digital image acquisition devices for PCI and PXI
- Digital devices – for parallel digital, Camera Link, and FireWire (IEEE 1394)
- Factory-calibrated hardware with real-time and embedded machine vision capabilities
- Integration with data acquisition and motion control
- NI-IMAQ driver software simplifies configuration and maintenance
- Analog devices – for monochrome and color; multichannel and nonstandard formats

11. High Performance, Analog, Monochrome NI PCI-1409/NI PXI-1409

- 4-channel monochrome image acquisition board for standard or nonstandard video sources
- 8 or 10-bit digitization
- Compatible with double-speed 60 frames/s progressive scan cameras
- Interlaced/noninterlaced acquisition
- Analog area and line-scan capabilities
- Unlimited acquisition window size with 16 MB of onboard memory

12. Single Channel, Analog, Color NI PCI-1411/NI PXI-1411

- Onboard real-time HSL color conversion for fast color matching
- Partial image acquisition with onboard programmable region of interest
- Onboard pixel decimation
- Programmable gain and offset
- 1-channel color or monochrome image acquisition board
- NTSC, PAL, S-Video, RS-170, or CCIR input

13. 32-bit, Parallel Digital NI PCI-1424

- Image acquisition boards for digital area and line-scan cameras
- Easy-to-use camera configuration utility
- Full 8, 10, 12, 14, 16, 24 and 32-bit resolution (grayscale or color)
- 50 MHz pixel clock rate with up to 200 Mbytes/s acquisition
- RS-422/TTL or LVDS/TTL camera compatibility
- 4 external triggers (digital I/O lines)

14. Compact Vision System

National Instruments image acquisition hardware and software, work in a seamless, integrated fashion with NI data acquisition, motion control, and CAN hardware. NI image acquisition devices for PCI and PXI have the RTSI bus and PXI trigger bus, respectively, which share timing signals among devices to synchronize and correlate images with other measurements. Use machine vision and motion control hardware to guide the placement of components into circuit boards and control microscope stages.

National Instruments CVS-1450 series systems give you flexibility, integration, and ruggedness for all of your inspection, alignment, gauging, and identification applications. A high-performance processor integrated with three FireWire ports means that

NI CVS-1450 systems are equipped to handle any inspection task. A diverse range of digital I/O options means that CVS-1450 can communicate with a wide range of automation devices like PLCs, relays, and robotics.

15. Multicamera Inspection

The NI Compact Vision System provides a low-cost way to inspect from several angles. With three FireWire ports, you can connect up to 15 cameras to the compact vision system with ease. Each camera shares a portion of the 400 Mb/s bandwidth.

16. Choice of Sensor

By using FireWire image acquisition, an NI Compact Vision system gives you the option of choosing the sensor that is right for your application. You can choose a low-cost, low-resolution sensor or a high-performance sensor. In addition, as new improved industrial FireWire cameras enter the marketplace, CVS-1454 systems are ready for them

17. External Device Control

The CVS-1454 has 29 digital I/O lines with built-in functionality for communicating with external devices, such as reading quadrature encoder inputs, generating strobe pulses, and writing to or reading from digital lines. Using these signals, you can dynamically control your lighting or cameras, synchronize with a conveyor belt, or communicate with relays that control solenoids and other actuators. Digital inputs and outputs are available for both 24 V and TTL-signaling levels. The CVS-1454 has 15 digital input lines – 13 isolated 24 V lines and two dedicated TTL lines. There are also 14 digital output lines – four 24 V and 10 dedicated TTL. In addition, CVS-1450 systems can send commands and data to other devices, such as PLCs, via Ethernet and RS-232 Serial.

18. Real-Time Display

Using the VGA output, you can see the product under inspection in real time, as well as pass/fail and inspection data. All of the overlays are user definable; with LabVIEW you can change the overlays programmatically and create custom user displays. Connect the system to any network to monitor the inspection. Send images over the network for viewing or store them in a database for future reference. In addition you can use NI VI Server technology to publish your data and results in real time to a Web browser.

19. Connect to Compact FieldPoint

CVS-1450 systems connect easily to Compact FieldPoint, the National Instruments real-time modular industrial control and measurement system. If you need analog signals, specialized

digital inputs, or other types of distributed control to interface to your NI Compact Vision System, you can connect them easily.

20. Rugged, Reliable Design

Run your application with confidence. The Compact Vision System uses the award-winning LabVIEW Real-Time engine, a reliable and embedded programming environment. Time-bounded algorithms ensure that you can meet the deterministic demands of your system.. The temperature range of 0 to 55 °C ensures that uptime is kept at a maximum. No fans, vents, or moving parts ensures reliable industrial inspection for robotics, packaging, or assembly applications.

National Instruments Hungary Kft
8000 Székesfehérvár
Palotai út 61. 5. em. 180.
Tel.: +36 30 655 7704
Fax.: +36 22 314 209
Email: ni.hungary@ni.com
<http://www.ni.com/hungary>

Building 3D Models Using Calibrated Stereo and Structured Light

Novák Norbert¹ and Megyesi Zoltán²

¹ Pantomat Ltd

`nnovak@freemail.hu`

² MTA SZTAKI and ELTE

`megyesi@sztaki.hu`

Abstract. We describe the algorithmic details of an efficient computer vision system for building the 3D model of a face. The system uses a calibrated stereo capturing device and structured light projected onto the measured object. The structured light is used for better precision and reliability, but the system can operate with conventional calibrated stereo images, as well.

The core of the system is a novel stereo matching algorithm that uses a data structure called the disparity space image [5]. The stereo correspondence problem is posed as search for optimal path in the disparity space image. Our algorithm exploits mutual visibility constraints to improve robustness and consistency. As a result, the method reliably reconstructs smooth surfaces even in case of occlusion.

1 Introduction

The measurement and model creation of objects is important in many industrial, medical and multimedia applications. These measurements can be performed by different kind of scanners and sensors, but tactile and laser scanners are considered to be the most accurate. Some objects are hard to measure with these methods though. Either because access to the subject is limited (eg. due to size or distance), the measurement itself would affect the subject, or it would simply take too much time.

The solution to these problems may lie in visual measurement. 3D reconstruction from images is becoming a widely applied technique for measuring objects and creating models. Its advantage, the capability of creating a model based on the **momentary appearance** of the object in many situations compensates for its limitations and inaccuracy.

Since in scene reconstructions from images the 3D information can be acquired from the differences between viewpoints, a relatively cheap hardware setup of at least two cameras is required. Depending on the setup, the reconstructed objects can be of arbitrary size and distance, and only visual contact is needed. On the other hand, certain issues are to be considered. For an automatic metric reconstruction, the cameras have to be calibrated. Without this, in the

general two-view case, only projective reconstruction is possible, using multiple views, reconstruction up to an arbitrary similarity transform is possible [6].

Beside errors coming from calibration, quantisation and other measurement related problems, there are algorithmic difficulties, too. Even though visual information for measurement has been in use for half a century in photogrammetry and two decades in computer vision, distortions, untextured regions, occlusions and the inherent ambiguity still present unsolved problems.

In this paper, we describe a stereo vision system for measuring and creating models of objects with smooth, Lambertian surfaces. The used algorithm requires short baseline canonical stereo configuration [7] and calibrated cameras, with exactly the same intrinsic parameters. The algorithm uses the Disparity Space Image (DSI) data structure that has been introduced by Bobick and Intille [5]. The algorithm is based on a maximum-surface technique [1] to create a smooth and consistent surface. Under the required conditions, the method achieves highly precise and consistent reconstruction of complex objects.

2 Reconstruction using Disparity Space Images

A classical stereo setup consists of two cameras, viewing the same scene. The depth information on scene points can be acquired by locating the projections of the points on the images taken by the cameras. The 3D coordinates than can be calculated by finding the intersection of lines connecting the camera centres and the projections. This requires that the cameras be calibrated, that is, both intrinsic and extrinsic parameters are known. Either by calibration or from a set of points the *epipolar geometry* can be built [7]. For each pixel of one view, it defines a line (the epipolar line) in the other view where its corresponding pair can be. If canonical configuration applies, that is, the camera axes are parallel to each other and perpendicular to the baseline, and the focal lengths are equal, then the views fit to the same plane, and epipolar lines are parallel. In this case, each pixel in one view has its corresponding pair in the same row of the other view.

When using canonical configuration, the Disparity Space Images can be constructed to cover the entire scene. If we consider a pair of corresponding pixels in the two views, we can connect them with the camera centres, and see where the lines meet in the 3D space. In fact, due to the discretisation, a pair of pixels define a projectively distorted cube in the scene from where any visible point is projected to the considered pixels. (See figure 1.)

Since each pixel has its corresponding pair in the same row of the other image, only the x coordinate of the pixel is used to calculate the depth. The difference in the x coordinate of a corresponding pair is called **disparity**.

The space can be sliced according to the rows of the images. Every 3D point in the slice can be projected to a corresponding pair of the two view, and represented as the coordinates of the left projection and a disparity value.

In this way the slices of the scene's visible region can be represented in an upper triangular matrix that has the X coordinates as columns and dispari-

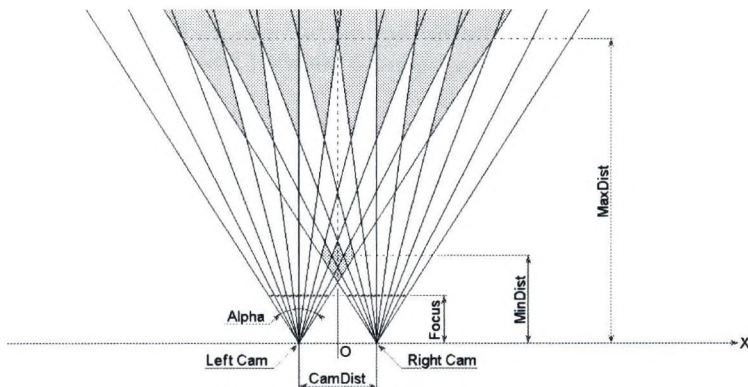


Fig. 1. Canonical Configuration. The figure shows cameras (LeftCam, RightCam) with canonical configuration. One row of image pixels are projected back, to cover a slice of space. Alpha is the cameras viewing angle, Mindist and Maxdist shows the significant region of space

ties as the rows. This representation is the **Disparity Space Image (DSI)**, demonstrated in figure 2. The DSIs belonging to different slices of space form the **Disparity Space (DS)**.

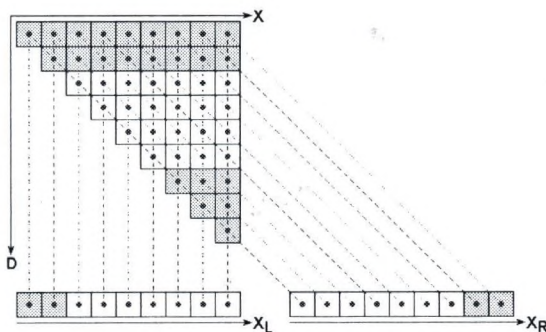


Fig. 2. Disparity Space Image. X is the x coordinate in the reference image and D is the disparity of a possible corresponding pair. Squares marked gray are outside of significant area.

The values of a DSI should be filled in according to how probable it is that the referenced pixel x and its pair with disparity d are the projection of a visible 3D point. The scores, as in many stereo algorithms [8, 9, 2] can be obtained by calculating the similarity or dissimilarity of the regions around the pixels pairs. Standard dissimilarity functions include Sum of Squared Differences (SSD) or Sum of Absolute Differences (SSD), while Normalised Cross Correlation (NCC) is the most widespread similarity function applied. From the camera parameters

and predefined limits, the visible and significant part of DS can be calculated, and only this part is filled up. Figure 3 shows a DSI.

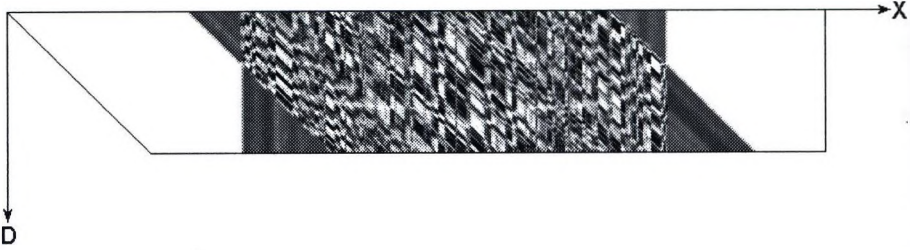


Fig. 3. DSI example. The gray-level shows the NCC score, of matching windows around reference pixels and pixels in the other image with given disparity. Note the thin white line spreading from left to right, showing a smoothly connected surface

The DS structure reflects the probability, that a 3D point is mutually visible. The next task is to locate the objects. Surfaces can be identified by smoothly connected curves of high values in the DSI, spreading in X direction and changing smoothly in D (See figure 3). These curves or paths can be located by maximal flow techniques with dynamic programming as proposed in [1, 5].

The located curves in all DSIs can be combined to form the **disparity map** of the scene viewed from the reference direction. In this map each pixel has the disparity value assigned to it based on which visible curve is it on. Knowing the camera parameters, a disparity map can be transformed to create a 3D model.

3 Consistent reconstruction of surfaces

In this section, we describe a novel method for evaluating DSI images to find consistently smooth surfaces. The method exploits the **surface constraint** as introduced by Geier [3]. If the images are projected back to the scene, at the depth of the original scene points, the texture is identical.

The method is based on finding a curve that satisfies a visibility constraint and has maximal score. The mutually visible part of the DS can be processed from two directions with dynamic programming similar to [4, 5]. If we take the left image as reference, different parts of the scene becomes occluded, thus we obtain a different curve than when the right image is used. The algorithm calculates both paths and joins them by strengthening the overlaps to yield a consistent curve [4]. To find the best surface curve from one side we have to consider what is visible from the reference image. We give a visibility score for each pixel in the DSI that is calculated based on the visibility of the curve they are possibly on. We fill in the visibility score arrays from left to right and right to left according to which one is the reference image.

When checking a new pixel it can support three paths:

1. If checked DSI pixel is occluded from the reference camera, the pixel's DSI score is not important, but the curve might still come through the pixel.
2. If checked DSI pixel occludes a pixel from the other camera, the occluded pixel's DSI score must not be considered, but the checked DSI score is supporting the path coming from that way.
3. If checked DSI pixel is on a mutually visible surface, it must support the path coming from the non occluded direction.

The pixel must support the best path going through it. To put it in a more formal way, let us call the arrays scoring the visibility Left Projection Image (LPrI) and Right Projection Image (RPrI).

A $LPrI(x, d)$ can be given as:

$$LPrI(x, d) = \max(LPrI(x, d + 1),$$

$$LPrI(x - 1, d - 1) - DSI(x - 1, d - 1) + DSI(x, d),$$

$$LPrI(x - 1, d) + DSI(x, d))$$

And consequently $RPrI(x, d)$:

$$RPrI(x, d) = \max(RPrI(x + 1, d + 1),$$

$$RPrI(x, d - 1) - DSI(x, d - 1) + DSI(x, d),$$

$$RPrI(x + 1, d) + DSI(x, d))$$

The arrays are filled up in the order shown in figure 4.

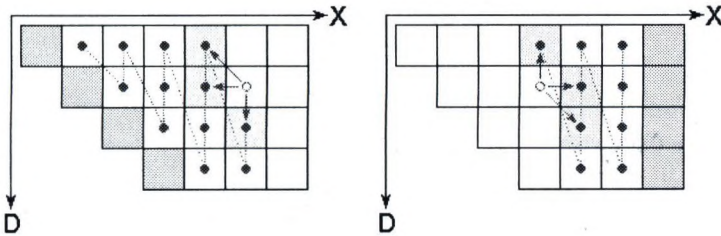


Fig. 4. Order of filling in Projection Images. The lines indicate the direction the Projection Images are filled up. Dark gray squares are filled in with DSI scores, the light grey squares show which pixels are considered when deciding on the visibility of a new pixel.

To join the two arrays we calculate $NPrI = LPrI + RPrI - DSI$ and normalise it to emphasise the best curve. Figure 5 shows a Normalised Projection Image (NPrI) Figure 6 is the same NPrI aligned to become symmetrical, the maximal curve is indicated.

A Disparity Map can be created by picking the best curves in all NPrI images, transforming the disparities into intensities, thus each slice becomes a row of the Disparity Map image.

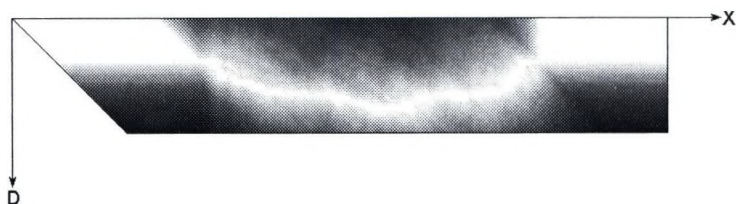


Fig. 5. Normalised Projection Image. The image shows the result of the search for the best curve.

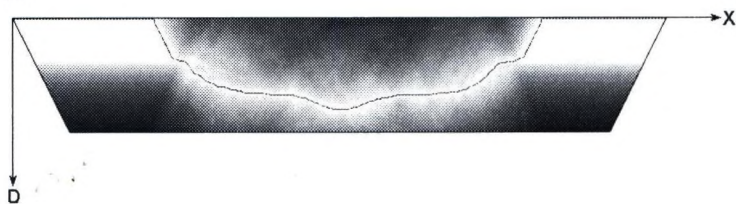


Fig. 6. Aligned Normalised Projection Image. Figure 5 is aligned to become symmetrical. Best curve indicated by red line.

4 Experiments

In this section experiments are shown demonstrating the reliability and accuracy of the method. Results on both synthetic (figure 7) and natural images (figure 8, 9) are shown. The latter are provided by a hardware setup constructed by Pantomat Ltd. that is built up of a structured light projector and two calibrated cameras in canonical configuration. The structured light reduces problems coming from untextured objects, but the algorithm can operate without it as well. Disparity maps for the input images are shown in figure 10. The created models can be seen in figure 11. The running time of the algorithm on a 1 GHz P4 with 256MB RAM was in case of 128x128 images 2 secs, on 512x512 images 133 secs.

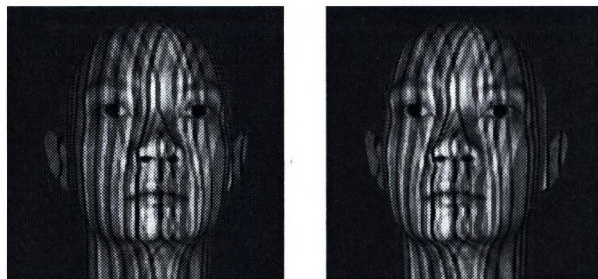


Fig. 7. Synthetic images with structured light



Fig. 8. Normal images with structured light



Fig. 9. Normal images without structured light

5 Conclusions

A stereo vision system has been presented capable of reconstructing complex objects with smooth, Lambertian surfaces. The method uses the DSI structure which allows for a smooth and robust reconstruction. The novelty of the method is the way it takes occlusions and visibility into account to find the best surfaces, and the incorporated left-right consistency.

The method is limited to smooth surfaces and short baseline images, but as experimental results show, together with the hardware setup created by Panomat Ltd, it is capable of precise model building of middle sized object even in case of low texturedness.

Acknowledgement.

This work was supported by the Hungarian Scientific Research Fund (OTKA) grants T038355 and M28078. The authors thank Pantomat Ltd. for providing data.

References

1. Changming Sun, Multi-Resolution Stereo Using Maximum-Surface Techniques. *Digital Image Computing: Techniques and Applications*, 1999.



Fig. 10. Disparity Images. From Left to right: synthetic images, normal images with structured light, normal image without structured light



Fig. 11. 3D Models created from disparity maps

2. Changming Sun, A Fast Stereo Matching Method. *Digital Image Computing: Techniques and Applications*, 1997.
3. János Geier, Surface Constraint: an exact criteria for solving the stereo correspondence problem and for testing the results. *ECVP*, 1998.
4. Chun-Jen Tsai, Aggelos K. Katsaggelos, Dense Disparity Estimation with a Divide-and-Conquer Disparity Space Image Technique. *IEEE*, 1999.
5. Aaron F. Bobick, Stephen S. Intille, Large Occlusion Stereo. *IJCV*, 1999.
6. R. Hartley, A Zisserman. Multiple View Geometry in computer vision. *Cambridge University Press*, 2000.
7. M. Sonka, V. Hlavac, R. Boyle. Image Processing, Analysis, and Machine Vision. Second Edition. *Brooks/Cole Publishing Company*, 1999.
8. E. Trucco, A. Verri. Introductory Techniques for 3-D Computer Vision. 150-175, *Prentice Hall International (UK) Ltd*, 1998.
9. Z. Zhang, R. Deriche, O. Faugeras, Q. T. Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. INRIA No2273 1994.

MedEdit: Műtéti tervezést segítő orvosi képfeldolgozó rendszer

Ollé Krisztián¹, Erdőhelyi Balázs¹, Kuba Attila¹, Varga Endre², Halmai Csongor¹

¹ Alkalmazott Informatikai Tanszék, Árpád tér 2.

ollek, ber, kuba, halmai @inf.u-szeged.hu

² Traumatológiai Klinika, Semmelweis u. 6.

varga@trauma.szote.u-szeged.hu

Szegedi Tudományegyetem, H-6720 Szeged

Kivonat ¹

Az utóbbi időben az autókba szerelt biztonsági rendszereknek köszönhetően egyre kevesebb halálos kimenetelű autóbaleset, ennek következményeként egyre több a nagyon bonyolult csontrendszeri sérülés. Az ilyen sérülések ellátása nagy pontosságot és precizitást igényel. Éppen ezért a műtéti beavatkozás előtt célszerű tervezést végezni arra vonatkozólag, hogyan lehetne legkevesebb műtéti beavatkozással megoldani a problémát. Ezen feladat megoldására fejlesztettünk ki egy számítógépes rendszert amelyet MedEdit-nek neveztünk el. A rendszer több főbb modulból épül fel, melyek közül a geometriai modell készítő és műtéti tervező modulok továbbfejlesztésében elért eredményeinket szeretnénk ismertetni. A MedEdit alapmoduljának feladata CT képek alapján a szegmentált csontmodell elkészítése és a különálló darabokból különálló geometriai objektumok felépítése. A műtéti tervező modulban ezen csontdarabokból építhető fel újra a sérült csont, majd megtervezhető a darabok rögzítése különböző implantátumok felhasználásával.

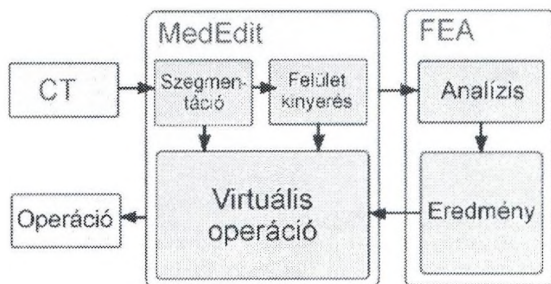
1. Bevezetés

A Szegedi Tudományegyetem traumatológiai és informatikai szakembereinek közös vállalkozásában egy projekt folyik melynek célja, hogy olyan eljárásokat ill. szoftvert fejlesszen ki, mely képes balesetben csontsérülést szenvedett betegeken biomechanikai tesztek elvégzésére, még a sebészi beavatkozás előtt. Az eljárás 3D képek segítségével, matematikai modellt épít fel, ezáltal lehetővé válik többféle stabilizációs lehetőség kipróbálása, mely optimális esetben javítja a sérült csonton végezhető műtét típusát.

2. A rendszer működésének áttekintése

Az 1. ábra mutatja a rendszer főbb komponenseit. A CT képek DICOM (Digital Imaging and Communications in Medicine) formátumban állnak rendelkezésre. A csont szegmentációja a voxelek szűrkeségi értékein alapul. [3]

¹ Ezt a kutatást az OTKA T37840 támogatta



1. ábra.

A rendszer vázlatja a fő komponensekkel, és a köztük lévő kapcsolatokkal.

A következő lépés a felületkinyerés. Itt az ún. Marching Cubes [4] algoritmust használjuk, mely a szegmentált 3D adatokból egy háromszög-hálós felületi modellt állít elő. Ezzel az algoritmussal gyakran túl sok háromszög keletkezik, mely nagyon erőforrás-igényes lenne, ezért egy felületegyszerűsítő eljárással [1] csökkentjük a modell méretét. Párhuzamosan, a szegmentált 3D képet is megtartjuk további felhasználásra.

Az így nyert geometriai modellen a sebész a műtéti tervezővel virtuális operációt végezhet, lyukakat fúrhat, különböző csont részeket illeszthet egymáshoz ill. implantátumokat (pl. csavar) helyezhet a csontba.

Kiegészítve a geometriai modellt anyagi tulajdonságokkal kapjuk a mechanikai modellt, amin már végrehajtható a véges elemes analízis (FEA). Ezek után megfigyelhető, hogyan viselkedik a kezelt csont terhelés hatására. Rendszerünk az orvos számára kényelmes felhasználói felületen keresztül biztosítja az exportálás lehetőségét.

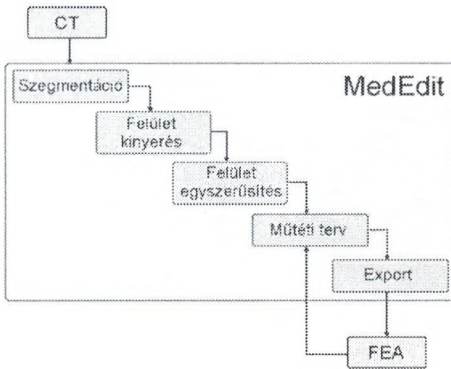
A terheléses feszültség-analízis a véges elemes programon belül történik. [5] A rendszer moduláris felépítésű, (lásd 2. ábra) minden modulnak speciális feladata van.

3. A MedEdit rendszer

3.1. Szegmentálás

Ezen modul célja az, hogy elválassza a csontszövetet a többi szövetről. Ez azt jelenti, hogy a szürke árnyalatú CT képekből álló, térfogatból egy bináris térfogatot hozunk létre, ahol 0 reprezentálja a háttérrel és 1 a csontot.

Itt a fuzzy összefüggő 3D kép-szegmentáló [3] algoritmus mellett döntötünk. Ez az algoritmus már néhány kiindulási pont megadása után automatikusan tudja szegmentálni a megfelelő szövetet. Miután az algoritmust lefuttattuk, lehetőségünk van a különálló csontdarabok szétválasztására ezt a feladatot úgy oldottuk meg, hogy ún. összefüggő komponensekre bontottuk a szegmentált



2. ábra. A MedEdit rendszer vázlata.



3. ábra.

A szegmentálás lépései. A medencecsont CT felvételének egy darabja (balról).

A fuzzy szegmentálás eredménye (középen). A szegmentálás eredménye

üregfeltöltés után (jobbról).

képet. Ez azért jó nekünk, mert a különböző darabokat külön-külön tudjuk kezelni a későbbiekben. Végül még végrehajtottunk néhány utófeldolgozó lépést. Pl.: üregfeltöltés, símitás (3. ábra).

3.2. Felület kinyerés

Ebben a lépésben hozzuk létre a geometriai modellt, a szegmentált térfogatból. Mivel a szegmentálásnál több darab is keletkezhet ezért ezekből külön-külön modellt generálunk. Itt a Marching Cubes [4] algoritmus egy általunk módosított változatát alkalmaztuk. Az eredeti algoritmust annyiban változtattuk meg, hogy a szegmentált térfogatot használja fel maszkként, de az eredeti szürke árnyalatú képeken futtatjuk le. Ellentétben az eredeti algoritmussal, mely egy küszöbértéket használ fel a felülethez tartozó elemek megállapítására. Sajnos az algoritmus igen nagy számú háromszöget és csúcspontot hoz létre (lásd 5. ábra bal oldala).

3.3. Felületegyszerűsítés

Itt csökkentjük az előző modulban keletkezett komplex, sok felületi elemből álló geometriai modellek elemeinek számát. Több okból is szükségünk van a felületegyszerűsítésre, egyrészt a megjelenítés sebességét növeljük vele, másrészt viszont a fizikai elemző program csak meghatározott számú felületi elemet képes kezelni. Jelenleg Garland és társai [1] által kidolgozott felületegyszerűsítő algoritmust használjuk fel ebben a lépésben (lásd 5. ábra jobb oldala).

3.4. Műtéti tervezés

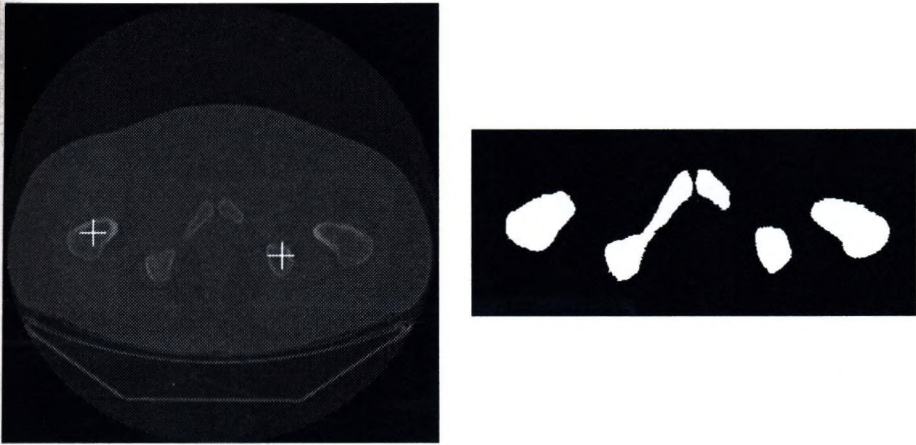
Ennek a modulnak a célja, hogy lehetőséget biztosítson a sebésznek, különböző műtéti eljárások megtervezésére és kipróbálására. A felhasználói felületen egyszerre négyféle nézőpontból, szögből és nagyításban kísérheti figyelemmel az adott kísérleti eljárást (lásd 6. ábra). A rendszer képes arra, hogy több 3D objektumot kezeljen egyszerre, ezáltal egy törés esetén, ahol a csont esetleg több darabra esett szét az orvos ezeket a különálló darabokat mozgatva össze tudja illeszteni a csontot. Ebben segítséget nyújt neki az is, hogy ütközésvizsgálatot tudunk végezni a különböző csont darabokra, ezáltal a valósághoz hasonlóan a virtuális térben sem tudja a darabokat egymásba tolni. Valamint dolgozunk olyan eljárásokon amelyek segítségével megpróbáljuk az illesztéseket automatikussá tenni. Ezen felül többféle 3D modell szerkesztési funkciót is biztosítunk, azaz lehetőség van egy modellen belül különböző részek kiválasztására, esetleg végleges letörlésére. Például egy furat létrehozása esetén először be kell állítani, hogy mely nézőpontból szeretné megközelíteni a csontot, majd a pontos pozíciót és a fúró dőlésszögét. Mindezt természetesen interaktív módon. Lehetőség van röntgenkép-szerű hatás elérésére is azáltal, hogy átlátszóvá teszi a csontot.

3.5. Export

Rendszerünkből exportálható a geometriai modell a véges elemes analízist végző program számára. A mechanikai modell előállításához azonban ezt ki kell egészítenünk bizonyos anyagi tulajdonságokkal, az anyagra vonatkozó fizikai állandókkal, terhelési- és elmozdulási vektorokkal. Ezeket az orvos egy kényelmes felületen meg tudja tenni. Mivel a véges elemes analízist végző program nemcsak felületet hanem térfogatot is képes felhasználni ezért mindkét adat kimentésére lehetőséget biztosítunk.

4. Példa

Egy műtéti tervezés fontosabb lépései követhetők a 4-6. ábrákon



4. ábra.

Balról a felhasználó a fehér kereszttek segítségével megadja a kiindulási pontokat a szegmentáláshoz. Jobbról a szegmentálás eredménye.

5. Eredmények

Rendszerünk egyelőre kísérleti jelleggel működik. Végigfut rajta a teljes folyamat, de vannak olyan helyek, ahol felhasználói beavatkozásra van szükség. Például a szegmentációhoz kiindulópontokra van szükség, valamint az eljárás eredményét az orvosnak ellenőrizni kell. A FEA rendszerrel a kommunikáció sem automatikus, hanem egy szkripten keresztül történik.

A felhasználói beavatkozást is beleértve egy átlagos vizsgálat esetén körülbelül 5 perc alatt elkészíti a geometriai és mechanikai modelleket. A véges elemes analízis durván 6 percig fut egy átlagos medencecsont mintára, ha a szegmentált 3D képet exportáltuk. Felület exportálása esetén ez az idő 30 másodperc alá csökken, hiszen ez esetben jóval kisebb a véges elemes háló mérete. A méréseket egy 2 GHz-es, 1.5 GB RAM-al rendelkező számítógépen végeztük.

Az eredmények nagyjából megfelelnek a klinikai elvárásoknak, igaz, a kvantitatív összehasonlító mérések még hátra vannak.

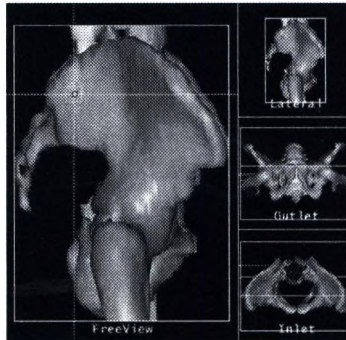
6. További tervek

Jelenleg a MedEdit segítségével lehetőség van lyukak furására, ill. virtuális csavarok behelyezésére. A későbbiek folyamán szeretnénk elérni, hogy a sebészek által használt szabványos implantátumok bármelyikét fel lehessen használni a virtuális operáció folyamán. Valamint, hogy a csontmodellen lehessen különböző méréseket végezni.



5. ábra.

Balról a medencecsont marching cubes algoritmus segítségével létrejött felületi modellje, mely több mint 200 000 háromszöget tartalmaz. Jobbról a medencecsont felületegyszerűsítés után. A felület itt 10 000 háromszögből épül fel.



6. ábra.

A műtéti tervezés fázisa. A medencecsont oldalirányú nézete és a fűrőhegy pozicionálása (jobbról). Ugyanezen objektum másik 3 irányú nézete (balról).

Hivatkozások

1. Garland M, Heckbert PS. Surface simplification using quadric error metrics. ACM SIGGRAPH' Computer Graphics; 1997.p. 209–16.
2. Geiger B. Three-dimensional modeling of human organs and its application to diagnosis and surgical planning. INRIA, BP ce. 1993
3. Nyúl LG, Falcao AX, Udupa JK. Fuzzy- Connected 3D Image Segmentation at Interactive Speeds. SPIE Medical Imaging 2000; 3979 p.212-23.
4. Lorensen WE, Cline HE. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM SIGGRAPH Computer Graphics; 1987 21(4)
5. Ollé K, Kuba A, Varga E. MedEdit: Computer-Assisted Orthopedic-Trauma Surgery Planner System. European Surgical Research 2002; 34(1)

Multimodalitású orvosi képfeldolgozás céljaira kidolgozott absztrakt szoftverkönyvtár

Opposits Gábor¹, Valastyán Iván², Trón Lajos³, Emri Miklós⁴

DE OEC PET Centrum
4026 Debrecen, Bem tér 18/c
ogabor@pet.dote.hu

Kivonat Intézetünkben a multimodalitású orvosi képfeldolgozáshoz kapcsolódó szoftverfejlesztéseinket egy C++ alapú volumetrikus szoftverkönyvtár segítségével végezzük. Ennek a könyvtárnak az alapja egy absztrakt, a generikus programozás irányelvei alapján kidolgozott multi-dimenzionális parametrikus sablonkönyvtár (MDTL). A tervezés és az implementáció során kidolgoztuk az n-dimenziós voxel-tömbök kezeléséhez szükséges alapsztályokat, iterátorokat és az image algebra alapjait képező operátorokat. Speciális elvárásként fogalmazzuk meg, hogy egy n-dimenziós tömb a könyvtár használója számára egyaránt kezelhető legyen a voxelek vagy az n-1 dimenziós tömbök tároló osztályaként. A C++ referenciátípus fogalmát kiterjesztettük a könyvtár minden komponensére, ami a memória- és időigényes algoritmusok optimális implementálását segíti. Az MDTL hatékonyságának elemzése céljából olyan tesztprogramokat készítettünk, amelyek segítségével igazoltuk, hogy a szoftverkönyvtárral implementált algoritmusok csak kismértékben növelik a futásidőt a feladat-specifikus C alapú implementációhoz viszonyítva. Az MDTL könyvtárra épülő komplex, platform független fejlesztői környezet segítségével a kép-regisztráció, képfúzió, agyatlasz-technika és a statisztikus parametrikus képfeldolgozás területéhez kapcsolódó, Linux- és XP operációsrendszerek alatt futtatható programokat készítettünk.

1. Bevezetés

A multimodalitású orvosi képfeldolgozás bemenő adatai a képpalkotó diagnosztikai eljárásokkal (CT, MRI, PET, SPECT, gammakamera, UH, ...) előállított, speciális fájl-formátumban tárolt, vagy hálózati kapcsolaton keresztül elért PACS archívumok adatai. A különböző fizikai elveken alapuló leképező módszerek absztrakt modellje szemleletesen úgy képzhető el, hogy a vizsgált objektumból egy képeletbeli térrács minden egyes pontjában meghatározunk egy, vagy több fizikai paramétert, ami a gyakorlatban egy adatgyűjtés és a képrekonstrukciós eljárás összetett folyamatának felel meg. Célkitűzésként fogalmazzuk meg, hogy a multimodalitású orvosi képfeldolgozás számára kidolgozzunk egy olyan C++ alapú, platformfüggetlen szoftverkönyvtárat, amelyben a képfeldolgozás alapobjektumait reprezentáló osztályokat az orvosi leképező eszközök absztrakt modellje szerint készítjük el. Ennek megfelelően a 'képek' és 'görbék' specializált

osztályait egy általános tudással rendelkező, n -dimenziós ($n=1, 2, \dots$) adathalmaz kezelését megvalósító alaposztály segítségével implementáltuk. A tervezés fontos szempontja volt a generikus programozás alapelveinek követése, ezek közül is kiemelt szerepet kapott az algoritmusok és a tárolók egymástól elkülönített implementációja. Ezt az elvet az STL-könyvtárral kapcsolatban így fogalmazták meg: "Annak oka, hogy az adatszerkezetek és az algoritmusok ilyen tökéletesen együtt tudnak működni az, ... hogy semmit sem tudnak egymásról." (Alex Stepanov).

2. MDTL könyvtár

Az alapkönyvtárat úgy készítettük el, hogy a C++ nyelvben megszokott referencia fogalmat kiterjesztettük a könyvtár minden elemére. Így elkerülhetővé válik a nagy mennyiségű adatmásolással járó lassulás, valamint a memória kihasználtsága is optimálisabb. Egy N -dimenziós tömb egyaránt kezelhető a voxelek valamint az $(N-1)$ -dimenziós tömbök tárolójaként is. Ez utóbbi segítségével pl. egy két dimenziós képfeldolgozó algoritmus az STL könyvtár segítségével használható 3D esetben is.

2.1. Tároló osztályok

Az MDTL-ben a rögzített méretű parametrizált vektorok (`_vector<int,N>`), valamint az n -dimenziós voxel-tömbök kezelésére kidolgozott `array<T>` osztályokat készítettük el. Ez utóbbi minden algoritmus számára egységes, az interfészt a különböző típusú bejárók biztosítják. A tárolókat sablon osztályként valósítottuk meg lehetővé téve a fordítási időben való típusparaméterezést.

```
array<float> A(128,128); // 128 x 128 2D float array
array<short> B(15, 128,128); // 15 x 128 x 128 3D short array
```

```
_vector<int,3> shape( 15, 128, 128); // int shape[] = 15, 128, 128
array<short> B( shape ); // 15 x 128 x 128 3D short array
array<short> C( B.shape() ); // 15 x 128 x 128 3D short array
```

Az `array` osztályban olyan adattagok és függvények találhatóak, amelyek segítségével a C++ nyelvben megszokott referencia fogalom az n -dimenziós tömböl számára is elérhető. Ezek szerint készíthető egy olyan tömb, amelyen keresztül egy másik tömb minden egyes eleme, vagy egy speciálisan strukturált részhalmozát képező elemei érhetőek el.

```
// D referencia B-re, D-n keresztül B elemei érhetőek el
array<float> D(&B);
D = 8; // B minden eleme 8 lesz
```

```
int_vector<2> offset(10,10); // 2D offset (10,10) indexű voxel
int_vector<2> shape (64,64); // 2D shape 64 x 64 matrix
// E egy 64x64-as 2D tömb amely az A bizonyos voxeleit kezeli
array<float> E( &A, offset, shape);
A = 2; // A minden eleme 2
E = 3; // A egy 64x64-es részhalmaza 3
```

2.2. Bejárók

Az MDTL a szokásos 'egyetlen elem elérésű' bejárókon (`voxel_iterator<T>`) túl az úgynevezett blokk bejárókat (`bulk_iterator<T>`, `edge_iterator<T>`) is támogatja. Megadva egy maszkot egy centrális elem körül (Kernel) a referencia operátor segítségével nem csak a központi elem, hanem a maszkban megadott összes elemet elérhető. Ez a fajta elembejárás megkönnyíti több képfeldolgozási algoritmus implementálását, úgymint élkimielés, simítás, stb.

```
while( i != A.end() ) // végigmegy minden elemen
cout <<*i++<<" "; // és kiírja
cout << endl
```

2.3. A voxel-itárotorok implementálásával elérhetővé vált az STL könyvtár több algoritmusa:

```
fill( A.voxel_begin(), A.voxel_end(), fill_value )
copy( A.voxel_begin(), A.voxel_end(), B.voxel_begin() )
find ( A.voxel_begin(), A.voxel_end(), find_value )
count( A.voxel_begin(), A.voxel_end(), count_value )
transform( A.voxel_begin(), A.voxel_end(), B.voxel_begin(), an_op )
for_each( A.voxel_begin(), A.voxel_end(), an_operator )
accumulate( A.voxel_begin(), A.voxel_end(), result )
mismatch( A.voxel_begin(), A.voxel_end(), B.voxel_begin() )
replace( A.voxel_begin(), A.voxel_end(), old_value, new_value )
```

A blokk iterátorok és egy speciális `conv` segítségével a könyvtár felhasználói számára könnyen implementálható több konvolúción alapuló algoritmus. A következő példa egy élkimielő algoritmust mutat be a Perwitt-féle konvolúciós kernell használatával. Az eredmény az 1. ábárn látható.

```
array<float> A; // 128 x 128 float tömb
A.load( filename ); // beolvasás ...
```

```
array<float> B(A.shape());
conv( A, B, Prewit );
```

A Prewit funktor konstruktora:

```
Prewit()
: KernelH(3,3), KernelV(3,3)
{
KernelH = -1, 0, 1,
-1, 0, 1,
-1, 0, 1;
KernelV = KernelH.swap(0,1);
}
```

A conv algoritmus:

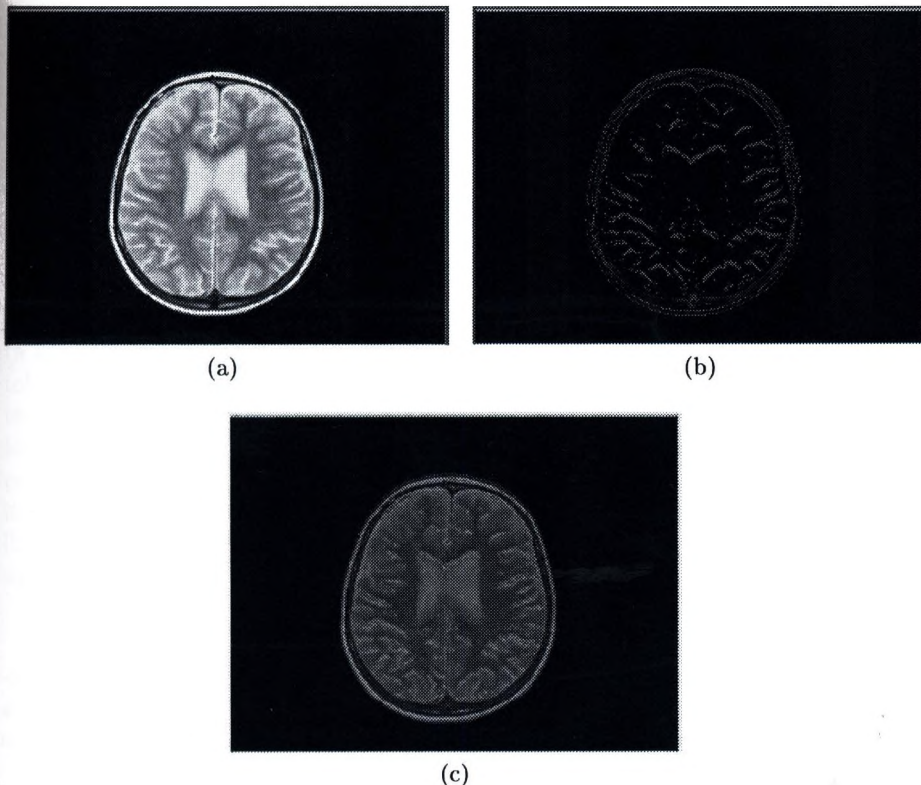
```
template< class kernel >
conv( array<T>& A, array<T>& B, kernel )
{
transform( A.bulk_begin( kernel.get_mask()),
A.bulk_end ( kernel.get_mask()),
B.bulk_voxel_begin(),
kernel);
transform( A.edge_begin( kernel.get_mask()),
A.edge_end ( kernel.get_mask()),
B.edge_voxel_begin(),
kernel);
}
```

A voxel- és blokk-iterátorokon túl létezik az úgynevezett altömb bejáró is. Az altömb bejáró segítségével adott irány mentén visszakaphatjuk akármelyik szeletet, megvalósítva ezzel pl. egy szelet bejárót 3D tömb esetén. Összefoglalás, eredmények A multimodalitású orvosi képfeldolgozás adatainak tárolására és kezelésére kidolgoztunk egy általános, a generikus programozás technikai elveit követő tárolót, valamint az elemek elérését lehetővé tevő bejárókat. Az MDTL könyvtárra épülő komplex, platform független fejlesztői környezet segítségével a képregisztráció, képfúzió, agyatlasz-technika és a statisztikus parametrikus képfeldolgozás területéhez kapcsolódó, Linux- és XP operációsrendszerek alatt futtatható programokat készítettünk.

Az eredmény az 1 ábárn látható.

3. Összefoglalás, eredmények

A multi modalitású orvosi képfeldolgozás adatainak tárolására és kezelésére kidolgoztunk egy általános - az stl filozófiáját követő - tárolót, valamint az elemek



1. ábra.

Élkiemelés az MDTL-ben, (a): nyers MR kép, (b): detektált élek, (c): fúziós kép

elérését lehetővé tevő bejárókat. Az MDTL könyvtárra épülő komplex, platform független fejlesztői környezet segítségével a képregisztráció, képfúzió, agyatlasz-technika és a statisztikus parametrikus képfeldolgozás területéhez kapcsolódó, Linux- és XP operációsrendszerek alatt futtatható programokat készítettünk.

Köszönetnyilvánítás

A projekt az IKTA-153/2003, IKTA-00006/2001 és NKFP-1/A/0010/2002 pályázatok támogatásával készült

Hivatkozások

1. Bjarne Stroustrup, A C++ programozási nyelv, Kiskapu Kft., 2001

Légutak vizsgálata 3D CT-képeken

Palágyi Kálmán

Szegedi Tudományegyetem, Képfeldolgozás és Számítógépes Grafika Tanszék
palagyi@inf.u-szeged.hu

Kivonat Egy összetett módszert mutatunk be kiterjedt 3D „csőszerű” és fa-szerkezetű bináris objektumok elemzésére. Az objektumok egy pont vastagságú középvonalának kivonására és az elágazási pontok kijelölésére javasolt eljárásunk szolgál alapul a fák kvantitatív leírásához, illesztéséhez, az egyes ágak térfogatának, átmérőjének meghatározásához és az egyes szegmensek címkézéséhez. A módszert légútfantomokon és élő szervezetből felvett (in vivo), CT-vizsgálatokból szegmentált légutakon validáltuk.

1. Bevezetés

3D „csőszerű” és fa-szerkezetű szervek, struktúrák (pl. légutak artériák és vénák) kinyerhetők CT- vagy MR-vizsgálatokból, mely objektumok képpontok (voxel-ek) összefüggő halmazaként adóttak. A kvantitatív analízis lényegesen egyszerűbbé válik, ha a voxel-szintű fákat formális fákkal reprezentáljuk. A formális leírásunk a kiterjedt objektumok egy voxel vastagságú középvonalának meghatározásán alapul.

2. Módszer

Az összetett módszerünk az alábbi fő lépésekből áll:

1. szegmentálás (adaptív régió-növesztéssel) és a szegmentálással kapott bináris objektum topológiai korrekciója (morfológiai zárással),
2. a fa gyökerének meghatározása (interaktívan vagy automatikusan),
3. egy voxel vastagságú középvonal kijelölése (szekvenciális vékonyítással),
4. a középvonal tisztítása (az ágak hossza és az ághoz tartozó elágazási pontok mélységének alapján),
5. az elágazási pontok meghatározása,
6. formális fa generálása (a középvonal ágainak címkézésével),
7. a szegmentált fa particionálása (a címkék izotropikus terjesztésével),
8. kvantitatív jellemzők meghatározása (hossz, térfogat, felszín és átmérő számítása az egyes szegmensekre, ágakra).

A fenti módszer egyes lépéseit korábban részletesen közöltük [4,5], így a jelen cikkben csak a legfontosabb lépéseket vázoljuk.

2.1. A középvonal kijelölése

A váz (*skeleton*) [2] általában felszín-szegmenseket (2D elemeket) is tartalmaz 3D-ben, amelyek határozottan nemkívánatosak „csőszerű” objektumok jellemzősekor. A vékonyítás nagy előnye a többi vázkijelölő módszerrel szemben az, hogy képes felszín-szegmensektől mentes, egy voxel vastagságú középvonalak kivonására „vonál-végpontokat” megőrző iteratív redukcióval [3]. A javasolt szekvenciális vékonyító algoritmusunk (26, 6) bináris képeket feltételez, vagyis a bináris képünk objektumainak „1” pontjaira a 26-, a „0” pontokra pedig a 6-szomszédsági/összefüggési reláció érvényes [2]. Szükségünk van néhány fogalom (a hat főirányhoz tartozó határpontok, a vonál-végpontok és az egyszerű pontok) meghatározására is. Egy „1” voxel a (26, 6) képen U-határpont, ha a felette lévő, vele 6-szomszédos voxel „0”. (Az N-, E-, S-, W-, és D-határpontok hasonlóképpen definiálhatók.) Egy „1” voxel vonál-végpont, ha a 26-szomszédai között pontosan egy „1” voxel található. Egy „1” voxel egyszerű (*simple*) pont, ha törlése („0”-vá változtatása) nem változtatja meg a kép topológiai tulajdonságait [2]. Megjegyzendő, hogy az „egyszerűség” lokális tulajdonság a (26, 6) képeken: elődönthető az adott voxel $3 \times 3 \times 3$ -as környezete ismeretében [2].

A javasolt algoritmust az alábbiakban vázoljuk:

repeat

for az U, az N, az E, az S, a W, és a D irányokra **do**
jelöljük meg az aktuális irányú határpontokat, melyek egyszerűek, de nem vonál-végpontok

for valamennyi megjelölt p pontra **do**

if p egyszerű az aktuális képen **then**

if p nem vonál-végpont **then**

töröljük a p pontot

else if (p „1”-es 6-szomszédainak száma) $\geq t$ **then**

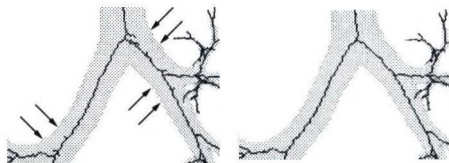
töröljük a p pontot

endfor

endfor

until változás történt a képen

Az eljárás lényeges eleme a végpont-ellenőrzés, mely segít abban, hogy a középvonalak kevesebb nemkívánatos parazita ágat tartalmazzanak (lásd a 1. ábrát). A végpont-ellenőrzést a $t \in \{0, 1, 2, 3, 4, 5, 6\}$ paraméter vezérli, javasolt értéke $t = 1$ vagy $t = 2$. Az algoritmus topológia-megőrző volta evidens, mivel csak egyszerű pontokat töröl szekvenciálisan. Garantált a maximálisan vékonyított (csak egy voxel vastagságú vonal-szegmensekből álló) középvonal kivonása is, mivel a vonál-végpontok kivételével minden egyszerű pontot eltávolítunk. A vékonyított képen csak háromféle objektumpont („1” voxel) fordul elő: vonál-végpont, vonal-pont (pontosan kettő objektumponttal 26-szomszédos) és elágazási-pont (kettőnél több objektumponttal 26-szomszédos).



1. ábra.

Egy szegmentált fa részlete és középvonalai. A baloldali középvonalakat végpont-ellenőrzés nélkül ($t = 6$) vontuk ki, a jobboldalikat pedig végpont-ellenőrzéssel ($t = 1$). A baloldali középvonal 176, a jobboldali pedig csak 128 ágat tartalmaz. A baloldali képen nyilakkal jelöltünk meg néhány parazita ágat, amelyeket nem növeszt az eljárás a végpont-ellenőrzésnek köszönhetően.

2.2. A középvonal tisztítása

Valamennyi vázkijelölő módszer érzékeny az objektum határának egyenetlenségeire, így a vékonyítással kapott középvonalak is korrekcióra szorulnak. A tisztításra az alábbi feltételelt fogalmaztuk meg: egy ág törlendő, ha a hossza (képpontokban mérve) rövidebb, mint „ l ” és a hozzátartozó elágazási pont mélysége legalább „ d ” (ahol a mélységi információt a kiterjedt objektum határára számított távolságtérképből [1] nyerjük.) A tisztítást több iterációs lépésben is elvégezhetjük — különböző küszöbértékek mellett. A tisztításra a 2. ábra mutat példát.



2. ábra.

Egy szegmentált fa részlete és középvonalai tisztítás előtt (bal) és tisztítás után.

2.3. Formális fa generálása és a partícionálás

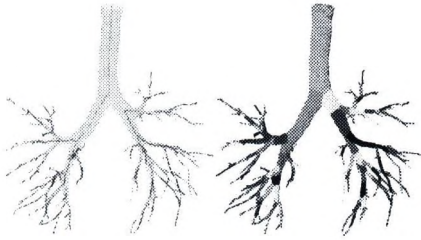
Az egy voxel vastagságú tisztított középvonalakat gráffá transzformáljuk, ahol a gráf szögpontjainak a középvonal voxelei felelnek meg, két szögpontot pedig él köt össze, ha a megfelelő voxelek 26-szomszédosak. Esetünkben az így kapott gráfok fák, melyek tovább egyszerűsíthetők: az eredeti gráf szögpontjai közül csak az elágazási-pontokat és a vonal-végpontokat tartjuk meg és a régi gráfon minden egyes vonal-pontokból álló élsorozat (utat) egyetlen címkézett él helyettesít (lásd 3. ábra). A fa-gráf címkézésével egyidejűleg a középvonalakat pontjait is címkézhetjük, mely címkék izotropikus terjesztésével a kiindulási kiterjedt objektumot (az eredeti képen található fa-szerkezetet) szegmensekre



3. ábra.

Egy emberi légútfa és a középvonalai (bal); a vonatkozó formális fa (jobb).

bontjuk, partícionáljuk. A partícionálásra a 4. ábra mutat példát. A formális fát XML formátumban tároljuk, ahol felsoroljuk a fa csúcspontjait és éleit. Az élekhez számos egyéb információ is tartozik: címke, generációs szám, a vonatkozó szegmenshez tartozó (a középvonalban szereplő) voxelek listája és négy kvantitatív index: vonatkozó ág/szegmens hossza, térfogata, felszíne és átmérője.

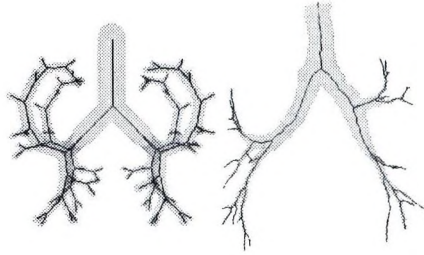


4. ábra.

Példa partícionálásra. A szegmentált légútfa és a címkézett középvonalak (bal), a partícionált objektum a címke-terjesztés után (jobb).

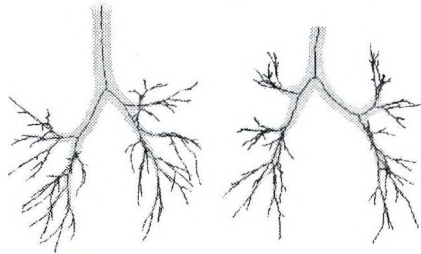
3. Eredmények

A módszert az Iowa-i egyetemen fejlesztettük ki (NIH grant HL-064368), ahol a jelen cikk szerzője egy évig vendégeskedett. Módszerünket számos adaton teszteltük és validáltuk. Az elágazási pontok lokalizációs hibáját egy matematikai fantom (5.a. ábra) 343 elforgatott példányán vizsgáltuk, ahol az elágazási pontok pontos pozíciói ismertek voltak. Az összetartozó elágazási pontok közötti euklidészi távolság átlaga és szórása voxel alatti pontosságot mutat (0.93 ± 0.41 voxel méret) [4]. A módszert kiértékeljük kétféle légútfantomra (5.b. és 6.a. ábra) is, melyeket 9–9 pozícióban digitalizáltunk CT-vel [4,5]. A módszer hatékony, robusztus és reprodukálható. Az in vivo (6.b. ábra) validációt 6 (CT-vel felvett) emberi légútfára végeztük el — jelenleg az eredmények publikálásán dolgozunk.



5. ábra.

Matematikai fantom és középvonalai (bal). Merev (műanyag téglába foglalt) légút-fantom és középvonalai (jobb).



6. ábra.

Burgonyakorpával töltött hengerbe helyezett, gumiból készült szegmentált légútfantom és középvonalai (bal). Egy CT-vel felvett szegmentált in vivo légút és középvonalai (jobb). Az in vivo validációt 6 vizsgálatból szegmentált légútfák elforgatott példányaira végeztük el.

Hivatkozások

1. G. Borgefors, G.: Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing* **27** (1984) 321–345.
2. T.Y. Kong, A. Rosenfeld, Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing* **48**, 357–393, 1989.
3. K. Palágyi, A. Kuba, A parallel 3D 12-subiteration thinning algorithm. *Graphical Models and Image Processing* **61**, 199–221, 1999.
4. K. Palágyi, J. Tschirren, M. Sonka: Quantitative analysis of three-dimensional tubular tree structures, *Proc. Medical Imaging 2003: Image Processing*, SPIE Vol. 5032, 277–287, 2003.
5. K. Palágyi, J. Tschirren, M. Sonka: Quantitative analysis of intrathoracic airway trees: methods and validation, *Proc. 18th Int. Conf. Information Processing in Medical Imaging*, IPMI 2003, Lecture Notes in Computer Science 2732, Springer, 222–233, 2003.

Valódi és szimulált dendrites struktúrák morfológiai jellemzése

Póliska Csaba¹, Gácsi Zoltán¹, Barkóczy Péter¹

Miskolci Egyetem, Fémtechnológiai Tanszék, 3515 Miskolc-Egyetemváros
¹{policisabi@yahoo.com

Kivonat A dolgozatban különböző módon előállított dendrites struktúrák morfológiai jellemzésével foglalkozunk. Évek óta kísérletezünk SCN (succinonitril) - acetone bázisú modellanyag dendrites kristályosításával, melynek során kialakítottunk egy jól használható képfeldolgozási és elemzési eljárást az alak jellemzésére.

Munkánk során többféle módon keletkezett dendriteket használtunk fel a mérések céljaira. Az általunk kristályosított dendriteken kívül különböző szimulációk eredményeképpen keletkezett dendriteken is végeztünk méréseket, és mint különleges dendritalakokat, többféle hópehelyformát is jellemeztünk.

Dolgozatunkban bemutatjuk, mennyire különbözőek, és mégis mennyire hasonlóak a különböző morfológiák.

1. Bevezetés

Dendrites, vagy ahhoz hasonló struktúra nagyon sokféle módon keletkezhet [1, 2, 3]. A fémek és ötvözeteik leggyakoribb kristályosodási formája a dendrites kristályosodás. Munkánk során először geometriai paraméterek mérésével határoztuk meg a különböző beállítások (kristályosodási sebesség, hőmérséklet gradiens, ...) mellett kristályosodó dendrites szerkezetekben kialakuló különbségeket [4], de ez nem minden esetben bizonyult elegendőnek. Szemmel látható volt, hogy a kristályosodás során kialakult dendritek nemcsak geometriai paramétereikben, hanem alakjukban, morfológiájukban is jelentősen különböznek egymástól. Ennek megfelelően olyan morfológiai paraméterek keresésébe fogtunk, melyekkel az alak kellőképpen leírható.

Orientációs rózsák diagram szerkesztése [5]. Az orientációs rózsák diagramot úgy kapjuk, hogy a detektált képre vízszintes irányú tesztvonalakat fektetünk, majd 0°-360°-os intervallumban forgatva a detektált képet, meghatározzuk a tesztvonalak és a dendrit területének a metszés számait. Ezek után polárkoordináta-rendszerben ábrázoljuk az összetartozó fok-metszés szám értékpárokat. A diagram alakja utal az alakzat szimetriaviszonyaira.

Orientációs faktor meghatározása [5]. A dendriteket, pontosabban a dendrit területének pontjait úgy tekintettük, mint részlegesen orientált, síkon elhelyezkedő vonalrendszer. Ez alapján a következő képlettel határozhatjuk meg az orientációs faktort (Ω_{vs}):

$$\Omega_{vs} = \frac{(P_L)_m - (P_L)_p}{(P_L)_m + \left(\frac{\pi}{2} - 1\right) (P_L)_p} \quad (1)$$

$(P_L)_m$: A dendritre, a hossz tengelyével merőleges irányban fektetett tesztvonalak, és a dendrit határoló görbájének a metszés számai,

$(P_L)_p$: A dendritre, a hossz tengelyével párhuzamos irányban fektetett tesztvonalak, és a dendrit határoló görbájének a metszés számai.

Ez a két jellemzési lehetőség sajnos csak az általunk kristályosított dendritéken volt alkalmazható.

A *Saltykov-féle alaktényező* [6] azt mutatja meg, mennyire körszerű az adott alakzat, jelen esetben a dendrit.

$$\Phi_s = 2\sqrt{\pi} \frac{\sqrt{T}}{K} \quad (2)$$

A Φ_s meghatározásához a dendrit T területének és K kerületének az ismeretére van szükség. Az egyszerű képlet alapján kapott érték minél jobban közelít az 1-hez, annál körszerűbb az alakzat (1 esetén körről van szó).

A *fajlagos felület* kiszámítása szintén egy nagyon egyszerű összefüggés segítségével történik, melyhez csak a dendrit területére (T) és kerületére (K) van szükség.

$$S_v = \frac{K}{T} \frac{4}{\pi} \quad (3)$$

Jelen dolgozat nem tartalmazza, de nagyon fontos és jelentős paraméter még a körszimmetrikus dendrites struktúrák jellemzésére a körkovariancia.

2. Képfeldolgozás és mérés

A mérésekhez felhasznált dendrites struktúrák négy különböző forrásból származtak. Általunk kristályosított dendritek nagy számban álltak rendelkezésünkre, és mivel többféle kísérleti beállítást is alkalmaztunk munkánk során, ezért ezek már különböztek morfológiájukban (1. ábra) [7]. A variálható kísérleti beállítások az alábbiak voltak:

- gravitációs vektor irányával megegyező irányú kristályosítás (továbbiakban "le"),
- a gravitációs vektor irányával ellentétes irányú kristályosítás (továbbiakban "fel"),
- a gravitációs vektor irányára merőleges irányú kristályosítás (továbbiakban "víz"),

illetve:

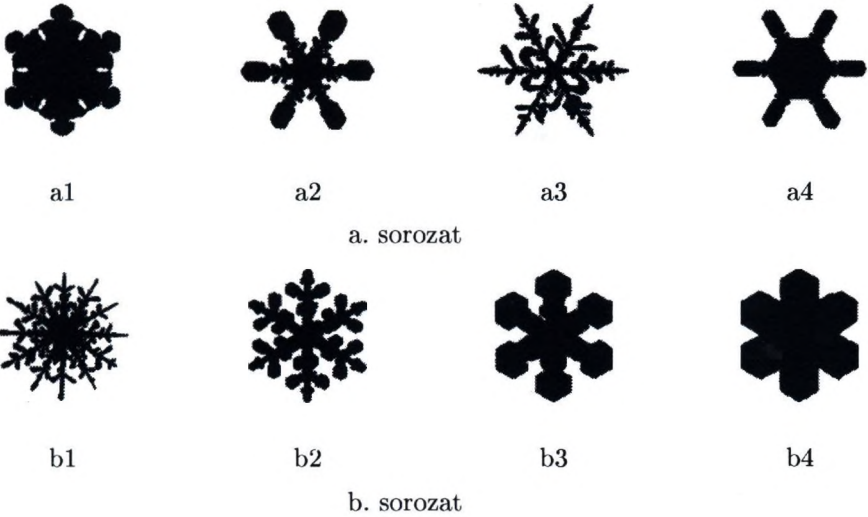
- s2 kristályosítási sebesség: 0,001 mm/s
- s4 kristályosítási sebesség: 0,003 mm/s

A fenti beállítások kombinálásával összesen hat kísérletet végeztünk el, és minden kísérletet 100-100 kép felhasználásával értékeltünk ki.



1. ábra. Succinonitrile-aceton dendritek (szürkekép és bináris kép)

A másik nagy alakzatsoprotot a hópelyhek alkották [11]. Ezek a képek gyakorlatilag nagy felbontású makrofotók, melyek sajnos azonban nincsenek kalibrálva (2. ábra).



2. ábra Hópelyhek (bináris képek)

Vizsgálataink harmadik részét különböző szimulációk során előállított dendrites képek jelentették. Két szimulációból összesen 1 illetve 4 dendritet vizsgáltunk meg (3. ábra). A c. szimuláció a molekuláris dinamika legújabb eredményeit felhasználva szimulálja a dendrites kristályosodást [8], a d. szimuláció pedig az ún. fázismező metódussal [9, 10] készült dendriteket mutatja be.



c1
c. szimuláció



d1



d2



d3



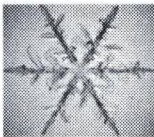
d4

d. szimuláció

3. ábra. Szimulált dendritek

Mivel a nagyítások nagyon különbözőek voltak, illetve egyes esetekben nem is ismertük a nagyítás értékét, ezért minden képet olyan méretűre alakítottunk, hogy egy 512x512 pixelnagyságú négyzetbe pontosan beférjenek.

A képátalakítások során az volt a célunk, hogy a dendritet elkülönítsük a környezetétől. Ez azt jelentette, hogy a kép fényerejét, kontrasztját addig változtattuk, amíg a dendrit határfelülete élesen el nem vált a háttértől. Ezek után a dendrit belsejét kitöltöttük fekete színnel. Egyes esetekben szükség volt a háttért is kitölteni, természetesen ez fehér színnel történt. Végeredményképpen minden esetben fehér alapon fekete dendriteket kaptunk. Az 4. ábrán bemutatjuk az átalakítások után kapott képeket.



hópehely, a3 kép



d. szimuláció



szimulált dendrit, d4 kép

4. ábra. Eredeti és az átalakított képek

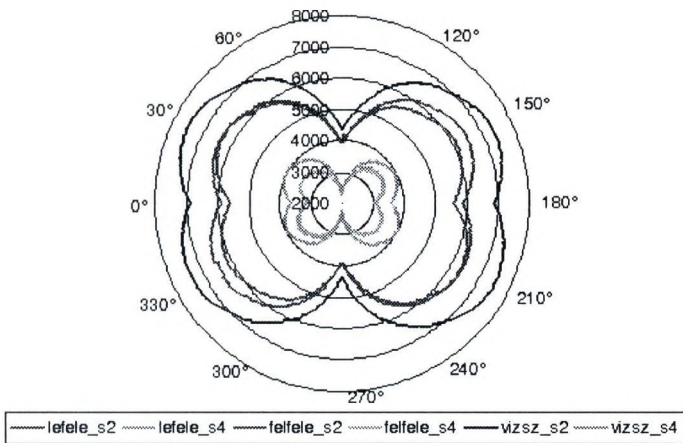
3. Eredmények

3.1. Orientációs rózsza diagram

Az orientációs rózsza diagram csak az általunk kristályosított dendrites szerkezetekre alkalmazható, mivel a körszimmetrikus alakzatokról nem ad használható információt.

Összesen hatféle beállítással végeztünk kísérleteket. A különböző beállításoknál kapott orientációs rózsza diagramok láthatóak az 5. ábrán.

A tendencia mindkét sebesség esetén ugyanaz. Amikor a kristályosítás iránya párhuzamos a gravitációs vektor irányával, akkor kisebbek a rózsza diagramok, mivel kevesebb metszés szám adódik (kevesebb és rövidebbek a szekunder dendritágak). A nagyobb sebességeknél a rózsza diagram kinyílik, és ha figyelmesebben megsejleljük, akkor látható, hogy bizonyos mértékben ki is kerekedik. Ha



5. ábra.

Az orientációs rózsza diagram alakulása az általunk kristályosított dendriteken

párhuzamosan kristályosítunk a gravitációs vektorral, akkor gyakorlatilag függetlenül az iránytól, közel hasonlóan néz ki a két diagram. A vízszintes irányú kristályosítás azonban mindkét sebességnél magasabb metszés számot, és ezáltal nagyobb rózsza diagramot produkál.

3.2. Orientációs faktor

Az orientációs faktor kiszámításának a képletéből adódik, hogy ez a paraméter sem alkalmazható körszimmetrikus alakzatok jellemzésére. A következő táblázatban az általunk kristályosított dendritek orientációs faktor értékeit vontuk össze.

1. táblázat. Az orientációs faktor értékei az általunk kristályosított dendriteken

orientációs faktor	a kristályosítás iránya és sebessége					
	Le s2	Le s4	Fel s2	Fel s4	Víz s2	Víz s4
	0,22	0,21	0,22	0,2	0,31	0,30

Ugyanaz olvasható ki az 1. táblázat számszerű értékeiből, mint az 5. ábra diagramjából. Az orientációs faktor utal az alakzat izotróp voltára, minél jobban közelít az értéke a nullához, annál izotrópabb az alakzat. Megfigyelhető, hogy mindkét sebesség esetén a vízszintes irányban kristályosított dendritek bírnak a legnagyobb irányítottsággal.

3.3. Saltykov-féle alaktényező

Az alább látható táblázatok bemutatják, hogyan alakultak az alaktényező értékei.

2. táblázat. Az általunk kristályosított dendritek Saltykov-féle alaktényezői

Saltykov-féle alaktényező	a kristályosítás iránya és sebessége					
	Le s2	Le s4	Fel s2	Fel s4	Víz s2	Víz s4
	0,14	0,19	0,14	0,18	0,12	0,31

3. táblázat. A hópehely dendritek Saltykov-féle alaktényezői

Saltykov-féle alaktényező	Hópehelyek, a. sorozat				Hópehelyek, b. sorozat			
	a1 kép	a2 kép	a3 kép	a4 kép	b1 kép	b2 kép	b3 kép	b4 kép
	0,30	0,27	0,18	0,40	0,13	0,19	0,32	0,44

4. táblázat. A szimulációval kapott dendritek Saltykov-féle alaktényezői

Saltykov-féle alaktényező	c. szimuláció		d. szimuláció			
	c1 kép	d1 kép	d2 kép	d3 kép	d4 kép	
	0,19	0,06	0,20	0,10	0,16	

Mint arról fentebb is szó volt, a Saltykov-féle alaktényező az alakzat körszerűségére utal, és kör esetén az értéke 1. Látható, hogy a hópehelyek sorozat a4, b3. illetve b4 ábrája (3. táblázat) közelít leginkább a körhöz, és ebben az esetben kaptuk a legnagyobb értékeket is a mérés során.

3.4. Fajlagos felület

A fajlagos felület értékeket mutatják az alábbi táblázatok. Az általunk kristályosított dendritek fajlagos felület értékei eléggé megbízhatatlanok voltak, elsősorban a képátalakításból adódó pontatlanságok miatt. A módszer ebben az alkalmazási esetben tökéletesítésre szorul.

5. táblázat. A hópehely dendritek fajlagos felület értékei

Fajlagos felület, 1/pixel	Hópehelyek, a. sorozat				Hópehelyek, b. sorozat			
	a1 kép	a2 kép	a3 kép	a4 kép	b1 kép	b2 kép	b3 kép	b4 kép
	0,05	0,05	0,06	0,03	0,09	0,07	0,05	0,04

6. táblázat. A szimulációval kapott dendritek fajlagos felület értékei

Fajlagos felület	c. szimuláció		d. szimuláció			
	c1 kép	d1 kép	d2 kép	d3 kép	d4 kép	
	0,05	0,22	0,05	0,12	0,08	

A képek alátámasztja azt a ténytet, hogy minél bonyolultabb, minél több szekunder, terciar ággal rendelkezik egy primer dendritág, annál nagyobb a fajlagos felülete. Amennyiben a szerkezet leegyszerűsödik, elveszti a finomságát, csökken a fajlagos felület értéke.

4. Összefoglalás

Az általunk kristályosított succinonitrile-aceton dendriteken bebizonyítottuk, hogy a dendrit alakja függ attól, hogy milyen szöveget zár be a kristályosodás iránya a gravitációs vektor irányával, illetve hogy milyen sebességgel történik a kristályosítás. Különbéle alakjellemzők segítségével megmutattuk, hogy vízszintes irányú kristályosodás esetén kapjuk a legdurvább, és legkevésbé izotróp dendriteket. A kristályosodási sebesség növekedésével is csökkennek a dendritek izotróp tulajdonságai.

A Saltykov-féle alakjellemző és a fajlagos felület mérésével különbséget tudunk tenni a szimulációkkal előállított dendrites szerkezetek morfológiai jellemzői között. Hasonló eredményre jutottunk, mintegy osztályozni tudtuk a különböző mértékben dendrites szerkezettel rendelkező hópelyheket.

A felhasznált paraméterek minden kétséget kizáróan alkalmasak a dendrites struktúrák morfológiai jellemzésére.

Hivatkozások

1. Glicksman M. E.: *Met. Trans. A*, Vol. 7A, pp. 1747-1759, November, 1976.
2. Réger M.: *Tranziens kristályosodási folyamatok kísérleti vizsgálata*, Budapest, 1997.
3. Bodenschatz E., Ragnarsson R., Utter B.: *Directional Solidification of Solvent/Polymer Alloys*, <http://milou.msc.cornell.edu/solidification.html>, 1999.
4. Cs. Pólliska, K. Tomolya, J. Kovács, Z. Gácsi, M. Réger: *Effect of Direction of Solidification on the Dendritic Structure*, *Materials Science Forum Vols. 329-330*, pp. 291-296, 2000.
5. *Sztereológia és Képelemzés*, egyetemi tankönyv, szerkesztette Gácsi Z., 2001.
6. Saltykov S. A.: *Stereometrische Metallographie*, VEB Deutscher Verlag für Grundstoffindustrie, Leipzig, 1974.
7. Weiss A., Zimmermann G., Rex S.: *Investigation on Morphological Instability during Directional Solidification on a Sounding Rocket*, *Third International Conference on Solidification and Gravity*, Hungary, Miskolc-Lillafüred, pp. 98-99, 1999.
8. Uehara T., Inoue T.: *A Molecular Dynamics Study on Solid-Liquid Interface and its Application to Phase-Field Simulation*. *Fifth World Congress on Computational Mechanics*. July 7-12, 2002, Vienna, Austria. Eds.: Mang H. A., Rammerstorfer F. G., Eberhardsteiner J. <http://force.energy.kyoto-u.ac.jp/uehara/publi.html>
9. Andersson C.: *Numerical simulation of dendritic solidification using a phase field model*. Licentiate's thesis. Department of Numerical Analysis and Computer Science. Royal Institute of Technology. Stockholm. (2000) p. 6. <http://www.nada.kth.se/christe/research/index.html>
10. Andersson C.: *Phase-field Simulation of Dendritic Solidification*. PhD thesis. Royal Institute of Technology. (2002) p. 51. <http://www.nada.kth.se/christe/research/index.html>
11. <http://www.its.caltech.edu/atomic/snowcrystals/>

Volume and surface models from CT/MR images

Ferenc Pongrácz and Gábor Renner

Computer and Automation Research Institute, HAS
pongaczf@axelero.hu, renner@sztaki.hu

Abstract. The paper presents image processing and model building procedures which are applied to the analysis of CT/MR medical image data stored in standard DICOM files. After evaluating DICOM data, the image series is stored in a voxel-based volume grid according to 16 bit grayscale range and sliced structure. Different spatial rearranging (reformatting) calculations are performed on the grid. Rigid-body and automatic registration algorithms are implemented to solve image fusion tasks. For the analysis of the shape and functionality of anatomical objects, continuous surface models based on surface points and grayscale gradients have been created. The mathematical representations of the continuous surfaces are computed by minimizing a functional, which is related to the squared distances of surface points and geometrical properties of the surfaces.

1 Introduction

3D analysis of anatomical objects and surgical planning create high demand for software tools which are suitable for converting raw DICOM data into a tractable form. Spread of modern navigation devices also influences the development of various methods toward fast 3D manipulation of CT/MR medical image sequences [1]. After locating image pixels at regular 3D grid points one important task is to realign grid planes as a function of a 3D cursor or, in a navigating situation, as a function of a 6 degree-of-freedom navigation device. Alternatively, the complementary task is to locate a target point in 3D by using several, spatially realigned volume grid samples.

In case of multiple modalities considered in the planning part of surgery, new problems of image fusion and registration arise. The actual data can create several difficulties with marker localization, surface reconstruction, voxel scaling, grid mapping, etc. The parameters of iso-surface reconstruction are important both for 3D scene visualization and collection of surface coordinates and grayscale gradients as well. The collected data from surface are used for creating continuous surface models which are important in clinical analysis of anatomical objects.

This article summarizes the results of our software development for constructing volume and surface models of CT/MR images and describes the first steps toward integrating these models into systems of computer surgery.

2 DICOM Images and Volume Models

CT/MR images are read in with a software modul suitable for the interpretation of standard DICOM tags in image header and handling different imaging formats. The read in procedure is divided into fast and slow versions: the fast version stores image pixels in a voxel grid without depth interpolation; the slow version makes depth interpolation and produces voxels with all edges identical to x-y pixel size of raw image. The volume model of the anatomical region under study is identical to the result of the actual sampling defined by the voxel size. Each vertices in a voxel are given a 16 bit grayscale value according to the original or interpolated values. The original volume model, due to its grid structure, can be resampled in different axis orders (Figure 1).

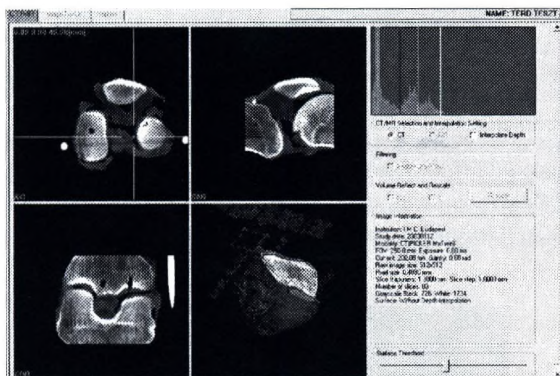


Fig. 1. CT/MR panel of the DICOM viewer and surgical planning program. Top left view shows the CT sequence according to original data accompanied with slice views from reformatted volumes. The iso-surface display is shown on the right bottom view.

The resampling procedure, usually referred as "reformatting" is performed by interpolation if the voxel size is not identical along the X-Y-Z axes (no depth interpolation added during read in). To meet speed and storage optimization conditions, this interpolation should be done "on the fly" i.e. only for the viewed slice position not for the whole volume grid. The slice viewing procedure is synchronized by cursor position shown on 3 orthogonal volume grid samples. This arrangement assures very fast localization of anatomical positions (Figure 2.).

3 Surface Reconstruction from Volume Grid Data

Iso-surface raytracing method with Phong illumination model is used to create surfaces from volume grid models according to a predefined value of grayscale threshold [2] [3]. To convert volume model into a rendered volume object on an OpenGL scene the grid size should be changed in case of non identical voxel edges. Several parameters are set for material model: opacity value, LUT

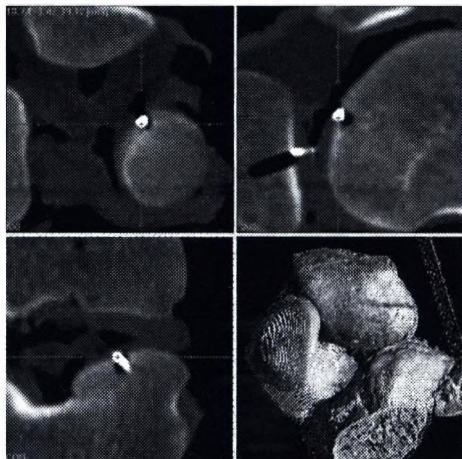


Fig. 2. Representation of the 3D cursor in 3 orthogonal volume grid samples. Marker selection in CT volume using enlarged views.

range, illuminations (diffuse, ambient, specular), threshold and shininess. Complex scenes of CT/MR volume objects and OpenGL graphical elements are created and viewed by OpenGL camera.

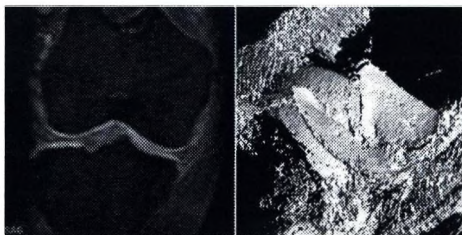


Fig. 3. Left panel shows a sagittal slice from the volume model of an MR sequence. Right panel displays a surface model after threshold-based segmentation of the knee contact region.

Our experience shows that the Hounsfield data stored in header of a single CT slice was usable as threshold range for the whole CT sequence (see surface view on Figure 2). However, MR sequences need minimum/maximum search in gray values before setting the threshold range for iso-surface raytracing (Figure 3).

4 Image Fusion and Registration Algorithms

Comparison of two image sequences is possible by implementing suitable registration algorithms and creating special image fusion views. In case of intermodality registration CT volume is always considered as the reference and the MR volume is chosen as "floating" volume to be registered. The voxel size matching is made for MR volume with 3D interpolation.

Marker-based registration of volumes is performed by known rigid-body algorithm [5]. In spite of the fact that many new registration methods have been recently developed, the marker-based approach is still widely used because of feasibility of accuracy tests of registration during image-guided surgery. The marker selection starts on the surface view by dropping the mouse location onto the surface model followed by projection into the 3 orthogonally sampled volume models. Fine tuning of marker coordinates is performed in the volumetric grid as shown in Figure 2. Three or four markers are localized this way. In case of 3 markers available, the 4th marker is set by calculation on the normal vector of the triangle defined by 3 real marker locations.

Mismatching order of registration markers between the two volumes is eliminated by a procedure based on the calculation of all possible distances between points. The method simply uses the permutation of point indices. The maximum number of index permutations is set according to point number and the optimum permutation in one of the volumes is sought. At the optimum, the correlation between the distances of centroid-based points in reference volume and points in floating volume is at maximum.

After successfully matching registration point pairs, the translational part of registration matrix is available from the 3D difference vector of centroids. The rotational submatrix is calculated according to [5] in a least-square sense by filling first the 3x3 matrix A with the centroid-based coordinates of markers:

$$a_{i,j} = \sum_{k=0}^n p_k^{i,R} p_k^{j,F} \quad i, j = 0 \dots 2 \quad (1)$$

where $p_k^{i,R}$ and $p_k^{j,F}$ are the i th and j th coordinates of points in reference and floating volumes, respectively and n is the number of markers. Using an SVD decomposition of A [4] the rotation submatrix can be calculated:

$$A = U \times [diag(w_j)] \times V^T \quad rot = V \times U^T \quad (2)$$

If the registration error is under the acceptance limit, the registered marker positions are projected into the reference volume and displayed (Figure 4.). Practice shows that SVD usually gives solution even if bad localization of markers occurs. However, handedness problems (correct ordering of coordinate axes) can arise from different slice orders of input sequences. This problem is solved during the read in procedure by automatic reslicing of DICOM data.

Markerless registration methods based on correspondences between graylevels of two volumes, measured by "mutual graylevel" information, are used lately in multimodality image fusion studies. Because of their clinical importance, these methods are currently intensively researched [6] [7] [8]. The method is able to align anatomical objects with different contrast properties as a consequence of the formulation of mutual graylevel information:

$$I(u(x), v(T(x))) = h(u(x)) + h(v(T(x))) - h(u(x), v(T(x))) \quad (3)$$



Fig. 4. Registration to a reference bone structure (femur of a knee) by 3 markers using the rigid-body method. 2 CT sequences of a cadaver knee are fused after rotating tibia. Crosses show markers on the reference volume and spheres are projected markers located on the registered volume. Blue signs represent the centroids. Transparency is added to one of the surfaces.

where $u(x)$ is the intensity level in the reference volume and $v(T(x))$ is the level in the floating volume observed through a coordinate transformation T . $h(u(x))$ and $h(v(T(x)))$ are the marginal entropies of signal probabilities in the overlapping regions. $h(u(x), v(T(x)))$ is the joint entropy. The maximizing mutual information is thought of as minimizing the joint entropy relative to the marginal entropies.

During implementation of the algorithm graphical tools are added to make registration between small subvolumes of the original CT/MR data. This approach is needed in those clinical studies where the 3D measures of relative displacement of different anatomical objects are important parameters. The simplex method is used for searching the maximum of I in (3), which sets 3 offset coordinates and 3 angles at each iteration and selects the best change towards optimum [4] [7]. The assessment of registration accuracy is difficult for markerless registrations. Mixed view representation of CT and MR modalities in the reference grid and compound surface model visualization can help but no quantitative measures are available. However, the initial results looks promising and the new approaches with combined mutual information and gradient information [8] can even support our effort to use this method for restricted 3D sections of CT/MR volumes (Figure 5.).

5 Continuous Surface Models from Volume Grid Data

For detailed analysis of shape and functionality of anatomical objects continuous surface models are usually required. Surface points extracted from isosurface models represent only rough shape information of the object to be investigated. The limitations of isosurface models extracted from direct volume rendering found especially apparent in studies of shape and contact properties of near surfaces. Therefore, we use continuous surface fits to in hardly detectable regions with complicated shape.

Continuous surface fit consists of two main parts. First point subsets belonging to different but close to each other surfaces must be carefully separated. This is done by the combination of point set segmentation and region growing

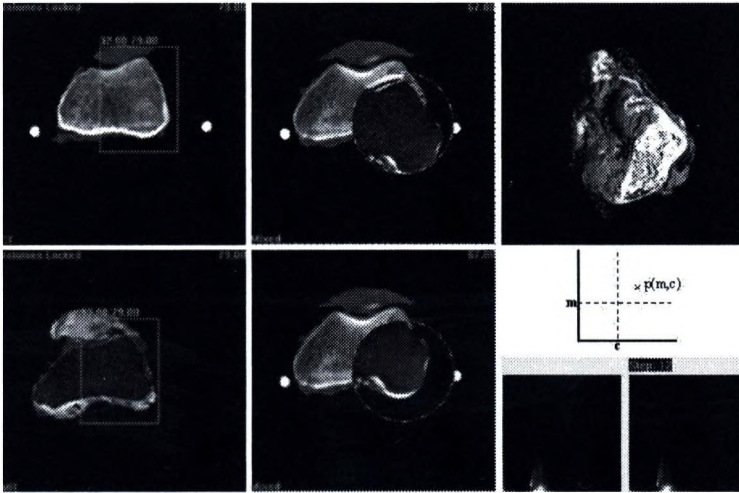


Fig. 5. Fusion of CT and MR modalities of the cadaver knee by automatic registration. First column: selecting registration subvolumes in CT and MR (selected slice range is shown). Second column: Unregistered and registered mixed views. Third column: surface view of fused models and joint histograms of probabilities of gray values before and after registration. Bright areas in the image of histogram correspond to large numbers of voxels with those MR and CT intensities. The strong vertical lines near the centre of the distribution are resulted from soft tissue features differentiated by MR in contrast to a small range of CT intensities. Small changes, especially in the low intensity region, can be seen in the histogram after registration.

techniques. Then the mathematical representation of the continuous surface is created. Subsequently we describe basic ideas for the two processes.

Segmentation of surfaces is based on surface points corresponding to an appropriate gray threshold (isosurface model). To facilitate segmentation of points belonging to surfaces which are near to each other or even touch each other we compute graylevel gradients in surface points from the volume model as well. Now we assume, that surfaces represent boundaries of different structures with different averages of graylevels, and there exists a global orientation for each structure. Consequently, gradients have alternate directions on opposite boundary surfaces. Now we investigate columns of small cross section (pixel size) and with orientation approximately perpendicular to the global orientation. Points in this column can be sorted as consecutive surface points, by taking increasing (or decreasing) distances in the column and alternate normal directions. In this way rough classification of the point set is possible. The remaining points are connected to already existing surface point sets by a region growing algorithm. Each point is glued to the nearest surface with the same normal orientation.

In Figure 6 point set is shown, which corresponds to several isosurfaces generated from a volume model (bone and cartilage surfaces of a joint). The surfaces are close to each other, some of them touch each other. Because amplitude of noise is in the magnitude of distances between the surfaces, separation using

purely positional information is not possible. Including neighbourhood and gradient information (normals to the surfaces) additionally, however, makes possible to group points belonging to the individual surfaces. Figure 6 also demonstrates points of one of the intermediate surfaces of the structure.

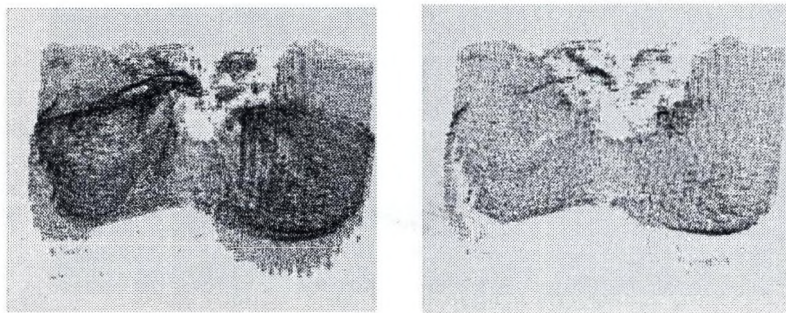


Fig. 6. Left shows isosurface points of four anatomical surfaces, right shows one intermediate surface points, extracted by segmentation.

Continuous surface fit starts with topological ordering of points belonging to one separated surface. This is done by creating a triangulation over the surface points. A good triangulation must meet several requirements: it must be topologically correct, must eliminate outlier points, triangles must have comparable side lengths and angles, their size must reflect the curvatures of the surface, etc. Automatic generation of triangular structures is difficult, we use a method described in [9]. For many application (e.g. graphical visualisation) triangular representation is sufficient. For more detailed investigation and analysis we may need surfaces with different degree of continuity.

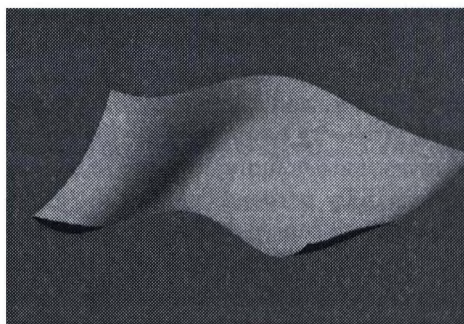


Fig. 7. Continuous surface for the point set of Figure 6 right.

For mathematical description of surfaces with high degree of continuity we use parametric surfaces commonly used in computer graphics and CAD (Bézier, B-spline, NURBS surfaces). Consequently, we have to parametrize the data points;

we associate parameter values of the continuous surface to them. This is usually done by projecting points onto a rough approximating surface [10].

The mathematical representation of continuous surfaces are computed by minimizing a functional. This contains the squared distances of data points to the continuous surface, and some geometrical quantities reflecting continuity and smoothness. The minimization is a complicated nonlinear iterative process, and outputs the describing parameters of the continuous surface. We use a tolerance driven method with parameter correction, the detail of which are described in [10]. Figure 7 shows the surface fitted for the previously separated isosurface point set displayed in Figure 6.

6 Conclusions

An overview and results of practical approach are given for medical DICOM data handling. The data processing starts by reading in DICOM files and displaying multimodality views of orthogonally reformatted volumetric models. The accurate and fast 3D localization in volumetric models help the topographic analysis of anatomical objects. Image fusion algorithm is implemented for 3D alignment of different volumetric models of the same structure. For the analysis of near surfaces continuous surface models were created.

Acknowledgements

The authors thank László Szobonya and Zoltán Bárdosi for giving support in programming work.

References

1. R.H. Taylor, S. Lavallée, G.C. Burdea and R. Mösges. *Computer-Integrated Surgery, Technology and Clinical Applications*. MIT Press, Cambridge, 1996.
2. J. Foley, A.van Dam, S.K. Feiner and J.F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1992.
3. VGL Online Manual. Documentation for VGL 3.1 generated at 30 Jun 2001 (Volume Graphics GmbH, Heidelberg).
4. Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge U. P., p. 994, 1992.
5. Arun, K.S., Huang, T.S. and Blostein, S.D.: Least square fitting of two 3-D point sets. *IEEE Trans.Patt.Anal.Machine Intell.* Vol. PAMI-9 (5), pp. 698-700, 1987.
6. Wells III,W.M., Viola,P., Atsumi,H., Nakajima,S. and Kikinis,R. Multimodal volume registration by maximization of mutual information. *IEEE Trans. Medical Imaging* 16(2), pp.187-198, 1997.
7. Studholme, C., Hill, D. L.G. and Hawkes, D.J.: Automated 3-D registration of MR and CT images of the head. *Medical Image Analysis.* 1(2), pp. 163-175, 1996.
8. Pluim,J.P.W., Maintz,J.B.A. and Viergever,M.A.: Image registration by maximization of combined mutual information and gradient information. *IEEE Trans. Medical Imaging* 19(8), pp.1-6, 2000.
9. Kos, G., Várady, T. : Parametrizing large triangular meshes, I. Magyar számítógépes grafika és geometria konferencia, Budapest, 2002. pp. 48-50
10. Weiss, V., Andor, L., Renner, G., Várady, T: Advanced surface fitting techniques, *CAGD*, Vol. 19, (2002), pp. 19-42

Körmetszeteket tartalmazó tárgyak rekonstrukciója néhány vetületből

Rodek Lajos¹, Kiss Zoltán¹, Kuba Attila¹

Szegedi Tudományegyetem, Képfeldolgozás és Számítógépes Grafika Tanszék
¹{rodek, kissz, kuba}@inf.u-szeged.hu
6720 Szeged, Árpád tér 2.,
06-62-544-000/6399

Kivonat Nemroncsoló anyagvizsgálat végzésekor jelentkezett a következő probléma: Rekonstruáljunk egy olyan homogén anyagú 3D tárgyat vetületi képeiből (pl. röntgen vagy neutronos felvételekből), amelyről tudjuk, hogy egy csőből és azon belül néhány tömör gömbből vagy hengerből áll (pl. csapágyak, csövek). A megoldásnál vegyük figyelembe azt, hogy a vetületi képek zajjal terheltek, továbbá szeretnénk a rekonstrukciót minél kevesebb számú vetületből elvégezni.

Megoldásként egy olyan eljárást választottunk, amely a rekonstrukciós problémát optimalizálásként fogalmazza meg. Keressük hengereknek és gömböknek azt a konfigurációját, amelynek a vetületei a lehető legkisebb mértékben térnek el a megadott vetületi adatoktól. Az eltérés mértékéért a négyzetes eltérést választottuk. A futási sebesség növelése és a rekonstrukció sikerességének biztosítása érdekében az eljárás egy alkalmas induló konfiguráció megkeresésével kezdődik. A szimulált lehűlés elvét használó algoritmust implementáltuk ennek az optimalizálási problémának a megoldására. Hogy a módszer hatékonyságát vizsgálhassuk, a programot beillesztettük a tanszéken fejlesztett DIRECT rendszerbe, amely diszkrét tomográfiai problémák megoldására alkalmas keretrendszer. Beszámolunk a program működésével kapcsolatos előzetes tapasztalatainkról és az eddig elért eredményeinkről.

1. Bevezetés

A következő problémát a berlini Hahn–Meitner Intitut kutatási atomreaktor munkatársai vetették fel: Adott egy üreges henger, melyben néhány tömör gömb található, ezek feltételezhetően kopó alkatrészek. A henger zárt, azaz a benne lévő gömbök helyzetéről csak valamilyen nemroncsoló (vetületi) képalkotó eljárással lehet információt szerezni. Erre a célra ők neutron leképezést használtak Neutronsugárral átvilágítva egy tárgyat hasonló képet kapunk, mint röntgen sugárzást használva. A lényegi különbség az, hogy a neutronsugárzást a fémek kevésbé nyelik el.

Egy ilyen vetületi kép elkészítése azonban igen időigényes és drága művelet így törekedni kell egy olyan rekonstrukciós eljárás kidolgozására, amely kevés-

Ezt a kutatást az OTKA T/12 032241 támogatta.

2–4 – vetületi képből viszonylag jó eredménnyel tud egy elfogadható 3D modellt előállítani.

Hasonló gyakorlati probléma merült fel a AEKI–KFKI-ben (Atomenergiai Kutatóintézet, Kísérleti Fizikai Kutatóintézet) is. Ebben az esetben a vizsgált tárgy egy tömör henger volt, amelybe kör keresztmetszetű lyukakat fúrtak.

Kezdetben mindkét problémát 3D-ről 2D-re visszavezetve közelítettük meg. Ebben az esetben a feladat redukálható a tárgy egy-egy síkmetszetének rekonstrukciójára, bemenetként 1D vetületi vektorokat használva. Eredményeinkről beszámoltunk konferenciákon [5,6,8], továbbá publikációként is összefoglaltuk azokat [7]. Az 1. ábra egy síkmetszet rekonstrukcióját mutatja klasszikus módszerrel és a saját eljárásunkkal.



1. ábra.

Sikeresen rekonstruált síkmetszet (berlini adatokból). **a)** 60 vetületből visszavetítéssel kapott kép (az intézettől kapott képről hiányzik a körgyűrű egy része); **b)** 4 vetületből saját módszerünkkel nyert kép

A 2D részfeladat vizsgálata során szerzett tapasztalatokra építve terjesztettük ki módszerünket az eredeti 3D feladat megoldására, és a továbbiakban is csak a 3D problémát fogjuk vizsgálni. Így a vetületek 2D mátrixok lesznek. Az általunk kidolgozott algoritmus tehát egy speciális rekonstrukciós eljárás, melyben kihasználtuk azt a plusz információt, hogy a rekonstruálandó 3D objektum egy csőből és azon belül néhány tömör gömbből vagy hengerből áll.

Az eljárás bemenetei a vetületi mátrixok, kimenete pedig egy 3D modell (kép) lesz. A megoldás során feltételeztük, hogy ténylegesen ismerjük a tárgyat alkotó gömbök ill. hengerek számát. Szintén feltettük, hogy a test minden objektuma homogénnek tekinthető, továbbá hogy csak négyféle anyag fordulhat elő: a tárgyat körbevevő levegő/vákuum, a cső anyaga, annak hengeres belsejét kitöltő anyag, ill. a gömbök és hengerek anyaga.

Tomográfiának nevezzük a képfeldolgozásnak azt a szakterületét, amely képeknek azok **vetületeiből** történő rekonstrukciójával (visszaállításával) foglalkozik. Ha a képfüggvény értékkeszlete véges sok, ismert elemet tartalmaz, **diszkrét tomográfiáról** beszélünk. A korábbi megállapítások alapján a diszkrét tomográfia jó módszernek tűnik a felvetett probléma megoldására. A sikeres rekonstrukció megköveteli, hogy az anyagokat jellemző **elnyelődési együtthatók** értékét legalább közelítőleg ismerjük. Az ily módon rekonstruált 3D képek négyértékűek lesznek.

Az algoritmus része a Szegedi Tudományegyetemen fejlesztett diszkrét rekonstrukciós technikákat összefogó DIRECT rendszernek [4].

2. A probléma buktatói, nehézségek

A rekonstrukció elvégzését több tényező is megnehezíti. Ezek közül a legfontosabb a rendelkezésre álló vetületek kevés száma, valamint a vetületeket terhelő zaj jelenléte. A **zaj** főleg az alkalmazott sugárzás fizikai tulajdonságaiból és a detektálás módjából fakad, viszont ennek jellemzőiről nem sokat lehet tudni. Ez a két dolog „garantálja”, hogy a klasszikus, általános tomográfiai algoritmusok – úgymint visszavetítés és Fourier-térbeli rekonstrukció – használhatatlanok legyenek a feladat megoldására.

Mint sok matematikai problémánál, a rekonstrukciónál is felmerül az egzisztencia (létezik-e megoldás) és az unicitás (egyértelmű-e a rekonstrukció) kérdése. A zaj miatt természetesen nem beszélhetünk szigorú értelemben vett egzisztenciáról. Ezen felül, még ha a zajtól el is tekintünk, a megoldás(ok) létezésének eldöntése nem oldható meg hatékonyan (ti. polinomiális időben).

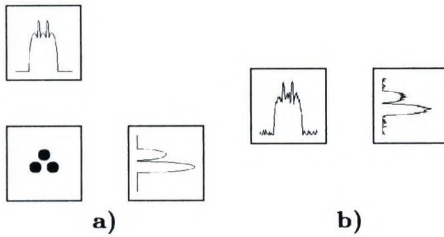
Az egyértelműség kérdését ugyancsak nehéz megválaszolni az ún. **kapcsoló komponensek** (pontosabban komponenspárok) létezése miatt. A 2D esetet tekintve, és bináris képet, valamint vízszintes és függőleges irányú vetületeket feltételezve erre mutat példát a 2. ábra. A függőleges vetítés során a kép oszlopösszegeit tartalmazó vektort kapjuk meg, míg a másik irány a sorösszegeket eredményezi. Mindkét kép 2–2 azonos sugarú körlapot tartalmaz, melyek középpontja egy négyzet átellenes csúcsain helyezkedik el (a két látszólagos négyzet nagysága és helyzete is megegyezik mindkét képen). Könnyen látható, hogy bár a két kép különböző, vetületeik azonosak lesznek. Ebben az esetben tehát pontosan két helyes megoldása van a rekonstrukciós problémának.



2. ábra. Egy 2D kapcsolókomponens-pár

A sok negatívum mellett két nagyon fontos segítségünk azért van. Egyrészt tudjuk, hogy a rekonstruálandó 3D képen egy cső valamint néhány, páronként diszjunkt gömb vagy henger található, továbbá bármely pixel értéke 4 adott érték valamelyike lehet. Azonban még ez sem elegendő ahhoz, hogy a feladatot pusztán geometriai úton oldhassuk meg. Egyrészt a zaj miatt a mért vetületek alakja az elméleti vetületekéhez képest jelentősen torzulhat. Másrészt ha egy vetítősugár kettő vagy több objektumon is áthalad (azaz a gömbök ill. hengerek **egymásra vetülnek**), akkor az adott vetületen a keresett objektumoknál kevesebb, de akár több maximumhely is megjelenhet. Ilyen eseteket mutat a 3. a) képen látható tárgy. A tárgy 3 gömböt tartalmaz, amelyek középpontja egy síkban fekszik. A képen az ezzel a síkkal vett keresztmetszet, valamint a neki megfelelő vízszintes és függőleges vetületek tekinthetők meg. Amint látható, a 3 gömb ellenére a vízszintes vetületen 2, a függőlegesen 5 maximumhely jelenik meg! Még rosszabb

a helyzet, ha 10%-nyi mesterséges zajt adunk a vetületekhez, amint ezt a 3. b) ábra is mutatja.



3. ábra.

Gömbök egymásra vetülése. **a)** a tárgy egy síkmetszete és két zajmentes vetülete; **b)** 10%-os zajjal torzított vetületek

3. Megvalósított algoritmus

A diszkrét tomográfiai rekonstrukciós módszereknek nagy irodalma van (ld. pl. [1]). Mi a vizsgált probléma megoldásaként egy *hibrid* eljárást választottunk: ez egyrészt a rekonstrukciót optimalizálásként fogalmazza meg, másrészt a rekonstruált 3D kép tartalmára és geometriai szerkezetére komoly megszorításokat vezet be.

Az optimalizáció alkalmazását a 2 fejezetben említett tényezők motiválták. Még ha a gyakorlatban előforduló zajtól el is tekintünk, a rekonstrukció sokszor akkor sem oldható meg korrektül, akár geometriai, akár egyéb módszerrel próbálkozunk.

Bármely gömböt egyértelműen meghatároznak a középpontjának koordinátái és a sugara. Ezeket az adatokat közös néven az adott **gömb paramétereinek** hívjuk. A hengerek hasonló módon paraméterezhetők. A tetet alkotó összes objektum paramétereinek összességét **paramétertérnek**, a tér elemeit pedig **konfigurációknak** nevezzük.

Ezek alapján az optimalizálás a paramétertérben folyik. Feladata, hogy egy **kezdő konfigurációból** kiindulva **globális** optimumot találjon, amely a lehető legjobban megközelíti a valódi (keresett) konfigurációt. Ez utóbbit természetesen nem ismerjük, de a megoldás pontosságát az eredeti vetületek és az aktuális konfiguráció vetületeinek összehasonlításával is megállapíthatjuk. Az eltérés mértékéeként a négyzetes eltérést választottuk, ez lesz tehát a minimalizálandó célfüggvény. A kezdő konfiguráció megállapításával idáig nem foglalkoztunk. Erről később egy külön programrész fog gondoskodni, hasonlóan a 2D-ben bevált esethez. Ideiglenesen a kezdőállapotot mi adjuk meg.

A célfüggvény minimalizálása egy iteratív eljárással történik, amely az aktuális konfiguráció (állapotgráfbeli) szomszédjai közül randomizáltan választja ki az optimum következő közelítését. Az érvénytelen konfigurációkat feltétel nélkül elutasítjuk. Érvényes konfiguráció esetén meghatározzuk a célfüggvény értékét

erre a konfigurációra. Ha a célfüggvény értéke javult (azaz csökkent) az előző iterációhoz képest, az új állapotot megtartjuk, ez lesz az optimum új közelítése. Ha a célfüggvény egy adott korlát alá csökkent, megállhatunk, megtaláltuk az optimumot. Egyébként új iteráció kezdődik.

A globális minimum elérése érdekében az előbbi algoritmust kiegészítettük az ún. **szimulált hűtés** technikájával [2,3].

Az algoritmust megvalósító programrendszert beillesztettük a tanszéken fejlesztett DIRECT rendszerbe [4], amely diszkrét tomográfiai problémák megoldására alkalmas keretrendszer. A kiinduló vetületek és a rekonstruált kép is DIRECT formátumban kerülnek eltárolásra.

Az algoritmus hatékonyságának és sikerességének tesztelése érdekében a programrendszert 3 különálló, de egymással szorosan együttműködő részre bontottuk:

- A PROJCONV program feladata a vizsálat során nyert vetületek átalakítása DIRECT formátumúvá.
- A PROJGEN program segítségével forrás vetületeket generálhatunk, amelyeket azután a rekonstrukcióhoz közvetlenül felhasználhatunk. Mind manuális (fix konfiguráció), mind automatikus (véletlenszerű konfiguráció) üzemmódban használható.
- A SPECREC program végzi a tényleges rekonstrukciót. Ennek fontosabb szakaszai:

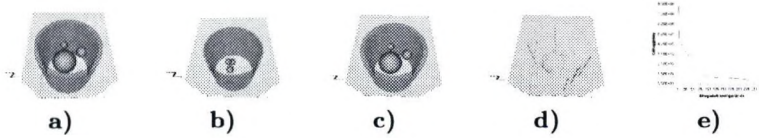
4. Eredmények

Az algoritmus működésének hatékonyságát többféle forrásból származó adatokon kipróbáltuk.

A 4. ábra egy olyan tárgy rekonstrukciójának menetét mutatja, amely három általános helyzetű gömböt tartalmaz. A tárgy paramétereit ill. a vetületeket a programunk generálta. A rekonstrukció négy vetületből történt, 10%-os zajszint mellett. Az ábra a) képe a keresett konfigurációt, b) képe a kezdőállapotot, c) képe pedig a rekonstruált eredményt mutatja. A rekonstrukció sikerességét jól tükrözi a d) képen látható különbség-konfiguráció, amely alig mutat eltérést a keresett tárgyhoz képest. Szintén figyelemre méltó, hogy nagyon jó eredményt kaptunk az aránylag rossz kezdőállapot dacára. A rekonstrukció kb. hét percig tartott. Ez alatt több mint 30000 konfigurációt próbált ki a program, amiből végül kb. háromszázat fogadott el. A célfüggvény értékének változását szemlélteti az ábra e) képe.

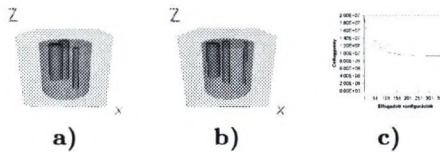
A szimulációs tesztek mellett valós, fizikai mérésekből származó adatokat is felhasználtunk. A 5. ábra annak a műanyag tárgynak a rekonstrukcióját mutatja be, amelyet a KFKI-ban vizsgáltak. A tárgy egy tömör hengerből és az abbé fűrt három hengeres furatból áll. A rekonstrukcióhoz itt is négy vetületet használtunk, a becsült zajszint pedig 10-15%-nak adódott. Az a) kép az általunk megadott kezdőállapotot, míg a b) kép a rekonstruált végeredményt mutatja. Ez utóbbi igen jól közelíti a tárgy valódi szerkezetét. Ez a rekonstrukció igen sokáig, több mint 17 percig tartott. Közben több mint 65000 konfiguráció kerül

kipróbálásra, amiből kb. négyszázat fogadott el a program. A hosszú futásidő tapasztalataink szerint a durva kezdőállapotnak tudható be. A célfüggvény értékének változását mutatja az ábra c) képe.



4. ábra.

Sikeresen rekonstruált szimulációs tárgy. **a)** a keresett konfiguráció; **b)** a kezdő konfiguráció; **c)** a rekonstruált konfiguráció; **d)** az a) és c) konfigurációk különbsége; **e)** a célfüggvény változása



5. ábra.

Sikeresen rekonstruált műanyag tárgy (KFKI). **a)** a kezdő konfiguráció; **b)** a rekonstruált konfiguráció; **c)** a célfüggvény változása

Hivatkozások

1. G. T. Herman and A. Kuba (Eds.): Discrete Tomography: Foundations, Algorithms and Applications. Birkhäuser: Boston, MA. (1999)
2. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller: Equation of state calculation by fast computing machines. J. Chem. Phys. **21** (1953) 1087–1092
3. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi: Optimization by simulated annealing. Science **220** (1983) 671–680
4. <http://www.inf.u-szeged.hu/~direct>, a DIRECT rendszer honlapja
5. Kiss Z., Kuba A., Rodek L.: Körmetszeteket tartalmazó tárgyak rekonstrukciója néhány vetületből. KÉPAF Konferencia kiadvány, Domaszék (2002) 253
6. A. Kuba, L. Ruskó, Z. Kiss, L. Rodek, E. Balogh, S. Zopf, A. Tanács: Preliminary Results in Discrete Tomography Applied for Neutron Tomography. COST Meeting on Neutron Radiography, Loughborough, England (2002)
7. A. Kuba, L. Ruskó, L. Rodek, Z. Kiss: Preliminary Studies of Discrete Tomography in Neutron Imaging. IEEE Trans. on Nuclear Sciences, közlésre benyújtva
8. A. Kuba, L. Ruskó, L. Rodek, Z. Kiss: Application of Discrete Tomography in Neutron Imaging. Proc. of 7th World Conference on Neutron Radiography, Rome, Italy (2002), közlésre elfogadva

Lézeres felületötvöztési technológia ellenőrzése sztereológiai vizsgálatok segítségével

Svéda Mária¹, Búza Gábor², Kovács Árpád³, Roósz Anrás³,

MTA-ME Anyagtudományi Kutatócsoport

¹{femmaria@uni-miskolc.hu

Bay Zoltán Anyagtudományi és Anyagtechnológiai Intézet

²

Miskolci Egyetem, Fémteni Tanszék

³

Kivonat Lézeres felületötvöztéssel Al-Si alapanyagon ólmot és bizmutot tartalmazó monotektikus felületi réteget alakítottunk ki. Az így előállított két rétegfű anyagot siklócsapágyként lehet a gyakorlatban alkalmazni. A sűrűlási tényezőt nagymértékben befolyásolja a lézeres felületkezeléssel létrehozott rétegben kialakuló ólom és bizmut cseppek mérete, eloszlása és mennyisége.

A felületi réteget scanning elektronmikroszkóppal vizsgálva a visszaszórt elektronokkal alkotott képeken jól elkülöníthetők a világos cseppeknek látszó magasabb rendszámú elemek (pl. Pb(Z=82), Bi (Z=83)) az alacsonyabb rendszámú sötét területnek látszó mátrixtól (pl. Al (Z=13), Si (Z=14)). A képeken Quantimet 570 C képelemző segítségével meghatároztuk az ólom illetve bizmut cseppek területhányadát illetve méret szerinti eloszlását.

1. Bevezetés

A monotektikus ötvözetrendszereket gyakran siklócsapágyak anyagaként alkalmazzák. Ezeknek a siklócsapágyaknak az előállítása történhet porkohászati úton [1-3], strip casting eljárással [4-7], planar flow casting eljárással [8] és rheocasting eljárással [9-11]. A siklócsapágyak előállítására lézeres felületkezelési technikát választottuk.

A lézeres felületkezeléssel előállított monotektikus ötvözeteknél nagy sebességgel történik a kristályosodás és olyan szerkezet alakul ki, ahol finom lágy kiválások találhatóak, amelyek szilárd kenőanyagként szolgálnak. A lézeres ötvöztés lényege, hogy a darab teljes tömegének tulajdonságait változatlanul hagyva csak a felületi réteg tulajdonságait változtatjuk meg.

A ma alkalmazott siklócsapágyak leggyakoribb anyag: ólom-ón-bronz és alumínium-réz-ón. Ezek a siklócsapágy anyagok már az igénybevételi határukat elérték, ezért van szükség nagyobb igénybevételű siklócsapágyak fejlesztésére

2. Kísérletek

A monotektikus felületi rétegek kialakítására a Bay Zoltán Anyagtudományi és Anyagtechnológiai Intézet 5 kW teljesítményű CO₂ lézerberendezését használtuk fel. Mindkét esetben Al-11%Si ötvözetet használtunk, mert az ilyen ötvözetekből gyakran készítenek különféle gépalkatrészeket. A próbadarabok mérete 15x20x130 mm, a lézerrel kezelt felület: 15x130 mm volt.

A bevonat beolvasztásánál az Al-Si hipoeutektikus ötvözet felületére kb. 0,1 mm vastagságú ólomréteget vittünk fel galvanizálással, majd különböző lézerteljesítménnyel (2; 2,5; 3; 3,5 kW) és 600 mm/min előtolási sebességgel összeolvasztottuk. A lézeres kezelés során 8 lézeres nyomot készítettünk 50%-os átfedéssel. A lézer fókuszátmérője 2 mm volt. A mintadarabokat grafitsprayvel vontuk be az alumínium ötvözetek kis abszorpciója miatt.

A por befűvások technológiánál a lézersugár által megolvasztott rétegbe Ar vívőgázzal, 40-60 μ m átmérőjű Bi szemcséket fűjtünk, különböző lézerteljesítménnyel (2; 2,5; 3; 3,5 kW) és különböző előtolási sebességek (300; 600; 900 mm/min) mellett. A lézeres kezelés során 5 lézeres nyomot készítettünk 50%-os átfedéssel. Az így kialakított monotektikus felületi rétegek megismerésére metalográfiai és scanning elektronmikroszkópos vizsgálatokat alkalmaztunk.

3. Próbaelőkészítés és vizsgálati módszerek

A fent említett módon kialakított monotektikus felületi rétegeket vezető műgyantába való beágyazás után, csiszoltuk, majd azokat az átolvasztott sávokra merőlegesen elvágtuk. A felületet nedvesen csiszoltuk, gyémánt pasztával políroztuk, majd 0.5% HF vizes oldatában marattuk.

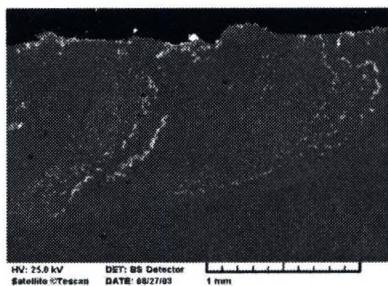
A bizmut és ólom részecskék mennyiségét, méret szerinti eloszlását Quantimet 570 C képelemző berendezéssel határoztuk meg. A mérésekhez scanning elektronmikroszkóppal készített visszaszórt elektron képeket használtunk. A visszaszórt elektron képeken a mátrixot alkotó kis rendszámú Al (Z=13) és Si (Z=14) sötétnek a monotektikus cseppeket alkotó nagy rendszámú Bi (Z=83) és Pb (Z=82) világosnak látszanak.

4. Eredmények

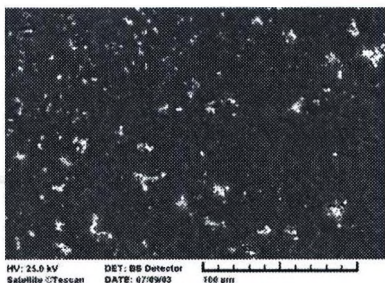
4.1. Ólom és bizmut eloszlás sajátosságai

Az 1. ábra a lézeresen felületötvözéssel készült Al-Si-Bi monotektikus felületi réteg átolvasztott zónáit mutatataja be.

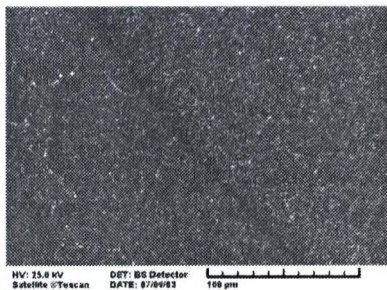
Az ólom és bizmut részecskék eloszlását az átolvasztott zónákban különböző nagytítások mellett a 2-5. ábrák szemléltetik.



1. ábra. Al-Si-Bi monotektikus felületi réteg.



2. ábra Bi részecskék eloszlása az átolvasztott zónában

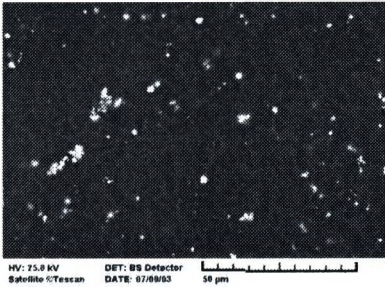


3. ábra Pb részecskék eloszlása az átolvasztott zónában

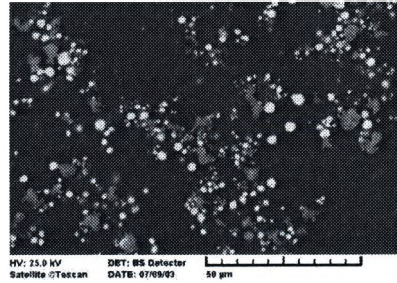
A nagyobb bizmut részecskék mellett kisebb néhány [106d]m-es bizmut részecskék helyezkednek el "felhőt" alkotva. Miközben a bizmut részecskék a felületi rétegbe beépülnek az áramlás miatt, a bizmut atomok illetve kisebb bizmut részecskék válnak le. A gyors kristályosodás miatt a részecskéket az áramlás nem tudja messzire vinni, így azok a nagyobb méretű részecskék mellett úgynevezett "felhőt" alkotva helyezkednek el. (2. ábra). Az ólom részecskék esetében ezt a jelenséget a nagyobb lézerteljesítménnyel készült próbadarabok estében tapasztaltuk.

Az ólom és bizmut részecskék alakját mutatja a 4-5 ábrák. Az ólomcseppek gömbölyű formát, a bizmut cseppek szögletes formát vettek fel, amely abból adódik, hogy a bizmut poliéderesen kristályosodik.

Az ólomcseppek estében energiadiszipatív mikroszondával meghatároztuk a cseppek összetételét. A fehér cseppek tisztán ólom tartalmúak, míg a szürke cseppek alumíniumot, ólmot és szilíciumot is tartalmaznak. (5. ábra)



4. ábra Bi részecskék eloszlása az átolvasztott zónában



5. ábra Pb részecskék eloszlása az átolvasztott zónában

Minden próbadarab estében az átolvasztott zónák területeiről nagyobb nagyítású képet készítettünk, hogy a képelemzős méréseknél a detektálási lépést segítsük. A Quantimet 570 C képelemző segítségével meghatároztam az ólom és bizmut részecskék területét, illetve méret szerinti eloszlását.

Az ólom és bizmut részecskék képelemző berendezés segítségével meghatározott átlagos terület %-a 2,6 és 3,1 %, és a sűrűségek figyelembevételével számított tömeg % 10,1 és 10,5 % volt.

Az 1. táblázat az ólom és bizmut részecskék tömeg% értékét mutatja be minden próbadarab estében.

1. táblázat. Az ólom és bizmut részecskék átlagos tömeg%-a

	ólom t%	bizmut t%		
	600 mm/min	300 mm/min	600 mm/min	900 mm/min
2 kW	6,2	5,2	8,6	9,8
2,5 kW	10,1	9,2	10,5	12,4
3 kW	8,9	7,4	9,1	11,8
3.5 kW	7,5	6,9	7,8	8,9

A 2. és 3 ábra az ólom és bizmut részecskék legjellegzetesebb méret szerinti eloszlását mutatja. Jól látszik, hogy az ólom részecskék leggyakoribb mérete 5-10 μm között található. A bizmut részecskék méret szerinti eloszlására az a jellemző, hogy jelentős részük 3-8 μm közötti mérettartományba esik.

5. Összefoglalás

A lézeres felületövezési technológia sztereológiai vizsgálatokkal történő ellenőrzése során megállapítottuk, hogy a bevonat beolvasztásos technológiával készült daraboknál a kisebb (2,5 kW) teljesítmény alkalmazása során a legegyszerűsebb ólom eloszlást kaptunk, a részecskék leggyakoribb mérete 5-10 μm között található és a bevitt ólom mennyisége átlagosan 10,1 tömeg %.



2. ábra. Ólom részecskék méret szerinti eloszlása.



3. ábra. Ólom részecskék méret szerinti eloszlása.

A porbefűvások technológiával készült daraboknál megállapítható, hogy a 2,5 kW teljesítmény és a 600 mm/min próba mozgási sebesség alkalmazása során a legegyszerűsebb bizmut eloszlást kaptunk, a részecskék leggyakoribb mérete 3-8 µm között található és a bevitt bizmut mennyisége átlagosan 10,5 tömeg %.

Az így bevitt ólom és bizmut mennyisége megfelelő a kellő siklási tulajdonságok biztosításához.

Hivatkozások

1. K. Nishiyama et al; Japan Soc. Power Metall. (1993) p. 645
2. D. P. Howe, A. A. Torrance, J. D. Williams; Mat. Sci Tech. Vol.7 (1991) pp. 330-333
3. M. Zhu et. al; Wear, 242 (2000) pp. 47-53
4. S. Mohan, V. Agarwala, S. Ray; Mater. Trans. JIM 33 (1992) p. 1057
5. L. Ratke, S. Diefenbach; Mat. Sci. Eng. Vol. R15 (1995)
6. S. K. Srivastava, S. Mohan, V. Agarwala, R. C. Agarwala; Metall. Mater. Trans. A 25A (1994) pp. 851-856
7. J. Z. Zhao, S. Drees, L. Ratke; Mat. Sci. Eng. A 282 (2000) pp. 263-265
8. T. Berrenger, P. R. Sahm; Z. Metallk. 87 (1996) 3 p. 188
9. R. Gray, S. Mohan, V. Agarwala, R.C. Agarwala; Z. Metallk. 84 (1993) p. 723
10. S. Mohan, V. Agarwala, S. Ray; Mater. Trans. JIM 33 (1992) pp. 1055-1057
11. V. Agarwala et al.; Mat. Sci. Eng. A 327 (2002) pp.186-202

Szemelvények a hazai képfeldolgozás (h)őskorából

Szabó József¹,

PICTRON Kft

¹1114 Budapest, Bartók Béla út 3.

szaboj.pictron@axelero.hu

Kivonat A hazai képfeldolgozás története mintegy harmincéves múltra tekinthet vissza. A cikk az alkalmazásorientált képfeldolgozás egyik legjelentősebb kutatóhelyén, az Szki-ban az 1970 évek közepétől 1980 végéig elért eredményeket kísérli meg összefoglalni. Ebben az időszakban sikerült megteremteni a képfeldolgozáshoz szükséges hazai gyártmányú eszközbázist, ennek legfontosabb elemeként kifejleszteni az első képmegejelntésre alkalmas és sorozatban is gyártott színes raszteres megjelenítőt (CDP). Erre az eszközbázisra alapozva számos úttörő alkalmazói rendszer kifejlesztésére is sor került. Ezek között feltétlenül említést érdemel az űr- és légifelvétel kiértékelésére kialakított első hazai számítógépes rendszer, a kutatóhajókra telepített tengerfenék kutatási célú képelemző, a laboratóriumi/mikroszkópi képkiértékelő és még számos további alkalmazás. A visszatekintés célja a lassan feledésbe merülő korai eredmények összegyűjtése, azok megőrzése érdekében, és a hihetetlen mértékű technikai fejlődés tudatosítása. A hivatkozásoknál megadott honlapon további információk, képek találhatóak a témáról ([11]).

1. Bevezetés

Már a hetvenes évek elején voltak lelkes emberek, alapvetően egyetemeken és kutató intézetekben, akik érdeklődéssel fordultak a képfeldolgozás felé, de megfelelő eszközök hiányában ez főleg elméleti kutatásokat jelentett. Az 1970-es évek végére a SZTAKI-ban, és a KFKI-ban, a Szegedi Egyetemen (JATE) valamint a BME egyes tanszékein alakultak kutató csoportok, és Szki-ban létrejött egy kifejezetten alkalmazásorientált képfeldolgozással foglalkozó műhely. Amint azt az előadás címében is jeleztem, nem kívánok a hazai képfeldolgozás (h)őskoráról, vagy kevésbé hangzatosan a kezdeti szakaszáról teljes körű áttekintést adni, kizárólag annak egy szeletét szeretném bemutatni, nevezetesen az Szki-ban az 1970-1990 között elért eredményeket.

2. Technikai és gazdaságpolitikai háttér

A hazai képfeldolgozási fejlesztések felgyorsulásához a következő tényezők járultak hozzá.

Számítástechnika robbanásszerű fejlesztése: A KFKI-ban az 1970 évek közepén elkészültek a különböző PDP gépek hasonmásai, és ekkor indult meg és teljesedett ki a KGST országok Egységes Számítógép Rendszerének (ESzR) fejlesztési programja. Ez utóbbi program hazai koordinációjára és a gépcsatlád legkisebb tagjának, az R-10 számítógép kifejlesztésre alakították meg 1969-ben a Számítástechnikai Koordinációs Intézetet, a később Számítástechnikai Kutató és Innovációs Központnak átnevezett Szki-t. A KFKI TPA 1140 gépe, és az R-10 gép már alkalmasak voltak képfeldolgozási feladatok megoldására is.

Embargó: A csúcstechnológiai eszközök keleti blokk országaiba irányuló exportjára vonatkozó korlátozások többek között a képfeldolgozási fejlesztéseknek is lendületet adtak, miután ezek iránt az eszközök iránt hallatlan kereslet mutatkozott.

Úrkutatási programokba való intenzív bekapcsolódás: Az úrkutatási együttműködésbe az akadémiai és egyetemi kutatóhelyek közvetlenül bekapcsolódtak. A 70-es évek végén, a 80-as évek elején beindultak az űr- és légifelvételek hasznosítását elősegítő központi programok is, amelyek lehetőséget adtak az Szki-nak is a képfeldolgozási fejlesztésekbe való intenzív bekapcsolódásra.

3. Az Szki Matematikai Laboratórium célkitűzése, profilja

Az Szki Matematikai Laboratóriumát (MAL) eredetileg abból a megfontolásból hozták létre, hogy az R-10 számítógép fejlesztése során felmerülő matematikai problémákra megoldást találjon. Rövidesen kiderült, hogy az alapvetően másolás, adaptálás jellegű fejlesztésnél nem igen merülnek fel matematikai problémák, vagy ha véletlenül mégis ilyen feladat adódott, azt az adott részlegnél lévő matematikus kollegák örömmel megoldották. E felismerést követően az Intézet vezetése az R-10 számítógépre alkalmazási programok fejlesztését jelölte meg a MAL feladatának.

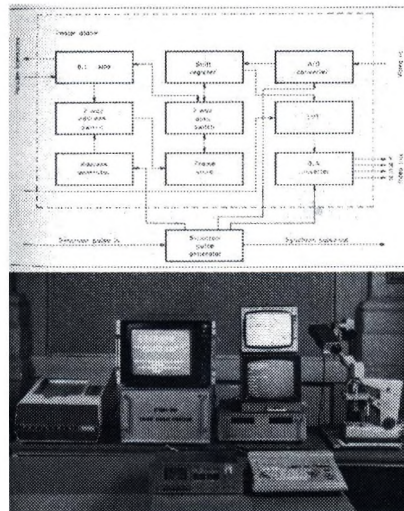
A laboratóriumvezető Dr. Dénes József érdeme, hogy a kötelező feladatok mellett, azonnal kép- és hangfeldolgozással kapcsolatos fejlesztéseket kezdeményezett. Ma közhelynek és természetesnek hat, de a lyukszalag/lyukkártya bemenettel rendelkező számítógépek korában többnyire értetlenséggel fogadták, amikor arról beszélt, hogy a számítástechnika akkor fog igazán elterjedni (válík "közüzemivé", ahogy egy későbbi cikkben megfogalmazta [1]), amikor a kép és hang be- és kimeneti perifériák könnyítik meg az ember-gép kapcsolatot. A hetvenes évek második feléig, tudatosan vállalt feladat megosztással, a fokozatosan 25-30 főre növekvő főosztály mintegy kétharmad része adatfeldolgozó programok készítésével (raktárgazdálkodás, információs rendszerek stb.) teljesítette a kirótt árbevételi kvótákat, egy harmad része pedig a kép és hangfeldolgozással kapcsolatos fejlesztéseket végezte. A megalapozó munkáknak köszönhetően a MAL olyan helyzeti előnybe került, hogy 1970 évek végén - képfeldolgozási rendszerek iránti igény ugrásszerű megnövekedésekor - teljes kapacitását a kép-

feldolgozási projektekre fordíthatta. A cikk címében szereplő szójátékot kibontva az "őskor" 1979-ig tartott, a "hőskor" 1980-ban kezdődött.

4. Eszközbázis

Az embargó korlátok miatt alapvető fontosságú volt hazai vagy KGST alkatrészekben alapuló, így a korlátozó intézkedéstől függetlenül reprodukálható eszköz kifejlesztése képbemenet és raszteres képmegjelenítés céljaira. Az 1976-ban kifejlesztett első, egyetlen példányban elkészült színes raszteres megjelenítő 16 színes 144x146 felbontású képet tudott megjeleníteni, de videojel digitalizálására még nem volt alkalmas. A sor-puffer 2x4 darab gyors memória integrált-áramkört (74S200, 256x1) tartalmazott, és felváltva szolgált a beírásra, és a megjelenítésre. A teljes kép a Szki mikrogép memóriájában tárolódott, ahonnan, 16 bites szavanként, 4 képpont adataival frissült a sor-puffer. A berendezés színes TV készüléken tudta megjeleníteni a memóriában tárolt képet ([2]).

Ezt követte a már valóban használható CDP (Color Display Processor). Először CCD shiftregisztert tartalmazó 16 Kbites dinamikus tároló áramkör (Intel 2416), majd 1978-tól a 16 Kbites DRAM megjelenésével (Intel4116), ez utóbbi képezte a CDP végző változata a videojelből 64 szűrkeárnyalatú, 288x384 felbontású képet hozott létre, amit a saját memóriájában tárolt. Ma már hihetetlennek tűnik, de a raszteres képmegjelenítéshez használt eszközök feladata a gyors TV technikai eszköz és a lassú, kis memóriával rendelkező számítógép közötti kapcsolat megteremtése volt. A CDP logikai felépítése és egy tipikus konfiguráció képe látható az 1. ábrán. A berendezés színes RGB (SECAM rendszerű) monitoron jelenítette meg a képet, alkalmas volt álszínes megjelenítésre, 2-2 biten ábrázolva a 3 alapszínt. A külső szakemberek bevonásával kifejlesztett berendezésben alkalmazott egyedi megoldásokat magyar és nemzetközi szabadalmi bejelentések védték ([3]). A fejlesztés Endrődi Béla, Manno Sándor és Szenes Zsuzsa nevéhez fűződik. A CDP különböző számítógépekhez való illesztésében, és egyéb hardver feladatokban Zsótér Jenő működött közre. Érdemes megemlíteni, hogy a maga korában igen értékes és korszerű berendezésről a szabadalmi bejelentésen kívül érdemi publikáció csak jóval később, a CDP-hez szoftvert fejlesztőktől jelent meg ([4]). Ez azzal magyarázható, hogy az Intézet a képfel-



1. ábra.
A CDP elvi rajza és egy konfigurációja

dolgozás témát nem szívesen hozta nyilvánosságra, miután attól félt, hogy a "fő fejlesztési csapáshoz", a számítógép fejlesztéshez szükséges beszerzési és információs forrásokat veszélyezteti, ha közismertté válik a képfeldolgozással kapcsolatos tevékenység.

A Híradástechnika Szövetkezet (HTSZ) vállalta CDP sorozatgyártását, és több mint száz példányt készült 1989-ig. A CDP-t a R-10 és TPA 1140 számítógéphez, valamint a HTSZ Motorola 6800 alapú mikrogéphez, később az Szki mikrogéphez illesztettük.

Már Szki fejlesztésként 1988-ra készült el az Imaging Technologies PC-Vision kártyájának utánérzéseként az IBM kompatibilis PC-be helyezhető 512x512x8 bit felbontású Provision digitalizáló és megjelenítő kártya. Három kártya egyidejű alkalmazásával 24 bites valódi színes kép előállítása és megjelenítése is megoldható volt. Az embargó korlátok megszűnésével a gyártás gazdaságtalanná vált és 25-30 darab után meg is szűnt.

Az alkalmazható számítógép kezdetben kizárólag a francia licenz alapján az Szki-ban adaptált, később a VIDEOTON által gyártott legkisebb ESZR számítógép, az R-10 gép volt. A később, az űr- és légifelvétel kiértékelésére fejlesztett MIP rendszer TPA 1140 számítógépre készült. Miután a KGST országokban gyorsan népszerűvé váló CM-4 gép szintén a PDP 1040 gép adaptációja volt, így a TPA gépre fejlesztett programok változtatás nélkül futtathatók voltak, ami megkönnyítette a Szovjetunióba történő értékesítést.

5. Alkalmazások

Reménytelen és értelmetlen vállalkozás lenne az összes alkalmazási projektet felsorolni, ezért csak a nagyobb, és néhány - valamilyen szempontból különleges - témát ismertetek. Egy 1987-ben készült összefoglaló cikkben néhány egyéb alkalmazás rövid összefoglalója megtalálható, a megfelelő hivatkozásokkal együtt ([5]),.

5.1. Kezdeti lépések

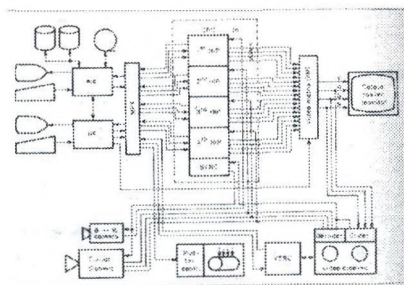
Ma már elképzelhetetlen feltételek között számos rendkívül érdekes kísérleti rendszert fejlesztett ki Kovács Emőke (aki később a RECOGNITA vezető fejlesztőjeként vált ismertté), Csornai Gábor és más fiatal kollegák. Ezek közül feltétlenül említést érdemel a kamerás szállítópálya vezérlő rendszer ([6]), amely egy demonstrációs célból épített görgősoros szállítópályán kifogástalanul vezérelte a tároló pályáról a üdítődobozos ládákat, a rájuk helyezett kódok felismerése alapján a célállomásra. Külön öröm számomra, hogy 25 évvel később, éles alkalmazásban PICTRON Kft színekben sikeresen oldottunk meg egy hasonló feladatot.

A katonai járműbe épített R-10 számítógépre (MOBIL R-10) telepített kamerás rendszerrel végzett első rendszám felismerési, sebességmérési és forgalomszámlálási kísérletek megfelelő technikai eszközök hiányában ugyan még nem hoztak megbízható eredményeket, de jó alapot és értékes tapasztalatokat adtak a későbbi munkákhoz.

5.2. Mintarendszer műhold- és légifelvételek kiértékelésére

A légi- és űrfelvételek hasznosításának, és az ehhez szükséges fejlesztések kerekeit a 3514/1979. MT határozat jelölte ki. Ebben a FÖMI-t jelölték ki elsődleges feldolgozó központnak. Tekintettel arra, hogy az Szki-MAL az OMSZ, Vízügyi szervezetek, HM megbízásából már kidolgozott R-10 számítógépre egy képfeldolgozó alaprendszert, ezért Szki-t jelölték ki a szolgáltató központok részére alkalmas számítógépes képfeldolgozó mintarendszer kifejlesztésére.

Az exportképesség és reprodukálhatóság (berendezés szinten KGST, lehetőleg magyar eszközök alkalmazása), és az országos szolgáltató központ kialakításának igénye egymásnak ellentmondó elvárásokat jelentett, ami előre kódolta a fejlesztő és fogadó intézmény konfliktusait. A FÖMI szerette volna az embargó korlátok alatt még átcsúszó legkorszerűbb eszközöket beszerezni, a remélt szovjet export miatt viszont a felhasználható eszközök elmaradtak a már nyugatról beszerezhető berendezések színvonalától. A nehézségek ellenére 1983 elején 256 Kbyte memóriával, 20Mbyte (ebből 10Mbyte cserélhető) merevlemezzel, 4 db mágnesszalag (800bpi-s) egységgel, 5 db alfanumerikus megjelenítővel, sor- és mátrixnyomtatóval felszerelt TPA 1140 számítógépen telepítettük a MIP (Modular Image Processing) többsávú képek érékelésére alkalmas rendszert. A képek megjelenítésére a CDP 4 képsíkos változata szolgált, amellyel színes képdigitalizálást és valódi színes megjelenítést is meg lehetett valósítani. A négy képsíkot az akkor használatos LANDSAT 4 műhold négy képsíkjának tárolása indokolta. Egy mikrogép vezérelte a képsíkok kiválasztását és megjelenítést (2. ábra).



2. ábra.

4 képsíkos CDP a MIP rendszerben

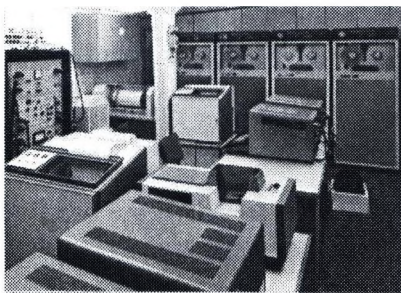
Színes képanyomtatáshoz az ELGI Corollpress 4 forgódobos nyomtatóját alkalmaztuk, amely 40x40 cm méretű színes papírképet tudott előállítani.

A rendszerfejlesztést végző osztály vezetője voltam, de a szoftver fejlesztés közvetlen szakmai irányítását Föglein János, Dr. Hegedűs Csaba és Kelemen Dezső, közel azonos időben végzett fiatal triumvirátus végezte. Az elkészült mintegy 180 000 (jelentős részben assembler) forrássorból, 52 programból álló rendszer adatkonverziós, képjavítási, geometriai korrekciós, ortogonális transzformációs, statisztikai, multispektrális képosztályozási, megjelenítési és szerviz funkciókat tartalmazott. A maximális képméret 1024x1024 felbontású volt, ami a konfiguráció paraméterei alapján érthető.

Az eredeti célkitűzésnek megfelelően néhány rendszert belföldön és a Szovjetunióban sikerült is értékesíteni. A fejlesztés alatt és az átadást követően a MAL egy ideig különböző felhasználók részére űr- és légifelvételek számítógépes kiértékelési feladatokat végzett.

5.3. Tengerfenék kutató rendszer

A téma jellege miatt gyakorlatilag semmiféle nyilvánosságot nem kapott az egyébként szakmailag rendkívül érdekes, és gazdaságilag is eredményes VIKING rendszer. A Dr. Álló Géza, Feró László és Staszny Gábor szakmai irányításával - mellesleg a Videoton fővállalkozásában - folyó fejlesztés célja a szovjet tengeri kutató flotta geológiai rendszerének kiegészítése volt a tenger alatti felvételek kiértékelésére szolgáló képfeldolgozó rendszerrel. Jellemző módon a Videoton házi lapjában megjelent egyetlen írás átkeresztelt néven ad hírt a "Videoton" képfeldolgozó rendszerről, és a javasolt felhasználási területek között meg sem említi a tényleges alkalmazást ([5]).



3. ábra.

R-11-es VIKING konfiguráció

ható a hajóra telepített számítógép, ami már 1 MB memóriával rendelkezett, és a szokásos periféria készlet egy 50MB-os mágneslemez-es egységgel is meglát erősítve. Az ábrán nem látható a konfiguráció érdekessége, a feladathoz jól alkalmazkodó két képsíkos CDP rendszer és az ELGI jelfeldolgozó processzor. Az eredetileg geofizikai regisztrátumok kiértékelésére fejlesztett jelprocesszor végzett kisebb módosításokkal néhány képfeldolgozási funkció futási idejét egy nagyságrenddel fel lehetett gyorsítani.

5.4. PC alapú rendszerek

Az 1980-as évek közepén a PC-k árának fokozatos csökkenése, teljesítményük egyre gyorsuló, napjainkig tartó növekedése és a PC-be helyezhető digitalizáló kártyák megjelenése egy nagyságrenddel lecsökkentette a képfeldolgozó konfigurációk árát, lezárult egy korszak. Lényegében ez már a képfeldolgozás következő fejezete, ezért csak utalok arra, hogy a MIP projekt lezárulásával felszabaduló csapat rövid idő alatt kifejlesztette PRIMA elnevezésű programot, amely 1989-ben a Kiváló Áruk Fóruma címet is megkapta. A program valószínűleg az egyetlen általános célúnak tervezett képelemző program, amely késztermékként mintegy 30-40 példányban kelt el. A 4. ábrán egy Provision kártyát tartalmazó PC konfiguráció látható a PRIMA programmal.

Az 1980-as években nagy reményeket fűztek a tengerfenéken helyenként fellelhető nagy tisztaságú nemesfém kitermeléshez. A rendszer célja, a tengerfenék felett a kutatóhajó által vonszolt kameráról érkezett képeken az ún. fém "konkréciók" megszámlálása volt. A hajóra telepített navigációs rendszer segítségével rögzíteni lehetett a felvétel készítési helyét. Az 1990-es évek elejéig több R-11 számítógépre alapozott rendszer telepítése megtörtént, de a szovjet piac és a Videoton összeomlása a további értékesítést lehetetlenné tette. A 3. ábrán lát-



4. ábra. PRIMA rendszer PC konfiguráción

A MAL-ban ebben az időszakban készültek a rendőrségi fantomkép összeállító és szakértői képelemzői rendszerek első változatai, és az első képarchiváló programok.

Ekkor már a MAL-tól szervezetileg külön folyt a karakterfelismerő program fejlesztése, ami a használható szkennerek megjelenését követően a szűk szakmán túl is ismert módon kiemelkedő sikereket ért el. Ez a korszak egy külön előadást érdemel, és jelen cikk kereteibe már nem fér be.

6. Oktatás

Az 1980-as évek közepén a képfeldolgozással kapcsolatba kerülő szakemberek döntő többsége a képfeldolgozással, de még a számítástechnikával sem találkozott egyetemi éve alatt. A BME Mérnöki Továbbképző Intézete felismerve ezt a hiányt 1984-ben indította be a "Bevezetés a számítógépes képfeldolgozásba" című 2 féléves tanfolyamait, amelyet éveken át a MAL munkatársai tartottak. A tanfolyam anyagából egyetemi jegyzet, majd később könyv is készült ([8], [9]). A jegyzet 1985-ben elnyerte BME Kiváló Egyetemi jegyzet kitüntető címet.

Jó tíz évvel később ezek a könyvek képezték - természetesen jelentősen átalakítva, gyakorló programmal és CD melléklettel kiegészítve - a Gábor Dénes Főiskolán rendszeresített tankönyv alapját, amikor az oktatásba bevezették a számítógépes képfeldolgozást. Az 1996-os első kiadást követően többször aktualizált utolsó változata 2002-ben készült ([10]). Az összes eladott példányszám 10 000 felett van.

Hivatkozások

1. Dénes, J.: Megvalósult újszerű ember gép kapcsolatok az Szki-ban Információ Elektronika. 1983.5. 257-262.
2. Szabó, J.: Mikrogépes színes display. Mérés és Automatika. 1977.5. 204-206.
3. Endrődi, B. Szenes, Zs.: Grafikus megjelenítő rendszer. Magyar szabadalmi sz.: 177.376, 1110/80. Európai sz.sz.: 51655. Osztrák sz.sz.: E15837
4. Álló, G., Feró, L., Staszny, G.: A Colour picture converter for image processing. Microprocessing and Microprogramming. 1987.1.27-35.

5. Szabó, J.: Képfeldolgozó rendszerek és alkalmazásaik. Magyar Elektronika. 1987.2.26-30.
6. Zsótér, J.: Darabárú szállító és mozgató berendezés vezérlése videojelek segítségével. Anyagmozgatás, csomagolás. 1975.8.234-236.
7. Tamás, K.: A Videoton színes képfeldolgozó rendszere. Videoton software tájékoztató. 1983.1-2.28-32.
8. Álló, G. - Föglein, J. - Hegedűs, Gy. Cs. - Szabó, J. (1985, 1989, 1993): Bevezetés a számítógépes képfeldolgozásba, Egyetemi jegyzet, Budapesti Műszaki Egyetem Mérnöki Továbbképző Intézet, Budapest ISBN 963 431 633 6
9. Álló, G. - Hegedűs, Gy. Cs. - Kelemen, D - Szabó, J. (1989): A digitális képfeldolgozás alapproblémái, Akadémiai Kiadó, Budapest, ISBN 963-05 4897 6.
10. Berke, J. - Hegedűs, Gy. Cs. - Kelemen, D. - Szabó, J. (2002): Digitális képfeldolgozás és alkalmazásai. Keszthelyi Akadémia Alapítvány - Pictron Kft., Keszthely, Budapest, ISBN 963 206 260 4.
11. www.pictron.hu/hoskor

Képfeldolgozás alkalmazása gömbgrafitos öntöttvas szövetszerkezetének vizsgálatára

Szalai Ibolya¹, Gácsi Zoltán²,

DUNAFERR Rt. Innovációs Menedzsment

1

Miskolci Egyetem, Fémtani Tanszék, 3515 Miskolc-Egyetemváros

2

Kivonat Az öntöttvas szövetszerkezetének hagyományos vizsgálata során a próbatestek mikroszkópos képét összehasonlítják egy szabványos mintaképsorozattal és megadják a grafit és a fémes alapanyag jellemző paramétereit. A vizsgálat objektivitásának ellenőrzésére különböző mechanikai tulajdonságokkal rendelkező gömbgrafitos öntöttvas mintákon és szabványos etalonfelvételeken számítógépes képelemzővel méréseket végeztünk, és az eredmények alapján kialakítottunk egy értékelési rendszert.

1. Bevezetés

Az ipari metallográfiai laboratóriumokban gyakran kerül sor különböző minőségű öntöttvasak szövetszerkezetének összehasonlító vizsgálatára. Az eljárás során a próbatestek szövetszerkezetét szabványban (MSZ ISO 945:1993, MSZ 5716:1974) közölt etalonkép sorozattal hasonlítják össze és megadják a jellemző paramétereket: a grafit- alakot, - méretet, - eloszlást és a fémes alapanyag összetevőit. A módszer pontosságának ellenőrzésére számítógépes képelemzés módszerével méréseket végeztünk gömbgrafitos öntöttvas próbatesteken és a szabvány etalonképein. Majd definiáltuk azokat a morfológiai paramétereket, melyek alkalmasak az alak leírására. A paraméterek alapján kialakítottunk egy olyan algoritmust, mely a számítógépes képelemzős eredményeket beolvassa, elvégzi a szükséges átalakításokat, végül megadja a próbatesten mért grafit szemcsék alakját külön-külön, és az egész próbatestre tekintve százalékosan is. A fémes alapanyag összetevőinek előhívására és azok mennyiségi meghatározására különböző színes maratási technikákat alkalmaztunk. A maratás hatására a különböző fázisok eltérő színben jelentek meg így lehetővé vált azok képelemzővel történő meghatározása.

2. Vizsgálati módszerek

2.1. Mikroszkópos vizsgálat

Az MSZ ISO 945:1993 és MSZ 5716:1974 szabványok a grafit és a fémes alapanyag meghatározását tartalmazzák. A grafit vizsgálata a grafit alakjának, méretének és az eloszlásának a megállapítását jelenti, a fémes alapanyag vizsgálata

során a ferrit mennyiségét, a perlit finomságát a foszfidos eutektikum szerkezetét és eloszlását, az eutektikus cellák méretét, valamint a cementit mennyiségének meghatározását végzik el. A vizsgálandó próbatestet előkészítik majd a maratlan csiszolaton az etalonképpel való összehasonlítás alapján meghatározzák a grafit alakját méretét és eloszlását. Ezt követően a legtöbbször "Nitallal" (2%-os salétromsavas alkohol) maratott próbatesten szintén az etalonkép segítségével jellemzik a fémes alapanyag mennyiségét [1].

2.2. Számítógépes módszer: képfeldolgozás és képelemzés

A gömbgrafitos öntöttvas szövetképén a képfeldolgozás segítségével olyan átalakítást végeztünk el, amelyekkel a próba előkészítés során jelentkező karcokat és a maratott szöveten fellépő, főleg a maratásból adódó foltokat távolítottuk el. Ugyanakkor a képelemzés alkalmával a grafit szemcsék alakját tudtuk számszerű adatokkal jellemezni.

3. Mérési eredmények

3.1. Számítógépes mérések eredményei

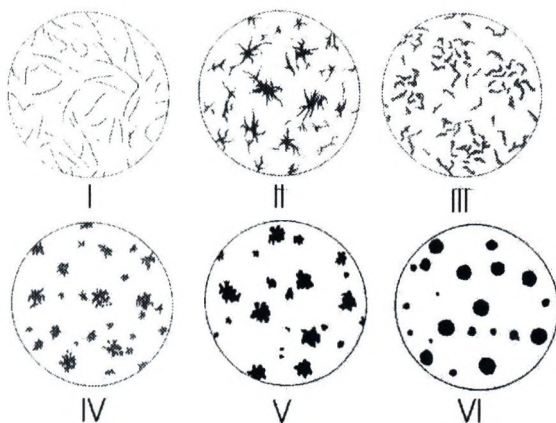
A vizsgálataink során egy LEICA MEF4A, REICHERT MD 4000E típusú fénymikroszkópot és a QUANTIMET 550MW típusú képelemzőt használtunk.

A grafit alakra vonatkozó etalon felvételeken elvégzett mérések. Ahhoz, hogy a próbatesteken mért grafit szemcséket be tudjuk sorolni egy-egy alak kategóriába az MSZ ISO 945:1993-as szabványban található gömbgrafit alakjának meghatározására alkalmazott etalonképeken is méréseket kellett elvégeznünk. Az 1. ábrán a szabványban található etalonképek láthatók [1].

Mértük a körszerűség értékeit (ez a paraméter a kör alakjától való eltérést mutatja, $kör=1$), majd meghatároztuk kitöltöttség értékeit (konvex terület és a terület hányadosa) és végül mértük a nyújtottság értékeit (hosszúság és a szélesség viszonya) [2]. Ezen paraméterek bizonyos értéktartományait rendeltük az egyes grafitalak típusokhoz, majd kialakítottunk egy kategorizálási rendszert. Ezeket a kategóriákat felhasználva készítettünk egy számítógépes osztályozási rendszert. Ez egy excel-makroprogram, amely a képelemző méréseket tartalmazó fájlokat beolvassa, elvégzi a szükséges átalakításokat, és végül megadja a próbatesten mért grafit szemcsék alakját külön-külön, és az egész próbatestre tekintve százalékosan is. A 2. ábrán a kategorizálási rendszerben szereplő paraméterek értékei láthatók az alak típusokra lebontva. A makroprogram lefuttatása után ebben a "panelben" jelennek meg a mérési eredmények.

Fémes alapanyag vizsgálata. A szabvány szerinti mikroszkópos vizsgálat tartalmazza (MSZ 5716:1974) a fémes alapanyag jellemzését is.

A legtöbb próbánál a ferrit mennyiségét kellett meghatározni. A szövet előhívására "Nitálos" (2%-os salétromsav) maratást alkalmaztunk, mivel a próbák



1. ábra. Az MSZ ISO 145:1993 szabvány alakra vonatkozó etalonfelvételei

Mérési eredmények				
Próbák száma	Kontraszt	Levegőtartalom	Átlagos érték	Standard deviancia
1	0,5	0,1	0	0,00
2	0,5	0,1	0	0,00
3	0,5	0,1	0	0,00
4	0,5	0,1	0	0,00
5	0,5	0,1	0	0,00
6	0,5	0,1	0	0,00
7	0,5	0,1	0	0,00
8	0,5	0,1	0	0,00
9	0,5	0,1	0	0,00
10	0,5	0,1	0	0,00
11	0,5	0,1	0	0,00
12	0,5	0,1	0	0,00
13	0,5	0,1	0	0,00
14	0,5	0,1	0	0,00
15	0,5	0,1	0	0,00
16	0,5	0,1	0	0,00
17	0,5	0,1	0	0,00
18	0,5	0,1	0	0,00
19	0,5	0,1	0	0,00
20	0,5	0,1	0	0,00

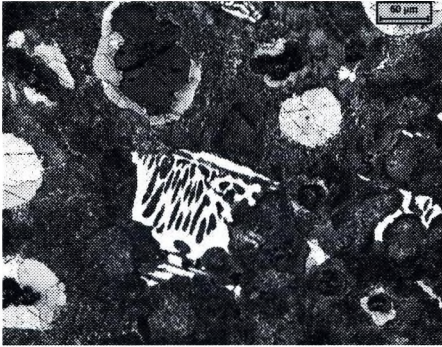
2. ábra. A kategorizálási rendszer értékei alaktípusokra lebontva

alpmátrixát alkotó két fázis (ferrit és perlit) ezzel a marószerral jól elkülöníthető és így a ferrit mennyisége képelemző segítségével jól mérhető. Azoknál a próbák-nál melyekben cementit is található a Nitálos maratással nem érhető el megfelelő kontraszt, hiszen mindkét fázis fehérnek látszik. Így ezekben az esetekben színes maratási eljárásokat alkalmaztunk.

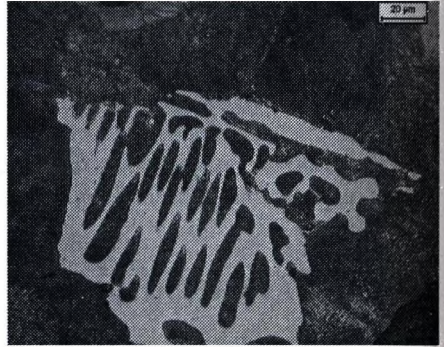
Színes maratás során a marószereket a megfelelő elszíneződés érdekében úgy állítják be kémiaiilag, hogy a minta felületén egy stabil általában 0,04–0,5 μm vastagságú filmet képezzen. Ezek a vékony oxid, szulfid és komplex molibdát filmek a próbában található fázisokon más-más vastagságúak lesznek és az interferencia hatásnak köszönhetően különbözőképpen színeződnek. Így a különböző fázisok jól elkülöníthetők és a képelemző által is jól mérhetővé válnak [3].

Többféle színes marószert kipróbáltunk, az első az úgy nevezett 1Beraha-reagens volt, melynek összetétele a következő: 1000 ml desztilláltvíz, 240 g Na-tioszulfát, 25 g Cd-klorid, 30 g citromsav. A marószer hatására ferrit fehérnek, a perlit kéknek a cementit pedig sárgás-barnának látszik. A második marószer 2Beraha-reagens, melynek összetétele: 1000 ml desztilláltvíz, 240 g Na-tioszulfát, 24 g Pb-klorid, 30 g citromsav. A maratás előtt pikralos (4%-os salétromsav) előmaratást végeztünk. A cementit nem maródott így az fehérnek látszik a ferrit és a perlit barna. A harmadik reagens összetétele a következő: 100ml desztillát

víz, 1 g Na-metabiszulfid, 1 ml HCl. A maratás hatására perlit és a ferrit barnának látszik a cementit pedig fehér. A 3-4. ábrán a Nitálos maratással előhívott szövet, 5-6. ábrán az 1 reagenssel a 7-8. ábrán a 2 reagenssel a 9-10. ábrán pedig a 3. reagenssel maratott szövet figyelhető meg.



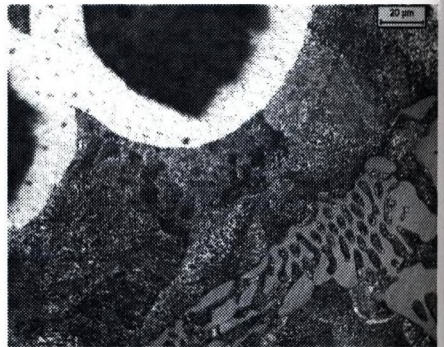
3. ábra
Nitallal maratott
szövetszerkezet (N=200x)



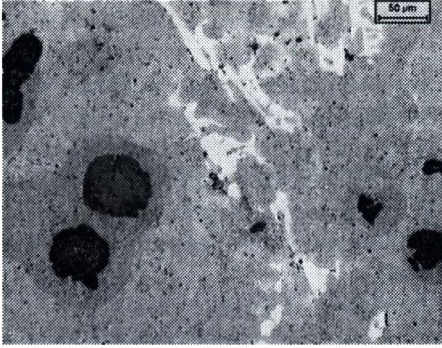
4. ábra
Nitallal maratott
szövetszerkezet (N=500x)



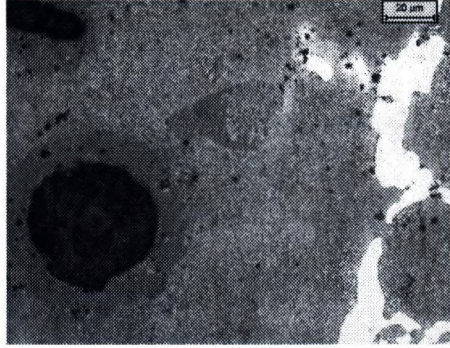
5. ábra
1Beraha-reagenssel maratott
szövetszerkezet (N=200x)



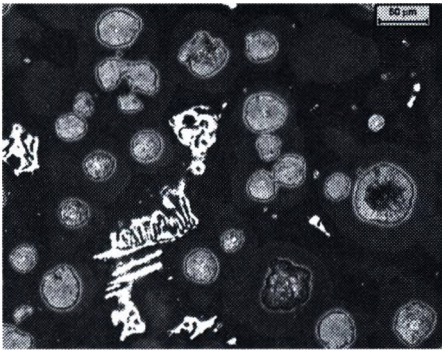
6. ábra
1Beraha-reagenssel maratott
szövetszerkezet (N=500x)



7. ábra
2Beraha-reagenssel maratott
szövetszerkezet (N=200x)



8. ábra
2Beraha-reagenssel maratott
szövetszerkezet (N=500x)



9. ábra
3-reagenssel maratott
szövetszerkezet (N=200x)



10. ábra
3-reagenssel maratott
szövetszerkezet (N=500x)

A fenti marószereket alkalmazva a cementit jól látható színnel különül el a ferritől (lásd 9-10 ábra). Így lehetőség van arra, hogy a színes digitális képen végezzük el a megfelelő szín (fázis) detektálását. Gyakran a detektálás nem volt megfelelő és így még némely képfeldolgozási művelettel (nyitás, zárás) pontosítottunk.

4. Összefoglalás

Az öntöttvas szövetszerkezetének jellemzésére szolgáló hagyományos etalonképes módszer megbízhatóságát 3 mérési sorozat eredményeinek összehasonlításával ellenőriztük. A grafit alak leírására szolgáló eredmények a legtöbb esetben nem voltak azonosak. A MSZ ISO 945:1993 szabvány etalonképein végzett számítógépes képelemzés alapján kialakítottunk egy osztályozási rendszert, amely

segítségével objektívebb eredményeket tudunk szolgáltatni a grafit alakra vonatkozóan.

A fémes alapanyag meghatározás során több színes maratási eljárást kipróbáltunk a szövetszerkezet előhívására. Mivel fázisok mennyiségének megállapítását szintén nagy mértékben befolyásolja a szubjektivitás, ezért ebben az esetben is képelemzős módszerrel ellenőriztük a hagyományos úton kapott eredményeket.

Az alakot és a fémes alapanyagot jellemző paraméterek jelentős eltérést mutattak az etalonképes és számítógépes meghatározásnál. A képelemzős módszerrel objektívebb eredményeket tudunk megadni.

Hivatkozások

1. MSZ ISO 945:1993 és MSZ 5716:1974 szabvány
2. Gácsi Zoltán, Sárközi Gábor, Réti Tamás, Kovács Jenő, Csepeli Zsolt, Mertinger Valéria: Sztereológia és képelemzés, Egyetemi Tankönyv, 2001. Well-Press-PHARE
3. Huszár Katalin: Színes maratás alkalmazása automatikus képelemzővel történő mérés esetén, Diplomaterv, Miskolc, 1996. május

Stochastic Radiosity on the Graphics Hardware

László Szirmay-Kalos and Gábor Szijártó¹

TU Budapest, Dept. of Control Engineering and Information Technology
szirmay@iit.bme.hu

Abstract. This paper presents a work in progress aiming at the solution of the diffuse radiosity problem, i.e. a Fredholm type integral equation, on the graphics hardware.

1 Introduction

We witness two kinds of advances of the graphics cards and their processing units (GPU). On the one hand, their speed is improving constantly, outperforming the Moore law. The average computation time of a pixel, including transformations local illumination, projection, clipping, texturing, blending and visibility determination, is about a few nanoseconds, i.e. close to the time of a single memory cycle. This incredible speed is the result of the massive pipelining, the parallelization and the special ALUs along the pipeline. For example, a GeForce 3 GPU may operate with 800 pipeline levels, while the Intel P4 processor has at most 20. The pipeline is broken to four parallel channels at difficult parts, and ALUs along the pipes can handle four floating point values in parallel and execute complex operations such as the multiplication of a 4×4 matrix and a 4-dimensional vector.

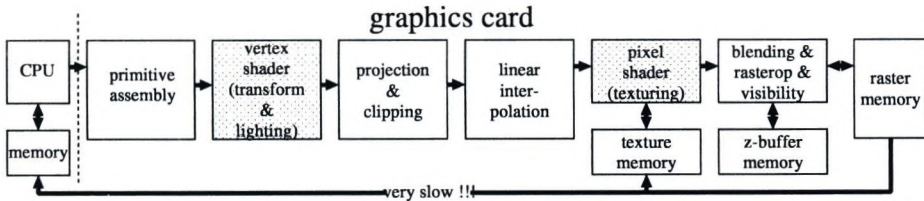


Fig. 1. GPU architecture including vertex and pixel shaders

On the other hand, the fixed pipeline has been developed further and has been turned to be partially programmable, thus the burnt in algorithms can be replaced by user specified ones. Two phases of the pipeline have become programmable, the vertex and pixel processing units (figure 1). Vertex processing, which was originally responsible for vertex transformation and local illumination computations at the vertices, can be controlled by a *vertex shader* program.

Pixel processing, which originally computed texturing, can be governed by a *pixel shader* program. The programmability together with the high speed have made many researchers think of how special, non-local illumination and even non-graphics algorithms can be ported from the CPU to the GPU. Examples include ray-shooting [8], Voronoi diagrams, FFT and the solution of linear equations, etc. (see <http://www.gpgpu.org>). In this paper we propose the solution of Fredholm type integral equations of the second kind by the GPU. The selected integral equation is the rendering equation describing the diffuse global illumination problem. Thus we use the graphics hardware, which was designed for local illumination rendering, to solve the global illumination problem.

When porting an algorithm onto the GPU we have to implement three programs, one for the CPU usually in C++, one for the vertex shader, and one for the pixel shader. Vertex and pixel shaders can be programmed in assembly, in Cg or in the high level shading language of DirectX. We used the Cg programming language [1]. The difficulty of the porting comes from the following restrictions. Vertex and pixel shaders form a stream processing architecture, where CPU feeds the vertex shader, which only modifies data items. Vertex shader results are passed to pixel shaders that can read textures and can only its target pixel stored either in the frame buffer or in the texture memory. Only pixel shaders can access memory (textures), but their program size, instructions and the number of texture accesses are limited. Reading back any data to the CPU destroys pipeline efficiency, thus should be avoided. Meeting all these limitations is the real challenge of GPU programming.

2 Solution of the rendering equation

Global illumination algorithms aim at the solution of the rendering equation. For the sake of simplicity, let us assume that the surfaces are diffuse. In this case, the rendering equation

$$L(\mathbf{x}) = L^e(\mathbf{x}) + (\mathcal{T}_{f_r} L)(\mathbf{x})$$

expresses the radiance $L(\mathbf{x})$ of point \mathbf{x} as a sum of the emission $L^e(\mathbf{x})$ and the reflection of all point radiances that are visible from here. The total reflection of the radiance of visible points is expressed by an integral operator

$$(\mathcal{T}_{f_r} L)(\mathbf{x}) = \int_S v(\mathbf{x}, \mathbf{y}) \cdot L(\mathbf{y}) \cdot f_r(\mathbf{x}) \cdot \frac{\cos \theta'_x \cdot \cos \theta_y}{|\mathbf{x} - \mathbf{y}|^2} d\mathbf{y}, \quad (1)$$

which is also called as the *light transport operator*. In this equation S is the set of surface points, $v(\mathbf{x}, \mathbf{y})$ is the mutual visibility indicator, which is 1 if points \mathbf{x} and \mathbf{y} are visible from each other and zero otherwise, f_r is the BRDF and θ'_x and θ_y are the angles between the surface normals and the direction between \mathbf{x} and \mathbf{y} .

The solution of the rendering equation requires general purpose instructions and is thus usually computed on the CPU. However, this is rather slow, and

the requirements of interactive rendering cannot be met. Our goal is to take advantage of the huge computation power of the GPU for the solution of the rendering equation. In order to do so, we transform the algorithm according to the capabilities of the GPU.

The CPU-based solution algorithms can be classified as random walk [6] and iteration techniques. The GPU support of random walk algorithms has been examined in [8]. Since iteration algorithms are conceptually closer to local illumination, which is originally supported by GPUs, we believe that iteration algorithms are better candidates for GPU implementation.

Iteration techniques are based on the fact that the solution of the rendering equation is the fixed point of the following iteration scheme:

$$L_m = L^e + \mathcal{T}_{f_r} L_{m-1}.$$

If this scheme is convergent, then the solution can be obtained as a limiting value:

$$L(\mathbf{x}) = \lim_{m \rightarrow \infty} L_m(\mathbf{x}).$$

Iteration simultaneously computes interaction between all surface elements, which is hard to implement in the GPU. The first attempt to port radiosity algorithm applied therefore Southwell iteration, which reduced the number of iterations [5]. This algorithm inherited the quadratic complexity of the original progressive radiosity and is quite complicated to implement.

We think that porting should also include the transformation of the original algorithm, to take into account the capabilities of the GPU. We propose that randomization offers a general strategy for such transformations. The formal basis of such approaches is the stochastic iteration, which was originally proposed for the solution of the linear equations, was presented in [7, 9, 4], then extended for the solution of integral equations [10, 11]. Stochastic iteration means that in the iteration scheme a random transport operator $\mathcal{T}_{f_r}^*$ is used instead of the light-transport operator \mathcal{T}_{f_r} . The random transport operator has to give back the light-transport operator in the expected case:

$$L_m = L^e + \mathcal{T}_{f_r}^* L_{m-1}, \quad E[\mathcal{T}_{f_r}^* L] = \mathcal{T}_{f_r} L.$$

Note that such an iteration scheme does not converge, but the iterated values will fluctuate around the real solution. To make the sequence converge, all previous iterated values are averaged:

$$\tilde{L}_m = \frac{1}{m} \cdot \sum_{i=1}^m L_i = \frac{1}{m} \cdot (L^e + \mathcal{T}_{f_r}^* \tilde{L}_{m-1}) + \left(1 - \frac{1}{m}\right) \cdot \tilde{L}_{m-1}. \quad (2)$$

Iteration works with the complete radiance function, thus its temporary version should be stored somehow. CPU algorithms usually use finite-element methods based on the decomposition of surfaces to triangular patches. Since in stream processing patches are processed independently, we store the radiance function in textures similar to the method of [3].

Having introduced the basic concepts we discuss the appropriate selection of the random iteration scheme.

3 Perspective ray-bundle shooting on the GPU

Perspective ray-bundle shooting chooses a point randomly and sends the radiance of this point from here towards all directions [2]. If point \mathbf{y} is selected with probability density $p(\mathbf{y})$, then the random transport operator is

$$(T_{f_r}^* L)(\mathbf{x}) = \frac{1}{p(\mathbf{y})} \cdot v(\mathbf{x}, \mathbf{y}) \cdot L(\mathbf{y}) \cdot f_r(\mathbf{x}) \cdot \frac{\cos \theta'_x \cdot \cos \theta_y}{|\mathbf{x} - \mathbf{y}|^2}. \quad (3)$$

It is easy to prove that this random operator really gives back the real transform operator in the expected case. In order to realize this random transport operator on the radiance function stored in a texture, two tasks need to be solved, including the random selection of a texel identifying point \mathbf{y} , and the update of the radiance at those texels which correspond to points \mathbf{x} visible from \mathbf{y} .

3.1 Random texel selection

According to importance sampling, it is worth setting the selection probability proportional to the integrand. Unfortunately, this is just approximately possible, and the point selection probability is set proportional to the radiance of the selected texel. If the light is transferred on several wavelengths simultaneously, the luminance of the radiated power should be used. Thus the selection probability of point \mathbf{y} is:

$$p(\mathbf{y}) = \frac{\mathcal{L}(L_j)}{\Phi}, \quad \Phi = \sum \mathcal{L}(L_i) \Delta S_i,$$

where texel j corresponds to surface point \mathbf{y} , \mathcal{L} is the luminance of a spectrum represented by red, green and blue components, Φ is the luminance of the integrated radiance, and ΔS_i is the area corresponding texel i . Note that if uniform parameterization is used then ΔS_i is similar to all texels.

The random selection can be supported by the *mipmapping hardware*. When the texture corresponding to the current radiance function is updated, the GPU is asked to compute the complete mipmap hierarchy. The mipmap can be imagined as a quadtree, which allows the selection of a texel in $\log_2 R$ steps, where R is the resolution of the texture. The top level of this hierarchy contains the average of all texels. The luminance of this value is multiplied by a random number uniformly distributed in the unit interval and also by four since on the next level this texel is decomposed to four texels. Then the next mipmap level is retrieved and the four texels corresponding to the upper level texel is obtained. The luminance of the four pixels are summed, the running sum is compared to the value obtained on the higher level. When the running sum gets larger, the summing is stopped. A new selection value is obtained as the difference of the previous value and the luminance of all texels before the found texel. Then the same procedure is repeated recursively on the lower mipmap levels. This procedure terminates at a leaf texel with a probability that is proportional to its luminance.

3.2 Update of the radiance texture

The points visible from \mathbf{y} can be found by placing a hemisphere around \mathbf{y} , and then using the z-buffer algorithm to identify the visible patches. Note that this step requires specialized vertex processing since the local illumination pipeline uses a rectangular and not a hemisphere window. Since it turns out just at the end, i.e. having processed all patches by the z-buffer algorithm, which points are really visible, the application of the random transfer operator requires two passes. In the first pass the center and the base of the hemisphere are set to \mathbf{y} and to the surface at \mathbf{y} , respectively, then the scene is rendered assuming that the color of patch i is exactly i . Having computed the image, the result is stored in a *visibility texture*. To demonstrate how the hemispherical projection can be done with the GPU, the vertex program is shown in the followings, where *modelview* is the eye transform of the camera put at \mathbf{y} , *IN.position* is the current vertex, and *OUT.hposition* contains the normalized pixel coordinates and the depth value:

```
float3 x          = mul(modelview, IN.position).xyz;
OUT.hposition.xy = normalize(x).xy;
OUT.hposition.z  = -(2*x.z + bp + fp)/(bp - fp); // [-fp,-bp]->[-1,1]
OUT.hposition.w  = 1;
OUT.color        = IN.color; // color now stores the patch index
```

In the second pass the same color values are set to identify the patches, but the BRDF and emission values are also passed, and now we render into the rectangle of the radiance texture. The vertex shader is set according to this transformation and also prepares the value of the randomly transported radiance in variable according to equation 3:

$$L_{ref} = L_{shoot} \cdot \frac{\cos \theta'_x \cdot \cos \theta_y}{|\mathbf{x} - \mathbf{y}|^2}, \quad L_{shoot} = \frac{L(\mathbf{y})}{p(\mathbf{y})} = \frac{L(\mathbf{y})}{\mathcal{L}(L(\mathbf{y}))} \cdot \Phi.$$

The CPU determines L_{shoot} and passes it to the vertex shader, which computes L_{ref} in the eye space where \mathbf{y} is in the origo:

```
float3 x = mul(modelview, IN.position).xyz; // y = origo
float3 ytox = normalize(x); // direction from y to x
float xydist2 = dot(x, x); // |x - y|^2
float costhetay = -ytox.z; // normal at y = axis -z
float3 xnorm = mul(modelviewIT, IN.normal); // normal at x = IN.normal
float costhetax = dot(xnorm, -ytox);
OUT.Lref = Lshoot * costhetax * costhetay / xydist2;
```

Note that the normal is transformed with the inverse transpose of the camera transform, because this way we can ignore any translation.

When a pixel is shaded, it is checked whether the same patch is stored in the visibility texture at the corresponding texel, that is, whether or not this pixel will be kept by the z-buffer [5]. The pixel shader code responsible for this check, for the multiplication with the visibility indicator and the BRDF (equation 3), for the update of the radiance texture, and also for the averaging operation (equation 2), is the following:

```

float3 visible_patch = tex2D(visibility_map, viscoord).rgb;
float visible = (abs(visible_patch.r - patch_index.r) +
                abs(visible_patch.g - patch_index.g) +
                abs(visible_patch.b - patch_index.b) < 0.001);
return tex2D(radiancemap, texcoord).rgb * (1-1/m) +
       (emission + visible * Lref * BRDF)/m;

```

The program gets the visibility texture coordinates (*viscoord*) and the patch index, *emission*, *BRDF* and the texture coordinate (*texcoord*) of the target point, as well as the prepared transported radiance *Lref* from the vertex shader. Iteration number *m* is passed directly from the CPU as a uniform parameter.

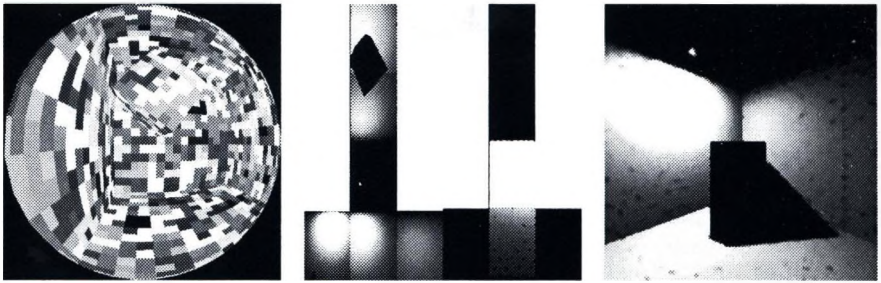


Fig. 2. The visibility map, the radiance map and the rendered scene after one stochastic iteration

4 Implementation results and conclusions

The proposed method has been implemented on an ATE Radeon 9700 Pro graphics card in C++/OpenGL/Cg environment. In the current implementation the random selection is made by the CPU and all other operations by the GPU. The implementation has been tested with the Cornell box scene and we concluded that a single iteration requires 10–20 msec for ten to ten thousand vertices and for 512×512 resolution radiance maps. This is about 5–10 times faster than the optimized CPU implementation using only the classical graphics pipeline for visibility checking. By porting even the random selection to the pixel shader we can eliminate all frame buffer read-backs, and thus expect further speedups. The drawback of this algorithm comes from the texture based radiance representation, which can result in dot artifacts. This problem will be attacked by interpolation similarly to [3].

References

1. http://developer.nvidia.com/page/cg_main.html.

2. Gy. Antal, L. Szirmay-Kalos, F. Csonka, and Cs. Kelemen. Multiple strategy stochastic iteration for architectural walkthroughs. *Computers & Graphics*, 27:285–292, 2003.
3. R. Bastos, M. Goslin, and H. Zhang. Efficient radiosity rendering using textures and bicubic reconstruction. In *ACM-SIGGRAPH Symposium on Interactive 3D Graphics*, 1997.
4. Ph. Bekaert. *Hierarchical and stochastic algorithms for radiosity*. PhD thesis, University of Leuven, 1999.
5. G. Coombe, M. J. Harris, and A. Lastra. Radiosity on graphics hardware. Technical report, Univ. of North Carolina, UNC TR03-020, 2003.
6. J. T. Kajiya. The rendering equation. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, pages 143–150, 1986.
7. L. Neumann. Monte Carlo radiosity. *Computing*, 55:23–42, 1995.
8. T. Purcell, T. Donner, M. Cammarano, H. Jensen, and P. Hanrahan. Photon mapping on programmable graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, pages 41–50, 2003.
9. M. Sbert. *The Use of Global Directions to Compute Radiosity*. PhD thesis, Catalan Technical University, Barcelona, 1996.
10. L. Szirmay-Kalos. Stochastic iteration for non-diffuse global illumination. *Computer Graphics Forum (Eurographics'99)*, 18(3):233–244, 1999.
11. L. Szirmay-Kalos. *Photorealistic Image Synthesis Using Ray-Bundles*. D.Sc. Dissertation, Hungarian Academy of Sciences, 2000. www.iit.bme.hu/~szirmay/diss.html.

Finding matching points based on motion-statistics from two views

Zoltán Szlavik¹, László Havasi², Tamás Szirányi¹, and

Analogical and Neural Computing Laboratory, MTA-SzTAKI, Budapest

¹[szlavik;sziranyi]@sztaki.hu

Department of Information Technology, PPCU,

Budapest²havasi@digitus.itk.ppke.hu

Abstract. In the paper a novel approach based on motion- statistics is presented to find matching points in a pair of overlapping images. According to the objectives of the 3D position estimation, the camera calibration requires matching points extraction for which the methodology is described in this article. The uniqueness of the approach is the methodology we applied provides good results even in the cases when traditional techniques are difficult to apply. The use of the motion statistics gives robust results.

1 Introduction

Observation of human or traffic activity from stationary camera is becoming of increasing interest for many applications, semi-mobile traffic control using automatic calibration or tracking humans in a surveillance system.

To estimate the object location in a scene we must know the calibration matrices for each camera in the system. The possibility of calibrating a camera based on the identification of matching points in several views of a scene has been shown in O. D. Faugeras, Q.-T Luong, and S.J. Maybank, "Camera self-calibration: Theory and experiments", ECCV '92, LNCS-Series, Springer-Verlag, 1992, Vol. 588 pp. 321-334.. Estimating camera parameters requires a set of matching points in two or several overlapping views R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd edition, Cambridge University Press, Cambridge, 2003. Matching different images of a single scene could be difficult, because of occlusion, aspect changes and lighting changes that occur from different views. Over the years numerous algorithms for image matching have been proposed. They can be classified into two categories. In "template matching" the algorithms attempt to correlate the grey levels of image patches, assuming that they are similar D. H. Ballard, C. M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1982.Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown Epipolar Geometry", Artificial Intelligence Journal, 1995, Vol.78, pp. 87-119.. This approach appears to be valid for image pairs with small difference; however it may be wrong at occlusion

boundaries and within featureless regions. In "feature matching" the algorithms first extract salient primitives from images (edges or contours) and match them in two or more views. S. T. Barnard, W. B. Thompson, "Disparity analysis of images", *IEEE Trans. PAMI*, 1980, Vol. 2(4), pp. 333-340. J. K. Cheng, T. S. Huang, "Image registration by matching relational structures", *Pattern Recog.*, 1984, Vol. 17(1), pp. 149-159. J. Weng, N. Ahuja, and T. S. Huang "Matching two perspective views", *IEEE Trans. PAMI*, 1992, Vol. 14(8), pp. 806-825. An image can then be described by a graph with primitives as nodes and geometric relations defining the links. The registration then becomes the mapping of the two graphs: subgraph isomorphism. These methods are fast because of the small subset of the image pixels, but they may fail if the chosen primitives cannot be reliably detected.

In our experiment we use standard digital cameras with wide angle. So the field of view is large, the size of features is small; the images are blurred and noisy. The common field of view of the two cameras is about 30°.

The approach we propose in the paper aims to use motion statistics to find matching points in image pairs. We first collect motion statistics and then choose global maximums as candidate matches. This step followed by an elimination of outliers from the set of candidate matches and an optimization to fine tune the locations of candidate pairs. Finally a projective transformation is estimated, which maps points from matching pairs from one image to another.

2 Collecting statistics from motions

Our application field has several special requirements to motion extraction. The videos were created with normal digital cameras with wide angle so the images are blurred and noisy. The size of moving objects has a great variety. There are small patches (walking people) and huge patches (trams or buses) too. The background cannot be extracted perfectly, because we do not want to assume any a priori knowledge about the scene.

In our approach the major assumption is the time synchronization between the cameras. When it exists the motion information can be transformed into motion-statistics. The motion patches are extracted by using simple running-average background subtraction. This method is fast and very sensible with low threshold value. The disadvantage is that it often detects noises and background flashings. In the preprocessing algorithm we are dilating the detected motion patches while these are reaching the local maximums of edge maps. This approach seems a good and simple solution to extract the significant contours of objects. In the future use of statistics we would like to retrieve the size information about objects. This aim needs the flood-fill algorithm with 4 neighbors connectivity thus inhibits the connection of distinct objects. The collected objects have to be filtered according to their size. Some results can be seen in Figure 1.

For each point in the images, two motion-statistics map are assigned: a local and a remote. After the motion detected on the local side, in the points defined

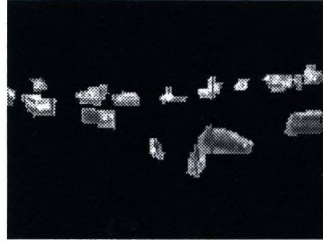


Fig. 1. The detected objects

by the local motion map the local statistics updated by the local motion map. For each point when motion detected on the local side, in the points defined by the local motion map the remote statistics updated by the motion map of the remote side.

The flowchart of the statistics collection algorithm is given in Figure 2:

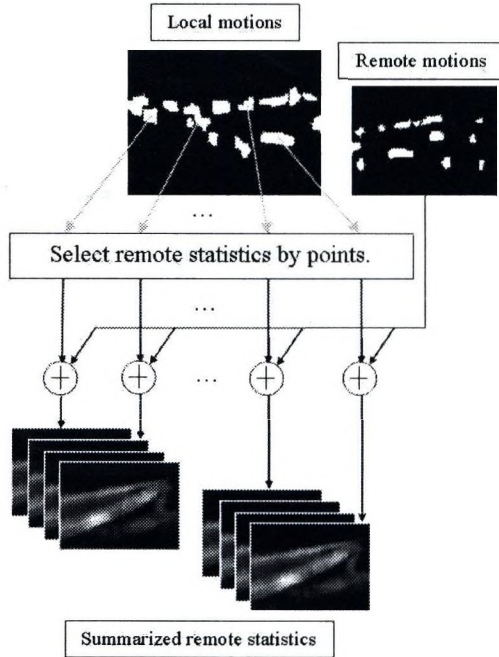


Fig. 2. Collecting motion statistics

3 Extracting candidate matches from motion statistics

As candidate matches we could choose global maximums on local and remote statistical images. However, several problems occur in this case:

1. From global statistics we know image regions where too many motion are present (see Figure 3). In many cases global maximum on remote statistical image could be in this region even in the case if the correspondent point should be in another place (see Figure 3). If we have a point in the first image where the correspondent scene location not in the field of view of the second camera, then the correspondent maximum in remote statistical image will be a point where value of the motion statistics is high.
2. Because of the size of the moving objects the global maximums could be shifted and will be somewhere in the neighborhood of the desired correspondent point. These "shifting" results in cases when different points from local statistical images mapped onto the same point on remote statistical images.

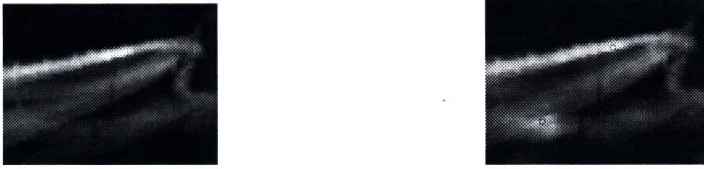


Fig. 3. Left image: global statistics for one of the views. Lighter parts represent regions where more motion is present. Right image: red circle marks the global maximum on this remote statistical image, while the point marked by blue circle should be the correspondent point.

To solve the first problem we need to eliminate points from the set of candidate matches if the global maximum on remote statistical images is a pixel where the value of the motion statistics is greater than some predefined parameter. To get over on second problem we also need to eliminate points from the set of candidate matches if the global maximum on remote statistical image is a pixel, which also present in another candidate pair. The obtained candidate pairs after these two simple steps are presented on next figure:

Consider a candidate match where is a point in the first image and is a point in the second image. Let and be the neighbors of and . If is a good match, we will expect to see many other matches, where if then .

To fine point correspondences we get the neighborhood of the candidate pair in the local statistical images and analyze their pairs in the remote statistical image. We construct clusters from these points using the nearest neighbor classifier.

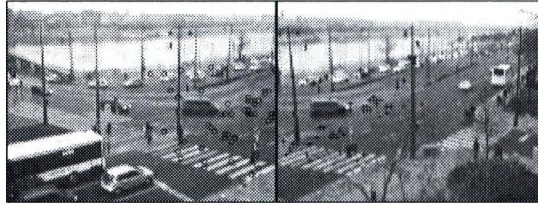


Fig. 4. The extracted candidate point pairs



Fig. 5. The constructed clusters from the points on remote statistical image.



Fig. 6. Candidates for matching pairs in local statistical image. Black plus sign is the point correspondent to local maximum in remote statistical image. Red circle refer to the point correspondent to the centroid of the largest cluster constructed from points in remote statistical image.

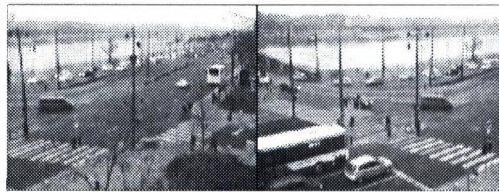


Fig. 7. The selected matching pair for the given remote and statistical images.

4 Determination of the transformation

Consider a set of matches m_1^i, m_2^i . It's required to find a two dimensional projective transformation P that maps each m_1^i to m_2^i . The projective matrix computed from the following linear equation:

$$KP = M_2 \tag{1}$$

Where

$$M_2 = (m_{11}^1, m_{11}^1, \dots, m_{11}^n, m_{12}^1, m_{12}^1, \dots, m_{12}^n) \tag{2}$$

and

$$K = \begin{pmatrix} m_{21}^1 & m_{22}^1 & 1 & 0 & 0 & 0 & -m_{11}^1 m_{21}^1 & -m_{11}^1 m_{22}^1 \\ m_{21}^2 & m_{22}^2 & 1 & 0 & 0 & 0 & -m_{11}^2 m_{21}^2 & -m_{11}^2 m_{22}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{21}^n & m_{22}^n & 1 & 0 & 0 & 0 & -m_{11}^n m_{21}^n & -m_{11}^n m_{22}^n \\ 0 & 0 & 0 & m_{21}^1 & m_{22}^1 & 1 & -m_{12}^1 m_{21}^1 & -m_{12}^1 m_{22}^1 \\ 0 & 0 & 0 & m_{21}^2 & m_{22}^2 & 1 & -m_{12}^2 m_{21}^2 & -m_{12}^2 m_{22}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & m_{21}^n & m_{22}^n & 1 & -m_{12}^n m_{21}^n & -m_{12}^n m_{22}^n \end{pmatrix} \tag{3}$$

If, then we have an overdetermined set of equations, which can be solved using Singular Value Decomposition.

5 Results

Two videos were captured 20 minutes of length from two different locations. The common field of view of the two cameras is about 30% . Then local and remote statistics is constructed and matching points is extracted by the above-described algorithm. The results can be seen on Figure 8.



Fig. 8. Using the above described algorithm 49 matches have found on to overlapping images

Using the obtained projective transformation from the extracted 49 matches a composite image constructed from two overlapping views of the scene:

6 Conclusions and ongoing work

We have shown that the use of motion statistics helps to find point correspondences between two overlapping views of the same scene. In further work we plan



Fig. 9. The constructed composite view from the two overlapping images. The alignment of the two images is not precise. In further work we plan to refine the obtained projective transformation through optimization.

to refine the obtained projective transformation through optimization. After construction of composite views tracks from multiple views could be combined into a single track. Analysis of the obtained projective distortion provides the basis for the calibration algorithm.

References

1. O. D. Faugeras, Q.-T Luong, and S.J. Maybank, "Camera self-calibration: Theory and experiments", ECCV '92, LNCS-Series, Springer-Verlag, 1992, Vol. 588 pp. 321-334.
2. R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd edition, Cambridge University Press, Cambridge, 2003
3. D. H. Ballard, C. M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1982.
4. S. T. Barnard, W. B. Thompson, "Disparity analysis of images", IEEE Trans. PAMI, 1980, Vol. 2(4), pp. 333-340.
5. J. K. Cheng, T. S. Huang, "Image registration by matching relational structures", Pattern Recog., 1984, Vol. 17(1), pp. 149-159.
6. J. Weng, N. Ahuja, and T. S. Huang "Matching two perspective views", IEEE Trans. PAMI, 1992, Vol. 14(8), pp. 806-825.
7. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown Epipolar Geometry", Artificial Intelligence Journal, 1995, Vol.78, pp. 87-119.

Kerámiarészecskék bevonatának jellemzése elektronmikroszkópos kép feldolgozásával

Tomolya Kinga¹, Gácsi Zoltán¹, Kovács Árpád¹,

Miskolci Egyetem, Anyagtudományi Intézet, Fémteni Tanszék

¹3515 Miskolc-Egyetemváros

tomolyakinga@freemail.hu

46-565-111-23-02

Kivonat A cél alumínium mátrixú SiC szemcsékkel erősített kompozitok gyártása és vizsgálata, amelyben a SiC szemcsék rézzel bevontak. A bevonási módszer a kémiai redukciós eljárás, amelyet két fő műveletből áll: az aktiválásból és a bevonásból. Első lépésben a szemcsék aktiválását végeztük el, amelyhez kétféle vegyszert használtunk. Az optimális aktiváló szer kiválasztását követően a bevonást különböző oldatban tartási idővel végeztük el, keresve a megfelelő mennyiségben bevont szemcsék mennyiségét és mértékét. A különböző paraméterek hatásának összehasonlítása céljából pásztázó elektronmikroszkópos felvételeket készítettünk, amelyeket ezt követően képelemző berendezés segítségével jellemeztünk. Vizsgáltuk a bevonás mértékét, illetve a bevont szemcsék mennyiségét.

1. Bevezetés

A monotektikus ötvözetrendszereket gyakran siklócsapágyak anyagaként alkalmazzák. Ezeknek a siklócsapágyaknak az előállítása történhet porkohászati úton [1-3], strip casting eljárással [4-7], plannar flow casting eljárással [8] és rheocasting eljárással [9-11]. A siklócsapágyak előállítására lézeres felületkezelési technikát választottuk.

A lézeres felületkezeléssel előállított monotektikus ötvözeteknél nagy sebességgel történik a kristályosodás és olyan szerkezet alakul ki, ahol finom lágy kiválások találhatók, amelyek szilárd kenőanyagként szolgálnak. A lézeres ötvözés lényege, hogy a darab teljes tömegének tulajdonságait változatlanul hagyva csak a felületi réteg tulajdonságait változtatjuk meg.

A ma alkalmazott siklócsapágyak leggyakoribb anyag: ólom-ón-bronz és alumínium-réz-ón. Ezek a siklócsapágy anyagok már az igénybevételi határukat elérték, ezért van szükség nagyobb igénybevételű siklócsapágyak fejlesztésére

2. SiC szemcsék bevonása rézzel

A SiC szemcsék rézzel történő bevonásának egyik módszere a kémiai redukciós eljárás. A kémiai redukciós bevonó oldat legfontosabb alkotói a fémsó és a megfelelő redukáló közeg. Az oldat a gyakorlatban termodinamikailag stabil, de ha egy megfelelően katalizált felülettel érintkezik, akkor instabil [Cottel, 1994].

Általánosságban elmondható, hogy a rézzel történő bevonás alapvetően két műveletre bontható. Az egyik műveletben a felületet alkalmassá teszik a bevonásra tisztítás, maratás, aktiválás által. A második fő lépésben a rezes oldatból viszik fel a felületre [Cottel, 1994].

A bevonás előtt a felület előkészítése szükséges. A felület előkészítése is több lépéses folyamat. Az előkészítés legfontosabb lépése a katalizálás, amely során atomi méretű fémmennyiséget visznek fel a felületre. A katalizáláshoz PdCl₂ nemesfém sósavas oldatát alkalmaztuk [Cottel, 1994], amelyben szobahőmérsékleten, 30 percen keresztül tartottuk a szemcséket. Ezt követően leszűrtük, desztillált vízzel tisztítottuk, majd 100 °C-on szárítottuk.

Rezes bevonás esetén az oldatban a réz forrása egy egyszerű rézsó, amely jelen esetben réz-szulfát. Esetünkben formaldehidet, azaz hangyasavat (HCO-OH) használtunk, amely a gyakorlatban is igen elterjedt redukáló közeg [Cottel, 1994].

A rézionok semlegesítődése során a réz-szulfát rézionjai a szemcsék felületére redukálódnak elektronfelvétel következtében. A formaldehid, mint redukáló közeg biztosítja a szükséges elektronokat a folyamathoz, amely alkalmazása azonban nagy pH-t igényel (pH>13), ezért az oldathoz lúgos kémhatású anyag hozzáadása szükséges, ami jelen esetben NaOH. Az egyszerű rézsók kb. 4-es pH érték fölött oldhatatlanok, és mivel bázikus közeg szükséges a bevonáshoz, így a rézsók komplexek segítségével vagy kelátképzőkkel tartják oldatban. [Cottel, 1994]

A bevonás megvalósítása során az aktivált szemcséket a megadott oldatban tartottuk. Az erősen kék színű oldat a SiC szemcsék beszorását követően pezsegni kezdett, a szemcsék színe szemmel láthatóan megváltozott, szürkés-vöröses színű lett.

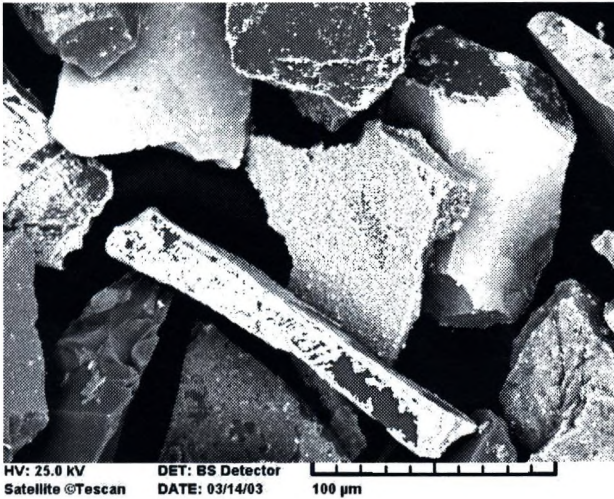
Ezzel a fürdővel létrehozott réteg finomszemcsés. A bevonat a szemcsék felületén véletlenszerűen szétszórt, katalizáló részecskéken képződik. A lerakódó réteg szemcseszerkezetét nagymértékben a felület morfológiája szabja meg. Emellett a fürdő összetétele is meghatározó faktor [Cottel, 1994].

3. SiC szemcsék bevonatának jellemzése

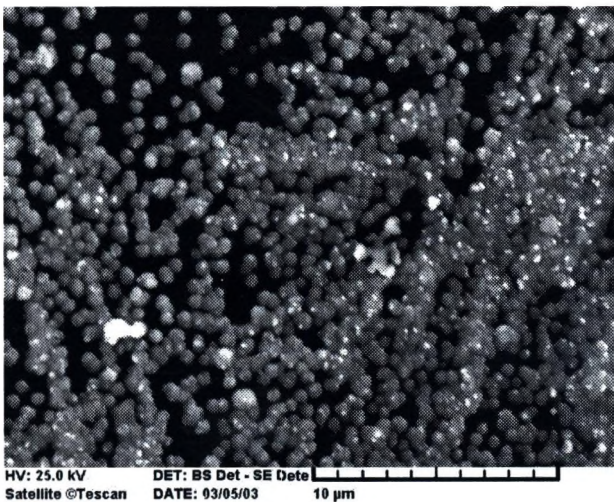
A létrehozott bevonatokról pásztázó elektronmikroszkóppal felvételeket készítettünk. Az 1. ábra a SiC szemcsék 500-szoros nagyítású képét mutatja. A sötét-szürke színű szemcséken a világos színű bevonat a réz. Látható, hogy a rézréteg először a szemcsék élein jelenik meg, illetve, hogy a réteg nem teljesen fedi a szemcséket, szigetszerűen rakódik le. A bevonatot nagyobb nagyításban megfigyelve látható a bevonat felépítése. A bevonat gömbszerű részecskékből épül fel (2. ábra). A szemcseméretet P220-ként adtuk meg, ami 80-120 µm méretű szemcséknek felel meg.

4. Bevonás mértékének jellemzése képelemző segítségével

A SiC szemcsék bevonását 30, 60 és 90 percen keresztül hajtottuk végre, azaz ennyi ideig tartottuk az oldatban a szemcséket kevergetés alkalmazása mellett.



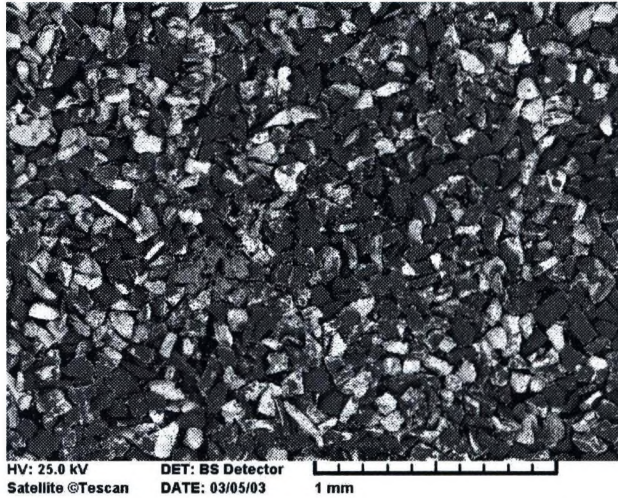
1. ábra. Bevont SiC szemcsék SEM képe, $N = 500x$



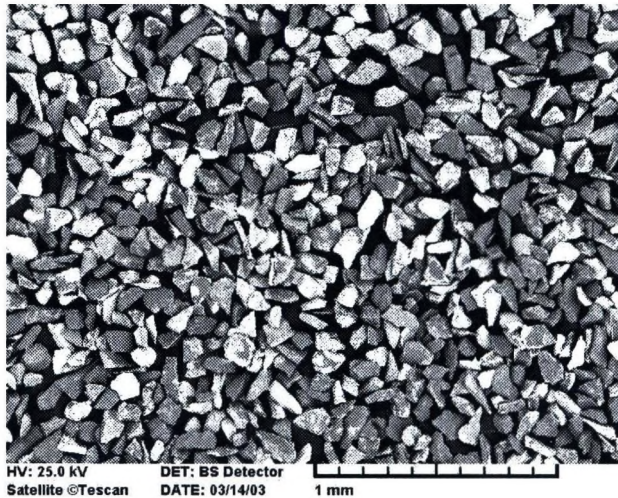
2. ábra. A bevonat pásztázó elektronmikroszkópos felvétele $N = 5000x$

A különböző bevonási idők hatását pásztázó elektronmikroszkópos felvételek képelemzős vizsgálatával elemeztük. A 3., 4. és 5. ábra a különböző ideig bevont porok felvételeit mutatja.

A pásztázó elektronmikroszkóp segítségével felvett képeken a létrehozott rézréteg világos részként, míg a SiC szemcsék sötét részként vannak jelen. A képekről szabad szemmel is megállapítható, hogy bevonási idő növelésével növekszik a bevont szemcsék száma és az egyes szemcsék bevonásának mértéke is. Cél az



3. ábra. Bevont SiC szemcsék SEM-os képe (Bevonási idő: 30 perc), $N=50\times$

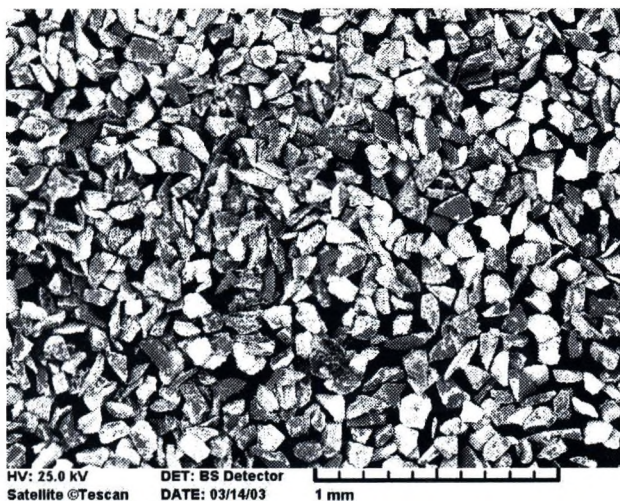


4. ábra. Bevont SiC szemcsék SEM-os képe (Bevonási idő: 60 perc), $N=50\times$

optimális bevonási idő megtalálása. Az optimális bevonási idő meghatározásánál nem csak a felvitt réz mennyiségét vesszük figyelembe, hanem a gazdaságosságot is. Ennek megállapítására képelemző segítségével történő mérésekre volt szükség.

Első lépésben a képek detektálását, azaz a mérni kívánt képrészletek elkülönítését végeztük el.

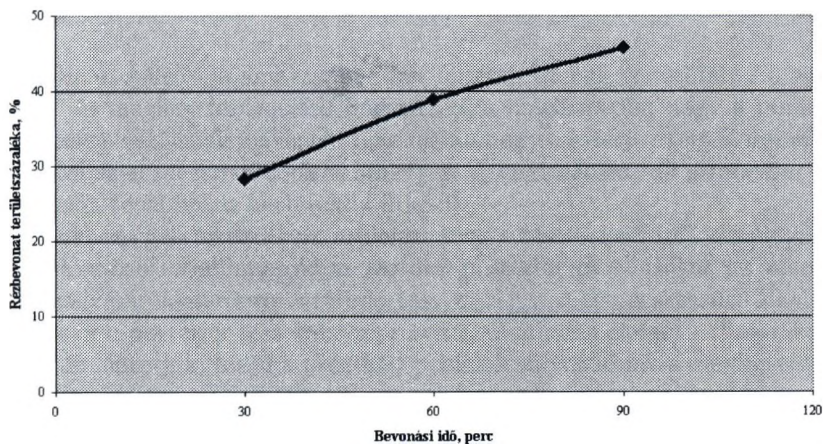
A detektált képeken ezt követően a területszázalék mérését végeztük el, amelynek eredményei a 6. ábrán láthatók. A területszázalék a detektált objektumok



5. ábra. Bevont SiC szemcsék SEM-os képe (Bevonási idő: 90 perc), $N=50\times$

látóterén belül mért összes területe osztva a mérőkeret nagyságával, százalékos formában megadva [Gácsi, 2001].

A mérési eredmények is igazolják, hogy a bevonási idő növekedésével a bevonás mértéke is növekedik. A bevonás mértéke azonban az idő növekedésével nem lineáris, hanem egy maximumhoz közelít. A 60 perces bevonási idő alatt a szemcsék mintegy 40



6. ábra. Egy-egy látómezőben mért rézréteg területi aránya

5. Összefoglalás

A dolgozat során SiC szemcsék rezes bevonásával foglalkoztunk. A cél az optimális bevonási idő kiválasztása volt. Ennek megállapításához a bevont szemcsékről pásztázó elektronmikroszkópos felvételeket készítettünk, amelyeken a jól elkülöníthető rézréteg mennyiségét képelemző segítségével mértük meg. A kapott eredményeket összehasonlítva megállapítható, hogy a bevonási idő növekedésével a bevonás mértéke is növekszik. Az optimális bevonási idő meghatározásához azonban nem csak a felvitt réz mennyiségét kell figyelembe venni, hanem a gazdaságosságot is. Így megállapítható, hogy a 60 perces művelet mondható optimális időtartamnak.

Hivatkozások

1. Cottel C. M., Sprague J. A., Smidt, Jr F. A.: Surface Engineering, Volume 5, ASM Handbook, 1994, pp. 311-322
2. Davidson A. M., Regener D.: A comparison of aluminium-based metal-matrix composites reinforced with coated and uncoated particule silicon carbide, Elsevier Science Ltd., 2000., pp. 865-869
3. Kandil Amr, Müller Katrin: Schmelzmetallurgische Herstellung einer SiC-partikelverstärkten Al-Legierung A7075, Zeitschrift für Metallkunde, 2000/10., pp. 812-821
4. Gácsi Zoltán, Sárközi Gábor, Réti Tamás, Kovács Jenő, Csepeli Zsolt, Mertinger Valéria: Sztereológia és képelemzés, Egyetemi Tankönyv, 2001, p. 198

Vizuális objektumkövetés részecskeszűrőkkel lokális, likelihood alapú mintavételezéssel

Torma Péter¹, Szepesvári Csaba²

¹ Eötvös Lóránd Tudományegyetem

tyus@tateyama.hu

² MTA SZTAKI

szcsaba@sztaki.hu

Kivonat A vizuális objektumkövetés modellezhető, mint egy nem-lineáris és nem Gauss-i dinamikával és megfigyelési függvénnyel rendelkező szűrési probléma, ahol a megfigyelési modell zaja tipikusan igen kicsi. Ha a dinamikai és a megfigyelési modell erősen nemlineáris és/vagy nem Gauss-i, a jól ismert Kálmán szűrő és kiterjesztései nem, vagy csak rossz eredményekkel alkalmazhatóak. Sajnos a szokásos részecskeszűrők az ilyen feladatokon hasonlóan gyenge teljesítményt mutatnak, mivel a megfigyelési modell zaja kicsi. Ebben a cikkben a részecskeszűrők egy alkalmas módosítását javasoljuk és megmutatjuk, hogy a módosított részecskeszűrő az említett speciális esetben a szokásos részecskeszűrőknél jobban teljesít. Egy rendszámkövetési feladaton mérve az új szűrő teljesítménye a szokásos szűrőkénél egy nagyságrenddel bizonyul jobbnak.

1. Bevezetés és motiváció

A jelfeldolgozás valamint a kotroll-elmélet egyik fontos, s ma már több évtizede intenzíven vizsgált területe, a dinamikus szűrők elmélete. A híres Kálmán-szűrő ennek a feladatnak az egyik első és máig gyakran használt megoldása. Fontosságát nem csupán elméleti szépsége és egyszerűsége, hanem gyakorlati haszna is adja.

A Kálmán-szűrő alkalmazásához két feltételnek kell teljesülni. Az egyik a dinamikai és megfigyelési modell linearitása, a másik pedig, hogy a modell zajok normális eloszlásúak legyenek. Igaz ugyan hogy a kiterjesztett Kálmán szűrő segítségével ezen megszorítások valamelyest gyengíthetőek, de a módszer alkalmazhatósága továbbra is korlátokba ütközik.

Ezen nehézségek leküzdésére született meg a részecskeszűrők elmélete [2] [1] [4]. A különböző időpillanatokhoz tartozó posterior eloszlásokat itt részecske-részecskesúly párokkal reprezentáljuk: $\{(x_i, w_i)\}_{i=1}^N$. A részecskeszűrő alkalmazásakor először a posterior részecske-reprezentáció minden elemét a dinamikai modell alapján léptetjük, majd a predikált pozíciók súlyozódnak a megfigyelési modell alapján.

Természetesen ennek az algoritmusnak is számtalan kiterjesztése ismert az irodalomban [3]. Ennek ellenére az algoritmus számítógépes látásbeli alkalmazásai gyakran nagyon gyenge teljesítmény mutatnak. Ennek oka, hogy ezen feladatokban a megfigyelési modell erős, azaz sokkal kevésbé zajos, mint a dinamikai

modell. Azt is jelenti ez, hogy a megfigyelési modell likelihood függvénye nagyon „csúcsos” az egyes módusok körül és ezeket a csúcsoakat a dinamikai modellből származó predikció csak nagy részecskeszám esetén képes „eltalálni”. Ennek eredményeképp a részecskék nem fognak lényeges információt hordozni [5].

Ebben a cikkben az ebből származó probléma kiküszöbölésére javasolunk egy algoritmust. Először a legnépszerűbb részecskeszűrő algoritmust, majd az új algoritmust ismertetjük. Ennek elméleti vizsgálata után megmutatjuk az absztrakt algoritmus egy sikeres alkalmazását japán rendszámok követésére. A cikket rövid összefoglalás zárja.

2. Részecskeszűrők

Legyen X_t az a Markov folyamat, amelynek időbeli változását követni szeretnénk. Feltesszük, hogy ismerjük $p(X_{t+1}|X_t)$ -t a folyamat dinamikai modelljét, valamint hogy minden pillanatban rendelkezésünkre áll egy megfigyelés Y_t , mely szintén sztochasztikus módon függ X_t -től: $p(Y_t|X_t)$, s szintén feltesszük hogy a $p(Y_t|X_t)$ sűrűségfüggvény is ismert. A cél a poszterior eloszlás becslése $p(X_t|Y_1, \dots, Y_t, X_0)$, erre a kifejezésre a következő rekurzió adódik:

$$p(X_t|Y_1, \dots, Y_t, X_0) \propto p(Y_t|X_t) \int p(X_t|X_{t-1})p(X_{t-1}|Y_1, \dots, Y_{t-1}, X_0) dX_{t-1}.$$

Tekintettel arra hogy ez általában egy magas dimenziós integrál, amely egyáltalán nem értékelhető ki, valamilyen approximációra van szükségünk. Reprezentáljuk ezért a t -edik időpontbeli poszteriorot egy részecskesereg segítségével ($\{(x_t^i, w_t^i)\}_{i=1}^N$):

$$p(X_t|Y_1, \dots, Y_t, X_0) \approx \sum_{i=1}^N w_t^i \delta_{x_t^i}(X_t).$$

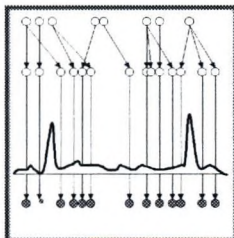
A fentiekből már egyszerűen adódik a szokásos részecskeszűrő algoritmus főciklusa:

1. **Részecske-predikció.** Az összes részecskének sorsoljuk ki az utódját $x_{t+1}^i \sim p(x_{t+1}|x_t^i)$.
2. **Részecske súlyozás.** A megfigyelési modell segítségével számoljuk ki az új súlyokat $w_{t+1}^i = p(Y_{t+1}|x_{t+1}^i)$.

A fentiekén kívül szükséges időnként a részecskehalmazokból kiszűrni a túl kis súlyú részecskéket. Ezt egy újramintavételezési lépés beiktatásával szokás megtenni. A részecskeszűrőről ld. bővebben a [2] [1] cikketet.

3. Az új algoritmus: Lokális mintavételezés a likelihood függvényből

Mint azt már a bevezetőben is említettük a szokásos részecskeszűrési algoritmus hatékonysága csekély, ha a megfigyelés zaja kicsi. A fellépő degradációra egy



1. ábra.

A szokásos részecskeszűrő algoritmus viselkedése kis megfigyelési zaj esetén. A felső sorban szereplő $t - 1$. időpontbeli posterior minden elemét a sztochasztikus dinamika segítségével léptetjük, majd az így kapott részecskéket a megfigyelési modell alapján súlyozzuk. Látható hogy a t . időpontbeli poszterior eloszlás már nem informatív.

példa látható az 1. ábrán. Természetesen ez a probléma kezelhető a részecskeszám növelésével, de ez az algoritmus sebességének rovására megy.

Egy alternatív megoldás, ha a részecskeszűrés főciklusát az alábbi lépésekre cseréljük ki:

1. **Részecske-predikció.** Az összes részecskének dobjuk ki az utódját $x_{t+1}^i \sim p(x_{t+1}|x_t^i)$.
2. **Lokális mintavételezés a megfigyelési likelihood alapján.** Dobjuk ki $z_{t+1}^i \sim \frac{p(Y_t|z)g(x_{t+1}^i - z)}{p(Y_t|\cdot)*g(x_{t+1}^i - \cdot)}$ -ből, ahol g egy megfelelően választott lokális tartójú ablakfüggvény.
3. **Részecske súlyozás.** A megfigyelési modell segítségével számoljuk ki az új súlyokat $w_{t+1}^i = p(Y_t|\cdot) * g(x_{t+1}^i - \cdot)p(z_{t+1}^i|x_t^i)/p(x_{t+1}^i|x_t^i)$

Az algoritmusban szereplő g ablakfüggvényről feltesszük, hogy kompakt tartójú, nem-negatív, és hogy $g \in L^1 \cap L^2$. Gondolhatunk itt kompakt tartóra megszorított Gauss függvényre, vagy egy kompakt halmaz karakterisztikus függvényére.

Az algoritmus első lépésben a dinamikai modellből predikál, akárcsak az eredeti. A második lépésben a részecskék a megfigyelési modell csúcsai felé mozognak (persze sztochasztikus módon). Ezen mozgás nagyságát szabályozza az ablakfüggvény. Az utolsó lépés biztosítja, hogy a kapott részecske-reprezentáció a posterior torzítatlan becslése legyen. Ez utóbbi tulajdonság bizonyításával foglalkozik a következő fejezet.

4. Az algoritmus torzítatlansága

1. **Tétel.** Rögzsítsünk egy tesztöleges t időpontot. Tegyük fel, hogy g kompakt tartójú függvény, amelyre $g \geq 0$ és $\int g = 1$. Tegyük fel, hogy $f_y = p(y|x)$ korlátos és integrálható. Ekkor a fenti algoritmus a posterior torzítatlan becslését adja,

az alábbi értelemben:

$$E[w_j h(z_j) | Y_1, \dots, Y_t, X_0] = \int h(x) p(x | Y_1, \dots, Y_t, X_0) dx,$$

ahol h egy tetszőleges integrálható függvény.

Bizonyítás: Legyen h egy tetszőleges függvény és legyen $I = E[w_j h(z_j) | H_t]$, ahol $H_t = Y_1, \dots, Y_t, X_0$.

A teljes valószínűség tétele szerint: $I = E[E[w_j h(z_j) | x_j, H_t] | H_t]$. A z_j és w_j definíciója alapján,

$$\begin{aligned} E[w_j h(z_j) | x_j, H_t] &= \int h(z) \frac{(f * g)(x_j) p(z)}{p(x_j)} \frac{f(z) g(x_j - z)}{(f * g)(x_j)} dz \\ &= \int h(z) \frac{p(z)}{p(x_j)} f(z) g(x_j - z) dz, \end{aligned}$$

ahol $f(z) = p(Y_t | z)$ és $p(x) = p(x | X_t)$. Így

$$\begin{aligned} I &= \int \int h(z) \frac{p(z)}{p(x)} f(z) g(x - z) dz p(x) dx \\ &= \int \int h(z) p(z) f(z) g(x - z) dz dx. \end{aligned}$$

Az integrálok felcserélhetőségéből már következik is az állítás, kihasználva, hogy $\int g = 1$.

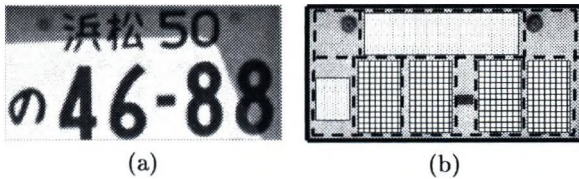
Természetesen fontos vizsgálni az algoritmus előnyeit is a korábbiakhoz képest. Ezen vizsgálatok, melyeket itt helyhiány miatt nem tudunk részletezni, azt igazolják, hogy nem túl szigorú feltételek mellett az algoritmus csökkenti a részecskesúlyok szórását, így növelve az algoritmus hatékonyságát.

5. Alkalmazás egy rendszámkövetési feladaton

A fenti algoritmust egy japán rendszámkövetési feladaton hasonlítottuk össze a szokásos részecskeszűrő algoritmussal. Kísérleteinkben az állapotterek a rendszám körvonalát reprezentáló paralelogramma konfigurációs teréből származtatuk.

A megfigyelési modell a japán rendszámok rendkívül kötött, skálafüggetlen geometriai struktúráját használja ki (2. ábra). Egy képen egy rendszám hipotézis likelihoodja az egyes területek likelihoodjainak szorzata. Az egyes területek likelihood-jait heurisztikus modellek segítségével számoltuk. Egy modell azt számolja, hogy rácsozott területek mennyire tűnnek számnak, míg egy másik pedig azt, hogy az élek milyen erősek a rendszám szélén, stb.

A dinamikai modell egy másodrendű auto-regresszív folyamat és egy inicializációs eloszlás keveréke, amelynek paramétereit kézzel hangoltuk. Itt az inicializációs eloszlás arra szolgál, hogy az újonnan feltűnő rendszámokra is rátaláljon a



2. ábra.

Egy japán rendszám(a) és a kötött geometriai struktúra(b). A kockás területek vastag vonalú betűk területét jelölik, a pöttyözöttek vékony vonalú betűket. A szaggatott vonalak az üres területeket jelölnék, míg a folytonos vonalak éleket.

rendszer, míg az auto-regresszív folyamat a már megtalált rendszámok követését szolgálja.

A megfigyelési eloszlás alapján történő lokális mintavételezést partícionált mintavételezéssel végeztük. Először a jellemzők 1 dimenziós hisztogramját számoltuk a predikált pozíció egy lokális környezetében, majd az így kapott új vertikális pozíció környékén számolt új horizontális pozíciót határoztuk meg hasonló módszerrel. Ennek megfelelően kísérleteinkben az ablakfüggvény(g) egy megfelelően skálázott karakterisztikus függvény volt.

6. Eredmények

A 100 részecskével működő új algoritmust hasonítottuk össze a 750 részecskével működő szokásos algoritlussal. Mindkét algoritmus valószínűben futott egy 1.7 GHz-s Intel gépen. Ezen belül az új algoritmus mintegy másfélszer gyorsabban és lényegesen megbízhatóbban futott. Míg a szokásos algoritmus 48 képet processzált másodpercenként, addig új algoritmusunk 65 képkocka feldolgozására volt képes.

A 3.ábra egy példát mutat az algoritmus működésére. A fekete praleogram-mák(téglalapok) mutatják a becsült posterior eloszlás várható értékét a különböző időpontokban.

Az algoritmus teljesítményét ugyanazon videón 100-szor lefuttatva teszteltük. Azt mértük, hogy milyen valószínűséggel esik a posterior várható értéke elég közel a rendszám valódi helyéhez. Ezt mutatja a 4. ábra. Jól látszik, hogy ugyan már a szokásos algoritmus is elég jó eredményt mutat, de az általunk konstruált módszer sokkal jobban viselkedik új rendszámok feltűnésekor, azokra sokkal gyorsabban rátalál.

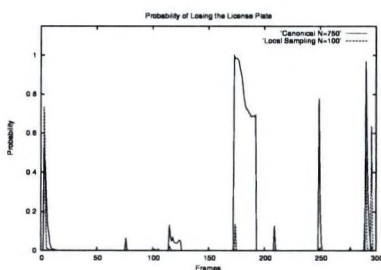
7. Összefoglalás

Cikkünkben ismertettünk egy új algoritmust amely a részecskeszűrők teljesítményét lényegesen megjavítja azon esetekben amikor a megfigyelési modell erős, azaz a sokkal kevésbé zajos mint a dinamikai modell. Megmutattuk, hogy az algoritmus torzítatlan becslést ad. Az algoritmus működését egy rendszámkövetési



3. ábra.

Képkockák a tesztvideóból. A felvétel egy NTSC kamerával készült. A fekete paralelogrammák(téglalapok) mutatják a becsült posterior eloszlás várható értékét a $t = 7, 25, 30, 86, 100, 115$ időpontokban.



4. ábra.

A rendszám követésének valószínűsége. A szokásos (Canonical) algoritlussal 750 részecskével, valamint az új algoritlussal(Local Sampling) 100 részecskével.

Látható, hogy az új algoritmus lényegesen jobb eredményt produkál.

feladaton teszteltük, és azt találtuk, hogy az új algoritmus mind sebesség, mind pontosság tekintetében jobb a szokásos részecskeszűrési algoritmusnál.

Hivatkozások

1. Arnaud Doucet: On Sequential Simulation Based Methods for Bayesian Filtering. *Statistics and Computing* (1998) 197-208
2. Jun S. Liu and Rong Chen: Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association* (1998) 1032-1044
3. Pitt, Michael K. and Neil Shephard: Filtering via simulation: Auxiliary particle filter. *Journal of the American Statistical Association*(1999) 590-599
4. Isard, Michael and Andrew Blake: CONDENSATION – conditional density propagation for visual tracking. *International Journal Computer Vision* (1998) 5-28
5. Péter Torma and Csaba Szepesvári: LS-N-IPS: an Improvement of Particle Filters by Means of Local Search. *Proc. Non-Linear Control Systems(NOLCOS'01)* St. Petersburg, Russia 2001.

Validált kvantitatív PET szimulátor használata az orvosi képfeldolgozásban és diagnosztikai kutatásokban

Valastyán Iván¹, Balkay László¹, Emri Miklós¹, Trón Lajos¹

DEOEC, PET Centrum, Debrecen vivan@pet.hu

Kivonat A szimulált Pozitron Emissziós Tomográfias vizsgálatok jelentős szerepet játszanak a valódi PET vizsgálati képek korrekciós, rekonstrukciós, valamint a képminőség javítását célzó algoritmusok fejlesztésében és tesztelésében. A korrekciókra példa a foton Compton szóródásának, a véletlen koincidenciák miatt bekövetkező események, a testszövetben fellépő foton gyengítés és a detektorok térbeli felbontásának korrekciója. A kvantitatív PET szimulátor alkalmas dinamikus radioaktivitás-eloszlás szimulálására az egyes szöveti régiókban tetszőleges strukturális információt tartalmazó (MR,CT) bemeneti képek megadása esetén. A validálás biztosítja, hogy a szimulált képek és a konkrét kameratípussal mért képek egymástól ne különbözzenek jelentős mértékben. A szimulátor másik felhasználási területe az agykutatás. Az agy egyes területeinek illetve a receptorok működésének szimulálása, a receptorok elhelyezkedése segíthet megérteni az agyban lejátszódó folyamatok egy részét.

1. Első fejezet

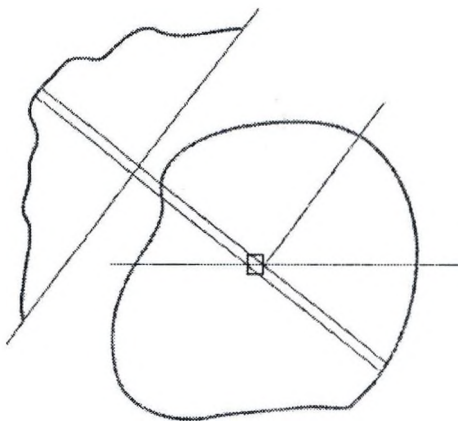
Analitikus PET szimulátor

A szimulátor fejlesztése Matlab környezetben történt. Az intézetünkben kifejlesztett platformfüggetlen orvosi keretrendszerre (VPP) épülő, C++-ban kifejlesztett verziónak a validálási szakaszában tartunk. Megkezdttük az MPI (Message Passing Interface) környezetben futtatható verzió alap algoritmusainak implementálását (rekonstrukciós, korrekciós módszerek).

Az analitikus módszer az adatgyűjtés folyamán a valódi vizsgálatok esetében fellépő torzító hatásokat (Compton szóródás, gyengítés, véletlen koincidenca, térbeli felbontás) analitikus függvényekkel írja le. A módszer gyorsabb, mint a Monte Carlo szimuláció, ahol az egyes fotonok útját követik nyomon az adatgyűjtés során, illetve ezek a függvények jól meghatározottak egy emberi agy esetében, mivel közelítőleg szimmetrikus és homogén szövetekből áll. A pontos radioaktivitás eloszlás egy PET vizsgálat esetén az optimális adatkorrekciótól, a rekonstrukciós módszertől és az analizáló algoritmusoktól függ. A program a valós radioaktivitás eloszlást a bemeneti strukturális információt tartalmazó (például szegmentált MR) képek és a tracer kinetikai modell alapján számítja ki. Perfúziós képek esetében a szimulált aktivitás eloszlást a kiválasztott vérgörbe, illetve az egyes régiókban a véráramlás együtthatói határozzák meg.

A szimulált szinogram adatok előállítása az alábbi fizikai modell alapján történik: $P_k = (T_k * A_k + S_k) * N_k + R_k$

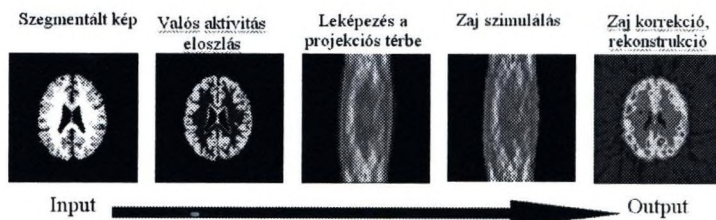
ahol P_k egy adott projekció k -adik detektor pozícióját jelöli; T_k az annihilációk száma; A_k a gyengítési tényező; S_k a Compton szóródás mértéke; N_k normalizációs faktor; R_k a véletlen koincidenciák száma. A szimulátor a képeket a projekciós térbe transzformálja a kamera pont-szétterjedés függvényének (PSF) segítségével. 1.



1. ábra. Az aktivitáseloszlás leképezése a projekciós térbe

2. A szimulált PET képek előállításának lépései

A strukturális információt tartalmazó bemeneti kép, a végörbe és a kinetikai modell paraméterek segítségével előállítjuk a valós aktivitás eloszlást, majd ezt a képet leképezzük a vetületi térbe (2. ábra). A projekciós térben a szinogramon elvégezzük a kamera torzító hatásait, majd hozzáadjuk a statisztikus zajt, utolsó lépésben végrehajtjuk a torzító hatások korrekcióját 3.



2. ábra. Szimulált kép előállításának lépései



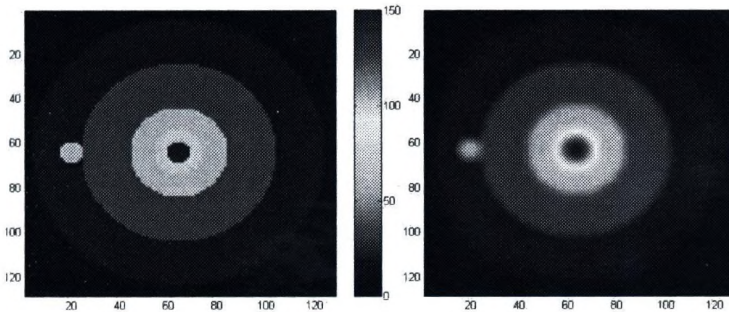
3. ábra. Torzító hatások és korrekciójuk a szinogramon

Dinamikus szimuláció esetén az aktivitás-eloszlás időbeli változását stacionárius aktivitás-eloszlások sorozatával modellezzük.

3. Képkorrekciók gyűrű fantom esetén

A detektor térbeli felbontás korrekciója a kép térben

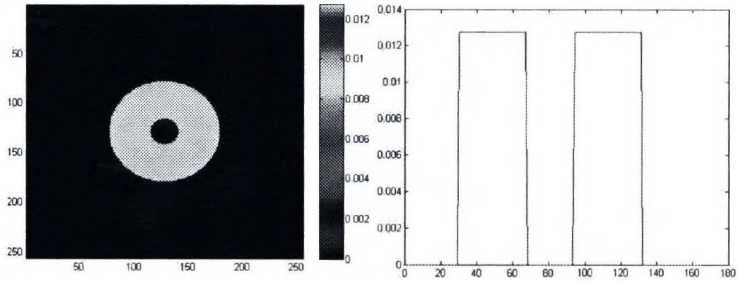
A kamerarendszer felbontása a gyűrű középpontjában a legjobb, míg távolodva a középponttól a detektorok felé, sugárirányban csökken. Ennek az egyébként kameratípusra jellemző paraméternek a segítségével végezzük el a felbontás korrekciót a kép térben. [1].



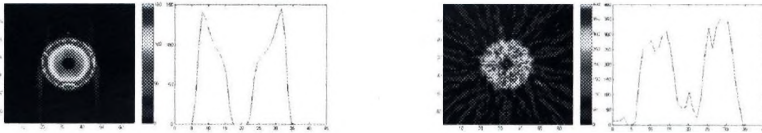
4. ábra. Térbeli felbontás korrekció

4. Fizikai (adatgyűjtési) korrekciók

A szimulátor alkalmas arra, hogy tanulmányozzuk a szimulált kameratípus adatgyűjtési és képfeldolgozási jellemzőit. Az adatgyűjtés során fellépő torzító hatásokat gyűrű alakú fantomon vizsgáltuk. A következő ábrák az egyes korrekciók lépéseit szemléltetik.



5. ábra. A gyűrű fantom és profilgörbéje



6. ábra. Gyengítési effektus és korrekciója fantom esetében

5. Eredmények

Az ábrán egy virtuális elliptikus fantom valós és szimulált PET vizsgálata látható. A szimulált és a valódi kép között csak kis mértékű eltérést lehet tapasztalni. 9

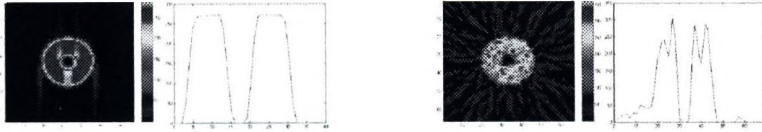
Perfúziós PET vizsgálatok szimulációját szegmentált MRI képek és a két-kompartmentes tracer kinetikai modell segítségével állítjuk elő. A szegmentációs technika segítségével az egyes agyi területekhez különböző perfúziós értékeket rendelünk. Legegyszerűbb esetben a szegmentált kép a fehér és szürke állományt tartalmazza. (10). A 11 ábra receptor specifikus ligandok kötődését szimulálja.

6. Köszönetnyilvánítás

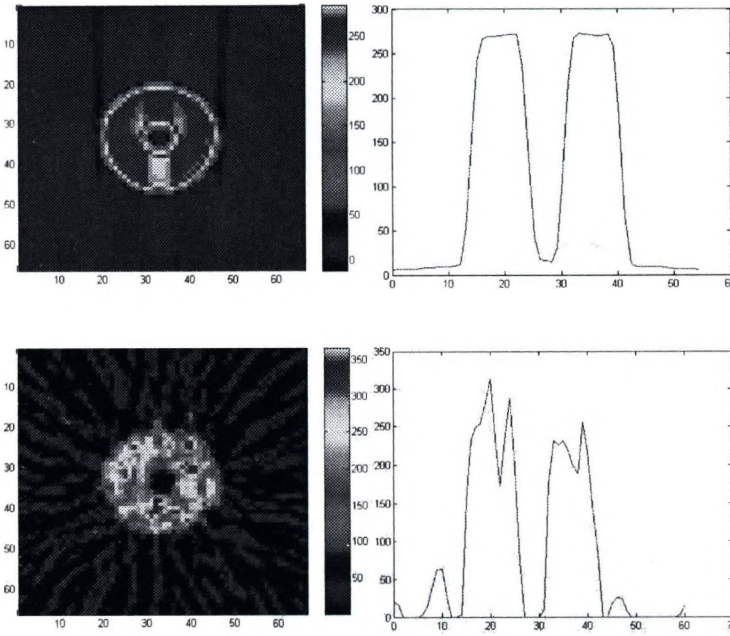
A projek az IKTA 153/2003, IKTA-00006/2001 és NKFP-1/A/0010/2002 pályázatok segítségével készült.

Hivatkozások

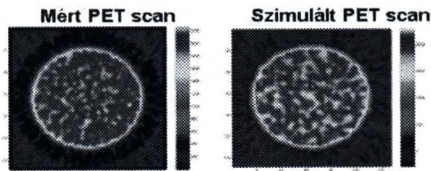
1. H. Zaidi: Scatter modeling and correction strategies in fully 3D PET. Nucl. Med. Communications, 2001,22,1181-1184
2. Y. Ma, C.A. Evans: Analytical Modeling of PET Imaging with Correlated Functional and Structural Images. IEEE Trans. Nucl. Sci. Vol.44,NO.6,2439-2444,1997



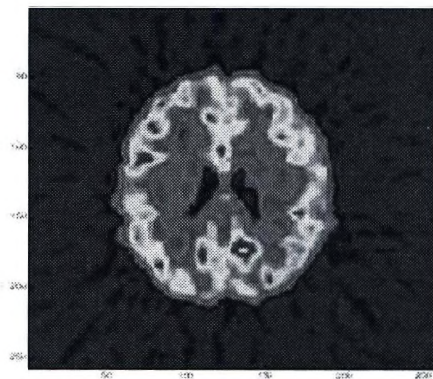
7. ábra. Véletlen koincidenca események és korrekciójuk



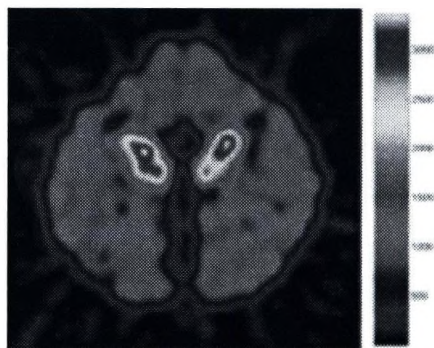
8. ábra. A Compton szóródás és korrekciója



9. ábra. Valódi és szimulált cylinder fantom vizsgálat



10. ábra. Szimulált perfúziós vizsgálat



11. ábra. Szimulált receptor jelölés az agy specifikus területében (putamen)

