

TDK-dolgozat

Sánta Zsolt

Szegedi Tudományegyetem
Természettudományi és Informatikai Kar
Képfeldolgozás és Számítógépes Grafika Tanszék

**Háromdimenziós objektumok
nemlineáris regisztrációja**

TDK-dolgozat

Készítette:

Sánta Zsolt

programtervező informatikus MSc
szakos hallgató

Témavezető:

Dr. Kató Zoltán

tanszékvezető
egyetemi docens

Tartalomjegyzék

Bevezetés	4
1. Regisztrációs keretrendszer	7
1.1. Matematikai áttekintés	7
1.2. A keretrendszer	8
1.3. Numerikus implementáció	10
1.3.1. Normalizálás	11
1.3.2. Diffeomorf eredmény	12
1.4. A módszer tesztelése	13
2. Polinomiális deformációs modell	15
2.1. A koordináták és a paraméterek szétválasztása	17
2.1.1. Esettanulmány	19
2.2. Teszteredmények	19
2.2.1. Szintetikus tesztek	20
2.2.2. Valós képek	21
3. Vékony fémlemez spline modell	23
3.1. Háromszögháló alapú kiterjesztés	24
3.1.1. Beillesztés a keretrendszerbe	28
3.2. Teszteredmények	29
3.2.1. Szintetikus tesztek	29
3.2.2. Valós képek	33
Konklúzió	35
Irodalomjegyzék	36
Publikációs jegyzék	39

Bevezetés

Képregisztrációnak azt az eljárást nevezzük, amikor kettő vagy több egymásnak megfelelő kép között keressük a geometriai kapcsolatokat. A regisztráció nagyon fontos előkészítő lépése lehet egy-egy képfeldolgozási feladat megoldásának. Ha leegyszerűsítjük a problémát két képre, adott egy *sablon* (template) és egy *megfigyelés* (observation) kép és a köztük ható geometriai transzformációt akarjuk meghatározni. A gyakorlati alkalmazásokban sok kikötést tehetünk az ismeretlen transzformációra, mint a transzformáció erőssége, invertálhatósága, topológia megőrzése.

A képek között ható transzformáció meghatározásához ismernünk kell a transzformáció erősségét, ami lineáris esetben lehet például merev test vagy affin, míg nem-lineáris esetben polinomiális vagy elasztikus transzformáció. Sokat segít a probléma kezelésében, ha ismerjük a képkészítés körülményeit. Például egy 2D-3D regisztrációs problémánál a 3D alakzatot próbáljuk illeszteni egy róla készült fényképhez, ezért ilyenkor affin és projektív transzformációkat keresünk.

Egy általános nemlineáris regisztrációs problémánál a deformációs mezőt egy tetszőleges $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ függvénnyel reprezentálunk. Sajnos a gyakorlatban nem mindig lehet meghatározni az általános deformációs mezőt, viszont *közelíthetjük* a feladatnak legjobban megfelelő transzformációs modellel. Egy ilyen modellt le lehet írni egy számítási eljárással (egy formulával) és egy N elemű vektorral, ahol N a modelltől függő természetes szám. A vektor elemei lesznek a modell paraméterei és a regisztrációs probléma megoldása során ezeket határozzuk meg.

A regisztrációs problémára adott megoldás függ a képek által hordozott információ-mennyiségtől [25]. Egy bináris képen mindössze annyi információnk van, hogy egy adott pozícióban található-e valamilyen anyag, így a regisztráció során kizárólag geometriai összefüggéseket tudunk vizsgálni. Ezzel szemben a szürkeárnyaltos képeknél rendelkezésre állnak az intenzitás értékek, amelyek alapján hasonlósági mértékek alkothatók és alkalmazhatók a probléma megoldásához.

A regisztrációs eljárásokat két nagyobb csoportba sorolhatjuk: *pontmegfeleltetés alapú* és *terület alapú* megoldások [25]. A pontmegfeleltetés alapú eljárásoknál egymásnak megfeleltethető pontpárok kinyerése után a transzformációs modelleknek megfelelően

meghatározzák a modell paramétereit (általában visszavezethető egyenletrendszerek megoldására). A pontpárok felírásához jól beazonosítható képjellemzőket használnak, mint élek, sarokpontok vagy orvosi képek esetén anatómiai pontok. Ezek meghatározása kézzel vagy automatikusan történik. A következő lépésben a kinyert pontmegfeleltetéseket párosítják, amiben fontos szerepe van az intenzitás értékek alapján készített statisztikáknak. Végül a megfeleltetett pontpárok alapján kiszámítják a transzformáció paramétereit.

Ezzel szemben a terület alapú megoldásoknál általában definiálnak egy hasonlósági mértéket a képintenzitások között, majd ebből készítenek egy optimalizálandó célfüggvényt, amely a megfelelő szélsőértékét a legjobban illesztő paraméterhalmazon veszi fel. Ilyen hasonlósági mérték lehet a keresztkorreláció vagy a kölcsönös információ. Ezek a problémák megoldhatók a klasszikus optimalizáló eljárásokkal (Gauss-Newton, Levenberg-Marquardt eljárások) [25].

Ugyanakkor mindkét eljárás függ a rendelkezésre álló képi intenzitásértékektől [25]. Amennyiben nincsenek (bináris képek) vagy nem tartalmaznak elegendő információt a képi intenzitások, az előbbi eljárások nem tudják hatékony megoldani a regisztrációs problémát. Orvosi képek illesztése során előfordulhat az is, hogy az illeszteni kívánt képek eltérő modalitással rendelkeznek, mert más képalkotó berendezéssel készültek (pl.: MR és ultrahang regisztráció). Ilyenkor a probléma megoldásához egy tisztán bináris objektumokkal dolgozó eljárásra van szükség.

A 3D objektumok regisztrációjával több korábbi cikkben is foglalkoztak. Gyakori problémák például az orvosi képek regisztrációja [14, 20] és 3D lézer szkennelével készült objektumok illesztése [21]. Ezek mellett sok általános célú keretrendszer is megtalálható [18, 10, 16].

A [10] cikkben megadott eljárásban pontfelhőket kevert Gauss-eloszlásként reprezentálnak, majd az eloszlások L_2 távolságán alapuló energiafüggvényt minimalizálva próbálják meghatározni a pontfelhők közötti transzformációt. A módszer előnye, hogy jól kezeli a szegmentálási hibákat és a kitakarásokat, valamint nem zárt alakzatok illesztését is meg tudja oldani. Hátránya, hogy a részletgazdag képek illesztése során a nagyméretű inputhalmazokon a futási idő jelentősen megnő. Ehhez hasonló eljárás található a [16] cikkben is.

A dolgozatban javasolt eljárás egy újszerű 2D illesztési módszeren alapszik, ami nem igényli pontmegfeleltetések felírását [6, 5, 23, 22]. Az eljárás azon az ötleten alapul, hogy a képek feletti térfogati (területi) integrálok és tetszőleges nem lineáris függvények együttes alkalmazásával elegendő feltételt kaphatunk a transzformációs modell paramétereinek a meghatározásához [6]. A módszer síkbeli alakzatok illesztésére alkalmas változatát alaposan megvizsgálták, mind lineáris affin [5], mind nem lineáris esetben síkhomográfia, polinomiális és vékony fémlemez spline modellek [6] esetén.

Az affin transzformációkra már megjelent a 3D kiterjesztés a [23, 22] művekben. Ugyanakkor a nem lineáris transzformációk esete 3D képeken még érintetlen terület, így a dolgozatban kísérletet teszünk a felmerülő kérdések megválaszolására.

Az eddigi munkák jó alapot szolgáltatottak valós alkalmazások megoldására is. Ilyen például a [15] műben publikált MR és ultrahang technikákkal készült prosztatata képszeletek regisztrációjára szolgáló nemlineáris keretrendszer vagy a törött objektumok újbóli összeillesztésére alkalmas, a [4] publikációban vázolt eljárás.

Az eddigi eredményeinket nemzetközi konferenciákon is publikáltuk. Jelenleg két elfogadott és egy benyújtott cikkünk van [27, 26, 28]. Ezek közül a két elfogadott cikk foglalkozik voxel alapú regisztrációval polinomiális és vékony fémlemez spline modellek esetén [27, 26], valamint a döntésre váró cikkünkben vezetjük be a háromszögháló alapú kiterjesztést [28].

A dolgozat egy matematikai áttekintéssel, majd a regisztrációs keretrendszer definiálásával indul. Ezt követően bemutatásra kerül két transzformációs modell beillesztése az általános keretrendszerbe.

Az első modell egy általános polinom függvény lesz, amin keresztül vázoljuk a voxel alapú képreprezentáció kezelését is, valamint megadunk egy hatékony numerikus számítási eljárást polinom függvények esetére.

A második modellben egy vékony fémlemez spline (*thin plate spline*, TPS) alapú technikát prezentálunk. Ebben a fejezetben a háromszögháló által határolt alakzatok illesztésén lesz a hangsúly, amivel tetszőleges transzformációs modell esetén hatékony eljárást kapunk.

Minden fejezet végén egy összefoglaló található a módszerek hatékonyságáról, összehasonlítva két pontfelhő alapú eljárással [10, 16]. Ezek mellett bemutatásra kerül egy lehetséges alkalmazási terület, amiben különböző időpontokban készült tüdő CT képek közötti geometriai összefüggéseket kellett meghatározni.

1. fejezet

Regisztrációs keretrendszer

1.1. Matematikai áttekintés

Az alfejezetben definiálni fogjuk a dolgozatban előkerülő fontosabb matematikai fogalmakat és összefüggéseket. A tételeket a bizonyításuk nélkül közöljük, megjelölve a szükséges szakirodalmat.

Először definiáljuk a többváltozós analízisben gyakran alkalmazott *Jacobi mátrix és determináns* fogalmát.

1.1. Definíció. [12] Adott differenciálható $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, ahol $\varphi = [\varphi_1, \varphi_2, \varphi_3]$ függvény esetén az elsőrendű parciális deriváltakból képzett J_φ mátrixot *Jacobi-mátrixnak* nevezzük és a következőképp definiáljuk egy tetszőleges $\mathbf{x} = [x_1, x_2, x_3] \in \mathbb{R}^3$ pontban:

$$J_\varphi(\mathbf{x}) = \begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1} & \frac{\partial \varphi_1}{\partial x_2} & \frac{\partial \varphi_1}{\partial x_3} \\ \frac{\partial \varphi_2}{\partial x_1} & \frac{\partial \varphi_2}{\partial x_2} & \frac{\partial \varphi_2}{\partial x_3} \\ \frac{\partial \varphi_3}{\partial x_1} & \frac{\partial \varphi_3}{\partial x_2} & \frac{\partial \varphi_3}{\partial x_3} \end{pmatrix} \quad (1.1)$$

A $J_\varphi(\mathbf{x})$ mátrix determinánsát *Jacobi determinánsnak* nevezzük és $|J_\varphi(\mathbf{x})|$ jelölést alkalmazuk rá.

A Jacobi determinánst leggyakrabban az *integráltranszformációnál* használjuk. Az *integráltranszformáció* az egyváltozós függvényeknél alkalmazott helyettesítéssel integrálás megfelelője a többváltozós függvényeknél és az alábbi tételben definiáljuk.

1.2. Tétel. [12] Adott két korlátos, zárt, Jordan-mérhető halmaz H és H^* és egy $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ függvény, amelyre igaz, hogy $\mathbf{y} = \varphi(\mathbf{x})$, ahol $\mathbf{x} \in H^*$ és $\mathbf{y} \in H$. Tegyük fel, hogy φ folytonosan differenciálható a H^* halmazon, továbbá $\forall \mathbf{x} \in H^* : |J_\varphi(\mathbf{x})| \neq 0$, végül a φ függvény kölcsönösen egyértelmű módon képezi le a H^* halmazt a H halmazra. Ekkor,

egy $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, a H halmazon folytonos függvény esetén teljesül a következő állítás:

$$\int_H f(\mathbf{y}) d\mathbf{y} = \int_{H^*} f(\varphi(\mathbf{x})) |J_\varphi(\mathbf{x})| d\mathbf{x}.$$

Folytassuk egy, a valós regisztrációs alkalmazások során gyakran előkerülő függvényosztály a *diffeomorfizmus* definiálásával.

1.3. Definíció. [9] Egy $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ leképezést diffeomorfizmusnak nevezünk, ha differenciálható és invertálható, illetve az inverze is differenciálható. Egy diffeomorfizmus Jacobi determinánsa nem lehet 0.

A diffeomorfizmusok nagy előnye, hogy invertálhatók és megőrzik az alakzatok topológiáját. Most definiáljuk egy transzformációs függvény *görbületi energiáját*.

1.4. Definíció. Adott $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ kétszeresen differenciálható függvény. Ekkor a

$$E_{bending} = \int_{\mathcal{F}_t} \left\{ \left(\frac{\partial^2 \phi}{\partial x_1^2} \right)^2 + \left(\frac{\partial^2 \phi}{\partial x_2^2} \right)^2 + \left(\frac{\partial^2 \phi}{\partial x_3^2} \right)^2 + 2 \left(\frac{\partial^2 \phi}{\partial x_1 \partial x_2} \right)^2 + 2 \left(\frac{\partial^2 \phi}{\partial x_2 \partial x_3} \right)^2 + 2 \left(\frac{\partial^2 \phi}{\partial x_3 \partial x_1} \right)^2 \right\} d\mathbf{x}, \quad (1.2)$$

integrált a függvény *görbületi energiájának* nevezzük.

A *görbületi energia* minimalizálásával kontrollálható a transzformáció nemlineáris hatása. Mivel az integrál csak a másodrendű parciális deriváltakat tartalmazza, a transzformációk lineáris tagjainak nincs hozzájárulása az értékhez.

Végül, a polinomokkal történő számolás során előkerülő *polinomiális tétel* kimondásával zárjuk a matematikai felvezetést.

1.5. Tétel (Polinomiális tétel). *Adott n nem negatív szám esetén*

$$(x_1 + x_2 + \dots + x_k)^n = \sum \binom{n}{r_1, r_2, \dots, r_k} x_1^{r_1} x_2^{r_2} \dots x_k^{r_k}. \quad (1.3)$$

1.2. A keretrendszer

A regisztrációs problémát két képre felírva, jelölje $\mathbf{x} \in \mathbb{R}^3$ a *sablon* és $\mathbf{y} \in \mathbb{R}^3$ a *megfigyelés* koordinátáit. Ekkor, adott egy φ diffeomorfizmus, amely a *sablon* képet a *megfigyelésbe* transzformálja, azaz:

$$\mathbf{y} = \varphi(\mathbf{x}) \iff \mathbf{x} = \varphi^{-1}(\mathbf{y}) \quad (1.4)$$

A javasolt eljárással mind a φ , mind az inverz φ^{-1} meghatározható. A továbbiakban a φ függvényt akarjuk meghatározni, tehát a *sablon* képet illesztjük a *megfigyelés*hez.

Egy konkrét esetben az általános transzformációt egy, a feladatnak legjobban megfelelő transzformációs modellel közelítjük és a modell paramétereit keressük. Ilyen modellek közül kettőt fogunk ismertetni a későbbi fejezetekben.

A paraméterek meghatározását leggyakrabban elegendő számú pontmegfeleltetések felírásával kezdik, azaz egymásnak megfeleltethető pontokat jelölnek ki a két képen, majd ezek segítségével meghatározzák a paramétereket. Ez a folyamat nem mindig automatizálható és van, hogy a pontmegfeleltetések kézzel történő meghatározása sem triviális feladat, továbbá minden esetben számolni kell a kijelölések pontatlanságával is. Ezek mellett, bináris képek estében nem állnak rendelkezésre az összehasonlítás alapját képező intenzitás értékek. Ezért a továbbiakban egy pontmegfeleltetés nélküli eljárást prezentálunk.

A pontmegfeleltetés probléma elkerülése végett integráljuk az (1.4) relációban található egyenleteket a *sablon* és *megfigyelés* pontjai felett:

$$\int_{\mathcal{F}_o} \mathbf{y} d\mathbf{y} = \int_{\varphi(\mathcal{F}_t)} \mathbf{z} d\mathbf{z}, \quad (1.5)$$

ahol $\mathcal{F}_t$ és $\mathcal{F}_o$ jelölik a *sablon* és *megfigyelés* objektumokat, valamint $\varphi(\mathcal{F}_t)$ jelöli a transzformált *sablon* képet. Az integrálási tartomány transzformációja nem mindig célravezető (pl.: voxel reprezentációnál), mivel az integrál kiszámításához először le kell generálni a módosított tartományt, ami nagyon megnöveli a számítási igényt. Ezért bizonyos esetekben érdemes *integráltranszformációt* alkalmazni az integrálok kiszámítása során, $\mathbf{z} \mapsto \varphi(\mathbf{x})$ és $d\mathbf{z} \mapsto |J_\varphi(\mathbf{x})|d\mathbf{x}$ helyettesítéseket alkalmazva:

$$\int_{\mathcal{F}_o} \mathbf{y} d\mathbf{y} = \int_{\mathcal{F}_t} \varphi(\mathbf{x}) |J_\varphi(\mathbf{x})| d\mathbf{x}. \quad (1.6)$$

Az *integráltranszformáció* végrehajtásához a jobb oldalon be kell szorozni a φ függvény Jacobi determinánsával, ami geometriailag a transzformáció végrehajtása során előforduló pontonkénti térfogat változást jelenti. Mivel a következőben mindkét számítási mód alkalmazására lesz példa, a levezetés fontosabb lépéseit mindkét eljárásra közöljük. Mindjárt fel is hívnánk a figyelmet egy fontos különbségre a két módszer között. Az *integráltranszformáció* alkalmazása megköveteli a reguláris Jacobi-mátrixot a kép összes pontjában, tehát lényegében a transzformációnak invertálhatónak kell lennie. Ezzel szemben az (1.5) egyenlet nem követeli meg az invertálhatóságot, így ez a számítási mód működhet akkor is, ha a φ függvény nem diffeomorfizmus.

A fenti egyenlethől koordinátánként egy, összesen három paraméter határozható meg. Viszont egy gyakorlatban alkalmazható transzformáció jóval több paramétert használ. Felhasználva az eredményeket a korábbi 2D keretrendszerből [6], az (1.5) és az (1.6)

egyenletek továbbra is teljesülnek, ha egy megfelelő nemlineáris $\omega : \mathbb{R}^3 \rightarrow \mathbb{R}$ függvényt alkalmazunk az egyenlet mindkét oldalán

$$\int_{\mathcal{F}_o} \omega(\mathbf{y}) d\mathbf{y} = \int_{\varphi(\mathcal{F}_t)} \omega(\mathbf{z}) d\mathbf{z} \quad (1.7)$$

$$\int_{\mathcal{F}_o} \omega(\mathbf{y}) d\mathbf{y} = \int_{\mathcal{F}_t} \omega(\varphi(\mathbf{x})) |J_\varphi(\mathbf{x})| d\mathbf{x}. \quad (1.8)$$

A [6] cikkhez hasonlóan, az eljárás alapötlete az, hogy generáljunk elegendő mennyiségű nemlineáris ω függvényt a paraméterek meghatározásához. Legyen N a szükséges paraméterek száma és jelöljük a felhasznált függvényeket a következőképp $\{\omega_i\}_{i=1}^\ell, \omega_i : \mathbb{R}^3 \rightarrow \mathbb{R}$. A probléma megoldásához legalább N függvényre lesz szükség, ezért legyen $\ell \geq N$. Alkalmazva az $\{\omega_i\}$ halmazt a következő egyenletrendszereket kapjuk az (1.7) és az (1.8) egyenletekből

$$\int_{\mathcal{F}_o} \omega_i(\mathbf{y}) d\mathbf{y} = \int_{\varphi(\mathcal{F}_t)} \omega_i(\mathbf{z}) d\mathbf{z} \quad (1.9)$$

$$\int_{\mathcal{F}_o} \omega_i(\mathbf{y}) d\mathbf{y} = \int_{\mathcal{F}_t} \omega_i(\varphi(\mathbf{x})) |J_\varphi(\mathbf{x})| d\mathbf{x}, \quad (1.10)$$

ahol $i = 1, \dots, \ell$. A fenti egyenletrendszerek megoldásával kapjuk a keresett transzformáció paramétereit.

1.3. Numerikus implementáció

A gyakorlatban csak a folytonos objektumok diszkretizált változatai állnak rendelkezésünkre, így az (1.5) egyenlet csak megközelítőleg teljesül. További problémát okozhatnak a szegmentálási hibák, ezért érdemes egy túlhatározott egyenletrendszert generálni, tehát $\ell > N$. A túlhatározottság miatt az egyenletrendszert legkisebb négyzetes értelemben oldjuk meg a *Levenberg-Marquardt* algoritmussal.

A megoldás keresése során az egyenletrendszer bal és jobb oldalainak a különbségeinek a négyzetösszegét iteratíván minimalizáljuk:

$$\sum_{i=1}^{\ell} \left(\int_{\mathcal{F}_o} \omega_i(\mathbf{y}) d\mathbf{y} - \int_{\varphi(\mathcal{F}_t)} \omega_i(\mathbf{z}) d\mathbf{z} \right)^2 \rightarrow \min \quad (1.11)$$

Ezt az értéket az eljárás algebrai hibájának nevezzük.

Az eljárás 2D változata több $\{\omega_i\}$ függvénycsaláddal tesztelésre került és egy összehasonlító elemzés készült a [6] cikkben. Ez alapján a megfelelő normalizálás mellett, az $\{\omega_i\}$ függvények nem befolyásolják az eljárás által szolgáltatott eredményeket. A nagyobb numerikus stabilitás és egyszerű kiszámíthatóság érdekében polinom függvényeket használtunk, azon belül is egyszerű hatvány függvényeket alkalmaztunk:

$$\omega_i(\mathbf{x}) = x_1^{m_i} x_2^{n_i} x_3^{o_i}, \quad (1.12)$$

ahol $(m_i, n_i, o_i) \in \mathbb{N}^3$ és $i = 1, \dots, \ell$.

1.3.1. Normalizálás

Növelhető az eljárás numerikus stabilitása azzal, ha normalizáljuk a képpontok koordinátáit a $[-0,5; 0,5]$ intervallumba [6]. Ehhez először az objektumok súlypontját az origóba kell tolni, majd a megfelelő skálafaktorokkal a koordinátákat a fenti intervallumba képezni. Ez elvégezhető az alábbi mátrixokkal

$$N_o = \begin{pmatrix} o_1 & 0 & 0 & -o_1 O_1 \\ 0 & o_2 & 0 & -o_2 O_2 \\ 0 & 0 & o_3 & -o_3 O_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad N_t = \begin{pmatrix} t_1 & 0 & 0 & -t_1 T_1 \\ 0 & t_2 & 0 & -t_2 T_2 \\ 0 & 0 & t_3 & -t_3 T_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1.13)$$

ahol T_i, O_i ($i = 1, \dots, 3$) a *sablon* és a *megfigyelés* pontjainak súlypontjai, míg t_i, o_i ($i = 1, \dots, 3$) a skálafaktorok.

A gyakorlatban az eljárás tetszőleges nemlineáris ω függvénnyel alkalmazható, ugyanakkor a lehetséges függvények értékkészlete eltérő lehet. Ez ahhoz vezet, hogy az egyes függvényeknek nem lesz egyenletes a hozzájárulása az egyenletrendszer legkisebb négyzetes hibájához. Ez további numerikus instabilitáshoz vezethet, ezért érdemes kiküszöbölni azzal, hogy normalizáljuk az integrálok értékeit is a $[-1; 1]$ intervallumba [6]. Ehhez a kiszámolt integrálokat le kell osztani egy, a maximális lehetséges integrálértéknek megfelelő konstanssal. Megfigyeléseink szerint, az iterációk során az objektumok pontjai nem transzformálódnak ki az egységkocka köré írható gömbből, így a normalizáláshoz kiszámítottuk az integrálokat az origó középpontú, $\frac{\sqrt{3}}{2}$ sugarú gömb felett

$$N_i = \int_{\|\mathbf{x}\| \leq \frac{\sqrt{3}}{2}} |\omega_i(\mathbf{x})| d\mathbf{x}. \quad (1.14)$$

A normalizálásokat alkalmazva az (1.9) egyenletrendszer a következőképpen módosul:

$$\frac{1}{N_i} \int_{N_o(\mathcal{F}_o)} \omega_i(\mathbf{y}) d\mathbf{y} = \frac{1}{N_i} \int_{\varphi(N_t(\mathcal{F}_t))} \omega_i(\mathbf{z}) d\mathbf{z}, \quad (1.15)$$

ahol $i = 1, \dots, \ell$ és a normalizáló transzformációkat az integrálási tartományon hajtjuk végre.

Ha az (1.10) egyenletrendszerből indulunk ki, a normalizált tartományok legenerálása helyett egy újabb *integráltranszformációt* alkalmazunk. A normalizáló transzformációk Jacobi determinánsai megegyeznek a(z) N_t és N_o mátrixok determinánsaival. Így az (1.10) egyenletrendszer a következő lesz:

$$\frac{|N_o|}{N_i} \int_{\mathcal{F}_o} \omega_i(\mathbf{Y}) d\mathbf{Y} = \frac{|N_t|}{N_i} \int_{\mathcal{F}_t} \omega_i(\varphi(\mathbf{X})) |J_\varphi(\mathbf{X})| d\mathbf{X}, \quad (1.16)$$

ahol $i = 1, \dots, \ell$.

Az eljárás eredményéül megkapott transzformáció a normalizált képek között hat. Ahhoz, hogy az eredeti képekre is alkalmazható legyen, denormalizálni kell azt. Mivel az eljárás során a φ diffeomorfizmus egy közelítését határozzuk meg, jelöljük a megtalált transzformációt $\hat{\varphi}^*$, a denormalizált transzformációt $\hat{\varphi}$ módon. Mivel $\hat{\varphi}^*$ a normalizált *sablont* illeszti a normalizált *megfigyeléshez*, a következő sorrendben kell végrehajtani a transzformációkat:

$$\begin{array}{ccc} \mathcal{F}_t & \xrightarrow{\hat{\varphi}} & \mathcal{F}_o \\ \downarrow & & \downarrow \\ N_t(\mathcal{F}_t) & \xrightarrow{\hat{\varphi}^*} & N_o(\mathcal{F}_o) \end{array} \quad \hat{\varphi} = N_o^{-1} \circ \hat{\varphi}^* \circ N_t \quad (1.17)$$

ahol N_o^{-1} a *megfigyelést* normalizáló transzformáció inverze:

$$N_o^{-1} = \begin{pmatrix} 1/o_1 & 0 & 0 & O_1 \\ 0 & 1/o_2 & 0 & O_2 \\ 0 & 0 & 1/o_3 & O_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.18)$$

A denormalizálást az adott transzformációs függvénytől függően kell végrehajtani.

1.3.2. Diffeomorf eredmény

A valós alkalmazásokban gyakran élnek azzal a kitételrel, hogy a fizikai deformáció invertálható és a függvény, valamint az inverze is differenciálható, más szóval egy diffeomorfizmust keresnek. Mint azt korábban jeleztük, az (1.10) egyenletrendszerben alkalmazott *integráltranszformációhoz* elengedhetetlen, hogy a φ függvény differenciálható és invertálható legyen, viszont az (1.9) egyenletrendszerben definiált irány ezt nem követeli meg. Mivel az (1.10) egyenletrendszerben foglaltak csak megközelítőleg teljesülnek, az esetleges szegmentálási és közelítési hibák miatt előfordulhat, hogy a meghatározott $\hat{\varphi}$ függvény nem lesz diffeomorfizmus.

A pontmegfeleltetésen alapuló regisztrációs eljárásoknál bizonyítható, hogy tetszőleges N pontpárra felírható egy olyan interpoláló diffeomorfizmus, ami a pontmegfeleltetéseket teljesíti [8]. Ugyanakkor a konstruktív bizonyítás során megadott diffeomorfizmus a gyakorlatban nem használható, ezért a pontmegfeleltetésre épülő eljárások is a gyakran alkalmazott transzformációs modellek megszorításával dolgoznak. Egy egyszerű javaslat a diffeomorf eredmények elérésére a nulla vagy negatív Jacobi determináns értékek büntetése az optimális eredmény meghatározása során [8, 3].

A dolgozatban javasolt eljárásnál a szintetikus generált teszt képeknél a tapasztalataink azt mutatták, hogy az elfogadható eredmények diffeomorfizmusok maradtak (pozi-

tív Jacobi determinánssal rendelkeztek), egyéb feltételek definiálása nélkül vagy a feltételek a képpontok elhanyagolható részén nem teljesültek. Ugyanakkor ezeknél a teszteknel ténylegesen diffeomorfizmus volt a fizikai deformáció.

Egy valós alkalmazás során szükséges lehet olyan kikötéseket tenni, amelyek garantálják a diffeomorf eredményt. Ilyen lehet például a fentebb említett nem-pozitív Jacobi determinánsok büntetése az egyenletrendszerben. Ez ugyanakkor negatívan befolyásolhatja az eljárás konvergenciáját.

Egy alternatív lehetőség a *görbületi energia* minimalizálása. A Jacobi determináns előjel váltása durvább transzformáció hatására jöhet csak létre, hiszen ilyenkor az objektum pontok eltűnéséről vagy „kifordulásáról” beszélünk. Ezt megakadályozza a *görbületi energia* minimalizálása, aminek használatával elérhető a diffeomorf eredmény a javasolt keretrendszerrel.

1.4. A módszer tesztelése

Az általános keretrendszerhez elkészült egy C++ alapú megvalósítás. A *Levenberg-Marquardt* algoritmushoz a *levmar* könyvtárat használtuk [13]. A teszteket Linux rendszeren futtattuk virtualizált Core i5 3.1 GHz architektúrán. A regisztrációs hibát minden esetben az eltranszformált *sablon* és a *megfigyelés* voxel képei segítségével értékeltük ki a δ -hibával:

$$\delta = \frac{|F_r \triangle F_o|}{|F_r| + |F_o|} \cdot 100\%, \quad (1.19)$$

ahol F_r és F_o jelölik a voxel képeket. A tapasztalataink alapján a 10% alatti eredmények adnak vizuálisan jó illesztést.

A tesztek során a generált *megfigyeléseket* és a *regisztrált* képeket a következő algoritmussal generáltuk (Matlab segítségével): egy nagy felbontású háromszögrácsot generálunk a megfelelő *sablon* képből, a Matlab beépített *isosurface* parancsával. Ezután eltranszformáltuk a háromszögrács pontjait, majd a transzformált háromszögrácsot újra voxel képpé alakítjuk a *binvox* programmal, ami letölthető a <http://www.cs.princeton.edu/~min/binvox> címről.

Az algoritmust összehasonlítottuk két pontfelhő alapú regisztrációs keretrendszerrel, két *kevert Gauss eloszlásra* épülő regisztrációs technikával (a GMMREG [10] és a *Coherent Point Drift* (CPD) [16]). A teszteléshez az eljárások C++ implementációját használtuk, ami elérhető a következő címen: <http://code.google.com/p/gmmreg>. A tesztelés során a paramétereket az alapértelmezett értékekre állítottuk. Először egy voxelizált alakzat összes felszíni pontjával próbáltuk tesztelni az algoritmusokat, viszont ez nem adott eredményt több, mint 12 óra futtatás után sem. Ezért az input halmazt leredukáltuk, változó felbontású háromszögrácsokat alkalmazva a [7] cikkben megadott

eljárással. Ebben a megoldásban a háromszögrács mérete kontrollálható az egyetlen háromszöget magába foglaló Delaunay gömb maximális sugarával. A teszteknel 2,5, 4,5 és 6,5 felbontásokat használtunk (az alacsonyabb felbontás több pontot eredményezett).

A gyakorlatban a szegmentált alakzat mindig tartalmazhat hibákat. Általában három főbb hibatípus fordul elő: hiányzó pontok, kitakarás és felszíni hibák. A robusztusság tesztek a [6] munkában igazolták, hogy a módszer könnyedén megoldja a hiányzó pontok problémáját, viszont mint minden terület alapú módszer, nem kezeli jól a kitakarásokat. Továbbá a nem lineáris transzformációs modellek az alakzatot a kitakarásokhoz tudják igazítani, hiszen nincs előzetes információnk a hibátlan alakzatról. Ezért a dolgozatban csak a felszíni hibák problémájával foglalkoztunk. Ennél a hibafajtánál az objektumok minden szeletén véletlenszerűen hozzáadtuk, illetve töröltünk adott méretű négyzeteket. A különböző transzformációs modellek eltérő módon kezelik ezt a hibatípust, így a konkrét eredményeket a megfelelő fejezetekben közöljük.

Végül egy szegmentált tüdő CT képekből álló tesztalmazon is teszteltük a javasolt eljárást. Itt, a szintetikus halmazoktól eltérően, nem ismert az eredeti transzformációs mező, tehát ténylegesen közelítésről beszélhetünk.

A tüdő képek regisztrációja egy nagyon fontos feladat a tüdőrák diagnosztizálásában és kezelésében. A kezelés során a betegről bizonyos időközönként CT felvételeket készítenek, amelyeknél meg kell mondani a daganat méretében beálló változást. A tüdő a légzési folyamat során is nemlineáris elváltozásokon megy keresztül, ami nagyon megnehezíti az automatikus regisztrációt. Egy bevett gyakorlat a megfeleltetések kézzel történő felvitele, viszont ez nem ad elegendően pontos eredményt a változások követéséhez.

2. fejezet

Polinomiális deformációs modell

Az elkövetkezőkben egy általános polinom alapú modell kerül bemutatásra, valamint ebben a fejezetben mutatjuk be a voxel alapú képreprezentáció használatát a keretrendszerben. A fejezet végén, a polinomiális $\{\omega_i\}$ függvények és a polinom modell együttes alkalmazása során felhasználható egyszerűsítési lehetőségeket is megadjuk.

Az egyik legegyszerűbb nem lineáris modell a polinom függvényeken alapuló modell. Fontos megjegyezni, hogy a spline alapú technikákkal ellentétben a polinom modell approximálja a két alakzat között ható deformációs mezőt, míg a spline technikáknál többnyire interpoláció történik. A paraméterek száma is jellemzően sokkal kevesebb, mint egy spline-nál, továbbá a paraméterek globálisan hatnak az alakzatra. Végül, polinomiális függvényekkel sok más transzformációs modell közelíthető, például Taylor-sorba fejtéssel.

Legyen $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, $\pi = [\pi_1, \pi_2, \pi_3]$ egy általános polinom függvény, ahol $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}$. Az általánosság megszorítása nélkül feltételezhetjük, hogy $d = \deg(\pi_1) = \deg(\pi_2) = \deg(\pi_3)$, ahol $\deg$ jelöli egy polinom fokszámát. Ekkor egy 3D polinom függvény a következő alakú:

$$\begin{aligned}\pi_1(\mathbf{x}) &= \sum_{i=0}^d \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} a_{ijk} x_1^i x_2^j x_3^k \\ \pi_2(\mathbf{x}) &= \sum_{i=0}^d \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} b_{ijk} x_1^i x_2^j x_3^k \\ \pi_3(\mathbf{x}) &= \sum_{i=0}^d \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} c_{ijk} x_1^i x_2^j x_3^k.\end{aligned}$$

A paraméterek száma kiszámítható a fokszám segítségével: $k = (d+3)(d+2)(d+1)/2$.

Ebben a fejezetben voxel alapú képreprezentációkkal foglalkozunk. Ennél a reprezentációnál az alakzatokat egy 3D rácson mintavételezzük és minden egységet azonos térfogattal látunk el. Az egységeket a középpontjuk koordinátaival jellemezzük. Mi-

vel az eltranszformált voxel alakzatok legenerálásának időigénye magas, a továbbiakban az (1.16) egyenletrendszerrel folytatjuk. Ekkor a normalizáló és a polinom transzformációkat a koordinátákon hajtjuk végre. Az egyenletrendszer alkalmazásához be kell vezetni a polinom függvények Jacobi determinánsát, amit a következő parciális deriváltakból képzünk:

$$\begin{aligned}\frac{\partial \pi_1}{\partial x_1} &= \sum_{i=1}^d \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} i a_{ijk} x_1^{i-1} x_2^j x_3^k \\ \frac{\partial \pi_1}{\partial x_2} &= \sum_{j=1}^d \sum_{i=0}^{d-j} \sum_{k=0}^{d-i-j} j a_{ijk} x_1^i x_2^{j-1} x_3^k \\ \frac{\partial \pi_1}{\partial x_3} &= \sum_{k=1}^d \sum_{i=0}^{d-k} \sum_{j=0}^{d-i-k} k a_{ijk} x_1^i x_2^j x_3^{k-1}.\end{aligned}$$

Hasonló formulák konstruálhatók π_2 és π_3 függvényekre. Most illesszük be a polinom függvényt az általános φ függvény helyére.

$$\frac{|N_o|}{N_i} \int_{F_o} \omega_i(N_o(\mathbf{Y})) d\mathbf{Y} = \frac{|N_t|}{N_i} \int_{F_t} \omega_i(\pi(N_t(\mathbf{X}))) |J_\pi(\mathbf{X})| d\mathbf{X}, \quad (2.1)$$

ahol $i = 1, \dots, \ell$. Jelölje F_t és F_o a *sablon* és a *megfigyelés* normalizált koordinátáit. Ekkor az integrál közelíthető a koordináták szummájaként:

$$\begin{aligned}\frac{|N_o|}{N_i} \int_{F_o} \omega_i(N_o(\mathbf{Y})) d\mathbf{Y} &\approx \frac{|N_o|}{N_i} \sum_{\hat{\mathbf{Y}} \in F_o} \omega_i(\hat{\mathbf{Y}}) \\ \frac{|N_t|}{N_i} \int_{F_t} \omega_i(\pi(N_t(\mathbf{X}))) |J_\pi(\mathbf{X})| d\mathbf{X} &\approx \frac{|N_t|}{N_i} \sum_{\hat{\mathbf{X}} \in F_t} \omega_i(\pi(\hat{\mathbf{X}})) |J_\pi(\hat{\mathbf{X}})|,\end{aligned}$$

ahol $i = 1, \dots, \ell$. Végül alkalmazva az (1.12) egyenletben definiált hatvány függvény alapú $\{\omega_i\}$ halmazt, a következő egyenletrendszert kapjuk:

$$\frac{|N_o|}{N_i} \sum_{\hat{\mathbf{Y}} \in F_o} \hat{Y}_1^{m_i} \hat{Y}_2^{n_i} \hat{Y}_3^{o_i} = \frac{|N_t|}{N_i} \sum_{\hat{\mathbf{X}} \in F_t} \pi_1(\hat{\mathbf{X}})^{m_i} \pi_2(\hat{\mathbf{X}})^{n_i} \pi_3(\hat{\mathbf{X}})^{o_i} |J_\pi(\hat{\mathbf{X}})|, \quad (2.2)$$

ahol $i = 1, \dots, \ell$.

A fenti egyenletrendszert megoldva megkapjuk a két alakzatot illesztő polinom függvényeket. Az eredmény denormalizálásához az N_t normalizálási transzformációt be kell építeni a polinomok bázisfüggvényeibe és az összes paramétert meg kell szorozni az N_o^{-1} inverz mátrixban található skálafaktorokkal. Végül az inverz mátrixban található eltolási

értékeket hozzá kell adni az eredményhez:

$$\hat{\pi}_1(\mathbf{x}) = O_1 + \frac{1}{o_1} \sum_{i=0}^d \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} a_{ijk}^* ((x_1 - T_1)t_1)^i ((x_2 - T_2)t_2)^j ((x_3 - T_3)t_3)^k \quad (2.3)$$

$$\hat{\pi}_2(\mathbf{x}) = O_2 + \frac{1}{o_2} \sum_{i=0}^d \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} b_{ijk}^* ((x_1 - T_1)t_1)^i ((x_2 - T_2)t_2)^j ((x_3 - T_3)t_3)^k \quad (2.4)$$

$$\hat{\pi}_3(\mathbf{x}) = O_3 + \frac{1}{o_3} \sum_{i=0}^d \sum_{j=0}^{d-i} \sum_{k=0}^{d-i-j} c_{ijk}^* ((x_1 - T_1)t_1)^i ((x_2 - T_2)t_2)^j ((x_3 - T_3)t_3)^k, \quad (2.5)$$

ahol a^* , b^* és c^* jelölik a meghatározott, normalizált együtthatókat. Az így kapott algoritmus komplexitása

$$\mathcal{O}(\ell|F_o| + I|F_t|(\ell + k + 9J)), \quad (2.6)$$

ahol I a *Levenberg-Marquardt* algoritmus futása során szükséges függvényhívások számát, k a transzformáció kiszámításához, J pedig Jacobi determináns kiszámításához szükséges lépések számát jelölik. Az algoritmus pszeudokódját megadjuk az 1. algoritmusban.

1. algoritmus. A voxel alapú algoritmus pszeudokódja

Bemenet: *sablon és megfigyelés voxel képek*

Kimenet: Az illesztő transzformáció, $\hat{\pi}$ paraméterei

- 1: Válasszunk $\ell > k$ nemlineáris $\{\omega_i\}_{i=1}^{\ell}$ függvényt és minden ω_i függvényre számítsuk ki a normalizáló konstansokat az (1.14) formulák segítségével
 - 2: Normalizáljuk a képek koordinátáit az N_o és N_t transzformációkkal a $[-0,5; 0,5]$ tartományba.
 - 3: Konstruáljuk meg a (2.2) egyenletrendszert.
 - 4: Oldjuk meg az egyenletrendszert legkisebb négyzetes értelemben a *Levenberg-Marquardt* algoritmussal. Inicializáljuk az algoritmust az identikus transzformációval.
 - 5: Denormalizáljuk a megkapott $\hat{\pi}^*$ transzformációt a (2.3), (2.4) és (2.5) formulák segítségével, ami megadja a keresett transzformációt
-

2.1. A koordináták és a paraméterek szétválasztása

A [6] munkában a szerzők bizonyítanak egy általános tételt, ami kimondja, hogy a polinomiális deformáció és polinomiális ω függvények esetében a kép koordináták és a transzformáció paraméterei szétválaszthatók, így a koordináták feletti integrálokat elegendő egyszer kiszámítani. A elkövetkezőkben megmutatjuk, hogyan adaptálható a 2D eljárás 3D-re és kielemezzük a hatékonyságát harmadfokú polinomok esetén.

Vizsgáljuk meg az előző alfejezetben kapott (2.2) egyenletrendszert. Az egyenletek bal oldalán egyértelműen polinom függvényeket találhatunk. A jobb oldalon található szummákban polinom függvények hatványait találjuk, amelyek így továbbra is polinomok lesznek. A Jacobi determináns kiszámításánál polinomok szorzatát és összegét kell kiszámítanunk, így ez is felírható polinom alakban. Ebből következően a jobb oldali szummában található tagok felírhatók a *polinomiális tétel* segítségével a következőképpen:

$$\pi_1^{n_i} \pi_2^{m_i} \pi_3^{o_i} |J_\pi(\hat{\mathbf{X}})| = \sum_{q=0}^{d_i} \sum_{r=0}^{d_i-q} \sum_{s=0}^{d_i-q-r} g_{iqrs} \hat{X}_1^q \hat{X}_2^r \hat{X}_3^s, \quad (2.7)$$

ahol $d_i = d(n_i + m_i + o_i) + 3(d - 1)$ lesz az új polinom fokszáma, d a transzformációs polinom fokszáma, g_{iqrs} függvények pedig az eredeti együtthatók polinomjai. Fontos megjegyezni, hogy a g_{iqrs} függvények nem tartalmazzák az alakzatok koordinátáit, így függetlenek az alakzattól. Visszahelyettesítve a (2.2) egyenletrendszer jobb oldalába kapjuk a következő egyenletet

$$\frac{|N_t|}{N_i} \sum_{\hat{\mathbf{X}} \in F_t} \pi_1(\hat{\mathbf{X}})^{m_i} \pi_2(\hat{\mathbf{X}})^{n_i} \pi_3(\hat{\mathbf{X}})^{o_i} |J_\pi(\hat{\mathbf{X}})| = \frac{|N_t|}{N_i} \sum_{q=0}^{d_i} \sum_{r=0}^{d_i-q} \sum_{s=0}^{d_i-q-r} g_{iqrs} \sum_{\hat{\mathbf{X}} \in F_t} \hat{X}_1^q \hat{X}_2^r \hat{X}_3^s. \quad (2.8)$$

A két formula matematikailag megegyezik, csak a számítási eljárásban térnek el. A jobb oldalon található formula sokkal több tagot tartalmaz, mint a bal oldalon lévő, viszont a szumma a koordináták felett kiszámítható előre és csak a megfelelő értéket kell behelyettesíteni. A módszer alkalmazásához meg kell határozni azt a minimális képméretet, aminél már megéri az együtthatókat és a koordinátákat ilyen módon szétválasztani.

Az elemzéshez először vezessünk be egy függvényt, ami egy fokszámhoz hozzárendeli az ilyen fokú teljes polinomokban található tagok számát: $\gamma : \mathbb{R} \rightarrow \mathbb{R}, \gamma(d) = (d + 3)(d + 2)(d + 1)/6$. Jelölje d_M a maximális fokú polinomot a kifejtés után, azaz $d_M = \max\{d_i \mid i = 1, \dots, \ell\}$. Ekkor a koordináták feletti szumma:

$$\sum_{\hat{\mathbf{X}} \in F_t} \hat{X}_1^q \hat{X}_2^r \hat{X}_3^s \quad (2.9)$$

kiszámítható $\mathcal{O}(|F_t| \gamma(d_M))$ időben, ha a magasabb fokszámú tagok számítása során felhasználjuk az alacsonyabb fokszámú tagokat.

Vizsgáljuk meg most a g_{iqrs} függvényeket. Legrosszabb esetben teljes polinomok lesznek a transzformáció paramétereiben, amelyek fokszáma $m_i + n_i + o_i + 1$, ahol m_i , n_i és o_i az ω_i függvényből, az utolsó tag pedig a Jacobi determinánsból jön. Így, ha feltételezzük, hogy a g_{iqrs} függvények egy tagját konstans időben tudjuk számítani, összesen $\mathcal{O}(\gamma(m_i + n_i + o_i + 1))$ műveletet igényel egy g_{iqrs} függvény kiszámítása. Egyetlen egyenletben ezekből $\gamma(d_i)$ található, így a komplexitása $\mathcal{O}(\gamma(d_i) \gamma(m_i + n_i + o_i + 1))$. Tehát, ha

a 2.2 egyenletrendszert a szétválasztási eljárás segítségével számítjuk, annak időigénye

$$\mathcal{O}(\ell|F_o| + \gamma(d_M)|F_t| + I \sum_{i=1}^{\ell} \gamma(d_i)\gamma(n_i + m_i + o_i + 1)) \quad (2.10)$$

lesz, ahol I a *Levenberg-Marquardt* algoritmus futása során szükséges függvényhívások száma.

2.1.1. Esettanulmány

A tesztjeink során harmadfokú polinomokat használtunk, így vizsgáljuk meg ezek viselkedését a szétválasztási eljárás során. Egy harmadfokú polinomnak összesen 60 paraméterre van szüksége, ezért használjuk a következő $\{\omega_i\}$ halmazt:

$$\{(n_i, m_i, o_i)\}_{i=1}^{64} = \{(a, b, c) \mid 0 \leq a, b, c \leq 3\}, \quad (2.11)$$

ami összesen 64 egyenletet tartalmaz. A korábban definiált formulák segítségével megadjuk a következő értékeket:

$$\begin{aligned} d_M &= 33 \\ \gamma(d_M) &= 7\,140 \\ \sum_{i=1}^{64} \gamma(d_i)\gamma(n_i + m_i + o_i + 1) &= 16\,453\,488 \end{aligned}$$

A tesztjeink során 380 volt az átlagos függvényhívások száma a megadott ω_i függvények és deformációs modell esetén. Fontos megjegyezni, hogy a függvényhívások száma megegyezik a két módszernél, mivel a két formula matematikailag ekvivalens. Behelyettesítve a kiszámított értékeket a (2.10) összefüggésbe a következőket kapjuk:

$$\mathcal{O}(64|F_o| + 7\,140|F_t| + 380 \times 16\,453\,488), \quad (2.12)$$

míg felhasználva a szétválasztás nélküli komplexitás elemzést a (2.6) összefüggésből:

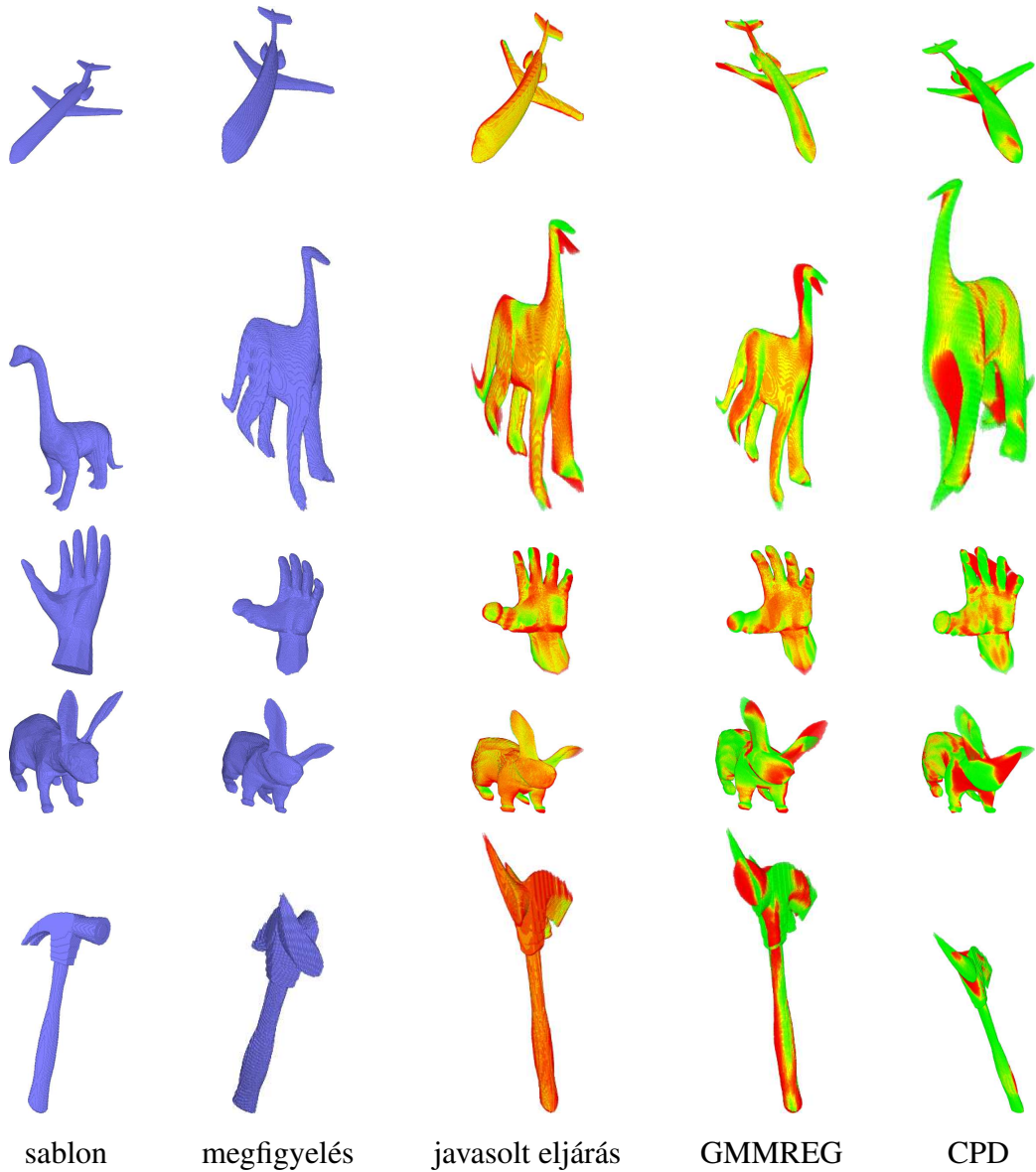
$$\mathcal{O}(64|F_o| + 380(64 + 60 + 90)|F_t|) \quad (2.13)$$

összefüggéseket kapjuk. Kifejezve a *sablon* koordinátáinak számában, a szétválasztást megéri végrehajtani, ha

$$|F_t| \geq 84\,286. \quad (2.14)$$

2.2. Teszteredmények

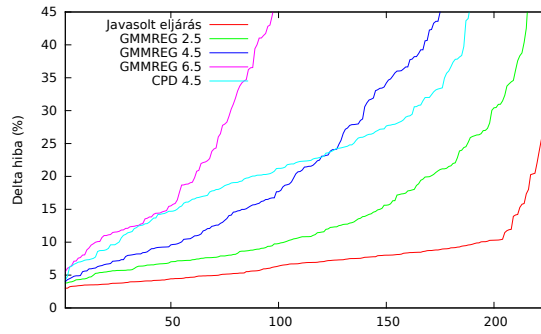
A tesztek során harmadfokú polinomiális modellel dolgoztunk (tehát $d = 3$), ami összesen $k = (d+3)(d+2)(d+1)/2 = 60$ paramétert használ. A megoldáshoz 64 egyenletből álló egyenletrendszert generáltunk, felhasználva a következő halmazt $\omega_i(\mathbf{x}) = x_1^{n_i} x_2^{m_i} x_3^{o_i}$, ahol $\{(n_i, m_i, o_i)\}_{i=1}^{64} = \{(a, b, c) \mid 0 \leq a, b, c \leq 3\}$.



2.1. ábra. Eredmények a tesztalmazból. A *megfigyelést* és *regisztrált* képeket egymásra vetítettük úgy, hogy a közös területet sárgával, különböző területeket zölddel és pirossal színeztük.

2.2.1. Szintetikus tesztek

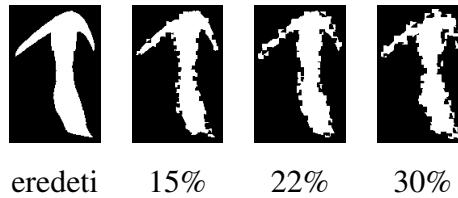
Az algoritmus teszteléséhez először egy 550 képből álló szintetikus tesztalmazt generáltunk. A generálás során másod- és harmadfokú polinomokkal transzformáltunk el *sablon* képeket. A polinomok paramétereit véletlenszerűen generáltuk a következő szabályok szerint: $a_{100}, b_{010}, c_{001} \in [0,5; 1,5]$, $a_{010}, a_{001}, b_{100}, b_{001}, c_{100}, c_{010} \in [-0,25; 0,25]$ és minden további paramétert a $[-0,5; 0,5]$ intervallumból vettük. Továbbá, $a_{000} = b_{000} = c_{000} = 0$ (tehát nincs eltolás), mivel a kezdeti normalizálás minden komolyabb eltolást eltüntet. Az eredményekre példákat prezentálunk a 2.1 ábrán és a statisztikák megtalálható a 2.1 táblázatban.



2.2. ábra. Összehasonlítás a GMMREG [10] és a CPD [16] eljárásokkal.

Időigény (perc)			$\delta(\%)$		
m	μ	σ	m	μ	σ
13,96	6,85	17,84	7,02	6,06	4,58

2.1. táblázat. A szintetikus teszt eredményei (m – medián, μ – átlag és σ – szórás).



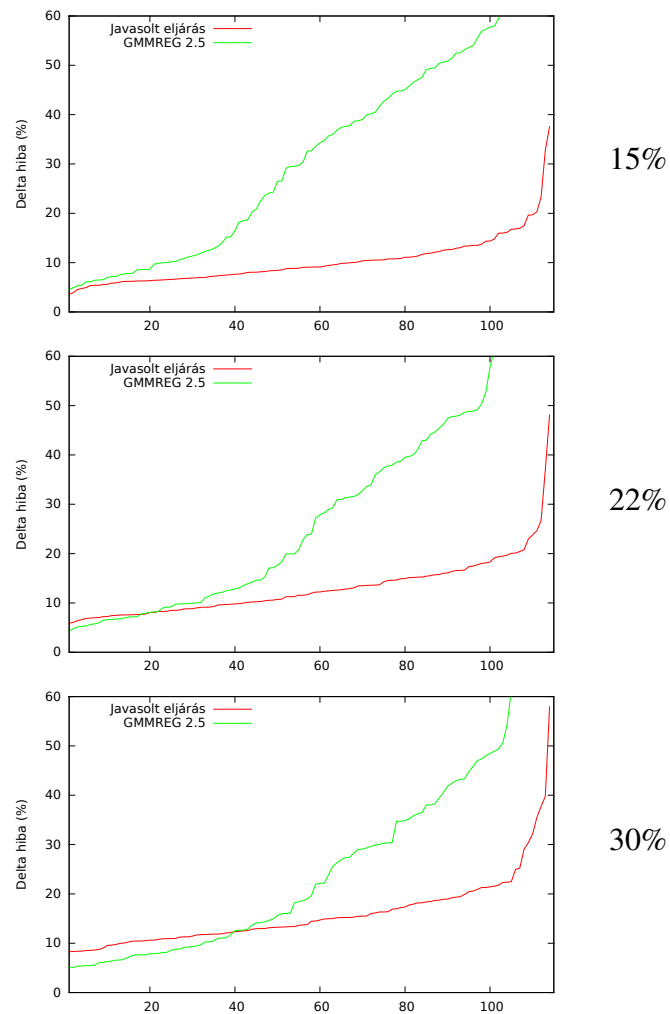
2.3. ábra. Példa a szegmentálási hibákra.

Az eljárást összehasonlítottuk egy kisebb tesztalmazon a GMMREG és CPD eljárásokkal. Az eredményeket a 2.2 diagramon ábrázoltuk. Az ábrán látszik, hogy a javasolt módszer jobb illesztést adott a tesztalmazon, viszont megjegyezzük, hogy ezek az algoritmusok nem polinomiális transzformációs modelleket alkalmaztak.

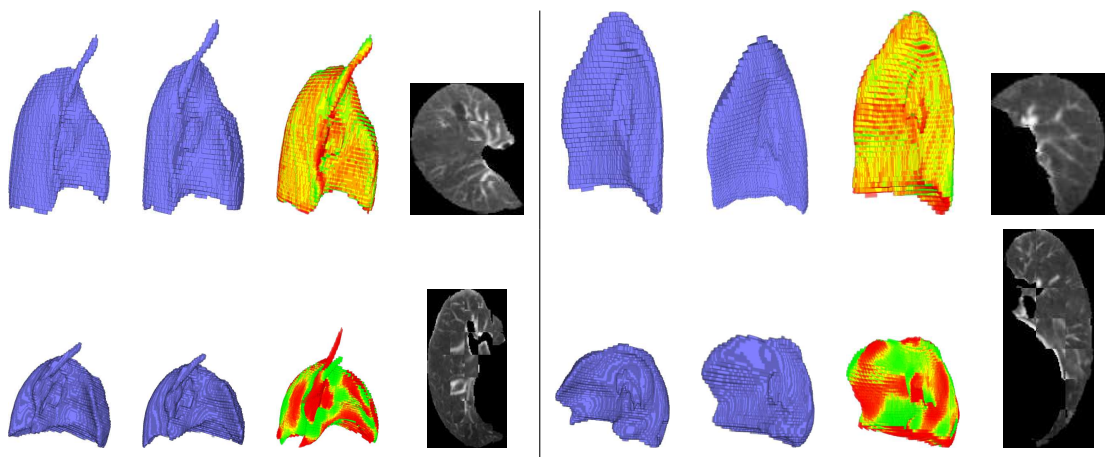
A robusztusság teszténél a teljes objektum 15%, 22% és 30% tette ki a generált hiba. Példa képszeletek láthatók a 2.3 ábrán. A tesztképeket illesztettük a legjobb GMMREG algoritmussal (ahol a Delenauy gömb sugara 2,5 volt) és az összehasonlítást megjelentettük a 2.4 ábrán. Ezek alapján az eljárás robusztusnak tekinthető, ha a szegmentálási hiba nem haladja meg az objektumpontok számának 22%-át.

2.2.2. Valós képek

A valós tüdő CT képeken is teszteltük az eljárás hatékonyságát a polinomiális modellel. Itt azt tapasztaltuk, hogy a polinomiális modell jól közelíti a valós deformációs mezőt. A kapott 8 képpárra a hibák mediánja 8,41%, átlaga 7,99%, szórása 3,03% volt. Az eredményképek a 2.5 ábrán láthatók, ahol egymásra vetítettük a regisztrált és kiindulási szürkeárnyaltos képeket is egy 8×8 sakktábla minta szerint.



2.4. ábra. Robusztusság teszt és összehasonlítás a GMMREG algoritmussal.



2.5. ábra. A tüdő képek illesztése. Az első sor a legjobb eredményeket, az utolsó a legrosszabb eredményeket tartalmazza. A szegmentált képeket a Mediso Ltd. *InterView Fusion* programja készítette.

3. fejezet

Vékony fémlemez spline modell

Ebben a fejezetben a *vékony fémlemez spline* modellel foglalkozunk, amelyet a továbbiakban az angol elnevezés alapján TPS-nek rövidítünk (*thin plate spline*). Továbbá adaptáljuk a keretrendszer háromszög felületi hálókkaal megadott alakzatok illesztésére.

A TPS egy széles körben alkalmazott nem lineáris interpolációs technika [24, 9, 2]. A pontmegfeleltetések meghatározása után egyszerű formulákkal számítható egy sima interpolációs függvény [2], ami alkalmas lehet magasabb dimenziós interpolációkra is. A számítások alapját előre definiált kontrolpontok és ezek képei adják (a klasszikus alkalmazásokban a pontmegfeleltetések). A javasolt eljárásban azonban nem használunk pontmegfeleltetéseket, tehát a TPS-t approximációs modellként használjuk.

Jelölje $\varsigma : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ a 3D TPS transzformációt. Adott K db kontrolpont: $c_k \in \mathbb{R}^3$ és a $a_{ij}, w_{ki} \in \mathbb{R}$ paraméterek esetén, ahol $i = 1, \dots, 3, j = 1, \dots, 4$ és $k = 1, \dots, K$, a transzformációs függvény felírható a következő alakban:

$$\varsigma_i(\mathbf{x}) = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + a_{i4} + \sum_{k=1}^K w_{ki} \|c_k - \mathbf{x}\|. \quad (3.1)$$

A TPS paramétereit két csoportba oszthatjuk. Az $a_{ij}, i = 1, \dots, 3, j = 1, \dots, 4$ paraméterek egy affin transzformációt írnak le és így a hatásuk azonos lesz a teljes képre nézve. A $w_{ki}, i = 1, \dots, 3, k = 1, \dots, K$ paraméterek határozzák meg egy adott kontrolpont hozzájárulását az $\mathbf{x}$ pont képének kialakulásában, a *radiális bázisfüggvény* segítségével. A bázisfüggvény 3D-ben az adott kontrolponttól vett Euklideszi-távolság lesz [9]. Minden kontrolpont hatása globális lesz, tehát ahhoz, hogy a TPS jól kezelje a lokális változásokat, több kontrolpontra van szükség a deformáció közelében. A szükséges paraméterek száma $N = 3(K + 4)$.

A kontrolpontok súlyparamétereinek további feltételeket is teljesíteniük kell [24]. Ezek a feltételek meggátolják, hogy a fémlemezre ható erőviszonyok hatására elmoz-

duljon vagy elforogjon valamelyik tengely irányába:

$$\sum_{k=1}^K w_{ki} = 0 \quad \text{és} \quad \sum_{k=1}^K c_{kj} w_{ki} = 0, \quad i, j = 1, 2, 3. \quad (3.2)$$

Mivel az eljárásunkban nem használunk pontmegfeleltetéseket, a kontrolpontokat tetszőleges pozíciókba lerakhatjuk. Ilyenkor érdemes a megoldandó feladathoz legmegfelelőbbben kialakítani a kontrolpontok helyzetét, figyelembe véve azt, hogy a hatásukat minden képpontra kifejtik. Amennyiben nem rendelkezünk előzetes információkról a kontrolpontok optimális elhelyezéséről, célszerű egy egyenletes ráccsal lefedni a kép területét, így a deformációs mező lokális hatásait is könnyebben közelíthetjük.

3.1. Háromszögháló alapú kiterjesztés

A továbbiakban feltételezzük, hogy a *sablon* és a *megfigyelés* alakzatok háromszöghálók-
ikkal adottak. Jelöljük ezeket rendre T_Δ és O_Δ , valamint a rácsok által kifeszített térfog-
atokat F_t^Δ és F_o^Δ módon.

Először terjesszük ki a ς transzformációs függvényt a háromszöghálókra. Legyen $\varsigma(T_\Delta)$ egy új háromszögháló, aminél

$$\varsigma(T_\Delta) = \{(\varsigma(A), \varsigma(B), \varsigma(C)) \mid (A, B, C) \in T_\Delta\}. \quad (3.3)$$

Jelölje a $\varsigma(T_\Delta)$ által meghatározott térfogatot $\mathcal{D}_\varsigma$. Mivel $T_\Delta \approx \mathcal{F}_t$ és $O_\Delta \approx \mathcal{F}_o$,

$$\int_{F_o^\Delta} \mathbf{y} d\mathbf{y} \approx \int_{\mathcal{F}_o} \mathbf{y} d\mathbf{y} \quad (3.4)$$

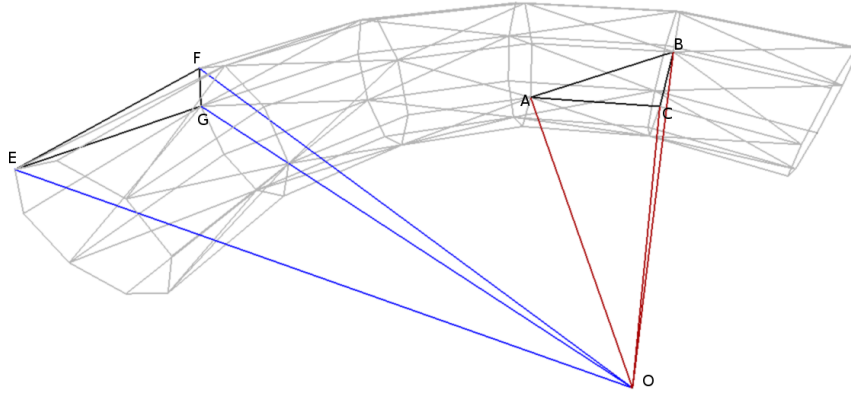
$$\int_{\mathcal{D}_\varsigma} \mathbf{x} d\mathbf{x} \approx \int_{\varsigma(\mathcal{F}_t)} \mathbf{z} d\mathbf{z}. \quad (3.5)$$

Feltételezzük, hogy minden háromszög csúcspontjait konzekvensen az óramutató járásá-
val ellentétes irányban sorolunk fel egy külső pontból nézve. Ekkor a (3.4) és a (3.5)
integrálok a háromszöghálók felett számíthatók megfelelően generált tetraéderek feletti
integrálok összegeiként:

$$\int_{F_o^\Delta} \mathbf{y} d\mathbf{y} = \sum_{o \in O_\Delta} \text{sign}(\text{vol}(\mathcal{T}_o)) \int_{\mathcal{T}_o} \mathbf{y} d\mathbf{y} \quad (3.6)$$

$$\int_{\mathcal{D}_\varsigma} \mathbf{z} d\mathbf{z} = \sum_{\pi \in \varsigma(T_\Delta)} \text{sign}(\text{vol}(\mathcal{T}_\pi)) \int_{\mathcal{T}_\pi} \mathbf{z} d\mathbf{z} \quad (3.7)$$

A következő fontos lépés a tetraéderek generálása: minden $o = (A, B, C)$ háromszög
esetén, legyen $T_o = (A, B, C, 0)$ egy olyan tetraéder, melynek alapját az o háromszög
pontjai és a negyedik csúcspontját az origó adja. Fontos, hogy a negyedik pont minden



3.1. ábra. Tetraéder generálás egy tórusz háromszöghálójából: a kék tetraéder pozitív, a piros negatív térfogatú.

tetraéder számára ugyanaz legyen. Ezen a feltételen kívül a negyedik pont tetszőleges pont lehetne, viszont célszerű az origót választani negyedik pontnak, mert így a további számításaink egyszerűsödnek. Ezzel a konstrukcióval a T_o tetraéder előjeles területe a számítható

$$\text{vol}(T_o) = \frac{1}{6} \begin{vmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{vmatrix}, \quad (3.8)$$

módon, ahol $A, B, C \in \mathbb{R}^3$ az o háromszög csúcspontjai. A térfogat előjele fontos, mert ez adja meg a háromszögek orientációját az origóból nézve. Egy példa látható a 3.1 ábrán, ahol egy tórusz háromszöghálójához adtunk meg tetraédereket. A tetraéderek térfogatának előjele eltér a két háromszög körüljárási irányának változása miatt.

Mivel a háromszögháló transzformációjakor csak a csúcspontok transzformációjáról beszélünk, a transzformált alakzat „legenerálása” is csak ebből áll. Továbbá egy háromszögháló lényegesen kevesebb csúcspontot tartalmaz, mint egy voxel kép, így a transzformáció is gyorsan számítható, akár iterációnként is. Ebből következően a keretrendszer (1.5) egyenletéből fogunk kiindulni, ami háromszöghálókra felírva a következőképpen változik:

$$\sum_{o \in O_\Delta} \text{sign}(\text{vol}(T_o)) \int_{T_o} \mathbf{y} d\mathbf{y} = \sum_{\pi \in \varsigma(T_\Delta)} \text{sign}(\text{vol}(T_\pi)) \int_{T_\pi} \mathbf{z} d\mathbf{z}, \quad (3.9)$$

ahol $\mathcal{T}_o$ és $\mathcal{T}_\pi$ jelölik a T_o és T_π tetraéderek által kifesztett térfogatot.

A következő lépés az ω függvények bevezetése. A fenti integráloknál egy ω függvény a tetraéderek feletti integrálba kerülne, amit általános esetben csak numerikus közelítéssel tudunk meghatározni. Ugyanakkor, ha az ω függvények polinomok, levezethetünk egyszerű, zárt formulákat a tetraéder feletti integrálok egzakt meghatározására. A továbbiakban alkalmazzunk egyszerű hatványfüggvényeket: $\omega_i(\mathbf{x}) = x_1^{n_i} x_2^{m_i} x_3^{o_i}$, ahol

$\{(n_i, m_i, o_i)\}_{i=1}^\ell = \{(a, b, c) \mid a + b + c = O\}$ és $O \in \{0, \dots, O_{max}\}$. Ekkor az (1.9) egyenletrendszer a következő lesz:

$$\sum_{o \in O_\Delta} \text{sign}(\text{vol}(T_o)) \int_{\mathcal{T}_o} y_1^{m_i} y_2^{n_i} y_3^{o_i} d\mathbf{y} = \sum_{\pi \in \zeta(T_\Delta)} \text{sign}(\text{vol}(T_\pi)) \int_{\mathcal{T}_\pi} z_1^{m_i} z_2^{n_i} z_3^{o_i} d\mathbf{z}. \quad (3.10)$$

Vegyük észre, hogy a jobb és bal oldalon található formulák megegyeznek, csak a be-
menetek változnak, továbbá a $\mathcal{T}_o$ és $\mathcal{T}_\pi$ térfogatok felett kiszámított integrálok egyszerű
geometriai momentumok, így alkalmazhatók a momentumok hatékony számítására java-
solt eljárások [11, 19].

Vegyük az előző egyenlet bal oldalát (a számítás szempontjából közömbös, hogy a
bal vagy a jobb oldallal foglalkozunk). Követve az [1, 19] cikkek gondolatmenetét, ír-
juk fel a tetraéderek pontjait a generáló háromszöghöz tartozó baricentrikus koordináta-
rendszerben. Ekkor egy $\mathcal{T}_o$ tetraéder tetszőleges $\mathbf{p} \in \mathbb{R}^3$ pontjának koordinátája megad-
ható az $\alpha, \beta, \gamma \in \mathbb{R}$ súlyok és a $\rho \in \mathbb{R}$ távolság segítségével:

$$\mathbf{p} = \rho \mathbf{p}' \quad \mathbf{p}' = \alpha A + \beta B + \gamma C,$$

ahol $\alpha, \beta, \gamma > 0$, $\alpha + \beta + \gamma = 1$ és $0 \leq \rho \leq 1$, valamint A, B, C az adott o háromszög
csúcspontjai. Az integrálok felírásához *integráltranszformációt* kell alkalmazni a bázis-
transzformáció elvégzése helyett, aminek Jacobi determinánsa $6 \text{vol}(T_o) \rho^2$. Ezután az új
bal oldal [1, 19]

$$\sum_{o \in O_\Delta} 6 \text{vol}(T_o) \int_0^1 \rho^{M_i+2} \int_0^1 \int_0^{1-\alpha} (\alpha A_1 + \beta B_1 + \gamma C_1)^{m_i} (\alpha A_2 + \beta B_2 + \gamma C_2)^{n_i} (\alpha A_3 + \beta B_3 + \gamma C_3)^{o_i} d\beta d\alpha d\rho, \quad (3.11)$$

ahol $\gamma = 1 - \alpha - \beta$ és $M_i = m_i + n_i + o_i$. Az első egyszerűsítési lehetőség abból
adódik, hogy a ρ távolság független a belső integráloktól és ezért az egyik integrál függet-
lenül számítható: $\int_0^1 \rho^{M_i+2} = \frac{1}{M_i+3}$. Helyettesítsük be a megkapott formulát a bal oldali
összegbe:

$$\frac{1}{M_i+3} \sum_{o \in O_\Delta} 6 \text{vol}(T_o) S_{m_i n_i o_i}, \quad (3.12)$$

ahol

$$S_{m_i n_i o_i} = \int_0^1 \int_0^{1-\alpha} (\alpha A_1 + \beta B_1 + \gamma C_1)^{m_i} (\alpha A_2 + \beta B_2 + \gamma C_2)^{n_i} (\alpha A_3 + \beta B_3 + \gamma C_3)^{o_i} d\beta d\alpha. \quad (3.13)$$

Következő lépésként fejtsük ki a hatványokat az $S_{m_i n_i o_i}$ formulában. Az egyszerűbb
jelölés érdekében, definiáljuk a trinomiális együtthatókat [19, 11]:

$$(a, b, c) = \frac{(a+b+c)!}{a!b!c!}.$$

Ekkor $S_{m_i n_i o_i}$ kifejtése és átrendezése után az alábbi formulát kapjuk:

$$S_{m_i n_i o_i} = \sum_{a_1+a_2+a_3=m_i} \sum_{b_1+b_2+b_3=n_i} \sum_{c_1+c_2+c_3=o_i} (a_1, b_1, c_1) A_1^{a_1} A_2^{b_1} A_3^{c_1} \times \\ (a_2, b_2, c_2) B_1^{a_2} B_2^{b_2} B_3^{c_2} (a_3, b_3, c_3) C_1^{a_3} C_2^{b_3} C_3^{c_3} \int_0^1 \int_0^{1-\alpha} \alpha^{a_1+a_2+a_3} \beta^{b_1+b_2+b_3} \gamma^{c_1+c_2+c_3}. \quad (3.14)$$

Az utolsó két integrál is kiszámítható:

$$\int_0^1 \int_0^{1-\alpha} \alpha^{m_i} \beta^{n_i} \gamma^{o_i} = \frac{m_i! n_i! o_i!}{(M_i + 2)!},$$

ami így függetlenné válik a szummáktól. Emeljük ki az $S_{m_i n_i o_i}$ összefüggésből a kiszámított integrált, így az alábbi formulát kapjuk a (3.12) formulából:

$$6 \frac{m_i! n_i! o_i!}{(M_i + 3)!} \sum_{o \in O_\Delta} \text{vol}(T_o) S_{m_i n_i o_i}, \quad (3.15)$$

ahol

$$S_{m_i n_i o_i} = \sum_{a_1+a_2+a_3=m_i} \sum_{b_1+b_2+b_3=n_i} \sum_{c_1+c_2+c_3=o_i} (a_1, b_1, c_1) A_1^{a_1} A_2^{b_1} A_3^{c_1} (a_2, b_2, c_2) B_1^{a_2} B_2^{b_2} B_3^{c_2} (a_3, b_3, c_3) C_1^{a_3} C_2^{b_3} C_3^{c_3}. \quad (3.16)$$

A fenti számítási mód komplexitása $\mathcal{O}(M_i^9)$. Az $S_{m_i n_i o_i}$ számítása egyszerűsíthető a következő formulák használatával:

$$C_{ijk}(\mathbf{p}) = (i, j, k) p_1^i p_2^j p_3^k \quad (3.17)$$

$$D_{abc}(\mathbf{p}, \mathbf{q}) = \sum_{i=0}^a \sum_{j=0}^b \sum_{k=0}^c C_{ijk}(\mathbf{p}) C_{a-i, b-j, c-k}(\mathbf{q}) \quad (3.18)$$

az alábbi módon [19]:

$$S_{m_i n_i o_i} = \sum_{a=0}^{m_i} \sum_{b=0}^{n_i} \sum_{c=0}^{o_i} C_{abc}(A) D_{m_i-a, n_i-b, o_i-c}(B, C), \quad (3.19)$$

amivel a számítási igény $\mathcal{O}(M_i^6)$ -ra csökkenthető. Ezt továbbgondolva a fenti formulák számíthatók rekurzívan a következő összefüggésekkel [11]:

$$C_{ijk}(\mathbf{p}) = p_1 C_{i-1, j, k}(\mathbf{p}) + p_2 C_{i, j-1, k}(\mathbf{p}) + p_3 C_{i, j, k-1}, \quad (3.20)$$

$$D_{ijk}(\mathbf{p}, \mathbf{q}) = \begin{cases} 0, & \text{ha } i < 0 \vee j < 0 \vee k < 0 \\ 1, & \text{ha } i = j = k = 0 \\ p_1 D_{i-1, j, k}(\mathbf{p}, \mathbf{q}) + p_2 D_{i, j-1, k}(\mathbf{p}, \mathbf{q}) + \\ p_3 D_{i, j, k-1}(\mathbf{p}, \mathbf{q}) + C_{ijk}(\mathbf{q}), & \text{egyébként} \end{cases} \quad (3.21)$$

$$S_{ijk}(T_o) = \begin{cases} 0, & \text{ha } i < 0 \vee j < 0 \vee k < 0 \\ 1, & \text{ha } i = j = k = 0 \\ A_1 S_{i-1, j, k}(T_o) + A_2 S_{i, j-1, k}(T_o) + \\ A_3 S_{i, j, k-1}(T_o) + D_{ijk}(B, C), & \text{egyébként} \end{cases}. \quad (3.22)$$

A rekurzív formuláknak köszönhetően a komplexitás lecsökkenthető $\mathcal{O}(M_i^3)$ -re [11].

3.1.1. Beillesztés a keretrendszerbe

A transzformáció legenerálásához hasonlóan, a normalizálás is végrehajtható a háromszöghálókön. Így az (1.15) egyenletrendszer felírható az alábbi alakban:

$$\frac{1}{N_i} \sum_{o \in O_\Delta} \text{sign}(\text{vol}(T_o)) \int_{T_o} y_1^{m_i} y_2^{n_i} y_3^{o_i} d\mathbf{y} = \frac{1}{N_i} \sum_{\pi \in \varsigma(N_t(T_\Delta))} \text{sign}(\text{vol}(T_\pi)) \int_{T_\pi} z_1^{m_i} z_2^{n_i} z_3^{o_i} d\mathbf{z}, \quad (3.23)$$

ahol $i = 1, \dots, \ell$ és az integrálókat az előző szakaszban kifejtett eljárás szerint számítjuk.

Mivel minden formula rekurzív az alacsonyabb fokszerű eredmények tetraéderenként letárolhatók, a magasabb fokszerű polinomok számításához. Így az integrálok kiszámítása egyetlen tetraéder fölött $\mathcal{O}(M^3)$ komplexitású, ahol M a momentumok maximális rendje a $\{\omega_i\}_{i=1}^\ell$ halmazra. A háromszögháló feletti regisztrációs algoritmus teljes komplexitása

$$\mathcal{O}(M^3|O_\Delta| + ((M^3 + \ell)|T_\Delta| + V(T_\Delta)N)I)), \quad (3.24)$$

ahol N a transzformáció végrehajtásához szükséges lépések száma, I a *Levenberg-Marquardt* optimalizáló által elvégzett függvényhívások száma, $|O_\Delta|$ és $|T_\Delta|$ jelöli a O_Δ és T_Δ hálók háromszögeinek számát, $V(T_\Delta)$ jelöli a T_Δ háló csúcspontjainak a számát.

Az előző alfejezetben jeleztük, hogy a TPS modell igényel bizonyos peremfeltételeket a (3.2) egyenletből. Mivel a feltételek lineárisak a modell paramétereiben (azaz az egyenletrendszer ismeretleneiben) a probléma formalizálható egy korlátozott legkisebb négyzetes problémaként. Az általunk alkalmazott `levmar` könyvtár tartalmaz ilyen megvalósítást is.

A transzformáció denormalizálásának két lépése van. Az egyszerűbb jelölés érdekében, legyen

$$\hat{A}^* = \begin{pmatrix} \hat{a}_{11}^* & \hat{a}_{12}^* & \hat{a}_{13}^* & \hat{a}_{14}^* \\ \hat{a}_{21}^* & \hat{a}_{22}^* & \hat{a}_{23}^* & \hat{a}_{24}^* \\ \hat{a}_{31}^* & \hat{a}_{32}^* & \hat{a}_{33}^* & \hat{a}_{34}^* \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3.25)$$

ahol $\hat{a}_{ij}^*$ és $i = 1, \dots, 4, j = 1, \dots, 3$ a meghatározott, normalizált $\hat{\varsigma}^*$ transzformáció lineáris paraméterei. Ekkor a denormalizált $\hat{A}$ mátrix megkapható a következő formulával:

$$\hat{A} = N_o^{-1} \hat{A}^* N_t \quad (3.26)$$

ahol N_o^{-1} és N_t a normalizálás során alkalmazott mátrixok. A kontrolpontoktól vett távolság súlyoknál a *sablon* normalizáló transzformációját végre kell hajtani a kontrolponton

és az aktuális ponton, majd a súlyokat meg kell szorozni a *megfigyelést* normalizáló mátrix inverzében található megfelelő skálafaktorrall, azaz $\hat{w}_{ki} = \hat{w}_{ki}^*/o_i$, ahol $i = 1, \dots, 3$ és $k = 1, \dots, K$. Tehát a (3.1) egyenlet a következő alakú lesz

$$\hat{\varsigma}_i(\mathbf{x}) = \hat{a}_{i1}x_1 + \hat{a}_{i2}x_2 + \hat{a}_{i3}x_3 + \hat{a}_{i4} + \sum_{k=1}^K \hat{w}_{ki} \|N_t(c_k - \mathbf{x})\|, \quad (3.27)$$

ahol az $\hat{a}_{ij}$ paraméter az előbbi $\hat{A}$ mátrix megfelelő eleme lesz és $i = 1, \dots, 3$, $j = 1, \dots, 4$.

A két hálót illesztő algoritmus pszeudokódja megtalálható a 2. algoritmusban.

2. algoritmus. A háromszögháló alapú algoritmus pszeudokódja

Bemenet: *sablon* és *megfigyelés* objektumok háromszöghálói

Kimenet: Az illesztő transzformáció, $\hat{\varsigma}$ paraméterei

- 1: Válasszunk egy O_{max} értéket úgy, hogy $\ell > k$ nemlineáris $\{\omega_i\}_{i=1}^{\ell}$ függvényünk legyen és minden ω_i függvényre számítsuk ki a normalizáló konstansokat az (1.14) formulák segítségével
 - 2: Normalizáljuk a csúcspontok koordinátáit az N_o és N_t transzformációkkal a $[-0,5; 0,5]$ tartományba.
 - 3: Konstruáljuk meg a (3.23) egyenletrendszert.
 - 4: Oldjuk meg az egyenletrendszert korlátozott legkisebb négyzetes értelemben a *Levenberg-Marquardt* algoritmussal. Inicializáljuk az algoritmust az identikus transzformációval.
 - 5: Denormalizáljuk a megkapott $\hat{\varsigma}^*$ transzformációt a megadott eljárás segítségével, ami megadja a keresett transzformációt
-

3.2. Teszteredmények

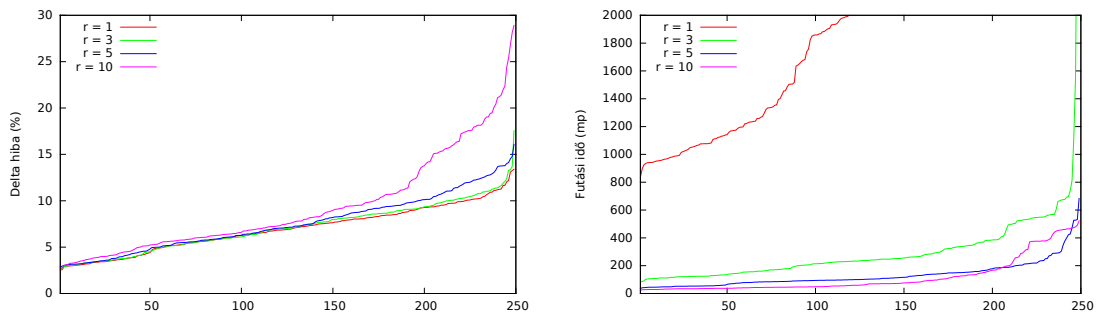
A tesztjeink során egy egyszerű TPS modellt alkalmaztunk 64 kontrolponttal. A kontrolpontok egyenletes rácsot alkotva lettek elhelyezve, dimenzióként 4 darab. Így a TPS modellnek összesen 204 paramétere lett, ami jelentősen több, mint a korábbi polinomiális teszteknel. Egy 220 egyenletből álló egyenletrendszert generáltunk a $\omega_i(\mathbf{x}) = x_1^{m_i} x_2^{n_i} x_3^{o_i}$ halmazból, ahol $(m_i, n_i, o_i)_{i=1}^{220} = \{(a, b, c) \mid a + b + c = O, O = 1, \dots, 9\}$.

3.2.1. Szintetikus tesztek

A teszteléshez egy 750 képből álló szintetikus halmazt készítettünk véletlenszerű TPS transzformációkkal. A generáláshoz 16, 32 és 64 kontrolpontot használtunk, egyenletes

rácsként elhelyezve. A transzformációk generálása során a kontrolpontokat véletlenül választott vektorokkal eltoltuk, majd az eredeti kontrolpontokra és az így kapott képeikre felírtuk az interpoláló TPS transzformációt (hasonlóan a klasszikus pontmegfeleltetési eljárásokhoz). A eltolások mértékét a $[-0,2; 0,2]$ halmazra korlátoztuk. A topológia megőrzését itt is kikényszerítettük a csak pozitív Jacobi determináns értékkel rendelkező transzformációk elfogadásával. A generált képekre példákat adunk a 3.3 ábrán.

Az első kísérletben az eljárás hatékonyságát vizsgáltuk a háromszögháló felbontásának függvényében. Ehhez az előzőleg generált voxel képeket, a pontfelhő alapú módszerekkel történő összehasonlítás során is alkalmazott, háromszögháló generáló eljárás segítségével különböző felbontású hálókra alakítottuk, majd inputként átadtuk a regisztrációs algoritmusnak. Az átalakítás során a felbontás kontrollálható az egy háromszöget magába foglaló Delaunay gömb sugarával, amit a teszthez a következő halmazból választottunk: $r \in \{1, 3, 5, 10\}$ (az alacsonyabb érték nagyobb felbontású hálót eredményez). Az eredmények láthatók a 3.2 ábrán. Jól látszik, hogy a felbontás nagyban befolyásolja az eljárás futási idejét. Ugyanakkor az alacsonyabb felbontás nem feltétlenül hoz nagyobb mértékű romlást a regisztrációs eredményekben. A tesztek alapján kijelenthető, hogy a legjobb eredmény/futási idő hányadossal az $r = 5$ eset rendelkezik, ezért a további tesztekben ezzel dolgoztunk.

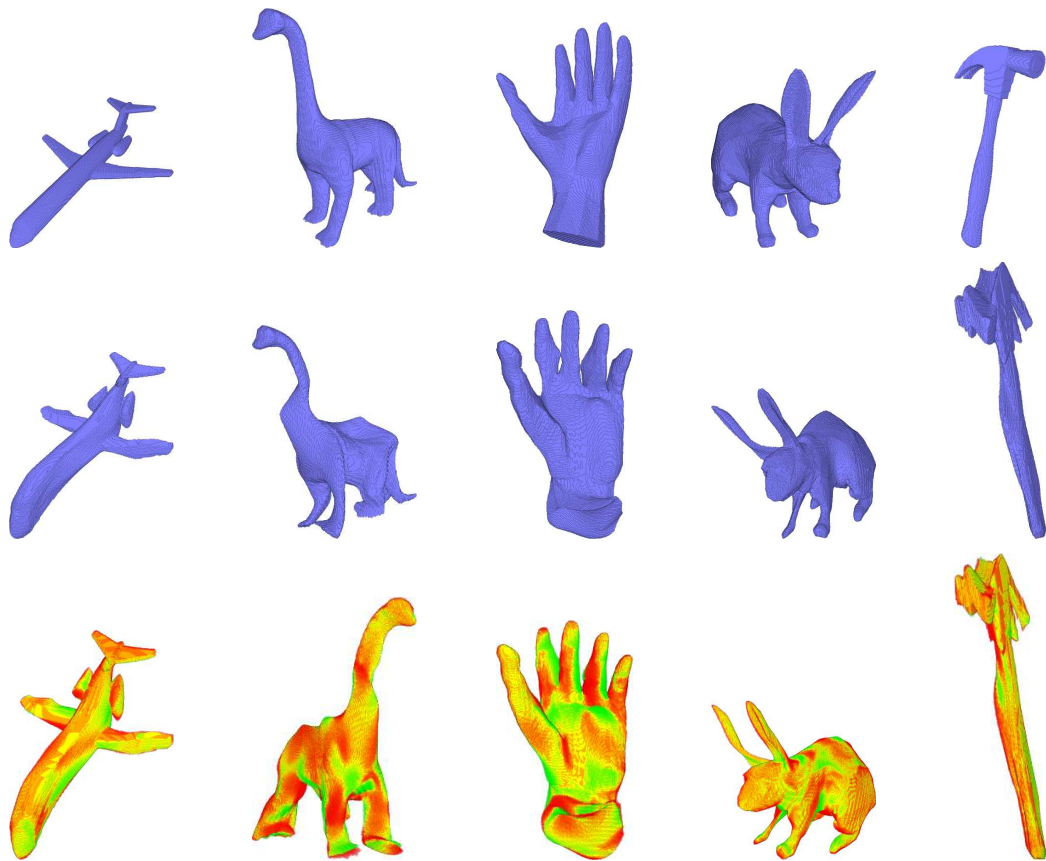


3.2. ábra. A háromszögháló alapú eljárás tesztelése a háló felbontásának függvényében.

Módszer	Időigény (perc)			$\delta(\%)$		
	m	μ	σ	m	μ	σ
Háromszögháló alapú	1,7	1,38	1,22	6,44	6,12	2,52
Voxel alapú	31,42	59,77	67,74	7,19	7,05	2,3

3.1. táblázat. A két módszer összehasonlítása a 750 darabos teszhalmazon (m – medián, μ – átlag és σ – szórás).

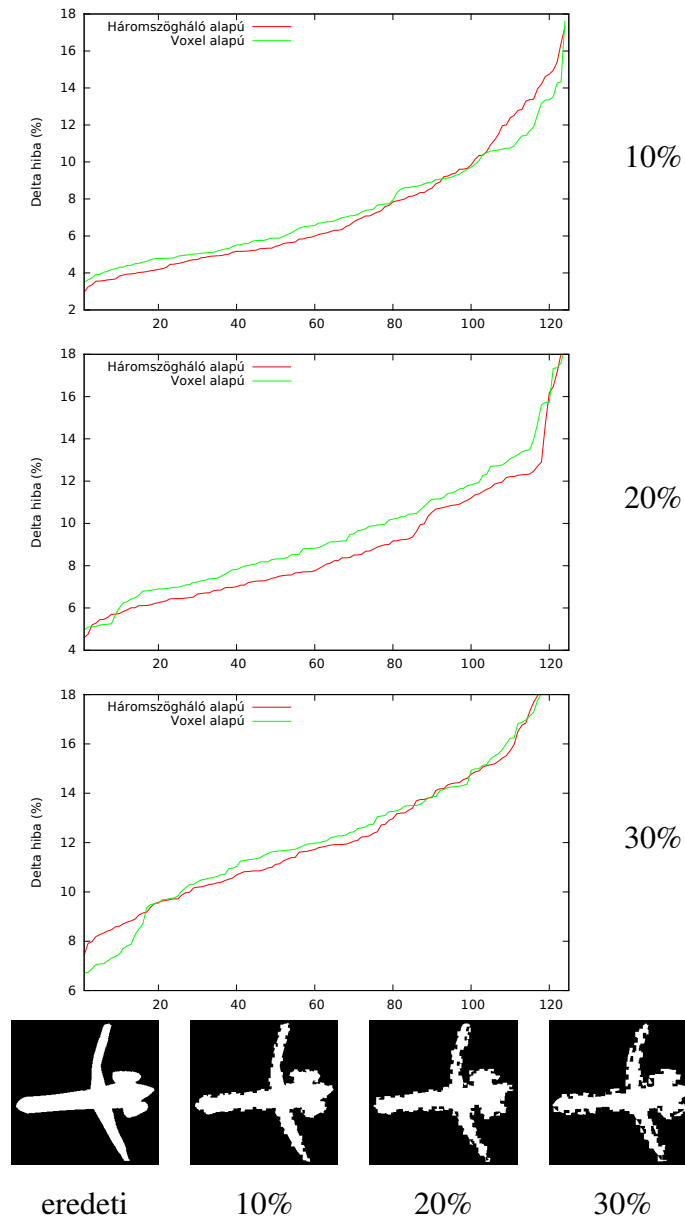
A hagyományos megoldással való összehasonlítás érdekében implementáltuk a voxel alapú eljárás TPS változatát is. A két módszer összehasonlítása megtalálható a 3.1



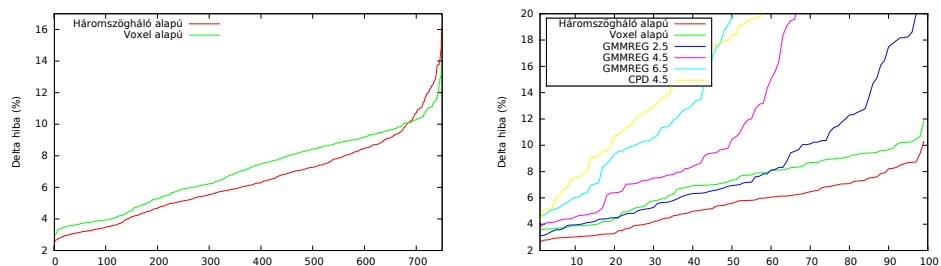
3.3. ábra. Eredmények a tesztalmazból. Az első két sor tartalmazza a *sablon* és a *megfigyelés* képeket. Az utolsó sorban az eredmények láthatók, a *megfigyelést* és a *regisztrált* képeket egymásra vetítettük úgy, hogy a közös területet sárgával, különböző területeket zölddel és pirossal színeztük.

táblázatban és a 3.5 ábrán. A teszt során a voxelek számának maximuma 2 524 180, a minimuma 135 433 volt, míg a háromszögháló felbontás 1 996 és 8 646 között volt. A táblázatból is látszik, hogy a háromszögrács alapú eljárás időigénye megközelítőleg huszad része a voxel alapú eljárásénak. Ami viszont érdekesebb, hogy a háló alapú eljárás alacsonyabb hibaértékeket produkált a tesztalmazon. Erre magyarázatul szolgálhat a mód, ahogyan a két eljárás a folytonos integrálokat közelíti. A voxel alapú eljárásnál a közelítési hiba adódik a reprezentáció és a valós objektum közötti eltérésből, valamint abból, hogy Jacobi determináns konstansnak tekintjük egyetlen voxelre nézve. Viszont a háromszögháló alapú eljárás nem igényli a Jacobi determináns számítását, emiatt csak a reprezentációs eltérésből adódik a közelítés hibája.

A TPS tesztnél alkalmazott felszíni hibák a teljes kép 10%, 20% és 30%-át tették ki (példák találhatók a zajjal terhelt képekre a 3.4 ábrán). A diagramok alapján az eljárás robusztusnak mondható, ha a szegmentálási hiba nem haladja meg a teljes képméret 20%-át.



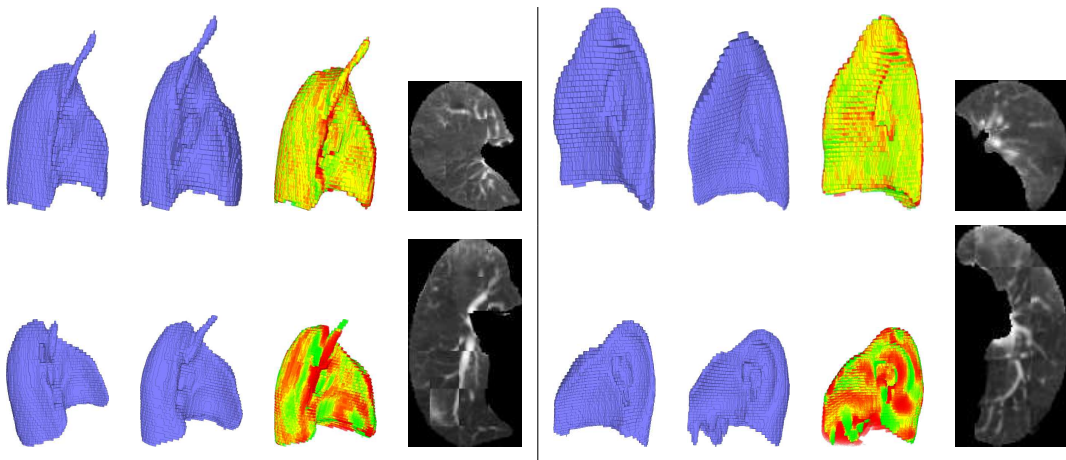
3.4. ábra. A robosztusság teszt eredményei változó erősségű felszíni hibákkal szemben. Az alsó ábrákon látható a különböző hibamértékekhez tartozó képszeletek.



3.5. ábra. A voxel és háromszögháló alapú eljárás összehasonlítása. A jobb oldali ábrán, a pontfelhő alapú eljárásokkal (GMMREG [10] and CPD [16]) való összehasonlítás.

Végül ezt a módszert is összehasonlítottuk a pontfelhő alapú eljárásokkal. A teszteléshez a $\{2,5; 4,5; 6,5\}$ halmazból választottuk ki a háló finomságát befolyásoló paramétert, ezzel a pontfelhő algoritmusok futási ideje megközelítőleg egy szintre került a voxel alapú algoritmuséval. A háló alapú algoritmushoz továbbra is az $r = 5$ értéket használtuk, tehát a javasolt algoritmus kevésbé részletes háromszögrácssal dolgozott. Az eredmények megtalálhatók a 3.5 ábrán. Ez alapján a háromszögrács alapú eljárás egyértelműen hatékonyabb bármely vizsgált algoritmusnál, időigénye pedig megközelítőleg a 6,5 felbontású GMMREG futtatással egyezik meg.

3.2.2. Valós képek



3.6. ábra. A tüdő képek illesztése TPS modellel. Az első sor a legjobb eredményeket, az utolsó a legrosszabb eredményeket tartalmazza. A szegmentált képeket a Mediso Ltd. *InterView Fusion* programja készítette.

A tüdő CT képeknél is teszteltük háromszögháló alapú TPS modellt. A szegmentált képeket a $r = 5$ paraméterrel konvertáltuk át a megfelelő háromszöghálókra. Ígéretes eredményeket értünk el, a delta hiba mediánja 5,41% volt, az átlag 5,83% és a szórás 2,09%.

A polinomiális modellel ellentétben a TPS könnyen adhat olyan illesztést, ami az alakzat felszínét jól illeszti, viszont a belső struktúráját nagyon eltorzítja. Ebből következően a szürkeárnyaltos képekre nem adhat jó illesztést. Ahhoz, hogy ezt javítsuk, regularizálni kell a modellt, így az egyenletrendszer megoldása mellett egy olyan megoldást keresünk, ami minimalizálja a transzformáció *görbületi energiáját*. Ehhez az (1.2) formulával megadott integrált hozzávesszük az egyenletrendszer algebrai hibájához egy alkalmazásfüggő λ paraméterrel súlyozva (a tesztjeink során a λ értéke 10^{-8} volt). További előnye a *görbületi energia* minimalizálásának, hogy az eljárásunkban alkalmazva

csak diffeomorf eredményeket kaptunk. A minimalizálás nélkül ez nem minden esetben fordult elő. A tūdő tesztek eredményeire a 3.6 ábrán adunk példákat.

Konklúzió

A dolgozatban bemutatásra került egy háromdimenziós alakzatok nemlineáris transzformációinak meghatározására alkalmas keretrendszer. Az eljárás alapötlete egy nemlineáris egyenletrendszer konstruálása, melynek ismeretlenjei megadják a transzformációt közelítő modell paramétereit. Az eljárás alapját egy 2D keretrendszer adja a voxel alapú képek esetén [6]. A háromszögháló alapú alakzatok bevezetése egy teljesen új és hatékony irány, amivel kezelhető a 3D kiterjesztés során felmerülő időbeli teljesítményromlás. Itt megjegyeznénk, hogy a háromszöghálóval elért eredmények alkalmazhatók 2D-ben is, ha a síkbeli alakzatok határát poligonokkal közelítjük, majd a poligonok csúcsaiból háromszögrácsot konstruálva levezetjük a formulákat a 3D esethez hasonlóan.

A tesztjeink kimutatták, hogy a módszer hatékonyan alkalmazható a klasszikus polinomiális és vékony fémlemez spline modellek esetén. Mindkét modellenél nagy teszt-halmazon igazoltuk a rájuk épülő módszerek hatékonyságát és robusztusságát a felszíni hibákkal szemben. Végül mindkét eljárást összehasonlítottuk két pontfelhő alapú regisztrációs módszerrel [10, 16], ahol a dolgozatban javasolt eljárások minden esetben jobb eredményt értek el, mint a korábbi megoldások.

Végül egy lehetséges alkalmazást prezentáltunk, ahol szegmentált tüdő CT képek illesztése a feladat. A feladatot mindkét modell alacsony hibaértékkel oldotta meg. Egy konkrét orvosi alkalmazáshoz azonban javasolt a modell további finomítása. A polinomiális modellenél szükséges lehet a futási idő redukálása és a megfelelő polinomiális függvény meghatározása, a TPS modell esetében a kontrolpontok optimális elhelyezése és a *görbületi energia* kezelésének finomhangolása lehetne a következő lépés egy konkrét alkalmazáshoz.

Irodalomjegyzék

- [1] G. C. Best. Helpful formulas for integrating polynomials in three dimensions. *Mathematics of Computation*, 18(86):310–312, Apr. 1964.
- [2] F. L. Bookstein. Principal warps: Thin-Plate Splines and the Decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, 1989.
- [3] G. Christensen. Consistent linear-elastic transformations for image matching. In *Proceedings of Information Processing in Medical Imaging*, pages 224–237. Springer-Verlag, 1999.
- [4] C. Domokos and Z. Kato. Affine puzzle: Realigining deformed object fragments without correspondences. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Proceedings of European Conference on Computer Vision*, volume 6312 of *Lecture Notes in Computer Science*, pages 777–790, Crete, Greece, Sept. 2010. Springer.
- [5] C. Domokos and Z. Kato. Parametric estimation of affine deformations of planar shapes. *Pattern Recognition*, 43(3):569–578, Mar. 2010.
- [6] C. Domokos, J. Nemeth, and Z. Kato. Nonlinear Shape Registration without Correspondences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):943–958, 2012.
- [7] Q. Fang and D. Boas. Tetrahedral mesh generation from volumetric binary and gray-scale images. In *International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1142–1145, Boston, Massachusetts, USA, June 2009. IEEE, IEEE.
- [8] H. Guo, A. Rangarajan, and S. Joshi. Diffeomorphic point matching. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, pages 205–219. Springer US, 2006.
- [9] M. Holden. A review of geometric transformations for nonrigid body registration. *IEEE Trans. Med. Imaging*, 27(1):111–128, 2008.

- [10] B. Jian and B. C. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, Aug. 2011.
- [11] P. Koehl. Fast Recursive Computation of 3D Geometric Moments from Surface Meshes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(Pre-Prints), 2012.
- [12] L. Leindler. *Analízis*. SZTE Bolyai Intézet, 2004.
- [13] M. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. [web page] <http://www.ics.forth.gr/~lourakis/levmar/>, Jul. 2004. [Accessed on 14 Jun. 2012.].
- [14] F. Michel, M. M. Bronstein, A. M. Bronstein, and N. Paragios. Boosted metric learning for 3D multi-modal deformable registration. In *International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1209–1214, Chicago, Illinois, USA, Mar. 2011. IEEE, IEEE.
- [15] J. Mitra, Z. Kato, R. Marti, A. Oliver, X. Llado, D. Sidibe, S. Ghose, J. C. Vilanova, J. Comet, and F. Meriaudeau. A spline-based non-linear diffeomorphism for multi-modal prostate registration. *Medical Image Analysis*, 16:1259–1279, aug 2012.
- [16] A. Myronenko, X. B. Song, and M. Á. Carreira-Perpiñán. Non-rigid point set registration: Coherent point drift. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Proceedings of the Annual Conference on Neural Information Processing Systems*, pages 1009–1016, Vancouver, British Columbia, Canada, Dec. 2006. MIT Press.
- [17] F. S. Nooruddin and G. Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9:191–205, 2003.
- [18] C. Papazov and D. Burschka. Deformable 3D shape registration based on local similarity transforms. *Computer Graphics Forum*, 30(5):1493–1502, Aug. 2011.
- [19] J. M. Pozo, M. C. Villa-Uriol, and A. F. Frangi. Efficient 3D Geometric and Zernike Moments Computation from Unstructured Surface Meshes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3):471–484, Mar. 2011.
- [20] K. Rohr, H. S. Stiehl, R. Sprengel, W. Beil, T. M. Buzug, J. Weese, and M. H. Kuhn. Point-based elastic registration of medical image data using approximating thin-plate splines. In *VBC*, pages 297–306, 1996.

- [21] R. Sagawa, K. Akasaka, Y. Yagi, H. Hamer, and L. Van Gool. Elastic convolved ICP for the registration of deformable objects. In *Proceedings of International Conference on Computer Vision*, pages 1558–1565, Kyoto, Japan, Oct. 2009. IEEE, IEEE.
- [22] A. Tanacs and Z. Kato. Fast linear registration of 3D objects segmented from medical images. In *Proceedings of International Conference on BioMedical Engineering and Informatics*, pages 299–303, Shanghai, China, Oct. 2011. IEEE.
- [23] A. Tanács, J. Lindblad, N. Sladoje, and Z. Kato. Estimation of linear deformations of 3d objects. In *Proceedings of International Conference on Image Processing*, pages 153–156, 2010.
- [24] L. Zagorchev and A. Goshtasby. A comparative study of transformation functions for nonrigid image registration. *Image Processing, IEEE Transactions on*, 15(3):529–538, march 2006.
- [25] B. Zitová and J. Flusser. Image registration methods: a survey. *Image Vision Comput.*, 21(11):977–1000, 2003.

Publikációs jegyzék

- [26] Zs. Sánta and Z. Kato. A Unifying Framework for Non-linear Registration of 3D Objects. In *Proceedings of IEEE International Conference on Cognitive Infocommunications*, Kosice, Slovakia, dec 2012. IEEE.
- [27] Zs. Sánta and Z. Kato. Elastic Registration of 3D Deformable Objects. In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*, Fremantle, Western Australia, dec 2012. IEEE.
- [28] Zs. Sánta and Z. Kato. Correspondence-Less Non-Rigid Registration of Triangular Surface Meshes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Portland, Oregon, jun 2013. Under review.