VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SEGMENTACE STRUKTUR MIKROSKOPICKÝCH DAT MOZKU

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. SAMUEL LÁSKA
AUTHOR

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SEGMENTACE STRUKTUR MIKROSKOPICKÝCH DAT MOZKU

SEGMENTATION OF MICROSCOPIC BRAIN STRUCTURES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. SAMUEL LÁSKA
AUTHOR

VEDOUCÍ PRÁCE Ing. RADIM BURGET, Ph.D.
SUPERVISOR

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Samuel Láska	***ID:***	106589
Ročník:	2	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Segmentace struktur mikroskopických dat mozku

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou signálového zpracování třírozměrných dat. V jazyce JAVA implementujte obrazový filtr, který bude rotovat a rozostřovat data. S pomocí takto rozostřených dat natrénujte model pro segmentaci struktur mozku a zhodnoťte přesnost takového řešení.

DOPORUČENÁ LITERATURA:

[1] Rafael C. Gonzalez and Richard E. Woods. 2006. Digital Image Processing (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[2] Milan Sonka, Vaclav Hlavac, and Roger Boyle. Image Processing: Analysis and Machine Vision. CL-Engineering, 2 edition, September 1998.

Termín zadání:	11.2.2013	***Termín odevzdání:***	29.5.2013

Vedoucí práce: Ing. Radim Burget, Ph.D.
Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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

Táto práca za zaoberá spracovaním obrazových medicínskych dát za pomoci programovacieho jazyka Java. Hlavný prínos práce je vytvorenie algoritmov pre extrakciu príznakov z 3D dát a následné overenie výsledkov pre problematiku zobrazovania mikroskopických 3D dát mozgu a vo vytvorení obrazových filtrov, ktoré sú vytvorené vo forme operátorov pre program RapidMiner. Následne je vytvorený segmentačný proces na úrovni 2D a 3D, ktorého výstupom sú 3D štruktúry nasegmentovaných buniek mozgu. Ďalej boli porovnané segmentačné algoritmy na základe výslednej formy segmentovaných štruktúr a porovnanie tohoto prístupu s inými prácami.

KĽÚČOVÉ SLOVÁ

segmentácia, 3D, Java, RapidMiner, spracovanie obrazu, elektrónový mikroskop, rotácia, bunkové štruktúry

ABSTRACT

This thesis is involved in image processing of medical data and its implementation using Java programming language. The main contribution of this thesis is creation of algorithms for feature extraction from 3D data and subsequent verification of the results for the issue of imagining 3D brain data, and creation of image filters and their implementation in the program RapidMiner. Consequently, the segmentation process is created at the 2D and 3D level, and output of 3D level segmentation are segmented brain structures. Furthermore, segmentation algorithms were compared on the basis of the final form of segmented structures and this approach was compared with other works.

KEYWORDS

segmentation, 3D, Java, RapidMiner, image processing, electron microscope, rotation, neurites

LÁSKA, Samuel *Extrakce pžíznaků z trojrozměrných obrazových dat*: diplomová práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 66 s. Vedúci práce bol Ing. Radim Burget, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju diplomovoú prácu na tému „Extrakce pžíznaků z trojrozměrných obrazových dat“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorském, o právoch súvisejúcich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhé, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno ....

(podpis autora)

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi Ing. Radimovi Burgetovi, Ph.D. za odborné vedenie, konzultáce, trpezlivosť a podnetné návrhy k práci.

Brno ....

(podpis autora)

OBSAH

Úvod **11**

1 Medicínske snímky **13**
1.1 Technológie tvorby a zobrazenia medicínkych dát 13
1.1.1 RTG 14
1.1.2 X-ray CT 14
1.1.3 PET 14
1.1.4 MRI 14

2 Obrazové spracovanie medicínskych snímkov **16**
2.1 Segmentácia medicínskych snímkov 16
2.2 Rozdelenie segmentačných techník 17

3 Metódy segmentácie medicínskych dát **18**
3.1 3D detekcia hrán 18
3.2 Morfologické techniky 18
3.2.1 Erózia 19
3.2.2 Dilatácia 20
3.3 Algoritmy prehľadávania grafov 21
3.4 Deformovateľné modely 22
3.4.1 Aktívne kontúry 22
3.5 Izoplochy a Level-set segmentácia 23
3.6 Metódy segmentácie založené na prahovaní 24
3.7 Metódy segmentácie založené na klasifikácii 25
3.8 Zhlukové algoritmy 26
3.8.1 K-means algoritmus 26
3.8.2 Fuzzy C-means algoritmus 27
3.9 Markov random field (MRF) 27
3.10 Narastanie oblastí 28
3.11 Split and merge 29
3.12 Segmentácia založená na registrácii 30
3.13 Umelá inteligencia 30

4 Realizácia operátorov **31**
4.1 RapidMiner 31
4.2 ImageJ 31
4.3 3D rotácia dát 31
4.4 Gaussovo rozostrenie obrazu 34

4.4.1 Operátor 3D rozostrenia dát 35
4.5 Priemerovací filter 37
4.5.1 Operátor 3D priemerovania dát 37
4.6 Sobelov filter 38
4.6.1 Operátor 3D detekie hrán 39
4.7 Mediánový filter 39
4.7.1 Mediánový operátor v 3D 41
4.8 Operátor zápisu 3D dát 41
4.9 Proces vytvorenia predspracovaných snímkov a ich uloženie 41

5 Trénovateľná segmentácia **43**
5.1 Trénovacie a testovacie snímky 43
5.2 Predspracovanie 44
5.3 Výber vstupných bodov trénovateľnej segmentácie 47
5.4 Extrakcia príznakov 48
5.5 Trénovateľná segmentácia 48
5.5.1 Druhy použitých učiacich sa algoritmov 50
5.5.2 Rozhodovací strom 51
5.5.3 Algoritmus náhodný les 52
5.5.4 SVM 53
5.5.5 Bayesova klasifikácia 54
5.5.6 Lineárna regresia 55
5.6 Tvorba trénovacieho modelu a aplikácia na nezávislé dáta 56
5.7 Multilevel segmentácia 57

6 Výsledké modely a zhodnotenie presnosti výsledkov **58**
6.1 3D modely bunkových štruktúr 58
6.2 Presnosť navrhnutého prístupu segmentácie 58

7 Záver **62**

Literatúra **63**

Zoznam príloh **65**

A Príloha A **66**
A.1 Obsah CD 66

ZOZNAM OBRÁZKOV

3.1 Často používané štrukturálne elementy. 19
3.2 Ukážka morfologickej operácie. 20
3.3 Ukážka morfologickej operácie. 20
3.4 Deformovateľné modely. Vývoj z počiatočného tvaru až po výslednú kontúru. Prevzaté z [10]. 22
3.5 Príklad Level-set funkcie (vpravo) pre 2D krivku K. 24
3.6 Príklad k-NN klasifikácie. 25
3.7 Rozdelenie zhlukov podľa algoritmu k-means. Prevzaté z [11]. 27
3.8 Narastanie oblastí. Obrázok vľavo značí začiatok algoritmu a obrázok v pravo proces narastania po pár iteráciách. 29
3.9 Dekompozícia quad-tree reprezentácie dát. 29
4.1 Proces 3D rotácie a rozostrenia dát zo vstupných obrázkov. 31
4.2 Kompozícia stacku 3D obrazu. 32
4.3 Pôvodný rez a zrotovaný stack. 34
4.4 3D Blur. 35
4.5 3D rozostrenie v smere xy, yz a xz 36
4.6 Average 3D. 38
4.7 Sobel 3D. 39
4.8 Triedenie pixelov v mediánovom filtri. 40
4.9 MedianFilter 3D. 41
4.10 Proces navrhnutej segmentácie 42
5.1 Testovacie dáta. 44
5.2 Obrázok vľavo originálny a vpravo rozostrený 2D blurom. 45
5.3 Obrázok vľavo originálny a vpravo rozostrený 3D blurom. 45
5.4 2D a 3D blur + detekcia hrán. 46
5.5 Získanie kostry z 2D a 3D rozostreného snímku. 47
5.6 Proces výberu bodov pre vstup trénovateľnej segmentácie. 47
5.7 Výber bodov pre trénovateľnú segmentáciu. 48
5.8 Lokálne príznaky. 49
5.9 Princíp tvorby tabuľky lokálnych príznakov. 50
5.10 Výsledný model rozhodovacieho stromu. 52
5.11 Výsledný model algoritmu Random Forest. 52
5.12 Výsledný model algoritmu SVM a typu jadra *dot*. 53
5.13 Výsledný model algoritmu SVM a typu jadra *anova*. 54
5.14 Výsledný model algoritmu Bayesovej klasifikácie. 55
5.15 Výsledný model algoritmu lineárnej regresie. 55
5.16 Výsledné štruktúry po aplikácii segmentačného modelu. 56

5.17 Výsledné označenie 3D štruktúr. 57
6.1 Výsledné 3D modely. 58

ZOZNAM TABULIEK

6.1 Presnosť jednotlivých segmentačných metód 60
6.2 Výsledky tímov realizujúcich multilevel segmentáciu 60

ÚVOD

Táto práca sa zaoberá problematikou segmentácie trojrozmerných obrazových dát. Segmentácia obrazových dát, hlavne tých medicínskych, má v súčasnosti veľký význam, hlavne, čo sa týka lepšej detekcie narušených tkanív, vhodnejšie zvolenej metódy liečby alebo pri navádzaní chirurga pri zložitej operácii.

V tejto práci sú prednesené súčasné techniky spracovania medicínskych dát. Ďalej je vysvetlený princíp fungovania segmentačných techník, ktoré sa vhodných spôsobom kombinujú za účelom čo najlepšieho výsledku spracovania obrazových dát pre daný problém. Ďalej je v tejto práci predstavený vlastný prínos ku extrakcií príznakov prostredníctvom vytvoreného obrazového filtru, ktorý umožňuje vstupný trojrozmerný snímok rotovať všetkými štyrmi smermi a následne snímok rozostriť prostredníctvom Gaussovej funkcie. Vyvorený filter je implementovaný v prostredí knižnice ImageJ a aplikovaný prostredníctvom operátora v programe RapidMiner, ktorý je jeden z najpoužívanejších programov pre dolovanie znalostí z dát. Po zapojenení operátora do segmentačneho procesu bude schopný extrahovať príznaky, ktoré sa využijú pri ďalšej analýze a spracovaní obrazových dát.

Hlavný prínos tejto práce spočíva vo vytvorení algoritmov pre extrakciu príznakov z 3D dát a následné overenie výsledkov pre problematiku mikroskopických 3D dát mozgu a vo vytvorení obrazových filtrov, ktoré sú vytvorené vo forme operátorov pre program RapidMiner. Boli vytvorené filtre rozostrovania obrazu pomocou Gaussovej funkcie, priemerovania hodnôt a filtre detekcie hrán. Tieto operátory využívajú vlastnosti priemerovania hodnôt, Gaussovej funkcie, superpozície a samotnej konvolúcie, ktorý je nezbytným operátorom pri aplikovaní filtru na vstupný obrázok. Sú využité znalosti z oblasti spracovávania obrazových signálov a ich reprezentácií a následnej tranformácie. Hlavnými prvkami aplikácie trojrozmerných filtov bolo uplatnenie superpozície pri konvolúcii jadra s transformačnými maticami pre každý smer x, y, z a spôsobe transformácie celého stacku snímku pomocou nahradzovania voxelov. Samotné operátory sa líšia v použití matematických operácií, a pričom princíp rotácie ostáva rovnaký.

Ďalším doležitým prínosom bolo vytvorenie vytvorenie segmentačného procesu, v ktorom sa využili dáta získané práve prostredníctvom operátorov vytvorených v programovacom jazyku Java. Bol teda vytvorený automatický segmentačný proces, ktorého výsledkom bolo vytvorenie modelu segmentácie, ktorý je aplikovatelný na nezávislé dáta. Účelom vytvorenia takéhoto modelu je práve odstránenie časovej náročnosti práce špecializovaného odborníka, ktorý musí jednotlivé segmenty určovať ručne.

Celý text práce je rozdelený nasledovne. Prvá kapitola obsahuje popis získavania obrazových dát pomocou rôznych zobrazovacích metód na získanie medicín-

skych snímkov. Tieto snímky sú použité na ďalšie spracovavanie obrazu a jeho analýzu. V ďalšej kapitole je rozobraný postup pri segmetácii týchto dát a rozdelenie metód podľa kritérií. Nasledujúca kapitola sa už podrobne venuje stručným popisom rôznych segmentačných techník na spracovanie obrazových dát, ktoré sa potom vhodným spôsobom kombinujú za účelom získania čo najlepšieho výsledku analýzy obrazových dát. Ďalšia kapitola sa už venuje vytvoreniu operátorov obrazového spracovania dát, v ktorej sa popisuje ich tvorba a je ukázaná fuknčnosť na konkrétnom príklade. Poslednou kapitolou je vytvorenie segmentačného procesu, ktorý aplikuje teóriu segmentácie a obrazového spracovania medicínskych dát.

Záver práce je venovaný zhodnoteniu jednotlivých segmentačných techník a ich porovnaním úspešnosti segmentácie bunkových štruktúr, kde je v tejto časti je ukázaný výsledný 3D model segmentovaných štruktúr mozgu.

1 MEDICÍNSKE SNÍMKY

Medicínske snímky sú akýmsi prostriedkom, ktorý zachytáva anatómiu a fyziológiu ľudského tela. Ide o jeho štruktúru, funkcie aj chemické zloženie. V súčasnej dobe sa získavajú snímky a veľké množstvá dát za účelom presnejšej diagnostiky, cielenejšej a účelnejšej liečby a pri tvorbe plánu chirurgického zákroku, ako aj pri prevencii najrozmanitejších ochorení. Tieto dáta sa získavajú z makroskopického hľadiska ako sú orgánové a tkanivové štruktúry a mikroskopického hľadiska, napríklad z pohľadu buniek a molekúl.

1.1 Technológie tvorby a zobrazenia medicínkych dát

Tvorba a zobrazovanie medicínskych snímok, z anglického **Medical Imagining**, je teda technika a postup, ktorý nám slúži na zobrazenie časti ľudského tela a jeho funkcií pre lekárske účely. Získavanie snímkov je založené na fyzikálnych vlastnostiach ľudského tela, najmä jeho priepustnosti, nepriehľadnosti, emisivite, odrazivosti, vodivosti, magnetizácii a zmene týchto parametrov v čase. Tkanivá sú väčšinou nepriepustné pre žiarenie so strednými vlnovými dĺžkami, ako je ultrafialové, infračervené a mikrovlnné vlnenie. Pomerne dobre však prepúšťajú krátkovlnnú radiáciu (röntgenové žiarenie), ktorá reaguje s elektrónmi v atómoch, z ktorých sú tkanivá zložené.

Jednoduchšie prípady medicínskych snímok sú šedo tónové snímky a zložitejšie sú RGB snímky. Pri rôznych druhoch týchto snímok sa potom odlišuje spôsob ich spracovania. V dnešnej dobe sú medicínske snímky zachytávané digitálne, pričom môžeme obraz zachytiť dvojrozmerný (2-D), kde sa skladá z pixelov, alebo trojrozmerný (3-D), kde je obrazová jednotka voxel. Voxel má vlastne plochu jedného pixelu a hĺbku rovnú hrúbke rezu [1].

Medicínsky snímok teda vzniká zachytávaním žiarenia, ktoré je vyžarované, šírené alebo odrážané tkanivami ľudského tela. Podľa druhu žiarení rozlišujeme rôzne typy medicínskych zobrazovacích techník. Ide napríklad o röntgen (RTG), počítačovú tomografiu (CT) využívajúcu vlny X, v nukleárnej medicíne ide o pozitrónovú emisnú tomografiu (PET) a jednofotónovú emisnú počítačovú tomografiu (SPECT), ktorá využíva rádiofarmaká a gama žiarenie. K technikám, ktoré pracujú s neionizujúcim žiarením patrí ultrasonografia, ktorá využíva mechanické vlnenie a magnetická rezonancia, ktorá využíva magnetické pole a neionizujúce žiarenie v rádiofrekvenčnej časti spektra.

1.1.1 RTG

Röntgenové žiarenie je forma elektromagnetického žiarenia. Ide o formu ionizujúceho žiarenia, ktoré je samo o sebe škodlivé. Röntgenové žiarenie môže byť využité pre zobrazovanie detailov kostí, zubov, poprípade za pomoci vhodných techník aj k skúmaniu mäkkých tkanív (tomografia).

Rádiológia je špecializovaný odbor lekárstva, kde sa využíva röntgenové žiarenie a toto zobrazovanie je jeho najčastejšie využitie.

1.1.2 X-ray CT

Počítačová tomografia (angl. Computed (axial) Tomography, skr. CT) je rádiologická vyšetrovacia metóda, ktorá pomocou röntgenového žiarenia umožňuje zobrazenie vnútra ľudského tela. Metóda sa využíva najmä v oblasti medicíny, kde slúži na diagnostiku širokého spektra poranení a chorôb.

Základným princípom metódy je zber a počítačové spracovanie veľkého množstva údajov o hodnote absorpcie röntgenového žiarenia. Tie sú získavané tak, že zdroj žiarenia - röntgenka a oproti nej nachádzajúce sa detektory rotujú okolo predmetu záujmu (napríklad pacienta). Následné počítačové spracovanie je potom schopné priradiť v danom reze každému bodu v priestore jeho príslušnú hodnotu absorpcie a zobraziť ju v stupňoch šedi na monitore.

Existuje niekoľko druhov počítačovej tomografie, ale táto téma nie je predmetom práce, je možné vyhľadať si doplňujúce informácie v odbornej literatúre.

1.1.3 PET

Pozitrónová emisná tomografia (PET) je lekárska zobrazovacia metóda, ktorá spadá do odboru nukleárnej medicíny. Princípom je lokalizácia miesta vzniku fotónov, ktoré v tele vznikajú pri anihiliácii pozitrónov - proces stretnutia častice s antičasticou, uvoľnených podanou rádioaktívnou látkou (rádiofarmakom) a elektrónov. PET nezobrazuje ani tak anatomickú štruktúru, ale skôr ochotu konkrétneho tkaniva vychytávať príslušné rádiofarmakum.

1.1.4 MRI

Zobrazovanie magnetickou rezonanciou (skr. MR alebo MRI - z anglického "magnetic resonance imaging") je moderná rádiologická metóda, ktorá umožňuje diagnostikovať patologické zmeny v ľudskom organizme bez toho, aby bolo nutné narušiť jeho integritu (operáciou či inou invazívnou metódou). Pacient je vložený do veľmi silného a homogénneho magnetického poľa, do jeho tela je vyslaný krátky rádiofrekvenčný

impulz a po jeho skončení sa sníma slabý signál, ktorý vytvára pacientovo telo, a ktorý sa následne použije na rekonštrukciu samotného obrazu.

Pri získavaní MRI snímku dochádza k modulácii (váhovaniu) snímku nasledujúcimi parametrami:

- hustotou protónových jadier (PD-weighted image)
- relaxačnou dobou T1, vznik T1-váhovaného obrazu (T1-weighted image)
- relaxačnou dobou T2, vznik T2-váhovaného obrazu (T2-weighted image) [5]

Pomocou MRI je teda možné získať rezy určitých častí tela a tie ďalej spracovávať. Tomuto ďalšiemu spracovávaniu sa budeme venovať v tejto práci, ktorej cieľom bude zo získaných snímok MRI segmentovať určité časti za účelom diagnózy ochorení. Procesu segmentácie sa budeme venovať neskôr v ďalších častiach práce.

Následné rezy, ktoré získame, je možno spájať do výsledného 3D modelu - toto je v súčasnej dobe najviac študovaným problémom modernej medicíny.

Magnetická rezonancia prináša určité výhody aj nevýhody. Medzi výhody patrí väčšia presnosť pri zobrazení väčšiny orgánov oproti ostatným zobrazovacím technikám. Dôsledkom väčšej presnosti je rozdielna intenzita signálov odlišných mäkkých tkanív. Nervy a mozgové tkanivá bolo možné neinvazívne zobrazovať až práve pomocou MRI. Vďaka rozsahu nastavenia vyšetrenia je možné dosiahnuť rozlíšenie, ktoré ďaleko presahuje možnosti röntgenu alebo CT.

Nevýhodou môže byť problém u pacientov s kardiostimulátormi, ktoré sa počas vyšetrenia môžu poškodiť. V porovnaní s CT sa artefakty vyskytujú častejšie a znižujú kvalitu výsledného obrazu. Nedajú sa teda vyšetrovať pohybujúce sa časti tela , okrem špecializovaných zariadení, ktoré sú určené na tento problém.

2 OBRAZOVÉ SPRACOVANIE MEDICÍNSKYCH SNÍMKOV

Rýchly pokrok v oblasti obrazového spracovania medicínskych snímok spôsobuje prevrat v medicíne. Počítačová tomografia (CT), magnetická rezonancia (MRI) a ostatné spôsoby poskytujú efektívny význam neinvazívneho mapovania anatómie rôznorodého objektu. Toto umožňuje vedcom a lekárom virtuálne pracovať s anatomickými štruktúrami a získavať informácie, ktorých výsledkom môže byť záchrana ľudského života.

V súčasnej dobe nie je rola medicínskeho zobrazovania obmedzená iba na jednoduché zobrazovanie a skúmanie anatomických štruktúr, ale presahuje sa tým, že poskytuje informácie o pacientovej diagnóze, pokročilom plánovaní a simulácii operačných zákrokov, plánovaní rádioterapie a podobne. Moderné vizualizačné techniky poskytujú vysokú presnosť a vysoko kvalitné 3D zobrazenie anatomických štruktúr. V mnohých prípadoch sa tieto vizualizačné techniky obmedzujú len na prosté zobrazovacie techniky a pokročilá analýza je zanedbávaná.

Hlavným dôvodov je komplexná vnútorná štruktúra ľudského tela s rozsiahlym množstvom anatomických orgánov zviazaných spolu, a to bráni lekárom pohľad viac ako jedným spôsobom. Niektoré vizualizačné triky, ako napríklad vytvorenie transparentného objektu v mnohých prípadoch nefunguje. Riešením tohoto problému je, že, anatomická štruktúra alebo región záujmu (ROI- region of interest) potrebuje byť vymedzená a oddelená, aby mohla byť zobrazená samostatne. Táto technika sa nazýva **obrazová segmentácia**. Keďže segmentácia ľudských orgánov alebo oblastí záujmu z jedného obrázku je významovo nepoužiteľná pre zväzkové zobrazovanie, zameriame sa teda na segmentáciu z 3D zväzkov (čo sú v podstate po sebe nasledujúce snímky zložené dohromady). Táto technika sa teda nazýva **zväzková segmentácia** [4].

2.1 Segmentácia medicínskych snímkov

Segmentácia v medicínskom zobrazovaní je všeobecne považovaná za veľmi zložitý problém. Zložitou je hlavne vďaka samotnej veľkosti súboru dát spojených so zložitosťou a premenlivosťou anatomických orgánov. Táto situácia je ešte zhoršená nedostatkami zobrazovacích techník, akými sú vzorkovacie artefakty, šum, nízky kontrast a podobne. Toto môže mať za následok, že hranice anatomických štruktúr môžu byť nezreteľné a prerušené. Hlavnou výzvou segmentačných algoritmov je teda presne extrahovať hranice určitého orgánu alebo ROI (Region of interest) a rozlíšiť ich od ostatných súborov dát.

Existuje mnoho prístupov k segmentácii. Tieto prístupy závisia od špecifickej aplikácie, zobrazovacej techniky (CT, MRI) a od ostatných faktorov. Napríklad, segmentácia pľúc má iné problémy ako segmentácia hrubého čreva. Ten istý segmentačný algoritmus, ktorý poskytuje vynikajúce výsledky pre jeden druh aplikácie, nemusí vôbec fungovať na iný prípad. Okrem toho, niektoré hlavné obrazové artefakty, ako napríklad šum a voxely tvorené viacerými druhmi tkanív (PVE - partial volume effect), majú významný vplyv na segmentačný algoritmus. Napríklad, segmentačný algoritmus môže byť veľmi odolný voči šumu, ale zároveň nepostačujúci na odstránenie prítomnosti artefaktu PVE. Táto premenlivosť je dôsledkom toho, že segmentácia je veľmi vyzývajúci problém. V súčasnosti neexistuje segmentačná metóda, ktorá by priniesla prijateľné výsledky pre každý súbor medicínskych dát. Existujú metódy, ktoré sú zovšeobecnené a môžu byť použité na rôzne druhy dát ale metódy, ktoré boli vytvorené špeciálne pre dané problémy, budú vždy generovať lepšie výsledky. Prístup k danej segmentácii môže občas pozostávať z viac ako jedného segmentačného algoritmu.

Z toho vyplýva, že výber správneho algoritmu alebo prístupu k segmentácii je dosť obsiahlym problémom a záleží na tom, čo chceme segmentáciou dosiahnuť.

2.2 Rozdelenie segmentačných techník

Počet segmentačných algoritmov, ktoré sa objavujú v rôznych literatúrach je veľký. Vďaka podstate problému segmentácie, väčšina tých algoritmov je špecifická práve vo vzťahu ku konkrétnemu problému, teda majú malý význam pre ostatné problémy. V tejto práci budú zovšeobecnené a pokryté základné techniky segmentácie, ktoré sa používajú a sú neustále vylepšované. Okrem toho sa budeme venovať iba 3D snímkam, teda budeme prezentovať algoritmy, ktoré sa aplikujú na 3D zväzky obrazových dát. Metódy segmentácie môžeme rozdeľovať podľa rôznych kritérií. Môžeme ich deliť podľa toho, či používajú binárnu alebo fuzzy logiku. V prípade binárnej logiky je rozhodujúce, či voxel patrí do určitej tkaninovej oblasti. V prípade, že je voxel tvorený viacerými ako jedným typom tkaniva (PVE), výsledky sú nepresné. Tento problém potom odstraňuje fuzzy segmentácia, kde je každému voxelu priradená množina pravdepodobnosti výskytu jednotlivých tkanív. Ďalej ich môžeme rozdeľovať na techniky hranovej detekcie, rôzne formy morfologických techník, algoritmy prehľadávania grafov, deformovateľné modely, izolplochy (Level sets), rôzne metódy prahovania, metódy založené na klasifikácii a štatistike, metódy zhlukovania a rôzne druhy hybridných metód.

3 METÓDY SEGMENTÁCIE MEDICÍNSKYCH DÁT

3.1 3D detekcia hrán

3D detekcia je jedna zo základných segmentačných techník. V rámci 3D reprezentácie je vhodnejšie hovoriť o povrchu ako o hrane. Teda tieto povrchy predstavujú hranice medzi jednotlivými tkanivami alebo anatomickými štruktúrami. Pri detekcii hrán sa využíva prvá alebo druhá derivácia obrazovej funkcie. Prvou deriváciu sa aproximuje Sobelov operátor, operátor Prewittovej, Robertov alebo Kirschov operátor. Tieto operátory určujú hrany ako maximá prvej derivácie. Pri druhej derivácii je výskyt hrany detekovaný, ak je priestorová zmena v polarite druhej derivácie dostatočne významná.

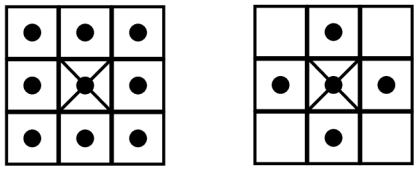
Techniky detekcie hrán v 3D priestore môžeme rozdeliť do dvoch častí:

- Lokálne hrany sú detekované podľa určitého rozdielu
- Tieto lokálne hrany sú zoskupované, aby tvorili určité okrajové obrysy uvažovaného regiónu od jedného voxelu k druhému.

Výhodou hranových detektorov je, že sa veľmi dobre aplikujú na súbory dát, ktoré majú dobrý kontrast medzi oblasťami. Tieto hrany sú potom detekované perfektne, čo sa dá overiť aj vizuálne. Na druhej strane, tieto algoritmy detekujú všetky hrany a následne je veľmi obtiažne nájsť vzájomný vzťah medzi hranami a oblasťami záujmu. Navyše sú tie algoritmy veľmi ťažko aplikovateľné na súbory dát, ktoré majú nízky kontrast medzi oblasťami. Tieto algoritmy sú tiež náchylné na šum a vo väčšine prípadov nie sú používané pre samotnú segmentáciu, ale sú združené do procesu s ostatnými segmentačným algoritmami, ktoré potom v tomto celku riešia konkrétny segmentačný problém. [4]

3.2 Morfologické techniky

Medzi morfologické techniky spracovávania obrazu patrí matematická morfológia (MM). MM je technika spracovávania geometrických štruktúr pôvodne založená na teórii množín. MM teda používa určité transformácie, ktoré slúžia na analýzu obrazu. MM získava vplyv určitého tvaru na celý obraz prostredníctvom štruktúrneho elementu (SE - structural element) 3.1. Tento element kóduje informáciu tohto primitívneho tvaru.



Obr. 3.1: Často používané štrukturálne elementy.

Tento tvar je určený ako súbor vektorov, ktoré odkazujú na určitý bod, väčšinou stred. Morfologickú transformáciu, teda priebeh morfologickej operácie, si môžeme predstaviť tak, ako by sme pohybovali štruktúrnym elementom (SE) systematicky po celom obraze. Bod obrazu, ktorý sa zhoduje s počiatkom súradníc SE, nazývame okamžitý bod. Výsledok vzťahu medzi obrazom a SE zapíšeme do tohto okamžitého bodu. Transformovaný obrázok je teda funkcia distribúcie SE v celom obraze. Základy morfologických operácií sú postavené na definovaní bodovej množiny E. Rozdeľujeme dva typy bodových množín a to binárny obraz a obraz s viacerými úrovňami jasu.

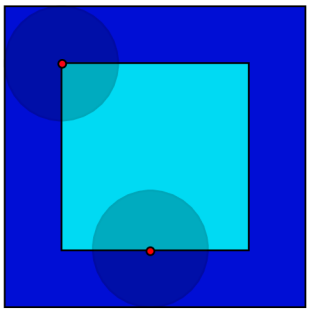
V MM existuje viacero morfologických operácií, ale medzi najzákladnejšie patria erózia a dilatácia.

3.2.1 Erózia

Erózia binárneho obrazu s vhodnou maskou, teda SE, je definovaná ako prienik množiny obrazu A s množinou všetkých posunov B (SE). Erózia je definovaná vzťahom:

$$A \ominus B = \bigcap_{b \in B} A_{-b} \tag{3.1}$$

Na obrázku 3.2 je vidieť príklad erózie tmavomodrého štvorca a disku, čoho výsledkom je svetlomodrý menší štvorec.



Obr. 3.2: Ukážka morfologickej operácie.

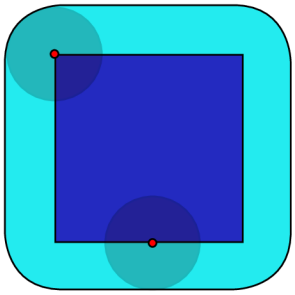
Erózia prevádza nezvratné operácie v obraze, a to zmenšovanie objektov, vyhladzovanie hrán, odstraňovanie izolovaných malých objektov a rozdeľovanie nežiadúcich prepojení. [2]

3.2.2 Dilatácia

Dilatácia binárneho obrazu je definovaná ako zjednotenie množiny obrazu A s množinou všetkých posunov SE.

$$A \oplus B = \bigcup_{b \in B} A_b \tag{3.2}$$

Na nasledujúcom obrázku možno vidieť príklad dilatácie tmavomodrého štvorca s diskom, kde je výsledkom vonkajší svetlomodrý štvorec.



Obr. 3.3: Ukážka morfologickej operácie.

Morfologické operácie sú pomerme jednoduché na pochopenie a implementáciu, ale zároveň sú v podstate zložité na kontrolu. Napríklad, je obtiažne kontrolovať dilatáciu, ak je daná jej horná hranica počtu operácií. Teda tieto algoritmy vyžadujú

na ich kontrolu v podstate určité externé kritériá. Tieto operácie majú navyše riziko, že zmenia morfológiu vstupného súboru dát. [2]

Týmto algoritmom by sme sa mali vyhýbať pokiaľ je našim hlavným záujmom je presnosť, existuje totiž riziko straty dôležitých dát. Algoritmy detekcie hrán, ako aj morfologické algoritmy nie sú sami o sebe používané na segmentáciu, ale ako súčasť segmentačného procesu.

3.3 Algoritmy prehľadávania grafov

V týchto technikách sú hrany a povrchy reprezentované grafmi a algoritmus sa snaží nájsť najkratšiu optimálnu cestu medzi uzlami grafu s použitím prehľadávacieho algoritmu A* , F*. Tie algoritmy sú špeciálne výhodné pri situáciách, keď určité časti medzi oblasťami v skúmanom segmentovanom zväzku nie sú korektne definované. F* (F star) algoritmus je často používaný v zobrazovaní biomedicínskych dát, preto bude stručne vysvetlený jeho princíp.

V princípe je F* algoritmus veľmi podobný A* algoritmu. F* algoritmus hľadá optimálnu cestu zo začiatočného bodu (z) do koncového bodu (k) s použitím určitého poľa (C), ktoré obsahuje všetky váhy a je iteratívne aktualizované poľom (P), ktoré obsahuje všetky cesty. Toto pole P je inicializované nekonečne okrem z bodu z, ktoré má priradené pole $C(z)$. Prvý krok v aktualizovaní pozostáva z pridávania všetkých prvkov y-tého riadku zľava doprava s využitím pravidla:

$$\begin{aligned} P(x,y) = \mathbf{min}\{&P(x-1,y-1)+C(x,y),\\ &P(c,y-1)+C(x,y),\\ &P(x+1,y-1)+C(x,y),\\ &P(x-1,y)+C(x,y),P(x,y)\} \end{aligned}$$

A následne pridávanie všetkých prvkov y-tého riadku sprava doľava s využitím pravidla:

$$P(x,y) = \mathbf{min}\{P(x+1,y)+C(x,y),P(x,y)\}$$

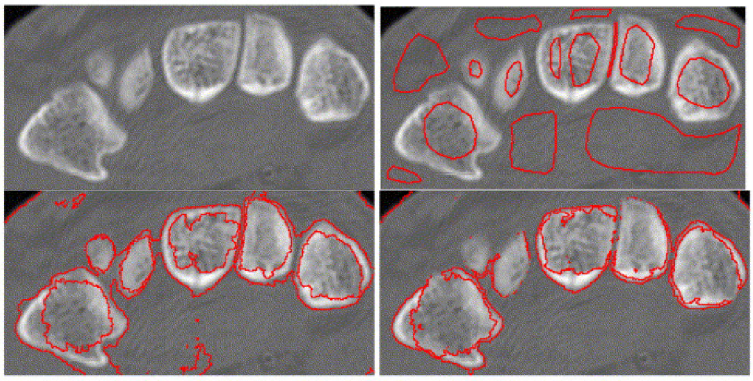
Keď sú všetky zmeny v poli P také, že nová hodnota je väčšia ako $P(k)$, algoritmus sa ukončí. Optimálna trasa sa potom dá nájsť ako cesta späť z koncového bodu k k minimálnej hodnote P každého susedného uzla, pokiaľ sa nedosiahne počiatočný bod z.

Hlavnou výhodou tejto metódy je, že je veľmi dobre aplikovateľná medzi oblasťami, ktoré sú prerušené. Zároveň sa vyžaduje, aby boli tieto povrchy reprezentované ako grafy, čo môže byť zložité. Ďalšia nevýhoda (z pohľadu zobrazenia dát z jednotlivých

zväzkov snímku) je, že algoritmus počíta s povrchmi. Aby sme dostali voxely ako reprezentáciu nejakého povrchu, je potrebný ďalší prístup na skonvertovanie povrchu do voxelov [6].

3.4 Deformovateľné modely

Pre algoritmy, ktoré využívajú deformovateľné modely, je charakteristický prístup *zhora-dole* , čím sa značne líšia od tradičných prístupov. Častejším postupom je teda extrahovať primitíva a potom sa snažiť ich zlepiť do vyšších celkov. Deformovateľné modely začínajú najskôr odhadom celého tvaru, ktorý následne porovnávajú a pod vplyvom fyzikálnych síl upravujú tak, aby lepšie odpovedali segmentovanému snímku 3.4. Princíp segmentácie môže byť, napríklad realizovaný tak, že deformovaný model popisuje uzavretou krivkou alebo plochou, ktorá oddeľuje segmentovanú oblasť vo vnútri od vonkajšej oblasti.



Obr. 3.4: Deformovateľné modely. Vývoj z počiatočného tvaru až po výslednú kontúru. Prevzaté z [10].

3.4.1 Aktívne kontúry

Aktívne kontúry, označované tiež ako hady, patria do skupiny deformovateľných modelov. Hady sú rovinné deformovateľné kontúry, ktoré sú významné pri niekoľkých úlohách z oblasti analýzy obrazu. Používajú formuláciu minimalizácie energie a úspechom tohto prístupu je nájdenie parametrického modelu, ktorý minimalizuje sumu vnútornej a potenciálnej energie. Vnútorná energia špecifikuje napnutie alebo hladkosť povrchu modelu. Potenciálna energia je definovaná cez celú oblasť zväzku a typicky obsahuje lokálne minimá na hranách, ktoré sa objavujú na hraničných oblastiach objektu. Minimalizácia celkovej energie prináša práve vnútornú a potenciálnu

energiu a ako výsledok sú tieto energie priťahované príznakmi snímku, akými sú čiary a hrany.

Deformovateľné modely majú tú výhodu, že ponúkajú koherentný a konzistentný matematický popis a sú odolné voči šumu a hraničným medzerám. Ďalšou výhodou je, že ponúkajú presnosť pre reprezentáciu hraníc, čo môže byť veľmi výhodné pre veľa aplikácií. [3]

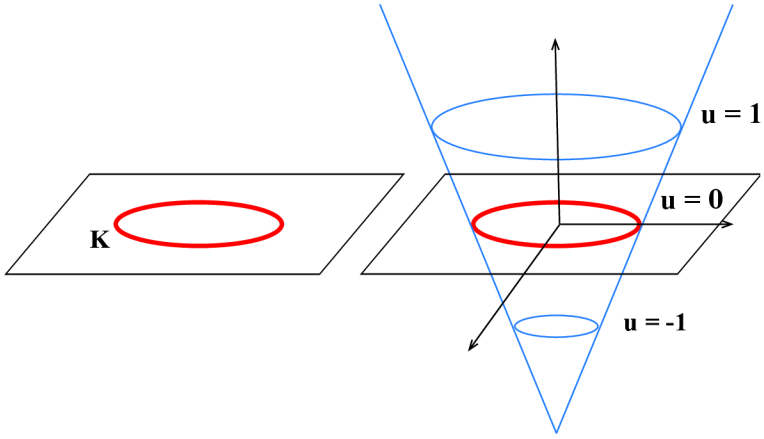
Nevýhodou je, že tieto metódy vyžadujú manuálny zásah kvôli inicializácii vstupných súborov dát. Tieto algoritmy tiež vyžadujú od užívateľa výber vhodných vstupných parametrov. Preto nie sú v rámci automatizácie techník veľmi vhodnou metódou.

3.5 Izoplochy a Level-set segmentácia

Izoplochy sú definované ako spojenie voxelov a ich intenzít rovnajúce sa izohodnote (tzv. *isovalue*) v 3D zväzku. Level-set bol prvýkrát predstavený Osherom a Sethianom v roku 1988 [7]. Táto technika izoplôch ako modelovacej technológie môže slúžiť ako alternatíva parametrickej metódy aktívnych kontúr popísaných v kapitole 3.4. Level-sets sú numerické techniky navrhnuté na sledovanie evolúcie rozhrania, teda v našom prípade izoplochy. Ostatné numerické techniky sa snažia sledovať pohybujúce sa hraničné oblasti tým, že na vyvíjajúci sa povrch vkladajú kolekciu značiek a následne menia ich pozíciu, aby sa s týmto povrchom zhodovali.

U Level-set metódy je krivka reprezentovaná nulovou hladinou rezu v rovine xy (tzv. *zero level set*). Táto funkcia sa nazýva *level set funkcia* a každému bodu roviny xy priraďuje jeho výšku u nad nulovou hladinou 3.5. Povrch funkcie sa postupne adaptuje vzhľadom k zadaným metrikám krivosti a obrazovým gradientom. Základným rozdielom level-set metódy oproti aktívnym kontúram je, že tvar krivky nemeníme priamo, ale prostredníctvom level-set funkcie.

Level-set je teda pokročilý algoritmus detekovnia hrán, ktorý pracuje v 3D. Tento algoritmus má za vstupné dáta objemové alebo diskrétne a výstup algoritmu je kontúra, poprípade povrchový model objektu. Tento algoritmus má svoje výhody a to, že metóda je vhodná pre veľmi zložité a komplexné tvary a umožňuje zaviesť podmienky krivosti a znalosti tvaru objektu. Na druhej strane má aj svoje nevýhody a to, že je nutná manuálna inicializácia.



Obr. 3.5: Príklad Level-set funkcie (vpravo) pre 2D krivku K.

3.6 Metódy segmentácie založené na prahovaní

Prahovanie je pravdepodobne najjednoduchšou segmentačnou technikou. V tejto technike je samostatná hodnota nazývaná *T-prah* použitá na vytvorenie binárneho rozdelenia intenzít voxelov. Všetky voxely s intenzitami väčšími ako prahovacia hodnota sú zoskupené do jednej triedy a ostatné voxely s intenzitou nižšou ako prahová hodnota opäť do druhej. Použitie jednej prahovacej hodnoty má teda za výsledok binárne segmentovaný zväzok.

Analýzu prevádzame pomocou histogramu, z ktorého porovnávame hodnoty s nastaveným prahom T. Prahovanie môžeme previesť podľa nasledujúcej funkcie:

$$R(x,y) = \begin{cases} 1, f(x,y) < T \\ 0, otherwise \end{cases} \tag{3.3}$$

Táto technika môže byť rozšírená na použitie viacerých prahovacích hodnôt, kde je oblasť definovaná dvoma prahovacími hodnotami, *spodnou* prahovacou hodnotou a *vrchnou* prahovacou hodnotou. Každý voxel zo vstupného zväzku následne náleží jednej z oblastí definovanej vlastnou intenzitou. Táto technika je známa ako *multitresholding*[8].

Napriek svojej jednoduchosti je táto technika veľmi efektívna v segmentácii zväzkov s veľmi dobrým kontrastom medzi oblasťami. Prahovanie je väčšinou prvý krok v segmentačnom procese zväzku snímkov.

Hlavným nedostatokom tejto techniky je, že výsledok je úzko spätý so zvoleným prahom. Každá zmena v hodnote zvoleného prahu má za následok rôzne osegmentované oblasti. Prahovacie hodnoty sú väčšinou zvolené interaktívne, podľa vizuálneho zhodnotenia dát. Ďalší nedostatok, ktorý vyplýva priamo z prvého je, že táto technika je veľmi citlivá na šum a nehomogenitu v hodnotách intenzít. Preto je veľmi obtiažne ju aplikovať na MRI a ultrazvukové snímky.

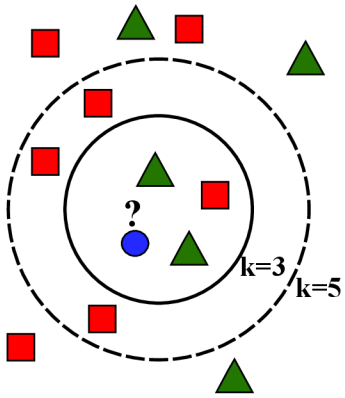
3.7 Metódy segmentácie založené na klasifikácii

Metódy založené na klasifikácii sú techniky založené na rozpoznávaní vzorov, ktoré prehľadávajú rozdelenú množinu príznakov odvodenú z dát konkrétneho zväzku snímok so známymi popismi. Množina príznakov je veľkosť N dimenziálneho vektoru príznakov vytvoreného pre každý voxel. Tento príznak môže obsahovať, napríklad informáciu o intenzite voxelu, gradientu voxelu alebo vzdialenosť voxelu od hraničnej oblasti zväzku [9].

Klasifikátory vyžadujú trénovacie dáta, ktoré sú predsegmentované buď manuálne alebo pomocou inej metódy. Predsegmentované dáta sú potom použité ako referenčné body, ktoré budú niesť nové dáta už použitej automatickej segmentačnej metódy.

Najjednoduchšia forma klasifikátorov sú klasifikátory najbližšieho suseda, kde je každý pixel alebo voxel klasifikovaný do tej istej triedy ako trénovacie dáta s blízkymi hodnotami intenzít. Klasifikátor *k-najližšieho-suseda (kNN- k-nearest-neighbor)* je zovšeobecnením tejto metódy, kde každý pixel je klasifikovaný podľa väčšiny k najbližších trénovacích dát. Podobný príklad klasifikácie je podľa Parzenovho okna, kde je klasifikácia vytvorená podľa trénovacích dát v Parzenovom okne, ktoré sa vytvorí nad klasifikovaným voxelom. Obe tieto klasifikácie sú neparametrické, pretože nerobia žiadny predpoklad o statickej štruktúre vstupných dát.

Ďalšími často používanými klasifikačnými technikami sú metóda maximálnej vierohodnosti *(ML-maximum likehood)* a Bayesovská klasifikácia.



Obr. 3.6: Príklad k-NN klasifikácie.

Na obrázku 3.6 môžeme vidieť, že napríklad pri zvolení vstupného vektora dát ako modrý krúžok, tak pri zvolení $k = 3$ bude tento vstupný vektor klasifikovaný ako zelený trojuholník a pri zvolení $k = 5$ bude klasifikovaný ako červený štvorec.

Klasifikátory vyžadujú, že štruktúra, ktorá má byť segmentovaná, musí vlastniť zreteľné a kvantitatívne schopné príznaky. Pretože trénovacie dáta môžu byť označo-

vané popismi, klasifikátory môžu preniesť tieto popisy na nové dáta, ak množina príznakov dostatočne odlišuje jednotlivé popisy. Tieto metódy sú relatívne výpočtovo výkonné a oproti prahovaniu môžu byť použité na viac kanálové obrazové dáta. Nevýhodou je, že nevykonávajú žiadnu priestorovú modeláciu. Ďalšia nevýhoda je manuálny zásah užívateľa pre získanie trénovacích dát.

3.8 Zhlukové algoritmy

Techniky založené na vytvorení zhlukov (clusters) dát používajú jednotlivé charakteristiky voxelov, pomocou ktorých ich zaraďujú do bezprostrednej blízkosti okolitých pixelov s podobnými charakteristikami. Zhlukovanie môže byť voľne preložené ako proces zoskupovania objektov do skupín, ktorých členovia vykazujú podobné vlastnosti. V našom prípade sú teda objekty dáta z voxelov a skupiny sú segmentované regióny. Pod pojmom podobné vlastnosti si môžeme predstaviť akékoľvek charakteristiky, ktoré vlastnia dáta získané z voxelov, teda napríklad intenzita voxelu, pozícia a vlastnosti okolia.

Segmentácia založená na zhlukovaní je podobná metódam založených na klasifikácii s výnimkou, že tieto metódy nepotrebujú žiadne trénovacie dáta. Tieto metódy spadajú teda pod označenie algoritmov učenia bez učiteľa. Môžeme povedať, že tieto algoritmy učia sami seba na dostupných dátach. Najviac používaným zhlukovým algoritmom pre segmentáciu je zhlukovanie **K-means**.

3.8.1 K-means algoritmus

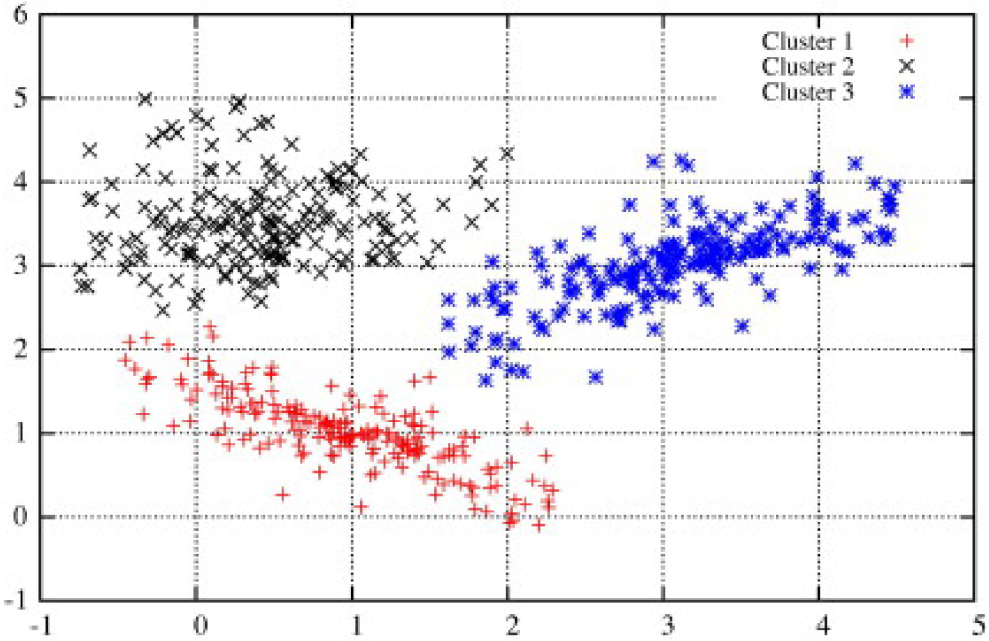
K-means algoritmus je teda neparametrická metóda analýzy zhlukov. Zhluky sú charakterizované centroidmi, teda stredmi zhlukov. Aj keď k-means algoritmus je automatickou metódou, je potrebné nastaviť počiatočné hodnoty centroidov, a to väčšinou ručne. Centroidy sú následne prepočítavané na základe voxelov do jednotlivých zhlukov. Voxely sú priraďované následne do zhlukov podľa najmenšej vzdialenosti od centroidov. Najčastejšie býva použitá Euklidovská vzdialenosť, ktorá je definovaná vzťahom:

$$d(a,b) = \sqrt[2]{\sum_{i=1}^{N} |x_a i - x_b i|^2}, \tag{3.4}$$

kde a a b sú objekty medzi ktorými vzdialenosť meriame. Vo všeobecnejšom prípade sú to vektory, ktoré obsahujú konečný počet prvkov (voxelov).

K-means algoritmus priraďuje každému voxelu práve jednu triedu, teda je binárny. Neberie do úvahy, že by mohlo dochádzať k prekrývaniu medzi jednotlivými tkanivami (PVE). Tento problém však odpadá pri použití iného neparametrického klasi-

fikátora, ktorý je založený tiež na učení bez učiteľa a je to algoritmus *Fuzzy C-means algoritmus*.



Obr. 3.7: Rozdelenie zhlukov podľa algoritmu k-means. Prevzaté z [11].

3.8.2 Fuzzy C-means algoritmus

Vo fuzzy zhlukovaní sa pre každý bod počíta, s akou pravdepodobnosťou patrí do ktorého zhluku. Takže body na kraji zhluku majú menší stupeň príslušnosti ako body v centre zhluku. Týmto spôsobom je popísané rozloženie objektov v zhlukoch. Voxely teda môžu patriť do viac zhlukov zároveň. Takto je možné identifikovať objekty, ktoré sa nedajú priradiť do žiadneho zhluku. Jednoduchšie je aj vytvorenie zhluku na základe pravdepodobnosti príslušnosti objektu k zhluku.

Fuzzy algoritmus môžeme jednoducho popísať tým, že najskôr sa vyberie počet zhlukov, a následne sa každému bodu priraďuje koeficient príslušnosti. Potom sa opakujú kroky, a to spočítanie stredu každého zhluku a pravdepodobnosti príslušnosti k určitému zhluku pre každý bod, až pokiaľ zmena koeficientu k príslušnosti nie je menšia ako zvolený prah citlivosti.

3.9 Markov random field (MRF)

Markovovo náhodné pole (angl. Markov random fields) samo o sebe nie je segmentačnou metódou, ale štatistickým modelom, ktorý používajú segmentačné techniky.

MRF modelujú priestorové väzby medzi susednými alebo blízkymi voxelmi. Tento lokálny vzájomný vzťah poskytuje mechanizmus pre modelovanie rôznorodých vlastností snímku [12]. V medicínskom zobrazovaní platí, že väčšina pixelov patrí do toho istého segmentu ako jeho susedia. Z tohto vyplýva, že každá anatomická štruktúra, ktorá sa skladá iba z jedného pixelu má veľmi malú pravdepodobnosť výskytu v obraze.

Metóda MRF je spolu s Bayesovským modelom často začlenená do algoritmov zhlukovej segmentácie, napr. ako K-means. Výsledkom segmentácie je potom maximalizácia aposteriórnej (na základe skúsenosti) pravdepodobnosti $p(C|X)$ (pravdepodobnosť, že vzorka X patrí do segmentu C) pre všetky body obrazu.

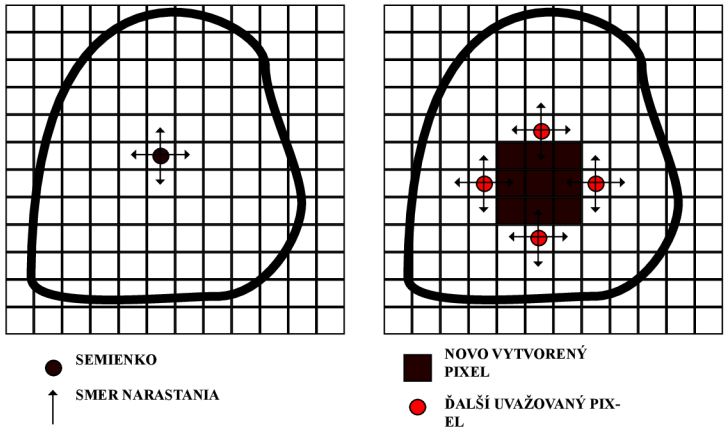
Nevýhodou MRF modelov je obtiažna voľba parametrov, ktoré ovplyvňujú silu priestorových väzieb medzi pixelmi obrazu. Príliš vysoké nastavenie parametrov môže spôsobiť, že hrany medzi segmentmi budú nadmerne vyhladené, a tým pádom dôjde ku strate dôležitých detailov o štruktúre. Okrem toho je táto metóda výpočtovo veľmi náročná. Navzdory týmto nevýhodám sú MRF modely veľmi rozšírené a používajú sa nielen na modelovanie štruktúry alebo textúry, ale aj na modelovanie nehomogenít v snímku a nerovnomerného osvetlenia.

3.10 Narastanie oblastí

V ďalšej časti sa budeme venovať hybridným algoritmom, ktoré kombinujú vlastnosti niektorých algoritmov popísaných vyššie. Metóda narastania oblasti je pravdepodobne asi najjednoduchšou technikou spomedzi hybridných metód. Táto technika vyberá predelenú oblasť z 3D zväzku na základe nejakého prednastaveného spojovacieho kritéria. Toto kritérium môže byť jednoducho intenzita voxelu alebo výstup iného segmentačného algoritmu [13]. Algoritmus narastania oblastí vyžaduje v podstate na začiatok *semienko* , od ktorého sa začína. Algoritmus potom narastá od počiatočného semienka, až pokiaľ sú splnené podmienky nastavenia spojovacieho kritéria.

Prahovanie ako aj metóda narastania oblastí nie je použitá ako samostatná segmentačná technika, ale ako súčasť segmentačného procesu pre konkrétny prístup k nejakému problému.

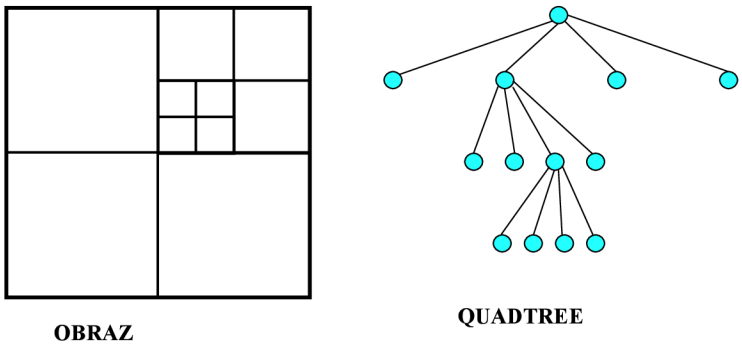
Hlavnou nevýhodou tohto algoritmu je manuálny zásah užívateľa pri inicializácii počiatočného bodu, teda semienka. Metóda narastania oblasti je taktiež citlivá na šum a PVE a tieto artefakty spôsobujú, že vybraný región obsahuje diery alebo nespojené časti.



Obr. 3.8: Narastanie oblastí. Obrázok vľavo značí začiatok algoritmu a obrázok v pravo proces narastania po pár iteráciách.

3.11 Split and merge

Táto metóda nazývaná delenie a spojovanie oblastí (angl. Split and Merge) je podobná metóde narastania oblastí. Je založená na quad-tree reprezentácii dát. Tento algoritmus vyžaduje, aby vstupné dáta boli organizované do pyramídovej mriežkovej štruktúry.



Obr. 3.9: Dekompozícia quad-tree reprezentácie dát.

Pre 3D dáta sa tieto regióny delia na osem častí. Teda pyramídová mriežková štruktúra sa delí na regióny a tieto regióny obsahujú skupinu po osem častí. Každý tento región potom môže byť rozdelený do ôsmich menších regiónov, alebo môže byť spojený do jedného väčšieho regiónu. Kritérium pre spojovanie v tejto metóde môže čokoľvek. Môže to byť intenzita pixelu alebo overovanie nejakej podmienky, založenej na výstupe predchádzajúcej segmentácie.

Definujme si parameter C, ktorý definuje kritérium spojovania oblasti. Algoritmus potom môžeme zapísať ako:

- Vyber regiónov R v mriežkovej štruktúre. Ak $C(R)$ je $false$, rozdeľ región na osem subregiónov. Ak pre osem regiónov $R_1, R_2, ..., R_8, C(R_1 \cup R_2 \cup ... \cup R_8) = true$, spoj ich do jedného regiónu. Skonči, keď žiadne regióny nie sú spojené.
- Ak susedné regióny R_i a R_j spĺňajú podmienku $C(R_i \cup R_j) = true$, tak potom spoj tieto regióny

Veľkou výhodou oproti metóde narastania oblastí je, že nie sú potrebné žiadne semienkové body, teda žiadna manuálna interakcia. Na druhej strane je ale potrebné, aby vstupné dáta boli vyjadrené pomocou pyramídovej štruktúry, čo môže byť nevhodné pre väčšinu súborov dát, ktoré sa používajú dnes.

3.12 Segmentácia založená na registrácii

Princípom týchto metód je registrácia vstupného obrazu na už segmentovaný obraz, tzv. *atlas*. Po prevedení registrácie obrazu na šablónu atlasu sa prevádza inverzná transformácia, pri ktorej sú anatomické alebo tkanivové popisy namapované z atlasu na vstupný obraz, čím je získaný segmentovaný obraz [14].

Segmentácia založená na registrácii je využívaná hlavne na MRI snímkach hlavy. Výhodou tejto metódy je, že popisy dát sú prevedené ako aj segmentácia. Hlavný nedostatok tejto metódy je kvôli anatomickej premenlivosti.

3.13 Umelá inteligencia

Konvenčné segmentačné algoritmy založené na štruktúrnych znalostiach často vyžadujú značnú užívateľskú odbornosť. Umelé neurónové siete (angl. Artificial Neural Networks-ANN) sa čiastočne snažia prekonať tieto nevýhody. ANN sú masívne paralelné siete spracovávajúce prvky alebo uzly, ktoré simulujú biologické učenie. Každý uzol v ANN je schopný vykonávať základné výpočty. Učenie je dosiahnuté prostredníctvom váh priradených k spojeniam medzi uzlami.

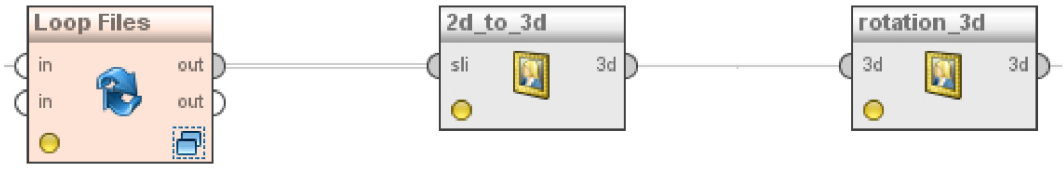
Existujú dve stratégie trénovania ANN. Prvý prístup hľadá charakteristické vlastnosti vstupných dát (väčšinou príznakové vektory) a klasifikuje ich do tried. Tento prístup nazývame *učenie bez učiteľa*. Sem môžu patriť rôzne prístupy trénovania dát, ako napríklad prostredníctvom *Kohonenových máp*.

Druhý prístup je *učenie s učiteľom*, ktoré vyžaduje ručne segmentované trénovacie dáta. Vstupom učiaceho sa algoritmu sú nielen príznakové vektory, ale aj funkcie, ktoré každému vstupnému vektoru priraďuje určitý segment snímku.

4 REALIZÁCIA OPERÁTOROV

4.1 RapidMiner

RapidMiner je mutli-platformový open-source software od spoločnosti Rapid-I. Je to veľmi užitočný nástroj, čo sa týka spracovania dát akýchkoľvek druhov. V tejto práci sa však budeme zaoberať iba spracovaním obrazových dát. RapidMiner poskytuje širokú škálu operátorov, od načítania dát zo súborov cez rôzne trénovacie modely až po samotný výstup, ktorý môžeme uložiť do súboru alebo vizuálne zobraziť pomocou grafov a tabuliek.



Obr. 4.1: Proces 3D rotácie a rozostrenia dát zo vstupných obrázkov.

4.2 ImageJ

ImageJ je program, ktorý sa využíva najviac na analýzu a spracovávanie biomedicínskych dát. Je šírený pod licenciou Public Domain a je vyvíjaný americkým National Institutes of Health (NIH). Obsahuje veľké množstvo algoritmov pre analýzu a spracovanie obrazu a je rozšíriteľný o mnoho operátorov, ktoré vyvíjajú autori po celom svete.

4.3 3D rotácia dát

Pri realizácii operátora sa táto časť skladá z dvoch častí, a to rotácie a následného rozostrenia zrotovaných dát. V tejto časti si popíšeme postup rotácie vstupného súboru dát, čo v našom prípade tvorí tzv. *stack* obrázkov. Stack môžeme chápať ako súbor viacerých obrázkov, ktoré nasledujú za sebou. Jednotlivé obrázky sa nazývajú *rezy* a ich zlúčením a zoskupením dostaneme výsledný 3D obraz.

Jednotlivé rezy stacku obrazu si teda predstavíme ako spojité rozloženie jasu v rovine, kde je toto rozloženie definované obrazovou funkciou $f(x, y)$. Táto funkcia f je spojitá s argumentmi x, y, ktoré definujú bod v rovine. Obor hodnôt, ktoré môže táto funkcia nadobúdať je definovaná vzťahom:

$$0 \leq f(x, y) \leq f_{max}, \tag{4.1}$$

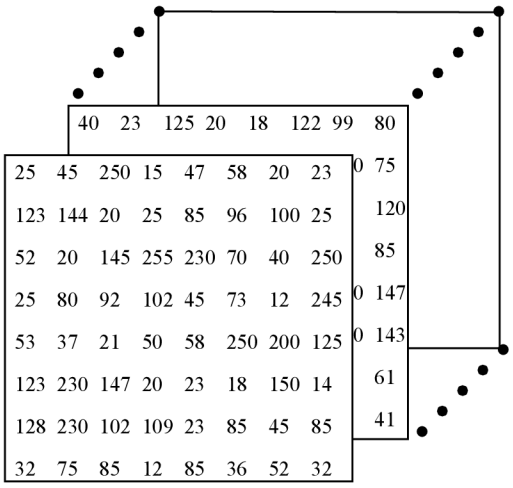
teda obrazová funkcia definuje obraz s premennou intenzitou jednej farby, teda obraz monochromatický. Pre viacfarebné modely sa dá obrazová funkcia vyjadriť ako súbor obrazových funkcií, kde jednotlivé funkcie reprezentujú obrazovú zložku daného farebného modelu. Aby sme mohli s obrazovou funkciu pracovať, potrebujeme ju zaradiť do N-rozmerného priestoru (systému), teda v našom prípade do 2D systému. Prepis signálu $f(x, y)$ pomocou zobrazenia σ na signál $g(x, y)$ môžeme zapísať ako:

$$g(x, y) = \sigma\{f(x, y)\}. \tag{4.2}$$

V spracovaní obrazových dát je ale vhodnejšie, aby hodnoty obrazovej funkcie f boli vyjadrené pomocou matice, ktorá obsahuje reálne hodnoty. V takom prípade už nepovažujeme obraz za spojitú funkciu, ale za množinu hodnôt, ktorú zapíšeme ako:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1, M-1) \end{bmatrix}$$

Na nasledujúcom obrázku 4.2 sú zobrazené rezy stacku, ktorej pixely sú definované obrazovou funkciou $f(x, y)$.



Obr. 4.2: Kompozícia stacku 3D obrazu.

Popísali sme teda kompozíciu rezu stacku obrázkov, ale pre náš prípad musíme túto funkciu rozšíriť o súradnicu z, ktorá nám udáva počet rezov v aktuálnom snímku. Uvažujeme teda o trojrozmernej funkcii:

$$f'(x,y,z) = \begin{bmatrix} f(0,0,z) & f(0,1,z) & \cdots & f(0,M-1,z) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0,z) & f(N-1,1,z) & \cdots & f(N-1,M-1,z) \end{bmatrix}$$

Rotácia v tomto prípade znamená, zamieňanie súradnic nasledovným spôsobom:

- Rotácia vľavo - zámena hodnôt súradnice x za súradnice z.
- Rotácia vpravo - princíp založený na rotácii vľavo, len v opačnom smere.
- Rotácia hore - zámena súradníc y za súradnice z.
- Rotácia dolu - princíp založený na rotácii hore, len v opačnom smere.

Po rotácii vľavo uvažujeme teda trojrozmernú funkciu s vymenenými osami.

$$f''(x,y,z) = \begin{bmatrix} f(0,M-1,0) & f(0,M-1,1) & \cdots & f(0,M-1,L-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,M-1,0) & f(N-1,M-1,1) & \cdots & f(N-1,M-1,L-1) \end{bmatrix}$$

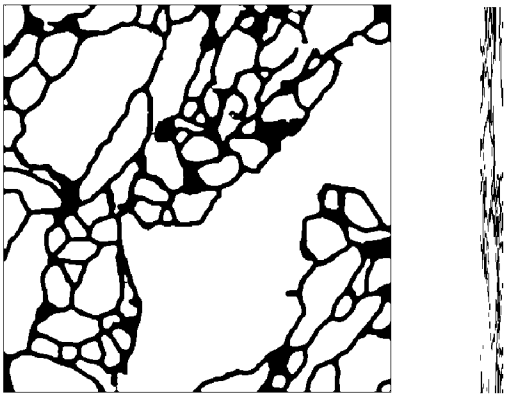
Pri rotácii vpravo je postup obdobný, len osi x a z sú vymenené, teda pri realizácii sa zrotovaný stack vľavo rotoval ešte dvakrát v smere osi x, aby bola dosiahnutá výmena osí v opačnom poradí. Využívali sa teda dve pamäte, v ktorých sa voxely medzi sebou nahradzovali.

Rotácia smerom hore je realizovaná obdobne ako rotácia vľavo s tým rozdielom, že v tomto prípade sa vymieňajú osi y a z. Opäť teda uvažujeme o trojrozmernej funkcii zapísanú v maticovej podobne ako:

$$f''(x,y,z) = \begin{bmatrix} f(N-1,0,0) & \cdots & f(N-1,M-1,0) \\ \vdots & \ddots & \vdots \\ f(N-1,M-1,L-1) & \cdots & f(N-1,M-1,L-1) \end{bmatrix}$$

Ďalej, proces rotácie smerom dole prebieha rovnako ako rotácia doprava, teda opäť sa rotuje ešte dvakrát v smere osi y a využívajú sa tiež dve pamäte, do ktorých sa zapisujú a prepisujú voxely.

Na nasledujúcom obrázku 4.3 je vidieť rotáciu stacku do ľavej strany, kde obrázok vľavo predstavuje jeden rez z celého stacku a obrázok vpravo stack už zrotovaný, teda osa x a z sa vymenili, teda tento konkrétny rez obsahuje v sebe všetky rezy pôvodného stacku.



Obr. 4.3: Pôvodný rez a zrotovaný stack.

4.4 Gaussovo rozostrenie obrazu

Po aplikovaní rotácie stacku na všetky smery môžeme následne začať dáta rozostrovať. Princíp Gaussového rozostrenia je v použití konvolúcie jadra, tzv.*kernel* s každým pixelom obrázku. Sila Gaussového rozostrenia spočíva v zvolení veľkosti jadra konvolučnej matice, kde je toto jadro vypočítané na základe Gaussovej funkcie:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{x^2}{2\sigma^2}} \tag{4.3}$$

V tomto prístupe využijeme princíp superpozície, kde výsledná konvolúcia je rovná konvolúciám s Gaussovým jadrom v smere x, y a z. Označme si teda konvolúciu ako K, potom princíp superpozície môžeme zapísať ako:

$$K = K_1 + K_2 + K_3 \ldots K_n = \sum_{i=1}^{n} K_i \tag{4.4}$$

V predloženom riešení ale využije indexovanie konvolúcie K len do troch, pretože máme len tri osi. Takže celková konvolúcia bude rovná:

$$K = K_x + K_y + K_z \tag{4.5}$$

Využívame teda Gausovu funkciu o jednej premennej pre každý smer. Spojitú konvolúciu vo všeobecnosti matematicky vyjadríme ako:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\alpha) g(x - \alpha) \mathrm{d}\alpha, \tag{4.6}$$

kde hodnota funkcie f s jadrom g v bode x je integrál súčinu funkcie f s otočenou funkciou konvolučného jadra posunutej do bodu x. V obrazovom spracovaní je teda funkciou $f(x)$ vstupný obrázok a funkciou $g(x)$ Gaussov filter.

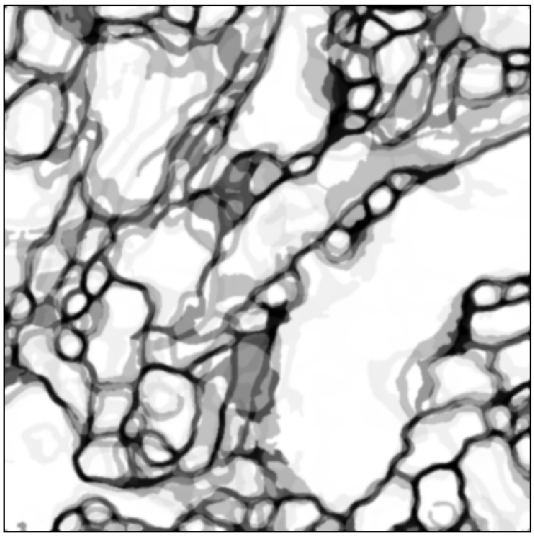
Pri obrazovom spracovaní využijeme ale konvolúciu diskrétnu, ktorá sa využíva v počítačovej grafike, a pre náš prípad to bude konvolúcia o jednej premennej, teda vyjadrenie konvolúcií pre osi x,y a z je nasledovný:

$$(f * g)(x) = \sum_{i=-k}^{k} f(x-i) \cdot h(i) \tag{4.7}$$

$$(f * g)(y) = \sum_{j=-k}^{k} f(y-j) \cdot h(j) \tag{4.8}$$

$$(f * g)(z) = \sum_{l=-k}^{k} f(z-l) \cdot h(l) \tag{4.9}$$

Výsledná aplikácia teórie spočíva v použití konvolúcie jadra Gaussovej funkcie pre smer x, y a následnej rotácie stacku o zvolený smer, kde sa prepíšu súradnice osí a aplikuje sa opäť konvolúcia v smere x alebo y (podľa smeru rotácie). Výsledkom je rozostrenie obrazu, ktoré obsahuje rozostrené pixely všetkých rezov stacku obrázkov 4.4.

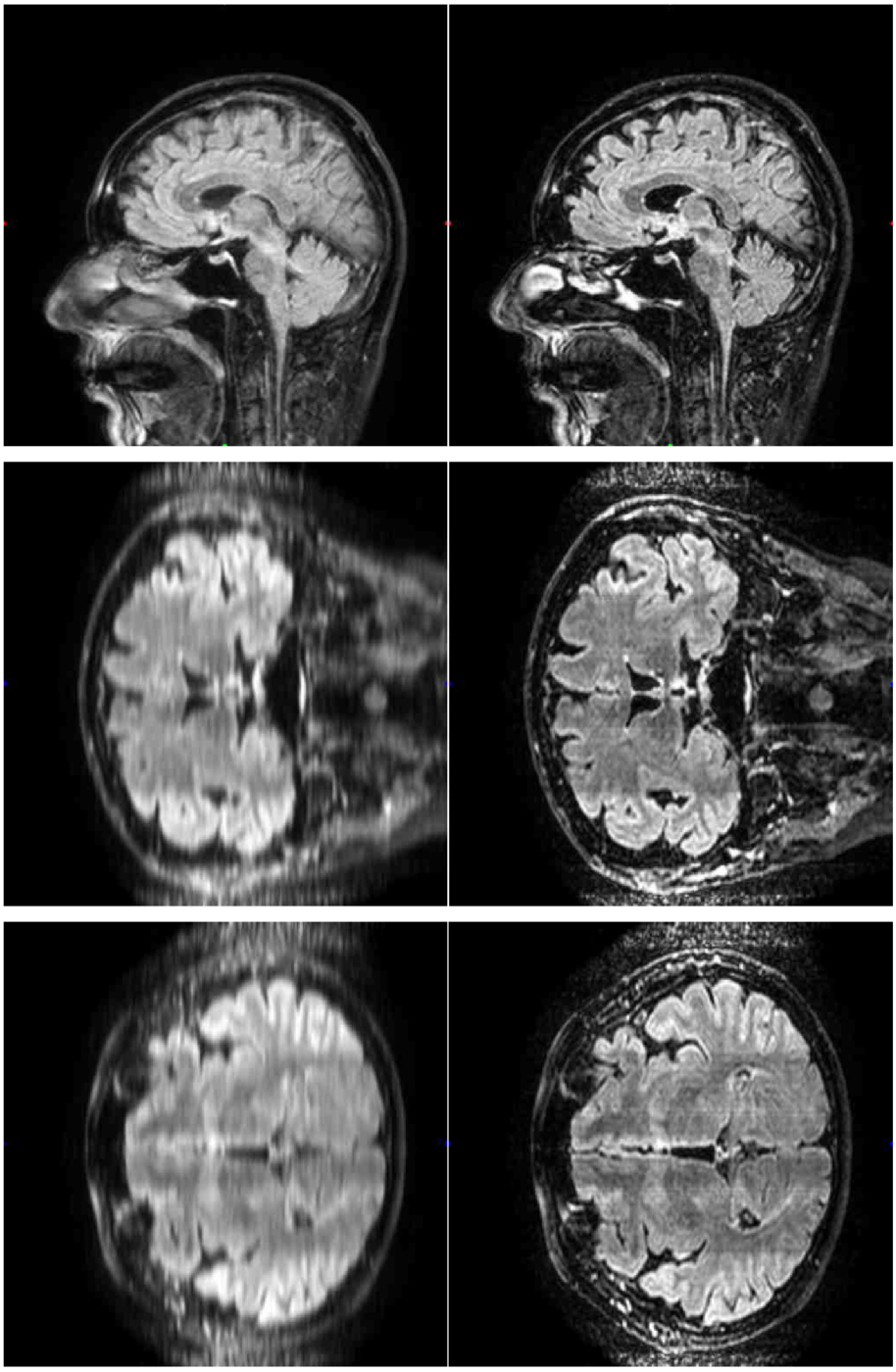


Obr. 4.4: 3D Blur.

4.4.1 Operátor 3D rozostrenia dát

Po aplikovaní teórie popísanej v častiach 4.3 a 4.4 bol vytvorený operátor pre prostredie RapidMiner. Tento operátor očakáva na vstupe 3D snímok, ktorý je možno získať prevodom 2D obrázkov, získaných niektorou metódou medicínskeho zobrazenia dát prevodom na 3D dáta pomocou operátora $2d_to_3d$. Následne vytvorený operátor rotácie zrotuje dáta, ktoré má na vstupe podľa zvoleného smeru, ktorý nastavíme v prostredí RapidMiner. Po zrotovaní nasleduje operácia rozostrenia s jadrom

Gaussovej funkcie, kde silu rozostrenia v smere x, y a z taktiež zvolíme v časti nastavovania parametrov operátora. Výsledné dáta zo vstupného snímku obsahujú teda všetky operácie popísané v častiach 4.3 a 4.4.



Obr. 4.5: 3D rozostrenie v smere xy, yz a xz

4.5 Priemerovací filter

Ďalší filter, ktorý bol vytvorený v jazyku Java a následne aplikovaný do prostredia RapidMiner je tzv. priemerovací filter. Tento filter patrí, ako aj Gaussov filter, do skupiny lineárnych filtrov. Lineárny filter je typ filtra, ktorý využíva konvolúciu určitej masky s pôvodnými hodnotami pixelov v obraze. Touto konvolúciou sa teda získa nová hodnota pixelu, ktorá sa nahradí hodnotou pixela v pôvodnom obrázku. Lineárne filtre teda kombinujú hodnoty pixelov v danom okolí týchto pixelov lineárne ako ich vážený súčet.

Aplikácia tohoto filtru spočívala vo vytvorení matice o rozmere $3 * 3$ a jedná sa podobne ako pri Gaussovom filtri o vyhladzovací filter, ktorý je založený na priemerovaní hodnôt. Možeme si teda zadefinovať maticu, ktorú budeme konvolúciou aplikovať na každý pixel povodného snímku.

$$H(i,j) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

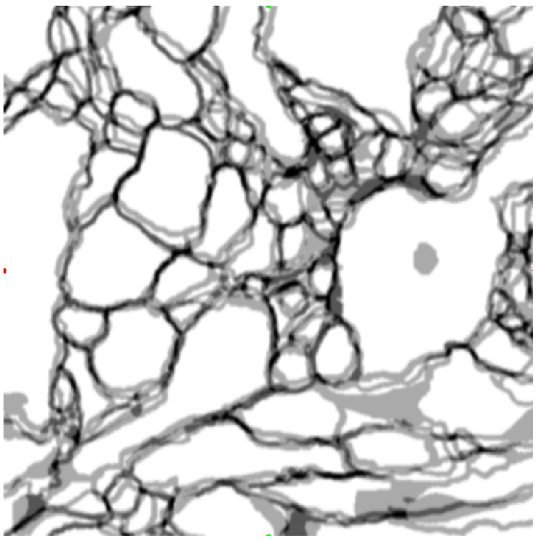
Veľkosť matice H je rovná okoliu samotného filtru a každý prvok $H(i,j)$ špecifikuje váhu daného pixelu v súčte. Ako priemerovacia hodnota sa berie hodnota 9, pretože máme okolie a tým pádom vlastne veľkosť filtrovacej matice $3*3$. Vo všeobecnosti by sme mohli napísať aplikáciu priemerovaného filtru pomocou rovnice:

$$I'(u,v) \leftarrow \frac{1}{9} \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j) \tag{4.10}$$

Priemerovací filter sa často nazýva **box** filter, pretože má tvar štvorca.

4.5.1 Operátor 3D priemerovania dát

Princíp popísaný vyššie bol následne aplikovaný v jazyku Java a aplikovaný vo forme operátora v prostredí RapidMiner. Vstupom operátora je *3d* snímok a výstupom opäť *3d* snímok po aplikovaní filtra.



Obr. 4.6: Average 3D.

4.6 Sobelov filter

Sobelov obrazový filter nám slúži na detekciu hrán v snímku. Na priblíženie fungovania tohoto operátora si musíme upresniť názvy ako *parciálna derivácia* a *gradient*.

Deriváciou viacrozmernej funkcie pozdĺž jednej z jej súradnicových osí sa nazýva *parciálna derivácia*, napríklad vyjadrená funkciou v smere osi x:

$$\frac{\partial f}{\partial x}(x, y) \tag{4.11}$$

Vyššie popísaná funkcia je teda parciálnou deriváciou obrazovej funkcie $f(x, y)$ pozdĺž osi x. Na základe tohoto si môžeme definovať funkciu:

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x, y) \\ \frac{\partial f}{\partial y}(x, y) \end{bmatrix},$$

ktorá sa nazýva *gradientný vektor* alebo skrátene *gradient* funkcie f na pozícii (x, y). Veľkosť toho vektoru (gradientu) potom môžeme následne vyjadriť ako:

$$|\nabla f|(x, y) = \sqrt{\left(\frac{\partial f}{\partial x}(x, y)\right)^2 + \left(\frac{\partial f}{\partial y}(x, y)\right)^2} \tag{4.12}$$

Táto funkcia je nemenná voči rotácii obrazu, teda je nezávislá na orientácii štruktúr vo vnútri snímku. Táto vlastnosť je veľmi dôležitá pre lokalizáciu hrán v obraze, teda funkcia $|\nabla f|(x, y)$ je základom mnohých metód na detekciu hrán.

Sobelov operátor [15] používa lineárne filtre v smere osi x:

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

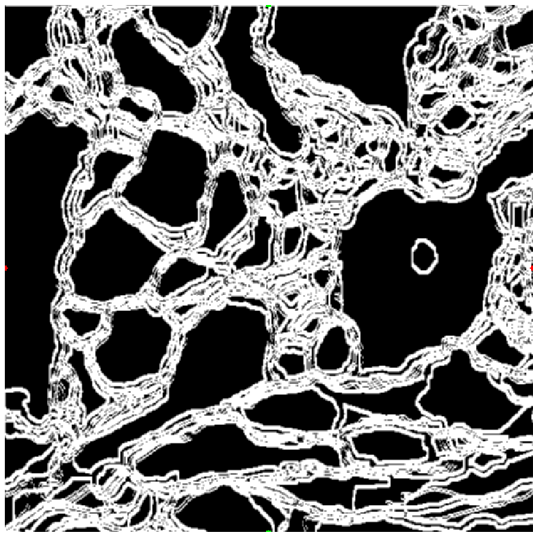
a v smere osi y

$$H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix},$$

ktoré počitajú gradient pozdĺž troch riadkov alebo stĺpcov. Na základe tohoto je zrejmé, že H_x^S vykonáva jednoduchý (box) filter popísaný v kapitole 4.5 predtým ako sa vypočíta gradient v smere osi x. Analogicky to platí samozrejme pre osu y.

4.6.1 Operátor 3D detekie hrán

Do prostredia RapidMiner bol implementovaný Sobelov operátor, ktorý funguje na princípe popísanom v kapitole 4.7. Využíva sa teda lineárnych filtrov v kombinácii počítania gradientu nových pixelov. Operátor má na vstupe 3d snímok a na výstupe snímok s detekciou hrán predchádzajúcich aj nasledujúcich snímkov.



Obr. 4.7: Sobel 3D.

4.7 Mediánový filter

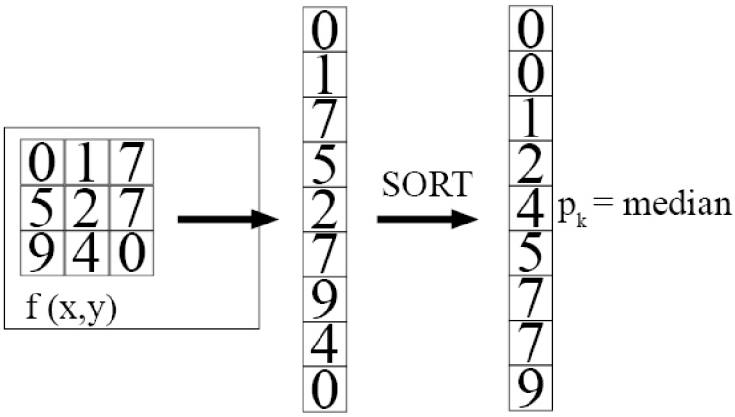
Ďalší operátorom vytvoreným v jazyku Java, bol mediánový filter, ktorý patrí medzi nelineárne filtre. Nelineárne obrazové filtre počítajú výslednú hodnotu pixelu na

pozícii v snímku (x, y) z pixelov, ktoré sa nachádzajú vo vnútri pohybujúceho sa regiónu R_x, y v pôvodnom snímku. Tieto filtre sa nazývajú teda *nelineárne*, pretože pôvodná hodnota pixelu je skombinovaná určitou nelineárnou funkciou.

Mediánový filter nahradzuje každý pixel *mediánovou* hodnotou pixelov v konkrétnom regióne R, teda pomocou rovnice môžeme daný vzťah napísať:

$$f'(x, y) \leftarrow median\{f(x + i, y + j)|(i, j) \in R \tag{4.13}$$

Princíp fungovania mediánového filtru popisuje aj nasledujúci obrázok 4.8. Oblasť filtru je okolie $3 * 3$ pixelov.



Obr. 4.8: Triedenie pixelov v mediánovom filtri.

Medián je definovaný ako prvok p_k z okolia p_i pixelov, a táto stredná hodnota je p_k, ak je poradie pixelov $(p_0, \ldots, p_2k)$ triedené $(p_i \leq p_i + 1)$.

4.7.1 Mediánový operátor v 3D

Podľa popisu v kapitole 4.9 sme definovali princíp fungovania mediánového filtru, ktorý bol následne naprogramovaný v jazyku Java a implementovaný do prostredia RapidMiner.

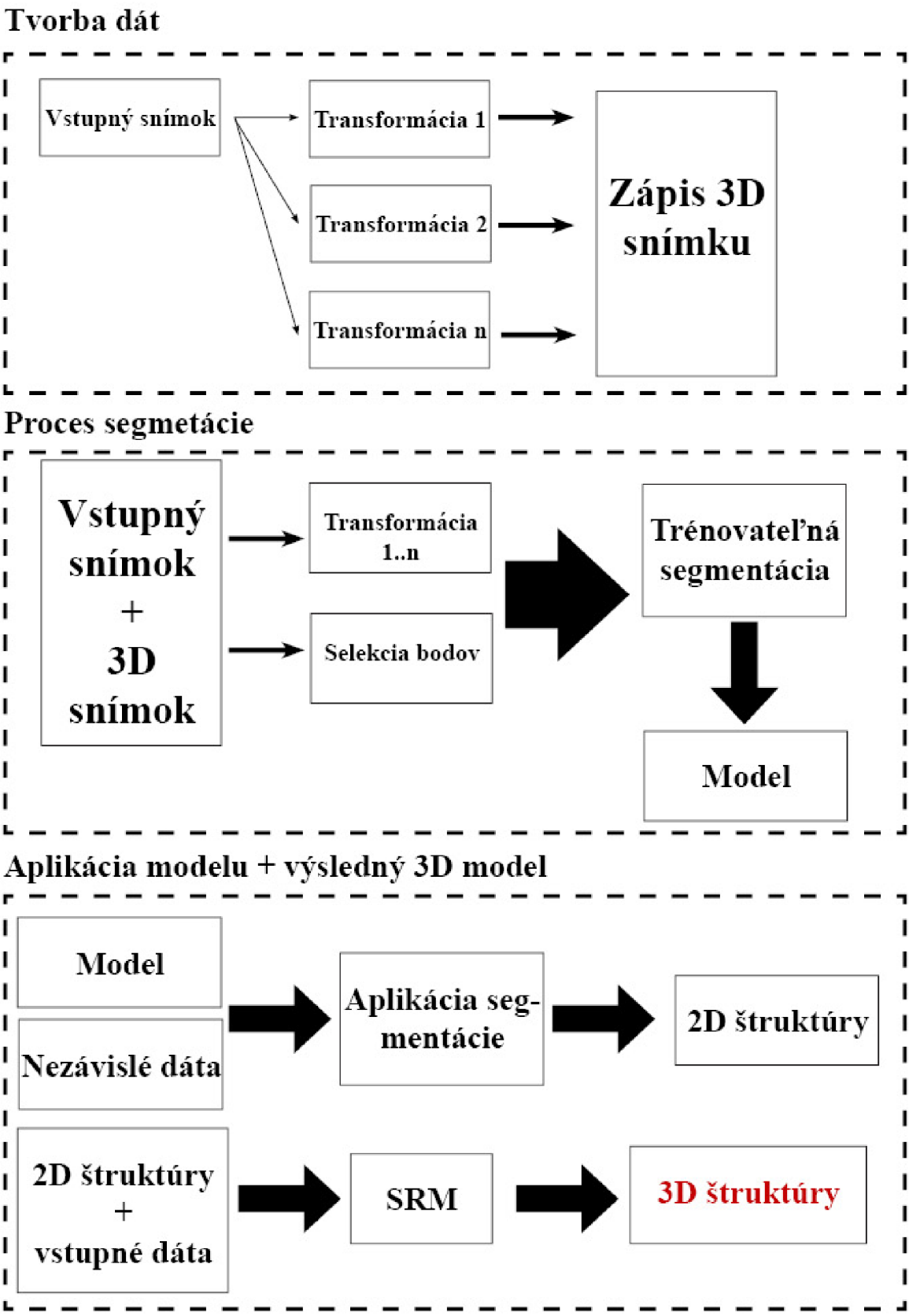
Obr. 4.9: MedianFilter 3D.

4.8 Operátor zápisu 3D dát

Pre potreby ďalšieho spracovávania snímkov bol vytvorený operátor zápisu výsledných predspracovaných snímkov. Operátor má na vstupe stack snímkov, na ktoré sú aplikované obrazové filtre popísane vo vyššých kapitolách. Výsledným výstupom je adresár jednotlivých slicov do samostatných súborov.

4.9 Proces vytvorenia predspracovaných snímkov a ich uloženie

Na nasledujúcom snímku 4.10 môžeme vidieť proces vytvorenia vstupných dát a následný proces segmentácie spolu s vytvorením a aplikovaním modelu, ktorých postup bude vysvetlený v nasledujúcich častiach. V časti tvorby dát na vstupe boli nami zvolené dáta vo forme mikroskopických snímkov štruktúr mozgu a jednotlivé sady snímkov sme upravovali filtrovacími technikami, ktoré sme popísali v kapitolách 4.4.1, 4.5, 4.7, 4.8. Výstupy z tranformačných operátorov sme uložili vo forme adresárov, v ktorých sa nachádzali jednotlivé slice vstupných stackov dát.



Obr. 4.10: Proces navrhnutej segmentácie .

5 TRÉNOVATEĽNÁ SEGMENTÁCIA

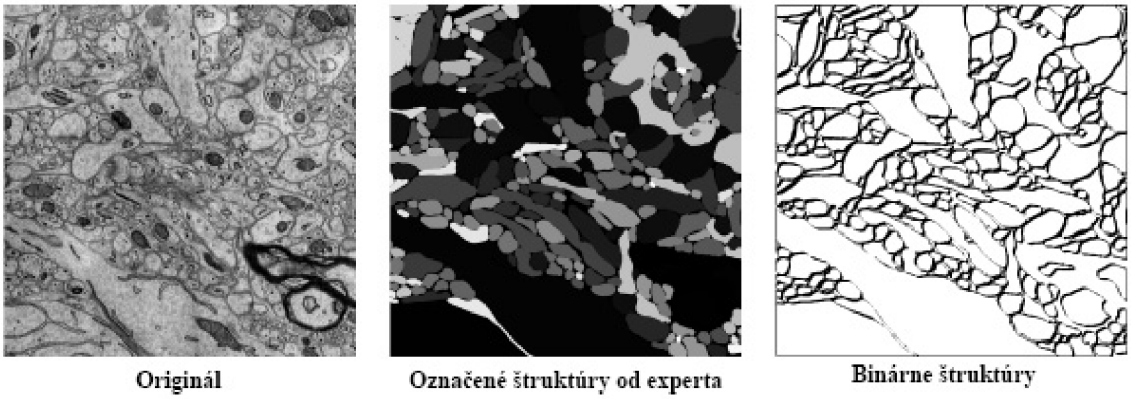
Hlavným účelom a prínosom diplomovej práce bolo na základe vytvorených operátorov v kapitole 4 a ďalších transformácií vytvoriť segmentačný model na základe informácií z predspracovaných snímkov trénovacích dát. Tento model bol následne aplikovaný na nezávislé testovacie dáta. Cieľom práce bolo zhodnotiť navrhnutý postup trénovateľnej segmentácie a jeho následné porovnanie s rôznymi algoritmami segmentačných metód. Celý princíp vychádza z extrakcie lokálnych príznakov vstupných snímkov, ktoré tvorili vstup pre trénovatelnú segmentáciu. Výstup tejto segmentácie bola vytvorená maska, v ktorej sú nasegmentované mikroskopické štuktúry mozgu.

Ďalším krokom bola aplikácia tejto masky spolu s originálnym vstupným snímkom v procese segmentácie pomocou metódy *Statistical region merging*, ktorá už pracuje v 3D. Pomocou tejto techniky bola opät získaná maska fo forme *multilevel segmentation* ktorej výstupom bola 3D štuktúra segmentovaných buniek mozgu. Výsledky práce boli poslané do súťaže ISBI 2013 challenge: 3D segmentation of neurites in EM images[16].

Trénovacie dáta, teda vstupné snímky segmentačného modelu, boli mozgové štruktúry zobrazené pomocou elektrónového mikroskopu [16]. Tieto dáta boli následne segmentované podľa štrukturálnych a funkcionálnych časti. Z dôvodu veľkej časovej náročnosti, pri ktorej špecializovaný odborník musí stráviť čas pri rozpoznávaní bunkových štruktúr bola navrhnutá práve automatická segmentačná metóda na rozpoznávanie týchto štruktúr.

5.1 Trénovacie a testovacie snímky

Dáta, ktoré sú spracovávané v rámci tejto diplomovej práce pozostávajú z testovacích a trénovacích sád snímkov. Každá sada (stack) obsahuje 100 slicov získaných elektrónovým mikroskopom. K trénovacej sade patria aj snímky, ktoré boli analyzované odborníkom, teda sú v nich vyznačené výsledné nasegmentované štuktúry. Bunkové štruktúry sú ale realizované tak, že každý segment má iné číslo, čím je dosiahnutá multilevel segmentácia. Masky v tejto práci sú prevedené na binárne masky, ktorých segmenty sú vyznačené bielou farbou (hodnota pixelu 255) a jeho hraničné oblasti, teda hranice bunky, vyznačené čiernou farbou (hodnota pixelu je 0). Ukážka výsledných masiek je zobrazená na obrázku 5.1.



Obr. 5.1: Testovacie dáta.

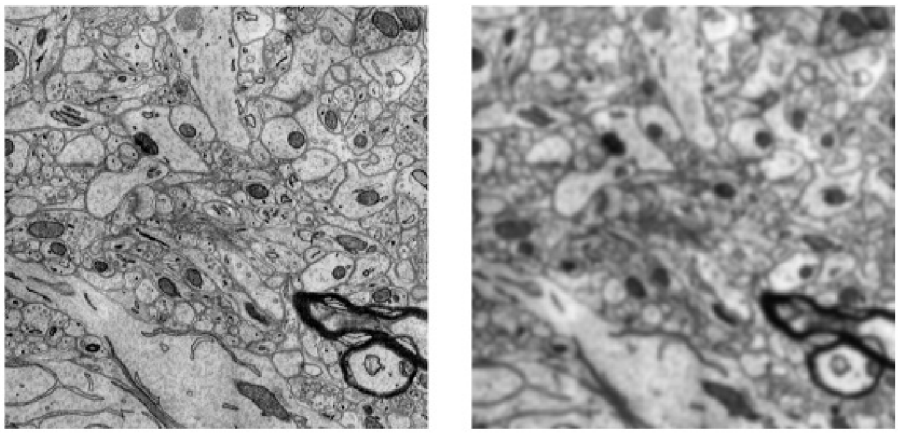
5.2 Predspracovanie

Predspracovanie snímkov je nevyhnutný krok potrebný kvôli získaniu presnejšieho výsledku segmentácie pre daný problém. Predspracovanie teda pozostáva z operátorov, napríklad detekcie hrán v snímku, filtrácie šumu a podobne. K tomuto môžeme využiť spomenuté techniky v kapitole 3. Sú to teda operátory prevádzajúce operácie detekcie hrán alebo rôznu fitráciu. V tejto časti však využijeme aj vytvorené operátory, ktoré sú popísané v kapitole 4.

Vstupný snímok sme teda určitým spôsobom transformovali za účelom získania príznakov, ktoré sú vhodné ako vstup pre trénovateľnú segmentáciu. Tým pádom boli získané viaceré informácie o každom pixeli v obraze. Tieto transformácie využívajú na výpočet hodnoty každého pixelu aj informácie, ktoré sa nachádzajú aj v okolí samotného skúmaného pixelu. Teda, čím viac transformácií použijeme, tým viac informácií môžeme poskytnúť samotnej segmentácii, čoho výsledkom je získanie presnejších hodnôt. V práci bolo použitých niekoľko transformácií, ktoré boli vždy použité na vstupný snímok, na ktorý ešte bola aplikovaná ekvalizácia histogramu. Tento krok bol spravený za účelom, aby všetky snímky danej testovacej alebo trénovacej sady snímkov mali znížený rozdiel intenzít v samotných snímkoch, pretože každý snímok mohol vzniknúť inak, teda do procesu snímania elektrónovým mikroskopom mohli vstupovať faktory, ktoré dali vznik týmto nepresnostiam.

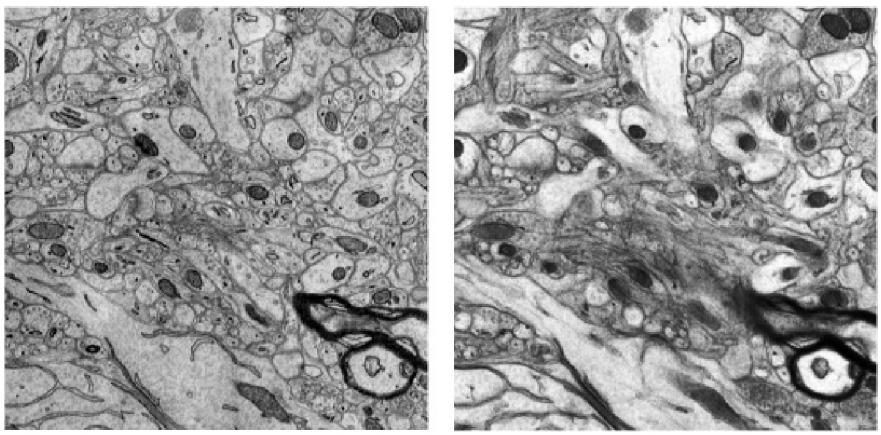
Ďalej uvedieme príklady transformácií, ktoré boli použité v rámci tejto práce a popíšeme ich. Na nami ekvalizovaný vstupný snímok trénovacej sady sme použili tranformácie, ktoré sa využívajú hlavne v obrazovom spracovaní dát. Aplikovali sme obrazové filtre, ktoré rozostrujú dáta, či už v 2D alebo v 3D 4.4.1. Týchto filtrov sme použili niekoľko za účelom získania čo najviac informácií o danom pixele. Použili sme jadrá Gaussovho rozostrenia v smere x, y o veľkostiach 1-32 a v smere

z o veľkostiach 1-7. Ďalšími transformáciami boli filtre, ktoré vykonávajú detekciu hrán a to Sobelov filter a tzv. *Laplacian of Gaussian*, ktorý vlastne rozostrí vstupný snímok, na ktorý aplikuje algoritmus detekcie hrán. Na vstupný snímok sme aplikovali aj operátory, ktoré vykonávajú morfologické operácie popísané v kapitole 3.2, a následne aplikovaný algoritmus skeletonizácie snímku [17]. Ďalšie použité filtre boli typu minimálneho, maximálneho, rozdielového filtru. Pri aplikácii filtrov sa využívalo aj prahovania, a to na základe ISODát [18]. Na nasledujúcich obrázkoch si ukážeme výber niektorých transformácií, ktoré boli aplikované v procese trénovateľnej segmentácie. Na nasledujúcich obrázkoch je možno vidieť snímky, na ktoré bol použitý operátor 2D a 3D rozostrenia v porovnaní s originálnym snímkom 5.2.



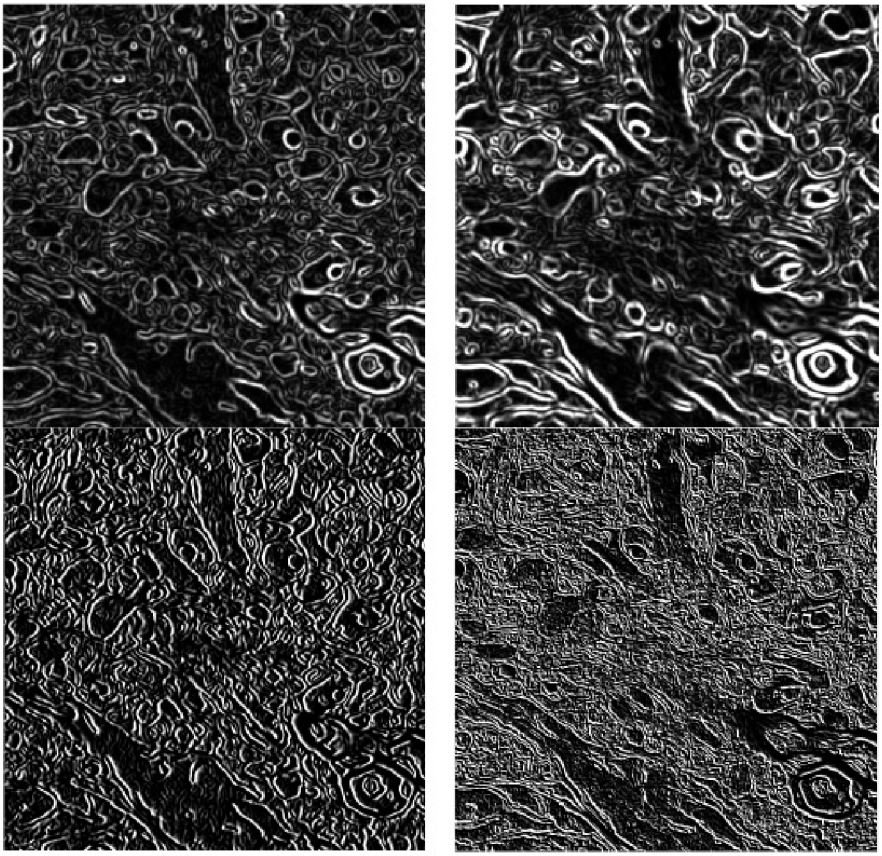
Obr. 5.2: Obrázok vľavo originálny a vpravo rozostrený 2D blurom.

Ďalšie dva obrázky znázorňujú použitie 3D rozostrenia na vstupný snímok, ktorý bol upravený ekvalizáciou histogramu 5.3.



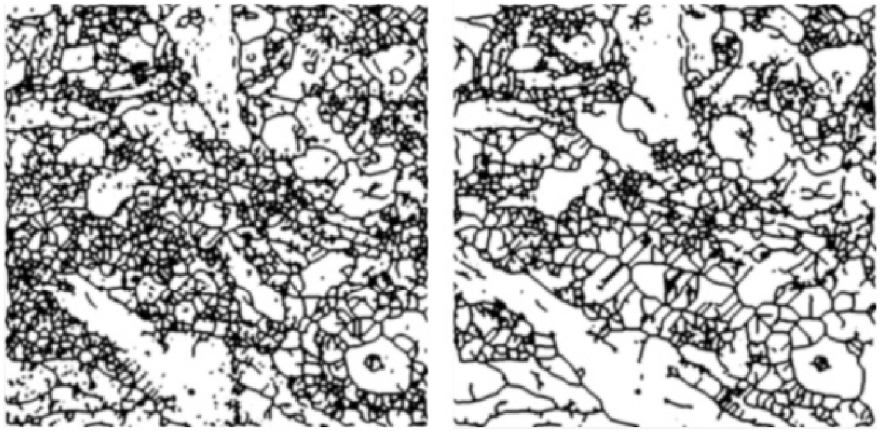
Obr. 5.3: Obrázok vľavo originálny a vpravo rozostrený 3D blurom.

Na ďalších snímkoch sú zobrazené výstupy z operátorov, ktoré vykonávajú funkciu hranových operátorov. Na vstupný snímok je aplikované rozostrenie v 2D a 3D a následne aplikácia hranovej detekcie. Využíva sa Sobelovho operátora a rozdielového filtra, ktorý každý pixel v obraze nahradzuje rozdielom hodnôt okolných pixelov a tým vzniká vyznačenie hrán v snímku 5.4. Na snímky v ľavom stĺpci bol aplikovaný 2D blur a následná detekcia hrán prostredníctvom rozdielového a Sobelovho filtra a na snímky vpravo 3D blur a následná detekcia hrán tiež pomocou rozdielového a Sobelovho filtru.



Obr. 5.4: 2D a 3D blur + detekcia hrán.

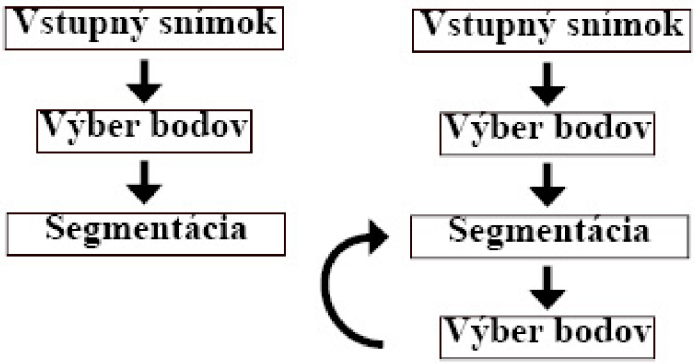
Ďalšie transformácie aplikované na vstupné dáta sú morfologického typu, ako erózia, dilatácia a skeletonizácia. Pri získavaní kostry (skeletu) snímku bola najskôr aplikovaná metóda prahovania a následne dilatácie. 5.5.



Obr. 5.5: Získanie kostry z 2D a 3D rozostreného snímku.

5.3 Výber vstupných bodov trénovateľnej segmentácie

Ďalšou dôležitou časťou celého procesu je výber bodov v snímkoch ako vstup pre trénovateľnú segmentáciu. Tieto body majú význam v tom, že nám pomáhajú pri upresnení výsledkov segmentácie tým, že rozhodujú, ktoré pixely patria do akej oblasti. Inými slovami môžeme povedať, že tieto body upresňujú hranice, kde by bol problém rozpoznania hranice bunky (membrány) od samotného segmentu bunky (vnútro). Tieto body budú tvoriť vstup trénovacieho algoritmu segmentácie. Teda podľa týchto bodov sa budú vyberať príznaky *(feature)*, podľa ktorých sa bude algoritmus rozhodovať, ktoré pixely budú tvoriť segment bunky a ktoré pixely budú tvoriť hraničnú oblasť bunky.

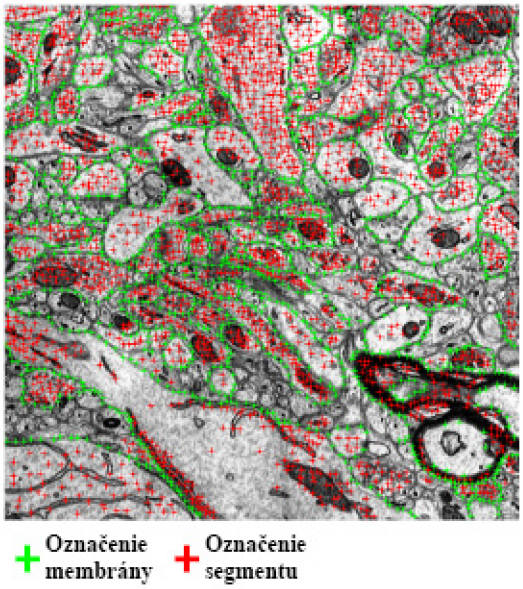


Obr. 5.6: Proces výberu bodov pre vstup trénovateľnej segmentácie.

Tieto body sa môžu zadávať pred samotným vstupom segmentácie ako nejaká sada parametrov segmentácie, alebo sa môžu voliť iteračne. To znamená, že výber

bodov aplikujeme aj po výstupe trénovateľnej segmentácie, a podľa presnosti výsledku ešte definujeme nové body, ktoré budú pri ďalšom kole segmentácie aplikované do procesu 5.6. Toto môže byť veľmi výhodné ak potrebujeme dolaďovať výsledky zo samotnej segmentácie.

V nasledujúcom obrázku 5.7 možno vidieť definovanie takých bodov pre samotnú segmentáciu. Červené body vyznačujú samotný segment bunky, teda jej vnútro a zelené body vyznačujú hranice bunky.



Obr. 5.7: Výber bodov pre trénovateľnú segmentáciu.

5.4 Extrakcia príznakov

Základom tohoto procesu je prevod dát z predspracovaného snímku na dáta vhodné ako vstup učiacich sa algoritmov. Tieto dáta si môžeme predstaviť ako tabuľku hodnôt, kde riadky sú popisky jednotlivých segmentačných častí a stĺpce sú príznaky, teda nejaký typ hodnôt, ktoré vystihujú danú oblasť a sú vypočítané v rámci celej oblasti. Tieto príznaky sa teda získali prostredníctvom transformácií v časti predspracovania 5.2. V nasledujúcich častiach práce bude popísaný princíp a formát týchto príznakov ako vstup učiacich sa algoritmov.

5.5 Trénovateľná segmentácia

V nasledujúcej časti bude rozobraná samotná problematika trénovateľnej segmentácie. Ako bolo spomínané v predchádzajúcich častiach, vstupom trénovateľnej seg-

mentácie boli snímky upravené pomocou ekvalizácie historamu a následne aplikované obrazové filtre, z ktorých boli získané lokálne príznaky. Celý proces trénovateľnej segmentácie možno teda rozdeliť do niekoľkých častí:

- získanie vstupných bodov trénovateľnej segmentácie
- aplikácia rôznych transformácií prostredníctvom obrazových filtrov za účelom získania lokálnych príznakov
- použitie týchto vstupných bodov a lokálnych príznakov ako trénovacie dáta samotnej segmentácie
- aplikácia trénovacích dát do modelov učiacich sa algoritmov
- získanie výsledného modelu segmentácie (masky bunkových štruktúr)
- získanie trénovacieho modelu a jeho aplikácia na nezávislé vstupné dáta
- aplikácia masky a vstupného snímku do následného procesu 3D segmentácie
- získanie výsledných bunkových štruktúr formou multilevel segmentov
- zhodnotenie a porovnanie výsledkov

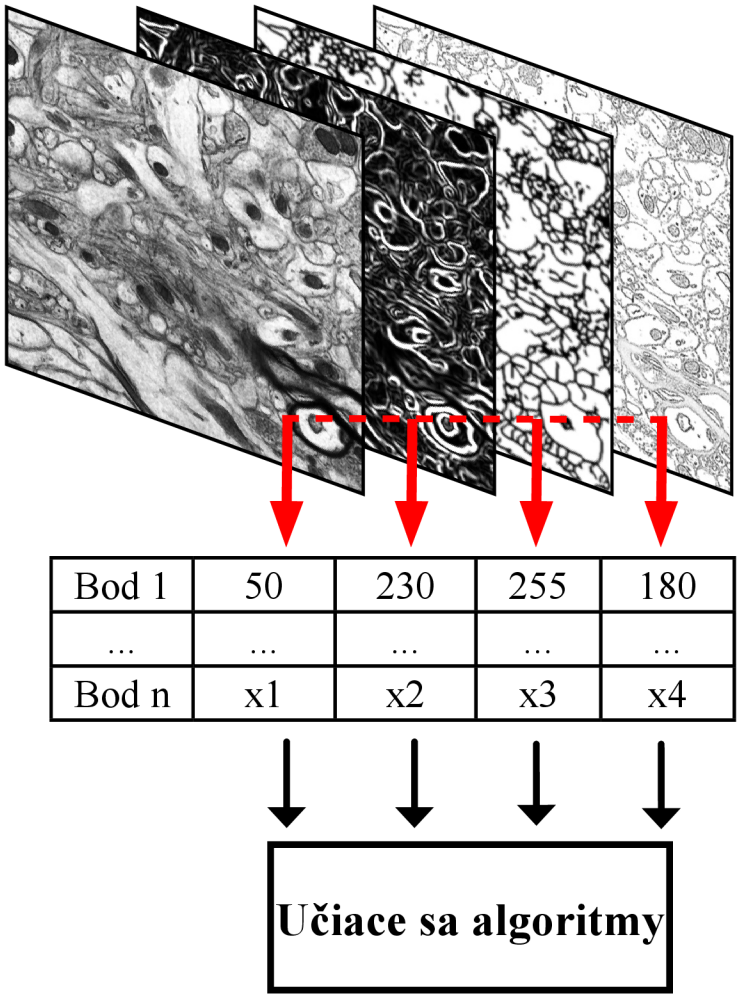
V tomto procese boli použité body, ktorými sme upresnili hranice a samotný segment bunky, a použitím niekoľkých transformácií boli získané lokálne príznaky. Tieto trénovacie dáta majú formu tabuľky, a na jej základe je natrénovaný model učiacich sa algoritmov. Na obrázku 5.8 je vidieť zápis týchto atribútov, s ktorými pracujú učiace sa algoritmy.

Row No.	label	Att0	Att1_g2.0	Att1_g3.0	Att1_g4.0
1	true	170	170	170	170
2	true	180	179	177	175
3	false	106	107	108	109
4	true	194	193	193	191
5	true	106	107	110	112
6	true	146	146	146	146
7	true	180	178	176	174
8	false	145	146	147	148
9	true	156	156	156	155
10	true	172	171	170	168

Obr. 5.8: Lokálne príznaky.

Princíp transformácie lokálnych príznakov môžeme popísať nasledovným spôsobom. Pre každý zvolený bod sa určí jeho hodnota v každom vstupnom snímku, teda originálnom aj transformovanom. Keďže týchto transformácií je n, tak je vypočítaný vektor, ktorý vyjadruje všetky hodnoty v daných transformáciách. Teda

v tabuľke lokálnych príznakov riadok tabuľky určuje bod, ktorý bol zvolený a jednotlivý stĺpec určuje hodnotu daného pixelu v snímku. Inými slovami jeden riadok je vyjadrený ako určitý vektor, ktorý sa vzťahuje k určitému zvolenému bodu. Tabuľka pracuje s hodnotami typu *integer* 5.9.



Obr. 5.9: Princíp tvorby tabuľky lokálnych príznakov.

5.5.1 Druhy použitých učiacich sa algoritmov

Pri procese tvorby segmentačného modelu boli využité niektoré dobre známe druhy učiacich sa algoritmov. Pretože každý algoritmus má svoje výhody aj nevýhody, tak kvôli tomuto faktoru bolo vyskúšaných niekoľko týchto algoritmov. V tejto práci boli použité nasledovné učiace sa algoritmy:

- algoritmus *Random Forests*
- algoritmus rozhodovacích stromov *Decision Free*

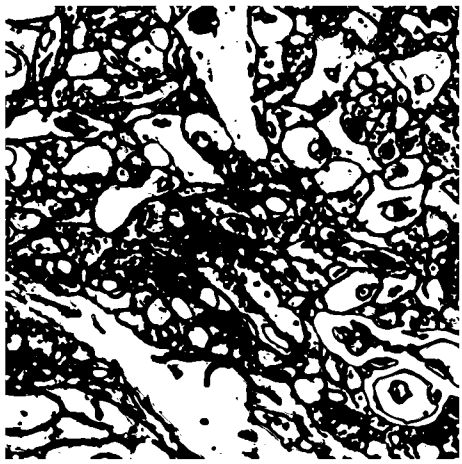
- *SVM* algoritmy a použitím niekoľkých jadier
- algoritmus *Naive Bayes Classification*
- technika lineárnej regresie

Následne budú stručne vysvetlené použité segmentačné techniky a ich dopad na vstupné snímky trénovateľnej segmentácie.

5.5.2 Rozhodovací strom

Tento typ algoritmu je založený na modele, ktorý využíva stromovo orientovaný graf. Je to typ tzv. inverzného stromu, pretože jeho koreň sa nachádza na vrchu samotného grafu a rastie smerom dolu. Táto dátova reprezentácia má výhodu v tom, že je veľmi ľahko interpretovateľná. Cieľom tejto metódy je vytvorenie klasifikačného modelu, ktorý predpovedá hodnoty cieľového atribútu, v našom prípade teda, či daný pixel bude vo vnútri segmentovanej bunky alebo bude tvoriť membránu bunky. Tento model používa ako vstupné hodnoty niekoľko atribútov, teda lokálne príznaky. Tieto atribúty nazveme *sada príznakov (Example Set)*. Každý uzol tohoto stromu reprezentuje jeden určitý vstupný atribút. Počet hraničných uzlov každého nadradeného uzlu (rodiča) je rovný počtu všetkých možných hodnôt vstupného atribútu. Každý list, teda uzol reprezentuje hodnotu atribútu, ktorá je daná hodnotami vstupných atribútov reprezentovaných jednoznačnou cestou od koreňa stromu po list (uzol). Rozhodovacie stromy sú generované rekurzívnym delením, to znamená, opakovateľné delenie na základe hodnôt atribútov. Tento proces sa skladá z niekoľkých častí, ale vo všeobecnosti môžeme povedať, že rekurzia končí v okamihu, keď všetky vzorky nadobúdajú rovnakú hodnotu cieľového atribútu. V ďalšom prípade rekurzia môže skončiť ak väčšina vzoriek má podobnú hodnotu cieľového atribútu. Ďalším procesom tohoto algoritmu je *zastrihávanie stromu*. Jedná sa o techniku, kde list stromu, ak nespĺňa určitý požiadavok, tak je zo stromu odstránený.

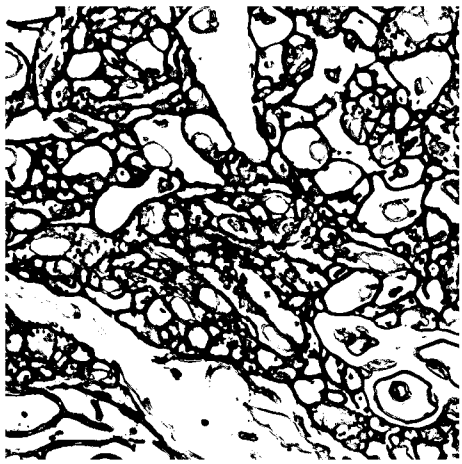
Na nasledujúcom obrázku 5.10 možno vidieť aplikáciu algoritmu rozhodovacieho stromu do procesu trénovateľnej segmentácie.



Obr. 5.10: Výsledný model rozhodovacieho stromu.

5.5.3 Algoritmus náhodný les

Random Forest algoritmus generuje súbor stromovo orientovaných grafov. Vychádza to aj z odvodenia, že viac stromov tvorí les. Takýto les tvoria tzv. *náhodné stromy.* Tento typ stromovo orientovaných grafov pracuje na podobnom princípe ako rozhodovacie stromy, avšak pre každé delenie je použitá náhodná podmnožina atribútov, ktoré sú dostupné. Výsledný model, teda les, obsahuje špecifikovaný počet modelov stromov. Cieľom tejto metódy je vytvoriť klasifikačný model, ktorý predikuje hodnotu cieľového atribútu, ktorého hodnota je založená na niekoľkých vstupných atribútoch vstupnej sady. Každý vnútorný uzol stromu zodpovedá jednému vstupnému atribútu. Počet hrán menovitého vnútorného uzlu je rovný počtu všetkých možných hodnôt, ktoré odpovedajú danému vstupnému atribútu.



Obr. 5.11: Výsledný model algoritmu Random Forest.

Na obrázku 5.11 je zobrazený výsledný model po aplikácii učiaceho sa algoritmu

Random Forest. Oproti rozhodovacím stromom je dosiahnuté lepšieho výsledku práve spôsobom popísaným vyššie. Nie je teda zvolený jediný strom, ale je ich niekoľko a tie berú náhodnú množinu vstupných atribútov.

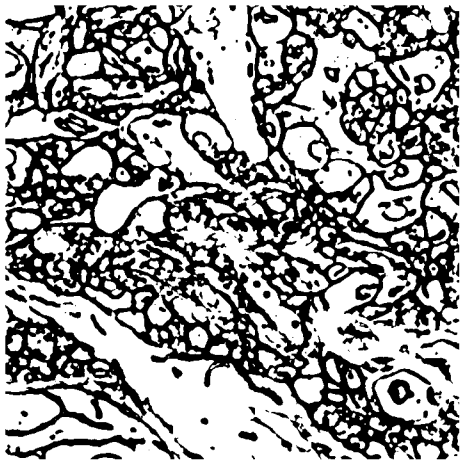
5.5.4 SVM

Metóda SVM (Support Vector Machines) je typ učiaceho sa algoritmu, ktorý berie sadu vstupných dát a pre každý daný vstup predikuje, ktorá z dvoch možných tried zaradení sa skladá z daného vstupu. Táto metóda je binárna, pretože rozdeľuje dáta do dvoch tried. Z danej sady vstupných vzoriek (príznakov), je každý označený do jednej alebo druhej triedy a následne učiaci sa algoritmus SVM vytvorí model, ktorý priradí nové vzorky do týchto dvoch tried. SVM je teda reprezentácia vzorkov ako body v rovine (teda nejaký priestor), ktoré sú mapované tak, aby táto rovina obsahovala na jej kraji čo najväčší priestor bez bodov. Nové vzorky sú následne mapované do takej roviny podľa toho, na ktorú stranu medzi rovinami (priestor bez bodov) dopadajú.

Dôležitou časťou tejto metódy je použitie tzv. jadrových operácií (kernel transformations) $K(x, y)$ [19]. Tieto transformácie prevádzajú priestor vzoriek, v našom prípade teda lokálne príznaky, do priestoru transformovaných príznakov, ktorý je vyššej dimenzie. V tejto práci sme použili jadrové tramsformácie typu *dot* a *anova*.

SVM jadro - dot

Toto jadro je definované $K(x, y) = x * y$. Na obrázku 5.12 je vidno aplikácia metódy SVM-DOT.



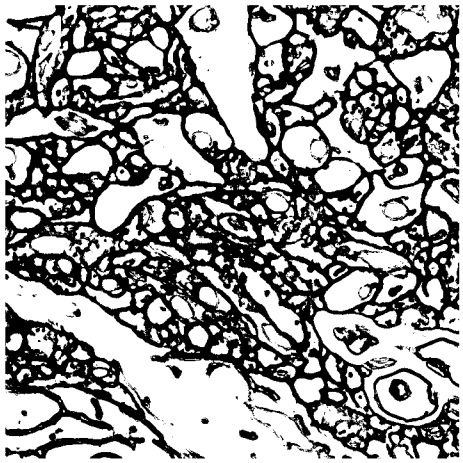
Obr. 5.12: Výsledný model algoritmu SVM a typu jadra *dot*.

SVM jadro Anova

Tento typ jadra je definovaný podľa nasledujúcej rovnice:

$$K(x, y) = \sum_{k=1}^{n} exp(-\sigma(x^k - y^k)^2)^d \tag{5.1}$$

Na nasledujúcom obrázku 5.13 je ukázaná aplikácia jadra typu ANOVA.



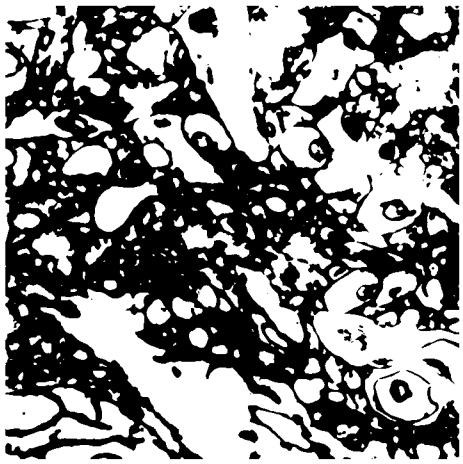
Obr. 5.13: Výsledný model algoritmu SVM a typu jadra *anova*.

5.5.5 Bayesova klasifikácia

Metóda je založená na princípe Bayesovej vety o pravdepodobnosti. Bayesovu vetu môžeme zapísať ako:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{5.2}$$

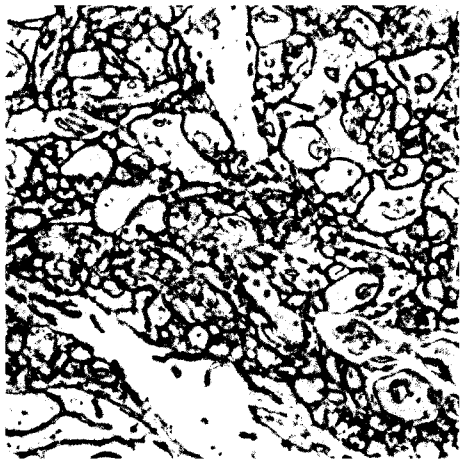
Bayesov klasifikátor predpokladá, že prítomnosť určitého javu (príznaku) atribútu nesúvisí s prítomnosťou nejakého ďalšieho javu (príznaku). Na obrázku 5.14 je vidieť aplikáciu tejto metódy v procese trénovateľnej segmentácie. Je vidieť, že štruktúry sa akokeby „zliali“ do seba, takže táto metóda nevykazuje veľmi perspektívne výsledky. Na druhej strane výhodou Bayesovej klasifikácie je, že vyžaduje pomerne malú časť trénovacích dát potrebných pre klasifikáciu.



Obr. 5.14: Výsledný model algoritmu Bayesovej klasifikácie.

5.5.6 Lineárna regresia

Lineárna regresia je metóda typu numerickej predikcie. Je založená na štatistickom meraní, ktoré sa pokúša určiť vzťah medzi jedným parametrom, teda hodnotou cieľového atribútu, a ostaných skupín parametrov. Regresia je použitá k predikovaniu priebežných hodnôt, zatiaľ čo klasifikácia je použitá k predikovaniu kategórie, do ktorej bude patriť cielový atribút. Nasledujúci obrázok 5.15 znázorňuje aplikovanie tejto metódy v procese trénovateľnej segmentácie.



Obr. 5.15: Výsledný model algoritmu lineárnej regresie.

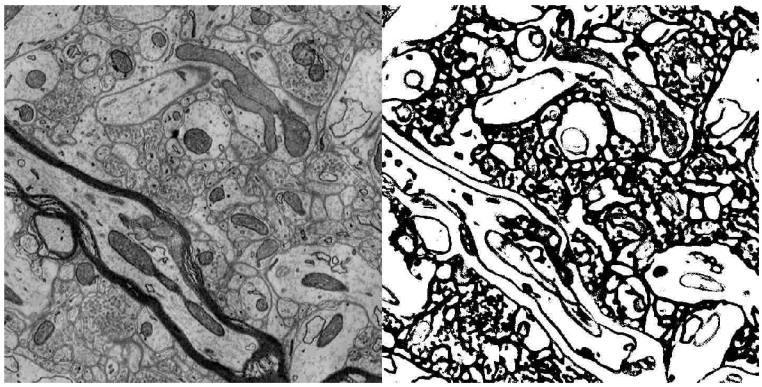
5.6 Tvorba trénovacieho modelu a aplikácia na nezávislé dáta

V tejto časti bude popísaný postup aplikácie výsledného segmentačného modelu na nezávislé testovacie dáta. V predchádzajúcich častiach boli popísané metódy učiacich sa algoritmov, ktoré boli použité v tejto práci. Následne je tento model pripravený aby bol aplikovaný na nezávislé dáta za účelom získania výsledkov a presnosti použitého učiaceho sa algoritmu a tým pádom aj celého procesu trénovateľnej segmentácie.

Trénovateľná segmentácia funguje v dvoch režimoch:

- trénovací režim
- režim aplikácie

Trénovací režim mal úlohu v natrénovaní dát zo vstupných snímkov za účelom získania nasegmentovaných bunkových štruktúr, zatiaľ čo režim aplikácie bude slúžit k aplikácii tohoto modelu na nezávislé vstupné dáta inej sady snímkov. Trénovací model je teda potrebné vytvoriť len raz a následne je tento model možné použit na akúkoľvek sadu vstupných snímkov. Z popísaných segmentačnych techník bola pre túto prácu vybraná metóda Random Forest, pretože vykazovala najlepšie výsledky, čo sa týka súvislého spojenia bunkových štruktúr. Samozrejme, použitie segmentačných techník závisí od rôznorodosti vstupných dát a pre každý daný problém je vhodnejšia iná technika. Nasledujúci snímok 5.16 zobrazuje aplikáciu výsledného segmentačného modelu na testovaciu sadu snímkov.

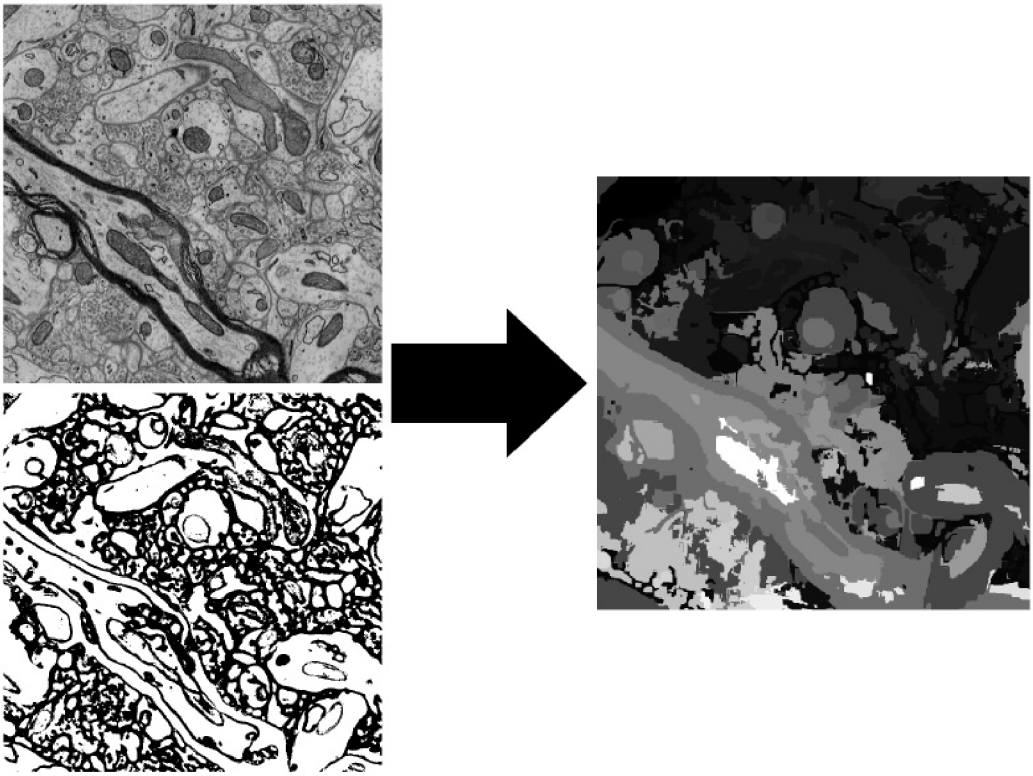


Obr. 5.16: Výsledné štruktúry po aplikácii segmentačného modelu.

5.7 Multilevel segmentácia

V tejto časti procesu budú využité výsledné masky trénovateľnej segmentácie, teda nasegmentované výsledné bunkové štruktúry. Výsledné masky štruktúr sú spojené s pôvodným snímkom a pomocou segmentácie *Statictical Region Merging* sú nasegmentvané výsledné štruktúry vo forme 3D. Na rozdiel od prechádzajúcej časti segmentácie, kde výsledná maska bola tvorená cieľovými atribútmi v binárnej podobe, teda hodnota cieľového atribútu bola *true* alebo *false*, v tomto druhu segmentácie sa už pracuje v 3D priestore, teda každá nasegmentovaná štruktúra má svoje špecifické označenie.

Týmto druhom segmentácie bolo teda dosiahnutého rozdelenia segmentovaných štruktúr do niekoľko vrstvového stacku snímkov. Vo výslednom zobrazení je možné jednotlivé štruktúry samostatne zobrazovať, kde je možné vidieť výsledný 3D model jednotlivých bunkových štruktúr. Na nasledujúcom obrázku 5.17 je vidieť výslednú multilevel masku, ktorá bola cieľovou štruktúrou pre súťaž ISBI 2013 challenge: 3D segmentation of neurites in EM images[16].



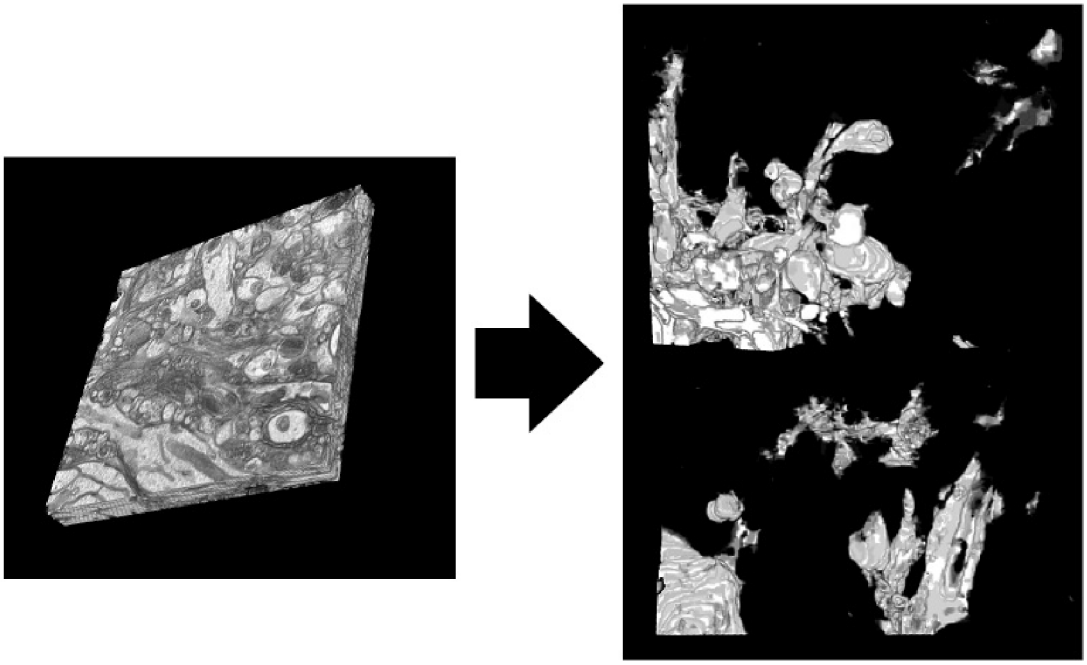
Obr. 5.17: Výsledné označenie 3D štruktúr.

6 VÝSLEDKÉ MODELY A ZHODNOTENIE PRESNOSTI VÝSLEDKOV

Záverom práce je zhodnotenie celkového výsledku navrhnutej metódy trénovateľnej segmentácie a zhodnotenie použitých algoritmov a zobrazenie výsledných 3D mikroskopických štruktúr mozgu získané elektrónovým mikroskopom.

6.1 3D modely bunkových štruktúr

Po aplikácii všetkých častí procesu trénovateľnej segmentácie bol získaný model segmentovaných štruktúr mozgu, ktorý po aplikácii následnej segmentácie na úrovni 3D je možno zobraziť vo forme jednotlivých bunkových štruktúr. Na nasledujúcom obrázku 6.1 je zobrazený originálny stack vstupných dát a výsledné niektoré segmenty bunkových štruktúr v trojrozmernom priestore.



Obr. 6.1: Výsledné 3D modely.

6.2 Presnosť navrhnutého prístupu segmentácie

Presnosť prístupu navrhnutej segmentačnej metódy môžeme rozdeliť do dvoch častí a to:

- presnosť výsledných štruktúr v 2D

- presnosť výsledných štruktúr v 3D

V procese zhodnotenia presnosti výsledkov 2D segmentácie boli použité nasledujúce 3 kritériá:

- Pixel Error
- Warping Error
- Rand Error

Pixel Error

Pixel Error [20] je metrika, v ktorej toto číslo udáva počet pixelov v snímku, na ktorých sa strojové učenie a označenia od špecializovaného odborníka nezhodujú. Stanovme si premennú L_i, ktorá udáva hodnotu označenia štuktúry L v obrázku i a premennú L_o, ktorá udáva hodnotu označenia štruktúry od odborníka. Pixel error je potom číslo, ktoré je rovné počtu pixelov v častiach snímku, kde sa tieto dve označenia líšia. Môžeme to zapísať pomocou Euklidovskej vzdialenosti ako $\|L_i - L_o\|^2$.

Táto metrika je častou využívaná pre svoju jednoduchosť, hoci trpí chybou v tom, že je veľmi citlivá na malé posuvy v okolí hraničných oblastí.

Wapring Error

Warping Error metrika, je typ metriky založenej na topológií porovnávania dvoch segmentoch. Ak L_o môže byť transformovaná na L_i pomocou sekvencie otáčania pixelov, kde každý pixel zachováva súbor požadovaných topologických vlastností a objavuje sa len v miestach v rámci masky M, tak hovoríme o tzv. *deformačnej chybe* (Warping Error). Táto deformácia môže byt zapísaná formou rovnice 6.1, v ktorej T značí pixelovú chybu vzťahujúcu sa na deformáciu L_o:

$$D(T\|L_o) = min_{L_i \triangleleft L_o}\|T - L_o\|^2 \tag{6.1}$$

Podobrobnejši popis tejto chyby je vysvetlený v literatúre[20].

Rand Error

Rand Error je známa metrika založená na Rand Indexe [20], ktorá pre výpočet využíva podobnosť dvoch dátovych klastrov. Využíva sa následného vzťahu:

$$R = \frac{a+b}{\begin{pmatrix} n \\ 2 \end{pmatrix}}, \tag{6.2}$$

kde a značí počet párových elementov (pixelov) v rovnakom objekte S a b značí počet párových elementov v rôznych objektoch S a S_1 a n vyjadruje počet pixelov daného snímku.

V nasledujúcej tabuľke 6.1 je uvedený prehlaď metrík jednotlivých algoritmov trénovateľnej segmentácie v porovnaní binárnych masiek výsledného segmentačného modelu a masky získanej prevodom multilevel masky, ktorá bola vytvorená špecializovaným pracovníkom a prevedená do binárnej podoby. Masky sú znázornené na obrázku 5.1.

Presnosť algoritmov			
Algoritmus	**Pixel Error**	**Warping Error**	**Rand Error**
Random Forest	0.214	0.026	0.916
Decision Tree	0.318	0.024	0.928
SVM - ANOVA	0.212	0.026	0.917
SVM - DOT	0.219	0.018	0.903
Bayes	0.292	0.017	0.865
Lineárna regresia MD5	0.218	0.038	0.869

Tab. 6.1: Presnosť jednotlivých segmentačných metód

Cieľom diplomovej práce bolo vytvorenie a klasifikovanie mikroskopických štruktúr v 3D, teda porovnávanie výslednej multilevel masky bolo realizované po nahratí výsledných štruktúr na server súťaže ISBI 2013 challenge: 3D segmentation of neurites in EM images[16]. Pre túto potrebu sú výsledky hodnotené podľa metriky:

- Adaptíny Rand Error - je definovaný ako $1 - F_{score}(max)$ metriky Rand Index [16].

SNEMI3D: 3D Segmentation of neurites in EM images		
Názov	**Meno skupiny**	**Rand Error**
Špecializovaný odborník	manuálne označené štruktúry	0.059975952
Link3D	SCI	0.151449312
Rhoana	Rhoana	0.180719720
SRM - bez masky	SPLab	**0.854746949**
SRM - s maskou	SPLab	**0.664958999**

Tab. 6.2: Výsledky tímov realizujúcich multilevel segmentáciu

Hodnoty v tabuľke 6.2 ukazujú priamo výsledky viacúrovňovej segmentácie vyhodnotené pre súťaž ISBI 2013 challenge.

Z výslednej tabuľky 6.2 je možno vidieť zlepšenie segmentácie v 3D za použitia vytvorenej binárnej masky bunkových štruktúr. Oproti segmentácii metódou Statistical Region Merging bez použitia masky vykazuje lepšie výsledky. Segmentácia teda prebiehala v dvoch časťiach a to na úrovni 2D, kde pomocou trénovateľnej segmentácie bola získaná binárna maska štruktúr a následná časť na úrovni 3D, kde táto maska bola použitá spolu s metódou SRM.

Oproti ostatným výsledkom multilevel segmentácie však táto metóda vykazuje nedostatky v oblastiach presného určenia membrán buniek. Je to sposobené tým, že v snímkoch získaných elektrónovým mikroskopom, sú niektoré štruktúry ťažko rozoznateľné a tým pádom učiaci sa algoritmus nevie rozhodnúť, ako má výslednú štruktúru označit. Tento faktor je možno čiastočne odstrániť v lepšom použití niektorých transformácií a vylepšeným procesom samotnej segmentácie.

Samotná aplikácia trénovateľnej segmentácie nie je obmedzená len na uvedený postup. Nie je teda obmedzená len na lekárske využitie, ale aj na všebecné použitie, čo sa týka obrazového spracovania dát. Pretože každý problém, ktorý ma segmentácia vyriešiť, je spojená vo vhodnom zvolení postupov a parametrov samotných blokov segmentácie. Vo všeobecnosti sa dá povedať, že postup zvolený v tomto procese je aplikovateľný na mnohé problémy tým, že odstraňuje manuálnu klasifikáciu dát tým, že po natrénovaní segmentačného modelu je možné použiť ho na navzájom nezávislé dáta.

7 ZÁVER

Táto práca sa zaoberá problematikou 3D segmentácie obrazových dát. Zameriava sa na metódy segmentácie, ktoré sa vhodnou kombináciou spájajú za účelom presnejšieho výsledku. Jednotlivé druhy segmentácie boli popísané v prvej kapitole. Ďalšia kapitola sa zaoberá vytvorením obrazových filtrov využitých v segmentačnom procese. V nasledujúcej kapitole bola navrhnutá metóda trojrozmernej segmentácie obsahujúca predspracovanie, ktorá slúži k príprave obrázku do vhodnejšej podoby pre spracovanie. Ďalej je to výber vstupných bodov ako vstup trénovateľnej segmentácie, extrakcia príznakov, ktoré reprezentujú určitú oblasť vstupného snímku a vyjadrujú to pomocou nejakej číselnej hodnoty. Následná segmentácia potom rozdeľuje časti vstupného snímku s podobnými vlastnosťami a potom nasleduje určovanie typov objektu vo vstupnom snímku. Metóda segmentácie je založená na rôzných segmentačných algoritmoch, ktoré boli v práci popísané a každý typ týchto algoritmov má svoje výhody aj nevýhody.

Hlavný prínos práce spočíva vo vytvorení algoritmov pre extrakciu príznakov z 3D dát a následné overenie výsledkov pre problematiku mikroskopických 3D dát mozgu a vo vytvorení obrazových filtrov v jazyku Java s použitím knižnice ImageJ. Tieto obrazové filtre majú na výstupe transformovaný vstupný stack snímkov. Stack je transformovaný pomocou konvolúcie. Konvolúcia je použitá v troch smeroch x, y, z a je pri tom využitý princíp superpozície. Následne boli tieto filtre implementované ako operátory v programe RapidMiner. Tieto filtre boli implementované do procesu trénovateľnej segmentácie za účelom rozpoznávania výsledných bunkových štruktúr. Segmentácia pracuje na úrovni 2D a 3D. Výstupom celého procesu aplikácie obrazových filtrov a segmentácie je získaný výsledný 3D model bunkových štruktúr, pričom sa využíva masiek štuktúr na úrovni 2D a 3D. Výsledná maska 3D buňkových štruktúr bola navyše poslaná do súťaže ISBI 2013 challenge: 3D segmentation of neurites in EM images, kde sa zúčastnilo len pár tímov celého sveta. Výsledná presnosť navrhnutého procesu je znazornená v časti 6.

LITERATÚRA

[1] HENDEE, William. R. *Imaging science in medicine* , Encyclopedia of Imaging Science and Technology. New York: John Wiley & Sons, 2002, 536 s. ISBN 978-04-714-6113-5.

[2] HARALICK, Robert M. Image Analysis Using Mathematical Morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 1987,vol. 9, no. 4, s. 532-550.

[3] WANG, Kejun., et.al. An Image Segmentation Method Based on the Improved Snake Model. *IEEE Internation Conference on Mechatronics and Automation*, June 2006, s. 532-536.

[4] LAKARE, Sarang. *3D Segmentation Techniques for Medical Volumes.* Stony Brook: State University of New York, Center for Visual Computing, 2000.

[5] DRASTICH, Aleš. *Tomografické zobrazovací systémy.* Brno: Vysoké Učení Technické, Fakulta Elektrotechniky a Komunikačných Technológií, Ústav biomedicínskeho inženírsktví, 2004, 207 s. ISBN 80-214-2788-4.

[6] PAVLIDIS, Theodosios. Segmentation of pictures and maps through functional approximation. *Computer Graphics and Image Processing*, December 1972, vol. 1, no. 4, strana 360-372.

[7] OSHER, S., SETHIAN J. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 1988, vol. 79 strana 12-49.

[8] SAHOO, P. K., et al. A Survey of Thresholding Techniques. ,*Computer Vision, Graphics, and Image Processing*, 1988, vol. 41, no. 2, strana 233-260.

[9] PHAM, Dzung. L., et al. Current Methods in Medical Image Segmentation. *Annual Review of Biomedical Engineering*, August 2000, vol. 2, strana 315-337.

[10] SEBASTIAN, T. B., et al. Segmentation of carpal bones from CT images using skeletally coupled deformable models. *Medical Image Analysis*, March 2003, vol. 7, no. 1, strana 21-45.

[11] CHIANG, M., et al. A time-efficient pattern reduction algorithm for k-means clustering. *Information Sciences*, 2011, vol. 181, no. 4, strana 416-431. ISSN: 0020-0255.

[12] LI, Stan. Z. *Markov Random Field Modeling in Image Analysis*. 3rd reprint, Springer, 2010, 384 s. ISBN 978-18-499-6767-9.

[13] HARALICK, Robert. M., et al. Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing*, January 1985, vol. 29, no. 1, strana 100-132.

[14] POMMERT, A., et al. Volume vizualization in Brain Mapping: The Methods. *Academic Press, San Diego*, 2002, vol. 26, strana 707-723.

[15] DAVIS, L. S. A survey of edge detection techniques. *Computer Graphics and Image Processing*, 1975, vol. 4, strana 248-270.

[16] SNEMI3D. *ISBI 2013 challenge: 3D segmentation of neurites in EM images* [online]. 2013, posledná aktualizácia 2013 [cit. 20. 4. 2013]. Dostupné z URL: <`http://brainiac2.mit.edu/SNEMI3D/home`>.

[17] TUN-WEN, P. Boundary-constrained morphological skeleton minimization and skeleton reconstruction. *IEEE Transactions on Patters Analysis and Machine Intelligence*, 1994, vol. 16, no. 2, strana 201-208.

[18] FLAVIO, R. Thresholding Using the ISODATA Clustering Algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, 1980, vol. 10, no. 11, strana 771-774.

[19] HOFFMAN, R., et. al. Kernel methods in machine learning. *The annals of statistics*, 2008, vol. 36, no. 3, strana 1171-1220.

[20] JAIN, V., et. al. Boundary Learning by Optimization with Topological Constraints. *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, strana 2488-2495.

ZOZNAM PRÍLOH

A Príloha A **66**
A.1 Obsah CD 66

A PRÍLOHA A

A.1 Obsah CD

Príloha obsahuje elektronickú verziu práce, zdrojové súbory napísané v jazyku Java a implementované v prostredí Eclipse a je odovzdaná elektronicky. Obsahuje tieto súbory:

- *main.pdf* - elektronická verzia práce,
- *GaussianBlurFilter3D.java* - navrhnutá metóda rotácie a rozostrenia dát v 3D,
- *Rotation.java* - operátor rozostrenia pre prostredie RapidMiner,
- *AverageFilter3D.java* - priemerovanie dát v 3D,
- *AverageFilterOperator.java* - operátor priemerovania pre prostredie RapidMiner,
- *createMask.java* - metóda vytvorenia binárnej masky,
- *MedianFilter3D* - stredná hodnota dát v 3D,
- *MedianFilter3DOperator* - mediánový operátor pre prostredie RapidMiner,
- *SobelFilter3D.java* - hranová detekcia v 3D,
- *SobelFilterOperator.java* - operátor detekcie hrán pre prostredie RapidMiner,
- *Write3D.java* - ukladanie jednotlivých transformovaných slicov.

Ďalej príloha obsahuje potrebné súčasti k spusteniu programu RapidMiner a spusteniu rozšírenia.

- *obrazky.zip* - trénovacie a testovacie snímky,
- *procesyRM.zip* - procesy 2D a 3D segmentácie,
- *rapidminer-IMMI3D-1.0.0.jar* - rozšírenie s vytvorenými operátormi do programu RapidMiner,
- *rapidminer-ImageMiner-1.3.3.jar* - rozšírenie ostatných ľudí podieľajúcich sa na vývoji,
- *RapidMinerUnuk.zip* - zdrojové kódy programu RapidMiner so spustiteľnou verziou bez inštalácie,
- *ImageProcessingExtension.zip* - zdrojové kódy rozšírenia,
- *IMMI3DExtension.zip* - zdrojové kódy rozšírenia.

Inštalácia rozšírení spočíva v nakopírovaní do zložiek RapidMineru *lib**plugins*.