

Affin Puzzle: Deformált objektumdarabok helyreállítása megfeleltetések nélkül*

Domokos Csaba¹, Kató Zoltán²

¹ `dcs@inf.u-szeged.hu`

² Képfeldolgozás és Számítógépes Grafika Tanszék, Szegedi Tudományegyetem
`kato@inf.u-szeged.hu`

Kivonat Ebben a cikkben törött objektumok helyreállításának problémájával foglalkozunk. Az eredeti objektum és annak törött darabjai között lineáris transzformációkat feltételezünk, és a javasolt módszer 2D és 3D affin transzformációkon keresztül mutatjuk be. A módszerünk alapötlete, hogy egy polinom egyenletrendszert írunk fel, melynek megoldása megadja az illesztés ismeretlen paramétereit. A javasolt algoritmust nagyméretű, két- és háromdimenziós képeket tartalmazó szintetikus adatbázison teszteltük. Az eredmények mutatják, hogy a módszer jó eredményeket szolgáltat, illetve robusztus a szegmentálási hibákkal szemben. Bemutatunk továbbá néhány 2D valós képen kapott eredményt, illetve olyan eredményeket, melyeket műtéti tervezés során használt 3D orvosi képek esetén kaptunk.

1. Bevezető

Ebben a cikkben törött objektumok helyreállításának problémájával foglalkozunk. Megjegyezzük, hogy a cikkben szereplő eredményeket [1]-ben már korábban publikáltuk. A feladat *puzzle* néven is ismert, mely nemcsak elméleti szempontból érdekes [2,3], hanem számos alkalmazási területe van, mint például az archeológia [4] vagy az orvosi képfeldolgozás [5] (pl. törött csontok helyreállítása [6,7,8]). A feladat a következőképpen fogalmazható meg: Adott egy bináris kép (*sablon*) és egy másik bináris kép (*megfigyelés*), mely a *sablon* képen lévő objektum törött darabjait tartalmazza. Célunk annak a geometriai transzformációnak a meghatározása, mely a törött darabokat a *sablon* képen lévő eredeti helyükre illeszti. A globális deformáció egy nemlineáris transzformáció a következő megszorításokkal: 1) az objektum részei egymástól el vannak szeparálva (például szegmentálás által vagy diszjunktak), 2) a *sablon* kép összes törött darabja ismert, de 3) mindegyikre *különböző* affin transzformáció hat.

Alakzatok parciális illesztése [9,10,11] hasonló probléma. Ekkor két (általában) különböző alakzat esetén azt a transzformációt keressük, melyet az egyik alakzatra alkalmazva a két alakzat közös része fedésbe kerül. Ez a klasszikus

* A cikk eredményei az alábbi publikációban jelentek meg: Csaba Domokos and Zoltan Kato: Affine puzzle: Realigning deformed object fragments without correspondences. In Proceedings of European Conference on Computer Vision (2010) 777–790.

alakzatillesztés egy különösen nehéz részterülete. Ebben az értelemben, a feladatunk tekinthető úgy is, mint a *sablon* és a *megfigyelés* darabjainak parciális illesztése. A jelenlegi megközelítések főként a Laplace-Beltrami keretrendszeren [12,11] alapulnak, de a klasszikus megközelítések, mint például az Iterative Closest Point (ICP) algoritmus [13] is használható alkalmas alakzat reprezentációt feltételezve [9]. Ezen algoritmusok meglehetősen nagy számításigénye miatt, ez a fajta megoldás nem optimális a problémánk esetén.

Egy másik kapcsolódó probléma nemlineáris torzulások lineáris transzformációkkal való szakaszonkénti közelítése. [14]-ben a torzulást lokálisan affin transzformációkkal modellezzük, mely egy globálisan sima deformációt eredményez, és az intenzitások lokális és globális változását is figyelembe veszi. A feladat klasszikus megoldása [15] során a képek között pontmegfeleltetéseket detektálnak, majd ezután egy (általában eléggé műveletigényes) optimalizálási eljárás segítségével vagy egy egyenletrendszer megoldásával kapják a deformáció ismeretlen paramétereit. Viszont a képek között megbízható pontmegfeleltetéseket találni már önmagában bonyolult feladat.

A puzzle feladat legtöbb megoldása során [2,3,4] szomszédos darabokat próbálnak illeszteni, majd ezen párosításokat felhasználva az eredeti objektumot merevtest transzformációkkal állítják elő. Kong és Kimia [2] egy olyan 2D görbeillesztésen alapuló technikát javasolt, mely a puzzle darabjainak geometriai tulajdonságát használja fel. A megoldást úgy kapják, hogy a "legjobbat először" keresési stratégia alapján az objektum darabjait háromszögekre bontva, azokat rekurzív módon csoportosítják. Ez a módszer kiterjeszhető például lézer szkennelrel letapogatott 3D darabokra is. Ekkor a 2D görbeillesztésen alapuló módszer általánosításával kapott eljárás segítségével állíthatók vissza az egyes darabok az eredeti helyükre. Papaioannou és *tsai*. 3D objektumok rekonstrukciójának problémáját tekintik, ahol az egyes darabok felszínének geometriáját használják ki, továbbá feltételezik, hogy nem áll rendelkezésre semmilyen információ a végső rekonstruálandó modellel kapcsolatban [3]. A módszer alapötlete, hogy két háromdimenziós darab esetén annál jobb illesztést kapunk, minél inkább a megfelelő helyükön vannak, így azon pontok közötti távolságot kell minimalizálni, melyek az adott darabok látható oldalain vannak. Az illesztett részek ezután merevtest transzformációval kerülnek az eredeti helyükre. A [4]-ben megadott görbeillesztés meglehetősen nagy számításigényű, mely a multiscale technika alkalmazásával egyszerűsíthető.

A klasszikus megközelítések során sokszor felhasználják a *sablon* objektumot, javítva ezzel a helyreállítás pontosságát és hatékonyságát. Viszont sokszor olyan problémákat céloznak meg, ahol a *sablon* nem elérhető, például archeológiai alkalmazások esetén [4]. Számos olyan alkalmazás létezik, ahol a *sablon* objektum ismert: Ipari alkalmazások esetén a feldolgozott darabok 3D modelljei könnyedén megkaphatók. Az orvosi képfeldolgozásban általában használható egy *atlasz*, vagy kihasználható az emberi test szimmetriája. Például törött csontok helyreállítása során az egészséges csont szolgálhat *sablonként*, amint azt a 5.3. fejezetben látni fogjuk. Emiatt a puzzle probléma ezen utóbbi változatát vizsgáljuk ebben a cikkben, és egy általános megoldást javaslunk, ahol 2D és 3D transz-

formációkat alkalmazunk. Az itt alkalmazott módszer nagyon hasonlít néhány affin regisztrációval kapcsolatos módszerhez [16,17]. Mindazonáltal ezen módszerek egyike sem konkrétan a puzzle problémával foglalkozik. [17]-ben azt feltételezik, hogy a képeken színinformáció is elérhető. Ezen intenzitás információk alapján, Hagege és Francos egy lineáris egyenletrendszert ír fel, melynek megoldása közvetlenül az ismeretlen transzformáció paramétereit eredményezi. [16]-ban egy elegáns megoldást mutattunk be, mellyel síkbeli alakzatok affin torzulásait lehet meghatározni. Viszont ez a módszer sem képes megoldani a puzzle problémát, mivel a deformáció nemlineáris, továbbá nem ismert a *sablon* objektum és annak *megfigyelt* darabjai közti megfeleltetés.

A 2. fejezetben egy általános keretrendszert adunk meg a probléma megoldására, melyet a 3. fejezetben a numerikus implemmentációs részletek követnek, majd a 4. fejezetben a módszer alkalmazásainak bemutatására kerül sor különböző lineáris transzformációk esetén. Végül a 5. fejezet a teszteredményeket mutatja be, melyeket 2D-s és 3D képeket tartalmazó szintetikus generált adatbázisokon és több különböző valós képen kaptunk. A 6. fejezet foglalja össze a cikket.

2. Törött objektumok helyreállítása

Adott egy n -dimenziós objektum a *sablon* képen és egy *megfigyelés* kép, mely a *sablon* deformált darabjait tartalmazza. Azt a transzformációt szeretnénk meghatározni, mely ezeket a darabokat a *sablon* képen lévő eredeti helyükre illeszti. Jelölje a *sablon* és a *megfigyelés* kép pontjainak homogén koordinátáit $\mathbf{x} = [x_1, \dots, x_n, 1]$ és $\mathbf{y} = [y_1, \dots, y_n, 1] \in \mathbb{P}^n$. Továbbá, legyen $\ell \in \mathbb{N}$ a *megfigyelés*en lévő törött darabok száma. A *megfigyelés* kép *sablon* képhez való illesztését meghatározó nemlineáris transzformáció ℓ darab lineáris (affin) transzformáció

$$\mathbf{A}_i = \begin{bmatrix} a_{i11} & a_{i12} & \dots & a_{i1(n+1)} \\ \vdots & & \ddots & \vdots \\ a_{in1} & a_{in2} & \dots & a_{in(n+1)} \\ 0 & 0 & \dots & 1 \end{bmatrix}, \quad i = 1, \dots, \ell \quad (1)$$

kompozíciójaként áll elő. Mivel a *megfigyelés* diszjunkt részeket tartalmaz, feltehetjük, hogy a $\lambda' : \mathbb{P}^n \rightarrow \{0, 1, \dots, \ell\}$ a deformált alakzatok címkézését adja, ahol a 0-t a háttérhez rendeljük. Nyilvánvalóan létezik a *sablon* képhez is egy *rejtett* $\lambda : \mathbb{P}^n \rightarrow \{0, 1, \dots, \ell\}$ címkézés, ahol az i címke a *sablon* kép i . darabjához tartozik. Célunk az affin transzformációkat megadó $\{\mathbf{A}_i\}_{i=1}^{\ell}$ mátrixok meghatározása. A legnagyobb kihívást az jelenti, hogy sem a *sablon* kép partícionálása (azaz a λ címkézés rejtett) sem pedig az alakzatok közötti megfeleltetés nem ismert.

2.1. Megoldás egy alakzat pár esetén

Először az i . alakzat pár közötti kapcsolatot adjuk meg. Jelölje a *sablon* és *megfigyelés* képen lévő alakzatok által meghatározott tartományokat $\mathcal{D}_i = \{\mathbf{x} \in$

$\mathbb{P}^n | \lambda(\mathbf{x}) = i \}$ és $\mathcal{D}'_i = \{\mathbf{y} \in \mathbb{P}^n | \lambda'(\mathbf{y}) = i\}$. Megjegyezzük, hogy \mathcal{D}'_i ismert, míg \mathcal{D}_i nem. Ezen tartományok pontjai az ismeretlen \mathbf{A}_i transzformációk alapján felelnek meg egymásnak:

$$\mathbf{x} = \mathbf{A}_i \mathbf{y} . \quad (2)$$

Egy lehetőség az \mathbf{A}_i meghatározására, hogy pontmegfeleltetéseket határozunk meg, majd a (2) egyenletben megadott egyenletrendszert oldjuk meg. Mivel \mathcal{D}_i ismeretlen, emiatt a megfeleltetések megtalálása szinte lehetetlen. Ezért egy olyan direkt módszert keresünk, mely nem igényel megfeleltetéseket. Vegyük észre, hogy a (2)-ben megadott összefüggés akkor is érvényben marad, ha egy $\omega : \mathbb{P}^n \rightarrow \mathbb{R}$ függvény hat az egyenlet mindkét oldalán [16]:

$$\omega(\mathbf{x}) = \omega(\mathbf{A}_i \mathbf{y}) . \quad (3)$$

Ebből integrálással kapjuk, hogy [16]

$$\int_{\mathcal{D}_i} \omega(\mathbf{x}) d\mathbf{x} = |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega(\mathbf{A}_i \mathbf{y}) d\mathbf{y} , \quad (4)$$

ahol $\mathbf{x} = \mathbf{A}_i \mathbf{y}$, $d\mathbf{x} = |\mathbf{A}_i| d\mathbf{y}$ integrál transzformációt alkalmaztunk, illetve $|\mathbf{A}_i|$ a Jacobi-féle determináns. A (4) egyenlet alapján nemlineáris $\{\omega_j\}_{j=1}^m$ (ahol $m \geq n(n+1)$) függvények alkalmazásával annyi egyenletet kaphatunk, amennyire szükség van. Az így kapott nemlineáris egyenletrendszer megoldása adja \mathbf{A}_i paramétereit [16].

2.2. Megoldás az összes alakzatra egyszerre

Most már ismerjük az i . alakzat pár közötti összefüggéseket, de sem az alakzatok közötti megfeleltetés, sem pedig a *sablon* kép \mathcal{D}_i partícionálása nem ismert. Ha ezeket ismernénk, akkor az alakzatokra páronként egy hagyományos bináris regisztrációs algoritmust alkalmazva megkaphatnánk az illesztés paramétereit. Sajnos ez a parciális illesztés [9] problémájának megoldását igényelné a *megfigyelésen* lévő összes alakzat és a *sablon* kép között, mely nem triviális feladat. A megoldás, hogy összeadjuk az összes \mathcal{D}_i alakzathoz tartozó egyenletet és egy egyenletrendszerben egyszerre, az összes paraméter egyidejű becslésével oldjuk meg a feladatot. Így (4)-ből kapjuk, hogy

$$\sum_{i=1}^{\ell} \int_{\mathcal{D}_i} \omega_j(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{\ell} |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega_j(\mathbf{A}_i \mathbf{y}) d\mathbf{y} . \quad (5)$$

Legyen $\mathcal{D} := \cup_{i=1}^{\ell} \mathcal{D}_i$, ahol $\mathcal{D} = \{\mathbf{x} \in \mathbb{P}^n | \lambda(\mathbf{x}) \neq 0\}$ a teljes *sablon* képen lévő objektumhoz tartozó tartományt jelöli. Ekkor az előző egyenlet bal oldala a következőképpen írható

$$\sum_{i=1}^{\ell} \int_{\mathcal{D}_i} \omega_j(\mathbf{x}) d\mathbf{x} = \int_{\cup_{i=1}^{\ell} \mathcal{D}_i} \omega_j(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{D}} \omega_j(\mathbf{x}) d\mathbf{x} , \quad (6)$$

mely közvetlenül számolható az input képek alapján anélkül, hogy a \mathcal{D}_i partícionálásokat ismernénk. Az így kapott egyenletrendszer $\ell n(n+1)$ ismeretlent tartalmaz:

$$\int_{\mathcal{D}} \omega_j(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{\ell} |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega_j(\mathbf{A}_i \mathbf{y}) d\mathbf{y}, \quad j = 1, \dots, m. \quad (7)$$

A (7) egyenletrendszer megoldásával megkapjuk a teljes deformáció minden paraméterét. Mivel minden ω_j egy egyenletet eredményez, így legalább $m \geq \ell n(n+1)$ lineárisan független függvényre van szükség az ℓ lineáris transzformáció megtalálásához. A gyakorlatban $m > \ell n(n+1)$, mely túlhatározott egyenletrendszerhez vezet, melyet legkisebb négyzetes értelemben oldunk meg.

3. Numerikus implementáció

Elméletileg bármilyen nemlineáris függvény használható a (7) egyenletrendszer megkonstruálásához, mely kielégíti a (3) egyenletet. A gyakorlatban az egyenletrendszer megoldását olyan minimalizálási eljárással kapjuk, mely iteratívan, legkisebb-négyzetes értelemben oldja meg a feladatot. Ilyen algoritmus például a *Levenberg-Marquardt* algoritmus [18], mely jól megválasztott numerikus szerkezetet igényel.

3.1. Normalizálás

Először mindkét kép koordinátáit a $[-0.5, 0.5]^n$ hiperkockába normalizáljuk, azaz $\cup_{i=1}^{\ell} \mathcal{D}'_i \mapsto [-0.5, 0.5]^n$ és $\mathcal{D} \mapsto [-0.5, 0.5]^n$. A normalizálás során eltoljuk az origót az alakzatok tömegközéppontjába, melyet egy közös, a *sablon* és *megfigyelés* maximális mérete alapján számolt skálafaktorral végrehajtott izotropikus skálázás követ. Természetesen az egyenletrendszer megoldásával kapott értékeket vissza kell normalizálni, hogy a valódi, eredeti alakzatok közötti transzformációt megkapjuk. Jelölje a *sablon* és *megfigyelés* képhez tartozó normalizáló transzformációkat \mathbf{N}_t és \mathbf{N}_o , továbbá a kapott megoldást pedig $\tilde{\mathbf{A}}_i$. Ekkor a valódi transzformáció $\hat{\mathbf{A}}_i = \mathbf{N}_t^{-1} \tilde{\mathbf{A}}_i \mathbf{N}_o$ alakban áll elő minden $i = 1, \dots, \ell$ esetén.

Mivel a legkisebb-négyzetes értelemben vett megoldással a (7) egyenletrendszer algebrai hibáját minimalizáljuk, azt szeretnénk, ha az összes egyenlet egyenlő mértékben járulna hozzá az egyenletrendszer algebrai hibájához. Ez az ω_j értékészletének $[-1, 1]$ intervallumba való normalizálásával megoldható. Tapasztalati úton azt kaptuk, hogy a legkisebb-négyzetes értelemben vett minimalizálás esetén az egyes iterációk során alkalmazott transzformációk az alakzatokat az origó középpontú $\sqrt{n}/2$ sugarú körön belül hagyják (az egység oldalú hiperkocka köré írható hipergömb). Tehát az ω_j normalizálása megvalósítható a (7) egyenletrendszerben szereplő integrálok megfelelő c_j konstanssal való leosztásával, melyet az integrálok előbbi tartományok (hipergömbök) feletti értéke alapján kapunk:

$$c_j = \int_{\|\mathbf{x}\| \leq \frac{\sqrt{n}}{2}} |\omega_j(\mathbf{x})| d\mathbf{x}, \quad j = 1, \dots, m. \quad (8)$$

3.2. Algoritmus és időbonyolultság

A gyakorlatban csak véges pontosságú digitális képek állnak rendelkezésünkre, így az integrálok csak *közelíthetően* az előtérpíxelek felett vett véges összegekkel adhatók meg. Ez a közelítés elhanyagolható hibát eredményez a számítások során. A \mathcal{D} és \mathcal{D}'_i tartományokat az előtérpíxelek D és D'_i véges halmazával reprezentáljuk. A normalizált (7) egyenletrendszer diszkrét változata az alábbi formában írható

$$\frac{1}{c_j} \sum_{\mathbf{x} \in D} \omega_j(\mathbf{N}_i \mathbf{x}) = \frac{1}{c_j} \sum_{i=1}^{\ell} |\mathbf{A}_i| \sum_{\mathbf{y} \in D'_i} \omega_j(\mathbf{N}_o \mathbf{A}_i \mathbf{y}), \quad j = 1, \dots, m. \quad (9)$$

A (9) egyenletrendszert *Levenberg-Marquardt* algoritmussal [18] oldjuk meg, mely iteratív minimalizáláson alapul és minden iterációban az összes egyenletet kiértékeli. Emiatt az algoritmus időbonyolultsága jelentősen csökkenthető lenne, ha a véges összegeket előre ki lehetne számolni elkerülve ezzel azt, hogy minden iterációban végig kelljen pásztázni a képet. Elméletileg tetszőleges olyan ω függvényhalmaz használható, melyek lineárisan független egyenleteket generálnak. [16]-ban megmutattuk, hogy polinomok választásával polinom egyenletrendszerhez jutunk, ahol a véges összegek előre kiszámolható konstansokká válnak. Figyelembe véve ezt a megállapítást polinomok következő halmazát fogjuk felhasználni

$$\{\omega_j : \mathbb{P}^n \rightarrow \mathbb{R}\}_{j=1}^m = \{\mathbf{x} \mapsto x_1^{u_1} \dots x_n^{u_n} \mid u_k \in \mathbb{N}, k = 1, \dots, n, 0 \leq \sum_{k=1}^n u_k \leq d\}, \quad (10)$$

ahol d a maximális fokszám és a polinomok számát $m = \frac{1}{n!} \prod_{i=1}^n (d + i)$ adja.

Az algoritmus pszeudokódját a (1) algoritmus mutatja. Mivel polinomiális függvényeket alkalmazunk a (9) egyenletrendszer generálása során, az ismeretlenek nem szerepelnek a véges összegekben belül [16]. Emiatt az algoritmus lineáris időbonyolultságú: a (9) egyenletrendszer megkonstruálásának időbonyolultsága $\mathcal{O}(|D| + \sum_{i=1}^{\ell} |D'_i|)$, illetve magának a megoldónak pedig elvileg konstans, mivel független az input képek méretétől.

4. Affin transzformációk

A javasolt regisztrációs keretrendszert a legfontosabb lineáris transzformációkra alkalmazzuk: két- és háromdimenziós affin és merevtest transzformációkra. A 2-dimenziós affin transzformációkat gyakran alkalmazzák projektív torzulások lineáris közelítéseként. A 3D merevtest transzformációk számos orvosi alkalmazás során játszanak fontos szerepet. Például, mikor CT képeken lévő csontokat kell illeszteni, hiszen a csontok biomechanikus tulajdonságai miatt csak merevtest transzformáció jöhet szóba.

Algorithm 1: A javasolt algoritmus pszeudokódja.

Input : A bináris *sablon* (D) és ℓ *megfigyelt* alakzat ($D'_i, i = 1, \dots, \ell$)

Output: ℓ darab illesztett $\hat{\mathbf{A}}_i$ lineáris transzformáció

- 1 Normalizáljuk az input koordinátákat az $[-0.5, 0.5]^n$ -be egy megfelelő \mathbf{N} hasonlósági transzformáció segítségével, úgy hogy az origó a tömegközéppontba kerül.
 - 2 Választunk egy $\omega_j : \mathbb{P}^n \rightarrow \mathbb{R}$ ($j = 1, \dots, m \geq \ell n(n+1)$) polinom függvényhalmazt.
 - 3 Megkonstruáljuk a (túlhatározott) (9) egyenletrendszert.
 - 4 Megkeressük a (9) egyenletrendszer legkisebb-négyzetes értelemben vett megoldását a *Levenberg-Marquardt* algoritmus felhasználásával. A megoldót az identikus transzformációval inicializáljuk.
 - 5 A kapott $\tilde{\mathbf{A}}_i$ megoldás vissznormalizálása $\hat{\mathbf{A}}_i = \mathbf{N}_i^{-1} \tilde{\mathbf{A}}_i \mathbf{N}_i$ alapján adja az illesztés paramétereit.
-

4.1. 2D affin transzformációk

Egy 2D affin transzformációnak 6 paramétere van, tehát $n = 2$ és 6ℓ ismeretlenünk van. Ahhoz, hogy elegendő egyenletet kapjunk, a (10)-ben leírt ω -k halmazát alkalmazzuk. Ekkor d -t úgy kell megválasztani, hogy

$$m = \frac{(d+1)(d+2)}{2} \geq 6\ell \quad \Rightarrow \quad d \geq \left\lceil \frac{\sqrt{1+48\ell}-3}{2} \right\rceil, \quad (11)$$

ahol $\lceil \cdot \rceil$ az felső egészrész jelöli. A (7) egyenlet bármely $j = 1, \dots, m$ -re úgy írható, hogy

$$\int_{\mathcal{D}} x_1^{u_1} x_2^{u_2} d\mathbf{x} = \sum_{i=1}^{\ell} \int_{\mathcal{D}'_i} |\mathbf{A}_i| (a_{i11}y_1 + a_{i12}y_2 + a_{i13})^{u_1} (a_{i21}y_1 + a_{i22}y_2 + a_{i23})^{u_2} d\mathbf{y}, \quad (12)$$

ahol a Jacobi-féle determináns könnyedén kiszámolható $|\mathbf{A}_i| = a_{i11}a_{i22} - a_{i12}a_{i21}$ alapján.

4.2. 3D affin transzformációk

A 2-dimenziós eset 3-dimenzióra való kiterjesztése eléggé nyilvánvaló. Ekkor a *sablon* objektum darabjai különböző 3D affin transzformációk alapján torzulnak, és 12ℓ ismeretlenünk van. d -t most úgy kell megválasztani, hogy

$$m = \frac{(d+1)(d+2)(d+3)}{6} \geq 12\ell \quad \Rightarrow \quad d \geq \left\lceil \frac{c}{3} + \frac{1}{c} - 2 \right\rceil, \quad \text{ahol} \quad (13)$$

$$c = \sqrt[3]{3 \left(324\ell + \sqrt{(324\ell)^2 - 3} \right)}.$$

A Jacobi-féle determináns hasonlóan számolható, mint a 2D-s esetben.

4.3. 3D merevtest transzformációk

A 3D merevtest transzformációk a 3D lineáris transzformációk fontos esetei. Egy 3D merevtest transzformáció hat szabadsági fokkal rendelkezik: $\alpha_1, \alpha_2, \alpha_3$ a három tengely körül vett forgatások szögei, illetve t_1, t_2, t_3 az egyes tengelyek mentén vett eltolások. Hasonló $\{\omega\}_{j=1}^m$ halmazt alkalmazunk, mint (13)-ban, csak most kevesebb polinomra van szükség:

$$d \geq \left\lceil \frac{c}{3} + \frac{1}{c} - 2 \right\rceil, \quad \text{ahol} \quad c = \sqrt[3]{3 \left(27 + 162\ell + \sqrt{(27 + 162\ell)^2 - 3} \right)}. \quad (14)$$

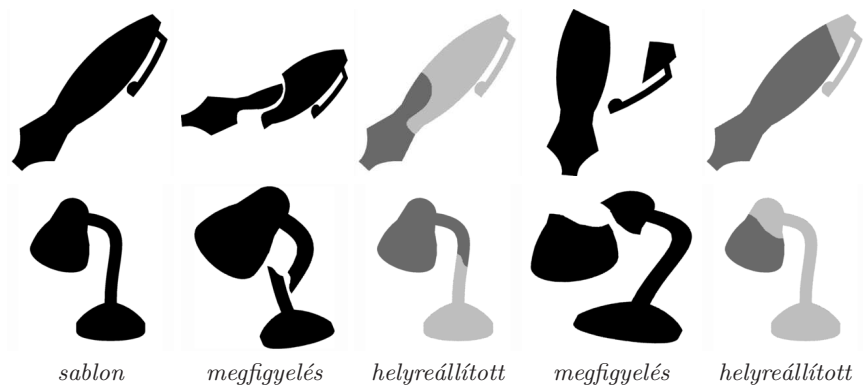
Mivel a merevtest transzformációk nem változtatják meg az objektumok méretét, a Jacobi-féle determináns értéke 1 lesz, emiatt nem szerepel az egyenletekben.

5. Eredmények

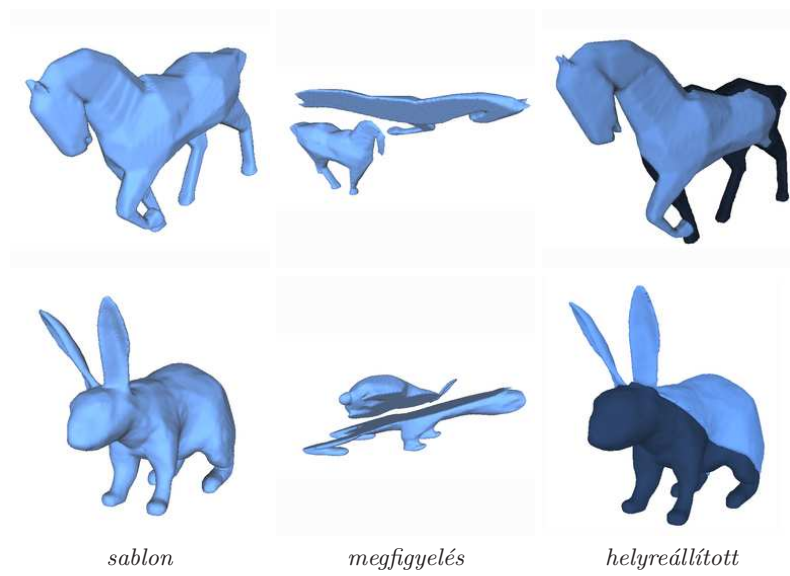
A javasolt algoritmust 2D és 3D képeket tartalmazó, szintetikus generált adatbázison teszteltük. A 2-dimenziós esetben az adatbázis 10 *sablon* objektumot tartalmazott. A *megfigyeléseket* úgy generáltuk, hogy minden objektumot négyféleképpen két részre vágunk, így minden *sablon* képhez négy képet kaptunk. Majd 600 darab átlagosan 700×700 -as méretű *megfigyelést* állítottunk elő. Ezeket az alábbi affin transzformációk véletlen kompozíciójaként adódó transzformáció alkalmazásával kaptuk, ahol a paramétereket a megadott intervallumokból véletlenszerűen választottuk: $[-\pi/4; \pi/4]$ mértékű forgatás, a két tengely mentén $[0, 75; 1, 25]$ -ből vett skálázás, $[-0, 1; 0, 1]$ nyírás és $[-25; 25]$ eltolás. A 3D esetben 10 *sablon* képet egy-egy véletlenszerűen megválasztott síkkal vágunk két részre úgy, hogy a kisebbik rész legalább az eredeti objektum 20%-a legyen. A képek öt különböző szétvágásával 50 képet kaptunk. Ezután véletlenszerűen megválasztott 3D affin transzformációk alkalmazásával összesen 200 darab, átlagosan $250 \times 250 \times 250$ -es méretű 3D *megfigyelést* generáltunk. Az alkalmazott transzformációk paraméterei a 2D esetenél leírt paraméterekhez hasonlóan lettek megválasztva (az egyetlen különbség, hogy az eltolás értékeit a $[-10; 10]$ intervallumból választottuk). Az eredmények kiértékelésére két hibamértéket definiáltunk: Az első, melyet ϵ jelöl, az átlagos távolságot méri a valódi \mathbf{A}_i és az illesztett $\hat{\mathbf{A}}_i$ transzformáció között. A másik a *sablon* és a *helyreállított* képek abszolút különbsége, melyet δ jelöl.

$$\epsilon = \sum_{\mathbf{p} \in D'_i, 1 \leq i \leq \ell} \frac{\|(\mathbf{A}_i - \hat{\mathbf{A}}_i)\mathbf{p}\|}{|D'_i|}, \quad \text{és} \quad \delta = \frac{|\hat{D} \Delta D|}{|\hat{D}| + |D|} \cdot 100\%, \quad (15)$$

ahol Δ két halmaz szimmetrikus különbségét, $D' = \cup_{i=1}^{\ell} D'_i$ és $\hat{D} = \cup_{i=1}^{\ell} \hat{D}_i$ pedig a *megfigyelés* és a *helyreállított* képek pixeleinek halmazát jelöli. Intuitíven ϵ az egy pixelre vetített átlagos transzformációs hibát mutatja. Megjegyezzük, hogy ϵ -t csak akkor tudjuk kiértékelni, ha a valódi transzformáció is ismert, ezzel szemben δ bármikor számolható. Viszont ϵ jobban kifejezi a transzformáció



1. ábra. Néhány 2-dimenziós képen kapott eredmény.



2. ábra. Néhány 3-dimenziós képen kapott eredmény.

hibáját. Tapasztalati úton azt találtuk, hogy a 2D esetben $\delta \leq 5\%$, a 3D esetben pedig $\delta \leq 10\%$ esetén állíthatjuk azt, hogy vizuálisan jó eredményt kaptunk.

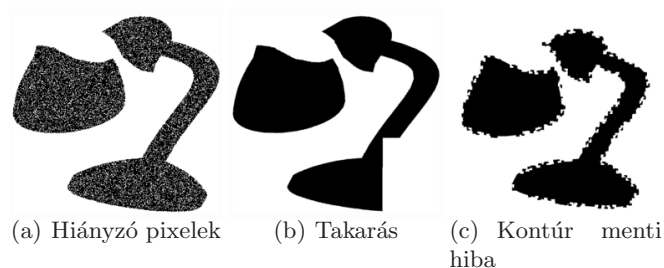
A javasolt algoritmus Matlabban lett implementálva. A tesztek Linux alatt 3GHz-es 3GB memóriával rendelkező gépeken futtattuk. A 2-dimenziós esetben a tipikus futási idő 3 mp 3D-ben pedig 10 mp volt. A 1. és 2. ábrán látható néhány futási eredmény. A 1. táblázatban láthatók a tesztesetek kiértékelése után kapott eredmények. Az eredmények világosan mutatják, hogy a javasolt módszer majdnem tökéletes eredményt szolgáltat mind a kétdimenziós mind a háromdimenziós esetben.

Torzulásmentes esetek	ϵ (pixel)				δ (%)			
2D affin transzformációk	0,11				0,13			
3D affin transzformációk	0,7				3,09			
(a) Hiányzó pixelek	1%	5%	10%	20%	1%	5%	10%	20%
2D affin transzformációk	6,57	21,1	32,83	56,26	2,09	6,24	8,39	12,62
3D affin transzformációk	1,22	4,65	9,71	19,02	3,99	8,67	15,8	23,54
(b) Takarás	1%	2,5%	5%	10%	1%	2,5%	5%	10%
2D affin transzformációk	9,91	20,45	35,04	58,68	3,54	6,35	9,51	13,75
3D affin transzformációk	3,27	7,7	14,73	22,74	8,07	13,08	18,47	26,13
(c) Kontúr menti hiba	1%	2,5%	5%	10%	1%	2,5%	5%	10%
2D affin transzformációk	1,9	3,91	6,65	12,23	0,59	1	1,73	3,08
3D affin transzformációk	0,99	1,44	2,33	4,03	3,23	3,65	4,44	5,8

1. táblázat. A javasolt algoritmus tesztelése során kapott hibamértékek mediánjai a 2D és 3D esetben. Az első két sorban a torzításmentes, szegmentálási hibát nem tartalmazó képeken kapott eredmények láthatók. A további sorokban a különböző típusú szegmentálási hiba esetén (lásd 3. ábra) kapott eredmények láthatók.

5.1. Robusztusság

A gyakorlatban általában nem áll rendelkezésre az alakzatok tökéletes szegmentálása. Most a módszer szegmentálási hibákra vett érzékenységet, azaz a robusztusságát elemezzük. Az első esetben a *megfigyelés* előtérpixeleinek 1%, ..., 20% töröljük. A második esetben az eredeti objektum 1%, ..., 10%-nak megfelelő négyzet alakú területet takarunk ki. Majd az alakzatok kontúrja mentén az eredeti objektum 1%, ..., 10%-nak megfelelő részt módosítunk úgy, hogy egyenletesen eloszlás alapján négyzet alakú területeket adunk az adott alakzatokhoz vagy törölünk belőlük. Néhány valós képen kapott eredményt is bemutatunk, melyek önmagukban tartalmaznak szegmentálási hibákat. Megjegyezzük, hogy az olyan hibák, mikor nem összefüggő területek törölődnek, morfológiai műveletekkel hatékonyan megszüntethetők. A 3. ábra mutat néhány példát a tekintett hibákra. A 1. táblázat mutatja, hogy a módszerünk elég ro-



3. ábra. Példák a vizsgált szegmentálási hibákra.

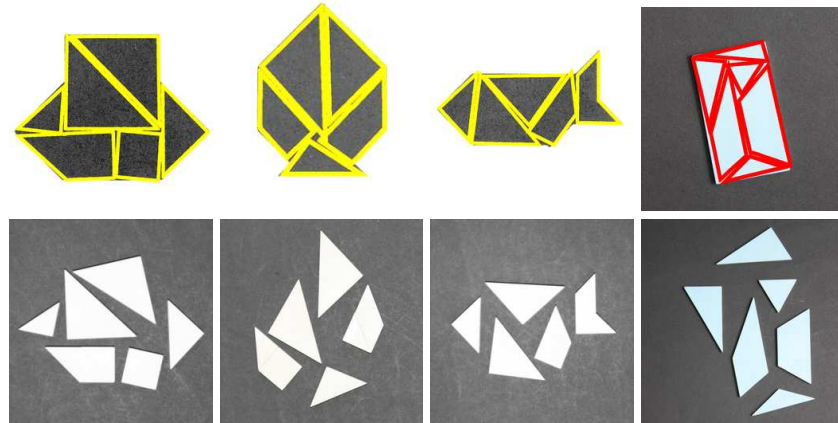
busztus, ha a hiba egyenletesen oszlik el az objektum felett (például hiányzó pixelek, kontúr menti hiba). Mindazonáltal, a módszer lokálisan jelentkező hiba esetén (pl. takarás) már kevésbé stabil. Ez jellemző tulajdonsága a terület alapú módszereknek, mivel ezen módszerek az objektumok felett számolt integrálokat használják, melyeket az ilyen típusú hibák nagyon elrontanak. Emiatt egy nagyobb hiányzó vagy takarásban lévő terület drasztikus hibákat eredményezhet az illesztés során. Mindazonáltal számos alkalmazás során a képeket kontrollált körülmények között rögzítik, ezzel garantálva, hogy a *megfigyeléseken* ne legyen takarás (például orvosi képfeldolgozás, ipari minőségellenőrzés).

5.2. A Tangram kirakó megoldása

A Tangram kirakó több síkidomot tartalmaz, melyeket alkalmas módon összerakva különböző alakzatokat kaphatunk. A cél egy előre megadott alakzat kirakása, melynek csak a körvonalát ismerjük. A 4. ábrán látható néhány példa ilyen alakzatokra, illetve a módszerünk által talált megoldásokra. A képek digitális fényképezőgéppel készültek, majd az ezekből küszöböléssel kapott alakzatokat használtuk fel *sablonként*. A 4. ábrán lévő első három esetében a Tangram játék megoldásfüzetében található megoldások szkennelt változatát használtuk, mint *sablon* képet, melyek csak a kirakandó alakzat körülbelüli megfelelői. Itt affin modellt használunk az alakzatok közötti síkhomográfia közelítésére.

Közismert tény, hogy a *Levenberg-Marquardt* algoritmus megragadhat a kezdeti érték közelében lévő lokális minimumokban. Jó inicializálást találni viszont alkalmazásfüggő feladat. Például ezeken a képeken (4.ábra) egy globális optimalizálási eljárás (pl. Spektrális gradiens módszer [19]) jó inicializálást adott, melyből indítva a *Levenberg-Marquardt* algoritmust jobb eredményt kapunk, mintha az identikus transzformációból indulnánk.

Végül megjegyezzük, hogy a 4. ábrán lévő képeken kismértékű átfedés látható. Ezt az okozza, hogy az átfedés láthatatlan az egyenletrendszer számára. Mindazonáltal, az átfedés problémája megoldható lenne úgy, hogy minden iteráció során ellenőrizzük, hogy az aktuális transzformáció milyen átfedést okoz, de ez nagyon időigényes művelet.

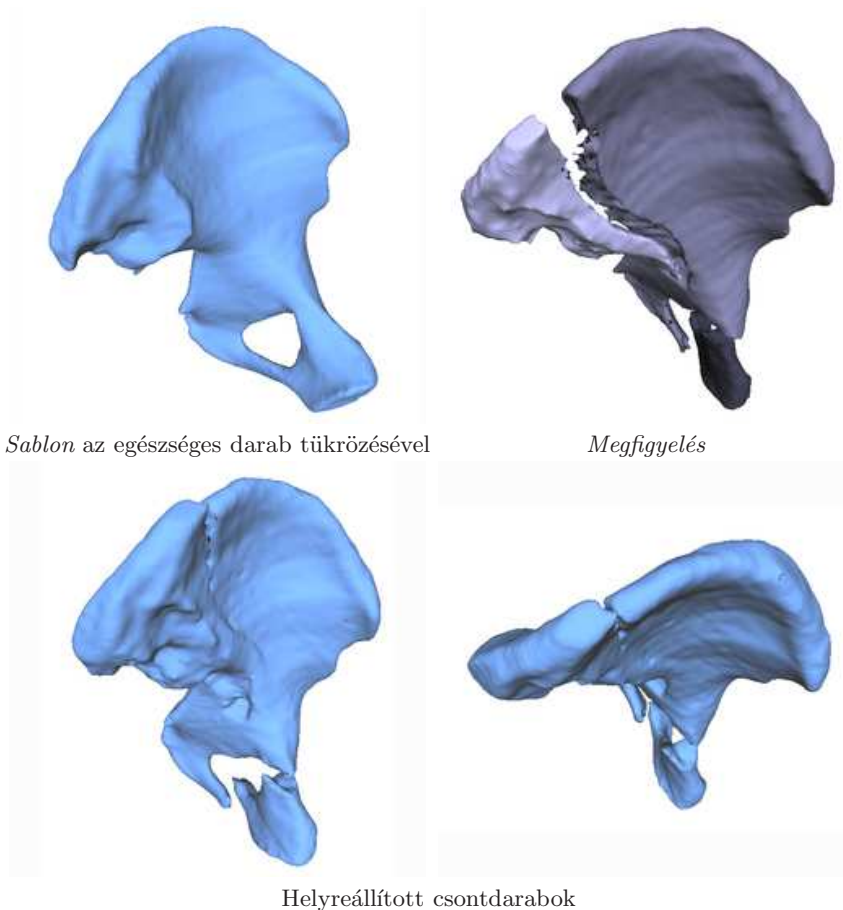


4. ábra. Tangram kirakós példák. **Fent:** *Sablon* képek és a törött darabok kontúrja a helyreállítás után. **Lent:** *Megfigyelések*.

5.3. Törött csontok helyreállítása

Bonyolultabb esetekben csontdarabok helyreállítása gyakran műtéti beavatkozást igényel, főképp ha az egyes darabok elfordulása vagy elmozdulása nagymértékű. Ilyen esetekben számítógéppel segített műtéti tervezést [6] alkalmaznak a konkrét műtéti beavatkozás előtt. Ezzel több információt nyernek a törött darabok elmozdulásáról és a beültetett implantátumok elhelyezkedését is egyszerű megvizsgálni. Az input adatok általában 3D CT képek, így az egyes részeket 3D térben kell mozgatni, mely drága 3D eszközöket, illetve sok manuális munkát igényel. Ezért a törött csontok automatikus helyreállítása jelentős időmegtakarítást eredményez, és egy olyan körülbelüli illesztést ad, melyet már könnyű manuálisan befejezni az adott anatómiai igényeknek megfelelően. A műtéti tervezés során a csontok biomechanikai tulajdonságát figyelembe véve, csak a merevtest transzformációk jöhetnek szóba. [6]-ban a klasszikus ICP algoritmust használják a csontok helyreállítására. Winkelbach és *tsai*. [7] jól ismert felszín regisztrációs technikákat alkalmaznak (pl. 2D mélység korreláció vagy ICP algoritmus) hengeres szerkezetű darabok helyreállítására. [8]-ban a regisztrációt úgy oldjuk meg, hogy egy kvadratikus szűrő fázis eltéréseinek lokális eltéréseit próbálják megbecsülni.

Most a javasolt keretrendszerünket 3D csípőcsontok helyreállítására alkalmazzuk, ahol 3D merevtest transzformációkat tételezünk fel az objektumok között. Azokban az esetekben, mikor csak az egyik oldalon van sérülés, a *sablon* egyszerűen az egészséges rész tükrözésével megkapható. A 5. ábra egy tipikus esetet mutat csípőcsonttörésre, ahol a csont három részre van törve. A legnagyobb kihívást a szegmentálási hibák, illetve a nem teljesen pontos *sablon* jelenti, mely az emberi test nem tökéletes szimmetriájából adódik. Ezen nehézségek ellenére a nagyobb részek illesztése elég pontos, csupán a kisebb darabnál van egy kicsi il-



5. ábra. Törött csontok helyreállítása (a CPU idő 15 mp volt ezen az 1 megavoxel CT képre).

lesztési hiba. Mivel a kis rész által okozott hiba viszonylag kicsi, ezért a megoldó nem találja meg a tökéletes transzformációt. Ha a (9) egyenletben a tagokat valahogy tudnánk normalizálni, akkor az algebrai hiba is jobban széteszlana az egyes egyenletek között, és pontosabb illesztést kaphatnánk. Sajnos ez kivitelezhetetlen, mert a megfelelő normalizáló konstansok kiszámításához ismerni kellene a *sablon* objektum partícionálását. Egy valós műtéti tervező rendszer nem működik emberi beavatkozás nélkül, emiatt a kisebb hibák nem annyira kritikusak és könnyen korrigálhatók.

6. Konklúzió

Egy újfajta keretrendszert mutattunk be a puzzle probléma megoldására, melyet 2D és 3D affin transzformációkra alkalmaztunk. A klasszikus megközelítések, melyek jellemző pontok kinyerésén és pontmegfeleltetéseken alapulnak, olyan megoldást javasolnak, ahol további információkat nem használnak fel. Alapjában a módszerünk egy polinom egyenletrendszer megkonstruálásán alapul, melynek megoldása közvetlenül megadja az ismeretlen paramétereket. Nyilvánvalóan, az objektumok száma és a deformáció erőssége befolyásolhatja az illesztés minőségét: Minél több törött részünk van, annál több egyenletre van szükség, mely befolyásolja a numerikus stabilitást. Továbbá, több törött rész több affin transzformációt eredményez, mely erősebb deformációhoz vezet. A törött részek teljesen véletlen elhelyezkedése egy bonyolult deformációnak felel meg, amikor a jó megoldást nehéz meghatározni. Másrészt, mikor a darabok viszonylagosan sorrendben vannak, sokkal pontosabb megoldást kapunk. Megjegyezzük, hogy a bemutatott orvosi alkalmazás során ez egy reális feltételezés a fizikai megszorítások miatt. A nagy számú teszteset, melyet 2D és 3D képeket tartalmazó szintetikus adatbázison futtattunk mutatják a módszer hatékonyságát és robusztusságát, illetve a valós képeken kapott eredmények megerősítik a módszer fontosságát több, különböző alkalmazási terület esetén.

7. Köszönetnyilvánítás

A cikk megjelenését részben a K75637 OTKA és a CNK80370 NKTH projekt támogatásával valósult meg. A CT képeket a Szegedi Tudományegyetem, Traumatológiai Osztálya bocsátotta rendelkezésünkre Prof. Varga Endre hozzájárulásával.

Hivatkozások

1. Domokos, C., Kato, Z.: Affine puzzle: Realigning deformed object fragments without correspondences. In Daniilidis, K., Maragos, P., Paragios, N., eds.: Proceedings of the European Conference on Computer Vision. Volume 6312 of Lecture Notes in Computer Science., Heraklion, Crete, Greece, Springer (2010) 777–790
2. Kong, W., Kimia, B.B.: On solving 2D and 3D puzzles using curve matching. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. Volume 2., Kauai, Hawaii, IEEE (2001) 1–8
3. Papaioannou, G., Karabassi, E.A., Theoharis, T.: Reconstruction of three-dimensional objects through matching of their parts. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 114–124
4. McBride, J.C., Kimia, B.B.: Archaeological fragment reconstruction using curve-matching. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop, Madison, Wisconsin, USA, IEEE (2003) 1–8
5. Hill, D.L.G., Batchelor, P.G., Holden, M., Hawkes, D.J.: Medical image registration. Physics in Medicine and Biology **46** (2001) R1–R45

6. Erdőhelyi, B., Varga, E.: Semi-automatic bone fracture reduction in surgical planning. In: Proceedings of the International Conference on Computer Assisted Radiology and Surgery. Volume 4 of International Journal of Computer Assisted Radiology and Surgery., Berlin, Germany, Springer (2009) S98–S99
7. Winkelbach, S., Westphal, R., Goesling, T.: Pose estimation of cylindrical fragments for semi-automatic bone fracture reduction. In Michaelis, B., Krell, G., eds.: Proceedings of Annual Symposium of the German Association for Pattern Recognition. Volume 2781 of Lecture Notes in Computer Science., Magdeburg, Germany, Springer (2003) 556–573
8. Pettersson, J., Knutsson, H., Borga, M.: Non-rigid registration for automatic fracture segmentation. In: Proceedings of International Conference on Image Processing, Atlanta, GA, USA, IEEE (2006) 1185–1188
9. Bronstein, A.M., Bronstein, M.M.: Regularized partial matching of rigid shapes. In Forsyth, D., Torr, P., Zisserman, A., eds.: Proceedings of the European Conference on Computer Vision. Volume 5303 of Lecture Notes in Computer Science., Marseille, France, Springer (2008) 143–154
10. Funkhouser, T., Shilane, P.: Partial matching of 3D shapes with priority-driven search. In: Proceedings of the Eurographics Symposium on Geometry Processing, Sardinia, Italy, Eurographics, ACM SIGGRAPH (2006) 1–12
11. Reuter, M.: Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions. *International Journal of Computer Vision* **89** (2010) 287–308
12. Rustamov, R.M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In: Proceedings of the Eurographics Symposium on Geometry Processing, Barcelona, Spain, Eurographics, ACM SIGGRAPH (2007) 1–9
13. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14** (1992) 239–256
14. Periaswamy, S., Farid, H.: Medical image registration with partial data. *Medical Image Analysis* **10** (2006) 452–464
15. Feldmar, J., Ayache, N.: Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision* **18** (1996) 99–119
16. Domokos, C., Kato, Z.: Parametric estimation of affine deformations of planar shapes. *Pattern Recognition* **43** (2010) 569–578
17. Hagege, R.R., Francos, J.M.: Estimation of affine geometric transformations of several objects from global measurements. In: Proceedings of IEEE International Workshop on Multimedia Signal Processing, Rio de Janeiro, Brazil, IEEE (2009) 1–5
18. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics* **11** (1963) 431–441
19. Birgin, E.G., Martínez, J.M., Raydan, M.: SPG: Software for convex-constrained optimization. *ACM Transactions on Mathematical Software* **27** (2001) 340–349