

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**  
**FAKULTA ELEKTROTECHNICKÁ**  
**KATEDRA KYBERNETIKY**



Diplomová práce

**Program pro prohlížení a segmentaci 3D MRI dat**

Jiří Mazanec

Praha 2008

**Vedoucí diplomové práce : Dr. Ing. Jan Kybic**

Katedra kybernetiky

Školní rok: 2006/2007

## ZADÁNÍ DIPLOMOVÉ PRÁCE

**Student:** Jiří M a z a n e c

**Obor:** Biomedicínské inženýrství

**Název tématu:** 3D prohlížečka pro data MRI

### Zásady pro vypracování:

- načítání souborů ve formátu DICOM,
- prohlížení obrazu ve třech na sobě kolmých řezech,
- grafické modifikace obrazů: jas, kontrast a reprezentace v různých škálách,
- registrace obrazu založená na principu rigidní transformace,
- segmentace obrazu založená na metodě GrabCut.


### Seznam odborné literatury:

- [1] Zaplatílek, K.; Doňar, B.: MATLAB – tvorba uživatelských aplikací. BEN - technická literatura, Praha, 2005.
- [2] Hlaváč, V., Sedláček M.: Zpracování signálů a obrazů. Skriptum FEL, Vydavatelství ČVUT, 2. vydání leden 2005.

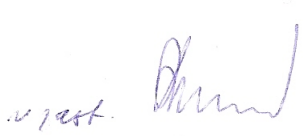
**Vedoucí diplomové práce:** Dr. Ing. Jan Kybic

**Termín zadání diplomové práce:** zimní semestr 2006/2007

**Termín odevzdání diplomové práce:** leden 2008

  
prof. Ing. Vladimír Mařík, DrSc.  
vedoucí katedry



  
prof. Ing. Zbyněk Škvor, CSc.  
děkan

V Praze dne 19.02.2007

# Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedeném příloze seznamu.

13.5.2008

---

V Praze dne

Jiří Maronec

---

Podpis

# Poděkování

Rád bych poděkoval Dr. Ing. Janu Kybicovi za odborné vedení diplomové práce a mnoho cenných rad a připomínek při její tvorbě. Dále děkuji svým rodičům, příbuzným a přátelům, kteří mi pomáhali při studiu.

## **Abstrakt**

V práci jsem vytvořit program v prostředí MATLAB schopný načítat snímky magnetické rezonance (MRI) ve formátu DICOM a zobrazit je ve třech na sobě kolmých rovinách.

Program umožňuje pomocí segmentační metody Graph Cut získat z načtených dat 2D a 3D struktury nejenom lidského mozku. Pro efektivnější zpracování dat byly využity knihovny C++. Segmentační metodu Graph Cut jsem aplikovat na syntetická i reálná data v 2D i v 3D prostoru.

Program je implementován ve třech modulech umožňujících prohlížení načtených dat, 2D segmentaci, 3D segmentaci. Všechny navržené moduly jsou integrovány v grafickém uživatelském prostředí MATLAB využívající metodu SwitchBoard Programming.

## **Abstract**

The aim of this thesis is to create a program in the MATLAB environment, which is able to process images in the DICOM-format obtained from magnetic resonance (MRI) and to display them in three orthogonal planes.

Through the use of the segmentation method Graph Cut the program can segment structures of the human brain in 2D and 3D. For more efficient data processing C++ libraries were used. Segmentation method Graph Cut was used for synthetic and real data in 2D and 3D space.

The program consists of three modules, one of which serves for the display of the data and the other two for 2D and 3D segmentation. These modules are integrated in a graphical user interface in MATLAB using the method of SwitchBoard Programming.

# Obsah

<b>1. Úvod</b>	<b>8</b>
<b>2. Magnetická rezonance</b>	<b>9</b>
2.1. Fyzikální principy magnetické rezonance	9
2.1.1. Klasický model	9
2.1.2. Kvantový model	10
2.1.3. Technická realizace	10
2.1.4. Rizika při vyšetření magnetickou rezonancí	11
<b>3. Standard DICOM</b>	<b>12</b>
<b>4. Metody segmentace</b>	<b>13</b>
4.1. Metody vycházející z detekce hran	13
4.1.1. Cannyho hranový detektor	13
4.2. Statistické metody	16
4.2.1. Prahování	16
4.2.2. Shluková analýza	17
4.3. Znalostní metody	18
4.3.1. Segmentační metoda - AAM	19
4.4. Neuronové sítě	20
4.4.1. Kohonenovy mapy	20
4.4.2. Neuronové sítě - GRBF	22
4.5. Hybridní metody	24
4.5.1. Watershed transformace	24
<b>5. Segmentační metoda Graph Cut</b>	<b>27</b>
5.1. Algoritmus Graph Cut	27
5.2. Max flow Algoritmus	29
5.2.1. Popis algoritmu Graph Cut	30
5.3. 2D Segmentace	31
5.3.1. Tvorba pravděpodobnostních modelů	32
5.3.2. Výpočet energie modelu	33
5.3.3. Nastavení pevných konstant	34
5.4. 3D segmentace	34
<b>6. Implementace prohlížečky</b>	<b>36</b>
6.1. Metoda implementace grafického prostředí	36
6.2. Načítací modul	36
6.3. Prohlížení 3D řezů	37
6.4. Grafické modifikace obrazů	38

6.5. 2D segmentační modul . . . . .	38
6.6. Kontrastní modul . . . . .	39
6.7. 3D Segmentační modul . . . . .	40
<b>7. Výsledky segmentace</b>	<b>41</b>
7.1. Testování na fantomech . . . . .	41
7.2. Výsledky 2D segmentace . . . . .	44
7.3. Výsledky 3D segmentace . . . . .	45
<b>8. Závěr</b>	<b>46</b>
<b>Literatura</b>	<b>47</b>
<b>Seznam obrázků</b>	<b>50</b>
<b>Seznam tabulek</b>	<b>52</b>
<b>Seznam zkratek</b>	<b>53</b>
<b>Přílohy:</b>	<b>53</b>
<b>A. Ovládání programu</b>	<b>54</b>
A.1. SlideViewer - 3D prohlížečka . . . . .	54
A.2. Segmentace - 2D modul . . . . .	55
A.3. Segmentace - 3D modul . . . . .	56
<b>B. Seznam funkcí</b>	<b>57</b>
B.1. Funkce Matlab . . . . .	57
B.2. Funkce Matlab - Wrapper . . . . .	58
B.3. Funkce C++ . . . . .	58

# 1. Úvod

Lidský mozek obsahuje struktury zodpovědné za určitou funkci a lokalizované do specifických oblastí. Tyto struktury se snažíme pro klinické i vědecké účely lokalizovat a to nejen ve 2D, ale i ve 3D prostoru. Zobrazování mozku magnetickou rezonancí (MRI) znamenalo velký posun v porozumění jeho struktury i funkce. Přesnější zachycení struktury, jejich objemového porovnání či posouzení dynamiky tvarových změn pomáhá nejen lépe léčit, ale i nahlédnout do nesmírně složitého fungování lidského mozku. Přesné vymezení oblasti bývá pro lékaře, ale i vědce problematické. Vzhledem k velké variabilitě a dalším faktorům zůstává každé zobrazení mozku originálem.

V dnešní době je možné se setkat s celou řadou přístrojů a zobrazovacích principů, ale ve své práci se budu zabývat pouze magnetickou rezonancí (Kap. 2). Vstupní data jsou rozdělena do souborů *DICOM* reprezentující milimetrové sagitální řezy hlavy člověka, popisem *DICOM* standardu se zabývám v kapitole 3. V lékařství se při segmentaci obrazu či objemu můžeme setkat s různými typy segmentačních metod, a proto jsem považoval za důležité, vytvořit seznam těch nejpoužívanějších na které můžeme při segmentaci obrazu v biomedicíně narazit (Kap. 4). Popis segmentačního algoritmu *Graph Cut*, jeho funkcí a implementací se zabývám v kapitole 5. Samotnou implementací grafického prostředí prohlížečky a segmentačních algoritmů se zabývám v kapitole 6. Výsledky a porovnání měření na fantomech, 2D obrázcích a 3D objemech s jinými segmentačními metodami můžeme srovnat v kapitole 7. V příloze je uveden popis ovládání programu (Kap. A) a seznam funkcí s jejich popisem (Kap. B).

Aplikace prohlížečky a moduly naprogramované v prostředí MATLAB jsou spolu s daty na přiloženém CD.



## 2. Magnetická rezonance

Magnetická rezonance (*MRI*) patří v současnosti k nejpoužívanější a nejpřesnější metodě zobrazení struktur lidského těla. Ve srovnání s jinými zobrazovacími metodami poskytuje mnohem detailnější informace o struktuře lidských tkání. Nevýhodou *MRI* je mohutná konstrukce a velká pořizovací i vyšetřovací cena, která z této diagnostické metody dělá v mnoha případech jen komplementární alternativu. Magnetická rezonance patří mezi zobrazovací metody, které dokáží lokalizovat velmi složité diagnostické problémy, bez ovlivňování buněčných struktur jakýmkoliv druhu záření [1].

### 2.1. Fyzikální principy magnetické rezonance

#### 2.1.1. Klasický model

Z hlediska klasické fyziky je atomové jádro s lichým počtem protonů a neutronů elementární koule rotující kolem své vlastní osy, která má kladný elektrický náboj. Jelikož každý pohyb elektrického náboje indukuje ve svém okolí magnetické pole, můžeme rotující kladně nabitě atomové jádro považovat za elementární magnetický dipól, jehož osa je orientována paralelně s jeho osou rotace.

Lidské tělo je z největší části složeno ze čtyř základních stavebních prvků H,C,O a N. Atom vodíku je v lidském těle zastoupen nejčastěji. Množina všech protonů vodíku v lidském těle nevykazuje jako celek, na rozdíl od jednotlivých protonů, žádný magnetický moment, neboť magnetické momenty jednotlivých protonů se díky náhodnému rozdělení směrů od osy rotace navzájem ruší. Tento stav se změní, vystavíme-li lidské tělo působení silného homogenního magnetického pole. Magnetické momenty jednotlivých spinů se nasměrují rovnoběžně k siločkovkám působícího pole, a to jak paralelně, tedy do směru vektoru působícího magnetického pole, tak i antiparalelně, tedy do směru opačného. Antiparalelní uspořádání je energeticky náročnějším stavem, a proto je antiparalelně orientovaných protonů o něco méně než protonů orientovaných paralelně s vektorem vnějšího magnetického pole. Při intenzitě vnějšího magnetického pole rovné 1,5 T odpovídá 1 000 000 antiparalelně uspořádaných protonů zhruba 1 000 007 protonů uspořádaných paralelně. Uvážíme-li, že v jednom krychlovém centimetru lidské tkáně je obsaženo zhruba  $10^{16}$  protonů, neudiví nás, že každý objemový element pacientova těla, každá částička lidské tkáně, jejíž vlastnosti vyšetřujeme, se stane díky takto vzniklé magnetické nerovnováze magnetickým dipólem s měnitelným magnetickým momentem, rotujícím kolem své osy s odpovídajícím magnetickým impulsem [1].

### 2.1.2. Kvantový model

Každý atom se skládá s jádra tvořeného protony a neutrony a z elektronového obalu, které se kolem jádra pohybují po specifických drahách. Obsahuje-li jádro lichý počet protonů a neutronů je jeho rotační impuls, který je roven rotačnímu impulsu všech jeho částí nenulový, což má za následek rotaci celého atomového jádra. Tento rotační impuls nazýváme jaderným spinem. Jaderný spin  $I$  může nabývat pouze diskretních hodnot. Každý jaderný spin vykazuje magnetický moment  $\mu$ . Mezi jaderným spinem a jeho magnetickým momentem platí vztah  $\mu = \gamma I$ , kde koeficient  $\gamma$  znamená gyromagnetickou konstantu, která je pro každé atomové jádro charakteristická (Tab. 2.1). Není-li atomové jádro vystaveno působením vnějšího magnetického pole, jsou všechny orientace jeho magnetického momentu energeticky rovnocenné. Vystavíme-li ho však působení statického magnetického pole o intenzitě  $B_0$ , které se zvýší o energie jádra a přídatnou potenciální energii. Změna energie probíhá tedy nespojitě, přeskokem mezi sousedními energetickými hladinami. Tyto diskretní energetické stavy atomového jádra se nazývají *Kern-Zeemanovy* energetické hladiny. V izolovaném homogenním magnetickém poli jsou přeskoky mezi energetickými hladinami z důvodů platnosti zákona o zachování energie zakázány. Mohou být indukovány pouze interakcí mezi tzv. magnetickou rezonancí, atomového jádra s vysokofrekvenčním elektromagnetickým signálem o frekvenci rovné frekvenci rotace magnetického momentu atomového jádra [1]. Rezonanční vlastnosti nejdůležitějších biologicky relevantních atomových jader:

Izotop	Kvantové číslo	Gyromag. konst.	Rezonanční frekvence [Hz]	Citlivost [%]
$H^1$	1/2	42,57	$26,75 \cdot 10^7$	10
$C^{13}$	1/2	10,51	$6,728 \cdot 10^7$	1,59
$F^{19}$	1/2	40,07	$25,18 \cdot 10^7$	83,34
$Na^{23}$	1/2	11,27	$7,08 \cdot 10^7$	9,35
$P^{31}$	1/2	17,25	$10,84 \cdot 10^7$	6,63

Tabulka 2.1.: Citlivost prvků v Magnetickém poli [1]

### 2.1.3. Technická realizace

Vývoj a konstrukce systému pro magnetickou rezonanci vyžaduje nejmodernější technické vybavení. Jádrem *MRI* přístroje je vysoce výkonný řídicí počítač, který umožňuje interakci s přístrojem, řídí všechny procesy během vyšetření a pomocí rychlé Fourierovy transformace (*FFT*) rekonstrukce v reálném čase celé série snímků vyšetřovaných topografických řezů. Centrální jednotkou scanneru je výkonný magnet, který vytváří silné homogenní magnetické pole. V závislosti na požadované intenzitě magnetického pole je možno užít tří typů různých magnetů, a to magnetů permanentních, supravodivých nebo odporových.

- *Permanentní magnety* jsou vhodné pro scannery s požadovanou intenzitou magnetického pole do 0,3 T. Mají velkou hmotnost, ve srovnání s ostatními dvěma typy však

poměrně nízkou pořizovací cenu.

- *Supravodivé magnety* k vyvolání supravodivosti je zapotřebí extrémě nízké teploty kolem  $-270\text{ }^{\circ}\text{C}$ , které je možné dosáhnout užitím kapalného hélia. Nákladnost zařízení se supravodivým magnetem je však více než dostatečně kompenzována možností pracovat s intenzitami v širokém rozmezí od 0,3 do 4 T.
- *Odporové magnety* pracují na elektromagnetickém principu a díky vysoké spotřebě elektrického proudu je jejich provoz vysoce nákladný. Umožňují pracovat s intenzitami do 0,5 T.

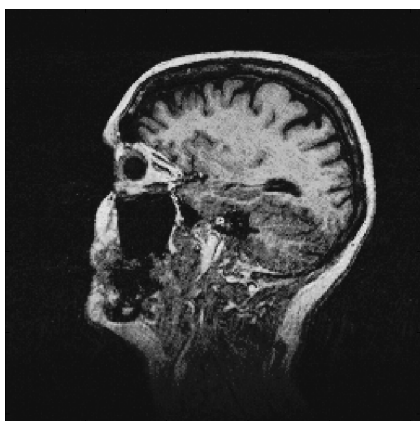
Třetí důležitou komponentou MRI systému jsou radiofrekvenční cívky, které slouží jednak jako antény vysílající elektromagnetický signál, jednak jako nejrůznější modifikátory magnetického pole v cílovém prostoru MRI systému. V závislosti na jejich funkci hovoříme o objemových, gradientních, vyrovnávacích cívkách a povrchových cívkách.

### 2.1.4. Rizika při vyšetření magnetickou rezonancí

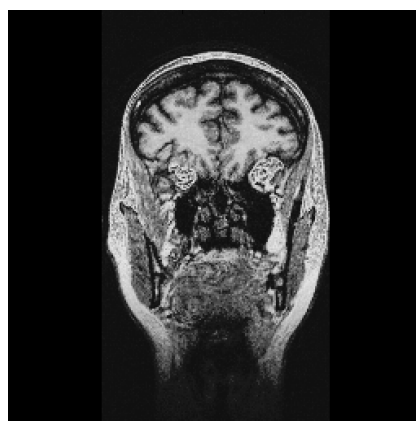
Nejen v cílovém prostoru ve kterém se nachází pacient, ale i v okolí MRI přístroje působí velice silné magnetické pole. U permanentního magnetu neustále, a u supravodivého a odporového magnetu po celou dobu provozu, tedy i v době, kdy neprobíhá žádné vyšetření. Každý kovový předmět je v magnetickém poli vystaven silám, úměrným intenzitě tohoto pole. Je tedy nutné se ubezpečit, že pacient nemá v těle žádné kovové implantáty. Malé kovové předměty vedou k znehodnocení snímku obrazovými artefakty, větší kovové předměty mohou být z těla pacienta působením magnetického pole dokonce i vytrženy. Stejná pravidla musí dodržovat i personál, který přijde do bezprostředního styku s MRI systémem. I na místě vzdáleném několik metrů od magnetu je intenzita magnetického pole tak vysoká, že dokáže přitáhnout i větší kovové předměty.

### 3. Standard DICOM

Standard *DICOM* (Digital Imaging and Communications in Medicine) byl vyvinut v National Electrical Manufacturers Association *NEMA* pro distribuci a zobrazování lékařských snímků *RTG*, *CT*, *MRI* atd. Kompletní standard je k dispozici na [3].



**Obrázek 3.1.:** Sagitální řez



**Obrázek 3.2.:** Frontální řez

Soubor ve formátu *DICOM* obsahuje hlavičku, která obsahuje informace o pacientovi, druhu snímku, velikosti obrazu a obrazová data, která mohou obsahovat informace ve třech rozměrech. Starší standardy ukládaly datové a obrazové informace do samostatných souborů. Data *DICOM* mohou být zkomprimována pro zmenšení velikosti souborů s využitím ztrátové (*Jpeg*) nebo bezztrátové komprese (*Run-Length Encoding*), identický s kompresí (*packed-bits*) v některých formátech *Tiff*. Společné uložení obrazových a identifikačních údajů prakticky odstraňuje možnost jejich vzájemné záměny nebo ztráty. První část hlavičky obsahuje formátovací informace rozměry a pomocné textové informace ke snímku. Velikost hlavičky závisí na tom, kolik informací je zde uloženo. Obrazová data jsou umístěna za hlavičkou a jsou uložena ve stejném souboru. Informace v hlavičce jsou organizovány do skupin. Skupina obsahuje parametry, vztahující se k použité zobrazovací metodě.

Zobrazovací programy přečtou tyto informace a nastaví se podle nich (je to jednodušší a úspornější, než zobrazování obrazových dat neobsahující informační části). Dále jsou uloženy informace o tom, zda obrazová data jsou komprimována, jakou metodou a s jakými parametry. Zároveň umožňují tato data správné zobrazení na různých počítačových platformách. Další část obsahuje například informace o fotometrické interpretaci (monochromatický nebo barevný snímek, stupnice šedých tónů, korekce sytosti/jasu snímku, barevná paleta *RGB* případně *CMYK* atd.) a hodnoty jasu a kontrastu.

## 4. Metody segmentace

V průběhu let bylo publikováno mnoho segmentačních algoritmů. V tomto přehledu jsem popsal nejčastěji používané segmentační metody v biomedicině včetně jejich charakteristik [8].

### 4.1. Metody vycházející z detekce hran

Lokální hrany jsou detekovány pomocí hranových detektorů na základě rozdílu hodnot okolních pixelů. Hranový detektor je algoritmus, který produkuje množinu hran (bodů, pixelů, nebo fragmentů) v obraze.

Detekce hran je jednou z nejdůležitějších oblastí nižší úrovně zpracování obrazu i přes to, že v reálných scénách je to složitý a dosud nevyřešený problém. Hranami rozumíme body obrazu, u kterých se hodnota jasu prudce mění. Hranu můžeme chápat jako vlastnost obrazového bodu započteného jako funkci obrazu v okolí tohoto bodu. Hrana je pak reprezentovaná velikostí a směrem.

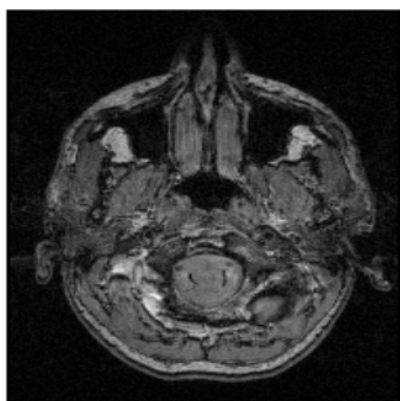
Změny či přerušení v jasu obrazu jsou jedny z nejzákladnějších charakteristik obrazu, protože naznačují fyzické rozmístění objektů v obraze. Lokální změny jasu obrazu z jedné úrovně na jinou se nazývají jasové hrany a globální změny pak jasové hraniční segmenty.

Model ideální hrany může být skoková funkce. V reálných obrazech je změna jasu postupná, nikoli skoková, takže je vhodnější použít šikmou funkci. Pokud se obě definované funkce objeví v obraze vedle sebe, vznikají ještě další dva typy hran: čára a střecha [2].

#### 4.1.1. Cannyho hranový detektor

Základní myšlenka Cannyho hranového detektoru vychází z představy, že skokovou hranu (ve 2D obrázku si ji můžeme představit jako schod) lze hledat filtrem. Návrh tohoto filtru je formulován jako úloha variačního počtu za podmínky, že budou splněny jisté požadavky na chování filtru. Detektor je optimální pro skokové hrany vzhledem ke třem kritériím:

1. Detekční kritérium požaduje, aby významné hrany nebyly přehlédnuty a aby na jednu hranu nebyly vícenásobné odezvy.
2. Lokalizační kritérium požaduje, aby rozdíl mezi skutečnou a nalezenou polohou hrany byl minimální.
3. Požadavek jedné odezvy zajišťuje, aby detektor nereagoval na jednu hranu v obraze vícenásobně. Toto očekávání je již částečně zajištěno prvním kritériem. Tento požadavek je zaměřen zejména na zašuměné a nehladké hrany, což první požadavek nezajistí.



**Obrázek 4.1.:** Originální obrázek  
*MRI* [10]



**Obrázek 4.2.:** Detekce hran Cannyho hranovým detektorem [10]

Odvození Cannyho detektoru je zdlouhavé, a proto se spokojíme s rekapitulací hlavních myšlenek:

1. Nejprve byl hranový detektor formulován pro 1D signál a první dvě kritéria optimalizace. Pomocí variačního počtu s využitím programu pro symbolické odvozování bylo nalezeno explicitní řešení.
2. Po přidání třetího kritéria, tj. zabránění vícenásobným odezvám, bylo nutné hledat optimální odezvu filtru numerickou optimalizací. Výsledný filtr lze s chybou menší než 20% aproximovat filtrací Gaussiánem.
3. Detektor hran je zobecněn do 2D. Hrana je zde dána polohou, orientací a velikostí (představme si ji jako výšku schodu). Lze ukázat, že konvoluce s 2D Gaussiánem (rov. 4.1) a derivace ve směru gradientu vytváří jednoduchý a účinný diferenciální operátor, který na rozdíl od přístupu poskytuje i orientaci hrany.

$$G(x, y | \sigma) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (4.1)$$

Nechť  $G$  je 2D Gaussián (rov. 4.1) a předpokládejme, že chceme obrázek podrobit konvoluci s operátorem  $G_n$ , který představuje první derivaci ve směru  $n$ , tj. směru gradientu,

$$G_n = \frac{\partial G}{\partial n} = n \cdot \nabla G \quad (4.2)$$

Směr gradientu není předem znám, ale existuje jeho robustní odhad. Je-li  $f$  obrázek, normála k hraně se odhaduje jako

$$\mathbf{n} = \frac{\nabla G * f}{|\nabla G * f|} \quad (4.3)$$

Pozice hrany odpovídá lokálnímu maximu konvoluce obrázku  $f$  s operátorem  $G_n$  ve směru  $\mathbf{n}$ .

$$\frac{\partial}{\partial n} G_n * f = 0 \quad (4.4)$$

V rovnici (4.4) dosadíme za  $G_n$  z rovnice (4.3) a získáme

$$\frac{\partial^2}{\partial n^2} G * f = 0 \quad (4.5)$$

Rovnice (4.5) ukazuje, jak najít lokální maxima ve směru kolmém na hranu. Tato operace se někdy nazývá potlačení odezvy mimo maxima. Jelikož konvoluce a derivace jsou asociativní operace, můžeme v rovnici (4.6) nejdříve realizovat konvoluci obrazové funkce s Gaussiánem a potom vypočítat směrovou druhou derivaci s využitím výpočtu ve směru podle rovnice. Síla hrany (velikost gradientu intenzity) se vypočte podle

$$|G_n * f| = |\nabla(G * f)|. \quad (4.6)$$

4. Vícenásobné odezvy na jedinou hranu způsobené šumem jsou v detekci hran běžným problémem. Výstup detektoru se obvykle prahuje, a tím se zvolí, která hrana je považována za významnou. Po pruhování jsou často hrany nesouvislé, ačkoliv by mnohdy měly tvořit souvislou hranici. Tuto potíž lze potlačit prahováním s hysterezí. Silné hrany s modulem gradientu nad vyšším prahem jsou považovány za hranové pixely pro dané měřítko. Osamocené slabé hrany s menším modulem gradientu než vyšší práh obvykle pocházejí z šumu. Pokud jsou ale souvislé se silnou hranou, je větší naděje, že mohou být hranovými body. Za ně jsou považovány, když odpovídající modul gradientu přesahuje nižší práh. Vyšší a nižší práh lze nastavit automaticky podle odhadnutého poměru signálu k šumu.
5. Správné měřítko pro hranový operátor záleží na velikosti objektů v obrázku. Jelikož obraz ještě není interpretován, a tudíž se neví, co jsou objekty, je potřebné vyzkoušet všechna měřítka a sdružit informaci z nich. Měřítko je dáno střední kvadratickou odchylkou Gaussiánu. Detektor může mít významnější odezvy na hranu ve více měřítkách, v tom případě je použit operátor v nejmenším měřítku, protože nejlépe lokalizuje hranu. Canny navrhl syntézu z odezvy detektoru v různých měřítkách. Nejdříve jsou označeny odezvy detektoru pro nejmenší měřítko a hrany pro větší měřítka hypotetického detektoru se syntetizují z nich. Syntetizovaná odezva se porovná se skutečnou odezvou pro příslušné větší měřítko  $\sigma$ . Hrany navíc proti syntetizovanému odhadu se zavedou jen tehdy, jsou-li silnější než předpokládala syntéza.

Cannyho hranový detektor představuje dnes používaný pokročilý hranový detektor. Běžné implementace většinou neobsahují zmíněnou syntézu odezev [2].

Název	Cannyho hranový detektor
Kategorie	Metoda hranové detekce
Vstupní data	Obrazová diskretní data
Výstupní data	Segmentovaný obraz (hodnota pixelu = index regionu)
Výhody	Jednoduchá implementace
Nevýhody	Velká citlivost na šum
Literatura	[16, 17, 18]

**Tabulka 4.1.:** Cannyho hranový detektor

## 4.2. Statistické metody

V tomto případě je základem segmentace statistická analýza obrazových dat, nejčastěji hodnot pixelů. Strukturní informace je obvykle zanedbávána.

### 4.2.1. Prahování

Distribuce úrovní šedi v obraze může pomoci určit práh pro převod obrazu do binární reprezentace - objekt, pozadí. Histogram obsahuje informace o počtu pixelů v obraze s konkrétní hodnotou šedi. Analýzou takového histogramu získáme práh  $T$ , resp. množinu prahů  $T_1, T_2, \dots, T_n$  pomocí kterých pak můžeme obraz rozdělit na podoblasti:

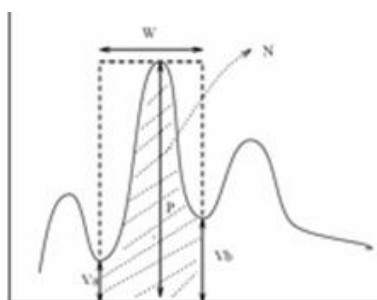
$$R(i, j) = \begin{cases} 1, & f(i, j) < T \\ 0, & \text{jinak} \end{cases} \quad (4.7)$$

resp.

$$R_k(i, j) = \begin{cases} 1, & T_{k-1} < f(i, j) < T_k \\ 0, & \text{jinak} \end{cases} \quad (4.8)$$

Získání prahu z histogramu ovšem není triviální úlohou. Histogramy reálných obrazů neobsahují ostré hrany a špičky, ale mohou mít několik vrcholů různé výšky a strmosti. K rozpoznání, jedná-li se o "dobrý" vrchol (ostrý a hluboký) můžeme použít poměr plochy vrcholu a plochy jeho obálky a tak s větší přesností určit, zda se jedná o minoritní nebo majoritní vrchol.





**Obrázek 4.3.:** Detekce vrcholu histogramu.

Na obrázku je označen histogram, kde  $P$  a  $W$  jsou výška a šířka oblasti vrcholu,  $V_a$  a  $V_b$  jsou minima okolí vrcholu a  $N$  je počet pixelů vrcholu. Následující rovnice 4.9 nám ohodnotí ostrost a výšku vrcholu (čím menší hodnota, tím lepší):

$$Q = \left(1 - \frac{V_a + V_b}{2P}\right) * \left(1 - \frac{N}{W * P}\right) \quad (4.9)$$

Výsledkem základního prahování je binární obraz, tzn. obsahuje pouze dva typy pixelů - buď "1" a nebo "0". Pro další zpracování je mnohdy důležité najít takové "1"-pixely, které jsou ve shlucích. O obecném shlukování pojednává kapitola 4.2.2. Nevýhodou histogramů je, že neobsahují prostorové informace o pixelech, takže obraz složený ze dvou obdélníků, bílého a černého, bude mít stejný histogram jako obraz složený s náhodně roztroušenými černými a bílými tečkami. Existuje též modifikace nazývaná polo-pracování. Kdy hodnoty pixelů nejsou nahrazeny "0", resp. "1", ale hodnoty větší než je práh zůstanou a ostatní se nastaví na "0" [8].

#### 4.2.2. Shluková analýza

Metoda shlukování pixelů podobných vlastností je vychází z měření provedených pro každý pixel. Každý pixel je prezentován vektorem  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  obsahujícím výsledky jednotlivých měření pro daný pixel. Měření mohou být barevné komponenty pixelu, vlastnosti okolí pixelu jako jsou střední hodnota okolních pixelů, rozptyl, apod. Je nutné navrhnout taková měření, aby pixely z jednoho segmentu byly ohodnoceny podobně a z různých segmentů rozdílně. Jinými slovy, data by měly být v pevném shluku v  $N$ -rozměrném prostoru. Mějme příklad, kdy provádíme pouze dvě měření. Úkolem segmentace je výpočet počtu shluků a přiřazení jednotlivých vektorů nejbližšímu shluku. Začneme pouze se dvěma shluky, každému přiřadíme vektory, které jsou shluku nejbliže a vypočteme, zda je nutné přidat další shluk. Pokud ano, vytvoříme nový shluk se středem v nejvzdálenějším vektoru a proces opakujeme. Nalezení kritéria pro přidání dalšího shluku ovšem není triviální. Jednou z možností je použití faktoru kvality [9], který se počítá v každém kroku a řídí počet výsledných shluků (při

dosažení maximální hodnoty faktoru proces končí). Faktor se počítá jako:  $\beta = tr\{S_w\}tr\{S_b\}$  kde  $S_w$ , resp.  $S_b$  jsou kovarianční matice vypočteny následovně:

$$S_w = \frac{1}{K} \sum_{k=1}^K \frac{1}{M_k} \sum_{x_i \in S_k} (x_i - \mu_k)(x_i - \mu_k)^T \quad (4.10)$$

$$S_b = \frac{1}{K} \sum_{k=1}^K (\mu_k - \mu_0)(\mu_k - \mu_0)^T \quad (4.11)$$

$$\mu_0 = \frac{1}{M'} \sum_{k=1}^{M'} x_i \quad (4.12)$$

kde  $K$  je počet shluků,  $M_k$  je počet vektorů přiřazených  $k$ -tému shluku,  $\mu_k$  je střední hodnota shluku,  $S_k$  je množina vektorů přiřazených  $k$ -tému shluku,  $\mu_0$  je střední hodnota všech vektorů a  $M'$  je počet pixelů určených pro shlukování [8].

Název	Shluková analýza
Kategorie	Základní statistická metoda segmentace
Vstupní data	Obrazová i objemová diskrétní data
Výstupní data	Segmentovaná diskrétní data
Výhody	Jednoduchá snadno rozšiřitelná metoda
Nevýhody	Problém stanovení cílového počtu shluků
Literatura	[9, 11]

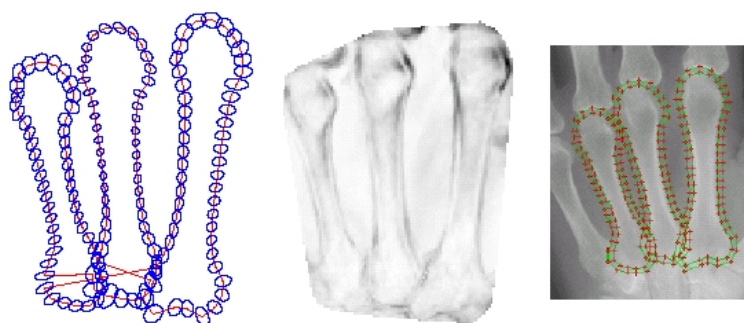
Tabulka 4.2.: Shluková analýza

### 4.3. Znalostní metody

Znalost vlastností segmentovaných objektů (tvar, barva, struktura, apod.) mohou segmentaci značně ulehčit. Metody patřící do této kategorie využívají atlas předloh či modelů segmentovaných objektů (v případě medicínských dat to může být atlas lidských tkání). Atlas je generován automaticky ze souboru trénovacích dat, nebo jsou do něj informace vloženy ručně, na základě lidské zkušenosti. V průběhu segmentace algoritmus hledá transformaci známých objektů, šablon v atlasu, na objekty nalezené v obraze. Tento proces se obvykle nazývá "atlas-warping".

### 4.3.1. Segmentační metoda - AAM

V posledních letech bylo dosaženo velmi dobrých výsledků v segmentaci medicínských i klasických obrazových dat pomocí segmentační metody AAM (AAM Active Appearance Models [14, 12]). Statistickou metodou zpracování dat, PCA (Principal Component Analysis) analýzou, je vytvořen model objektů z manuálně segmentovaných trénovacích dat. Parametry získaného modelu je možné přizpůsobit jakémukoliv novému obrazu a ověřit tak přítomnost objektu v obraze.



**Obrázek 4.4.:** Segmentace záprstních kostí pomocí AAM Zleva: nezávislá analýza hraničních bodů, analýza změn v textuře kostí a vypočtený AAM model [12].

Mezi modelované vlastnosti patří tvar objektu a intenzity pixelů. Příprava trénovacích vzorů spočívá v manuálním zadávání hraničních bodů. V průběhu trénování je zaznamenán vzájemný vztah mezi změnou polohy hraničních bodů a změnou intenzity pixelů v dané množině vzorů. Tímto způsobem natrénovaný model umožňuje velice rychlé porovnání modelu s objekty v novém obraze.

Název	Active Appearance Models - AAM
Kategorie	Speciální template-based segmentační algoritmus pracující s 2D obrazovými daty. Statistickou PCA analýzou dat je získán model segmentového objektu
Vstupní data	Ručně anotovaná trénovací množina vzorů
Výstupní data	Poloha detekovaného objektu v obraze
Výhody	Rychlá adaptace modelu na nový obraz
Nevýhody	Metoda vyžaduje poměrně přesnou počáteční inicializaci
Literatura	[13, 15, 14, 12]

**Tabulka 4.3.:** Metoda segmentace - AAM

První nevýhodou segmentace pomocí AAM je samotné trénování. Musíme totiž sestavit reprezentativní množinu vzorů, což není jednoduché. Další problém je, že vzory v trénovací množině musíme ručně anotovat. V případě rozsáhlé množiny dat je taková anotace velmi

časově náročná. Nepříjemnou vlastností *AAM* algoritmu je zvýšená možnost selhání při porovnávání modelu s novým obrazem. Velmi důležitá je proto poměrně přesná inicializace, nebo-li odhad polohy objektu v testovaném obraze [8].

### 4.4. Neuronové sítě

Většina metod segmentace obrazu je založena na znalostech a zkušenosti. Opakem je segmentace obrazu neuronovými sítěmi (dále jen *NN* - Neural Network), která není založena na podobných meta-pravidlech. Trénování *NN* čistě orientované na data probíhá podle principu "učení příklady". Na druhé straně leží algoritmy analyzující data vzhledem k zadané množině pravidel a příznaků.

V zásadě existují dvě strategie trénování umělé *NN*. První přístup hledá charakteristické vlastnosti vstupních dat (obvykle příznakové vektory) a klasifikuje je do tříd bez jakékoliv další interpretace. Tento přístup nazýváme učení bez učitele a setkali jsme se s ním v podobě trénování *SOM* neuronových sítí v kapitole 4.4.1 o shlukové analýze obrazu a statistických metodách segmentace.

Druhý přístup je trénování s učitelem, vyžaduje ručně segmentovaná trénovací data. Vstupem učícího algoritmu jsou nejen příznakové vektory, ale i funkce, která každému vstupnímu vektoru přiřazuje určitý segment obrazu.

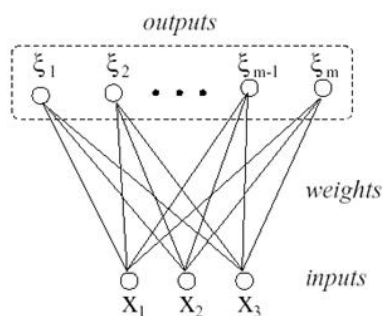
#### 4.4.1. Kohonenovy mapy

Kohonenovy mapy [19] jsou neuronové sítě typu *SOM*. Využívají soutěžní strategie učení (competitive learning). Principem tohoto modelu je, že výstupní neurony sítě spolu soutěží o to, který z nich bude aktivní. V určitém čase je tedy aktivní pouze jeden neuron. Důležitou vlastností těchto sítí je shlukování. Vstupy sítě jsou tříděny do skupin dle vítězného (aktivního) neuronu.

Každý neuron výstupní vrstvy je v Kohonenově mapě propojen vazbou se všemi neurony vrstvy vstupní Obr. 4.5. Charakteristickou vlastností každé vazby je její váha  $w$ . Index vítězného neuronu pak odpovídá číslu segmentu v obraze. Vstupem sítě může být jas pixelu, případně další příznaky extrahované z obrazu.

Soutěžení výstupních neuronů spočívá ve výpočtu vzdálenosti vektoru vah každého neuronu od vstupního neuronu. Neuron s nejnižším výstupem je prohlášen za vítěze a právě jeho index  $c$  je pro nás podstatný. Vzdálenost obou vektorů lze spočítat jako Eukleidovskou vzdálenost:

$$c = \operatorname{argmin} D(x - w_i) \quad (4.13)$$



Obrázek 4.5.: Architektura 1D Kohonenovy mapy

Učení sítě probíhá následujícím způsobem. Postupně procházíme celou tréninkovou množinu a při předložení jednoho tréninkového vzoru dochází ke konkurenci neboli soutěži neuronů. Po vyhodnocení vítězného neuronu jsou pak upravovány váhy nejen tohoto neuronu, ale i neuronů v jeho okolí. Smyslem takového postupu je posunutí váhových vektorů neuronů v okolí vítěze směrem k aktuálnímu vstupu tak, aby tyto neurony ještě více vylepšily svou pozici vůči novému tréninkovému vzoru. Samotná adaptace vah se pak řídí podle vztahu: kde  $c$  je index vítězného neuronu a jsou parametry učení. Reálný parametr určuje míru změny vah, na počátku učení je blízký jedné a postupně se zmenšuje. Velikost s okolí byla na počátku učení velká (srovnatelná s velikostí sítě) a postupně klesá až na hodnotu "0". Tento způsob učení je označován také jako učení bez učitele.

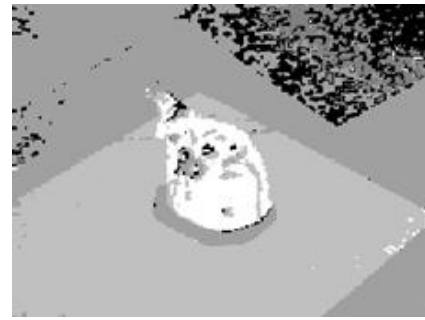
$$w_t^{(c)} = \begin{cases} w_t^{(c)}, \alpha(x_k - w_t^{(c)}), c \in N_s \\ w_t^{(c)}, \text{jinak} \end{cases} \quad (4.14)$$

kde  $c$  je index vítězného neuronu a  $\theta$  a  $s$  jsou parametry učení. Reálný parametr  $0 < \theta \leq 1$  určuje míru změny vah, na počátku učení je blízký jedné a postupně se zmenšuje. Velikost s okolí byla na počátku učení velká (srovnatelná s velikostí sítě) a postupně klesá až na hodnotu "0". Tento způsob učení je označován také jako učení bez učitele.

Jedním z problémů většiny clusterovacích algoritmů je počet shluků a tedy i počet výstupních neuronů. Úspěšnost těchto metod značně závisí na složitosti vstupu (počtu shluků ve vstupu) a zvoleném počtu shluků. Větší počet neuronů znamená lepší rozlišovací schopnosti, bohužel při nižším počtu shluků ve vstupním obraze jen obtížně nalezneme odpovídající správně segmentovaný obraz. Toto platí i pro opačný případ, kdy počet shluků ve vstupním obraze je vysoký, avšak síť se snaží třídít pixely do příliš malého počtu shluků. Řešením tohoto problému je upravit učící algoritmus sítě tak, aby byl schopen přidávat a případně odebírat neurony ve výstupní vrstvě dle potřeby [20, 23, 8].



Obrázek 4.6.: Originál [22]



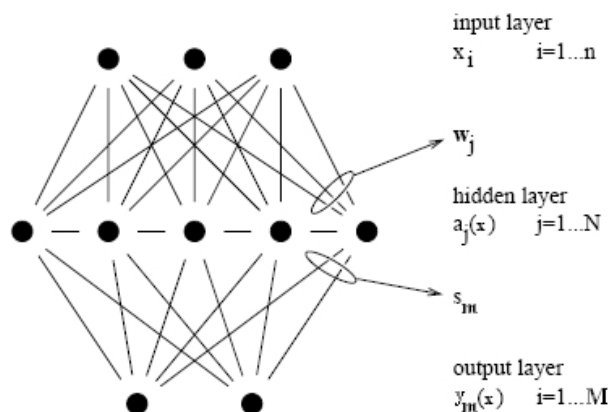
Obrázek 4.7.: Segmentace Kohonenovými sítěmi [22]

Název	Kohonenovy mapy (neuronové sítě typu SOM)
Kategorie	V základní variantě jednoduchá statistická metoda segmentace
Vstupní data	Obrazová i objemová diskrétní data
Výstupní data	Segmentovaná diskrétní data
Výhody	Jednoduchá snadno rozšiřitelná metoda
Nevýhody	Pevný počet neuronů ve skryté vrstvě = pevný počet segmentů
Literatura	[19, 21, 22, 20, 23]

Tabulka 4.4.: Kohonenovy mapy

#### 4.4.2. Neuronové sítě - GRBF

GRBF (Generalized Radial Generalized Basis Functions) neuronové sítě [24, 25] spojují výhody obou přístupů jak učení bez učitele, tak učení s učitelem.



Obrázek 4.8.: Struktura neuronové sítě s radiální bázovou funkcí [24]

Architektura GRBF sítě je třívrstvá (Obr. 4.8). Skládá se ze vstupní vrstvy, jedné skryté vrstvy a výstupní vrstvy. Propojení neuronů mezi vrstvami je úplné. Každý neuron skryté

vrstvy je spojen se všemi neurony vstupní i výstupní vrstvy pomocí synaptických vah. Vstupní vrstva neuronů slouží pouze k přiložení klasifikovaného příznakového vektoru  $x$  na vstup sítě. Hodnoty přiloženého vektoru propaguje do skryté vrstvy, kde jsou použity neurony s radiální bázovou funkcí *GRBF* [19]. Vektor vah  $w_j$  neuronu ve skryté vrstvě lze interpretovat jako bod v  $n$ -rozměrném prostoru hodnot vstupního vektoru. Aktivační funkce skrytého neuronu je obdobná jako u dříve popisovaných soutěživých *SOM* sítí. Určí se vzdálenost (nejčastěji eukleidovská) vektoru vah  $w_j$  neuronu od příznakového vstupního vektoru  $x$ :

$$d_j(\vec{x}) = \|\vec{x} - \vec{w}_j\| \quad (4.15)$$

Výsledná aktivační funkce *GRBF* neuronu má charakter Gaussovy funkce, např.

$$a_j(\vec{x}) = e^{-\frac{d_j^2(\vec{x})}{2\rho_j^2}} \quad (4.16)$$

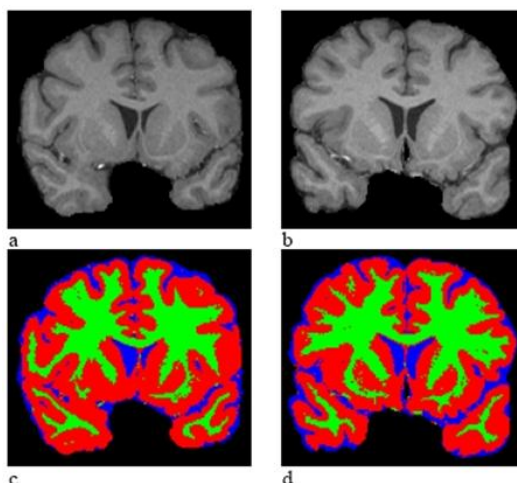
kde  $\rho$  je směrodatná odchylka Gaussovy funkce - parametr skrytého neuronu. V některých případech se používá normalizovaný tvar aktivační funkce [25]. Kompetitivní chování skryté vrstvy je zřejmé. Neuron, jehož vektor vah je nejbližší vstupnímu příznakovému vektoru má nejvyšší hodnotu aktivační funkce. Výstupy neuronů skryté vrstvy jsou lineárně propagovány do vrstvy výstupní. Vektor vah  $s_m$  výstupního neuronu slouží pro výpočet vtažované sumy aktivačních funkcí skrytých neuronů:

$$y_m(\vec{x}) = \sum_{j=1}^N s_m a_j(\vec{x}) \quad (4.17)$$

Tato struktura výstupní vrstvy odpovídá velmi známé architektuře NN zvané perceptron [19]. Trénování *GRBF* sítě probíhá ve třech fázích. První dvě probíhají automaticky bez učitele, třetí fáze obvykle s učitelem.

1. V první fázi jsou ustanoveny váhy  $w_j$  mezi vstupní a skrytou vrstvou. Pro tento účel se používají algoritmy shlukové analýzy (Kap. 4.4.1).
2. Jakmile jsou hodnoty vah  $w_j$  známy, můžeme určit parametry skrytých neuronů (směrodatné odchylky). Nejčastěji se používají jednoduchá heuristická pravidla.
3. V poslední fázi jsou určeny váhy mezi skrytou a výstupní vrstvou sítě. Nejvýhodnější je minimalizace celkové chyby neuronové sítě pomocí metody gradientního sestupu. Detailní informace ohledně architektury a trénování *GRBF* neuronových sítí naleznete v již citované knize [25].

Vlastnosti *GRBF* neuronových sítí jejich autoři ověřovali na automatické segmentaci šedé a bílé kůry mozkové v *MRI* snímcích. Svoje výsledky hodnotí jako velmi povzbudivé, avšak stále jsou tu problémy, které je nutné řešit [8].



**Obrázek 4.9.:** Výsledek plně automatické segmentace šedé a bílé kůry mozkové v MRI řezu pomocí GRBF neuronové sítě [24]

Název	Neuronové sítě - GRBF
Kategorie	Pokročilý hybridní segmentační algoritmus založený na neuronových sítích s radiální bázovou funkcí.
Vstupní data	Obrazová i objemová diskrétní data.
Výstupní data	Segmentovaná data.
Výhody	Spojuje výhody dvou druhů algoritmů pro trénování neuronové sítě - s učitelem i bez učitele.
Nevýhody	Ručně anotovaná trénovací data
Literatura	[24, 25, 19]

**Tabulka 4.5.:** Neuronové sítě - GRBF

## 4.5. Hybridní metody

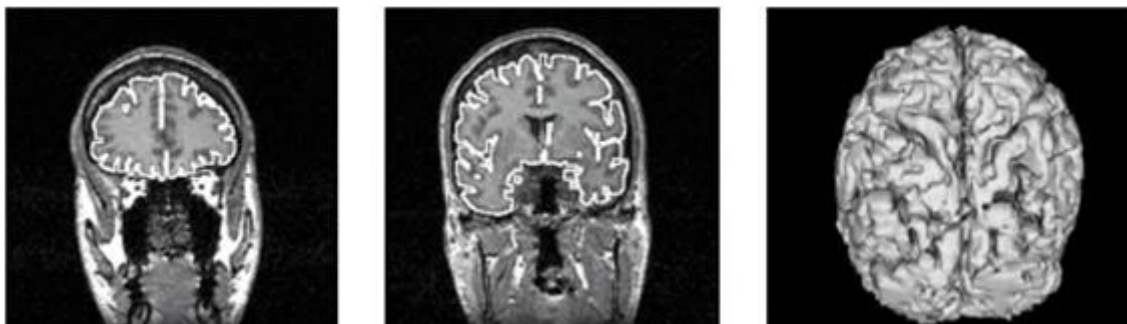
Některé segmentační techniky je těžké zařadit do jedné z předchozích kategorií, protože obsahují prvky každé z nich. Mluvíme tedy o tzv. hybridních metodách. Mezi hybridní řadíme také metody založené na matematické morfologii. Jedná se o skupinu metod, která pro segmentaci využívá matematických charakteristik obrazu, např. průběh gradientu.

### 4.5.1. Watershed transformace

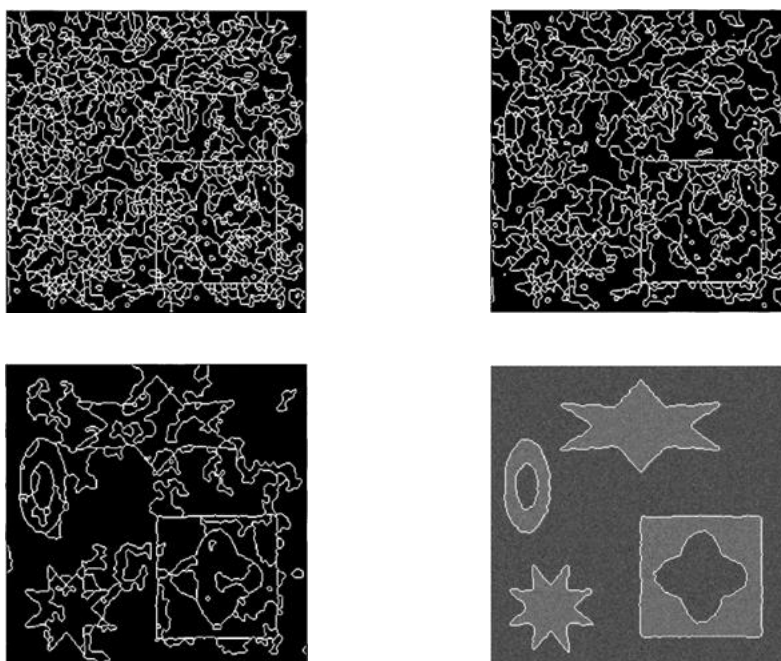
Watershed transformace (watershed = rozvodí, povodí či vodní předěl [17]) je možné zařadit mezi segmentační přístupy založené na propojování regionů. Tato morfologická metoda segmentace je postavena na myšlence pocházející z geografie. Obraz je chápán jako terén nebo topografický reliéf, který je postupně zaplavován vodou. Povodí jsou z počátečních bodů



(lokálních minim obrazu) zaplňována vodou. V místech, kde by se voda ze dvou různých povodí mohla slít jsou vytvořeny hráze. Proces postupného zaplavování je zastaven ve chvíli, kdy dosáhneme nejvyššího bodu terénu (maxima obrazu). Výsledkem je obraz rozdělený do regionů, jednotlivých povodí oddělených hrázemi. Vzniklé hráze jsou nazývány *watershed lines*, nebo jednodušeji *watershed* [8].



**Obrázek 4.10.:** Výsledek watersheds segmentace lidského mozku a jeho 3D rekonstrukce. Obrázky pocházejí z článku [17].



**Obrázek 4.11.:** Výsledek watershed transformace (vlevo nahoře) a postupné spojování vzniklých regionů metodou popsanou v článku [16], odkud pocházejí i použité obrázky.

Výraz watershed transformace znamená označení všech pixelů obrazu tak, že všechny body daného povodí jsou označeny stejným unikátním indexem. Speciální index, odlišný od všech ostatních, je přiřazen všem bodům tvořícím hráze - watersheds. Příklad segmentace

Název	Watershed transformace
Kategorie	Pokročilý hybridní segmentační algoritmus pracující s 2D obrazovými daty. Metoda morfologické segmentace
Vstupní data	Obrazová diskrétní data
Výstupní data	Segmentovaný obraz (hodnota pixelu = index regionu)
Výhody	Osvědčená a efektivní metoda segmentace mozku v medicínských aplikacích
Nevýhody	Je-li obraz zašuměný, produkuje watershed transformace velký počet regionů
Literatura	[16, 17, 18]

**Tabulka 4.6.:** Watershed transformace

medicínských obrazových dat touto metodou je znázorněn na Obr. 4.10. Metoda produkuje velký počet regionů. Výsledkem je nadměrná segmentace obrazu. Obvyklým řešením je proces spojování regionů, jenž následuje po transformaci. Regiony patřící do stejného segmentu či obrazové struktury jsou spojovány (Obr. 4.11) [16, 18].

## 5. Segmentační metoda Graph Cut

Interaktivní metody segmentace se stávají stále použitelnějšími a to právě díky možnosti uživatele zasáhnout do procesu segmentace. *Graph Cut* metoda umožňuje interaktivní segmentaci, která rozděluje obraz do dvou oblastí *předmět* a *pozadí*. Tato metoda je použitelná také v 3D prostotu.

### 5.1. Algoritmus Graph Cut

Grafové algoritmy patří mezi významné metody segmentace obrazu. Hlavní myšlenkou grafových metod segmentace je vytvořit z obrázku ohodnocený graf  $G = (V, E)$  s množinou uzlů  $V$  a množinou hran  $E$ , které spojují jednotlivé uzly grafu. Uzly  $v \in V$  odpovídají jednotlivým pixelům v obraze a hrany  $\langle v_i, v_j \rangle \in E$  spojují vždy sousední uzly v závislosti na tom jaký zvolíme způsob propojení sousedů. Pixel v klasickém 2D obrázku může mít buď 4 nebo 8 sousedních pixelů. Každý uzel  $v$  a hrana  $\langle v_i, v_j \rangle \in E$  má cenu v závislosti na *předmětu*, který se má segmentovat.

Nechť množina všech pixelů obrázku je označena jako  $I$  a množina  $N$  všech pixelů  $(p, q) \in I$  reprezentuje sousední pixely. Nechť  $O$  je množina všech pixelů patřících do *předmětu*,  $B$  je množina všech pixelů patřících do *pozadí*. Nechť každý pixel obrázku  $i_k$  patří do jedné z binárních tříd  $L_k \in \{O, B\}$ , kde  $O$  a  $B$  reprezentují množiny pixelů *předmětu* a *pozadí* (5.3.1). Potom  $L = (L_1, L_2, \dots, L_{|I|})$  definují výslednou segmentaci.

Metodu je možné inicializovat interaktivně nebo automatickou identifikací jednoho nebo více bodů reprezentující *předmět* nebo jednoho nebo více bodů reprezentující *pozadí*. Tyto body nebo množiny bodů se nazývají zrna a slouží jako speciální konstanty nazývané *pevné konstanty*. Zbývající konstanty nazývané *měkké konstanty* reprezentují pixely ve zbytku obrázku.

K nalezení požadovaného výsledku použijeme, stejně jako ostatní globální optimalizační grafové vyhledávací techniky, kriteriální funkci (rov. 5.1). K minimalizaci kriteriální funkce  $C(L)$  (rov. 5.1) je použit speciální druh grafu  $G_{st} = (V \cup \{s, t\}, E)$ . K tomuto grafu  $V$ , jehož uzly odpovídají pixelům obrázku  $I$ , jsou navíc připojeny dva speciální koncové uzly  $s$  a  $t$ . Tyto koncové uzly jsou spojeny s uzly grafu reprezentující pixely obrázku a to tak, že ke každému uzlu grafu existuje právě jedna hrana spojující tento uzel s koncovým uzlem  $s$  a právě jedna hrana spojující tento uzel s koncovým uzlem  $t$  (Obr. 5.1). Hrany  $E$  v grafu  $G_{st}$  jsou klasifikovány do dvou kategorií:  $n$  – hrany a  $t$  – hrany. Množina  $n$  – hran spojuje dvojice sousedních pixelů, jejichž cena je odvozena z hraničního členu  $B(L)$  (rov. 5.1). Množina hran  $t$  – hran spojuje uzly grafu reprezentující pixely obrázku a koncové uzly, jejichž celková cena je dovozena z  $R(L)$  (rov. 5.1). Řez  $s - t$  grafem  $G_{st}$  je množina hran, které tento řez protíná a rozděluje tento graf na dvě disjunktní podmnožiny  $S$  a  $T$  (Obr. 5.1), takže  $s \in S$  a  $t \in T$  a přitom nesmí existovat přímá cesta, která by spojovala koncové uzly  $s$  a  $t$ . Minimální celková

cena řezu se nazývá minimálním řezem grafu (Obr. 5.2). Nalezení minimálního řezu grafem je problém, který se dá převést na klasický optimalizační problém nalezení maximálního toku (maximum flow algoritmus).

$$C(L) = \lambda R(L) + B(L), \quad (5.1)$$

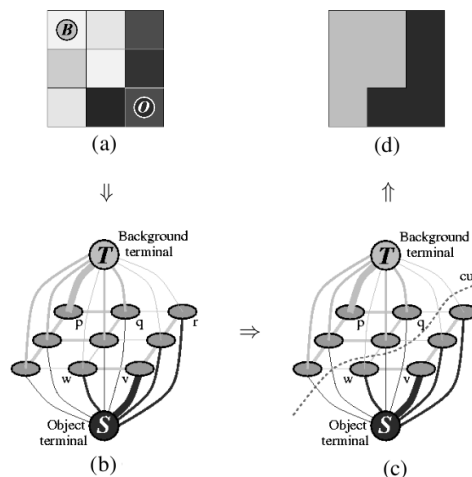
kde

$$R(L) = \sum_{p \in R_p} R_p(L_p) \quad (5.2)$$

$$B(L) = \sum_{(p,q) \in N} B_{(p,q)} \delta(L_p, L_q) \quad (5.3)$$

a

$$\delta(L_p, L_q) = \begin{cases} 1, & L_p \neq L_q \\ 0, & \text{jinak} \end{cases} \quad (5.4)$$



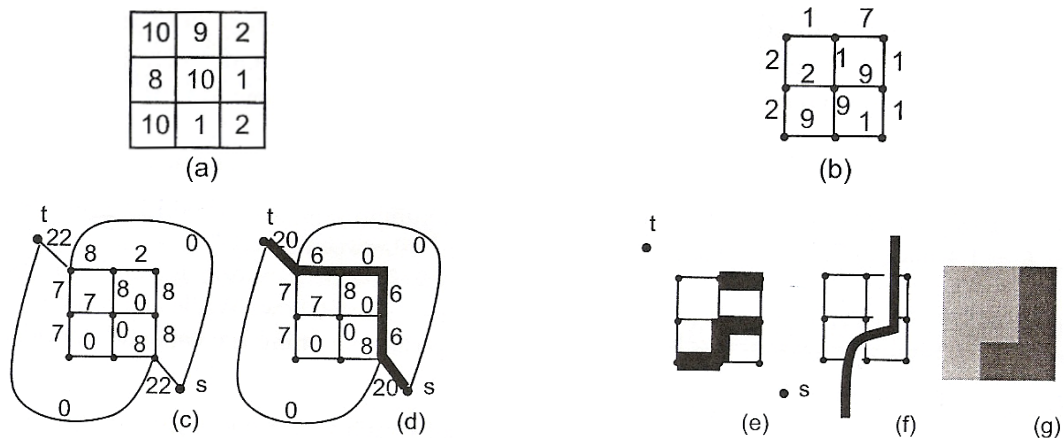
**Obrázek 5.1.:** Příklad 2D segmentace obrázku o velikosti 3x3. Cena každé hrany je reprezentována tloušťkou spojnice. Člen  $R(L)$  spolu s pevnými konstantami definují cenu  $t$ -hran a člen  $B(L)$  definuje cenu  $n$ -hran [4]

Kde  $C(L)$  je celková cena odpovídající aktuální segmentaci  $L$ , můžeme se setkat také s pojmem energie. Člen  $R(L)$  odpovídá součtu  $t$ -hran rozdělených segmentací  $L$  do dvou disjunktních množin (Obr. 5.1 c). Cena  $t$ -hran je vyjádřena členem  $R_p$  (Kap. 5.3.1). Člen  $B(L)$  je ohodnocení nespojitosti (rozdílu intenzit) mezi sousedními pixely  $p, q$  v daném uspořádání sousedů (4 sousedi nebo 8 sousedů pro 2D obrázek). Člen  $B_{(p,q)}$  také nabývá velkých hodnot jsou-li pixely  $(p,q)$  stejné a malých hodnot jsou-li pixely  $(p,q)$  rozdílné. Parametr  $\lambda$  určuje poměr členu  $R(L)$  vůči členu  $B(L)$ , čímž můžeme ovlivňovat významnost vyznačených oblastí

oproti vlastnostem hranice oblastí. Je-li tedy  $\lambda$  rovna nule nebude se člen  $R(L)$  uvažovat a výsledná segmentace bude závislá pouze na vlastnostech členu  $B(L)$ , tudíž na hranici, která je výsledkem minimalizace  $C(L)$  (rov. 5.1) definované *max flow* algoritmem.

## 5.2. Max flow Algoritmus

Problém získání minimálního řezu je řešen algoritmem nalezením maximálního toku (max flow algoritmus) z koncového uzlu  $s$  do koncového uzlu  $t$ . Jedná se o problém, který se dá převést na kombinatorickou optimalizační úlohu, která hledá maximální "průtok vody" přes jednotlivé hrany grafu, kde cena jednotlivých hran určuje množství vody, které proteče vždy maximem jednotlivých hran grafu. Maximální průtok je ohodnocení množiny hran grafu mezi koncovými uzly  $s$  a  $t$ , která odpovídá nalezení minimálního řezu grafu. Existuje mnoho algoritmů, které řeší tuto optimalizační úlohu (5.3.1).



**Obrázek 5.2.:** (a) - originální data odpovídající (obrázku 5.1) (b) - Cena hran, rozdíl intenzit pixelů obrázku (4 sousedi) (c) - vytvoření  $G_{st}$  grafu dle (tabulky 5.1) (d) - Zbytkový (reziduální) graf znázorňující nejkratší cestu (e) - zvýraznění satureovaných hran (f) -nalezení minimálního řezu grafu (g) - výsledná segmentace [27]

Proces je inicializován s nulovým tokem, to znamená, že neexistuje přímé spojení mezi koncovými uzly  $s$  a  $t$  (Obr. 5.2 c). V průběhu algoritmu vedoucího k maximalizaci toku se aktuální stav toku průběžně udržuje v podobě zbytkového (reziduálního) grafu  $G_f$ , kde  $f$  je aktuální tok (Obr. 5.2 d). Dokud je topologie grafu stejná s topologií grafu  $G_{st}$ , algoritmus pokračuje v rozšiřování zbytkového grafu vzhledem k aktuálnímu stavu toku. Při každé iteraci algoritmu se hledá nejkratší cesta  $s \rightarrow t$  podél neprošlých hran zbytkového grafu. Tok podél této cesty se zvyšuje až do stavu, kdy je kapacita některé z hran satureovaná. Zvýší-li se tok grafu o  $\Delta f$ , sníží se tok zbytkového grafu  $\Delta f$ . V každém kroku se tedy zvyšuje celkový tok

z koncového uzlu  $s$  do koncového uzlu  $t$  až do stavu, kdy neexistuje žádná cesta z nenasaturovanými hranami  $s \rightarrow t$ . Bylo tedy dosaženo maximálního toku a optimalizační problém je vyřešen (Obr. 5.2 f). Oddělením  $S$  a  $T$  grafových uzlů je potom definovaná segmentace - minimální řez  $s \rightarrow t$  odpovídá saturevaným hranám grafu (Obr. 5.2 e).

Hrana	cena (cost)	pro oblasti
$p, q$	$B_{\{p,q\}}$	$p, q \in N$
$\{s, p\}$	$\lambda.R_p(\text{"pre"})$	$p \in I, p \notin (O \cup B)$
	$K$	$p \in O$
	$0$	$p \in B$
$\{p, t\}$	$\lambda.R_p(\text{"poz"})$	$p \in I, p \notin (O \cup B)$
	$K$	$p \in O$
	$0$	$p \in B$

**Tabulka 5.1.:** Přiřazení odpovídajícího ohodnocení hrany vytvořeného grafu  $G_{st}$

Tabulka 5.1 popisuje inicializační stav algoritmu *GraphCut*. Počáteční ohodnocení  $n$ -hran i  $t$ -hran a definuje speciální konstanty nazývané *pevné konstanty*, které dodefinují ohodnocení  $t$ -hran (Kap. 5.3.3). Konstanta  $K \geq 1 + \max_{p \in I} \sum_{q: \{p,q\} \in N} B_{\{p,q\}}$  definuje minimální hodnotu koncové  $t$  - hrany jako součet všech  $n$  - hran vycházejících právě z tohoto uzlu v grafu  $G_{st}$ , tato hodnota definuje *pevné konstanty* množině pixelů *předmětu* nebo *pozadí* matice  $R_p$ , která definuje cenu  $t$  - hran v grafu  $G_{st}$ .

### 5.2.1. Popis algoritmu Graph Cut

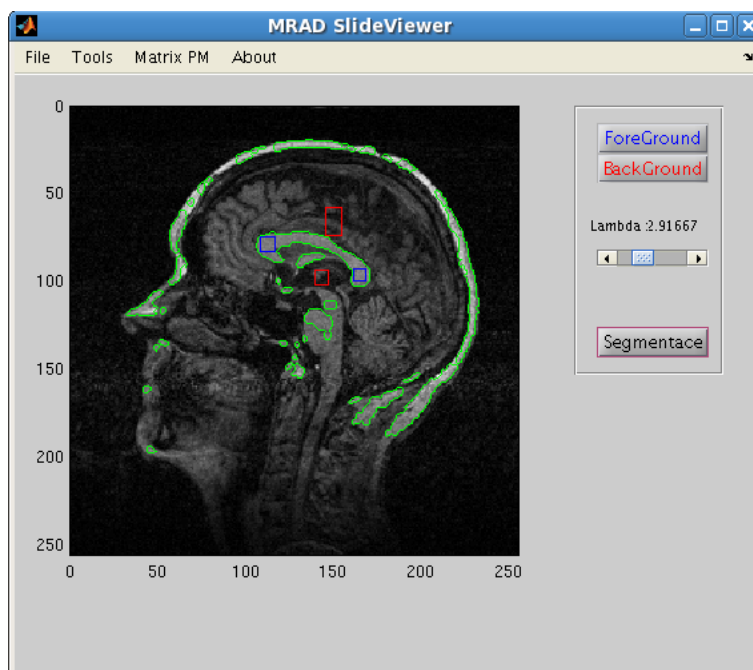
1. Vytvoří ohodnocený orientovaný graf odpovídající velikosti a dimenzi obrázku (objemu), který má být segmentován.
2. Získání *předmětu* a *pozadí* jako oblastí, které byly vyhodnoceny jako *předmět* nebo *pozadí*. Vytvoření dvou speciálních uzlů  $s$  - hran a  $t$  - hran, které jsou napojeny na všechny pixely obrázku.
3. Přiřazení odpovídající ohodnocení hrany každého linku vytvořeného grafu dle tabulky 5.1.
4. Použití algoritmu *max flow*, který je optimalizovaný pro získání minimálního řezu vytvořeného grafu.
5. Výsledný řez grafu, který je minimálním řezem vytvořeného grafu oddělujícím *předmět od pozadí*.

Předpokládejme, že uživatel inicializovat počáteční stav a po segmentaci není spokojený s výsledkem. Uživatel může zlepšit segmentaci přidáním doplňkových dalších oblastí

a tím vylepšit výsledek segmentace. V mé implementaci algoritmu nepracuji s předcházejícím výsledkem segmentace, ale při každé interakci uživatele je algoritmus spouštěn z novou inicializací. Samozřejmě je možné vycházet z předcházejícího výsledku segmentace, ale v mém případě to nebylo nutné.

### 5.3. 2D Segmentace

Implementaci segmentační metody můžeme rozdělit do následujících kroků, vytvoření pravděpodobnostních modelů z označených oblastí *předmětu a pozadí* (Obr. 5.3). Z těchto oblastí (množiny pixelů) vytvoříme buď histogramový nebo Gaussovský odhad pravděpodobnostní hustoty intenzity obrazu (Kap. 5.3.1). Normalizací tohoto histogramu získáme pravděpodobnostní rozdělení jednotlivých pixelů v celém obrázku (Obr. 5.4). Máme-li vytvořené pravděpodobnostní modely aplikujeme tyto modely na vstupní obrázek. To jest, každému pixelu vstupního obrázku přiřadíme hodnotu pravděpodobnosti námi vytvořených modelů. Výsledkem tedy jsou dvě pravděpodobnostní matice jejich hodnoty odpovídají pravděpodobnosti modelu *předmět* a *pozadí* (Obr. 5.6). Z pravděpodobnostních matic získáme energetické matice (Obr. 5.7). Takto vytvořené matice předáme MATLAB Wrapperu, který má za úkol zprostředkovat komunikaci mezi prostředím MATLAB a knihovny C++. Nejdůležitější částí této segmentační metody je efektivní implementace algoritmu *max flow* (Kap. 5.2). Detailní popis algoritmu GraphCut nalezneme v Kap. 5.2.1. Výsledkem segmentace je binární oblast  $L$  odpovídající struktuře segmentované oblasti (Obr. 5.3).

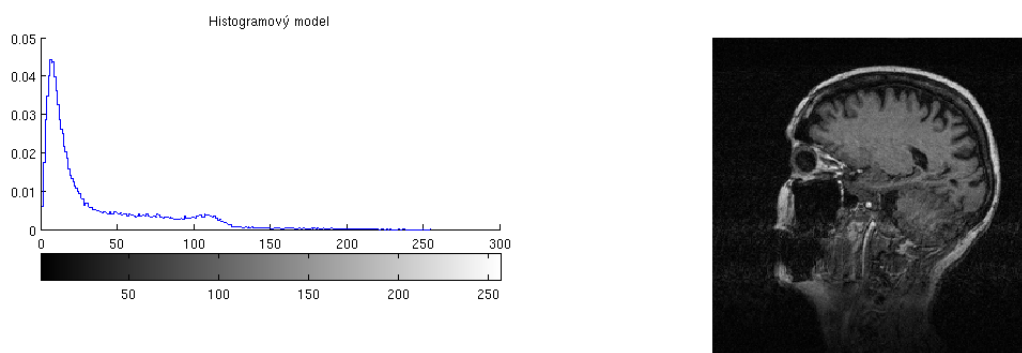


**Obrázek 5.3.:** Definice oblastí *předmět* (modrá) *pozadí* (červená), segmentace (zelená)

### 5.3.1. Tvorba pravděpodobnostních modelů

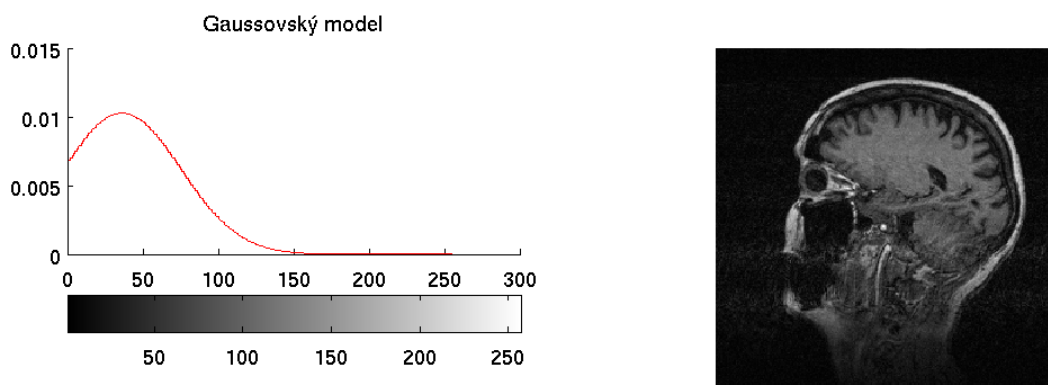
Histogram lze chápat jako odhad pravděpodobnosti hustoty intenzity v obraze. Histogram jasu  $h_f(z_i), i = 0, \dots, L - 1$  je vektor  $h$  s počtem složek rovným počtu jasových úrovní  $L$ . Z označených oblastí vypočítáme histogramy a normalizací vytvoříme pravděpodobnostní model pro jednotlivé oblasti *předmět* a *pozadí*.

**Histogramový model** Ukázka histogramového modelu, popisuje pravděpodobnost výskytu pixelů obrázku 5.4.



**Obrázek 5.4.:** Histogramový odhad výskytu pravděpodobnosti hustoty intenzity v obrázku

**Gaussovský model** Slouží k aproximaci pravděpodobnosti hustoty intenzity v obrázku, bodovým odhadem  $\mu$  a  $\sigma$  můžeme aproximovat parametry. Metoda nám umožňuje získat odhad histogramové modelu. Její výhodou je možnost vyhladit lokální extrémy. Na obrázku 5.5 můžeme vidět Gaussovským odhad výskytu pravděpodobnosti hustoty intenzity v obrázku, který získáme z rovnice 5.5.



**Obrázek 5.5.:** Gaussovským odhad výskytu pravděpodobnosti hustoty intenzity v obrázku

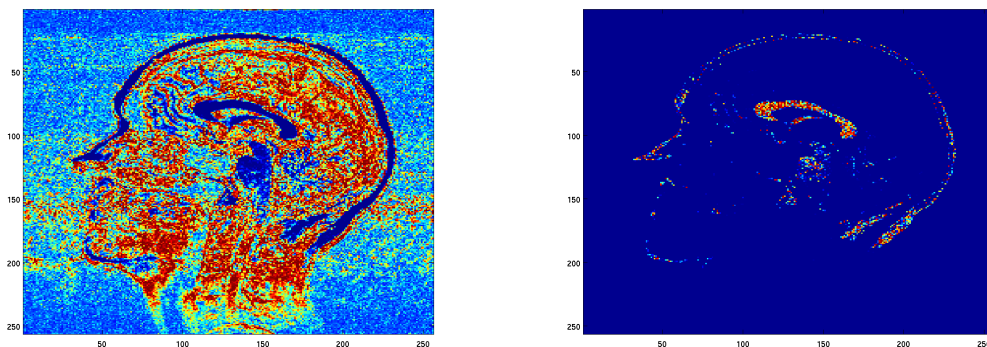


$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \text{ kde} \quad (5.5)$$

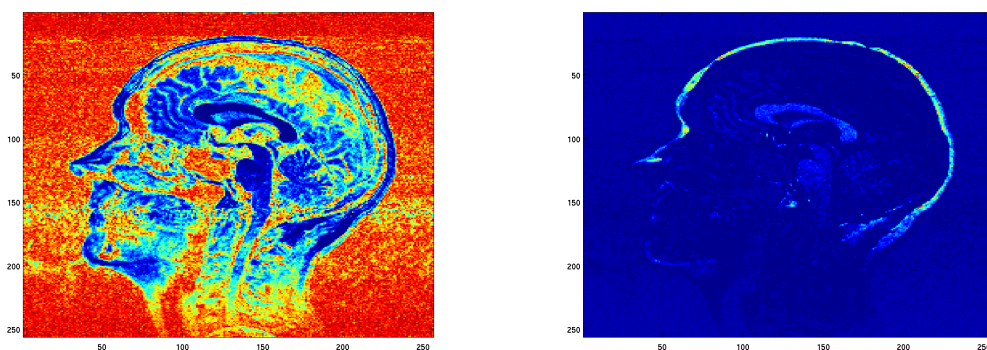
$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2}, \quad \mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (5.6)$$

### 5.3.2. Výpočet energie modelu

Z pravděpodobnostních matic modelů je potřebné získat energetické matice. Aplikováním pravděpodobnostních modelů dostane matice pravděpodobností ze kterých následně vypočítáme energie dle vztahu 5.8. Výběr oblastí *předmětu* a *pozadí* odpovídá obrázku 5.3 na kterém můžeme také vidět výsledek segmentace pro tento konkrétní případ.



**Obrázek 5.6.:** Pravděpodobnostní matice - *předmět* (vlevo), *pozadí* (vpravo)



**Obrázek 5.7.:** Energetické matice - *předmět* (vlevo), *pozadí* (vpravo) 5.8

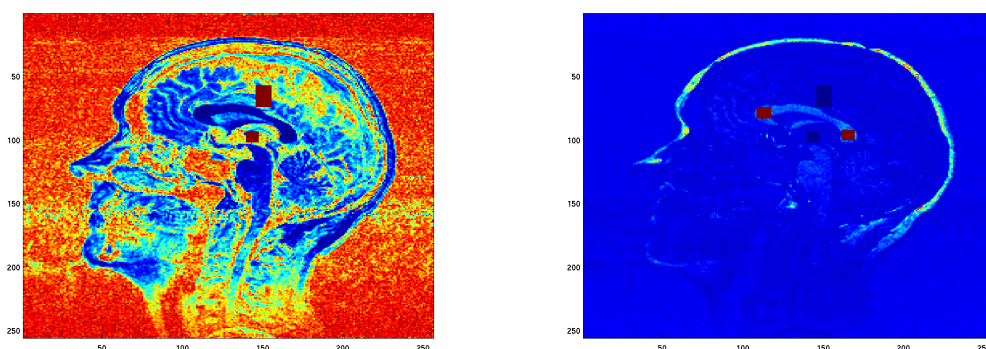
$$R_p(\text{"pre"}) = -\ln\Pr(I_p|O) \quad (5.7)$$

$$R_p(\text{"poz"}) = -\ln\Pr(I_p|B) \quad (5.8)$$

Tyto energetické matice odpovídají cenám vah koncových uzlů t-hran (rov. 5.2). Energie vytvořená na základě modelu *předmět* odpovídá cenám hran uzlů grafu  $G_{st}$  spojených z koncovým  $s$  uzlem a energie vytvořená na základě *pozadí* modelu odpovídá cenám hran uzlů grafu  $G_{st}$  spojených s koncovým  $t$  uzlem.

### 5.3.3. Nastavení pevných konstant

Stejně oblasti, které jsem použil pro tvorbu pravděpodobnostních matic, jsem použil také pro definování *pevných konstant*. Jedná se o speciální konstanty, které dodefinují hodnotu koncových hran v matici  $R_p$  (rov. 5.2) uzlům grafu  $G_{st}$ , které jsme označili jako *předmět* a *pozadí* (Obr. 5.3). Koncové hrany nabývají hodnot 0 nebo  $K$  dle tabulky 5.1.



Obrázek 5.8.: Energetické matice - *předmět* (vlevo), *pozadí* (vpravo)

## 5.4. 3D segmentace

Segmentace v 3D prostoru je analogická se segmentací ve 2D prostoru (Kap. 5.3). Vytvoříme pravděpodobnostní modely (Kap.5.3.1) z námi označených oblastí *předmětu* a *pozadí* (Obr. 5.9). Na základě těchto modelů přepočítáme celý objem dat pro *předmět* i *pozadí*. To jest, každému pixelu segmentovaného objemu přiřadíme hodnotu pravděpodobnosti námi vytvořených modelů. Výsledkem tedy jsou dvě pravděpodobnostní matice jejichž hodnoty odpovídají pravděpodobnosti modelu *předmět* a *pozadí* a to jako pro *předmět* tak pro *pozadí*. To znamená, že dostaneme dvě trojrozměrné matice dimenze odpovídající segmentovanému objemu. Z těchto matic vypočítáme energetické matice (Kap. 5.3.2). Do těchto matic dodefinujeme *pevné konstanty* (Kap. 5.3.3) dle tabulky 5.2.1. Výsledkem jsou tedy dvě trojrozměrné



**Obrázek 5.9.:** Definice oblastí *předmět* (modrá) *pozadí* (červená), výběr (zelená)

matice, které odpovídají dle vztahu 5.8 energiím získaných z pravděpodobnostních matic. Segmentace ve 3D prostoru je časově mnohem náročnější než segmentace ve 2D prostoru a z tohoto důvodu jsem implementoval nástroj, kterým je možné zvolit část objemu nad kterým se bude provádět segmentace a tím časově urychlit dobu segmentace.

## 6. Implementace prohlížečky

### 6.1. Metoda implementace grafického prostředí

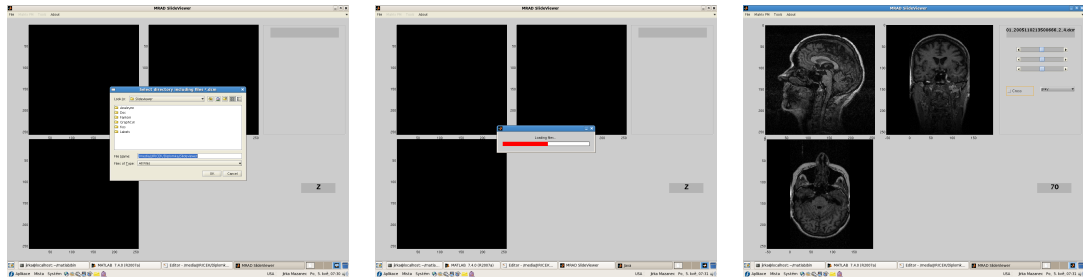
Tato programovací technika jazyku *MATLAB* využívá příkazy `switch` a `case`. Základem je využití vlastností funkcí volat samu sebe s různými parametry. Mimo této vlastnosti je dobré znát také vlastnosti systému Handle Graphics a mít běžné základy programů. Na první pohled velmi jednoduchá a intuitivní technika se mi v praxi předvedla jako efektivní a spolehlivá metoda, kterou je možné použít pro tvorbu i složitého GUI. Na tomto principu pracuje převážná část mého grafického prostředí.

```
1 function Program1(vstpar)
2     if nargin==0
3         Monitor=get(0, 'ScreenSize');
4         F = figure('Units','Pixels','Name','Hlavni obrazek',...
5             'Position',[100 .4*Monitor(4) .75*Monitor(3) .4*Monitor(4)],...
6             'Tag','Figure','Color',[1 1 1],'NumberTitle','off',...
7             'Resize','on','Menubar','none');
8         h1 = uicontrol('Units','Normalized','Style','Push',...
9             'Position',[.6 .5 .25 .18],'Tag','Tlacitko','String','Stiskni',...
10            'Callback','Program1 stisk','ForegroundColor','White',...
11            'BackgroundColor',[.7 .2 0],'FontWeight','Bold','FontSize',12);
12        h2 = uicontrol('Units','Normalized','Style','Edit',...
13            'Position',[.1 .5 .3 .18],'Tag','editace','String','nezmacknul',...
14            'ForegroundColor','White','BackgroundColor',[.7 .20],'FontWeight','
15            Bold','FontSize',12);
16    else
17        switch(vstpar)
18            case('stisk')
19                set(findobj('Tag','editace'),'String','Stiskni me');
20                set(findobj('Tag','editace'),'FontWeight','Bold');
21                set(findobj('Tag','editace'),'FontSize',14);
22        end
23    end
```

### 6.2. Načítací modul

Modul slouží k načítání souborů ve formátu *DICOM*. Snímky, které jsem měl k dispozici, byly uloženy v DICOM souborech takto: 1 snímek v souboru, 16 bitová barevná hloubka, rozlišení 256 x 256 pixelů. v každé sadě bylo 140 souborů. Data jsou rozdělena do souborů

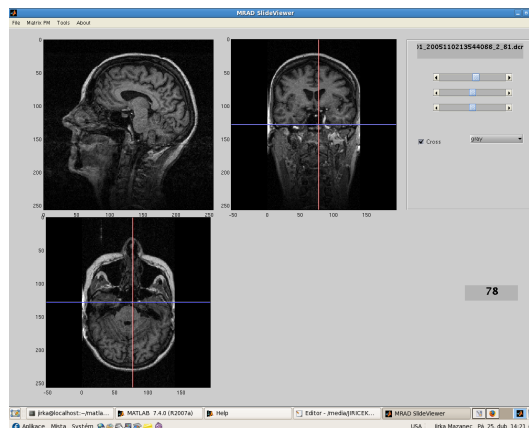
reprezentující milimetrové sagitální řezy hlavy člověka. Tyto soubory jsem načel pomocí prostředí *MATLAB* a tímto jsem vytvořil vstupní matici dat. Pomocí funkce *MATLABu dicomread* jsem načel vstupní soubory a podle toho jsem nastavil parametry prohlížeče (velikost obrázku, hodnoty jasu, atd.). Na obrázku 6.1 je zobrazen postup při nahrávání souborů.



Obrázek 6.1.: Výběr adresáře (vlevo), Nahrávání dat (uprostřed), Stav po nahrání (vpravo)

### 6.3. Prohlížení 3D řezů

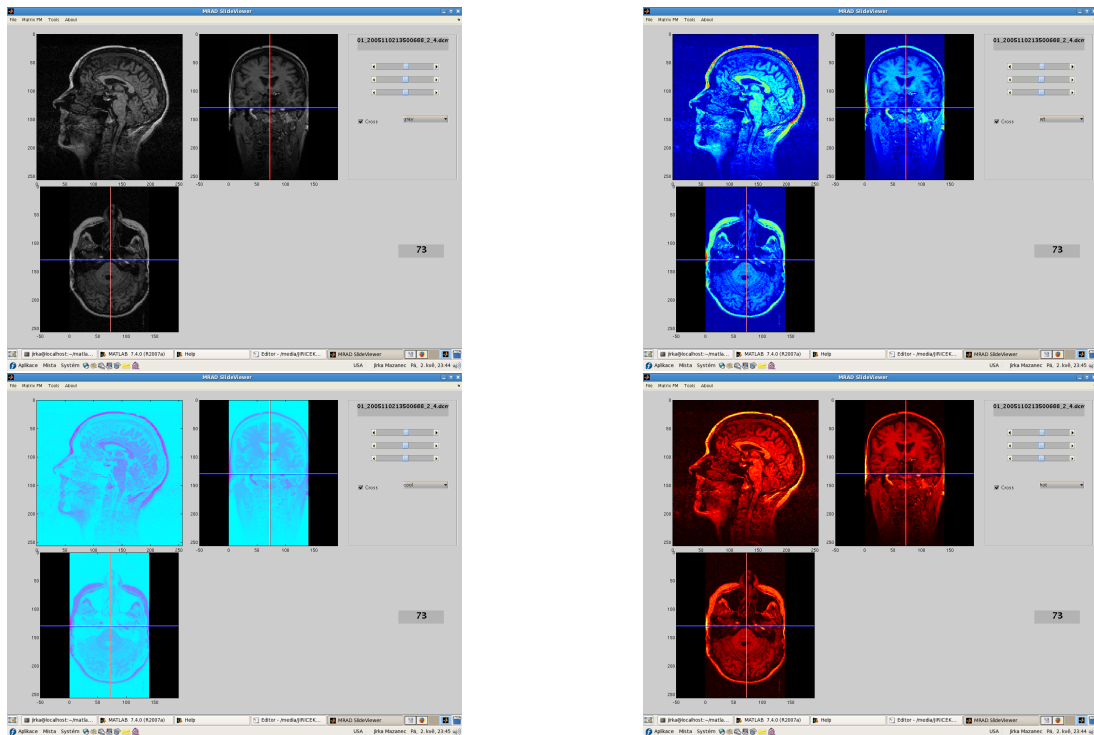
Prohlížečka umožňuje práci s obrazem ve třech na sobě kolmých řezech, po načtení vstupní matice se prohlížečka přepne do pozice ve které zobrazí vždy řez, který se nachází uprostřed dané roviny. Ke změně mezi jednotlivými řezy slouží posuvníky (slidery), jejichž posunem lze měnit aktuální pozici a tím procházet vstupní matici dat v kterémkoliv řezu. K označení aktuální pozice řezu nám slouží poziční kříž, který je možno aktivovat zaškrtnutím políčkem (checkbox 6.2).



Obrázek 6.2.: Prohlížení řezů - sagitální (vlevo), frontální (vpravo), transversální (dole)

## 6.4. Grafické modifikace obrazů

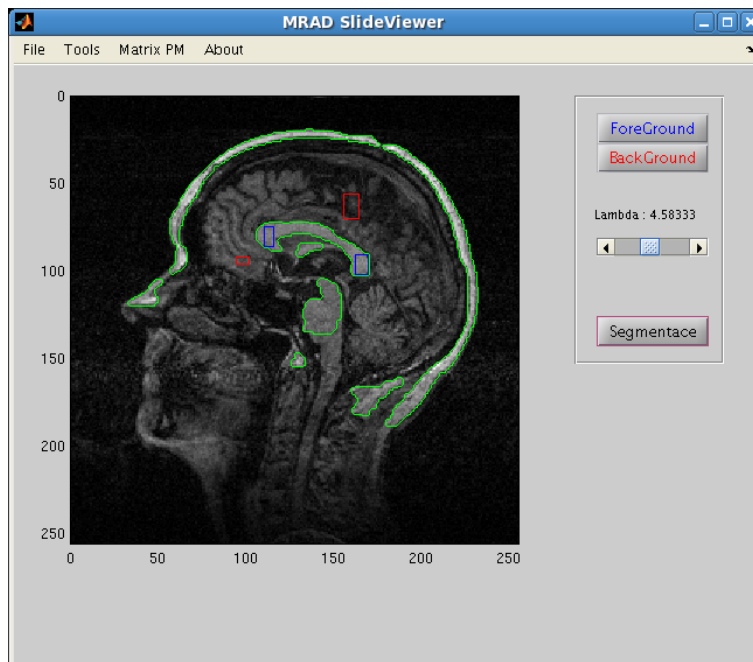
Pro zvýraznění kontur jsem použil funkci pro zobrazení obrazu v různých barevných mapách. Jednotlivým úrovním intenzit obrazu se přiřazují různé barvy dle definovaných barevných škál. V mé prohlížečce jsem použil čtyři základní druhy barevných map. Na obrázcích jsou zobrazeny různé škály stejného řezu. Pro srovnání můžeme porovnat



Obrázek 6.3.: Reprezentace dat v různých škálách

## 6.5. 2D segmentační modul

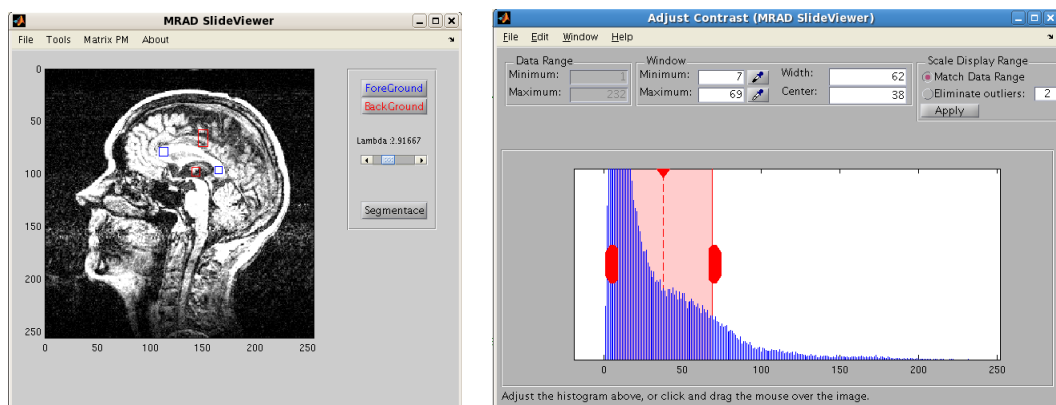
V segmentačním okně je možné vybrat jeden ze tří zobrazených řezů (sagitální, frontální, transversální), použít funkci pro nahrání externího souboru. Pomocí funkce *MATLAB imread* je možné nahrát obrázky v různých formátech (např. Giff, Bmp, Jpeg, Png atd.). Pomocí tlačítek s názvy Foreground a Background je možné označit oblasti, které považujeme za *předmět* a *pozadí*. Parametr  $\lambda$  je parametr definující váhu  $R(L)$  (rov. 5.1). V tomto okně je také možné měnit kontrast obrázku, což nám může pomoci pro získání lepšího výsledku segmentace (Kap. 6.6). Výsledný labeling je možné uložit na disk v podobě souborů *\*.mat* a následně tento binární obraz zase nahrát ke konkrétnímu obrázku.



Obrázek 6.4.: Výsledek 2D Segmentace (modrá oblast *předmět*, červená oblast *pozadí*)

## 6.6. Kontrastní modul

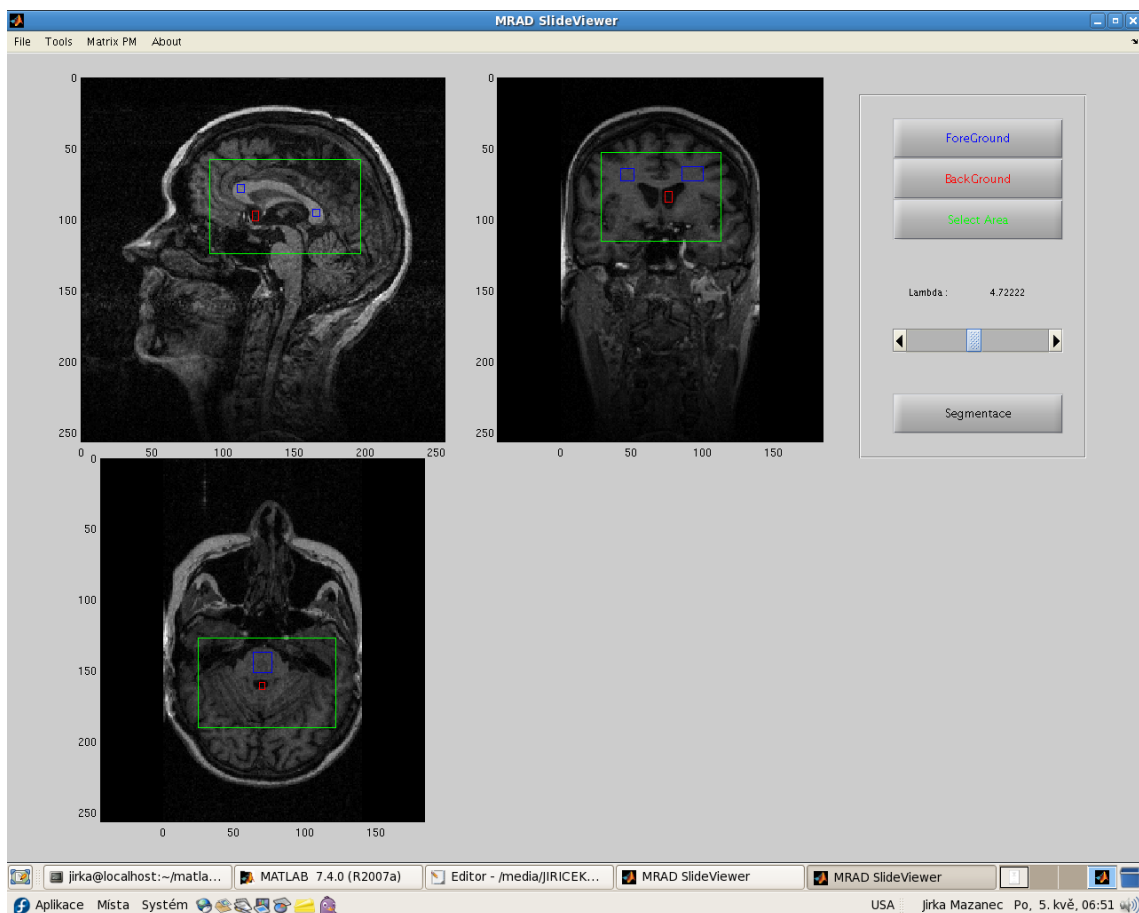
Kontrastní modul slouží k grafickým modifikacím obrázku úpravou histogramu obrázku. Takto upravený obrázek můžeme manuálně zbavit šumu nebo pro nás nejdůležitých částí spektra histogramu. Tímto způsobem můžeme dosáhnout ještě kvalitnějších výsledků segmentace. Kontrastní modul je součástí prostředí MATLAB.



Obrázek 6.5.: Změna kontrastu obrázku

## 6.7. 3D Segmentační modul

Segmentace obrazu se skládá z vytvoření pravděpodobnostních modelů z námi označených oblastí *předmět* a *pozadí* (kap. 6.5). Jde znovu o vytvoření pravděpodobnostních modelů, které je možno díky více oknům definovat ve 3D prostoru, což vede k lepším výsledkům při segmentaci. Navíc jsou zde zelená pole, která slouží k výseku objemu, nad kterým se bude provádět segmentace.



Obrázek 6.6.: 3D Segmentační modul

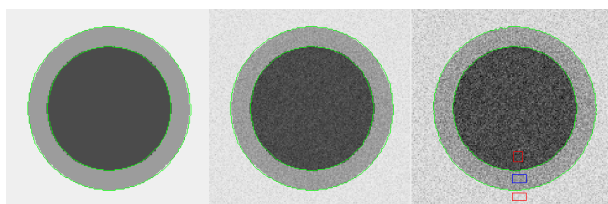


## 7. Výsledky segmentace

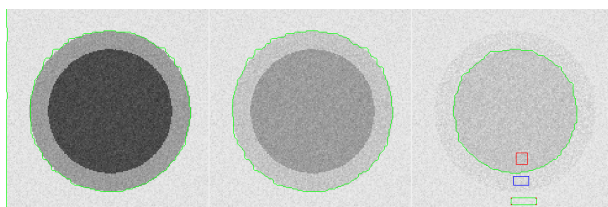
V této kapitole se budu zabývat testováním segmentační metody *Graph Cut* a výsledky segmentací, které jsem testoval na datech (fantomu), získaných ze zdroje [26]. Hlavní motivací je přesvědčit se, zdali segmentační metoda pracuje správně a porovnat syntetické data zašuměné různými druhy šumu s referenční segmentací. Úspěšnost segmentační metody jsem definoval jako poměr počtu správně nalezených pixelů s referenčním počtem pixelů vyjádřeným v procentech.

### 7.1. Testování na fantomech

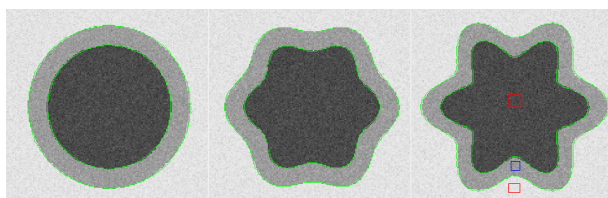
Na následujících obrázcích jsou zobrazeny výsledky segmentací obrázků, kdy jsou vybrány oblasti pro vytvoření modelů - *předmět* (modře) *pozadí* (červeně).



**Obrázek 7.1.:** Na těchto obrázcích je zobrazen signál/šum různých intenzit (SNR)

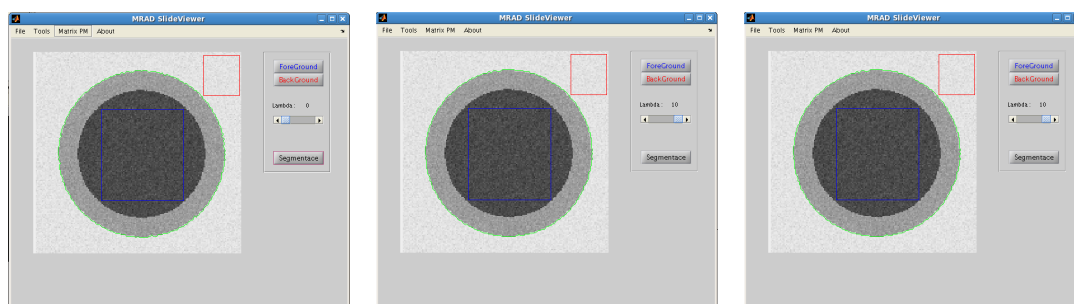


**Obrázek 7.2.:** Na těchto obrázcích je zobrazen kontrast/šum různých intenzit (CNR)



**Obrázek 7.3.:** Zobrazení segmentace proměnné tvarové komplexity

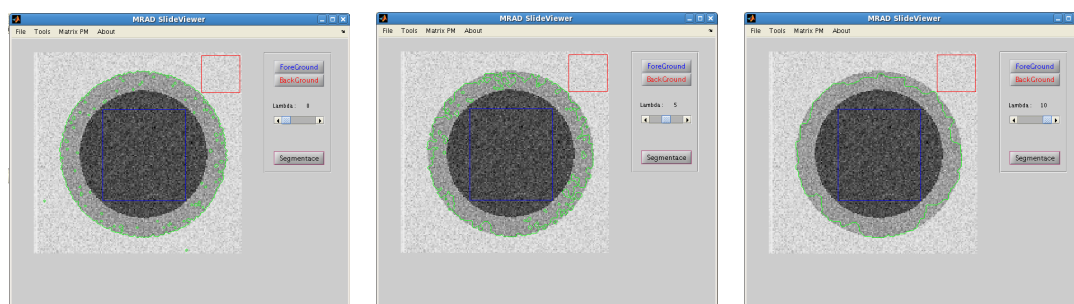
## 7. Výsledky segmentace



**Obrázek 7.4.:** Výsledky segmentace obrázků zašuměných SNR pro různé parametry  $\lambda$  (5.1), číselné porovnání tabulka 7.1

$\lambda$	čas [s]	úspěšnost [%]
0	0,0934	94,26
5	0,1029	98,69
10	0,1503	100,73

**Tabulka 7.1.:** Výsledky segmentace SNR

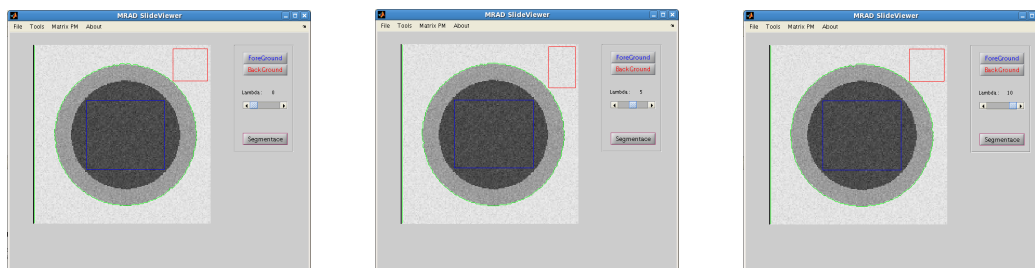


**Obrázek 7.5.:** Výsledky segmentace obrázků zašuměných SNR pro různé parametry  $\lambda$  (5.1), číselné porovnání tabulka 7.2

$\lambda$	čas [s]	úspěšnost [%]
0	0,0865	79,89
5	0,0805	80,50
10	0,0900	83,47

**Tabulka 7.2.:** Výsledky segmentace SNR

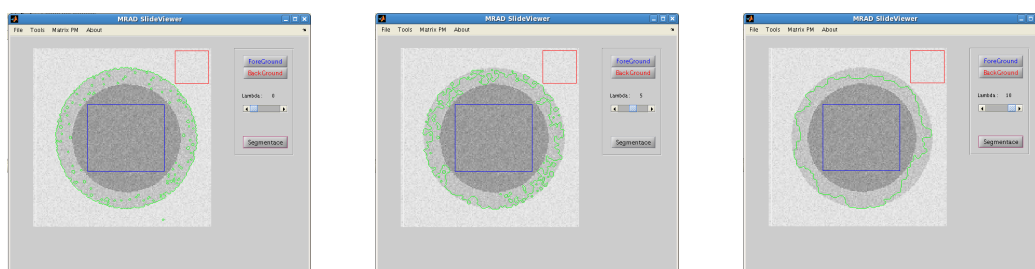
## 7. Výsledky segmentace



**Obrázek 7.6.:** Výsledky segmentace obrázků zašuměných CNR pro různé parametry  $\lambda$  (5.1), číselné porovnání tabulka 7.3

$\lambda$	čas [s]	úspěšnost [%]
0	0,0780	99,02
5	0,0693	98,04
10	0,0819	99,57

**Tabulka 7.3.:** Výsledky segmentace CNR



**Obrázek 7.7.:** Výsledky segmentace obrázků zašuměných CNR pro různé parametry  $\lambda$  (5.1), číselné porovnání tabulka 7.4

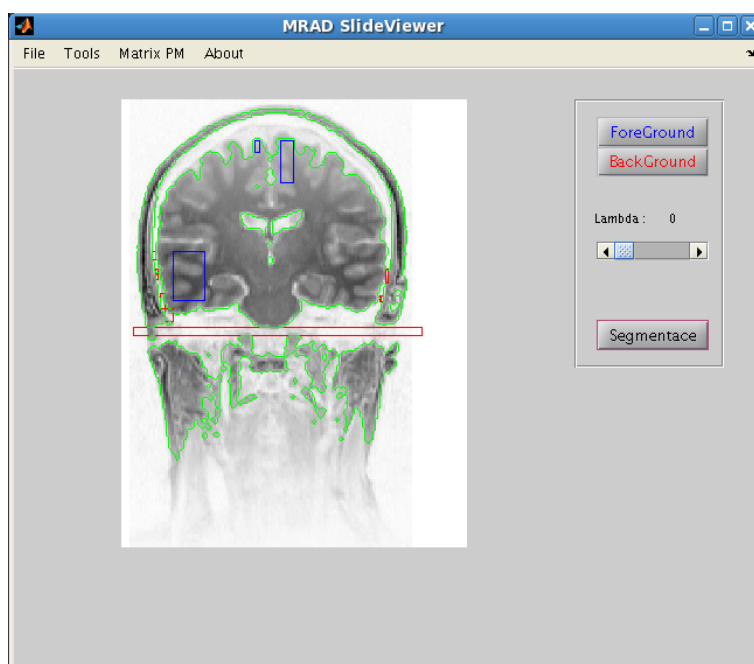
$\lambda$	čas [s]	úspěšnost [%]
0	0.0679	90,84
5	0,0833	86,00
10	0,1022	80,96

**Tabulka 7.4.:** Výsledky segmentace CNR

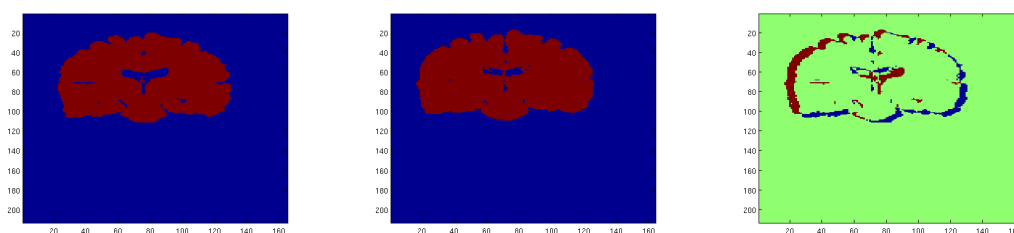
Na těchto fantomech jsem testoval schopnost algoritmu segmentovat mezikruží (prostřední kruh), ve kterém jsem neoznačil ani *předmět* ani *pozadí*. Výsledky by samozřejmě dopadly mnohem lépe, kdy jsme označili přesněji oblast našeho zájmu.

## 7.2. Výsledky 2D segmentace

Při 2D segmentaci jsem jako referenci použil data z [26], ve kterých jsem zvolil frontální řez. K těmto datům byly přiřazeny i jednotlivé výsledky polo-automatické segmentace programu *CardView* používaného jako retenční segmentační nástroj. Výsledek segmentace je zobrazen na obrázku (7.9 vlevo).



Obrázek 7.8.: Segmentační modul



Obrázek 7.9.: Výsledky segmentací CardView (vlevo), Graph Cut (uprostřed), Subtrakce obou masek (vpravo)

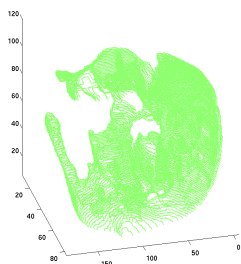
Porovnání vychází ze subtrakce masek, to je masky referenční s maskou získanou ze segmentace metodou *GraphCut*. Poměr výsledného počtu pixelů subtrakční masky s počtem pixelů masky referenční definuje výslednou odchylku. Tato odchylka je úměrná kvalitě výsledné segmentace.

čas [s]	celkem [px]	pixelů špatně [px]	úspěšnost [%]
0,0934	34932	1754	99.04
0,1029	34932	1854	99.05
0,1503	34932	1590	99.06

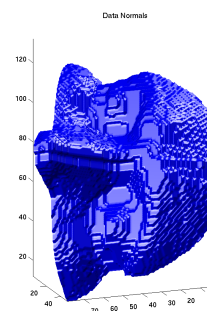
**Tabulka 7.5.:** Výsledky 2D segmentace pro 3 tři různé obrázky

### 7.3. Výsledky 3D segmentace

Výsledky 3D segmentace metodou *Graph Cut* nejsou až tak uspokojivé, důvodem je zřejmě špatná kvalita snímku (artefakty), které se ve snímcích objevují. Z hlediska 2D segmentace jsou zanedbatelné, ale při 3D segmentaci už hrají důležitou roli. Segmentace je v prostoru časově náročnější a než se segmentace 2D, což jsem vyřešil omezením objemu nad kterým se segmentace prováděla.



**Obrázek 7.10.:** Výsledek 3D segmentace - binární oblast



**Obrázek 7.11.:** Výsledek 3D segmentace - základní renderování

## 8. Závěr

Zobrazovací metody již v dnešní době dosahují opravdu kvalitních výsledků především zásluhou magnetické rezonance, kterou můžeme považovat za jednu z nejsofistikovanějších zobrazovacích metod. V klinické praxi se s tímto pokrokem vyskytly další možnosti využití takto získaných dat. Jednou z nich je i segmentace jednotlivých struktur v lidském těle a jejich oddělení od svého okolí.

Přesné vymezení struktur mozku zůstává stále procesem, který není jednoduché zcela zautomatizovat. Důvodem jsou rozdílné tvary struktur orgánů, které jsou u každého člověka jedinečné, proto nebývá jednoduché najít segmentační metodu, která by byla schopna automaticky vyhledat požadovanou strukturu v dané oblasti. Existuje mnoho segmentačních metod, které jsem ve své práci popsal, ale najít opravdu universální metodu, která je schopna segmentace různých druhů lidských orgánů je v dnešní době zatím problematické. Chceme-li struktury v lidském těle přesněji lokalizovat vyžadujeme nějaký vstup od uživatele, který tak má možnost výsledky segmentace obrazu ovlivňovat. Při svém studiu jsem se setkal z mnoha různými segmentačními technikami, ale nejvíce mě oslovila právě interaktivní metoda *Graph Cut* a proto jsem se rozhodl tuto metodu implementovat.

Hlavním cílem mé práce bylo vytvořit program umožňující načítání, prohlížení a segmentaci lékařských dat. Vzhledem k charakteru vstupních dat nebylo možné implementovat princip rigidní transformace. Proto byla práce obohacena o rozšíření segmentačního algoritmu *Graph Cut* do 3D prostoru. V prostředí MATLAB jsem implementoval program ve třech modulech - modul Prohlížeče, modul 2D-segmentace, modul 3D-segmentace. Prohlížečský modul je schopný pomocí prostředí MATLAB načíst vstupní soubory ve formátu *DICOM* a zobrazit tyto data ve třech na sobě kolmých řezech. Dále je schopný zobrazení vstupních dat v různých barevných škálách a umožňuje zapnutí či vypnutí pozičního kříže.

Segmentační algoritmus jsem testoval na fantomech, které byly zatíženy různými druhy šumu. Testování na fantomech dosáhlo uspokojivých výsledků jak na fantomech testující tvarovou komplexitu, tak i na fantomech, které byly zatíženy různými druhy šumu. Výsledky segmentace jsem porovnával s poloautomatickým segmentačním programem *CardView* používaným komunitou zabývající se segmentací mozkových struktur. Výsledky 3D segmentace metodou *Graph Cut* nejsou až tak uspokojivé, důvodem je zřejmě špatná kvalita snímku (artefakty). Tyto výsledky ukazují nejen ověření funkčnosti této metody, ale především to, že segmentační metoda *Graph Cut* se jeví jako velice perspektivní metoda pro segmentaci obrazu v oblasti biomedicíny a to nejen ve 2D, ale i v 3D prostoru.

Grafové algoritmy sloužící k segmentaci obrazu se jeví jako perspektivní metoda segmentace struktur lidského mozku.

Na přiloženém CD lze nalézt data pacientů použítá k segmentaci, implementovanou aplikaci a elektronickou verzi diplomové práce.

# Literatura

- [1] ZUNA I., PAPOUŠEK L.: *Úvod do zobrazovacích metod v lékařství*, ČVUT Praha, 2002, (str. 35-37), ISBN-80-01-02152-1
- [2] HLAVÁČ V., SEDLÁČEK M.: *Zpracování signálů a obrazů*, ČVUT Praha, 2005, (str. 152), ISBN-80-01-03110-1
- [3] *Informace o formátu DICOM:*  
<ftp://medical.nema.org/medical/dicom>
- [4] KOLMOGOROV V., BLAKE A.: *Interactive Foreground Extraction using Iterated Graph Cut*, Microsoft Research Cambridge, UK, 2002
- [5] BOYKOV Y., JOLLY M.: *Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images*, Siemens corporate research, Princeton, USA, 2001
- [6] *Download Max flow algoritmu:*  
<http://www.adastral.ucl.ac.uk/~vladkolm/software.html>
- [7] *Download Matlab Wrapperu:*  
<http://www.wisdom.weizmann.ac.il/~bagon/matlab.html>
- [8] *Přehled Segmentačních metod používaných v medicíně:*  
<http://www.fit.vutbr.cz/~spanel/segmentace>
- [9] COLEMAN G. B., ANDREWS H. C.: *Image Segmentation by Clustering*, Proc. IEEE, 1979, 773-785
- [10] HAŠKOVEC V., MUDROVÁ M.: *Detekce hran v biomedicínských obrazech*, VŠCHT Praha, Ústav počítačové a řídicí techniky, [www.phobos.vscht.cz](http://www.phobos.vscht.cz)
- [11] HARALICK R. M., KELLY G. L.: *Pattern Recognition with Measurement Space and Spatial Clustering for Multiple Images*, Proc. IEEE, (str. 654-665), 1969
- [12] STEGMANN M. B.: *Active Appearance Models: Theory, Extensions and Cases. Master Thesis, 2nd edition*, Technical University of Denmark, IMM, 2000
- [13] STEGMANN M. B.: *Active Appearance Models*, Technical University of Denmark, 2004 - [http://math.berkeley.edu/~sethian/level\\_set.html](http://math.berkeley.edu/~sethian/level_set.html)
- [14] COOTES T. F., EDWARDS G. J., TAYLOR C.J.: *Active Appearance Models. In Proceedings of European Conference on Computer Vision 1998*, Springer, 1998
- [15] COOTES T. F.: *Active Appearance Models. Division of Imaging Science and Biomedical Engineering*, University of Manchester, UK, č. 2, (str. 484-498), 2006

- [16] HARIS K., EFSTRATIADIS N., MAGLAVERAS N., KATSAGGELOS A.K.: *Hybrid Image Segmentation Using Watersheds and Fast Region Merging*, IEEE Transactions on Image Processing, 1998
- [17] GRAU V. ET AL.: *Improved Watershed Transform for Medical Image Segmentation Using Prior Information*, IEEE Transactions on Medical Imaging, č. 23, (str. 4), 2004
- [18] COOTES T. F.: *Image Segmentation and Mathematical Morphology*, Center of Mathematical Morphology home page, University of Manchester, UK, č. 2, str. 484-498 2000 - <http://cmm.ensmp.fr/~beucher/wtshed.html>
- [19] NOVÁK M. A KOL.: *Umělé neuronové sítě, teorie aplikace*, Praha, C. H. Beck, 1998.
- [20] WU Y., LIU Q., HUANG T. S.: *An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization*, University of Illinois at Urbana-Champaign, 2000
- [21] WU Y.: *Segmentation: Clustering, Graph Cut and EM. ECE510*, Computer Vision Notes Series 5, 2001
- [22] COSTA L. F.: *Neural-based Color Image Segmentation and Classification using Self-Organizing Maps*, IX SIBGRAPI, (str. 47-54), 1996
- [23] MA F. ET AL.: *Probabilistic Segmentation of Volume Data for Visualization Using SOM-PNN Classifier*, IEEE, 1998
- [24] WISMÜLLER A., VIETZE F. ET AL.: *A Neural Network Approach to Adaptive Pattern Analysis - the Deformable Feature Map. European Symposium on Artificial Neural Networks, IESANN' - Belgium, Bruges, (str. 189-194), 2000*
- [25] BANKMAN I. N.: *Handbook of Medical Imaging: Processing and Analysis. Academic Press, San Diego, CA, USA, 2000*
- [26] *Zdroj dat z magnetické rezonance mozku a výsledky jejich segmentací:*  
<http://www.cma.mgh.harvard.edu/ibsr/>
- [27] SVOBODA T., KYBIC J. *Image Processing, Analysis, and Machine Vision: A MATLAB Companion*, Thomson Learning, 2007, (str. 291-306), ISBN - 0495295957
- [28] KOLMOGOROV V., ZABIH R.: *What Energy Functions can be Minimized via Graph Cuts?*, To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Earlier version appeared in European Conference on Computer Vision (ECCV), 2002



- [29] BOYKOV Y., KOLMOGOROV V.: *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision*, To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2004

# Seznam obrázků

3.1. Snímek magnetické rezonance mozku - Sagitální řez . . . . .	12
3.2. Snímek magnetické rezonance mozku - Frontální řez . . . . .	12
4.1. Původní obrázek magnetické rezonance . . . . .	14
4.2. Detekce hran - Cannyho hranový detektor . . . . .	14
4.3. Detekce vrcholu histogramu. . . . .	17
4.4. Segmentace záprstních kostí pomocí AAM . . . . .	19
4.5. SOM neuronové sítě . . . . .	21
4.6. Originál (Kohonenovy sítě) . . . . .	22
4.7. Segmentace Kohonenovými sítěmi . . . . .	22
4.8. Struktura neuronové sítě s radiální bázovou funkcí . . . . .	22
4.9. Automatická segmentace pomocí GRBF neuronové sítě . . . . .	24
4.10. Watershed segmentace lidského mozku . . . . .	25
4.11. Výsledek watershed transformace . . . . .	25
5.1. Graph Cut - Příklad 2D segmentace obrázku . . . . .	28
5.2. Graph Cut - popis segmentace . . . . .	29
5.3. Definice oblastí - 2D segmentace . . . . .	31
5.4. Histogramový odhad . . . . .	32
5.5. Gaussovský odhad . . . . .	32
5.6. Pravděpodobnostní matice . . . . .	33
5.7. Energie modelu . . . . .	33
5.8. Nastavení pevných konstant . . . . .	34
5.9. 3D Segmentace . . . . .	35
6.1. Načítací modul . . . . .	37
6.2. Prohlížení řezů . . . . .	37
6.3. Reprezentace dat v různých škálách . . . . .	38

6.4. Výsledek 2D Segmentace . . . . .	39
6.5. Změna kontrastu obrázku . . . . .	39
6.6. 3D Segmentační modul . . . . .	40
7.1. Zobrazení poměru signál/šum různých intenzit (SNR) . . . . .	41
7.2. Zobrazení poměru kontrast/šum různých intenzit (CNR) . . . . .	41
7.3. Zobrazení segmentace proměnné tvarové komplexity . . . . .	41
7.4. Segmentace obrázků zašuměných SNR . . . . .	42
7.5. Segmentace obrázků zašuměných SNR . . . . .	42
7.6. Segmentace obrázků zašuměných CNR . . . . .	43
7.7. Segmentace obrázků zašuměných CNR . . . . .	43
7.8. Segmentační modul . . . . .	44
7.9. Porovnání výsledků Graph Cut se segmentací CardView . . . . .	44
7.10. Výsledek 3D segmentace - binární oblast . . . . .	45
7.11. Výsledek 3D segmentace - základní renderování . . . . .	45
A.1. Prohlížení řezů . . . . .	54
A.2. 2D Segmentační modul . . . . .	55
A.3. 3D Segmentační modul . . . . .	56

# Seznam tabulek

2.1. Citlivost prvků v Magnetickém poli . . . . .	10
4.1. Cannyho hranový detektor . . . . .	16
4.2. Shluková analýza . . . . .	18
4.3. Metoda segmentace - AAM . . . . .	19
4.4. Kohonenovy mapy . . . . .	22
4.5. Neuronové sítě - GRBF . . . . .	24
4.6. Watershed transformace . . . . .	26
5.1. Graph Cut - ohodnocení hran . . . . .	30
7.1. Výsledky segmentace SNR . . . . .	42
7.2. Výsledky segmentace SNR . . . . .	42
7.3. Výsledky segmentace CNR . . . . .	43
7.4. Výsledky segmentace CNR . . . . .	43
7.5. Výsledky 2D segmentace pro 3 tři různé obrázky . . . . .	45

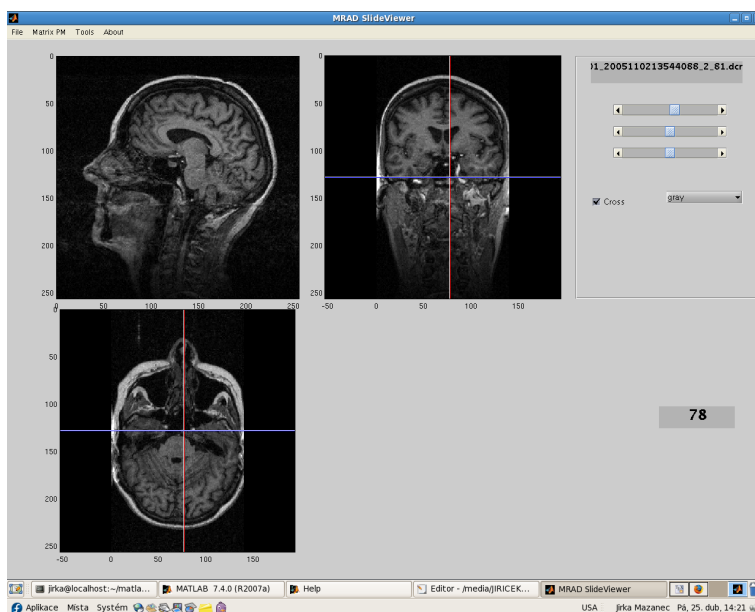
# Seznam zkratek

AAM	Active Appearance Models
CNR	Contrast/Noise Ratio
DICOM	Digital Imaging and Communications in Medicine
GRBF	Generalized Basis Functions
GUI	Graphics User Interface
MRI	Magnetická rezonance
NEMA	National Electrical Manufacturers Association
PCA	Principal Component Analysis
SNR	Signal/Noise Ratio
SOM	Self-organizing Maps

# A. Ovládání programu

## A.1. SlideViewer - 3D prohlížečka

Celý postup je detailně popsán v kapitole 6.5. V této kapitole jsou popsány uživatelské funkce pro základní orientaci a ovládání programu. V programu je možné spustit celkem 3 okna a to SlideViewer (prohlížeč), segmentační okno pro 2D segmentaci a segmentační okno pro 3D segmentaci.

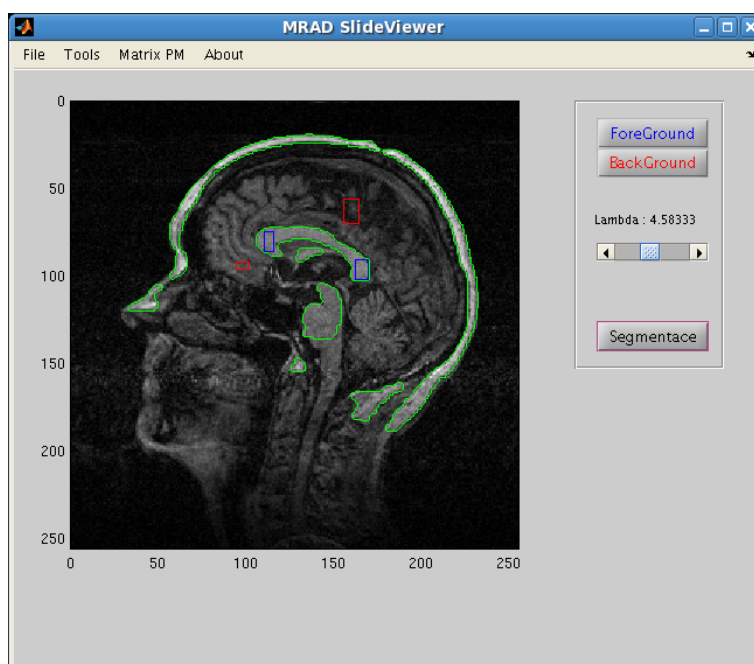


**Obrázek A.1.:** Prohlížení řezů - sagitální (vlevo), frontální (vpravo), transversální (dole)

1. V liště *File/Load Directory* lze nahrát adresář z obsaženými soubory DICOM
2. V liště *File/Matrix PM* lze ze souboru nahrát výslednou segmentaci
3. V liště *Tools* lze zvolit segmentační nástroje pro 2D a 3D segmentaci
4. V liště nástroje se zobrazí informace o programu
5. Posuvníky slouží pro prohlížení řezů načtených dat
6. Zaškrtnuté políčko slouží k aktivaci pozičního kříže

## A.2. Segmentace - 2D modul

Umožňuje zobrazit 3 základní řezy z prohlížeče SlideViewer (sagitální, frontální, transversální) nebo načíst externí obrázek ve formátu *DICOM*, *Bmp*, *Jpeg* (kap. 6.5). Umožňuje spustit kontrastní nástroj pro základní modifikace obrazu a po konečné segmentaci uložit nebo nahrát výslednou binární oblast jako soubor *\*.mat*.

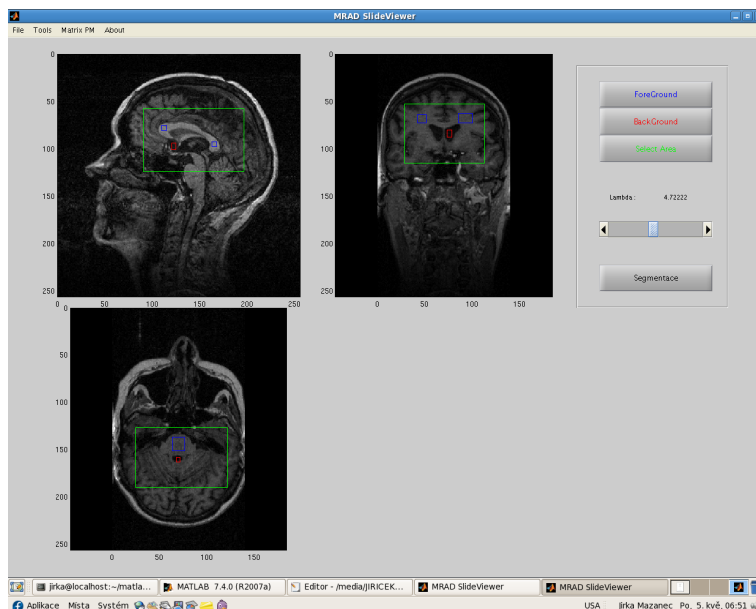


Obrázek A.2.: 2D Segmentační modul

1. V liště *File/Load Image* lze nahrát externí soubor a nebo lze zvolit aktuální řez z 3D prohlížeče (sagitální, frontální, transversální)
2. Zvolením *předmět* nebo *pozadí* lze vybrat oblast která bude považována za *předmět* nebo *pozadí* pro následnou segmentaci (pro vybrání nové oblasti je nutné znovu kliknout na příslušné tlačítko)
3. Nastavením parametru lambda je definována konstanta udávající velikost smoothness termu (5.1) - jemnost segmentace
4. V liště *Tools/Contrast* lze vybrat kontrastní nástroj pro úpravu obrázku před následnou segmentací
5. Po vybrání všech oblastí je možné zmáčknout segmentaci a tím spustit segmentační algoritmus
6. Výslednou binární oblast je možno uložit na disk v podobě *\*.mat* souboru

### A.3. Segmentace - 3D modul

Umožňuje zobrazit 3 základní řezy z prohlížeče SlideViewer (sagitální, frontální, transversální) a po konečné segmentaci uložit nebo nahrát výslednou binární oblast jako soubor \*.mat.



Obrázek A.3.: 3D Segmentační modul

1. Zvolením *předmět* nebo *pozadí* lze vybrat oblast která bude považována za *předmět* nebo *pozadí* pro následnou segmentaci (pro vybrání nové oblasti je nutné znovu kliknout na příslušné tlačítko)
2. Zvolením *Select area* se definuje výsek objemu nad kterým se bude provádět segmentace
3. Nastavením parametru lambda je definována konstanta udávající velikost smoothness termu (5.1) - jemnost segmentace
4. Po vybrání všech oblastí je možné zmáčknout segmentaci a tím spustit segmentační algoritmus
5. Výslednou binární oblast je možno uložit na disk v podobě \*.mat souboru a je možné ho nahrát v základním okně prohlížeče



## B. Seznam funkcí

Funkce programu lze rozdělit do tří základních skupin na funkce *Matlabu*, funkce *Wrapperu* a funkce *C++* (po kompilaci *mex*). Ke každé funkci je uveden krátký popis pro základní orientaci v programu.

### B.1. Funkce Matlab

- *About.m* - zobrazí informace o programu
- *Energie\_Slider.m* - vrací hodnotu posuvníku segmentačních modulů
- *GC.m* - definuje vstupní parametry pro 2D segmentaci
- *GC3D.m* - definuje vstupní parametry pro 3D segmentaci
- *Load\_Image.m* - načítají modul 2D segmentace
- *Matice.m* - načítají modul prohlížeče Slideviewer
- *Retc\_For.m* - definuje oblast *předmět* pro 2D segmentaci
- *Retc\_For\_3D.m* - definuje oblast *předmět* pro 3D segmentaci
- *Retc\_Back.m* - definuje oblast *pozadí* pro 2D segmentaci
- *Retc\_Back\_3D.m* - definuje oblast *pozadí* pro 3D segmentaci
- *Retc\_Select.m* - definuje objem nad kterým se provede 3D segmentace
- *Segmentace2D.m* - hlavní okno 2D segmentace
- *Segmentace3D.m* - hlavní okno 3D segmentace
- *Setmap.m* - nastavení barevné škály prohlížeče Slideviewer
- *Show\_A1.m* - zobrazení řezu prohlížeče Slideviewer
- *Show\_A1\_Cross.m* - zobrazení pozičního kříže řezu prohlížeče Slideviewer
- *Show\_A2.m* - zobrazení řezu prohlížeče Slideviewer
- *Show\_A2\_Cross.m* - zobrazení pozičního kříže řezu prohlížeče Slideviewer
- *Show\_A3.m* - zobrazení řezu prohlížeče Slideviewer
- *Show\_A3\_Cross.m* - zobrazení pozičního kříže řezu prohlížeče Slideviewer
- *Show\_Image\_Frontal.m* - zobrazení frontálního řezu prohlížeče

- *Show\_Image\_Sagital.m* - zobrazení sagitálního řezu prohlížeče
- *Show\_Image\_Transversal.m* - zobrazení transversálního řezu prohlížeče
- *SlideViewer.m* - spouští program Slideviewer
- *Start.m* - pomocná funkce pro 2D segmentaci
- *Start3D.m* - pomocná funkce pro 3D segmentaci

## B.2. Funkce Matlab - Wrapper

- *GraphCut.m* - komunikace mezi prostředím *Matlab* a knihovnamy *C++*
- *OpenGraphCut.m* - komunikace mezi prostředím *Matlab* a knihovnamy *C++*

## B.3. Funkce C++

- *graph.h*, *block.h*, *maxflow.cpp* - implementuje max-flow algoritmus
- *energie.h* - implementuje efektivnější konstrukci grafu
- *GCoptimization.h* - implementuje minimalizaci energie v grafu
- *GraphCut.h* - hlavní funkce