

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

SEGMENTACE HIPOKAMPU V MRI DATECH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

OLDŘICH KODYM

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

SEGMENTACE HIPOKAMPU V MRI DATECH

SEGMENTATION OF HIPPOCAMPUS IN MRI DATA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

OLDŘICH KODYM

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR WALEK

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor
Biomedicínská technika a bioinformatika

Student: Oldřich Kodym

ID: 155584

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Segmentace hipokampu v MRI datech

POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s teorií grafů a zejména pak s její aplikací v oblasti zpracování a analýzy obrazových dat. 2) Proveďte literární rešerši publikovaných algoritmů a metod pro segmentaci obrazových dat využívajících řez grafem. Zaměřte se zejména na segmentaci medicínských objemových dat. 3) Prakticky se seznamte s grafovou reprezentací obrazových dat a se základními operacemi z teorie grafů v programovém prostředí MATLAB. 4) Navrhněte strukturu softwaru pro segmentaci hipokampu s využitím znalostí z oblasti teorie grafů a prakticky jej realizujte. 5) Software doplňte vhodným grafickým uživatelským rozhraním, které obsluhu umožní (v případě potřeby) zasahovat do procesu segmentace. 6) Výslednou metodu otestujte na reálných datech jak zdravých, tak pacientů postižených Alzheimerovou chorobou. Výsledky diskutujte.

DOPORUČENÁ LITERATURA:

[1] LÉZORAY, O., GRADY, L. J. Image processing and analysis with graph: theory and practice. Boca Raton: CRC Press, c2012, 537 s. ISBN 978-1-4398-5507-2.

[2] FELZENSZWALB, P. F. a HUTTENLOCHER, D. P. Efficient Graph-Based Image Segmentation. International Journal of Computer Vision. 2004, roč. 59, č. 2, s. 167-181.

Termín zadání: 9.2.2015

Termín odevzdání: 29.5.2015

Vedoucí práce: Ing. Petr Walek

Konzultanti bakalářské práce:

prof. Ing. Ivo Provazník, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá využitím grafových metod pro segmentaci málo kontrastních obrazových dat, konkrétně pro segmentaci hipokampu ze snímků magnetické rezonance. Nejprve je uvedena základní problematika a terminologie teorie grafů. Následně je popsána metoda minimálního řezu grafem včetně algoritmů schopných tento minimální řez nalézt. Následuje popis její implementace pro segmentaci 2D a 3D obrazových dat. Metoda byla testována na zkušebních datech a poté implementována jako modul pro software 3D Slicer. Zde byla testována na snímcích hipokampu zdravých pacientů stejně jako na pacientů trpících Alzheimerovou chorobou. Nastíněny jsou nejčastější problémy vyskytující se při segmentaci a možné postupy jejich řešení.

KLÍČOVÁ SLOVA

Teorie grafů, Řez grafem, Min-cut/Max-flow teorém, Ford-Fulkersonův algoritmus, Segmentace hipokampu

ABSTRACT

The thesis deals with application of graph-based methods in segmentation of low contrast image data, specifically hippocampus segmentation from magnetic resonance data. Firstly, basics and terminology of graph theory is introduced. Next, minimum graph cut method is explained along with algorithms capable of finding this cut. After that comes the description of its implementation for 2D and 3D image data segmentation. Method was tested on sample data and then implemented as a 3D Slicer software module. Here the method was tested on the hippocampus data of healthy patients as well as patients suffering from Alzheimer's disease. Most common problems occurring during the segmentation were forshadowed as well as possible ways to solve them.

KEYWORDS

Graph theory, Graph cut, Min-cut/Max-flow theorem, Ford-Fulkerson algorithm, Hippocampus segmentation

KODYM, Oldřich *Segmentace hipokampu v MRI datech*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2015. 42 s. Vedoucí práce byl Ing. Petr Walek,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Segmentace hipokampu v MRI datech“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Petru Walkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	9
1 Segmentace hipokampu	10
1.1 Hipokampus	10
1.2 Význam algoritmizace segmentace hipokampu	10
2 Úvod do teorie grafů	12
2.1 Definice grafu	12
2.2 Vlastnosti grafů důležité pro hledání řezu	13
2.2.1 Váhováný graf	13
2.2.2 Orientovaný a směřovaný graf	13
2.3 Tok sítě	14
2.4 Cesta a řez grafem	15
3 Min-cut/max-flow algoritmy	17
3.1 Ford-Fulkersonův augmenting paths algoritmus	17
3.1.1 Popis algoritmu	18
3.1.2 Vliv způsobu vyhledávání cest na výpočetní náročnost	20
3.2 Boykov-Kolmogorovův algoritmus	21
4 Implementace algoritmu pro segmentaci obrazu	23
4.1 Vhodná konverze obrazových dat na graf	23
4.2 Segmentační energie	24
4.2.1 Region term	24
4.2.2 Boundary term	25
4.2.3 Celková energie	26
4.2.4 Pevná omezení	26
4.3 Publikované metody segmentace hipokampu využívající řez grafem . .	27
4.4 Testování na zkušebních 2D a 3D datech	28
5 Implementace v programovém prostředí MATLAB®	31
5.1 Popis metody zpracování vstupních dat	31
5.2 Uživatelské rozhraní programu	32
5.3 Implementace jako modul pro software 3D Slicer	34
6 Diskuze úspěšnosti algoritmu	36

7 Závěr	38
Literatura	39

SEZNAM OBRÁZKŮ

1.1	Znázornění polohy hipokampu a amygdaly v lidském mozku [28].	10
1.2	Hipokampus na laterálním snímku magnetické rezonance [29].	11
2.1	Příklad grafu pro znázornění vztahů mezi objekty.	12
2.2	Příklad váhovaného grafu.	13
2.3	Příklad orientovaného grafu a směřovaného grafu.	14
2.4	Příklad sítě.	15
2.5	Příklady dvou s-t řezů grafem.	16
3.1	Reziduální síť a cesty.	18
3.2	Iterace Ford-Fulkersonova algoritmu.	19
3.3	DFS a BFS metody vytváření vyhledávacích stromů pro hledání cest.	20
3.4	Cesta nalezená ve fázi růstu.	21
4.1	Model grafu reprezentujícího obrazová data pro segmentaci.	24
4.2	Test algoritmu pro 2D vstupní data.	29
4.3	Test algoritmu pro 3D vstupní data.	30
5.1	Uživatelské rozhraní navrženého softwaru (vlevo) a zobrazení pravděpodobnostních funkcí na základě histogramu označených voxelů (vpravo).	32
5.2	Chybná detekce šedé kůry mozkové.	33
5.3	Schéma procesu interaktivní segmentace.	33
5.4	Slicelet vytvořen pro grafovou segmentaci.	35
6.1	Původní snímek hipokampu zdravého pacienta (vlevo), lékařem označený hipokampus (uprostřed), výsledek výpočtu segmentace s označenými voxely (vpravo). Snímek byl podobným způsobem označen v 7 řezech před prvním výpočtem a poté v dalších 5 řezech před finálním výpočtem segmentace s koeficientem $\lambda = 0,03$	37
6.2	Původní snímek hipokampu pacienta s Alzheimerovou chorobou (vlevo), lékařem označený hipokampus (uprostřed), výsledek výpočtu segmentace s označenými voxely (vpravo). Snímek byl podobným způsobem označen v 8 řezech před prvním výpočtem a poté v dalších 8 řezech před finálním výpočtem segmentace s koeficientem $\lambda = 0,02$	37
6.3	3D vizualizace segmentovaného hipokampu.	37

ÚVOD

Tato práce se zabývá využitím grafových metod pro segmentaci málo kontrastních obrazových dat hipokampu. Vychází přitom především z [3] a [2], které využívají algoritmus nalezení minimálního řezu grafem pro optimální segmentování obrazu do objektu a pozadí. Minimalizují přitom segmentační energii, která vyjadřuje penalizaci za přidělení bodu do jedné nebo druhé množiny.

V první kapitole je uvedena motivace pro automatizaci segmentačních metod. V případě segmentace hipokampu ze snímků magnetické rezonance může usnadnit například včasnou diagnózu Alzheimerovy choroby. Ruční rozměření je totiž kvůli velmi nejasným hranicím velmi obtížné a časově náročné.

Druhá kapitola slouží k objasnění základních pojmů v teorii grafů, jejíž metody se stále častěji používají mimo jiné pro různé úkony obrazové analýzy. Jsou vysvětleny pojmy vrchol, hrana, váha, orientace, tok a řez, přičemž je snaha dodržet terminologii tak, jak je popsána v [24], jelikož různé zdroje základní pojmy často definují s drobnými odlišnostmi.

Ve třetí kapitole je výčet nejpoužívanějších algoritmů schopných najít v grafu minimální řez využitím min-cut/max-flow teorému. Podrobněji je popsán Ford-Fulkersonův algoritmus založený na metodě *augmenting paths*. Dále je uveden základní princip Boykov-Kolmogorova algoritmu představeném v [2], který z Ford-Fulkersonova algoritmu vychází, ale upravuje způsob, jakým v grafu vyhledává cesty, a tím dosahuje kratších výpočetních časů.

Čtvrtá kapitola ukazuje postup převedení obrazových dat do grafu tak, aby v něm mělo smysl hledat minimální řez. Jsou definovány *region term* a *boundary term* složky segmentační energie. První odráží nakolik si daný bod vlastnostmi odpovídá s množinou objektu nebo pozadí, druhá potom odlišnost od sousedního pixelu na rozhraní objekt/pozadí. Celkově tak vyjadřují penalizaci za vedení hranice daným místem. Nakonec jsou uvedeny výsledky použití metody na testovací data, které poukazují na vhodnost metody pro využití při náročných segmentacích, jakými je vymezení hipokampu ve snímcích magnetické rezonance.

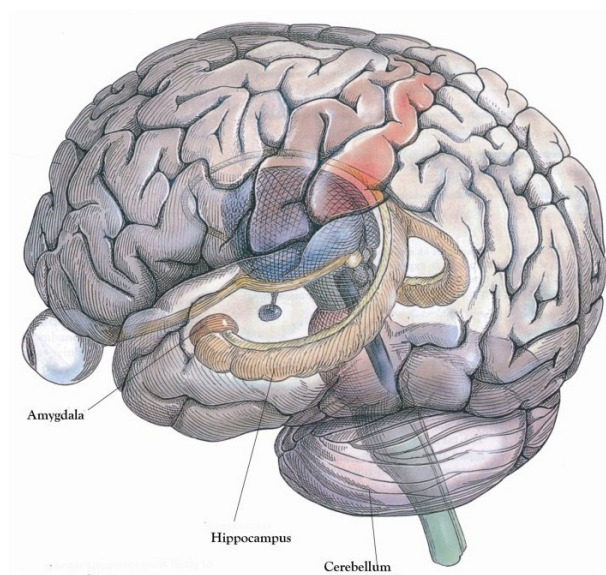
V páté kapitole je popsán způsob, jakým byl algoritmus implementován v programovém prostředí MATLAB®. Poté jsou uvedeny důvody, proč je pro tvorbu grafického prostředí programu pro segmentaci medicínských dat vhodnější použít místo MATLABu software 3D Slicer a podrobně popsán postup práce v navrženém grafickém prostředí.

Šestá kapitola obsahuje diskuzi úspěšnosti segmentačního programu v porovnání se zlatým standartem u zdravých pacientů a pacientů trpících Alzheimerovou chorobou. Uvedeny jsou také problémy, se kterými se lze při segmentaci setkat a způsoby, jakým jim předcházet nebo je minimalizovat.

1 SEGMENTACE HIPOKAMPU

1.1 Hipokampus

Hipokampus (lat. *hippocampus*) je párový orgán limbického systému umístěný ve střední části spánkového laloku. Důležitou roli hraje především ve zpracování informací a uchovávání krátkodobé paměti. Segmentace hipokampu, tedy rozdělení obrazu na oblast obsahující hipokampus a oblast obsahující okolní tkáň, má význam především pro diagnostiku onemocnění, která způsobují ztrátu objemu hipokampu. Klíčovou roli hraje především při diagnostice Alzheimerovi choroby, kde lze změnu objemu sledovat již ve velmi časných stádiích onemocnění a diagnózu lze provést presymptomaticky [12].



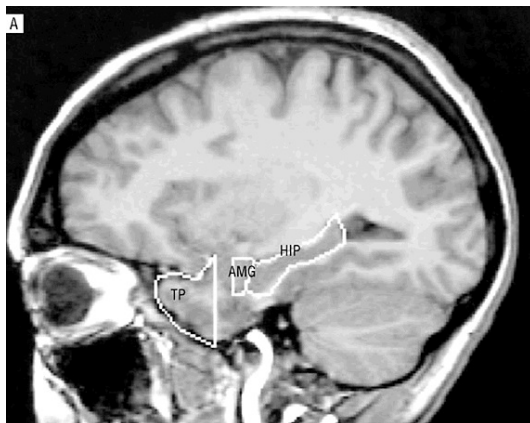
Obr. 1.1: Znázornění polohy hipokampu a amygdaly v lidském mozku [28].

Mezi onemocnění spojená s atrofií hipokampu dále patří například deprese, schizofrenie, rozvinutý alkoholismus či posttraumatický stres. Kromě diagnózy těchto onemocnění se objem hipokampu stanovuje taky v souvislosti s evolučními stromy či sledováním chování zvířat a další souvislosti jsou předmětem výzkumu [1] [17] [20] [5].

1.2 Význam algoritmizace segmentace hipokampu

Tkáň hipokampu je strukturně velmi podobná okolním mozkovým tkáním a proto je velmi těžké ji vymezit i na snímcích z magnetické rezonance, které dosahují u měkkých tkání nejvyššího kontrastu. Odborníci nejsou plně jednotní ani při ručním

definování, které automatické a poloautomatické metody považují za zlatý standard, především kvůli nejasné hranici mezi tkání hipokampu a sousední amygdalou. Tyto problémy způsobují, že určení objemu hipokampu je pro lékaře velice časově náročné a proto je zde snaha nalézt algoritmy schopné segmentace automaticky či s minimálním zásahem lékaře [22].



Obr. 1.2: Hipokampus na laterálním snímku magnetické rezonance [29].

Vzhledem k výše zmíněným úskalím segmentace hipokampu se však algoritmy běžně používané pro segmentaci často kvůli své jednoduchosti nehodí, jelikož se často řídí pouze jedním parametrem. Pro dostatečně přesný výsledek je v případě hipokampu však třeba kombinovat požadavky na dodržení kombinace podmínek jako je určitá hodnota jasů a tvar obrysu. Mezi algoritmy vhodnými pro tyto účely v poslední době roste popularita těch, které využívají k formulaci problému hledání řezu grafem.

2 ÚVOD DO TEORIE GRAFŮ

Grafové metody jsou stále častěji využívány pro řešení mnoha situací v mnoha různých oblastech jako je biochemie, telekomunikační sítě či algoritmizace, kde reprezentace dat grafem umožňuje k řešení problému využít širokou škálu vyvinutých optimalizačních metod. Vhodné jsou obecně například pro aplikace, ve kterých jde o nalezení minima funkce, kde vynikají tím, že nalezené minimum je vždy globální.

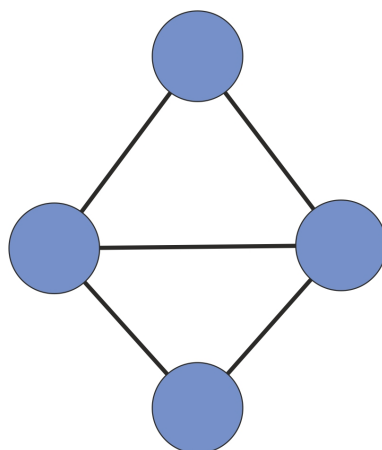
V posledních letech, kdy kvůli stále větším objemům obrazových dat rostou vysokým tempem i výpočetní nároky v oblasti zpracování obrazu, byly grafové metody využity i v této oblasti pro úlohy jako indexování obrazových databází, obrazové restaurace, rekonstrukce stereoskopického obrazu, texturní syntéza a segmentace [24] [3].

V této kapitole budou uvedeny a vysvětleny základní pojmy a operace teorie grafů.

2.1 Definice grafu

Graf můžeme chápat jako strukturu tvořenou dvěma prvky; množinou vrcholů či uzlů (*vertices, nodes*) a množinou hran (*edges*), které tyto vrcholy propojují. Takovou situaci můžeme představit jako krajinu s městy reprezentovanými vrcholy, které jsou propojeny cestami reprezentovanými hranami. Matematicky je graf G s vrcholy V a hranami E definován jako uspořádaná dvojice množin $G = (V, E)$, kde $E \subseteq V \times V$, tedy prvky množiny E jsou dvojprvkové podmnožiny množiny V [24] [8].

Nejčastěji používané značení v literatuře je $v_i \in V$ pro i -tý vrchol a $e_{ij} \in E$ pro hranu spojující i -tý a j -tý vrchol. V této práci bude použito značení stejné.



Obr. 2.1: Příklad grafu pro znázornění vztahů mezi objekty.

2.2 Vlastnosti grafů důležité pro hledání řezu

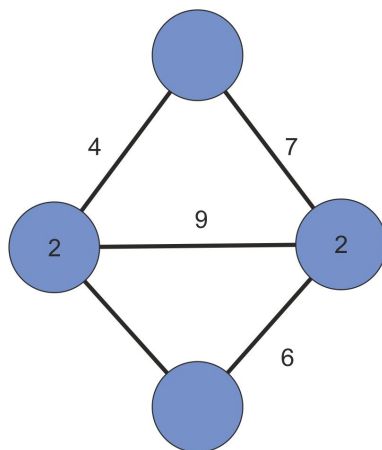
I přes tento na první pohled jednoduchý koncept existuje množství parametrů, kterými se různé grafy vytvořené pro různé účely liší. V této kapitole budou uvedeny vlastnosti grafu důležité pro proces nalezení řezu grafem, který vyjadřuje nejméně penalizovanou množinu hran a tedy pravděpodobnou hranici objektu v obraze, kterému graf odpovídá. Podrobněji bude tento pojem vysvětlen v kapitole 2.4.

2.2.1 Váhováný graf

Ve váhovaném grafu mají vrcholy a hrany přiřazenu reálnou váhu, která vyjadřuje jejich hodnotu vůči ostatním prvkům. V naší analogii s městy a cestami by tedy váhy grafu zřejmě představovaly velikost měst a šířku či počet pruhů cesty.

Zavádíme tedy váhovací funkci $\hat{w} : V \rightarrow \mathbb{R}$ pro vrcholy a analogicky $w : E \rightarrow \mathbb{R}$ pro hrany, která každému prvku množiny přiřazuje reálnou hodnotu. Konkrétní váhu i -tého vrcholu potom budeme značit w_i a hrany spojující i -tý a j -tý vrchol w_{ij} , ačkoliv v literatuře se můžeme setkat i s označením $w_{(e_{ij})}$ nebo $w_{(v_i, v_j)}$.

I neváhovaný graf ovšem můžeme chápat jako graf váhovaný tak, že všechny jeho váhy jsou rovny jedné. Protože do algoritmu hledání řezu vstupují pouze váhy hran, budou v našem případě váhy všech vrcholů $w_i = 1, \forall v_i \in V$. Váhy hran potom mohou nabývat libovolných hodnot. Pokud má hrana spojující i -tý a j -tý vrchol váhu $w_{ij} = 0$, považujeme tuto hranu za neexistující [7] [24].



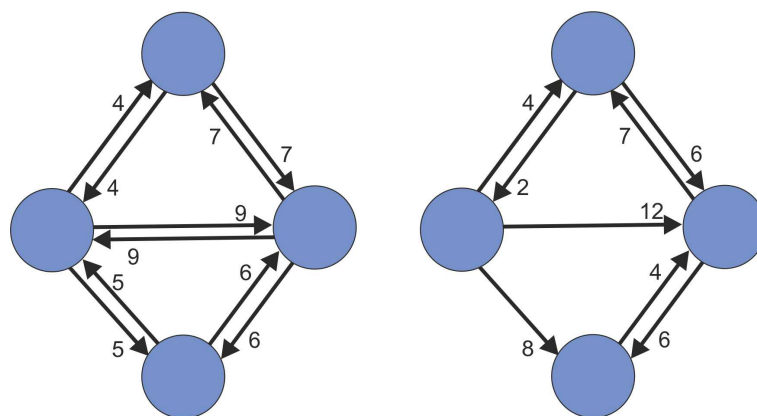
Obr. 2.2: Příklad váhovaného grafu.

2.2.2 Orientovaný a směrovaný graf

V orientovaném grafu přiřazujeme hranám kromě váhy také směr, tedy ke každé dvojici vrcholů (v_i, v_j) je kromě hrany e_{ij} definována také hrana e_{ji} . Jinými slovy

rozlišujeme hrany směřující od vrcholu v_i a hrany směřující k vrcholu v_i . Vrátime-li se k analogii s cestami mezi městy, můžeme takto na cestu přidat navíc dopravní pruh pouze pro jeden směr. Obdobně můžeme také vytvořit „jednosměrku“, nastavíme-li váhu hrany v opačném směru na nulu.

Směřovaný graf je potom takový orientovaný graf, u jehož hran platí, že $w_{ij} \neq w_{ji}$. I orientovaný graf tedy může být nesměrovaný v případě, že pro jeho hrany v opačných směrech platí, že mají stejné váhy. Takto je terminologie definována v [24] a bude dodržena i v této práci, ačkoliv se definice v různých literaturách mohou lišit [24]. Příklad orientovaného grafu je v obrázku 2.3 vlevo, směrovaný graf potom vpravo.



Obr. 2.3: Příklad orientovaného grafu a směrovaného grafu.

2.3 Tok sítě

Sítěmi rozumíme v teorii grafů speciálně konstruované grafy pro práci s tokem (*flow*). Váhy hran v síti, které jsou vždy orientované a váhované, se nazývají kapacity a značí se pro tento účel c_{ij} . Platí že $c_{ij} = w_{ij}$. Tyto kapacity vyjadřují maximální možnou hodnotu toku, který danou hranou může v daném směru procházet a jsou tedy vždy kladné. Kromě těchto vah je každé hraně existující v této síti přiřazen také tok f , pro který platí, že:

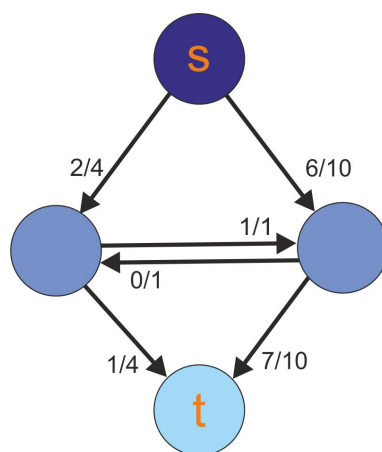
$$0 \leq f_{ij} \leq c_{ij}, \forall w(e_{ij}) \in E. \quad (2.1)$$

Platí-li pro hranu, že:

$$f_{ij} = c_{ij}, \quad (2.2)$$

mluvíme o této hraně jako o saturované.

Vrcholy v síti jsou specifické tím, že pro ně platí, že součet toků všech hran, které směřují do uzlu je roven součtu toků všech hran, které vedou z uzlu ven. Jde tedy o jistou obdobu prvního Kirchhoffova zákona. V síti však existují minimálně dva vrcholy, které tuto podmínku nesplňují. Jedná se o zdroj (*source*) značený s , u něhož je vždy tok směřující od něj větší než tok směřující k němu a stok (*sink*) značený t , pro který platí že naopak tok směřující od něj je menší než do něj. Obecně může být těchto speciálních terminálních vrcholů v síti víc, v této práci však bude v síti vždy právě jeden zdroj a jeden stok [24].



Obr. 2.4: Příklad sítě.

Na obrázku 2.4 je síť, ve které teče z tmavě modrého zdroje s do světle modrého stoku t tok. Ve vahách je zapsán tok/kapacita příslušné hrany. Prostřední hrana směřující doprava je saturovaná.

2.4 Cesta a řez grafem

Cesta mezi dvěma vrcholy je v teorii grafů obecně definovaná následovně:

$$\pi(v_1, v_n) = (v_1, e_1, v_2, \dots, v_{n-1}, e_{n-1}, v_n).$$

Jde tedy o propojení dvou vrcholů, v našem případě jde o hledání cesty $\pi(s, t)$ ze zdroje do stoku. Každá hrana či vrchol je v cestě obsažen nejvýše jednou. Obecně mohou cesty obsahovat jakkoliv orientované hrany [7].

Řezem grafem (*graph cut*) $G = (V, E)$ je označována taková množina hran, která rozděluje graf V na dvě části tak, že jedna část obsahuje zdroj a druhá stok, tedy ze zdroje nevede do stoku žádná cesta, která by neobsahovala aspoň jednu hranu z množiny G . Označíme-li množinu vrcholů propojených po provedení řezu se zdrojem $S \in V$ a množinu vrcholů propojených se stokem $T = V - S$, potom hrany řezu jsou

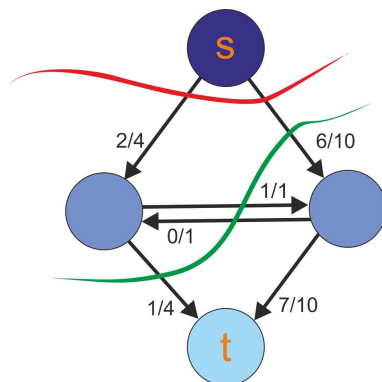
$\{e_{ij} \in E \mid v_i \in S, v_j \in T\}$. Důležitý je také fakt, že hrany v řezu jsou orientované od zdroje ke stoku, hovoříme tedy o s-t řezu (*s-t cut*).

Celková hodnota řezu $c(S, T)$ je rovna součtu kapacit všech hran, které jej tvoří, tedy

$$c(S, T) = \sum_{v_i \in S, v_j \in T} c_{ij}. \quad (2.3)$$

V obrazové analýze má smysl především hledání takového řezu, který má hodnotu nejmenší možnou (*minimum cut*, *min-cut*). Takový řez při vhodném sestavení sítě představuje právě hranice segmentovaného objektu. Je-li splněna podmínka, že kapacity hran jsou kladné, lze toto minimum relativně spolehlivě nalézt [24] [7].

Jak bylo definováno výše, pokud mezi zdrojem a stokem existuje hranami tvořená cesta, může přes ni téct tok. Pokud má však být zachována podmínka, že všechny vrcholy kromě zdroje a stoku mají mít součet příchozích toků shodný se součtem toků odchozích, může mít tento tok maximálně hodnotu nejmenší kapacity hrany v této cestě. Jak ukazuje důkaz například v [10], velikost maximálního toku, který může ze zdroje do stoku v celém grafu téct, je shodná s velikostí minimálního řezu (*min-cut/max-flow theorem*). Právě této duality často využívají algoritmy určené pro hledání minimálního řezu grafem [2].



Obr. 2.5: Příklady dvou s-t řezů grafem.

Na obrázku 2.5 má červený řez hodnotu 14, zelený řez má hodnotu 15. Ze dvou prostředních hran byla do zeleného řezu započítána pouze ta směřující doprava, opačná totiž směřuje z množiny T se stokem do množiny S se zdrojem.

3 MIN-CUT/MAX-FLOW ALGORITMY

Pro nalezení minimálního řezu ve směrovaném grafu s dvěma terminálními vrcholy (zdroj a stok) existují algoritmy, které jsou schopny spolehlivě najít jeho globální minimum (resp. globální maximum toku). Jednotlivé algoritmy se mohou lišit základní metodou i konstrukcí samotného grafu, většina jich však spadá do jedné ze dvou hlavních kategorií. První z nich jsou algoritmy na základě *push-relabel* metody [18] a druhou potom algoritmy na základě *augmenting paths* algoritmu [11]. Algoritmy první kategorie jsou obecně rychlejší, avšak konkrétně pro segmentaci obrazu se ukázal jako nejúspěšnější Boykov-Kolmogorovův algoritmus, který se řadí do druhé kategorie, ačkoliv využívá některých výhod *push-relabel* metody [2] [32].

3.1 Ford-Fulkersonův augmenting paths algoritmus

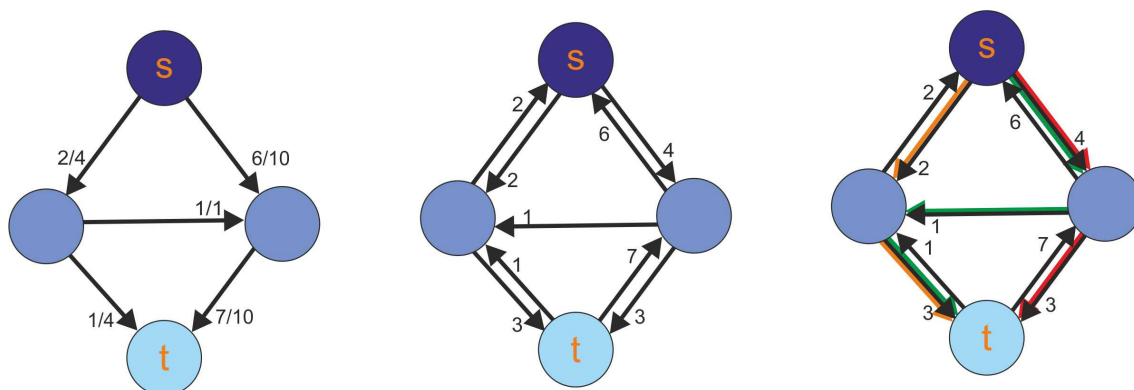
Vstupem do Ford-Fulkersonova algoritmu je síť s nulovým tokem, ve které není žádná antiparalelní hrana, tzn. mezi každou dvojicí vrcholů je hrana jen v jednom směru. Při samotných iteracích se ale pracuje s tzv. reziduální sítí (viz níže), ve které algoritmus hledá cestu ze zdroje do stoku a tuto cestu poté „posílí“ (*augmenting*) tím, že skrz ni pošle největší možný jednotný tok, který nepřekročí žádnou kapacitu v této cestě. Při hledání cesty v reziduální síti povede cesta pouze ve směru hran orientovaných od zdroje ke stoku.

Máme-li síť $G(V, E)$, potom její reziduální síť $G_f(V, E_f)$ obsahuje stejné vrcholy, hrany mají však jiné kapacity, tzv. reziduální kapacity. Tyto reziduální kapacity vyjadřují nevyužitou kapacitu hrany v původní síti. Pokud původní hranou protéká tok který ji nesaturuje, bude mít její reziduální kapacita hodnotu rozdílu toku a původní kapacity hrany. Zároveň se do reziduální sítě zavede hrana opačného směru, která bude mít hodnotu původního toku. Platí tedy, že:

$$c_f(i, j) = \begin{cases} c(i, j) - f(i, j) & \text{pokud } e_{i \rightarrow j} \in E, \\ f(j, i) & \text{pokud } e_{j \rightarrow i} \in E, \\ 0 & \text{ostatní.} \end{cases} \quad (3.1)$$

Poslední řádek znamená, že neexistuje-li hrana v původní síti, potom neexistuje ani v síti reziduální.

Na obrázku 3.1 je znázorněna reziduální síť (uprostřed) vytvořená podle sítě s tokem (vlevo). Reziduální hrany mají pouze kapacitu. Vpravo jsou v reziduální síti vyznačeny všechny tři možné cesty ze zdroje do stoku.



Obr. 3.1: Reziduální síť a cesty.

3.1.1 Popis algoritmu

Celý algoritmus lze pro graf $G(V, E)$ shrnout následujícím pseudo kódem [6].

```

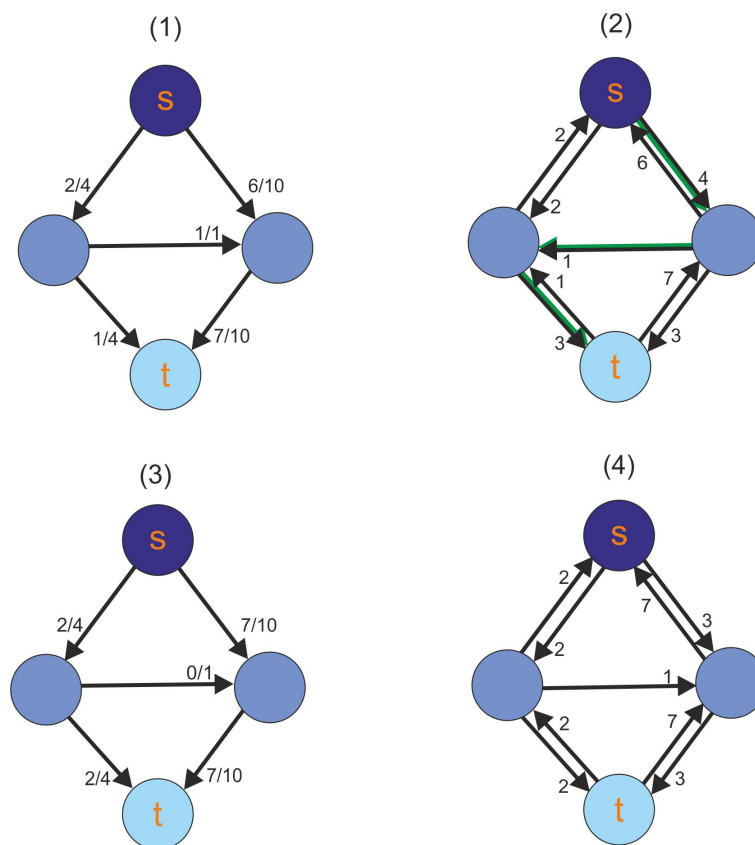
FORD-FULKERSON( $G, s, t$ );
for each edge  $(u, v)$  in  $E(G)$  do
     $f[u, v] = 0$ ;
     $f[v, u] = 0$ 
end
while there is a path  $p$  from  $s$  to  $t$  in the residual network  $G_f$  do
     $m = \min\{c(u, v) - f[u, v] : (u, v) \text{ is on } p\}$ ;
    for each edge  $(u, v)$  on  $p$  do
         $f[u, v] = f[u, v] + m$ ;
         $f[v, u] = -f[u, v]$ 
    end
end

```

V prvním kroku algoritmus každé hraně ve vstupní síti přiřadí nulový tok v obou směrech. V tomto kroku je tedy reziduální síť shodná se sítí původní. Následně se algoritmus pokusí v síti najít cestu ze zdroje do stoku (možnosti hledání cest budou podrobně popsány níže), kterou může poslat tok (*augmenting path*). Najde-li cestu, určí se největší možný tok jako nejmenší kapacita, kterou cesta obsahuje (*bottleneck capacity*). Tento tok se přičte všem jejím hranám orientovaným ve směru cesty a všem hranám orientovaným proti směru se odečte.

V další iteraci se vytvoří nová reziduální síť, která již neobsahuje v předchozím kroku saturovanou hranu (jelikož v ní nezbývá žádná reziduální kapacita, obsahuje ji pouze v opačném směru) a nová cesta se již po zbytek běhu algoritmu vyhledává v síti reziduální, která se po každém posílení cesty vytvoří. Tok se však upravuje

stále v původní vstupní síti. Takto jsou cesty vyhledávány a posilovány, dokud není v reziduální síti zdroj zcela oddělen od stoku a žádná cesta již tady neexistuje. Celkový tok ze zdroje v této síti je maximální.



Obr. 3.2: Iterace Ford-Fulkersonova algoritmu.

Na obrázku 3.2 je příklad jedné iterace Ford-Fulkersonova algoritmu. Původní síť (1) zde má tok, nejde tedy o první iteraci. Je vytvořena reziduální síť a v ní je určena jedna cesta (2), která se v původní síti posílí o nejmenší kapacitu (3). Následující reziduální síť (4) již neobsahuje saturovanou hranu v původním směru.

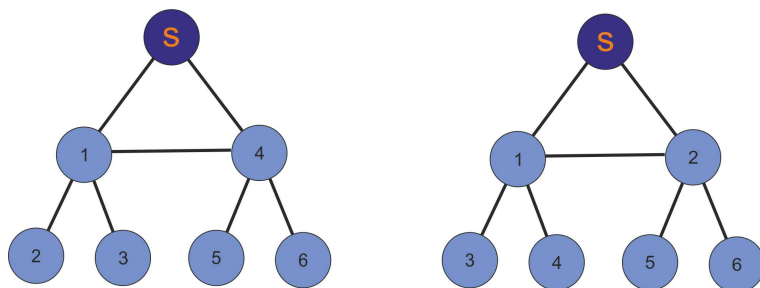
Po posledním kroku algoritmu dostaneme v reziduální síti pouze zdroj s množinou vrcholů na něj napojených a tuto množinu označíme S . Množinu stoku a všech ostatních původních vrcholů sítě, které v poslední reziduální síti nejsou, označíme T . Vrcholy množiny T byly odděleny saturací hran ve všech cestách, které k němu vedly. Skrz takto rozdělenou síť nyní teče nejvyšší možný tok, protože každé další zvýšení toku by již překročilo minimálně jednu kapacitu). Řez z takovéto sítě získáme jako množinu všech hran, jejichž počáteční vrchol leží v množině S a konečný vrchol v množině T . Podle min-cut/max-flow teorému je tento řez minimální [7] [11].

3.1.2 Vliv způsobu vyhledávání cest na výpočetní náročnost

Metoda, jakou jsou ve fázi hledání cesty ze zdroje do stoku v reziduální síti vybírány vrcholy vnáší do algoritmu vliv náhodnosti. Její vhodná volba je proto důležitá pro optimalizaci výpočetního času. Pro jednotlivé metody hledání cest existují vztahy pro závislost výpočetního času na počtech hran, vrcholů a velikosti maximálního toku v nejhorších možných případech, tyto závislosti však nejsou vhodné pro porovnávání algoritmů nebo hodnocení jejich celkové výkonnosti. Pro zhodnocení výkonnosti pro účely segmentace je třeba se zamyslet nad vhodností metod pro využití konkrétně v grafech s mřížkovou strukturou a empiricky otestovat jejich chování [31].

Při hledání cesty algoritmus postupně prohledává vrcholy napojené na zdroj nebo na předchozí prohledaný vrchol a kontroluje, zda daný vrchol není stokem. Při kontrole napojených vrcholů může postupovat dvěma způsoby. Prvním je *depth-first search* (DFS), při kterém se, jak název napovídá, postupuje po vrcholech prioritně do hloubky sítě. Vede-li z kontrolovaného vrcholu hrana k ještě neprozkoumanému vrcholu, pak se jako další zkontroluje tento vrchol (prioritně vrchol vzdálenější od zdroje než předchozí) a k ostatním sousedům předchozího vrcholu se vrátí teprve v případě, že by se takto zkoumaná cesta ukázala jako slepá. V případě nalezení slepé cesty se vrátí k poslednímu vrcholu, ze kterého vedla alternativní hrana a vrcholy mezi ním a slepým koncem se již znovu nekontrolují. Příklad pořadí, ve kterém by byly vrcholy v síti zkontrolovány je na obrázku 3.3 vlevo. Tuto strukturu označujeme jako vyhledávací strom s kořenem ve zdroji.

Druhým možným způsobem je *breadth-first search* (BFS). Vrcholy se v tomto přístupu nekontrolují podél cest, ale po „vrstvách“ bodů se stejnou vzdáleností od zdroje. Tato metoda zajišťuje, že jako první budou k posílení vybrány nejkratší cesty ze zdroje do stoku a teprve po jejich saturaci se algoritmus posune k delším cestám, což vede ke snížení výpočetní náročnosti algoritmů. Tento způsob zařazování vrcholů do stromu je ilustrovaný v obrázku 3.3 vpravo [9] [34].

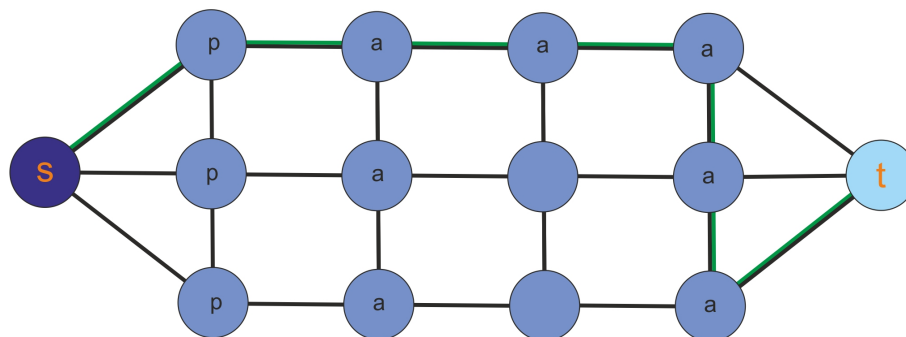


Obr. 3.3: DFS a BFS metody vytváření vyhledávacích stromu pro hledání cest.

3.2 Boykov-Kolmogorovův algoritmus

Algoritmus využívaný v této práci představil v roce 2004 Yuri Boykov a Vladimir Kolmogorov v [2]. Vychází z Ford-Fulkersonova algoritmu a inovaci vnáší právě ve způsobu hledání cesty. Podobně jako v [9] k prohledávání vrcholů využívá metodu *breadth-first search*, po nalezení cesty ze zdroje do stoku v jedné iteraci však vyhledávací stromy vrcholů uchovává v paměti a v další iteraci je znovu využívá. Stromy navíc vytváří dva, jeden s kořenem ve zdroji a jeden ve stoku. Algoritmus se dá rozdělit do tří opakujících se fází: fáze růstu, fáze posílení toku a fáze adopce.

V průběhu první fáze růstu se ze zdroje a stoku začnou vytvářet stromy tak, že se body přímo napojené na terminální vrcholy označí jako aktivní. Následně se podle BFS strom rozšíří o všechny vrcholy přímo na tyto aktivní vrcholy napojené (je důležité si uvědomit, že ve stromu s kořenem ve stoku jsou vrcholy napojené přes hranu opačného směru). Tímto se vytvoří nová vrstva aktivních vrcholů a aktivní vrcholy z předchozího kroku se označí jako pasivní. Takto se fáze růstu z obou terminálních vrcholů opakuje, dokud jeden z aktivních vrcholů nedetekuje ve svém okolí aktivní vrchol protějšího stromu. V takovém případě je detekována cesta ze zdroje do stoku a nastává fáze posílení toku tak, jak je popsáno v kapitole 3.1.1. Příklad cesty nalezené při růstu je zeleně zvýrazněn na obrázku 3.4. Pasivní vrcholy jsou označeny p, aktivní jsou označeny a.



Obr. 3.4: Cesta nalezená ve fázi růstu.

Ještě než se algoritmus posune k fázi růstu další iterace, je třeba ošetřit situace, při kterých může saturace cesty oddělit část stromu v reziduálním grafu od obou terminálních vrcholů. V takovém případě jsou kořeny oddělených stromů označeny jako sirotci (*orphans*). Ve fázi adopce se algoritmus pokusí každému sirotkovi najít náhradní rodičovský vrchol (*parent*) ve stromu, kterému původně náležel. Pokud se náhradní rodič nenajde, všechny vrcholy jeho stromu jsou označeny jako neprohledané. Fáze adopce končí, když v grafu nezůstávají žádní sirotci. Algoritmus se poté vrací k fázi růstu. Pokud již růst není možný (graf neobsahuje žádné aktivní

vrcholy), oba stromy jsou odděleny pouze saturovanými hranami a bylo dosaženo maximálního toku.

4 IMPLEMENTACE ALGORITMU PRO SEGMENTACI OBRAZU

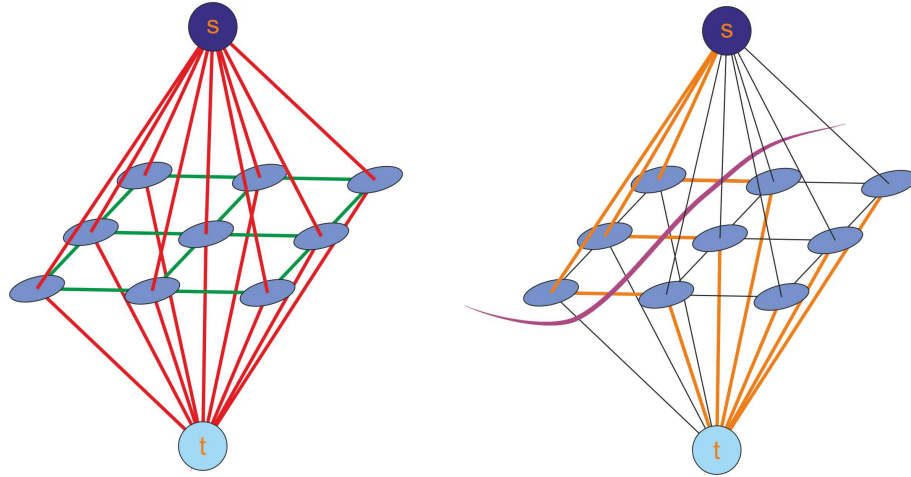
Aby bylo možné řez grafem aplikovat na segmentaci N -rozměrného obrazu, je třeba ze vstupních dat vhodně vytvořit graf a zakomponovat apriorní znalosti. Zároveň musíme vlastnosti obrazu formulovat jako diskrétní energii tak, aby mělo smysl hledat její minimum. V této kapitole bude uvedena metoda konverze obrazových dat do grafu, definována energie segmentace a popsán způsob, jakým může uživatel do segmentace zasahovat.

4.1 Vhodná konverze obrazových dat na graf

V případě dvojrozměrného obrazu bude mít vytvořený graf formu mřížky, kde každý vrchol bude reprezentovat jeden pixel. Hranami bude každý vrchol propojen s předem definovaným počtem sousedních vrcholů. Ty mohou představovat buď 4-okolí, tedy pixely nad, pod, nalevo a napravo, nebo 8-okolí, kde je pomyslný čtverec kolem pixelu doplněný o rohy. Tyto hrany budeme nazývat n -hrany (n -links, od *neighborhood*) a nesou informaci o rozdílu sousedních pixelů. Samotné vrcholy, jak již bylo zmíněno výše, neobsahují žádnou informaci a jejich váha je tedy vždy 1.

Dále do grafu zavedeme dva terminální vrcholy a napojíme je na všechny ostatní vrcholy tzv. t -hranami (t -links, od *terminal*). t -hrany vrcholu představují vztah pixelu k segmentovanému objektu nebo k pozadí, což bude rozvedeno níže. Většinou se uvádí, že zdroj představuje objekt a stok pozadí, reálně by však záměna těchto dvou vrcholů neměla na běh algoritmu vliv. Každý vrchol je tedy napojen čtyřmi nebo osmi n -hranami na své sousedy, jednou t -hranou na zdroj a jednou t -hranou na stok, jak je znázorněno na obrázku 4.1, kde jsou n -hrany vyznačeny zeleně a t -hrany červeně. Vpravo je v takto sestrojeném grafu vyobrazen příklad řezu tvořeného oranžovými hranami. Je třeba si uvědomit, že aby byla splněna podmínka úplného oddělení stoku řezem od zdroje, musí být každý vrchol oddělen od právě jednoho terminálního vrcholu.

U trojrozměrných dat je situace velmi podobná. Mřížka je zde taktéž trojrozměrná, každý vrchol představuje jeden voxel. Sousední vrcholy mohou představovat základní 6-okolí, dále 18-okolí či kompletní 26-okolí. Terminální vrcholy jsou opět propojeny se všemi ostatními vrcholy. Představit si lze i vícerozměrná data, kde by například v časové sekvenci byl každý voxel navíc n -hranou propojen s voxelu stejné pozice v předchozím a následujícím snímku. Metodou řezu grafem lze segmentaci řešit v libovolném počtu dimenzí [3].



Obr. 4.1: Model grafu reprezentujícího obrazová data pro segmentaci.

4.2 Segmentační energie

Segmentační energií se podobně jako v metodách *active contours* [4] nebo *intelligent scissors* [26] vyjadřuje penalizace za vedení hranice objektu konkrétním místem. Algoritmus v této práci bude pracovat se dvěma složkami energie, kde jedna vyjadřuje příslušnost voxelu k objektu nebo pozadí (*unary, region term*) a druhá rozdíl mezi sousedními voxely (*binary, boundary term*). Obecně však může být v energii zahrnuto více složek, například odchylku od předem daného tvaru [13] nebo vzdálenost od předpokládané pozice podle atlasu [33]. Úspěšnost segmentace potom závisí na vhodném vyjádření složek celkové energie a poměru mezi nimi.

4.2.1 Region term

Mějme v mřížce P vytvořený binární vektor všech voxelů $A = (A_1, \dots, A_p, \dots, A_{|P|})$, kde každý voxel A_p může nabývat hodnoty „obj“ pokud náleží ve finální segmentaci objektu nebo „bkg“ pokud náleží pozadí. Hodnota *region term* složky energie je potom definovaná jako

$$R(A) = \sum_{p \in P} R_p(A_p), \quad (4.1)$$

kde $R_p(A_p)$ vyjadřuje penalizaci za přidělení pixelu do množiny objektu nebo pozadí na základě příslušnosti do jejich histogramů. Předpokládá tedy apriorní znalost těchto histogramů. Konkrétně je použito vyjádření

$$R_p(\text{„obj“}) = -\ln \Pr(I_p \mid \text{„obj“}) \quad (4.2)$$

$$R_p(\text{„bkg“}) = -\ln \Pr(I_p \mid \text{„bkg“}). \quad (4.3)$$

Tato definice vychází z vyjádření problému jako *maximum a posteriori* odhadu Markovských náhodných polí podle Bayesovské pravděpodobnosti (*MAP-MRF*) tak, jak je formulován v [24] nebo [19]. Jde tedy o záporný logaritmus pravděpodobnosti, že intenzita pixelu patří do daného histogramu. Penalizace přiřazení do objektu pixelu s intenzitou, která spíš náleží do histogramu pozadí bude tedy větší, než kdyby intenzita vyhovovala příslušnému histogramu objektu.

Do grafu je toto kritérium zahrnuto jako váhy t-hran. Za předpokladu, že zdroj reprezentuje objekt a stok pozadí, bude mít váha t-hrany od voxelu ke zdroji hodnotu $R_p(\text{„obj“})$ podle 4.2 a váha t-hrany od voxelu ke stoku hodnotu $R_p(\text{„bkg“})$ podle 4.3 [3].

4.2.2 Boundary term

Nyní přidejme ke každému prvku z výše definovaného vektoru voxelů A ještě všechny jeho sousední voxely definované okolím jako v kapitole 4.1 a takto vzniklý vektor neuspořádaných dvojic sousedních voxelů $\{p, q\}$ označme N . *Boundary term* je potom definovaná následovně:

$$B(A) = \sum_{\{p,q\} \in N} B_{p,q} \cdot \delta_{A_p \neq A_q}, \quad (4.4)$$

kde člen $\delta_{A_p \neq A_q}$ zajišťuje, že započítány budou do energie pouze dvojice voxelů, které leží na rozhraní objekt/pozadí. Nabývá hodnot

$$\delta_{A_p \neq A_q} = \begin{cases} 1 & \text{pokud } A_p \neq A_q, \\ 0 & \text{pokud } A_p = A_q. \end{cases}$$

Člen $B_{p,q}$ je samotná funkce přidělující energii ve tvaru

$$B_{p,q} = \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{\text{dist}(p, q)}. \quad (4.5)$$

Kromě samotného rozdílu je její hodnota závislá taky na rozptylu kvůli normalizaci dat a euklidovské vzdálenosti vrcholů.

Boundary term složka segmentační energie tedy vyjadřuje penalizaci za nesouvislost dvou sousedních voxelů na rozmezí objektu a pozadí, ačkoliv vhodnější formulace by byla penalizace za souvislost. Funkce této složky totiž nabývá vysokých hodnot v případě, že jsou voxely, mezi kterými je vedena hranice co se jasu týče podobné a nízkých hodnot v případě, že jsou odlišné.

V grafu je toto kritérium zahrnuto jako váhy n-hran, kde váha hrany e_{pq} nabývá hodnot $B(A)$. Zároveň této hodnoty nabývá i váha opačné hrany e_{qp} , jelikož rozdíl je stejný, ačkoliv obecně se dá funkce modifikovat i tak, aby více penalizovala například přechody z tmavého na světlejší voxel či naopak [3].

4.2.3 Celková energie

Celkovou energii segmentace sdružující obě své složky definujeme následovně:

$$E(A) = \lambda \cdot R(A) + B(A). \quad (4.6)$$

Koeficient λ zde vyjadřuje podíl, ve kterém bereme jednotlivé složky v úvahu při stanovování celkové energie. Pokud λ zvolíme výrazně větší než 1, bude algoritmus upřednostňovat segmentaci především na základě intenzity jednotlivých pixelů i za cenu toho, že bude výsledek náchylný k šumu a segmentovaný objekt nesouvislý. Pokud λ zvolíme naopak menší, výsledný segmentovaný objekt bude brát v potaz především ucelenost objektu, což může mít za následek například nežádoucí odseknutí výběžků objektu [3].

V praxi je třeba koeficient λ správně empiricky zvolit pro každou konkrétní aplikaci, jak bude popsáno níže.

4.2.4 Pevná omezení

Výše uvedené vyjádření segmentační energie je dostačující při méně náročných segmentacích, kde je správné řešení často na první pohled zřejmé. To však často neplatí a algoritmus samotný potřebuje k uspokojivému provedení segmentace další informace. Hlavní apriorní informací, kterou může uživatel snadno předložit, je vyznačení polohy několika bodů objektu a pozadí. To můžeme uživateli umožnit například formou interaktivního štětce. Tuto informaci můžeme taky využít pro přesnější vytvoření nebo úpravu histogramu, avšak v našem případě budou histogramy objektu i pozadí předem známy.

Algoritmu tedy předkládáme navíc množinu voxelů náležících do objektu \mathcal{O} a množinu voxelů patřících do pozadí \mathcal{B} . Kromě přiřazení příslušných hodnot ve finálním vektoru segmentace A je však třeba zajistit vliv těchto omezení na celou segmentaci. Hlavní funkcí těchto pevných omezení je tedy především úprava hodnot vah t-hran následujícím způsobem:

$$\{p, S\} = \begin{cases} K & \text{pokud } p \in \mathcal{O} \\ 0 & \text{pokud } p \in \mathcal{B} \end{cases} \quad (4.7)$$

$$\{p, T\} = \begin{cases} 0 & \text{pokud } p \in \mathcal{O} \\ K & \text{pokud } p \in \mathcal{B}, \end{cases} \quad (4.8)$$

kde

$$K = 1 + \max_{p \in P} \sum_{q: (p,q) \in N} B_{p,q}. \quad (4.9)$$

Stejný význam by mělo proměnné K přiřadit hodnotu nekonečno, jak se v některých zdrojích uvádí. Obě definice zaručují, že daná t-hrana voxelu bude mít vždy větší váhu než součet vah všech jeho n-hran. To znamená, že tato hrana nebude nikdy v reziduální síti saturovaná, protože se dříve saturují všechny n-hrany spojující daný voxel s jeho sousedy a řez touto t-hranou tedy nikdy nebude procházet.

Pokud se uživatel rozhodne, že výsledek segmentace není dostatečně uspokojivý, má navíc možnost dodatečně označit nesprávné voxely objektu či pozadí a nechat výsledek upravit. V [3] byla představena metoda, při které v takovém případě může algoritmus znovu využít reziduální graf s maximálním tokem předchozího výsledku, což vede ke značně kratší výpočetní době oproti prvotní segmentaci. V případě přidání nového bodu objektu zvětšíme váhy t-hrany příslušného vrcholu (a tedy i reziduální kapacity v reziduálním grafu) tak, že k váze hrany spojující vrchol se zdrojem připočteme $K + \lambda R_p(\text{„obj“})$ a k váze hrany spojující vrchol se stokem $\lambda R_p(\text{„bkg“})$. Tím dosáhneme toho, že rozdíl mezi vahami t-hran bude opět K a zároveň se v žádné hraně nesníží tok, což by mohlo vést k selhání algoritmu. V takto upraveném grafu může algoritmus pokračovat ve hledání maximálního toku od kroku, ve kterém přestal v prvním běhu.

Použití interaktivního štětce uživatele omezuje na práci ve 2D prostoru navzdory tomu, že zpracovávána data mohou být 3- a více rozměrná. Možným řešením je vyznačit oblasti objektu a pozadí ve více na sebe kolmých rekonstruovaných řezech nebo v celé sérii řezů, což ale může být v případě objemných dat problém. Obecně platí, že čím bližší bude počáteční vymezení objektu a pozadí požadovanému výsledku, tím přesnější bude výsledná segmentace a kratší výpočetní doba.

4.3 Publikované metody segmentace hipokampu využívající řez grafem

V posledních letech bylo diskutováno množství metod segmentace hipokampu, žádná však zatím nebyla úspěšně zavedena v klinické praxi. Některé úspěšné metody byly založeny na atlasové registraci, kde se využívala označená data k sestavení apriorní mapy pravděpodobné polohy hipokampu. Tato mapa byla poté registrována na cílový obraz a nová segmentace vznikala na základě této mapy. Tato metoda však ignoruje informaci o rozložení intenzit v objektu, která může být potenciálně využita pro přesnější segmentaci [15] [14].

Metoda segmentace vycházející z řezu grafem byla prokázána jako vhodná pro segmentaci bílé a šedé mozkové hmoty v [21]. V grafu sestaveném podobně jako v této práci byly přítomny 3 terminální vrcholy, které představovaly bílou hmotu, šedou hmotu a mozkomíšní mok. Navržená metoda střídala iterace segmentace s

iteracemi korekce nehomogenity pole magnetické rezonance, které v každém kroku upravují hodnotu jasu každého voxelu. Metoda se ukázala jako vhodná pro segmentaci mozkových tkání, ačkoliv nebyla potvrzena a srovnána s jinými metodami.

Plně automatická metoda segmentace hipokampu, která v sobě kombinuje atlasovou registraci a metodu minimalizace řezu grafem byla představena v [33]. Graf obsahoval v n -hranách informaci o rozdílu intenzit sousedních voxelů stejně jako v této práci. V t -hranách však byla kromě pravděpodobnosti příslušnosti daného voxelu k objektu či pozadí na základě jeho intenzity obsažena také příslušnost na základě rozložení pravděpodobnostní mapy získané atlasovou registrací. Metoda svou úspěšností předčila metody založené čistě na atlasové registraci.

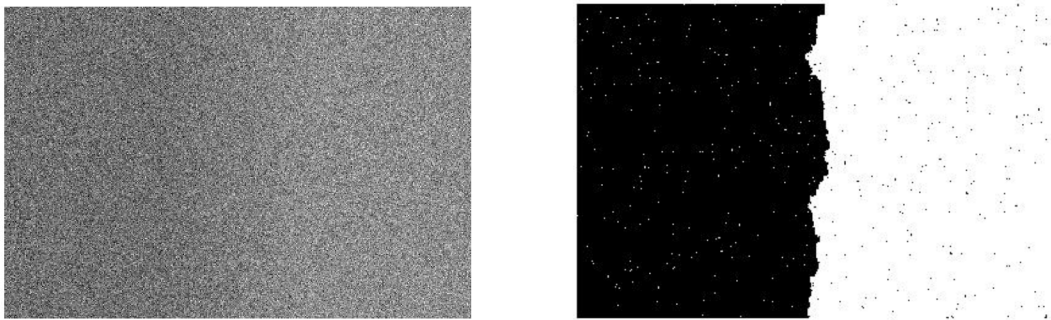
Metoda navržená v [23] doplňuje kombinaci atlasové registrace a minimalizace řezem grafem o následnou optimalizaci výsledku segmentace aplikováním morfologické operace *opening*. Ta má za účel odstranit ze segmentace velmi úzké úseky následované širokými úseky. Výsledná kontura objektu se tak vyhladí a zabrání se nežádoucímu „protékání“ segmentovaného objektu mimo skutečný objekt. Bylo také prokázáno že výsledek grafové segmentace je přesnější, bere-li v potaz potenciální zastoupení více tkání v jednom voxelu, které je způsobené nedostatečným rozlišením snímků magnetické rezonance. Metoda byla skutečně úspěšnější než metoda publikovaná v [33].

Úspěšnost výše uvedených automatických metod se pohybuje při srovnání s ručně segmentovanými snímky hipokampu kolem 90% a dají se tedy považovat za poměrně robustní a vhodné pro rozsáhlé studie. Nelze u nich však dosáhnout naprosto přesného výsledku, protože uživatel nemá možnost výsledek segmentace upravit. Pro tyto účely jsou vhodnější metody poloautomatické. Mezi těmito však zatím metoda minimalizace řezu grafem není příliš rozšířená.

4.4 Testování na zkušebních 2D a 3D datech

Nejprve byl algoritmus otestován na jednoduchém Gaussovsky zašuměném černobílém obrázku s přechodem ve formě šedého klínu, viz 4.2 vlevo. Hodnoty vah t -hran byly určeny jako kumulativní pravděpodobnosti vycházející z Gaussových rozložení intenzit v referenčních regionech objektu (u pravého okraje) a pozadí (u levého okraje). Hodnoty vah n -hran ve 4-okolí každého vrcholu byly přiřazeny podle vztahu 4.5. Těmito hranami je graf kompletně definován a je předložen algoritmu. Výsledná matice s hodnotami výsledného vektoru A je na obrázku 4.2 vpravo. Je vidět, že segmentace proběhla i přes silné zašumění poměrně úspěšně při koeficientu λ zvoleném 0,55.

Dále byla jako testovací data vytvořena série různě velkých elips pro vytvoření

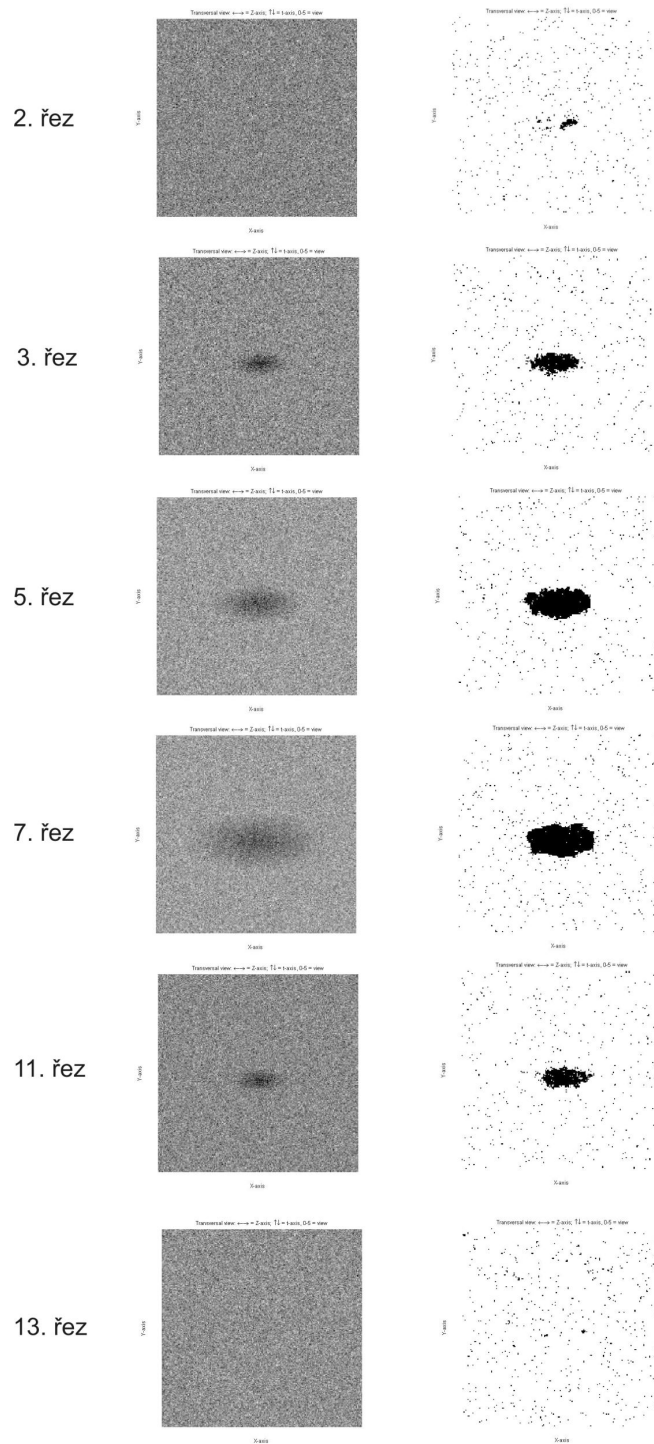


Obr. 4.2: Test algoritmu pro 2D vstupní data.

trojrozměrného modelu oválu. Hranice objektu jsou opět rozmazané a celý objem je zašuměný. Podmínky a rozměry objemu jsou tedy podobné, jako u snímku hipokampu z magnetické rezonance. Výsledky segmentace v jednotlivých řezech jsou uvedeny v obrázku 4.3 vpravo s původními daty vlevo.

Algoritmus byl opět schopen získat hrubý tvar objektu i přes značné zarušení a nejasné hranice. Lze si všimnout, že objekt ve výsledku zasahuje i do řezu, ve kterém by být neměl. Toto „přetékání“ je u automatických segmentačních metod častým jevem, který je však možno řešit zavedením možnosti označení těchto segmentovaných voxelů jako chybných způsobem, jaký byl popsán v kapitole 4.2.4. Protože histogramy objektu a pozadí jsou si v těchto vstupních datech blíží, koeficient λ byl za účelem zvýraznění rozdílu zvolen 1,5.

Díky těmto výsledkům lze tedy metodu segmentace nalezením minimálního řezu grafem pokládat za vhodnou pro použití při poloautomatickém či automatickém segmentování hipokampu ze snímků magnetické rezonance.



Obr. 4.3: Test algoritmu pro 3D vstupní data.

5 IMPLEMENTACE V PROGRAMOVÉM PROSTŘEDÍ MATLAB®

V této práci je použita knihovna pro implementaci Boykov-Kolmogorova algoritmu do MATLABu® zveřejněná v [30]. Vstupem do algoritmu jsou dvě matice nesoucí informace o váhách t-hran a n-hran. Výstupem samotného algoritmu je jednorozměrný binární vektor A . Jeho prvky představující voxely A_p nabývají hodnot 0, pokud byly zařazeny do pozadí a 1, pokud byly zařazeny do objektu.

5.1 Popis metody zpracování vstupních dat

Pro účely této práce byla dále vytvořena funkce pro tvorbu vektorů souřadnic všech dvojic pixelů, mezi kterými existuje n-hrana. Jejím vstupem jsou pouze rozměry snímku. Základním způsobem je, v případě trojrozměrných dat, brát v potaz 6-okolí každého voxelu. Je třeba si uvědomit, že ve směrovaném grafu existuje mezi dvěma sousedními voxely hrana v obou směrech, což znamená, že s každým voxelem stoupne počet hran o šest, kde každá obsahuje informace o souřadnicích počátku, souřadnicích konce a hodnotě váhy. V případě snímků magnetické rezonance o rozměrech $512 \times 512 \times 160$ to je přes 250 milionů hran. Operace s tímto vektorem jsou tedy paměťově vysoce náročné.

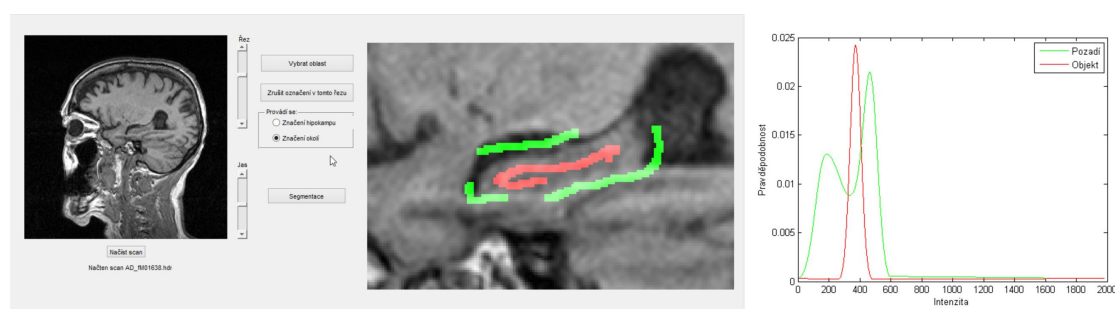
Protože jde o metodu interaktivní, byla také vytvořena funkce pro zpracování vstupních uživatelských dat do pravděpodobnostních funkcí. Vstupem do ní je kromě patientského snímku také nulová matice o velikosti shodné se snímkem, ve které jsou na pozicích označených uživatelem jako objekt (hipokampus) hodnoty 1 a na pozicích označených jako okolí hodnoty 2. Funkce nejprve z každé množiny označených bodů vytvoří histogram originálního obrazu, který se poté pokusí co nejpřesněji proložit distribuční funkcí. V případě množiny bodů objektu odpovídá distribuční funkce normálnímu rozložení, u množiny bodů pozadí jsou zde však dva píky odpovídající bodům s vyšší intenzitou než hipokampus (bílá hmota alveu) a bodům s nižší intenzitou (postranní mozková komora). Takového průběhu distribuční funkce můžeme dosáhnout pomocí jádrového odhadu hustoty pravděpodobnosti (*kernel density estimation* [16]). Dvě získané funkce dále normalizujeme vydělením jejich kumulativním součtem, který představuje jejich plochu pod křivkou. Takto dosáhneme funkcí odpovídajících pravděpodobnostem, že bod dané intenzity náleží do histogramu objektu a histogramu pozadí, které jsou potřeba při výpočtu vah t-hran. Dále je z těchto bodů získána lokální směrodatná odchylka potřebná při výpočtu vah n-hran.

Výstupy dvou výše zmíněných funkcí jsou společně s originálními patientskými snímky, maticí uživatelem zadaných pevných omezení vstupem do funkce, která pro-

vádí samotnou grafovou segmentaci. Nejprve v ní dojde k sestavení matice t-hran a jejímu naplnění příslušnými hodnotami z pravděpodobnostní funkce objektu resp. pozadí podle rovnic 4.2 a 4.3. Dále jsou t-hrany vedoucí z voxelů označených uživatelem jako objekt či pozadí přepsány podle rovnic 4.7 a 4.8. Následně se z vektoru souřadnic n-hran vytvoří matice a naplní se příslušnými hodnotami podle rovnice 4.5. Tyto dvě matice vstupují do samotného min cut/max flow algoritmu, jehož výstupem je binární vektor, který po převedení zpět do tvaru vstupních dat představuje příslušnost jednotlivých voxelů k objektu nebo pozadí.

Vzhledem k povaze snímků mozku však dojde kromě segmentace hipokampu také k segmentaci okolních tkání, které mají podobný intenzitní profil jako hipokampus. Z tohoto důvodu vstupuje výsledek ještě do funkce, která z výsledku segmentace odstraní všechny segmentované objekty, které nejsou přímo propojeny s žádným z uživatelem definovaných voxelů hipokampu. Takto separovaný výsledek segmentace obsahuje pouze voxely skutečně vyhodnocené jako hipokampus.

5.2 Uživatelské rozhraní programu



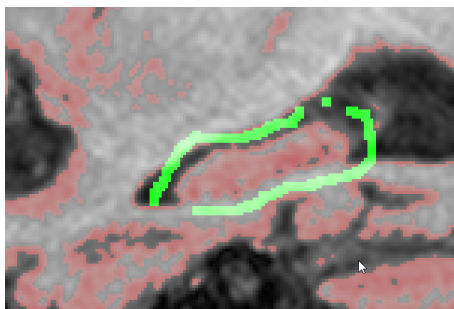
Obr. 5.1: Uživatelské rozhraní navrženého softwaru (vlevo) a zobrazení pravděpodobnostních funkcí na základě histogramu označených voxelů (vpravo).

Uživatelské rozhraní prototypu navrženého softwaru (obrázek 5.2 vlevo) se skládá ze tří základních bloků. Vizualizace načtených dat, panel s kontrolními prvky a editor pro zadávání bodů objektu a pozadí. Uživatel nejprve zvolí pacienta, jehož data si přeje načíst. Ty se zobrazí v levé části grafického rozhraní. Po nastavení jasu si uživatel vybere, ve kterém řezu chce začít zadávat body a nastaví vhodnou úroveň jasu snímku. Poté pomocí tlačítka „Vybrat oblast“ zvolí přibližnou pozici středu hipokampu a tím zobrazí zvětšenou oblast hipokampu do pravé části rozhraní. V této části nyní může pomocí štětce zvolit libovolné množství bodů hipokampu a jeho okolí.

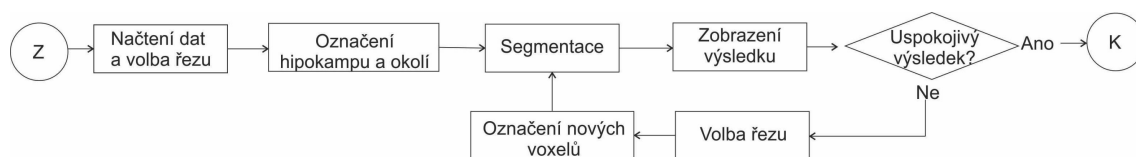
Kvalita výsledku segmentace je samozřejmě tím vyšší, čím precizněji uživatel určí pevná omezení v problematických částech snímku. To ovšem v kombinaci s odezvou,

se kterou program zpracovává pozici kurzoru znamená časté chybné označení jednoho či více bodů. V tomto případě lze označení v daném řezu zrušit a začít znovu pomocí tlačítka „Zrušit označení v tomto řezu“.

Ve chvíli, kdy je hipokampus označen v dostatečném množství řezů, lze provést grafovou segmentaci tlačítkem „Segmentace“. V novém okně se zobrazí pravděpodobnostní funkce získané z histogramu a výsledek segmentace se potom zobrazí do pravé i levé části rozhraní, kde jsou všechny voxely vyhodnocené jako hipokampus zvýrazněny červeně. Výsledek první segmentace je však často nedostatečně kvalitní. Často dochází například k nechtěnému propojení hipokampu s okolní tkání šedé kůry mozkové skrze malou část špatně detekované hranice. V takovém případě je jako hipokampus chybně označena i velká část okolní tkáně, kterou je třeba odstranit (viz obrázek 5.3). K řešení nastalé situace stačí navíc k původním pevným omezením blíže specifikovat problematickou část hranice několika dalšími pevnými omezeními v malém množství řezů a segmentaci provést znovu. Takto lze postupovat, dokud není výsledek segmentace dostatečně přesný. Postup celého procesu je shrnut ve schématu na obrázku 5.4.



Obr. 5.2: Chybná detekce šedé kůry mozkové.



Obr. 5.3: Schéma procesu interaktivní segmentace.

Vzhledem k nespolehlivému a nepohodlnému způsobu značení pevných omezení, nutnosti zdlouhavě vybírat oblast zájmu v každém řezu hipokampu a občasné nestabilitě prostředí MATLAB je však nepravděpodobné, že by se dal software v této podobě použít v klinické nebo výzkumné praxi.

5.3 Implementace jako modul pro software 3D Slicer

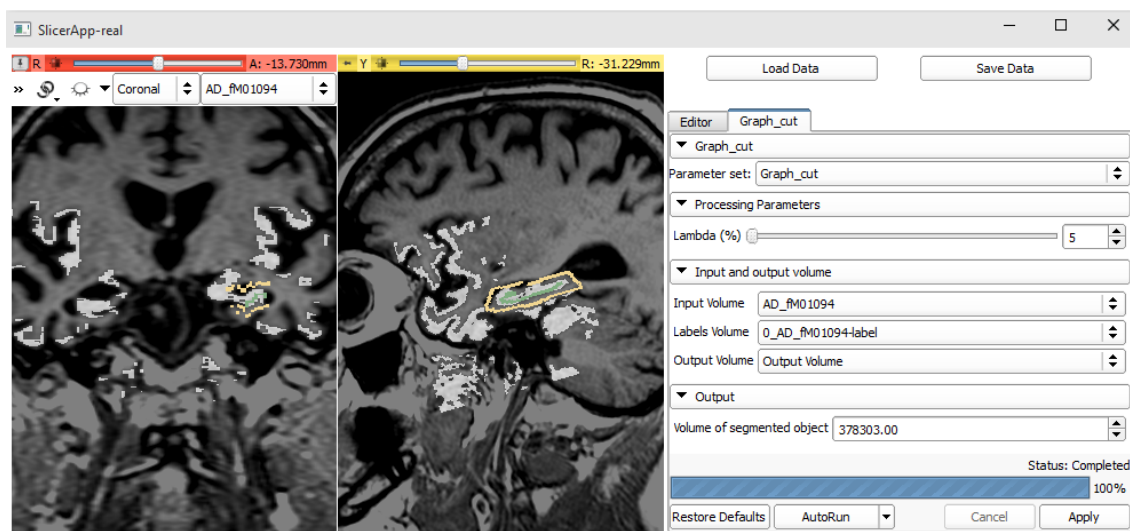
3D Slicer je softwarová platforma pro analýzu a vizualizaci medicínských obrazových dat původně vyvinutý při spolupráci laboratoře umělé inteligence MIT a laboratoře operačního plánování v nemocnici Brigham and Women's Hospital. 3D Slicer je zdarma, open-source a přístupný na většině operačních systémů. Kromě základní manipulace s daty je k dispozici široká nabídka zásuvných modulů s dalšími funkcemi jako jsou různé možnosti renderování objemů, registrací multimodálních dat nebo automatických segmentací. Jedním ze zásuvných modulů pro 3D Slicer je modul Matlab Bridge, který umožňuje použít obrazová data ze 3D Sliceru jako vstup funkce Matlabu. Po provedení funkce je výstupní objem zobrazen zpět ve 3D Sliceru.

3D Slicer využívá ve svém rozhraní jazyk MRML (Medical Reality Markup Language), který se snaží sjednotit standardy pro manipulaci s medicínskými obrazovými daty. Mimo jiné umožňuje uživatelsky přátelskou manipulaci s 3D objemy jako ovládání pozice, přiblížení, jasů a kontrastu na myši tak, jak je lékařská odborná obec zvyklá. Jedním ze základních modulů je také editor, který právě díky této celkově pohodlné manipulaci umožňuje velmi intuitivní značení dat. Původním účelem tohoto modulu je ruční segmentace.

Software je celkově velmi snadno rozšiřitelný o další moduly programované v Pythonu nebo C++. Protože pro výzkumné účely je však jen málokdy potřebné mít přístup ke všem modulům najednou, existuje také možnost definovat uživatelské prostředí obsahující pouze využívané moduly, tzv. *slicelet*. Slicelety jsou programovány v Pythonu s knihovnou PyQt, která umožňuje v Pythonu využívat multiplatformní rámec pro vytváření uživatelských prostředí Qt.

Pro účely této práce byl vytvořen slicelet obsahující modul pro načtení a vizualizaci objemu ve dvou projekcích, editor a Matlab Bridge modul pro grafovou segmentaci definovaný v Matlabu a popsáný v předchozí kapitole (obrázek 5.5). Práce s programem vypadá opět podobně. Nejprve jsou načtena obrazová data pomocí tlačítka Load Data, potom je v prostředí editoru dvěma různými štětci (zeleným a žlutým, ty totiž v matici označených bodů 3D Sliceru odpovídají hodnotám 1 a 2) označeno několik voxelů hipokampu a pozadí. Následně se zvolí modul pro grafovou segmentaci, zvolí se vstupní objemy a koeficient λ a spustí se samotná segmentace, která opět zobrazí také pravděpodobnostní funkce.

Výsledná binární matice se uloží do nového objemu. Ten lze zobrazit přes původní data s volitelnou průhledností, což nám umožní posoudit úspěšnost segmentace podobně jako v uživatelském prostředí definovaném v Matlabu. Zobrazí se také informace o celkovém objemu segmentovaného objektu, která však nese pouze in-



Obr. 5.4: Slicelet vytvořen pro grafovou segmentaci.

formaci o počtu voxelů a ne o skutečných rozměrech. Jestliže je výsledek nepřesný jako na obrázku 5.5, je zde opět možnost provést další iteraci označení voxelů a nového výpočtu segmentace a postup opakovat tak dlouho, dokud nebude výsledek přijatelný.

6 DISKUZE ÚSPĚŠNOSTI ALGORITMU

Pro dosažení uspokojivé segmentace hipokampu bylo při testování programu většinou zapotřebí označit 10 až 15 sagitálních řezů, popřípadě vymezit hranici mezi hipokampem a amygdalou v koronálních řezech. Časová náročnost výpočtu jedné segmentace se pohybuje kolem dvaceti sekund, segmentace byly zapotřebí často upravit a výpočet proto probíhal dvakrát nebo třikrát.

Koeficient λ se pro jednotlivé pacienty lišil, vždy se však pohyboval v řádech setin. To znamená, že pro segmentaci hipokampu je vhodnější preferovat podmínku *region term*. Obecně potom platilo, že s rostoucím koeficientem λ stoupal objem segmentovaného tělesa a tím také pravděpodobnost nechtěného napojení segmentovaného hipokampu na sousední amygdalu a okolní šedou kůru mozkovou jako na obrázcích 5.3 a 5.5. Volbou nižšího koeficientu λ se tomuto lze vyhnout, cenou je však výsledek segmentace, který nemusí úplně přiléhat až na hrany hipokampu. Při hledání vhodného koeficientu λ je proto nutné najít vhodný kompromis.

Ukážou-li průběhy pravděpodobnostních funkcí, že se histogramy objektu a pozadí příliš překrývají, lze postupovat dvěma způsoby. Prvním je, že se zvolí vyšší koeficient λ a algoritmus tak začne především hledat hrany (klást větší důraz na *boundary term* podmínku). V případě hipokampu to však většinou způsobí selhání algoritmu způsobené ztrátou důležité apriorní informace. Vhodnější je proto pokusit se ovlivnit tvar pravděpodobnostní funkce získané z histogramu pozadí tím, že se označí přibližně stejné množství pixelů výrazně tmavších a výrazně světlejších než intenzita hipokampu. To způsobí snadnější rozlišení dvou intenzitních píků pozadí v histogramu a tím často vynikne pík hipokampu, který se nachází mezi nimi, viz obrázek 5.2 vpravo. U některých pacientů jsou však histogramy hipokampu a okolních tkání tak podobné, že segmentace není možná.

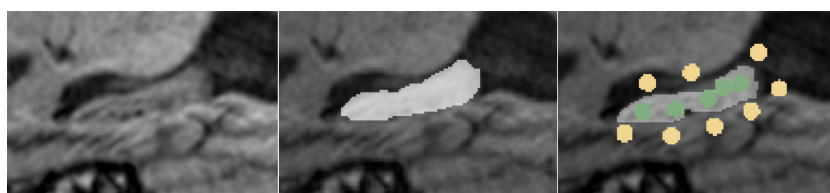
Jak již bylo výše několikrát zmíněno, jednou z nejproblematictějších částí je z pohledu segmentace hranice s amygdalou. Tu je často potřeba definovat označením hranice jako voxelů pozadí v několika řezech. Protože to však zanáší do histogramu pozadí voxely s intenzitou podobnou hipokampu, je vhodné pro oddělení amygdaly použít menší štětec. Zároveň je však třeba brát v potaz, že do výpočtu segmentace v MATLABu vstupuje pouze matice masky s označenými voxely o stejných rozměrech jako patientský snímek, tedy $512 \times 512 \times 160$. Při volbě průměru štětce menšího než 1px tedy často dojde k tomu, že jej algoritmus nebere v potaz jako pevné omezení i přesto, že se v programu vyznačená hranice zobrazí. To je způsobeno tím, že 3D Slicer data vyhlazuje interpolací na jemnější rozlišení.

Objektivní statistické hodnocení úspěšnosti segmentačního algoritmu bohužel není možné vzhledem k velmi malému množství lékařem označených snímků, které jsou v práci považovány za zlatý standard, se kterým se výstupy algoritmu srovná-

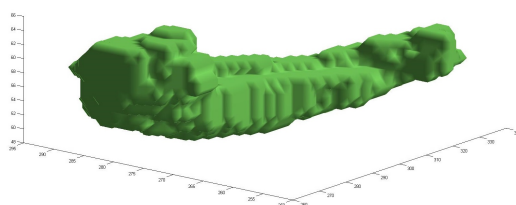
vají. Přesto je z výsledků zřejmé, že algoritmus je ve většině případů schopen dosáhnout požadovaného výsledku a to jak u pacientů ze zdravé kontrolní skupiny, tak u pacientů trpících Alzheimerovou chorobou. Pro úspěšnou segmentaci je však zapotřebí mít dostatečné znalosti anatomie mozku a problematické hranice hipokampu ve snímku aspoň v několika řezech nalézt a vyznačit. Na následujících obrázcích jsou zobrazeny výstupy segmentačního algoritmu srovnané s lékařem označenými daty v ukázkových řezech včetně popisu náročnosti a parametrů segmentace.



Obr. 6.1: Původní snímek hipokampu zdravého pacienta (vlevo), lékařem označený hipokampus (uprostřed), výsledek výpočtu segmentace s označenými voxely (vpravo). Snímek byl podobným způsobem označen v 7 řezech před prvním výpočtem a poté v dalších 5 řezech před finálním výpočtem segmentace s koeficientem $\lambda = 0,03$.



Obr. 6.2: Původní snímek hipokampu pacienta s Alzheimerovou chorobou (vlevo), lékařem označený hipokampus (uprostřed), výsledek výpočtu segmentace s označenými voxely (vpravo). Snímek byl podobným způsobem označen v 8 řezech před prvním výpočtem a poté v dalších 8 řezech před finálním výpočtem segmentace s koeficientem $\lambda = 0,02$.



Obr. 6.3: 3D vizualizace segmentovaného hipokampu.

7 ZÁVĚR

Cílem práce bylo seznámení s teorií grafů a publikovanými algoritmy grafových metod pro segmentaci obrazových dat a realizace programu pro poloautomatickou segmentaci hipokampu. Vysvětlení základních pojmů a operací je obsaženo v kapitole 2. Graf byl definován jako struktura tvořená vrcholy a hranami o daných vlastnostech, dále byl vysvětlen řez grafem a jeho minimalizace. Způsob, jakým lze minimalizací řezu reprezentovat problém segmentace obrazu je uveden v kapitole 4 včetně vhodného způsobu konverze obrazových dat do struktury grafu a definic segmentačních energií. Dále kapitola obsahuje rešerši publikovaných segmentačních metod hipokampu využívající řez grafem.

Kapitola 3 obsahuje popis algoritmů vyvinutých pro minimalizaci řezu grafem. Představena je metoda *augmenting path* a Ford-Fulkersonův algoritmus a dále na jeho základě vytvořený Boykov-Kolmogorovův algoritmus, který je díky své optimalizaci výpočetní náročnosti použit i v této práci.

Praktické seznámení s funkčností a úspěšností algoritmu na testovacích datech je shrnuto na konci kapitoly 4. Metoda je při vhodné reprezentaci a definici segmentačních energií hodnocena jako dostatečně přesná na to, aby byla použita pro segmentaci hipokampu z dat magnetické rezonance, jejíž význam je shrnut v kapitole 1.

Kapitola 5 obsahuje popis implementace algoritmu pro poloautomatickou segmentaci hipokampu na základě poznatků z kapitoly 4 pomocí programového prostředí MATLAB®. Představila také dva vytvořené typy grafického uživatelského prostředí umožňujícího do procesu zasahovat vyznačením několika bodů hipokampu a pozadí. Grafické prostředí vytvořené pomocí 3D Sliceru bylo stanoveno jako vhodnější varianta pro práci s programem.

Výsledky a úspěšnost segmentace byly diskutovány v kapitole 6. I přes omezenou možnost objektivního hodnocení úspěšnosti segmentace je zřejmé, že přes občasné komplikace metoda u většiny pacientů usnadňuje náročnou segmentaci hipokampu ze snímků magnetické rezonance.

LITERATURA

- [1] BERESFORD, T. P., ARCINIEGAS, D. B., ALFERS, J., CLAPP, L., MARTIN, B., DU, Y., LIU, D., SHEN, D. and DAVATZIKOS, C. (2006), Hippocampus Volume Loss Due to Chronic Heavy Drinking. *Alcoholism: Clinical and Experimental Research*, 30: 1866–1870. doi: 10.1111/j.1530-0277.2006.00223.x
- [2] BOYKOV, Y. a V. KOLMOGOROV. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2004, vol. 26, issue 9, s. 1124-1137. DOI: 10.1109/TPAMI.2004.60. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1316848>
- [3] BOYKOV, Yuri a Gareth FUNKA-LEA. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision*. 2006, vol. 70, issue 2, s. 109-131. DOI: 10.1007/s11263-006-7934-5. Dostupné z: <http://link.springer.com/10.1007/s11263-006-7934-5>
- [4] CASELLES, V., KIMMEL, R., and SAPIRO, G. 1997. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79
- [5] CLAYTON, Nicky S, Juan C REBOREDA a Alex KACELNIK. Seasonal changes of hippocampus volume in parasitic cowbirds. *Behavioural Processes*. 1997, vol. 41, issue 3, s. 237-243. DOI: 10.1016/S0376-6357(97)00050-8. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0376635797000508>
- [6] CORMEN, Thomas H. Introduction to algorithms. 2nd ed. Cambridge: MIT Press, c2001, xxi, 1180 s. ISBN 02-620-3293-7
- [7] DEMEL, Jan. Segmentace 3D obrazových dat s využitím grafové reprezentace. Brno, 2014. Diplomová práce. Vysoké učení technické v Brně
- [8] DIESTEL, Reinhard. Graph theory [online]. 2nd ed. New York: Springer, c2000, xiv, 312 p. [cit. 2014-10-25]. ISBN 03-879-8976-5. Dostupné z: <http://www.esi2.us.es/~mbilbao/pdffiles/DiestelGT.pdf>
- [9] E.A. Dinic, “Algorithm for Solution of a Problem of Maximum Flow in Networks with Power Estimation,” *Soviet Math. Dokl.*, vol. 11, pp. 1277-1280, 1970
- [10] FIGUEIRA, Arnaldo. Princeton University - Course: Algorithms, Part II: Ford-Fulkerson Algorithm. In: Youtube [online]. 2013 [cit. 2014-11-26]. Dostupné z: <https://www.youtube.com/watch?v=-8MwfgB-lyM>

- [11] FORD, L a D FULKERSON. Flows in networks. New Jersey: Princeton University Press, 2011, xix, 194 s. princeton landmarks in mathematics. ISBN 978-0-691-14667-6
- [12] FOX, N. C., E. K. WARRINGTON, P. A. FREEBOROUGH, P. HARTIKAINEN, A. M. KENNEDY, J. M. STEVENS a M. N. ROSSOR. Pre-symptomatic hippocampal atrophy in Alzheimer's disease. *Brain*. 1996, vol. 119, issue 6, s. 2001-2007. DOI: 10.1093/brain/119.6.2001. Dostupné z: <http://brain.oxfordjournals.org/cgi/doi/10.1093/brain/119.6.2001>
- [13] FREEDMAN D., T. ZHANG. Interactive Graph Cut Based Segmentation With Shape Priors. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005. IEEE Computer Society. June 2005, Vol. 1, pp. 755-762. ISSN 1063-6919
- [14] GOUTTARD, Sylvain, Martin STYNER, Sarang JOSHI, Rachel G. SMITH, Heather CODY HAZLETT, Guido GERIG, Josien P. W. PLUIM a Joseph M. REINHARDT. . *NeuroImage*. 2006, vol. 33, issue 1, 65122J-65122J-11. DOI: 10.1117/12.708626. Dostupné z: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1299623>
- [15] HECKEMANN, Rolf A., Joseph V. HAJNAL, Paul ALJABAR, Daniel RUECKERT a Alexander HAMMERS. Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. *NeuroImage*. 2006, vol. 33, issue 1, s. 115-126. DOI: 10.1016/j.neuroimage.2006.05.061. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1053811906006458>
- [16] VOPATOVÁ, Kamila. Volba vyhlazovacích parametrů pro jádrové odhady hustot [online]. 2012 [cit. 2015-05-21]. Disertační práce. Masarykova univerzita, Přírodovědecká fakulta. Vedoucí práce Ivanka Horová. Dostupné z: http://is.muni.cz/th/63985/prif_d/
- [17] FRODL, T. Hippocampal Changes in Patients With a First Episode of Major Depression. *American Journal of Psychiatry*. vol. 159, issue 7, s. 1112-1118. DOI: 10.1176/appi.ajp.159.7.1112. Dostupné z: <http://ajp.psychiatryonline.org/article.aspx?articleID=175630>
- [18] GOLDBERG, Andrew V. a Robert E. TARJAN. A new approach to the maximum-flow problem. *Journal of the ACM*. vol. 35, issue 4, s. 921-940. DOI: 10.1145/48014.61051. Dostupné z: <http://portal.acm.org/citation.cfm?doid=48014.61051>

- [19] GREIG, D., PORTEOUS, B., and SEHEULT, A. 1989. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279
- [20] GURVITS, Tamara V., Martha E. SHENTON, Hiroto HOKAMA, Hirokazu OHTA, Natasha B. LASKO, Mark W. GILBERTSON, Scott P. ORR, Ron KIKINIS, Ferenc A. JOLESZ, Robert W. MCCARLEY a Roger K. PITMAN. Magnetic resonance imaging study of hippocampal volume in chronic, combat-related posttraumatic stress disorder. *Biological Psychiatry*. 1996, vol. 40, issue 11, s. 1091-1099. DOI: 12091188. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0006322396002296>
- [21] Z. SONG, N. TUSTISON, and B. AVANTS et al., “Integrated graph cuts for brain MRI segmentation,” *MICCAI 2006, Pt 2*, vol. 4191, pp. 831–838, 2006
- [22] KONRAD, C., T. UKAS, C. NEBEL, V. AROLT, A.W. TOGA a K.L. NARR. Defining the human hippocampus in cerebral magnetic resonance images—An overview of current segmentation protocols. *NeuroImage*. 2009, vol. 47, issue 4, s. 1185-1195. DOI: 10.1016/j.neuroimage.2009.05.019. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1053811909004972>
- [23] KWAK, Kichang, Uicheul YOON, Dong-Kyun LEE, Geon Ha KIM, Sang Won SEO, Duk L. NA, Hack-Joon SHIM a Jong-Min LEE. Fully-automated approach to hippocampus segmentation using a graph-cuts algorithm combined with atlas-based segmentation and morphological opening. *Magnetic Resonance Imaging*. 2013, vol. 31, issue 7, s. 1190-1196. DOI: 10.1016/j.mri.2013.04.008. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0730725X13001513>
- [24] LÉZORAY, Olivier a Leo GRADY. *Image Processing and Analysis with Graphs: Theory and Practice*. CRC Press, 2012. ISBN 978-1-4398-5507-2
- [25] VAN DER LIJN, Fedde, Tom DEN HEIJER, Monique M.B. BRETELER a Wiro J. NIESSEN. Hippocampus segmentation in MR images using atlas registration, voxel classification, and graph cuts. *NeuroImage*. 2008, vol. 43, issue 4, s. 708-720. DOI: 10.1016/j.neuroimage.2008.07.058. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1053811908008860>
- [26] MORTENSEN, E.N. and BARRETT, W.A. 1998. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60:349–384
- [27] PIRZADA, Shariefuddin a Ashay DHARWADKER. Applications of Graph Theory. *Journal of the Korean Society for Industrial and Applied*

- Mathematics [online]. 2007, Vol.11 NO.4 [cit. 2014-10-04]. Dostupné z: <http://www.dharwadker.org/pirzada/applications/>
- [28] PiuChePui. MEMORIA: NEUROFISIOLOGIA, RECUPERO E OBLIO [online]. PiuChePui, [cit. 21. 11. 2013]. Dostupné z URL: www.piuhepuoi.it
- [29] PHELPS, Jim. PsychEducation [online]. 2000 [cit. 2014-11-08]. Dostupné z: <http://www.psycheducation.org/index.html>
- [30] RUBINSTEIN, Miki. A wrapper library for Boykov and Kolmogorov max-flow/min-cut implementation. In: Mathworks [online]. 2008 [cit. 2014-12-22]. Dostupné z: <http://www.mathworks.com/matlabcentral/fileexchange/21310-maxflow>
- [31] SEDGEWICK, Robert. PRINCETON UNIVERSITY. Finding Paths in Graphs [online]. 2004 [cit. 2014-12-19]. Dostupné z: <http://www.cs.princeton.edu/rs/talks/paths.pdf>
- [32] STICH, Timo. NVIDIA. Graph Cuts with CUDA. San Jose, 2009. Dostupné z: <http://www.nvidia.com/content/gtc/documents/>
- [33] VAN DER LIJN, Fedde, Tom DEN HEIER, Monique M. B. BRETELER a Wiro J. NIESSEN. Hippocampus segmentation in MR images using atlas registration, voxel classification, and graph cuts. Neuroimage. 2008, č. 43.
- [34] ZADEH, Norman. Theoretical Efficiency of the Edmonds-Karp Algorithm for Computing Maximal Flows. ISSN 10.1145/321679.321693