

Orvosi képfeldolgozási algoritmusok  
megvalósítása kiloprocesszoros  
architektúrákon



Tornai Gábor János  
*A Ph.D. Disszertáció tézisei*

Pázmány Péter Katolikus Egyetem  
Információs Technológiai és Bionikai Kar

Témavezető:  
Dr. Cserey György Ph.D  
Konzulens:  
Dr. Roska Tamás  
Az MTA rendes tagja

Budapest, 2014

# 1. Bevezetés

Az orvosi képanalízis hatékonyan segíti az orvosi munkát a diagnózis kialakításától a terápiás kezelések monitorozásáig. Képesek vagyunk szöveti anomáliát detektálni, esetenként számszerűsíteni tudjuk a különböző daganatos elváltozások valószínűségét. Az analízisek mögött ott találjuk a különböző algoritmusokat és a félvezető eszközöket, amelyek lehetővé teszik a szükséges sebességet és pontosságot. A legszemléletesebb példák közé tartoznak a monitorozott, minimális beavatkozású műtétek, ahol a mûtéti eszköz trajektóriáját folyamatosan hasonlítják a referenciához.

Az említett új félvezető eszközök az utóbbi években megjelent és nagy fejlődésnek indult sokprocesszoros rendszerek. A korábbi egymagos rendszereket felváltották a sokmagos rendszerek. Gondolhatunk sokmagos CPU-ra, a GPU-k magszámának drasztikus növekedésére, a szenzorprocesszor chipekre vagy akár FPGA-kra.

Mindegyik eszköz több alapvető tulajdonsággal rendelkezik a következőkből: nagy számítási teljesítmény, nagy sávszélesség, alacsony disszipáció, kis chip felület. Ahhoz, hogy hatékonyak ki lehessen használni az eszközök nyújtotta előnyöket és a kitűzött feladatokat megfelelően el lehessen végezni, új algoritmusok és új optimalizációs módszerek szükségesek. A disszertációmban két feladatra mutatok be megoldást, melyek a sokprocesszoros rendszerek előnyeit használják ki.

Az első feladatomban volt a digitálisan rekonstruált radiográf (DRR) renderelése, amely kulcsfontosságú nagyon sok képpel támogatott terápiás alkalmazásnál. A sok alkalmazás közül az operáció előtti 3D CT felvétel és az operáció közbeni 2D röntgen képek összekapcsolására fókuszálók [1]. Ezt az eljárást nevezik 2D-3D regisztrációnak. Maga a DRR egy virtuálisan generált röntgenkép. Egy DRR-pixel értéke a képzeletbeli forrás és a pixel közötti út mentén vett vonalintegrálja a CT-nek. Az irodalomból ismert, hogy a DRR-renderelés igényli a regisztráció idejének a legnagyobb részét [1, 2, 3], így kulcsfontosságú a renderelés idejének minimalizálása.

A második feladatomban volt a nívóhalmaz alapú algoritmusok gyorsítása, lévén számos alkalmazási területük van a geometriától kezdve a kristály növekedés modellezésen keresztül a gépi látásig [4, 5, 6]. A módszer lényege, hogy egy görbét, felületet, sokaságot vagy képet egy parciális differenciál egyenlettel (PDE) evolválunk és az eredményt a fejlődés egy pontján megkapjuk. A nívóhalmazokkal megoldható feladatoknak létezik olyan részhalmaza, ahol csak az evolúció végállapotára vagyunk kíváncsiak. Ilyen a szegmentáció, az alak modellezés és a detekció. Erre a

halmazra ad igen gyors megoldást a Shi-féle módszer [7], mely módszernek a lépésszámával és CNN-UM illetve GPU-rendszerekre való hatékony leképezésével foglalkoztam.

## 2. Módszerek

A DRR-renderelést Nvidia CUDA C környezetben valósítottam meg. Több optimalizációs szabályt állítottam fel és vizsgálati paramétert alkalmaztam, hogy a feladatot az adott típusú hardverre jól leképezsem. Több különböző generációhoz tartozó GPU-n mértem (8800 GT, 280 GTX, Tesla C2050, 570 GTX, 580 GTX), valamint két különböző driver és fordítóprogram esetében is végeztem méréseket (3.2-es fordító + 260.16.21-es driver, illetve 5.5-ös fordító és 331.67-es driver). Két adathalmazon dolgoztam. Egyik egy radiológiai fantomról (Radiology Support Devices, Newport Beach, CA, modell RS-330) készült  $512 \times 512 \times 72$  felbontású CT felvétel, a másik egy kutatók által annotált adatbázisból [8] egy sertésfejéről készült  $512 \times 512 \times 825$  felbontású CT felvétel. A fantom az emberi mellkast imitálva a röntgen-spektrumban hasonló elnyelési tényezővel rendelkezik, mint a csont, tüdő, ér, porc vagy egyéb belső szervek. Végeztem méréseket teljes és véletlenszerűen mintavételezett DRR esetében is. A renderelés paramétereit elsősorban a műtéti regisztrációnál releváns tartományban vizsgáltam, és állítottam be (célrégio mérete, mintavételezés sűrűsége).

A nívóhalmazokhoz kapcsolódó munka során a probléma megértéséhez szükséges volt a PDE-k családjából a hiperbolikus egyenletek, illetve a viszkózus limesz kapcsolódó részeinek elemi szintű ismerete. A viszkózus limeszt a következő nemlineáris PDE megoldásából kapjuk  $G(u)u_x + u_t = \epsilon u_{xx}$ , amennyiben  $\epsilon$  tart 0-hoz. A nívóhalmazok elméletének alkalmazása nélkülözhetetlen objektum-segmentációnál és görbék változásának követését igénylő feladatoknál. Ezen felül a diszkrét topológia és a konvex halmazok elméletének elemeit használtam fel, és azok segítségével konstruáltam a bizonyítást a kimondott állításokhoz. A futtatási kísérleteket Eye-RIS v1.3 látórendszeren és Nvidia 780 GTX GPU-n végeztem.

### 3. Új tudományos eredmények összefoglalása

1. Tézis 1. *Megalkottam egy szabályrendszert (1)-(4), mellyel a 2D-3D regisztrációs eljárások lassúságáért felelős digitálisan rekonstruált radiográf (DRR) renderelés hatékonyan számolható Nvidia GPU-n. A DRR-renderelés során véletlen irányú vonalintegrálok számolásához alkalmaztam a megadott szabályrendszert és szisztematikusan végigvizsgáltam a blokkméret paramétert az elméletileg lehetséges teljes tartományban. Ezek alapján az elméleti ajánlásokkal ellentétben a hatékony számításhoz a nagyon alacsony blokkonkénti szálszám (8-16) szükséges. Így a korábbi módszerekkel ellentétben, a 2D-3D regisztrációs eljárás a műtéti körülményeknek megfelelően valós időben elvégezhető, alkalmazási területtől függően 0.5-10 fps sebességgel. Megmutattam, hogy a DRR-renderelés 0.2-2.2 ms alatt elvégezhető egy háti gerinccsigolyát teljesen tartalmazó célrégió esetén ( $16 \times 9 \text{ cm}^2$ ,  $400 \times 225$  felbontás).*

- 1. a lassú, feltételes ugrás 'parallel thread execution' (PTX) utasításra forduló 'if else' kifejezések három operandusú feltételes utasítással helyettesítendőek, (ha lehetséges) melyek sokkal gyorsabb select' PTX utasításra fordulnak,*
- 2. azokat az adatokat, melyekre gyakran, lokálisan, nem-coalesced' módon olvasunk, és nem írunk, a textúra memóriába kell helyezni,*
- 3. az osztás művelet kerülendő, ha mégis szükséges, (akkor, ha a pontosság lehetővé teszi) a gyorsabb változatot érdemes használni (div.approx, div.full a div.rnd helyett),*
- 4. amennyiben egy osztót többször felhasználunk, akkor reciprokot kell számolni, és azt követően azzal szorozni.*

Elsőként közöltem pontos méréseket GPU-n futó, véletlenszerűen mintavételezett DRR-renderelésről [9]. A bemutatott optimalizációs szabályok közül az első kettő hatékonyságát az 1. táblázat szemlélteti. A harmadik és negyedik szabály hatását és a blokkméret-függőséget az 1. ábra mutatja be.

A feltételes utasítás optimalizálása átlagban 6-11% gyorsulást okozott a Tesla C2050 GPU-n, míg ez az arány az 570 GTX GPU-n 6-13%. A lineáris memória használata a textúra memória helyett 1.7-2.4-szeres lassulást okozott.

1. táblázat. Az első két optimalizációs szabály hiányának hatása a futási időre. Az optimalizált futási idők a  $t_{opt}$  oszlopokban láthatóak. A feltételes utasítás optimalizálásának hiánya 6-13% lassulást ( $t_{branch}$  oszlopok), míg a lineáris memória használata átlagosan kétszeres lassulást okozott ( $t_{linear}$  oszlopok). Az idők  $\mu s$ -ben értendők.

szálszám	Tesla c2050			570 GTX		
	$t_{opt}$	$t_{branch}$	$t_{linear}$	$t_{opt}$	$t_{branch}$	$t_{linear}$
1024	234	258	553	181	206	408
1536	319	339	639	263	295	462
2048	466	502	1094	358	403	656
3072	648	689	1275	572	617	1101
4096	969	1082	1935	693	742	1310
full DRR	2666	2763	5278	2259	2375	5221

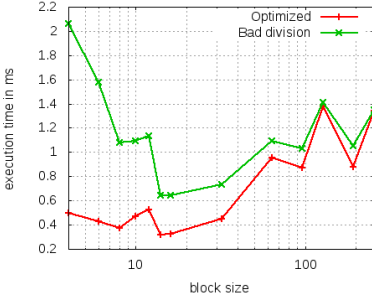
Az optimális blokkméret az optimalizált kernel esetén mindig a blokkonkénti 8-16 szál volt. Ezt a tulajdonságot egy korábbi fordító, illesztőprogram kombinációval is megvizsgáltam négy különböző csúcs GPU-n (8800 GT, 280 GTX, Tesla C2050, 580 GTX). A szabályok szerint optimalizált kernel futási ideje ebben a régebbi szoftverkörnyezetben is hasonló karakterisztikát mutatott.

A tézishez tartozó publikáció: [I]. A tézis a disszertáció második fejezetében kerül kifejtésre.

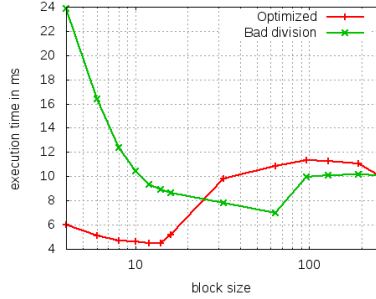
2. Téziscsoport A Shi-féle nívóhalmaz alapú algoritmus [7] szükséges lépésszámára adok korlátot, amely csak az evolúció kezdeti feltételétől függ. Olyan kezdeti feltétel-családot mutatok be, amellyel ezt az elméleti korlátot lehet flexibilisen és effektíven csökkenteni. Ráadásul sokprocesszoros rendszereken ebből a kezdeti feltétel családból indítva az algoritmust a konvergáláshoz szükséges idő töredékére csökken.

*2.1 Tézis: Két új elméleti korlátot állítottam fel (általános és konvex esetre) a Shi-féle nívóhalmaz alapú algoritmus maximálisan szükséges lépésszámának meghatározásához, amely korlátok csak a kezdeti feltételtől függenek. A megadott korlátokat elméleti úton bizonyítottam és ennek eredményeit kísérletekkel is megerősítettem. Az eredmények a 2.2-es al-tézisben felhasználásra kerülnek.*

Legyen adott a diszkrét rács ( $\mathbb{Z}^n$ ) egy részhalmaza  $D$ .  $D$ -n belül útnak nevezzük a pontok egy halmazát  $\mathbf{x}_l (l = 0, 1, \dots, L) \in D$ , ha  $\mathbf{x}_l \in N(\mathbf{x}_{l+1})$  és  $\mathbf{x} = \mathbf{x}_0$  valamint  $\mathbf{y} = \mathbf{x}_L$ , ahol  $N(\cdot)$  megadja a pont szomszédainak



(a) 1536 szál



(b) 20480 szál

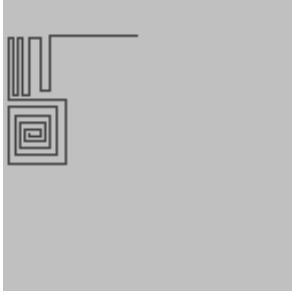
1. ábra. A harmadik és a negyedik szabály együttes hatása a futási időre a blokkméret függvényében.

halmazát. A pontok egy halmaza egy *összefüggő régiót* definiál, ha minden pontjából minden pontjába vezet út úgy, hogy csak a halmazbeli pontokat használjuk fel. *Minimum útnak* hívjuk azokat az utakat, amelyeknél a két végpont között nincsen rövidebb út. Ez a minimum út nem egyértelmű, és függ a választott szomszédságtól. Egy összefüggő régió átmérője  $B$  egy olyan leghosszabb minimum út, melynek a két végpontja az összefüggő régióban található.

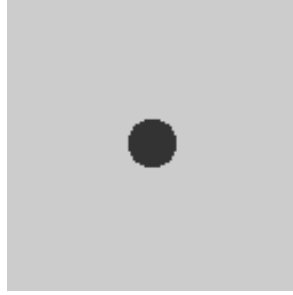
**1. Állítás** (általános korlát). *Álljon az igazi objektum régió  $\Omega^*$   $P$  darab,  $\Omega_p^*$  (ahol  $p = 1 \dots P$ ) összefüggő régióból és az igazi háttér régió  $\Gamma^*$   $Q$  darab,  $\Gamma_q^*$  (ahol  $q = 1 \dots Q$ ) összefüggő régióból. Legyen  $F > 0$   $\Omega^*$ -ban and  $F < 0$   $\Gamma^*$ -ban. A kiinduló állapotot úgy válasszuk meg, hogy  $\Omega = \cup_i \Omega_i$ ,  $\Gamma = \cup_j \Gamma_j$  és  $\Omega_p^* \cap \Omega \neq \emptyset$ ,  $\forall p = 1 \dots P$  és  $\Gamma \cap \Gamma_q^* \neq \emptyset$ ,  $\forall q = 1 \dots Q$ . Ekkor a Shi-féle algoritmus konvergál  $\Omega^*$ -hoz legfeljebb  $N_{\text{iteráció}}$  számú iteráció alatt, ahol  $N_{\text{it}} \leq \max(\max_i(|\Omega_i|), \max_j(|\Gamma_j|))$  és  $|\cdot|$  a régió elemszámát jelöli.*

**2. Állítás** (konvex korlát). *Tegyük fel, hogy  $\Omega^*$ ,  $\Gamma^*$ ,  $F$  és a kiinduló állapot rendelkezik az előbbi tételben leírt tulajdonságokkal, továbbá vagy  $\Omega^*$  vagy  $\Gamma^*$  legyen (diszkrét értelemben) konvex. Ekkor a következő korlát lép életbe a maximálisan szükséges iterációra:  $N_{\text{iteráció}} \leq \max(\max_i(B_{\Omega_i}), \max_j(B_{\Gamma_j}))$ .*

A 2. ábrán két példa objektumot látunk. Míg a 2(a) ábrán egy konkáv objektum látható, melynek a teljes „felfedezéséhez” legrosszabb esetben a pixelszámával arányos iterációra van szükség, addig a 2(b) ábrán egy



(a) konkáv objektum



(b) konvex objektum

2. ábra. A lehetséges objektumok két szélsősége. Amennyiben a kezdeti feltétel pusztán egy pixelnyit metsz bele mindkét objektumba, (b) a körhöz annak pixelben számított átmérőjénél nem szükséges több iteráció, ezzel szemben (a) a labirintusnál ez a mennyiség a konkáv objektum pixeleinek a száma.

2. táblázat. Az állítások bemutatása két  $128 \times 128$  felbontású kép (2(a)-(b)) esetében. A kezdeti feltétel  $n \times n$ -es sakktábla szerű minta volt.

görbeszám ( $n^2$ )	1	$2^2$	$4^2$	$8^2$	$16^2$	$24^2$	$32^2$	$64^2$
általános korlát	$64^2$	$32^2$	256	64	16	9	4	1
konvex korlát	127	63	31	15	7	5	3	1
2(a) ábra $N_{it}$	145	68	18	7	6	3	3	1
2(b) ábra $N_{it}$	26	16	9	6	4	3	3	1

konvex objektum látható, amelyhez a pixelben mért átmérőjével arányos lépésszámmal konvergálunk legrosszabb esetben.

A 2. táblázat egy példán keresztül mutatja be, a kezdeti feltétel korlátokra gyakorolt hatását. A képméret  $128^2$  pixel, a kezdeti feltétel konfiguráció egy sakktábla szerű minta volt, melyben az oldalankénti négyzetek számát ( $n$ ) változtattam az első sorban levő értékek szerint. A második és a harmadik sorban rendre az adott konfigurációhoz tartozó általános és konvex korlát látható. Az utolsó két sorban a 2. ábrán látható két tesztképhez tartozó szükséges lépésszámot tüntettem fel.

*2.2 Tézis: Igazoltam, hogy az 2.1 Tézisben megfogalmazott korlátokat minimalizáló kezdeti feltétel típusból indított Shi-féle evolúció hatékonyan leképezhető sokprocesszoros architektúrákra, melyet két rendszeren CNN-UM-en és GPU-n is megvalósítottam, az eredmények az elméletet*

*igazolták.*

A 2.1 Tézisből következik, hogy minél kisebb az összefüggő régiók mérete a kezdeti feltételben, annál kevesebb iteráció szükséges a konvergenciához. Többek között azért nem használták ezt a kezdeti feltétel típust, mivel így egy  $N \times M$  felbontású kép esetén  $O(N \times M)$ -el arányos mennyiségű pixelt kell feldolgozni egy iteráció során, hiszen a sok kis görbe majdnem teljesen befedi az egész képet. Ezzel szemben az egyetlen görbéből induló evolúcióknak a görbe hosszával arányos számú pixelt kell feldolgozniuk, ami  $O(N + M)$ -el felülről becsülhető. Fel kell hívnom a figyelmet arra, hogy amennyiben egy egyetlen görbéből álló kezdeti feltétel „távol van” a valós objektumtól, akkor a feldolgozandó pixelek száma  $O(k(N + M))$ -re változik, ahol  $k \sim \max(N, M)$ , ami  $O(N \times M)$ . Mivel a kezdeti feltételek szegmentálás esetén többnyire „távol vannak” a valós objektumtól, ezért a különböző típusú kezdeti feltételek hasonló mennyiségű műveletet igényelnek aszimptotikusan.

A 2.1 Tézisből világosan következik, hogy az egymás mellett sűrűn elhelyezett, kis átmérőjű görbék a megfogalmazott állításoknak megfelelően alacsonyan tartják az elméletileg lehetséges maximális lépésszámot. Az Eye-RIS rendszeren egy iteráció mérete független volt a kezdeti feltétel típusától, míg a GPU-n egy enyhe hullámnázás volt tapasztalható a lépésszám drasztikus csökkenésével egyidejűleg.

A CNN-UM-re leképezett algoritmust Eye-RIS 1.3-as rendszeren validáltam meg. A megvalósítás egyszerű templéteket használ, egy iteráció  $400 - 440 \mu s$ -ot vesz igénybe, ebből a tényleges számolás  $60 - 70 \mu s$ , a többi memóriatranszfer, amely abból fakad, hogy a Q-Eye chip-en rendelkezésre álló lokális logikai memóriánál többre van szüksége az algoritmusnak, így a látórendszer fő memóriájából kell áttölteni az adatot, mely egy másik nyomtatott lapon van.

A GPU-ra leképezett algoritmus futási ideje a 3. táblázatban látható. Megfigyelhető, hogy a javasolt kezdeti feltételből indított evolúciók minden esetben sokkal jobban teljesítenek, mint a hagyományos kezdeti feltételből indított társaik. Extrém esetben 24-szeres gyorsulás figyelhető meg a javasolt kezdeti feltétel javára ( $2,048 \times 2,048$  kép esetén  $210 \cdot 560$  vs.  $7 \cdot 684$ ).

Így látható, hogy mind a CNN-UM-en, mind pedig a GPU-n jelentős gyorsulás érhető el a Shi-módszer esetében, amennyiben a javasolt kezdeti feltételeket használjuk.

A téziscsoporthoz tartozó publikációk: [II, III, IV]. A tézis a disszertáció harmadik fejezetében kerül részletes bemutatásra és kifejtésre.



3. táblázat. Futási idő összehasonlítása GPU-ra leképezett NVIDIA GTX 780 GPU-n futó Shi algoritmusnak egy szálon futó eredeti Shi-algoritmussal Intel core i7 CPU-n. A gyorsulás az egy iterációra vonatkozó adat.

képméret	kezdeti feltétel	$t_{\text{GPU}} (\mu\text{s})$	$t_{\text{CPU}} (\mu\text{s})$	$N_{\text{it}}$	gyorsulás
256 × 256	1 × 1	129	1,610	32	12.5
256 × 256	2 × 2	126	2,242	59	17
256 × 256	8 × 8	140	3,164	20	22
256 × 256	32 × 32	143	8,874	8	62
512 × 512	1 × 1	317	3,190	64	10
512 × 512	4 × 4	167	8,724	40	52
512 × 512	16 × 16	157	12,897	25	82
512 × 512	64 × 64	123	16,246	18	132
1,024 × 1,024	1 × 1	534	6,431	129	12
1,024 × 1,024	8 × 8	548	27,461	55	50
1,024 × 1,024	32 × 32	590	43,739	32	74
1,024 × 1,024	128 × 128	490	84,078	12	171
2,048 × 2,048	1 × 1	560	14,972	210	26
2,048 × 2,048	16 × 16	703	79,920	79	113
2,048 × 2,048	64 × 64	830	198,980	28	239
2,048 × 2,048	256 × 256	684	327,541	7	478

Az idők 100 konvergált futtatás átlagát mutatják.

## 4. Alkalmazási területek

Egyrésztől megmutattam, hogy lehetséges valós időben (0.5-10 fps sebességgel!) elvégezni műtéti vagy egyéb terápiás körülmények között a 2D-3D regisztrációt. Ez nagyon fontos, és rendkívül nagy jelentősége lehet az elkövetkező évtized műtéti technológiájának fejlődésében. Másrésztől a problémát a GE Healthcare szakértőinek folyamatos konzultációja mellett vizsgáltuk és a technikai eredményeket és anyagokat a francia kutatás-fejlesztési csoportnak továbbítottam.

A második téziscsoport állításait gyorsabb szegmentáláshoz, detekcióhoz lehet felhasználni. Ezen feladatok felhasználási köre nagyon széles, ezért ezt itt most minden részletre kiterjedően nem tárgyalom. Kiemelem az orvosi képek analízisét, mert ezen a területen több kísérletet végeztem. Nagy jelentősége van annak is, hogy az algoritmus leképezésén túl olyan kezdeti feltételt használtam, amelyet eddig nem vettek számításba. Emellett a 2.1 Tézis elméleti eredményei garanciát biztosítanak az időkorlátra, amelyek időkritikus feladatoknál jelenthetnek nagy előnyt.

## 5. Köszönetnyilvánítás

„*Mi mindnyájan az Ő teljességéből kaptunk kegyelmet kegyelemre halmozva.*” (Jn1,16)

Köszönöm *Roska Tamás*nak, és *Szolgay Péter*nek a doktori iskola korábbi és jelenlegi vezetőjének a bátorítását, tanácsait és igényességét, melyek nagy erőt adtak a munkám folytatásához. Nehéz pontosan megfogalmazni, hogyan és mennyire köszönöm *Cserey György*nek, témavezetőmnek a lassan 7 éve tartó közös, izgalmas munkát, amely átfogja az egyetemi tanulmányok végét és az egész doktori időszakomat. Odafigyelése, ötletei és inspirációja nélkül nem lett volna sikeres az elmúlt időszak. A témavezetői tevékenységen túl a bennünket (meg)érintő törvények megvitatására is mindig sikerült alkalmat találni.

A doktorandusz társaimnak és a már végzeteknek is nagy hálával tartozom a közös beszélgetésekért, a segítségükért, vagy épp a kritikus véleményükért: *Horváth András, Stubendek Attila, Gergely Domonkos, Radványi Miska, Fülöp Tamás, Rák Ádám, Zsedrovits Tam, Koller Miki, Józsa Csaba, Nemes Csaba, Borbély Bence, Tuza Zoli, Rudan Jani, Bihari Dóri, Horváth Anna, Gelencsér András*. Köszönöm a pénteki szemináriumokat, a focit és a jó beszélgetéseket az ebéd mellett.

Köszönöm *Vida Tivadarné Katinkának* a lelkes, szíves és alázatos munkáját, nagy rugalmasságát és a sok nevetést, mellyel szüntelenül segítette gördülékenyen az adminisztrációs kötelezettségeinket. A kar tanulmányi és gazdasági osztályának, a dékáni hivatalnak és az összes többi munkatársnak is köszönet a háttérben elvégzett sok sok munkáért.

Köszönöm *Édesanya, Édesapa, Piri, Tamás, Ildi, Tibi*; az egész családnak köszönöm. Mindent.

Niki, Magdus s Blanka, miattatok érdemes küzdeni, öröm hozzátok hazatérni.

## A szerző publikációi

[I] **G. J. Tornai**, G. Cserey, and I. Pappas, „Fast DRR generation for 2D to 3D registration on GPUs,” *Medical Physics*, vol. 39, no. 8, pp. 4795–4799, 2012.

[II] **G. J. Tornai** and G. Cserey, „Initial condition for efficient mapping of level set algorithms on many-core architectures,” *EURASIP Journal*

on *Advances in Signal Processing*, 2014:30.

[III] **G. J. Tornai**, G. Cserey, and A. Rák, „Spatial-Temporal level set algorithms on CNN-UM,” in *International Symposium on Nonlinear Theory and its Application, (NOLTA 2008)*, pp. 696–699, 2008.

[IV] **G. J. Tornai** and G. Cserey, „2D and 3D level-set algorithms on GPU,” in *Cellular Nanoscale Networks and Their Applications (CNNA), 2010 12th International Workshop on*, p. 1–5, 2010.

A. Gelencsér-Horváth, **G. J. Tornai**, A. Horváth and G. Cserey, „Fast, parallel implementation of particle filter on GPU,” *EURASIP Journal on Advances in Signal Processing*, 2013:148.

## Hivatkozások

- [1] P. Markelj, D. Tomazevic, B. Likar, and F. Pernus, „A review of 3D/2D registration methods for image-guided interventions,” *Medical Image Analysis*, Mar. 2010.
- [2] G. P. Penney, P. G. Batchelor, D. L. Hill, D. J. Hawkes, and J. Weese, „Validation of a two-to three-dimensional registration algorithm for aligning preoperative CT images and intraoperative fluoroscopy images,” *Medical physics*, vol. 28, p. 1024, 2001.
- [3] J. Wu, M. Kim, J. Peters, H. Chung, and S. S. Samant, „Evaluation of similarity measures for use in the intensity-based rigid 2D-3D registration for patient positioning in radiotherapy,” *Medical Physics*, vol. 36, no. 12, p. 5391, 2009.
- [4] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. No. 3 in Cambridge monographs on applied and computational mathematics, New York: Cambridge Univ Pr, 2nd edition ed., 1999.
- [5] G. Sapiro, *Geometric partial differential equations and image analysis*. New York: Cambridge Univ Pr, 1st edition ed., 2001.
- [6] V. Caselles, R. Kimmel, and G. Sapiro, „Geodesic active contours,” *International Journal of Computer Vision*, vol. 22, pp. 61–79, Feb. 1997.

- [7] Y. Shi and W. Karl, „A Real-Time algorithm for the approximation of Level-Set-Based curve evolution,” *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 645–656, 2008.
- [8] S. A. Pawiro, P. Markelj, F. Pernus, C. Gendrin, M. Figl, C. Weber, F. Kainberger, I. Nobauer-Huhmann, H. Bergmeister, M. Stock, D. Georg, H. Bergmann, and W. Birkfellner, „Validation for 2D/3D registration i: A new gold standard data set,” *Medical Physics*, vol. 38, no. 3, p. 1481, 2011.
- [9] G. J. Tornai, G. Cserey, and I. Pappas, „Fast DRR generation for 2D to 3D registration on GPUs,” *Medical Physics*, vol. 39, no. 8, pp. 4795–4799, 2012.