

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## INTERAKTIVNÍ SEGMENTACE MEDICÍNSKÝCH OBRAZOVÝCH DAT

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

RADIM KŘÍŽ

BRNO 2010



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **INTERAKTIVNÍ SEGMENTACE MEDICÍNSKÝCH OBRAZOVÝCH DAT**

INTERACTIVE MEDICAL IMAGE SEGMENTATION

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**RADIM KRÍŽ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. MICHAL ŠPANĚL**

BRNO 2010

## Abstrakt

Práce se zabývá možnostmi interaktivní segmentace medicínských obrazových dat pomocí knihovny *ITK* a segmentační metody *level-set*. Dále diskutuje možnosti interaktivního ovlivnění průběhu segmentace pomocí uživatelem definovaných hran a usnadnění segmentace pomocí módů umožňujících automatické nastavení parametrů. Nedílnou součástí práce je uživatelsky přívětivý segmentační nástroj postavený na knihovně *ITK* využívající segmentační metodu *level-set*, který názorně předvádí možnosti módů a přidáných hran v praxi. Návrh a implementace tohoto nástroje je také součástí této práce.

## Abstract

This work deals with the possibilities of interactive segmentation of medical image data using a *ITK* library and segmentation method *level-set*. It discusses the possibilities of an interactive influencing of segmentation with user-defined edges and facilitate segmentation using modes allow automatic adjustment of parameters. An integral part of this work is user-friendly segmentation tool built on *ITK* library and segmentation method *level-set*, which clearly demonstrates the possibilities of modes and added edges in practice. Design and implementation of this tool is also part of this work.

## Klíčová slova

metoda Level-set, segmentace, interaktivita, ITK, Qt, medicínská obrazová data, segmentace obrazu, uživatelsky přívětivé GUI

## Keywords

Level-set method, segmentation, interactivity, ITK, Qt, medical image data, image segmentation, user-friendly GUI

## Citace

Radim Kříž: Interaktivní segmentace medicínských obrazových dat, bakalářská práce, Brno, FIT VUT v Brně, 2010

# Interaktivní segmentace medicínských obrazových dat

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Španěla a uvedl jsem všechny prameny, ze kterých jsem čerpal

.....  
Radim Kříž  
16. května 2010

## Poděkování

Děkuji panu Ing. Michalu Španělovi za odbornou pomoc, bez jeho nemalého přispění bych bych měl s bakalářskou prací jistě mnohem větší problémy. Dále bych rád poděkoval tvůrcům výborné knihovny ITK, kterou jsem úspěšně použil ve své práci.

© Radim Kříž, 2010.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Interaktivní segmentace medicínských dat</b>	<b>5</b>
2.1	Metoda active appearance models . . . . .	5
2.2	Metoda aktivních kontur . . . . .	5
2.3	Metoda level-set . . . . .	7
<b>3</b>	<b>Současné nástroje pro spracování obrazových medicínských dat</b>	<b>9</b>
3.1	Knihovna OpenCV . . . . .	9
3.2	Knihovna MDSTk . . . . .	9
3.3	Knihovna ITK . . . . .	10
3.4	3DSlicer . . . . .	11
3.5	ITK-Snap . . . . .	11
<b>4</b>	<b>Použití knihovny ITK</b>	<b>13</b>
4.1	Metody level-set v knihovně ITK . . . . .	13
4.2	Popis pipeline . . . . .	13
4.3	Načtení obrázku . . . . .	14
4.4	Změna rozměrů obrázku . . . . .	14
4.5	Vyhazení obrázku . . . . .	15
4.6	Detekce hran . . . . .	16
4.7	Vytvoření inicializační zero level-set hladiny . . . . .	17
4.8	Hlavní level-set filtr . . . . .	18
4.9	Zpracování výstupu level-set filtru . . . . .	19
<b>5</b>	<b>Návrh segmentačního nástroje</b>	<b>21</b>
5.1	Všeobecné předpoklady návrhu . . . . .	21
5.2	Hlavní prostředky implementace nástroje . . . . .	21
5.3	Princip použití metody level-set . . . . .	22
5.4	Návrh grafického uživatelského rozhraní . . . . .	22
5.5	Předdefinované módy . . . . .	23
5.6	Zadávání nových uživatelsky definovaných hran . . . . .	24
<b>6</b>	<b>Implementace</b>	<b>25</b>
6.1	Základní rozvržení implementace grafického uživatelského rozhraní . . . . .	25
6.2	Propojení ITK s grafickým uživatelským rozhraním . . . . .	26
6.3	Módy segmentace . . . . .	26
6.4	Semínka a řídicí body . . . . .	28

6.5	Zobrazení výsledků . . . . .	28
<b>7</b>	<b>Výsledky</b>	<b>29</b>
7.1	Ukázka evoluce metody level-set . . . . .	29
7.2	Ukázka použití uživatelem definovaných hran . . . . .	30
7.3	Ukázka použití přednastavených módů . . . . .	30
7.4	Zhodnocení výsledků aplikace a možnosti vylepšení . . . . .	31
<b>8</b>	<b>Závěr</b>	<b>33</b>
<b>A</b>	<b>Obsah CD</b>	<b>36</b>
<b>B</b>	<b>Manual</b>	<b>37</b>
B.1	Překlad a spuštění segmentačního nástroje . . . . .	37
B.2	Práce se segmentačním nástrojem . . . . .	37
<b>C</b>	<b>Ukázka výstupů segmentačního nástroje</b>	<b>39</b>
<b>D</b>	<b>Plakát</b>	<b>41</b>

# Kapitola 1

## Úvod

Oblast počítačového zpracování medicínských obrazových dat je dnes velmi atraktivní a dynamicky se rozvíjící obor. V lékařství se využívá při zpracování obrazových dat získaných například z *RTG* (*Rentgen*), *CT* (*Počítačova tomografie*), *MR* (*Magnetická rezonance*) a dalších zdrojů obrazových dat. Software na zpracování medicínských obrazových dat jde ruku v ruce s těmito fyzickými elektronickými zařízeními. Tyto přístroje jsou sice velmi přesné a symbolizují pomyslný vrchol možností současné slaboproudé techniky, nicméně obrazová data, jaká tyto přístroje poskytují, často obsahují chyby nebo jsou různě fragmentovaná, posunutá apd. Tyto nedostatky nejsou povětšinou způsobeny chybami přístrojů, ale jde zkrátka o důsledek neustálého pohybu a fungování lidského organismu, který i v průběhu vyšetření nesmí být nijak postižen. Výsledná data je často nutné dodatečně upravovat, registrovat <sup>1</sup> a segmentovat <sup>2</sup>. Právě v této chvíli se ke slovu dostávají softwarevé programy, které jsou schopny uvést data do použitelného stavu a *vdechnout jim tak život*. Příkladem takto zpracovaných dat mohou být různé virtuální modely tkání použitelné pro výuku nebo například virtuální interní vyšetření bez ohrožení pacienta. Tyto programy musejí být ideálně velmi robustní a jednoduše ovladatelné. Dotažení těchto technologií na úroveň jednoduchého každodenního použití, by způsobilo nemalou revoluci v lékařství a diagnóze.

Tématem bakalářské práce je interaktivní segmentace medicínských obrazových dat. Cílem práce bylo implementovat a otestovat jednoduchý segmentační nástroj pro medicínská obrazová data a to především pro výstupy z *CT* a *MR*. Dalším požadavkem bylo implementovat velmi intuitivní a jednoduché uživatelské rozhraní s možností ovlivnění průběhu segmentace. Pro implementaci programu je použita velmi kvalitní knihovna *ITK* společně se segmentační metodou *level-set*. Tyto technologie nejlépe vyhověli požadavkům výsledného segmentačního nástroje.

Příložený segmentační nástroj nemá ambice stát se v praxi využívanou aplikací při segmentaci medicínských obrazových dat. Jedná se především o studii a nastínění možností usnadnění segmentace pro běžného uživatele, který nemá podrobné znalosti a povědomí o vnitřních funkcích programu.

Technická zpráva se pokouší nejprve uvést čtenáře do problematiky segmentace obrazových medicínských dat se zaměřením na metodu *level-set* a knihovnu *ITK*. Následně se zabývá návrhem, implementací a zhodnocením přiloženého segmentačního nástroje.

Celá textová část práce je členěna do několika tematických kapitol. První pod názvem *Interaktivní segmentace medicínských dat* se zabývá teorií problematiky a zaměřuje se pře-

<sup>1</sup>Registrace je proces zarovnání, nebo nalezení spojitosti mezi různými daty. Například je možno k sobě přiřadit obraz získaný z *CT* a obraz stejné tkáně získaný pomocí *MR*.

<sup>2</sup>Segmentací se nazývá proces identifikace dat v digitální reprezentaci daného objektu.

devším na metodu *level-set*. Následující kapitola *Současné nástroje pro interaktivní segmentaci obrazových medicínských dat* pojednává především o knihovně *ITK* a také stručně popisuje již implementované aplikace postavené nad touto knihovnou, které slouží jako zdroj inspirace. Způsob použití knihovny ve výsledném nástroji *ITK* je diskutována v kapitole pod názvem *použití knihovny ITK*. V další kapitole *Návrh segmentačního nástroje* se obecně popisuje návrh výsledného programu. Dále následuje kapitola *Implementace*, kde je popsána samotná implementace vybraných částí segmentačního nástroje. V kapitole *Výsledky* se vyhodnocují dosažené výsledky a zda aplikace splnila stanovené cíle. V poslední kapitole *Závěr* je zhodnocení celé práce jako celku. Nedílnou součástí práce je seznam použité literatury, příloha o obsahu přiloženého CD, velké obrázky výstupů nástroje, ukázka plakátu reprezentujícího práci a příloha o překladu, spuštění a ovládání přiložené aplikace.



## Kapitola 2

# Interaktivní segmentace medicínských dat

Tato kapitola je zaměřená na teoretický výklad problematiky. Obsahuje výběr několika metod, které se s úspěchem používají k segmentaci dat. Jsou to metody, které by bylo možné využít při implementaci výsledného nástroje.

### 2.1 Metoda active appearance models

*Active appearance models* (AAM) [1] je rozšířením metody *Active Shape Models* (ASM). Je metoda segmentace, kdy se v obraze hledá výskyt předem natrénovaných vzorů. Program využívá seznam dříve natrénovaných modelů vzorových objektů, které se vytvoří pomocí statistické analýzy *Principal Component Analysis* (PCA) na základě manuálně zadaných bodů (anglicky *landmarks*), které označují významná místa v obraze. Ukázka na obrázku 2.1.

Nevýhoda spočívá především v pracném natrénování vzorů, které navíc musí být velmi precizní, aby se předešlo chybným výsledkům [18]. Tato přípravná část, spočívající ve vytvoření *knihovny vzorů*, je časově velmi náročná. Při použití metody AAM se také může stát, že selže. Metoda totiž rozezná pouze podobné vzory, což ji vylučuje při vyhledávání různých nestandardních morfologických artefaktů v segmentované tkáni.

Tato metoda není příliš vhodná pro implementaci výstupní aplikace. Přesto metoda reprezentuje zajímavý přístup k segmentaci dat, a proto je zde zmíněna.

### 2.2 Metoda aktivních kontur

V anglické literatuře jsou někdy aktivní kontury označovány jako *snakes* [3]. Při této metodě se kontura (jako implementace kontury se často používá *splajn*), kterou inicializuje uživatel, postupně tvaruje směrem k hranám objektu. Tohoto se docílí minimalizováním energie kontury. Tato energie se skládá ze tří složek, jak je vidět na rovnici 2.1. Máme-li konturu definovanou parametricky jako  $\mathbf{v}(s) = (x(s), y(s))$  můžeme psát rovnici 2.1.

$$E_{snake}^* = \int_0^1 E_{snake}(\mathbf{v}(s))ds = \int_0^1 E_{internal}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))ds \quad (2.1)$$



Obrázek 2.1: Ukázka *AMM* algoritmu. Tvář označena 122 *landmark* body. Převzato z [1]

**Interní energie**  $E_{internal}$  definuje energii *splajnu* (kontury). Pro tuto energii platí uvedená rovnice 2.2.

$$E_{internal} = E_{cont} + E_{curv} \quad (2.2)$$

Kde  $E_{cont}$  symbolizuje energii kontury a  $E_{curv}$  symbolizuje energii zakřivení *splajnu*.

**Energie obrazu**  $E_{image}$  se skládá ze tří komponent.

- **Hran** – určuje míru intenzity obrazu
- **Okrajů** – určuje míru přitahování kontury k výrazným přechodům (gradientům) obrazu
- **Omezení** – slouží k vyhlazování kontury

Pro energii obrazu platí uvedená rovnice 2.3.

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \quad (2.3)$$

Kde koeficienty  $w$  jsou váhy k příslušným energiím, kterými lze ovlivnit výsledek.

**Energie omezení**  $E_{cont}$  slouží k případnému uživatelskému ovlivnění chování kontury. Její použití je volitelné.

Existuje spousta různých variant implementací aktivních kontur podle potřeby cílových aplikací. Segmentace pomocí této metody se hojně využívá, jak při zpracování statického

obrazu tak i pro videa. Metoda je implementována například v knihovně *OpenCV*, o které je řeč v kapitole 3.1. Nevýhodou *aktivních kontur*, je, že na složitějších objektech nedosahují velké přesnosti a někdy vytvářejí nežádoucí artefakty, jako například smyčky. Bylo však vyvinuto několik modifikací, které tento nedostatek odstraňují [5]. Příklad je na obrázku 2.2.



Obrázek 2.2: Ukázka *active contours* algoritmu. Na prvním obrázku je inicializace, na druhém neupravená implementace vytvářející nežádoucí smyčky a na třetím upravená implementace bez smyček. Převzato z [18]

## 2.3 Metoda level-set

Numerická technika pro segmentaci a sledování evoluce objektů v obraze. Poprvé byla představena vědci Osherem a Sethianem [8]. Metoda přistupuje k problematice obdobným způsobem jako aktivní kontury. I zde je nutná manuální inicializace. Avšak s konturou nepracujeme přímo, ale kontura je definovaná jako nulová hladina  $\Gamma$  (anglicky *zero level-set*)  $N + 1$  rozměrné *level-set funkce*  $\varphi$ , kde  $N$  je počet rozměrů segmentovaných dat (viz. rovnice 2.4) [7].

$$\Gamma = \{(x, y) | \varphi(x, y) = 0\} \quad (2.4)$$

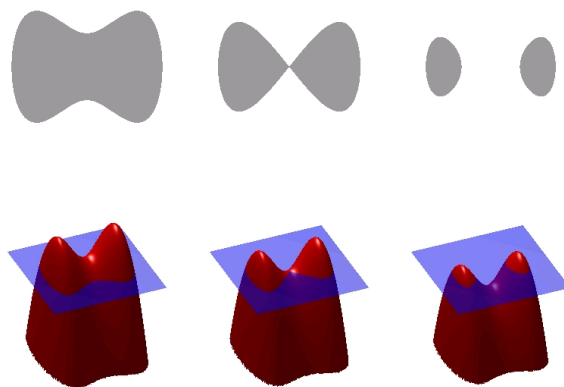
Na obrázku 2.3 je zobrazena názorná ukázka fungování metody *level-set*. Červenou barvou je znázorněn graf funkce  $\varphi$ , modrou barvou je znázorněna nulová (*zero level-set*) hladina a v horní části obrázku jsou vyobrazeny odpovídající průměty funkce  $\varphi$  do *zero level-set* hladiny.

Pokud budeme pohybovat nulovou hladinou ve kolmém směru rychlostí  $v$  bude platit parciální diferenciální rovnice 2.5 (tzv. *level-set rovnice*).

$$\frac{\partial \varphi}{\partial t} = v |\nabla \varphi|. \quad (2.5)$$

Řešení této rovnice je velmi komplikované a je potřeba zavést speciální metody. Existují různé modifikace této rovnice, podle toho co právě vyžaduje cílová aplikace.

*Fast marching* využívá jednodušší variantu *level-set* rovnice. Vychází z předpokladu, že expanze kontury ve všech bodech probíhá stejným směrem. Pro představu lze použít například nafukující balón, kde se jeho stěny ve všech bodech od sebe vzdalují.

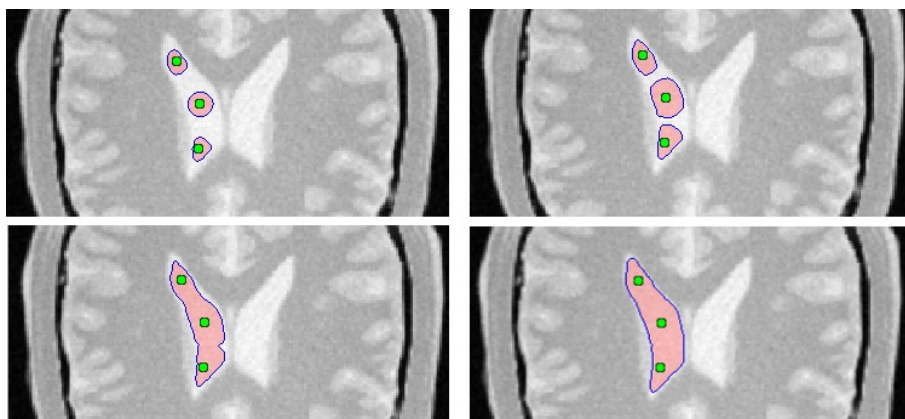


Obrázek 2.3: Ukázka fungování metody *level-set*. Převzato z [10]

V obecné definici metody *level-set* je možné, aby některé body expandovaly a jiné se smršťovaly, podle aktuální potřeby. Lze si představit například kapku oleje na neklidné hladině vody. Řešení takové rovnice je ovšem mnohem složitější.

Důležitou vlastností funkce  $\varphi$ , je, že na hranách objektu má hodnotu 0, uvnitř objektu jsou hodnoty funkce kladné a vně objektu záporné. Pomocí této vlastnosti je možné určit, kde se nachází aktuální kontura (*zero level-set*), kde vnitřek objektu a kde jeho okolí.

Mezi výhody metody patří to, že je poměrně jednoduše implementovatelná do více rozměrů (myšlenka vychází z podstaty  $N + 1$  *level-set funkce*). Dále metoda implicitně segmentuje více objektů najednou, objekty s dírou nebo rozdělující se objekty, jak můžeme pozorovat na obrázku 2.3. Metoda *level-set* je také obecně schopnější lépe detekovat jemné a členité objekty, než metoda *aktivních kontur*.



Obrázek 2.4: Ukázka evoluce *level-set* kontury, při segmentaci mozku. Výstup je získán z výsledného segmentačního nástroje.

## Kapitola 3

# Současné nástroje pro spracování obrazových medicínských dat

V této kapitole jsou v krátkosti popsány knihovny zaměřené na zpracování obrazu. Práce je zaměřena především na knihovnu *ITK*, která oproti ostatním nabízí mnohé výhody. V dnešní době je používání knihoven běžnou praxí. Tímto způsobem se ušetří spoustu drahocenného času a problémů spojených s vymyšlením a implementováním potřebných algoritmů. Navíc lze (většinou) předpokládat, že algoritmy fungují správně, jsou dobře implementovány a otestovány. Místo vymyšlení již vymyšleného je tedy možno se pustit přímo do řešení konkrétního problému.

Dále jsou zde popsány a dvě kompletní aplikace založené na knihovně *ITK*, které slouží jako inspirace při tvorbě vlastního nástroje.

### 3.1 Knihovna OpenCV

*Open Source Computer Vision Library (OpenCV)* je velmi pokročilá knihovna pro zpracování obrazu a podporu počítačového vidění vyvíjená firmou Intel od roku 1999. Je napsána v jazyce C/C++ a obsahuje více než 500 funkcí. Knihovna je samozřejmě 100% přenositelná mezi všemi běžnými operačními systémy. Z tohoto důvodu je hojně využívána jak ve výzkumu tak v profesionálních aplikacích.

*OpenCV* implementuje algoritmy pro kalibrační techniky, detekci pohybu, sledování, analýzu tvaru, segmentaci a rozpoznávání objektu [4]. Umí pracovat z valnou většinou běžných obrazových formátů jako JPEG, PNG, GIF, BMP, TIFF a další. Není však primárně určena pro zpracování medicínských dat, proto například v základu neumí pracovat s formátem DICOM. Tento formát je hojně užíván v lékařství jako výstup z *CT* nebo *MR*. Část této knihovny dokonce implementuje i základní GUI.

### 3.2 Knihovna MDSTk

*Medical Data Segmentation Toolkit* [17] je knihovna vyvíjená na fakultě Informatiky Vysokého učení technického v Brně. Jedná se o kolekci nástrojů na zpracování 2D/3D obrazových dat a to především medicínských. Je napsána v jazyce C++ a je tedy 100% přenositelná mezi všemi běžnými operačními systémy. Umí pracovat s obrazovými formáty JPEG, PNG a především s formátem DICOM. Implementuje podporu pro propojení s knihovnou *OpenCV*.

Knihovna implementuje DICOM parser a objemové zobrazení dat, obrazové filtry, hranové detektory a spoustu dalších užitečných funkcí pro zpracování obrazu (například *LBP*, *Watersheds*).

### 3.3 Knihovna ITK

*Insight Toolkit* [15] je multiplatformní systém určený pro zpracování, segmentaci a registraci obecně n-rozměrných obrazových dat se zaměřením na data medicínská. Systém je napsán v jazyce C++ a v maximální možné míře využívá objektového a generického přístupu k programování. Systém je velmi rozsáhlý avšak velké úsilí je věnováno zachování přehlednosti. Knihovna neimplementuje žádná uživatelská rozhraní. Toto je ponecháno na jiný specializovaný software jako jsou například *FLTK*, *Qt* pro tvorbu GUI a *VTk*, *OpenGL* pro vizualizaci 3D obrazových dat.

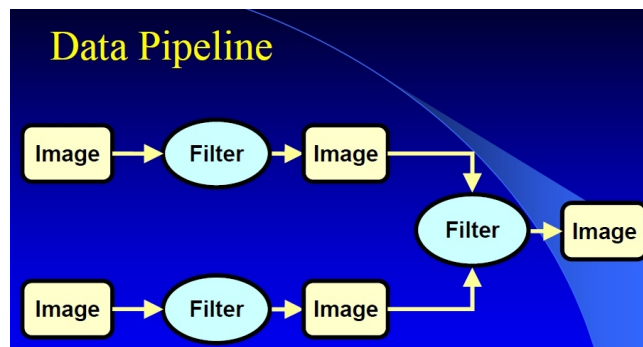
*ITK* umí pracovat se všemi běžnými formáty včetně formátu DICOM. A ty, které nepodporuje, se dají přidat vytvořením nové třídy odvozené od třídy `itk::ImageIOBase` a přepsat její virtuální metody. Tento způsob přizpůsobení funguje pro všechny třídy knihovny *ITK* a dává knihovně velkou flexibilitu. Dále obsahuje vlastní memory management pomocí tzv. *Smart pointerů*. Pro složité matematické operace využívá knihovny *VNL*.

*ITK* obsahuje obrovské množství funkcí a stále se přidávají nové, protože kdokoli na světě se může zapojit do vývoje této knihovny. Výpis všech funkcí přesahuje rámec této práce, takže zde jsou vypsány pouze hlavní skupiny funkcí s krátkým popisem [2].

- **Reprezentace a přístup k datům** – podporovány jsou dva základní strukturované typy. Jsou to obecně n-dimenzionální obrázky a nestrukturované sítě tzv. *meshes*.
- **Prostorové objekty** – slouží pro modelování prostorových objektů (například segmentovaných tkání).
- **Filtry** – do této skupiny patří značný počet různých filtrů s různým určením. Jmenujme například prahovací filtry (*Binary threshold*), hranové detektory (*Canny edge detection*), vyhlazovací filtry (*Curvature anisotropic diffusion filter*), transformační filtry (*Resample filter*) a další.
- **Čtení a zápis do souborů** – obsahuje funkce pro souborový vstup a výstup. Zvládá práci s velmi širokou paletou formátů.
- **Statistika** – obsahuje různé statistické funkce (histogramy, klasifikace, clustering).
- **Registrace** – slouží k propojení podobných obrázků (v medicíně například obrázků z CT s obrázkem z MG).
- **Segmentace** – slouží k identifikaci a klasifikaci dat v obraze. *ITK* obsahuje segmentační metody založené na metodách *watersheds*, extrakce příznaků a *level-set*. Obsahuje celkem 6 různých implementací metody *level-set*.

Jednotlivé funkce jsou realizovány jako proudové filtry, jejichž spojením do tzv. *pipeline* dostaneme požadovaný výsledek viz. Obrázek 3.1.

Knihovna *ITK* se jednoznačně jeví pro potřeby bakalářské práce jako nejvhodnější z dostupných knihoven. Neobsahuje sice metodu segmentace pomocí *aktivních kontur*, což ovšem kompenzuje přítomností hned několika implementací metody *level-set*, která je nakonec



Obrázek 3.1: Ukázka *pipeline* zpracování dat v knihovně *ITK*. Převzato z [6]

použita ve výsledném segmentačním nástroji. *ITK* obsahuje veškerou potřebnou funkčnost, má kvalitní dokumentaci a je velmi chválena odbornou veřejností. Největší výhoda oproti ostatním je nejlepší implementace segmentační metody *level-set* ze všech dříve zmíněných knihoven a přímé zaměření na medicínská data. Jistý problém může tvořit její možná až zbytečně velká obsáhlost, ale na druhou stranu toto nám poskytuje široké možnosti případného rozšíření nástroje v budoucnu.

Existuje již několik aplikací implementujících segmentaci obrazových dat na základě metody *level-set* a postavené na knihovně *ITK*. Jmenovitě například univerzální medicínský segmentační program *3DSlicer* nebo už konkrétněji zaměřená aplikace *ITK Snap* (které jsou podrobněji popsány níže v sekcích 3.4 a 3.5). Tyto programy už v zásadě umí mimo jiné i to, co je cílem této práce, avšak jejich pochopení pro běžného uživatele je poměrně náročné. Cílem tedy je implementovat segmentační nástroj pracující s metodou *level-set*, který by byl jednoduchý na pochopení a obsluhu a zároveň přidat do segmentace prvky interaktivity ve smyslu možnosti ovlivnění evoluce metody *level-set*. Je ovšem jasné, že kvalit výše zmíněných aplikací stěží dosáhnout v rámci této bakalářské práce. Stojí za nimi zkušené týmy lidí a dlouholetý vývoj.

### 3.4 3DSlicer

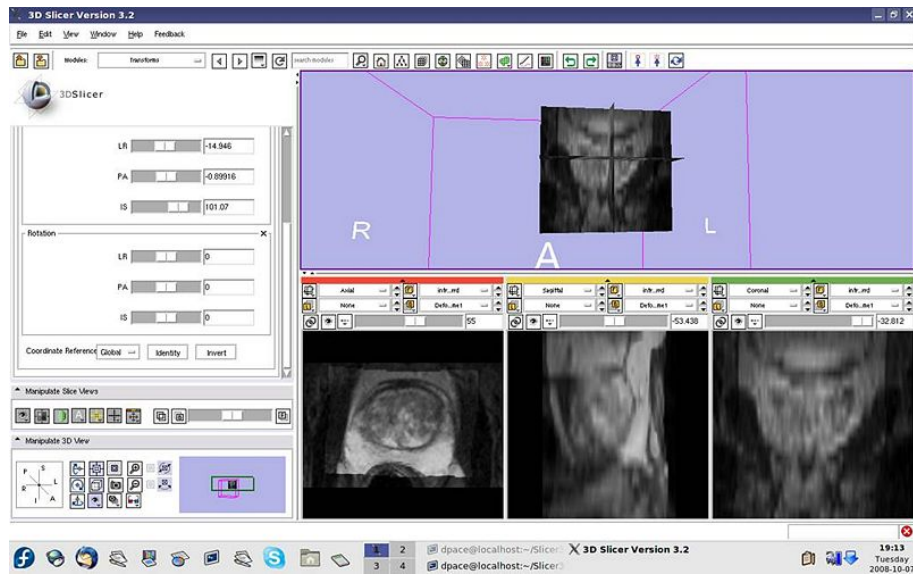
Jedná se o *open source* multiplatformní aplikaci pro registraci, segmentaci a vizualizaci více rozměrných obrazových medicínských dat. Obsahuje pokročilé algoritmy zpracování obrazu. Program je postaven především nad knihovnami *ITK* a *VTK* [11]. Ukázka pracovního prostředí je na obrázku 3.2.

*3DSlicer* je velmi rozsáhlý projekt a tvoří jej více než milion programových řádků v jazyce C++. Je velmi výkonný a všestranný, ale právě tyto jeho vlastnosti jej předurčují pro použití pouze lidem velmi dobře zasvěcených do této problematiky. Nastavení různých parametrů je velmi zdoluhavé a nepřehledné a vyžaduje odborné znalosti jak z medicínského tak informatického odvětví.

### 3.5 ITK-Snap

Je již konkrétněji zaměřená *open source* aplikace na segmentaci 3D medicínských dat postavená nad knihovnou *ITK* [9]. Tato aplikace má velmi podobnou filosofii a klade si podobné cíle jako tato práce. Také implementuje segmentaci pomocí metody *level-set* a obsahuje

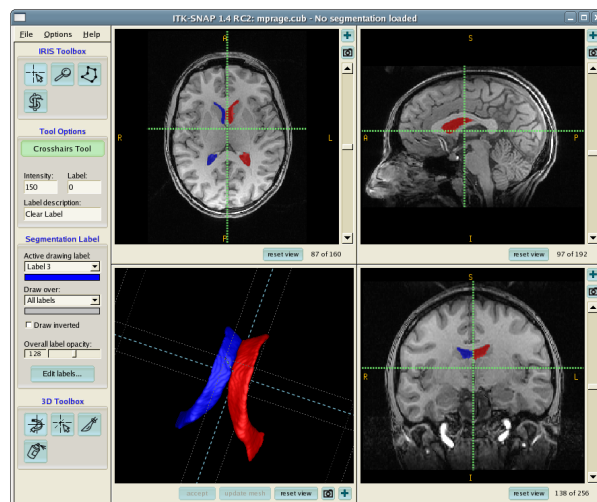




Obrázek 3.2: Ukázka pracovního prostředí aplikace *3DSlicer*. Převzato z [11]

sofistikovanou nápovědu při nastavování parametrů, ale poněkud jiným způsobem, než implementuje výsledný segmentační nástroj. *ITK-Snap* využívá posuvníků a nápomocných textů, naopak výsledný segmentační nástroj se pokouší stanovit módy, které jednotlivé parametry nastavují a odpovídají jednotlivým situacím, které během segmentace nastávají. Výsledný nástroj, taky na rozdíl od této aplikace, umožňuje kontrolovat evoluci *level-set* algoritmu pomocí krokování a přidávání nových hran.

Každopádně tato aplikace slouží jako jistý vzor a to převším z pohledu 3D zpracování a vizualizace. Ukázka pracovního prostředí je na obrázku 3.3.



Obrázek 3.3: Ukázka pracovního prostředí aplikace *ITK-Snap*. Převzato z [9]



## Kapitola 4

# Použití knihovny ITK

V této části práce je popsáno použití knihovny *ITK* ve výsledném segmentačním nástroji. Nejdříve je zde popsán způsob implementace metod *level-set* v knihovně *ITK* a poté jsou zde rozepsány jednotlivé filtry a jejich propojení do *pipeline*. Při popisu jednotlivých filtrů jsme vycházeli z [2].

### 4.1 Metody level-set v knihovně ITK

Metody *level-set* implementované v knihovně *ITK*. Se obecně řídí podle rovnice 4.1.

$$\frac{\partial \varphi}{\partial t} = -\alpha \mathbf{A}(\mathbf{x}) \cdot \nabla \varphi - \beta P(\mathbf{x}) |\nabla \varphi| - \gamma Z(\mathbf{x}) \kappa |\nabla \varphi| \quad (4.1)$$

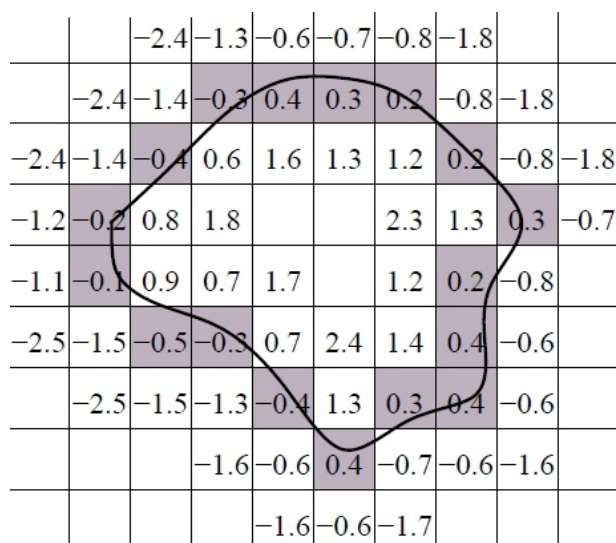
kde  $\mathbf{A}$  je přitahovací (advection) parametr,  $P$  (propagation) je parametr určující míru expanze a  $Z$  je vyhlazovací parametr. Konstanty  $\alpha$ ,  $\beta$  a  $\gamma$  určují váhy příslušných parametrů. Tato rovnice odpovídá *Geodesic active contour level-set filtru*, který je diskutován v sekci 4.8. Jiné způsoby implementací metod *level-set* využívají rovnici v jiném (často zjednodušeném) tvaru. V rovnici je možno vidět jistou podobnost s aktivními konturami. Tento filtr totiž tvoří propojení dobrých vlastností *aktivních kontur* s metodou *level-set* [7].

Většina *level-set* filtrů vyžaduje na vstupu, kromě zadání vah jednotlivým parametrům rovnice, i inicializační konturu, pro kterou platí rovnice 2.4, a hranový obrázek. Který je vlastně upraveným obrázkem, který se bude segmentovat. Všechny tyto vstupy ovlivňují evoluci metody *level-set*. Výstupem je poté obrázek se spočtenou *zero level-set* hladinou viz. obrázek 4.1.

### 4.2 Popis pipeline

Veškeré funkce v knihovně *ITK* jsou vytvořeny jako filtry na proudové zpracování dat. Tyto filtry se propojují do *pipeline*, kde na vstupu je původní obrázek a na výstupu požadovaný výsledek. Pipeline použita při vývoji nástroje, včetně vstupních parametrů jednotlivých filtrů, je zobrazena na obrázku 4.2.

Z obrázku 4.2 je patrné, že nejdůležitější součástí představuje *geodesic active contour level-set filtr*, který implementuje samotnou funkčnost segmentační metody *level-set*. Tento filtr má dva vstupy. Prvním je inicializační *zero level-set* hladina, definována jako obrázek vzdáleností. Tento obrázek se vytvoří pomocí filtru *fast marching*. Druhým vstupem je hranový obraz, který se spočítá přímo ze segmentovaného obrázku. Vytvoření tohoto obrázku



Obrázek 4.1: Ukázka výstupu *level-set* filtrů z knihovny *ITK*. Převzato z [2]

předchází použití několika filtrů, které obrázek nejprve vyhladí (*anisotropic diffusion filter*), následně v něm detekují hrany (*gradient magnitude filter*), tyto hrany se ještě upraví a převedou na požadovaný rozsah hodnot (*sigmoid filter*) a na závěr se uměle přidají hrany zadané uživatelem (tato část ovšem zcela nesplňuje princip proudového zpracování a je v diagramu uvedena jen pro úplnost (viz kapitola 5.6). Podrobněji jsou filtry z knihovny *ITK* popsány v následujícím textu.

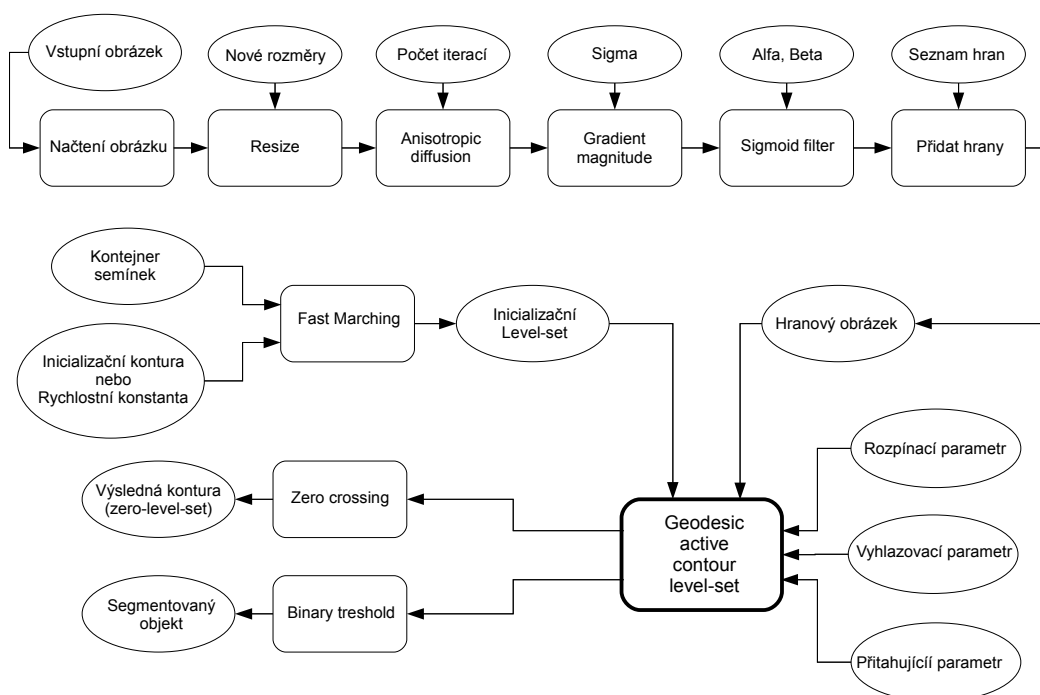
### 4.3 Načtení obrázku

Načtení se provete pomocí *image file reader filtru*. Tento filtr umí na základě jména souboru s obrázkem načíst obrazová do interní datové datové podoby *ITK*. Zvládá velké množství obrazových formátů (BMP, PNG, JPEG, TIFF, DICOM, GIPL, RAW) včetně formátu DICOM. Obecně je filtr uzpůsoben k načítání *n*-rozměrných dat. Výsledný segmentační nástroj, ale využívá pouze načítání dvou rozměrných obrazových dat.

V programu interně pracujeme s obrázkem, kde jsou hodnoty jednotlivých pixelů symbolizovány datovým typem *float*. Některé filtry totiž vyžadují značnou přesnost.

### 4.4 Změna rozměrů obrázku

Tento filtr není nutný pro správnou funkčnost výsledného nástroje, ale zpříjemňuje práci uživateli. Umožňuje segmentovaný obrázek zvětšovat nebo zmenšovat podle potřeby. Tuto funkčnost implementuje filtr *resample* z knihovny *ITK*. V parametrech filtru jsou očekávány nové rozměry stran a poměr nových rozměrů vůči původnímu rozměru obrázku. Tento filtr implementuje i spoustu jiných obrazových transformací (rotaci, výřez, ...). Pro funkčnost nástroje však není příliš podstatný, proto jej nebudeme dále podrobněji popisovat.



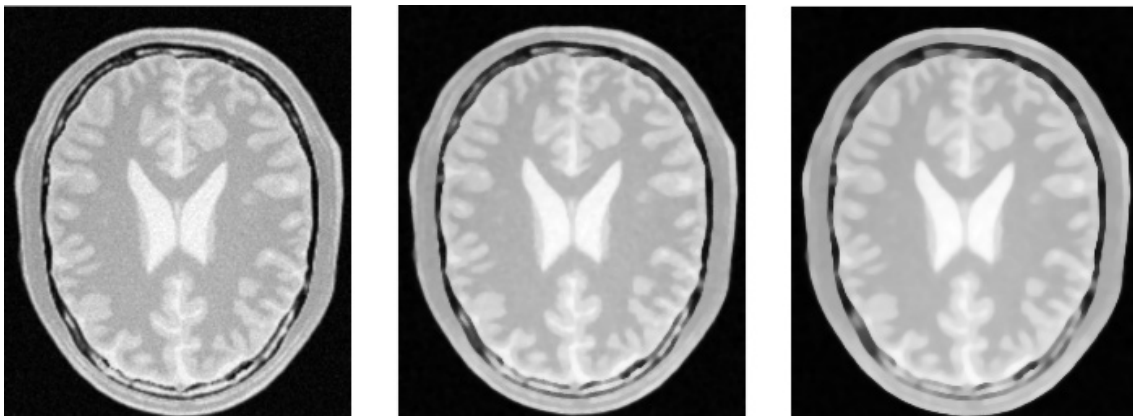
Obrázek 4.2: Digram propojení filtrů (*pipeline*) při implementaci segmentačního nástroje.

## 4.5 Vyhlazení obrázku

První ze série přípravných filtrů. Vyhladí obrázek, zbaví jej nežádoucího šumu a drobných nepřesností. Z nabídky podobných filtrů v *ITK* vybereme *curvature anisotropic diffusion filter*, který je pro potřeby aplikace nejvhodnější.

Filtr je implementován na základě *modified curvature diffusion equation (MCDE)*, která netrpí nedostatky klasických filtrů využívajících anizotropní difuze, které za určitých podmínek podléhají negativní difuzi. Díky tomuto rozrušují a potlačují hrany, což je pro výsledný nástroj nežádoucí.

Na vstupu je očekáván jediný parametr počet iterací, který určuje jak intenzivně bude obrázek vyhlazen. Obecně platí čím více šumu a nepřesností, tím více iterací je třeba provést. Je však třeba mít na paměti, že každá iterace trvá dvakrát tak dlouho než předchozí, a zvláště u větších obrázků může být velký počet iterací (více než 5) časově velmi náročný. Další důležité parametry vodivostní parametr (určuje jak radikálně bude obrázek rozmazaný a jak moc zachovat hrany) a parametr časové známky (jeho nastavení má smysl u více rozměrných dat) jsou nastaveny na výchozí hodnoty a pro aplikaci není jejich nastavování potřeba. Ukázka výstupu je na obrázku 4.3.



Obrázek 4.3: Ukázka výstupu *curvature anisotropic diffusion filtru*. První zleva je původní obrázek, uprostřed je obrázek po 5 a vpravo po 10 iteracích algoritmu. Jednotlivé ukázky byly pořízeny přímo z výsledného segmentačního nástroje.

## 4.6 Detekce hran

Detekci hran a vytvoření finálního hranového obrázku obstarávají dva filtry: *gradient magnitude recursive gaussian filtr* a *sigmoid filtr*.

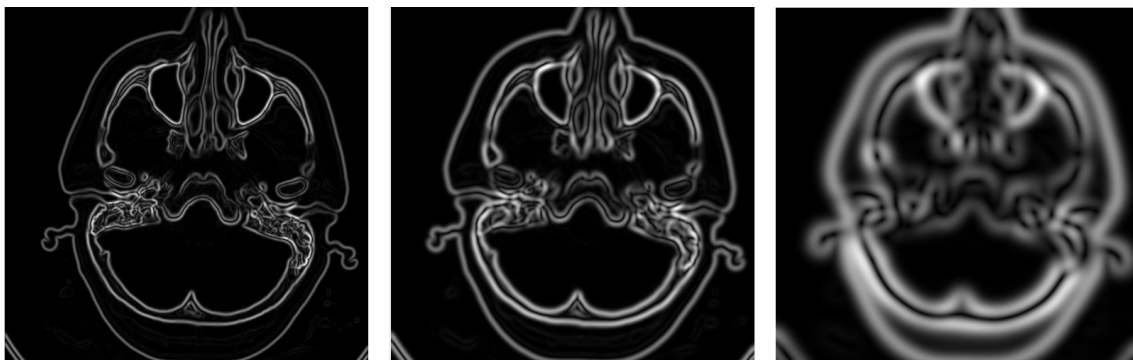
			-1
-1	0	1	0
			1

Obrázek 4.4: Ukázka derivačního jádra *gradient magnitude recursive gaussian filtru*. Převzato z [2]

**Gradient magnitude recursive gaussian filtr** provádí konvoluci s gaussovským jádrem a následně aplikuje derivační operátor, který je zobrazen na obrázku 4.4. Předcházející vyhlazení zajistí vyhlazené a pěkné hrany. Jediný parametr filtru je sigma, která se používá ke kontrole rozsahu vlivu filtru na hranách obrazu. Určuje vlastně míru rozmazání gaussovským jádrem. Důležitým faktem je, že tento filtr invertuje barvy výsledného obrázku. Příklad výstupu filtru je zobrazen na obrázku 4.5.

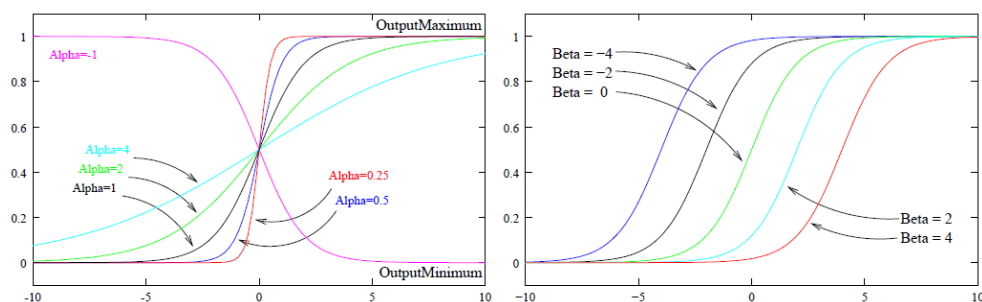
**Sigmoid filtr** patří do kategorie nelineárních mapovacích filtrů a můžeme se na něj dívat jako na variantu casting filtrů. Mapuje specifický rozsah intenzit na nový rozsah a zajišťuje hladký a plynulý přechod na hranách obrazu. Jednotlivé nové hodnoty pixelů počítá podle rovnice 4.2.

$$I' = (Max - Min) \frac{1}{\left(1 + e^{-\left(\frac{I - \beta}{\alpha}\right)}\right)} + Min \quad (4.2)$$



Obrázek 4.5: Ukázka výstupu *gradient magnitude recursive gaussian filtru*. Zleva je nastavený parametr sigma postupně na hodnoty 3, 5 a 10. Jednotlivé ukázky byly pořízeny přímo z výsledného segmentačního nástroje.

Ve výše uvedené rovnici značí  $I$  intenzitu vstupního pixelu,  $I'$  intenzitu výstupního pixelu,  $Min$  a  $Max$  definují minimální a maximální možné hodnoty výstupního pixelu,  $\alpha$  definuje rozsah vstupní intenzity a  $\beta$  intenzitu, okolo které se rozsah centruje. Pokud je  $\alpha$  nastavena na zápornou hodnotu invertuje ve výsledku barvy výstupního obrázku, což nám napraví nedostatek předchozího *gradient magnitude recursive gaussian filtru*.



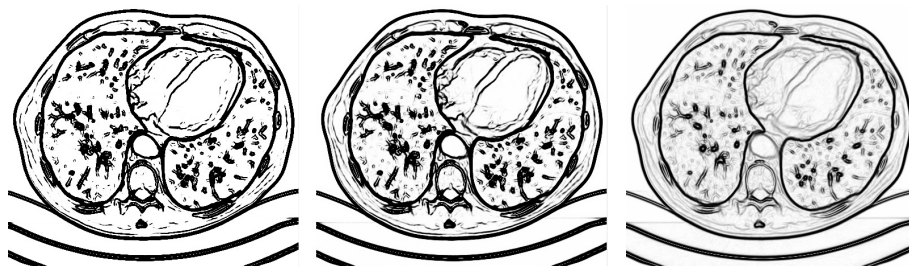
Obrázek 4.6: Grafy, které znázorňují vlivy parametrů alfa a beta na výstup *sigmoid filtru*. Převzato z [2]

V tomto případě je filtr využit ke zvýšení kontrastu mezi různými částmi obrazu. Tento filtr také překóduje obrázek na rozsah  $< 0.0, 1.0 >$ . V ideálním případě pak bude na hranách hodnota 0 a na anatomických strukturách 1.

## 4.7 Vytvoření inicializační zero level-set hladiny

K tomuto účelu poslouží *fast marching filtr*, který vytvoří mapu vzdáleností na základě zadaných parametrů a semínek. Tento filtr je vlastně zjednodušená varianta metody *level-set*, kde řídicí diferenciální rovnice má velmi jednoduchou formu a evoluce *zero level-set* hladiny probíhá jen jedním směrem.

*Fast marching filtr* očekává jako parametr semínka. To jsou body v obraze určené svou polohou a počáteční hodnotou udávající, jejich počáteční rozsah, která vlastně určuje vzdálenost inicializační nulové hladiny od semínka. Tyto semínka vkládá uživatel přes uživatelské



Obrázek 4.7: Obrázky ukazující vliv parametru  $\alpha$  na hranový obrázek ve výsledném nástroji. Parametr  $\beta$  je nastaven na 3,0 a  $\alpha$  postupně zleva na hodnoty  $-0,1$ ,  $-0,5$ ,  $-5$



Obrázek 4.8: Obrázky ukazující vliv parametru  $\beta$  na hranový obrázek ve výsledném nástroji. Parametr  $\alpha$  je nastaven na  $-0,5$  a  $\beta$  postupně zleva na hodnoty 10, 5, 1

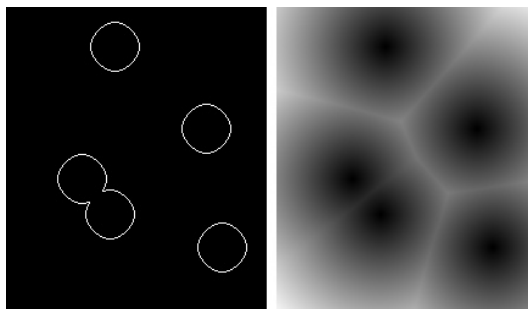
rozhraní, kde jsou symbolizovány seznamem objektů, které se pohybují přes plochu obrázku a jejichž vlastnosti jsou právě pozice a počáteční hodnota (počáteční hodnotu nastavuje přímo aplikace nikoli uživatel). Uživatel může zadat teoreticky libovolný počet těchto semínek. Častokrát je to žádoucí především při segmentaci podélných a velkých objektů, kdy dobře zadaná semínka výrazně urychlí i vylepší proces segmentace.

Aplikace využívá dva možné způsoby vstupů do tohoto filtru. Jedním je prosté zvolení rychlostní konstanty, na základě které filtr automaticky vytvoří obrázek vzdáleností. Dalším možným vstupem je uživatelem definovaná inicializační kontura, která symbolizuje inicializační *zero level-set* hladinu. První způsob se využívá v případě, že chceme, aby evoluce *level-set* algoritmu probíhala především ve směru rozpínání, druhou naopak zvolíme v případě, že chceme, aby evoluce probíhala především ve směru smršťování. Kontura je implementována podobně jako přidané hrany (opět jde o seznam řídicích bodů). Rozdíl je však v tom, že kontura vždy tvoří uzavřený kruh a je nutné propojit první a poslední bod kontury. Zde může uživatel zanechat velkou mezeru, která se v případě potřeby dopočítá úsečkou vytvořenou opět pomocí *Bresenhamova algoritmu* [12]. Dopočítané řídicí body potom leží na této úsečce v nástrojem definovaných rozestupech. Ukázky výstupů jsou na obrázku 4.9

## 4.8 Hlavní level-set filtr

*Geodesic active contour level-set filtr* je hlavní část segmentační pipeline. Provádí samotnou *level-set* metodu. Filtr vyžaduje dva vstupy, jejichž vytvoření jsme popsali výše. Jedná se segmentovaný obrázek převedený do hranového tvaru a inicializační *zero level-set* hladinu,





Obrázek 4.9: Ukázka výstupu *fast marching filtru* pro 5 semínek. První obrázek ukazuje rozložení nulové level-set hladiny a druhý je výstupní obrázek vzdáleností.

která je reprezentována mapou vzdáleností.

Knihovna *ITK* obsahuje celkem 6 různých implementací segmentační metody *level-set*. Od jednoduchých (například *fast marching*) až po složitější (například právě *geodesic active contour level-set*), které vyžadují více nastavení a více upravených vstupů, avšak umožňují lépe kontrolovat evoluci algoritmu a dávají lepší výsledky. *Geodesic active contour level-set filtr* byl vybrán právě pro tyto jeho vlastnosti.

Parciální diferenciální rovnice, která řídí evoluci *zero level-set* hladiny, je doplněna o tři parametry. Tyto parametry ovlivňují evoluci a je třeba nastavit je v závislosti na situaci a aktuálních potřebách segmentace.

**Parametr propagation scaling** – Rozpínací parametr určuje intenzitu rozpínání výsledné kontury. V případě zadání záporných hodnot se kontura bude smršťovat.

**Parametr curvature scaling** – Vyhlažovací parametr vyhlazuje výslednou konturu a vyrovnává se s drobnými nepřesnostmi.

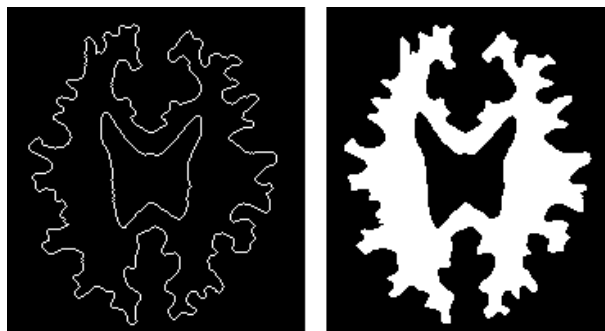
**Parametr advection scaling** – Přenosový parametr přitahuje *level-set* konturu k hraně objektu. V principu často směřuje proti rozpínacímu parametru. Zabraňuje přeskočení kontury.

Dalšími parametry které je třeba nastavit jsou *RMS chyba* (*root mean squared error*) a maximální počet iterací filtru. Na základě těchto parametrů se algoritmus ukončí dosáhne-li zadané přesnosti nebo maximálního počtu iterací.

Výstupem algoritmu je obraz, kde pixely s nulovou hodnotou symbolizují *zero level-set* hladinu, tedy naši výslednou konturu, pixely s kladnými hodnotami symbolizují vnitřek segmentovaného objektu a pixely se zápornou hodnotou naopak vněšek objektu.

## 4.9 Zpracování výstupu level-set filtru

O zpracování výstupu geodesic active contour filtru se starají dva filtry. Těmito filtry jsou *zero-crossing filtr* a *binarythreshold filtr*. Ukázka výstupů těchto filtrů je na obrázku 4.10.



Obrázek 4.10: Ukázka výstupu zero-crossing filtr (vlevo) a binarythreshold filtr (vpravo).

**Zero-crossing filtr** vyhledává v obraze nulové přechody, které v našem případě symbolizují výslednou *zero level-set* hladinu. Tato funkce tedy vrací výslednou konturu segmentovaného objektu.

**Binarythreshold filtr** převádí hodnoty uvnitř objektu na hodnotu 255 a hodnoty vně na hodnotu 0. Výstup z tohoto filtru využíváme při vybarvování segmentovaného objektu.



## Kapitola 5

# Návrh segmentačního nástroje

Tato kapitola popisuje jakým způsobem postupoval návrh programu. Během návrhu bylo třeba se zaměřit především na následující body.

- Maximální využití vlastností a výhod segmentační metody *level-set*.
- Využití co nejvíce možností a začlenění knihovny *ITK* do výsledného nástroje.
- Jednoduchost a přívětivost uživatelského rozhraní nástroje.
- Použití módů při nastavování parametrů segmentace.
- Možností ovlivnění evoluce metody *level-set* pomocí uživatelem definovaných hran.

Jednotlivé body jsou podrobněji diskutovány v následujících částech této kapitoly.

### 5.1 Všeobecné předpoklady návrhu

Program by měl mít maximálně uživatelsky přívětivé grafické rozhraní, aby se v něm vyznal i uživatel nepodrobně zasvěcený do této problematiky. Implementace by se měla obejít bez složitého nastavování spousty parametrů. Veškerá nastavení, se budou provádět pomocí popsaných přednastavených módů. Měla by však být i zachována možnost vše nastavit manuálně. Samotná uživatelská kontrola segmentace a ovládání programu se bude provádět pomocí myši (klávesnice pouze na zadávání hodnot do polí).

### 5.2 Hlavní prostředky implementace nástroje

Pro implementaci výsledného segmentačního nástroje jsou využity dva hlavní prostředky metoda *level-set* a knihovna *ITK*.

**Metoda level-set** je využita v implementaci výsledného nástroje pro několik výhod, které nabízí oproti jiným segmentačním metodám. Mezi tyto výhody patří poměrně snadné a intuitivní rozšíření na segmentaci vícerozměrných dat, dále implicitní segmentace více objektů najednou a rozdělujících se objektů. Nejzákladnější princip metody je pro uživatele velmi pochopitelný, navíc lze u této metody snadno implementovat princip postupného krokování segmentace. Tyto výhody ji určují jako nejvhodnějšího kandidáta na segmentaci obrazových medicínských dat, kde se tyto vlastnosti uplatní. Důkazem vhodnosti metody

pro využití v této oblasti je její začlenění do knihovny *ITK*, která je přední světovou knihovnou v segmentaci obrazových medicínských dat.

**Knihovna *ITK*** je použita jako knihovna implementující segmentační metodu *level-set*. Tato knihovna je velmi obecná a populární v oblasti segmentace medicínských obrazových dat. Pro implementaci nástroje je podstatné, že obsahuje hned několik způsobů implementace metody *level-set*, proto je tato knihovna použita k vypracování výsledného segmentačního nástroje.

### 5.3 Princip použití metody *level-set*

Při začlenění metody *level-set* do výsledného nástroje se využije, v co možná nejširší míře, jejich vlastností. Přitom se vychází z metody *level-set*, tak jak je implementována v knihovně *ITK*.

Metodu *level-set* je možné ovlivnit nastavením jejich parametrů, semínky, inicializačními konturami nebo přidáním nových uživatelem definovaných hran. Parametry se nastavují buďto přes módy nebo manuálně, zadávání semínek a inicializačních kontur probíhá přímo do obrázku přes uživatelské rozhraní.

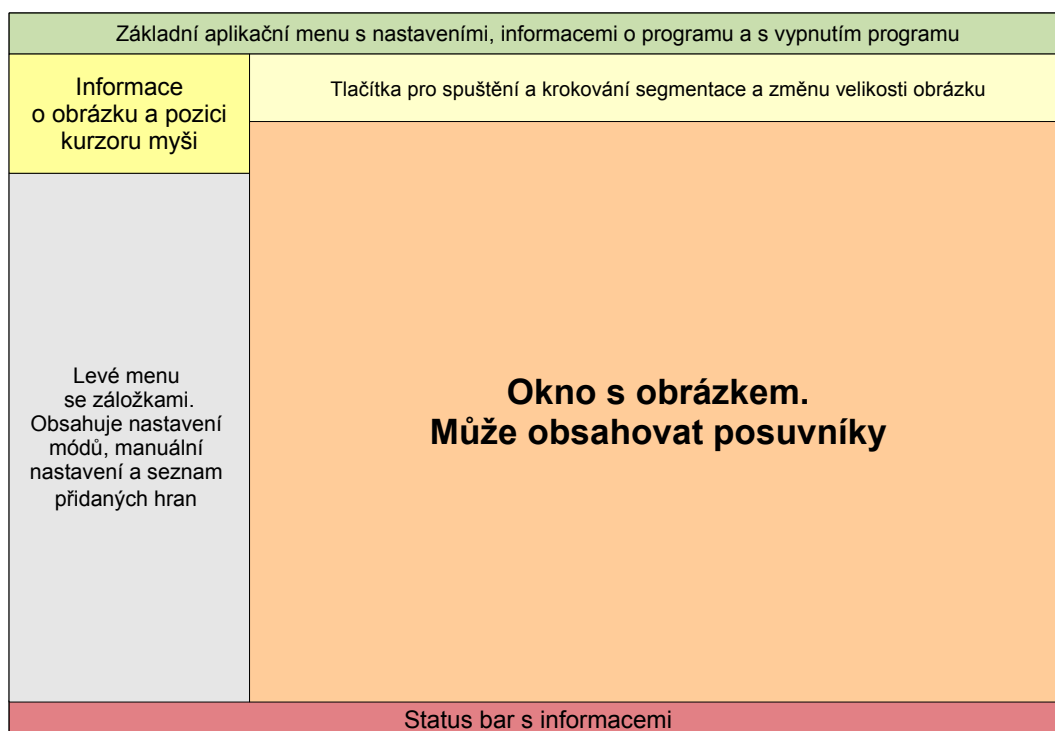
Nové semínko lze přidat kliknutím do obrázku (po kliknutí na tlačítko spouštějící přidávání semínek). Semínek je možno zadat i více pro segmentaci více objektů najednou nebo pro segmentaci větších či podlouhlých objektů. Pozici semínka lze dodatečně upravovat.

Inicializační kontury se budou zadávat pouze v případě, že chceme využít smršťování metody *level-set*. Kontura se zadá kliknutím do obrázku a pohybem myši (obtáhnutí zadaného objektu). Důležité je, že vždy bude muset tvořit uzavřený kruh, bude tedy potřeba propojit první a poslední bod. I inicializační kontury budou dodatečně modifikovatelné, aby co nejlépe obemykaly segmentovaný objekt.

### 5.4 Návrh grafického uživatelského rozhraní

Velmi důležitým bodem aplikace je kvalita a přehlednost grafického uživatelského rozhraní. Obecné nastavení programu, otevírání obrázků, vypínání aplikace, mazání výsledků a mazání nastavení bude umístěno standartním způsobem v horní liště aplikace. V horní liště budou umístěny i odkazy na okna obsahující základní nápovědu a informace o programu. Z hlavního okna budou dostupné informace o obrázku (název a originální rozměry) a aktuální pozici kurzoru myši v obraze. V levé části okna budou tři záložky. První s nastavením módů segmentace, druhá bude obsahovat formulář pro manuální nastavení segmentace a ve třetí bude zobrazen aktuální seznam uživatelem přidaných hran. Nad oknem s obrazem budou tlačítka pro spuštění segmentace, krokování segmentace a tlačítka pro změnu velikosti obrázku. Změna bude probíhat podle dříve nastavených konstant a bude udržovat poměr původního rozměru obrázku. V případě, že obrázek bude větší než je možno zobrazit v okně, budou se v okně s obrázkem zobrazovat posuvníky. Během průběhu segmentace se v okně s obrázkem bude zobrazovat současný stav programu. V dolní části programu se bude zobrazovat *status bar* s pomocnými informacemi. Rozvržení uživatelského rozhraní je zobrazeno na obrázku 5.1.

Jak už bylo řečeno v úvodní teoretické části knihovna *ITK* interně neobsahuje žádné nástroje pro vizualizaci ani grafické uživatelské rozhraní. K tomuto účelu slouží jiné specializované nástroje, které se s knihovnou *ITK* propojí. Výsledný nástroj pracuje pouze ve 2D



Obrázek 5.1: Návrh grafického uživatelského rozhraní segmentačního nástroje.

a tedy nevyžaduje 3D vizualizaci, což situaci poněkud zjednodušuje. Vizualizaci segmentovaného obrázku a výsledků i grafické uživatelské rozhraní je zajištěno pouze kvalitním frameworkem *Qt*, který obsahuje veškerou požadovanou funkčnost.

Framework *Qt* [16] je *open source* multiplatformní knihovna pro vytváření uživatelských rozhraní vyvíjená společností Nokia (třívě Trolltech) pod licencí LGPL. Primárně je psána v jazyce C++, ale existuje i pro jiné jazyky. V knihovně *Qt* je napsáno například prostředí KDE nebo programy Skype, Opera, Google Earth a další. Součástí *Qt* je velmi kvalitní dokumentace. Dále v dokumentaci knihovny *ITK* je *Qt* uvedeno jako jedna doporučených možností vytvoření uživatelského rozhraní nad *ITK*. Především z těchto důvodů je využit tento framework při tvorbě výsledného nástroje.

## 5.5 Předdefinované módy

Protože manuální nastavení segmentační metody *level-set* je poměrně obtížné. Je vytvořeno několik (tato množina není zcela jistě konečná a může se postupně rozrůstat) módů, které nastavují parametry metody *level-set* a dalších pomocných filtrů tak, aby evoluce *zero level-set* v maximální možné míře proběhla podle přání uživatele. Tyto módy se budou nastavovat z rolovací nabídky. U každého módu bude stručný popis o tom v jakém případě jej využít. Může existovat i více současně nastavitelných skupin módů, avšak ne tolik, aby to bylo zbytečné nebo mátló uživatele. Parametry segmentace se pak nastaví pomocí kombinace těchto módů.

## 5.6 Zadávání nových uživatelsky definovaných hran

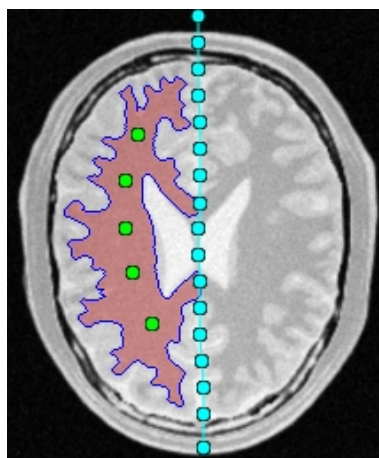
Uživatel bude moci ovlivnit průběh evoluce metody *level-set* přidáním nových hran. Tímto způsobem lze například vytvářet nové hrany, tam kde před tím nebyly, spojovat segmentované objekty, opravovat poškozené nebo neúplné hrany a aplikovat další požadované úpravy segmentovaných dat.

Nové hrany se budou zadávat myší přímo do obrázku. Půjdou editovat pomocí řídicích bodů. Bude dostupný jejich seznam (třetí záložka v levém menu), přes který bude možné dohledat, zvýraznit a smazat jednotlivé hrany.

Jak je možno vidět na digramu 4.2 tato funkčnost je součástí pipeline. V případě této funkce se však nedá zcela mluvit o proudovém zpracování. Tato funkce je implementována původně pro výsledný segmentační nástroj a není přímou součástí knihovny *ITK*, pouze využívá její rozhraní při přístupu k obrazovým datům.

Tato část úzce spolupracuje s uživatelským rozhraním nástroje. Přikreslí uživatelem zadané nové hrany do již hotového hranového obrázku, který je jedním ze dvou vstupů do *geodesic active contour level-set filtru*. Tímto způsobem lze velmi lehce a přehledně ovlivnit evoluci *zero level-set* hladiny.

Nové hrany zadává (po kliknutí na příslušné tlačítko spouštějící možnost zadávání nových hran) uživatel přímo do obrázku pomocí stisknutí tlačítka myši a následným pohybem kurzoru. Po uvolnění tlačítka se zobrazí nově přidaná hrana definovaná seznamem řídicích bodů, které slouží k další případné editaci hrany. Po startu segmentace program spojí body ze seznamu úsečkami pomocí *Bresenhamova algoritmu* [12] a body těchto úseček se poté zapíší do hranového obrázku, pomocí přístupových metod knihovny *ITK*. Příklad použití uživatelem definovaných hran je na obrázku 5.2.



Obrázek 5.2: Obrázek demonstrující možnosti segmentace s přidáváním vlastních hran. Zde se jedná o segmentaci pouze pravé hemisféry mozku. Modrozeleně je zbarvena přidaná dělicí hrana.

## Kapitola 6

# Implementace

Tato kapitola se zabývá konkrétní implementací programu. První část kapitoly tvoří popis použití knihovny *ITK* a jejich filtrů, které jsou použity v programu. Dále je zde popsána implementace stěžejních částí uživatelského rozhraní, zakomponování knihovny *ITK* a podrobněji rozebrány zajímavé části programu.

Aplikace je napsána v C++ a využívá objektový přístup k programování. Jednotlivé kompaktní části programu jsou rozděleny do tříd. Každá třída je definována ve dvou souborech. V hlavičkovém souboru, který obsahuje prototyp třídy spolu se všemi jejími metodami a vlastnostmi, a zdrojovém souboru, který definuje jednotlivé metody třídy. Podrobnější popis jednotlivých tříd lze nalézt v programové dokumentaci, která se nachází na přiloženém CD.

### 6.1 Základní rozvržení implementace grafického uživatelského rozhraní

Grafické uživatelské rozhraní je rozděleno do několik provázaných tříd. Z nichž nejvíce důležité jsou třídy: `mainWindow`, `imageWidget`, `pointWidget` a `topLayer`.

**Třída `mainWindow`** definuje celkový design hlavního okna a obsahuje veškeré prvky, kterými uživatel nastavuje parametry segmentace. V této třídě se také sdružují veškeré objekty a tato třída definuje jejich zobrazení.

**Třída `ImageWidget`** definuje widget s obrázkem. Tato třída zobrazuje obrázek a odchytává jednotlivé události myši při zadávání semínek, inicializační kontury nebo přidávání nových hran do obrázku. Tato vrstva obsahuje tři objekty třídy `topLayer`, do kterých vykreslují výsledky, inicializační kontury a přidané hrany.

**Třída `pointWidget`** implementuje objekty, které vytvářejí body, se kterými lze volně pohybovat metodou drag and drop. Používá se k definování semínek nebo úpravám inicializační kontury a nových hran (kde fungují jako řídicí body).

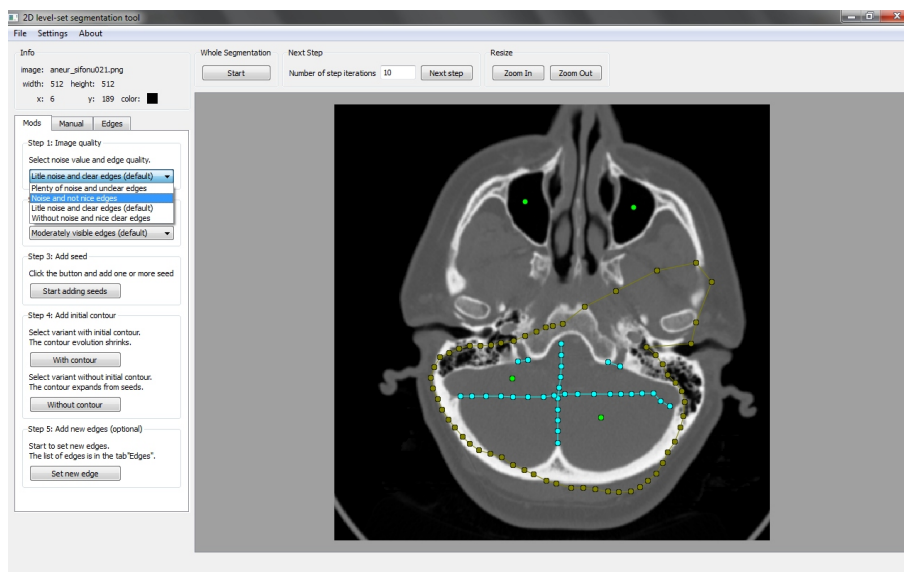
**Třída `topLayer`** se používá při vytvoření vrchní vrstvy nad widgetem s obrázkem. Do této vrstvy se poté mohou vykreslovat výsledky. K tomuto účelu obsahuje potřebné vykreslovací metody. Je velmi výhodné vykreslovat výsledky do speciálních vrstev. Během

úprav nemusíme znova přepisovat obrázek, ale jen potřebnou vrstvu, což výrazně urychlí a zpřehlední program.

Zdrojový kód aplikace obsahuje ještě další pomocné třídy.

- `myLabel` – upravení a přizpůsobení standardní `Qt` třídy `QLabel`
- `myButton` – upravení a přizpůsobení standardní `Qt` třídy `QPushButton`
- `imageViewer` – používá se k jednoduchému zobrazení obrázků, využívá se při ladění

Výsledná podoba uživatelské rozhraní, tak jak je implementována, je na ukázce 6.1.



Obrázek 6.1: Ukázka implementovaného grafického uživatelského rozhraní.

## 6.2 Propojení ITK s grafickým uživatelským rozhraním

Ve výsledném nástroji pracuje knihovna *ITK* na pozadí. Z grafického uživatelského rozhraní přijímá příkazy a datová komunikace probíhá přes tzv. *Pixel buffery*, ze kterých si uživatelské rozhraní (v tomto případě knihovna *Qt*) obrazovou informaci vytáhne.

Veškerá potřebná funkčnost knihovny *ITK* je zabalena do speciálního objektu definovaného třídou `levelSet`, která obsahuje metody jak pro veškerá nastavení, tak pro kontrolu průběhu segmentace. Tyto metody jsou volány z uživatelského rozhraní a uživatel je tedy schopen segmentaci jednoduše ovládat.

## 6.3 Módy segmentace

Jak už bylo předesláno dříve, výsledný nástroj se pokouší zjednodušit segmentaci tím, že využívá možnosti automatické segmentace pomocí jednoduše zvolitelných módů. Jednotlivé módy vlastně ovlivňují parametry jednotlivých filtrů tak, aby pokud možno co nejlépe odpovídali požadavkům segmentace. Módy jsou rozděleny do dvou skupin – do dvou roletkových menu (implementovaných pomocí `Qt` třída `QComboBox`)

**Definici kvality obrázku** Zde se udává míra zašumění obrázku a kvalita hran objektu, který chceme segmentovat. Tohoto nastavení existují 4 typy podle úrovně šumu a kvality hran od nekvalitních obrázků a hran až po velmi kvalitní, kde se hodnoty parametrů nastavují na velmi nízké hodnoty. Tyto módy nastavují parametr vyhlazování pro *geodesic active contour level-set filtru* a počet iterací pro *curvature anisotropic diffusion filtru*.

Jednotlivé módy této skupiny jsou popsány v následujícím seznamu. Názvy módů jsou v angličtině, tak jak je to ve výsledné implementaci nástroje.

- **Plenty of noise and unclear edges** – Objekt je nezřetelný a zašuměný, nastavuje vysoký počet iterací vyhlazovacího filtru a vysokou míru vyhlazování hran.
- **Noise and not nice edges** – O málo zřetelnější objekt a méně šumu, ale stále je potřeba data poměrně hodně upravit.
- **Little noise and clear edges** – Výchozí nastavení, obraz obsahuje pouze nepatrný šum a poměrně výrazné hrany.
- **Without noise and nice clear edges** – Bez šumu a zřetelné hrany. Obrázek ani kontura nebudou téměř vůbec vyhlazeny.

**Definici zřetelnosti hran** Tato skupina módů definuje míru viditelnosti hran, na které se má brát zřetel. V případě, že při segmentaci záleží pouze na hlavních hranách, nastaví se parametry na takové hodnoty, aby potlačili hrany méně výrazné a naopak zvýraznili pouze hlavní výrazné hrany. Pro toto nastavení existuje 5 různých módů. Od módu, kdy záleží jen na nejhlavnějších hranách až po mód beroucí zřetel i na hrany velmi málo výrazné. Tyto módy nastavují parametry propagace a přilnavosti pro *geodesic active contour level-set filtr*. Dále parametr sigma pro *gradient magnitude recursive gaussian filtr* a parametry  $\alpha$ ,  $\beta$  pro *sigmoid filtr*.

Jednotlivé módy této skupiny jsou popsány v následujícím seznamu. Názvy módů jsou opět v angličtině, tak jak je to ve výsledné implementaci nástroje.

- **Only main edges** – Segmentace se zaměřuje pouze na hlavní hrany obrazu, méně výrazné hrany a objekty ignoruje. Lze využít chceme-li například segmentovat celou tělní dutinu včetně některých fragmentů, které chceme zanedbat.
- **Main and little les** – Segmentace se zaměřuje opět na hlavní hrany obrazu, ovšem rozpozná i některé méně výrazné. Například tepny v orgánech.
- **Moderate visible edges** – Výchozí nastavení, citlivé zhruba na průměrné hrany v obraze.
- **Moderate visible and little less** – o něco citlivější než moderate.
- **Care even about bland edges** – evoluci ovlivní i velmi slabě viditelné hrany. Využití při segmentaci podobných tkání.

Hodnoty jednotlivých filtrů jsou zvoleny na základě empirických pokusů na poskytnutých medicínských datech. Úspěšnost módů je diskutována v následující kapitole 7.

## 6.4 Semínka a řídicí body

Semínka i řídicí body jsou definovány stejným způsobem. Každý bod nebo semínko je reprezentováno jako jeden objekt třídy `pointWidget`. S tímto objektem lze libovolně pohybovat po obrázku metodou `drag and drop`. Toto je implementováno pomocí metod frameworku *Qt* na odchyťování událostí myši. Tento objekt si také vnitřně jako vlastnost uchovává svou pozici nad obrázkem.

Nové hrany a inicializační konury jsou definovány seznamem těchto objektů, které se chovají jako řídicí body. Abychom mohli najednou definovat více hran a kontur jsou tyto seznamy řídicích bodů součástí nadřazených seznamů, které obsahují jednotlivé hrany a kontury.

## 6.5 Zobrazení výsledků

Jednotlivé výsledky segmentace (výstupy segmentační *pipeline*) se zobrazují do speciálních vrstev, které jsou umístěny nad segmentovaným obrázkem. Tyto výsledky předává objekt třídy `levelSet` v pixel-bufferech a o jejich překreslování do nadřazených vrstev se stará objekt třídy `topLayer`.



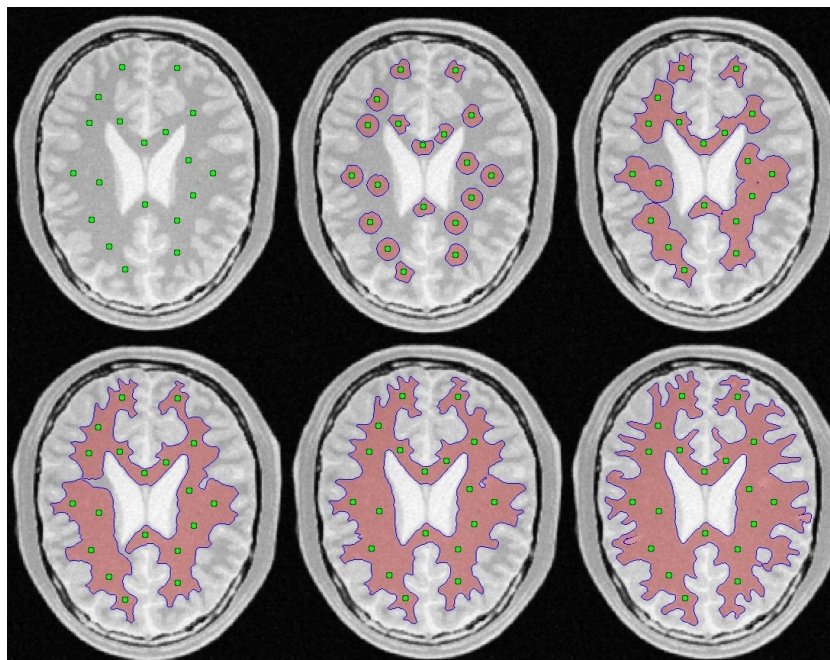
## Kapitola 7

# Výsledky

V této sekci je uvedeno několik segmentací a jejich výsledků, tak jak jich dosáhl výsledný segmentační nástroj. Jednotlivé objekty jsou barevně odlišeny. Modrou barvou je vyznačena aktuální *zero level-set hladina*. Červená vyplňuje vnitřek segmentovaného objektu. Zelené body symbolizují semínka pro tvorbu obrázku vzdáleností. Tmavě žluté body spojené úsečkami označují uživatelem definovanou inicializační konturu. A světle modré body spojené úsečkami označují uživatelem přidané nové hrany.

V závěru kapitoly je poté shrnutí výsledků aplikace a zhodnocení naplnění počátečních cílů. Také jsou zde uvedeny návrhy na možná zlepšení a rozšíření stávající podoby aplikace.

### 7.1 Ukázka evoluce metody level-set

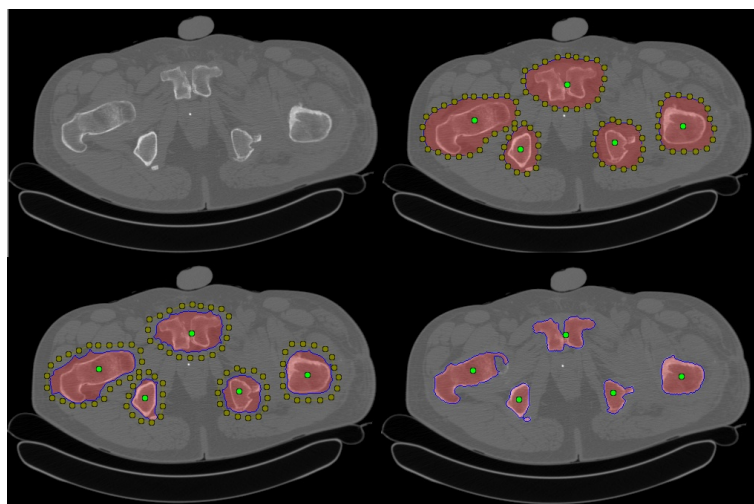


Obrázek 7.1: Ukázka rozpínací evoluce *zero level-set* hladiny při segmentaci mozku na snímku z MR

Na obrázku 7.1 je zobrazena evoluce rozpínací varianty metody *level-set*. Příklad ukazuje jednotlivé výstupy po 50 krocích iterace a názorně předvádí schopnosti metody. I tak členitý a oproti okolí poměrně málo výrazný objekt dokáže výsledný segmentační nástroj poměrně velmi pěkně vyznačit. Dodejme, že byly použity módy *noise and not nice edges* ze skupiny *Image quality* a *care even about bland edges* ze skupiny *Edge visibility level*. Z použitých módů vyplývá, že abychom dosáhli požadované jemnosti výsledku bylo nutné brát v úvahu i velmi jemné hrany.

Této ukázky 7.1 jsme dosáhli pomocí funkce krokování, kterou výsledná aplikace implementuje. Jednotlivé kroky jsou po 50 iteracích filtru *geodesic active contour level-set filtru*.

Na dalším příkladu výstupu 7.2 je opět ukázka evoluce kontury *zero level-set*. Tentokrát se však jedná o proces smršťování. Na obrázku si všimněte uživatelem inicializované kontury a následnému přilnutí kontury k okraji objektu. Tento způsob segmentace se velmi hodí, jak je konečně vidět na příkladu, například pro kosti ze snímků CT nebo RTG. Na posledním obrázku v sérii byli odstraněny inicializační kontury pro lepší přehlednost.



Obrázek 7.2: Ukázka smršťovací evoluce metody *level-set* na segmentaci pánevních kostí na snímku z CT

## 7.2 Ukázka použití uživatelem definovaných hran

Na obrázku 7.3 je ukázka využití uživatelsky definované hrany. V tomto případě je přidána hrana využita na zdůraznění rozmazaného okraje objektu. Tímto způsobem lze upravovat různé chyby, které by jinak negativně ovlivnili segmentaci.

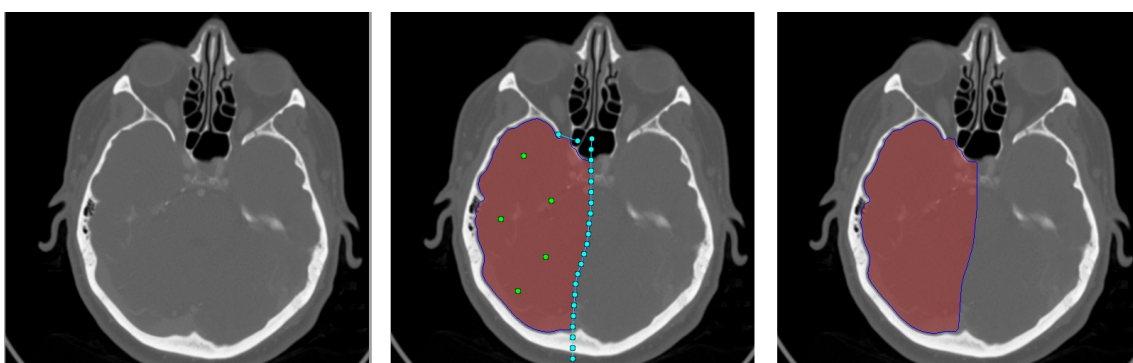
Na dalším příkladu použití 7.4 uživatelem definovaných hran je zobrazena segmentace pouze levé půlky lidské lebky. Jak lze vidět, přidáním hran lze vysegmentovat i různé části stejných tkání nebo dutin.

## 7.3 Ukázka použití přednastavených módů

Zde najdete ukázku možných způsobů využití přednastavených módů segmentace. V první části ukázky 7.5 je segmentace všech tkání mimo plicní dutinu. Ve druhé části je segmentace



Obrázek 7.3: Ukázka využití uživatelsky definovaných nových hran na upřesnění rozmazané hrany.



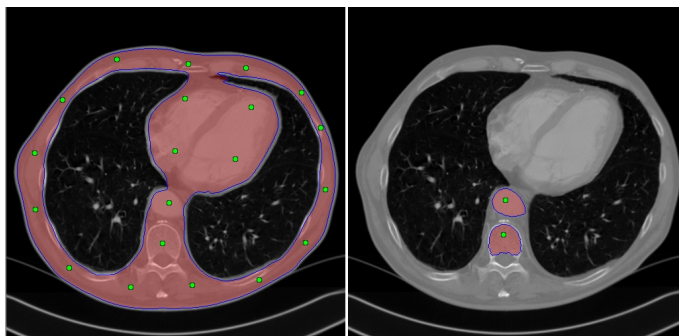
Obrázek 7.4: Ukázka využití uživatelsky definovaných nových hran při segmentaci částí stejných tkání nebo dutin.

pouze hltanové trubice a hlavice obratle. V první části byl použit mód *Only main edges* který se zaměřuje pouze na velmi ostré hrany a méně výrazné přechody ignoruje. Naopak mód *Care even about bland edges*, použitý na druhou část, bere zřetel i na velmi nevýrazné hrany.

## 7.4 Zhodnocení výsledků aplikace a možnosti vylepšení

Jak je vidět z ukázek výsledná aplikace funguje správně a je schopna zvládnout i těžší segmentační problémy. Nicméně i přesto lze ještě mnohé zlepšovat. Každopádně s výsledky segmentace jsem spokojen. Celková funkčnost aplikace odpovídá předsevzaným cílům.

Pro efektivnější práci by bylo vhodné nástroj rozšířit do 3D prostoru a pracovat přímo s objemovými daty. Principy tohoto segmentačního nástroje (módy, přidávání nových hran) by se však dali využít i u případného rozšíření o práci ve 3D. S ohledem na takové budoucí rozšíření byla zvolena i metoda *level-set*, která umožňuje poměrně snadný přechod do 3D. Dalším prvkem, který by velmi zjednodušil přechod do více rozměrů, je použití knihovny *ITK*. Tato knihovna přímo v její implementaci předpokládá použití pro spracování obecně více rozměrných dat. Největší problém v přechodu do 3D by tedy určitě představovala vhodná vizualizace.



Obrázek 7.5: Ukázka použití přednastavených módů při segmentaci různě výrazných hran a přechodů

Rozšíření o možnost práce v 3D rozměrném prostoru by z výsledného nástroje vytvořilo skutečně kompletní aplikaci. Nicméně vytvoření takové aplikace nebylo plánem bakalářské práce a není ani v silách autora. Principy nástroje, by se ovšem dali ve 3D prostoru využít. Nejprve by se vytvořila 2D inicializační kontura pomocí principů výsledného nástroje a poté by se z této kontury spustila *level-set evoluce* do 3D prostoru.

Dalším vhodným rozšířením by bylo zvýšit počet dobře okomentovaných módů, případně je roztrždit do více skupin. Toto by se však vytvářelo až na základě požadavků uživatelů. Navíc různé typy vstupních obrázků se chovají různě – mají například různé rozložení intezit. Bylo by vhodné například vytvořit speciální sadu módů pro data z *MR* a z *CT* data.

## Kapitola 8

# Závěr

Stanovené cíle, tak jak jsou definovány v úvodu, se povedlo splnit. Výsledný segmentační nástroj je funkční a jeho výsledky odpovídají původním představám. Uživatelské rozhraní se díky implementaci módů velmi zjednodušilo. Z nutnosti nastavování mnoha číselných parametrů se zredukovalo pouze na výběr daného módu, zadání semínek a případně inicializačních kontur. Umělá korekce evoluce metody *level-set* se také osvědčila a umožnila segmentaci i složitějších nebo poškozených objektů.

Metoda *level-set* společně s knihovnou *ITK* se ukázala jako velmi dobrá volba pro implementaci nástroje. Metoda *level-set* úspěšně funguje i na velmi obtížně segmentovatelných objektech s nejasnými hranami a poskytuje velkou variabilitu způsobů segmentace. I knihovna *ITK* přesvědčila o svých nesporných kvalitách. S knihovnou se velmi dobře pracuje a je velmi dobře implementována. Díky její obecnosti a robustnosti poskytuje do budoucna velmi široké možnosti případných rozšíření.

I přes splnění zadaných cílů však výsledný nástroj zdaleka nedosahuje kvality pokročilých aplikací na zpracování medicínských obrazových dat. Určitě největším nedostatkem je absence práce se 3D obrazovými daty. Nicméně výsledný nástroj může velmi dobře posloužit jako základ pro takovou aplikaci a především speciální principy nástroje jako je přidávání nových hran a nastavování parametrů pomocí módů, se jeví jako zajímavé rozšíření funkčnosti složitějších aplikací.

Díky využití knihovny *ITK*, která je obecně uzpůsobena pro práci s n-rozměrnými obrazovými daty, by případné rozšíření o práci ve 3D nemělo být příliš složité. Nicméně práce se 3D daty vyžaduje kvalitní systém vizualizace, který knihovna *ITK* neimplementuje. Bylo by tedy potřeba do nástroje zakomponovat vhodný vizualizační nástroj (např. Knihovnu *VTk* nebo *OpenGL*). Tato úprava by však byla velmi náročná a přesahuje rámec této práce, je ale výrazným námětem pro rozšíření aplikace v budoucnu.

# Literatura

- [1] Cootes, T. F.; Edwards, G. J.; Taylor, C. J.: *Active appearance models*. Springer Berlin / Heidelberg, 1998, iSSN 0302-9743.
- [2] Ibáñez, L.; Schroeder, W.; Ng, L.; aj.: *The ITK Software Guide, Second Edition*. 2005-11-21.
- [3] Kass, M.; Witkin, A.; Terzopoulos, D.: Snakes: Active Contour Models. *International Journal of Computer Vision*, 1988: s. 321 – 331.
- [4] Kulková, E.: Prehľad knižníc na podporu spracovania obrazu. 2007, fakulta matematiky, fyziky a informatiky, Univerzity Komenského v Bratislave, diplomová práca.
- [5] L. Ji, H. Y.: Loop-Free Snakes for Image Segmentation. *Proc. 1999 International Conference on Image Processing*, 1999: s. 193 – 197.
- [6] Luis Ibáñez, W. S.; Consortium, I. S.: ITK Architecture. Presentation.
- [7] Paragios, N.: Level Set Methods in Medical Image Analysis: Segmentation. Presentation.
- [8] Sethian, J. A.: Fast Marching Methods and Level-Set Methods [online]. [http://math.berkeley.edu/~sethian/level\\_set.html](http://math.berkeley.edu/~sethian/level_set.html), 2006 [cit. 2010-05-04].
- [9] WWW stránky: ITK Snap [online]. <http://www.itksnap.org/>, 2009-02-09 [cit. 2010-05-04].
- [10] WWW stránky: Level set method [online]. [http://en.wikipedia.org/wiki/Level\\_set\\_method](http://en.wikipedia.org/wiki/Level_set_method), 2010-03-12 [cit. 2010-05-04].
- [11] WWW stránky: 3DSlicer [online]. <http://www.slicer.org/slicerWiki>, 2010-05-03 [cit. 2010-05-04].
- [12] WWW stránky: Bresenham's line algorithm [online]. [http://en.wikipedia.org/wiki/Bresenham% 27s\\_line\\_algorithm](http://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm), 2010-05-04 [cit. 2010-05-05].
- [13] WWW stránky: CMake Tutorial [online]. [http://www.cmake.org/cmake/help/cmake\\_tutorial.html](http://www.cmake.org/cmake/help/cmake_tutorial.html), [cit. 2010-05-04].
- [14] WWW stránky: Doxygen [online]. <http://www.doxygen.org>, [cit. 2010-05-04].
- [15] WWW stránky: ITK Insight Toolkit [online]. <http://www.itk.org>, [cit. 2010-05-04].

- [16] WWW stránky: Qt Reference Documentation [online].  
<http://doc.qt.nokia.com/4.6/>, [cit. 2010-05-04].
- [17] Španěl, M.: Medical Data Segmentation Toolkit: A Brief Guide. 2010, presentation.
- [18] Španěl, M.; Beran, V.: Obrazové segmentační techniky, Přehled existujících metod  
[online]. <http://www.fit.vutbr.cz/~spanel/segmentace/>, 2006-01-19 [cit.  
2010-05-04].

~

# Dodatek A

## Obsah CD

Příložené CD obsahuje veškerá data, která byla vytvořena v rámci této bakalářské práce a použité knihovny. CD je rozděleno do několika složek popsaných níže v seznamu.

- **složka bin** – obsahuje přeložený binární kód segmentačního nástroje, tak že jde spustit pod systémem *MS Windows* (Testováno na *MS Windows XP* a novější)
- **složka src** – obsahuje zdrojové kódy aplikace a překládací skript pro CMake
- **složka doc** – obsahuje textovou část bakalářské práce ve formátu PDF
  - **pod složka src** – obsahuje zdrojový text technické zprávy pro program  $\text{\LaTeX}$
- **složka program\\_doc** – obsahuje programovou dokumentaci ve formátu HTML vygenerovanou programem Doxygen [14]
- **složka presentation** – obsahuje prezentaci ve formátu PDF
  - **pod složka src** – obsahuje zdrojový text prezentace pro program  $\text{\LaTeX}$
- **poster** – obsahuje plakát ve formátu JPEG tisknutelný ve velikosti A3
- **images** – obsahuje obrázky, na kterých lze otestovat funkčnost programu
- **utils** – obsahuje použité knihovny a to jak pro operační systém *Linux* tak pro *MS Windows*
  - **pod složka Qt** – obsahuje nejnovější distribuci programu *Qt*
  - **pod složka ITK** – obsahuje nejnovější distribuci knihovny *ITK*



## Dodatek B

# Manual

V této části technické správy se nacházejí užitečné informace o překladu, spuštění a práci se segmentačním nástrojem.

### B.1 Překlad a spuštění segmentačního nástroje

Program využívá knihoven *ITK* a *Qt* a vyžaduje jejich přítomnost na cílovém systému. Dále je nutné, aby systém obsahoval překladač jazyka C++, například g++. K vygenerování souboru *makefile* využívá systému *CMake*.

Systém *CMake* je *open source* multi-platformní generátor souborů *makefile*. Mimo jiné podporuje i knihovny *ITK* a *Qt*. Výsledný *makefile* soubor je sestaven na základě skriptu v souboru `CmakeLists.txt`, který se nachází v kořenovém adresáři projektu. Systém *CMake* zajišťuje bezproblémový překlad na všech platformách vybavených potřebnými knihovnami a překladači [13].

Po vygenerování souboru *makefile* se program jednoduše přeloží spuštěním programu `make` bez parametru. V případě, že je potřeba program nainstalovat do složky s programy, použije se ještě po úspěšném překladu příkaz `make install`. Pokud je potřeba vymazat všechny vygenerované soubory, použije se příkaz `make clean`.

Po přeložení se program spustí binárním souborem *2D\_level-set\_segmentation\_tool*. Po spuštění se zobrazí dvě okna. Příkazový řádek, kde se budou vypisovat chybové, případně ladící informace a samotné okno s projektem obsahující GUI aplikace. Okno s příkazovým řádkem může běžný uživatel klidně ignorovat.

### B.2 Práce se segmentačním nástrojem

Zde je jednoduchý návod jak program používat.

Grafické uživatelské rozhraní je rozděleno na několik oblastí, které usnadňují navigaci. Celé rozhraní je popsáno anglickým jazykem, především kvůli kódování a taky tomu, že angličtina je universální jazyk.

V horní části okna aplikace je pruh s vyjíždějícími nabídkami. První menu obsahuje možnost otevření souboru, smazání částí nebo celého výsledku segmentace a ukončení aplikace. Druhé obsahuje možnost zapnutí nebo vypnutí ladících výstupů a vyplňování výsledku segmentace. V posledním jsou informace o programu a krátká nápověda.

V levé části aplikace se nacházejí tři záložky. První obsahuje formulář pro automatické nastavení segmentace, druhá formulář pro manuální nastavení segmentace a třetí seznam

uživatelé přidáním hran.

Největší prostor zabírá okno se segmentovaným obrázkem. Nad tímto oknem se nacházejí tlačítka pro spuštění segmentace, spuštění krokování segmentace a zvětšení nebo zmenšení segmentovaného obrázku. Popis průběhu segmentace

Nejprve je třeba načíst segmentovaný obrázek. Aplikace podporuje formáty PNG, JPEG a BMP.

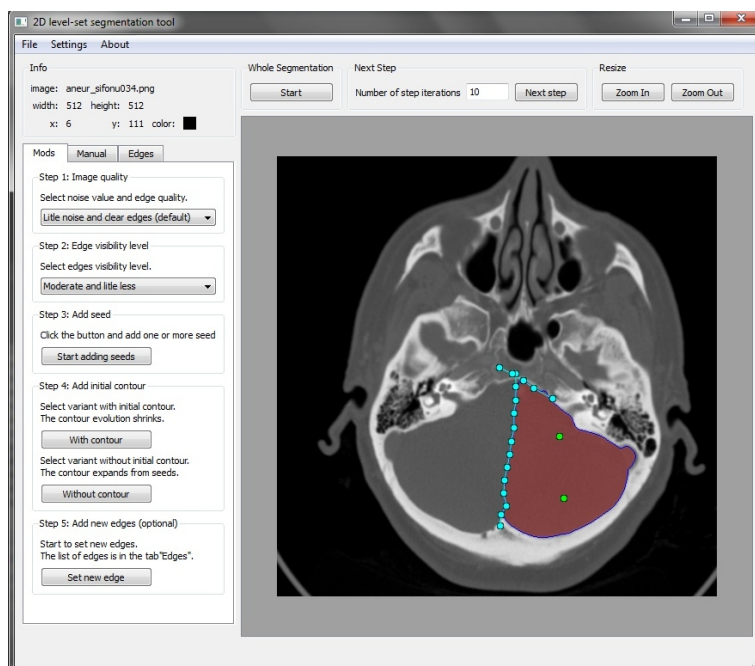
Následně je nutné zvolit mezi manuální a automatickou segmentací. Manuální je určena pro pokročilé uživatele, kteří jsou zasvěceni do dané problematiky. Automatická je určena pro běžné uživatele.

Po vybrání požadovaných módů nebo po vyplnění formuláře s parametry je nutné do obrázku vložit jedno nebo více semínek. Semínka reprezentují body, ze kterých kontura expanduje, popřípadě se do nich smršťuje. Na velké a protáhlé objekty je lepší použít více semínek. Semínka je možno upravovat metodou drag and drop.

Po zadání semínek je volitelně možno vložit inicializační konturu (je možno vložit i více než jednu inicializační konturu). V případě, že je inicializační kontura zadána bude se výsledná kontura od této smršťovat. Jednotlivé řídicí body kontury je opět možno upravovat metodou drag and drop.

Pokud je třeba z obrázku segmentovat nějaký specifický objekt, který nemá zcela jasné hrany, je možností využít přidání nových hran do obrázku. Tyto hrany lze opět editovat metodou drag and drop případně smazat. Jejich množství není nijak omezeno. Kliknutím na číslo hrany je hranu možno pro lepší orientaci zvýraznit.

Na obrázku B.1 je zobrazena ukázka segmentace.

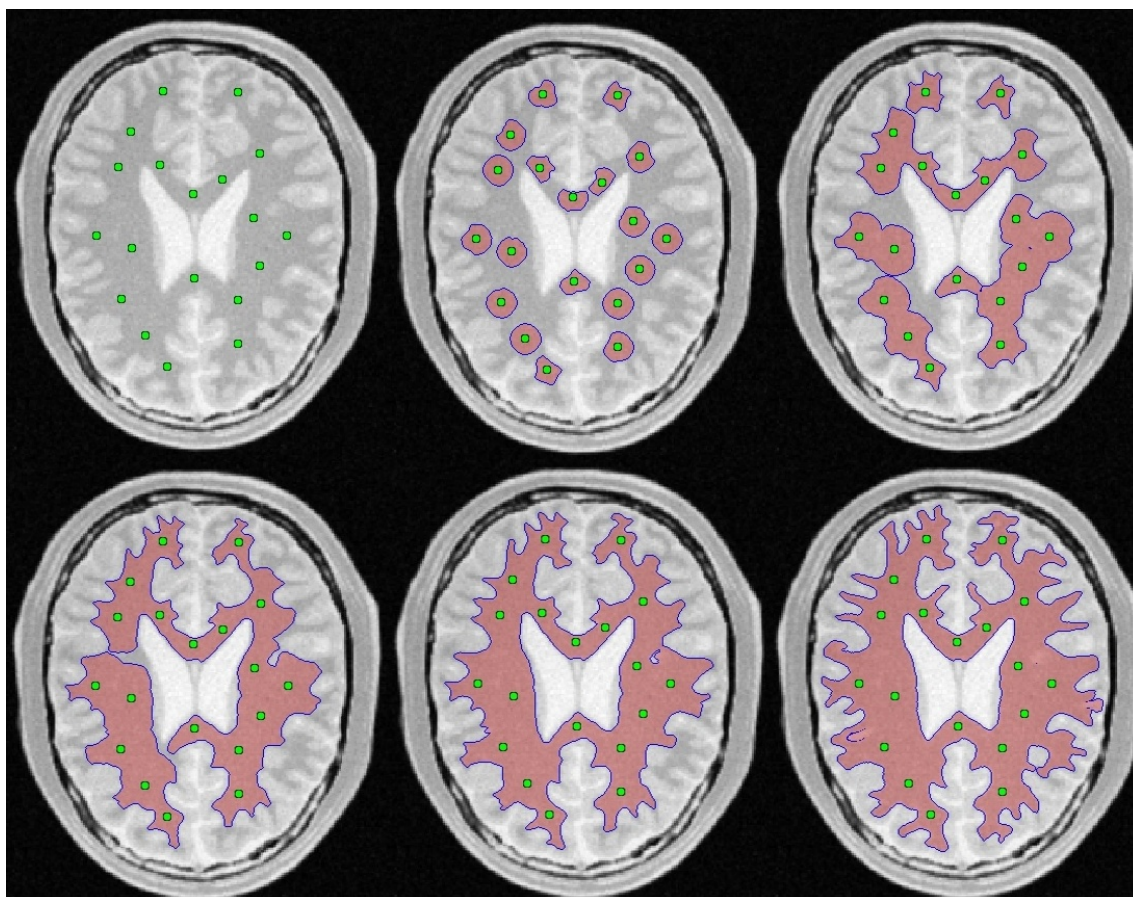


Obrázek B.1: Ukázka použití uživatelem definovaných hrak k ovlivnění *level-set* evoluce. Nové hrany je možno začít přidávat stisknutím tlačítka *Set new edge*. Jednotlivé přidání hrany se zobrazují v seznamu ve třetí záložce *Edges*. Zde je ukázka segmentace části lebeční dutiny.

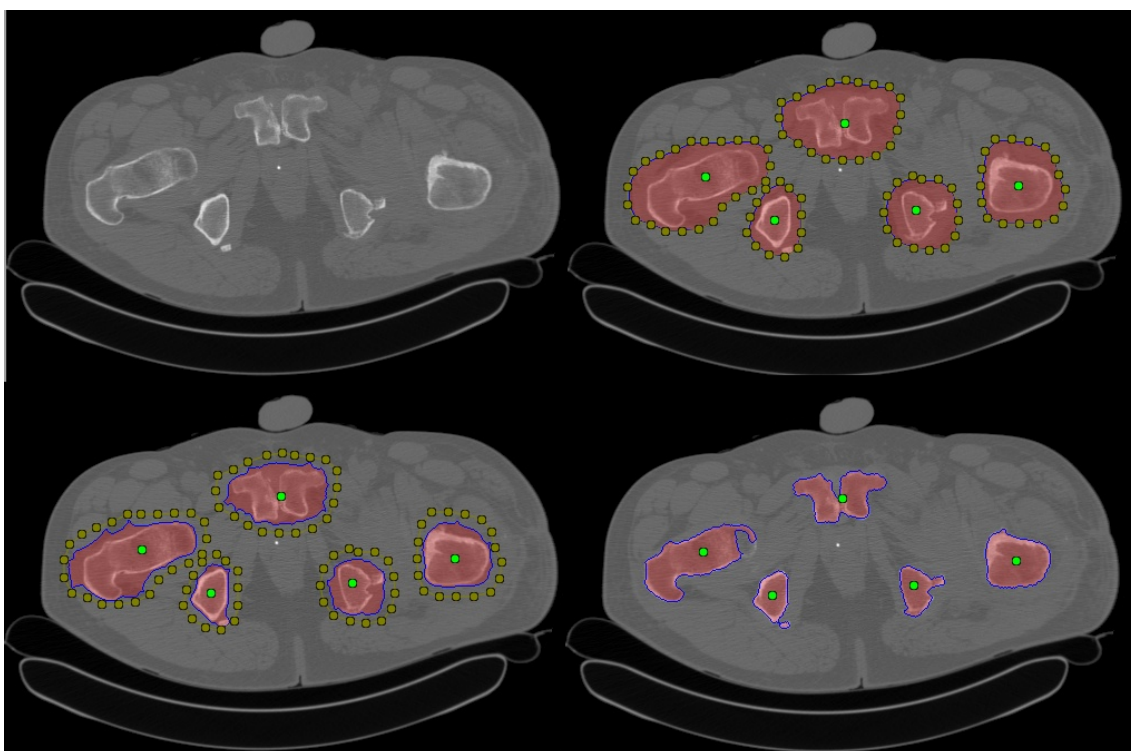
## Dodatek C

# Ukázka výstupů segmentačního nástroje

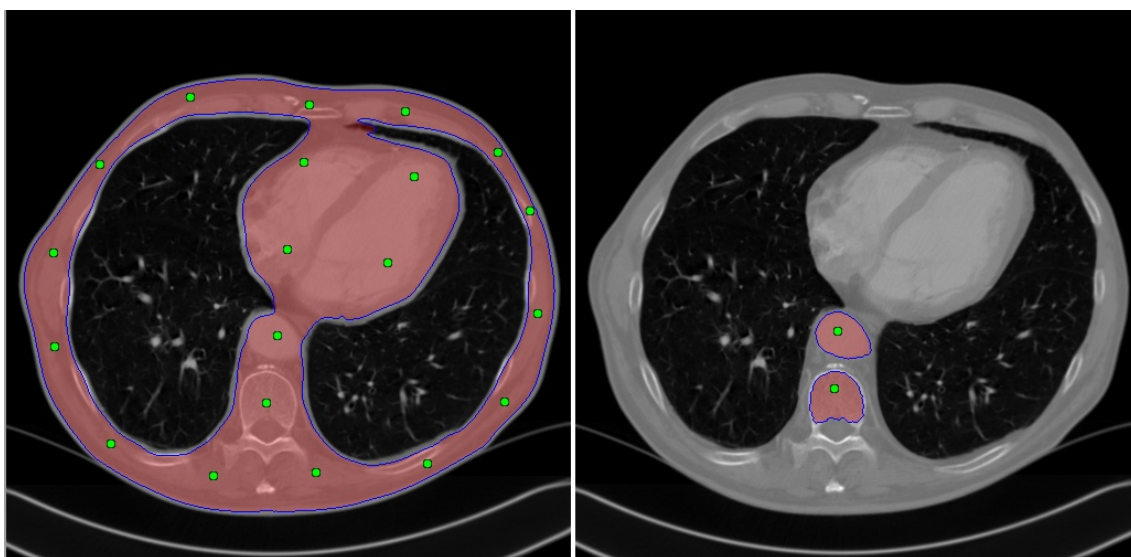
Zde jsou některé ukázky výstupů programu ve velké velikosti.



Obrázek C.1: Ukázka rozpínací evoluce *zero level-set* hladiny při segmentaci mozku na snímku z MR



Obrázek C.2: Ukázka smršťovací evoluce metody *level-set* na segmentaci pánevních kostí na snímku z CT



Obrázek C.3: Ukázka použití přednastavených módů při segmentaci různě výrazných hran a přechodů

## Dodatek D

## Plakát



# INTERAKTIVNÍ SEGMENTACE MEDICÍNSKÝCH OBRAZOVÝCH DAT

### TECHNOLOGIE

#### LEVEL-SET

- segmentační metoda
- používaná v medicínské oblasti
- numerická metoda
- velmi robustní a variabilní

#### KNIHOVNA

- pro spracování obrazových dat
- zaměřena na medicínská data
- implementuje level-set metodu
- velmi obecná a obsáhá

### CÍLE

- navrhnout segmentační nástroj
- využít ITK a level-set
- usnadnění pomocí módů segmentace
- možnost ovlivnit průběh segmentace
- přehledné GUI

### UKÁZKA EVOLUCE METODY LEVEL-SET

- při segmentaci šedé kůry mozkové



### UKÁZKA GUI

- segmentace části lebky pomocí přidávaných hran



### SEGMENTACE S VYUŽITÍM INICIALIZAČNÍ KONTURY

- ukázka segmentace pánevních kostí



Radim Kříž (xkriz03@stud.fit.vutbr.cz) - - vedoucí: Ing. Michal Španěl - - Bakalářská práce 2010

Obrázek D.1: Náhled plakátu reprezentujícího bakalářskou práci