

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5220-47912

Bc. Marek Sobôtka

**SEGMENTÁCIA OBRAZU S POUŽITÍM
METÓD DISTRIBUOVANÝCH
AGENTOV PRE ZÍSKAVANIE
INFORMÁCIE Z LEKÁRSKÝCH
SNÍMKOV**

Diplomová práca

Študijný program: Softvérové inžinierstvo
Študijný odbor: 9.2.5 Softvérové inžinierstvo
Miesto vypracovania: Ústav informatiky a softvérového inžinierstva, FIIT STU
Bratislava
Vedúci práce: Ing. Vanda Benešová, PhD.

máj 2012

Zadanie diplomovej práce

Meno študenta: **Bc. Sobôtka Marek**

Študijný program: Softvérové inžinierstvo

Študijný odbor: Softvérové inžinierstvo

Názov práce: **Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskeho snímkov**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážete, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie, zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riadte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti alebo o priebežné výsledky Vášho riešenia alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnete zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav aplikovanej informatiky FIIT STU v Bratislave

Vedúci práce: **Ing. Vanda Benešová, PhD.**

Termíny odovzdania:

podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti:
home > Informácie o > štúdiu > organizácia štúdia > diplomový projekt

V Bratislave dňa 14. 2. 2011



prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky a softvérového
inžinierstva

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce¹

Študent

Meno, priezvisko, tituly:	Marek Sobôtka, Bc.
Študijný program:	Softvérové inžinierstvo
Kontakt:	m.sobotka@azet.sk

Výskumník:

Meno, priezvisko, tituly:	Vanda Benešová, Ing. PhD.
----------------------------------	---------------------------

Projekt:

Názov:	Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskeho snímkov
Miesto vypracovania:	Ústav aplikovanej informatiky, FIIT STU, Bratislava
Oblasť problematiky²:	Digitálne spracovanie obrazu

Text zadania

Segmentácia obrazu pre rôzne aplikácie je vo všeobecnosti zložitá úloha a pre jej riešenie sú spravidla potrebné sofistikované algoritmy. V literatúre sa v poslednej dobe objavujú rôzne metódy segmentácie obrazu používajúce systém multi-agentov pre aplikácie segmentácie obrazu predovšetkým v oblasti medicíny. Analyzujte stav problemovej oblasti so zameraním na využitie segmentácie v oblasti medicíny. Rešeršujte publikované metódy segmentácie obrazu s použitím metód distribuovaných agentov a vyberte algoritmus vhodný pre segmentáciu objektov na dvojrozmerných snímkoch z lekárskej praxe ako sú napr. röntgenové snímky alebo snímky získané metódou magnetickej rezonancie (MRI). Implementujte vybranú metódu segmentácie pracujúcej so systémom multi-agentov v jazyku C++. Navrhajte možné vylepšenia pre danú metódu a následne ju optimalizujte pre danú aplikáciu. Vyhodnoťte robustnosť a efektívnosť implementovanej metódy na vhodnom súbore testovacích obrazov. Sledujte závislosti implementovanej metódy od jej vstupných parametrov a vyhodnoťte ich. Porovnajte úspešnosť ako i časovú náročnosť implementácie porovnaním so základnými segmentačnými metódami ako napr. segmentácia v pyramíde.

150-200 slov, ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

¹ Veľkosť jednotlivých polí pre vyplňanie nemožno meniť. Vytlačiť obojstranne na jeden list papiera.

² Identifikácia oblasti v rámci odboru štúdia, na ktorú sa projekt primárne viaže

Literatúra

- Liang, J., Xu, Z., Mao, D., Cheng, X.: Multi-agent based evolutionary algorithm in medicine image segmentation. In: *Journal of Communication and Computer*, vol. 6, June 2009, no. 6 (serial no. 55), ISSN 1548-7709, 31-35.
- Chitsaz, M., Seng, W.C.: A Multi-agent System Approach for Medical Image Segmentation. In: *Future Computer and Communication, 2009. ICFCC 2009. International Conference on*. Washington DC: IEEE Computer Society, 2009, 408-411.
- Shariatpanahi, H.F., Batmanghelich, N., Kermani, A.R.M., Ahmadabadi, M.N., Soltanian-Zadeh, H.: Distributed Behavior-based Multi-agent System for Automatic Segmentation of Brain MR Images. In: *Neural Networks, 2006. IJCNN '06. International Joint Conference on*. Washington DC: IEEE Computer Society, 2006, 4535-4542.

2-3 vedecké zdroje, každý na samostatnom riadku a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uvedte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval Bc. Marek Sobôtka, konzultovala a osvojila si ho Ing. Vanda Benešová PhD. a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave

dňa

18. 1. 2011



Podpis študenta



Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / nie³

Dňa: 31. 1. 2011



Podpis garanta predmetov

³ Nehodiace sa prečiarknite

Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Softvérové inžinierstvo

Autor: Bc. Marek Sobôtka

Diplomová práca: **Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskeho snímkov**

Vedúci diplomovej práce: Ing. Vanda Benešová, PhD.

máj, 2012

V dnešnej dobe každodenne pracujeme s množstvom informácií. Vznikajú rôzne systémy na ich spracovanie, od manuálnych až po plne automatické. Jedným z typov informácií sú obrazové informácie. Denne sa na celom svete vytvorí nespočetné množstvo fotografií a natočí sa veľa hodín videa. Pri takom množstve záznamov prirodzene vzniká potreba ich rozpoznávania, triedenia a ďalšieho spracovania. Jednou z oblastí, kde je nutné spracúvať obrazové dáta, je medicína. Používajú sa tu viaceré zariadenia, ktoré produkujú rôzne typy výstupov. K typickým príkladom patria röntgenové snímky alebo snímky získané metódou magnetickej rezonancie (MRI). Už v súčasnosti existuje nesmierne množstvo systémov na ich spracovanie. Nedielnou súčasťou spracovania je segmentácia. Práca opisuje segmentáciu obrazu a jej základné metódy, analýzu multi-agentových systémom a niektoré vybrané existujúce riešenia multi-agentových systémov z medicínskej oblasti. Multi-agentový systém, ktorý skúmame, je založený na kombinovaní viacerých základných obrazových informácií. Využíva sa centralizovaná koordinácia agentov. Agenty sú riadené moderátorom.

Annotation

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Software Engineering

Author: Bc. Marek Sobôtka

Diploma thesis: **Image segmentation using methods of distributed agents to obtain information from medical images**

Supervisor: Ing. Vanda Benešová, PhD.

2012, May

Nowadays, a large amount of information is worked with every day. Different systems for data processing have been created. These systems can be manual or fully automatic. One type of data being processed is image data. Daily, countless numbers of photos are created and hours of videos are recorded around the world. With so many records there naturally arises the need for its recognition, sorting and further processing. Medicine is one area where it is necessary to process image data. Several devices are being used here that produce different types of output. Typical examples include X-rays or pictures obtained by the magnetic resonance imaging (MRI). There is already a vast array of systems for image processing. An integral part of image processing is image segmentation. This work describes the image segmentation and its basic methods, analysis of multi-agent system and some selected existing solutions of multi-agent systems in the medical field. Multi-agent system which we are researching combines several types of basic image information. Centralized coordination of agents is used. Agents are controlled by a moderator.

Čestne prehlasujem, že som túto prácu vypracoval samostatne, s použitím uvedenej literatúry,
konzultácií s vedúcim diplomovej práce a znalostí, získaných počas štúdia.

Bratislava, máj 2012

.....

podpis

Ďakujem Ing. Vande Benešovej PhD. za jej ústretový prístup, cenné rady, pripomienky a odborné vedenie pri vypracovávaní diplomovej práce.

Obsah

1	Úvod.....	1
2	Segmentácia obrazu.....	2
2.1	Prahové metódy.....	2
2.2	Metódy založené na detekcii hrán.....	3
2.3	Regionálne metódy.....	4
2.4	Metódy zhlukovania.....	5
2.5	Iné metódy.....	7
3	Multi-agentové systémy	8
4	Existujúce riešenia.....	11
4.1	Multi-agentový systém autorov Chitsaz a Seng [3]	11
4.2	Multi-agentový systém autorov Liang, Xu, Mao a Cheng [9]	13
4.3	Multi-agentový systém autorov Shariatpanahi, Batmanghelich, Kermani, Ahmadabadi a Soltanian-Zadeh [10]	14
4.4	Multi-agentový systém autorov Idir, Merouani a Tlili [11]	16
4.5	Multi-agentový systém autorov Duchesnay, Montois, Jacquélet a Kinie [12].....	18
5	Špecifikácia prototypu.....	19
6	Návrh riešenia	21
6.1	Identifikácia hráčov systému.....	21
6.2	Interný operátor – moderátor.....	21
6.2.1	Predspracovanie	22
6.2.2	Inicializácia dát	23
6.2.3	Inicializácia agentov.....	23
6.2.4	Beh agentov.....	24
6.2.5	Post spracovanie.....	24
6.2.6	Vyfarbenie regiónov.....	27
6.3	Interný operátor – agent	27
6.3.1	Vykonaj krok pre každé dieťa agenta.....	29
6.3.2	Aktualizuj dáta agenta.....	29
6.3.3	Označ agenta za neaktívneho	29
6.3.4	Vypočítaj počet možných pohybov	29
6.3.5	Presuň sa na miesto možného pohybu / Presuň sa na miesto nasledovného pixlu.....	30
6.3.6	Vytvor agenta pre každé miesto možného pohybu.....	30
6.3.7	Prepni agenta do spájacieho módu	30
6.3.8	Označ aktuálny pixel za výsledný	30

6.3.9	Označ všetky navštívené pixle od posledného rozdelenia za nevýsledné	30
6.3.10	Získaj pozíciu nasledovného pixlu	31
6.4	Interakcia používateľa so systémom	33
7	Implementácia riešenia	34
7.1	Funkcionalita moderátora	35
7.1.1	Predspracovanie	35
7.1.2	Inicializácia dát	35
7.1.3	Inicializácia agentov	36
7.1.4	Beh agentov	36
7.1.5	Post spracovanie	37
7.1.6	Vyfarbenie regiónov	37
7.2	Funkcionalita agenta	37
7.2.1	Spájanie na základe vzdialenosti	37
7.2.2	Model aktívnych kontúr	37
7.2.3	Kombinácia vlastností obrazu	37
7.2.4	Použitie rovnice pre rozhodovanie agentov	38
7.3	Funkcionalita externého používateľa	38
7.3.1	Zmena nastavenia segmentácie	38
7.3.2	Nastavenie vstupných dát	39
7.3.3	Segmentovanie jedného obrazu z možnosťou rekonfigurácie	39
7.3.4	Segmentovanie dátovej množiny obrazov s vyhodnotením	40
7.3.5	Zobrazenie vizualizácie práce pre agentov	40
7.3.6	Získanie medzivýsledkov	40
8	Overenie riešenia	42
8.1	Opis dátovej množiny obrazov	42
8.2	Spôsob vyhodnotenia	43
8.3	Nastavenie parametrov	43
8.4	Porovnanie výsledkov voči iným metódam	44
8.5	Výstupy systému	45
8.6	Vyhodnotenie časovej náročnosti	45
9	Zhodnotenie	47
10	Zoznam použitej literatúry	49
A	Inšalačná príručka	51
B	Používateľská príručka	52
B.1	Spracovanie jedného obrazu	52
B.2	Dávkové spracovanie viacerých obrazov	53
B.3	Nastavenia segmentácie	54

C	Implementačná dokumentácia – diagramy tried	56
D	Obsah CD nosiča.....	58
E	Príspevok na konferenciu IIT.SRC 2012	59

Zoznam obrázkov

Obr. 2.1 Segmentácia prahovaním [6]	3
Obr. 2.2 Hranový obraz získaný pomocou Canny detektora [6]	4
Obr. 2.3 Postupná minimalizácia energie hada pri segmentácii modelom aktívnych kontúr	4
Obr. 2.4 Rozdelenie obrazu do viacerých regiónov [4]	5
Obr. 2.5 Kroky segmentácie metódou K-Means [6]	6
Obr. 2.6 Posuny vyhľadávacích okien pri segmentácii Mean-Shift	6
Obr. 2.7 Segmentácia rozvodnením [4]	7
Obr. 3.1 Interná štruktúra agenta [3]	10
Obr. 4.1 Architektúra multi-agentového systému [3]	12
Obr. 4.2 Ukážka segmentácie multi-agentového systému [3]	12
Obr. 4.3 Ukážka segmentácie multi-agentového systému [9]	13
Obr. 4.4 Prispôsobovanie modelu objektu [10]	14
Obr. 4.5 Architektúra agenta v multi-agentovom systéme [10]	15
Obr. 4.6 Ukážka segmentácie multi-agentového systému [10]	16
Obr. 4.7 Ukážka predspracovania multi-agentového systému [11]	17
Obr. 4.8 Ukážka segmentácie multi-agentového systému [12]	18
Obr. 5.1 Ukážka predspracovania prototypu	19
Obr. 5.2 Príklady akcií prototypu	20
Obr. 6.1 Diagram aktivít segmentácie obrazu	22
Obr. 6.2 Kategorizácia pixlov podľa počtu hrán v okolí	24
Obr. 6.3 Ukážka vytvorenia kostry [15]	25
Obr. 6.4 Inicializácia kruhov a ich uzlov [16]	25
Obr. 6.5 Proces rozrastania kruhov [16]	26
Obr. 6.6 Korekcia vyčnievajúcich uzlov [16]	26
Obr. 6.7 Rovnica interpolácie [16]	26
Obr. 6.8 Diagram aktivít kroku agenta	28
Obr. 6.9 Vymazanie krátkych hrán z obrazu	31
Obr. 6.10 Rovnica výpočtu skóre pixlu	31
Obr. 6.11 Rovnica výpočtu kontrolovaného uhlu	32
Obr. 6.12 Rovnica výpočtu kontrolovaného regiónu	32
Obr. 6.13 Rovnica výpočtu penalty	32
Obr. 6.14 Diagram prípadov použitia externého používateľa	33
Obr. 7.1 Diagram komponentov	34
Obr. 7.2 Ukážka získateľných výstupov	41

Obr. 8.1 Graf výsledkov segmentácie rôznych metód	44
Obr. 8.2 Segmentácia obrazu s vyšším rozlíšením.....	45
Obr. B.1 Úvodné okno programu	52
Obr. B.2 Okno s výsledkami pri spracovaní jedného obrazu	53
Obr. B.3 Okno s výsledkami pri dávkovom spracovaní viacerých obrazov	54
Obr. C.1 Diagram tried segmentačnej knižnice	56
Obr. C.2 Diagram tried grafického rozhrania.....	57
Obr. D.1 Logická stromová štruktúra obsahu CD nosiča.....	58

Zoznam tabuliek

Tab. 8.1 Počet snímkov jednotlivých dátových množín	42
Tab. 8.2 Vyhodnotenie segmentácie v závislosti od parametrov	43
Tab. 8.3 Porovnanie výsledkov voči iným metódam	44

Register použitých skratiek

CT	Computed Tomography – rádiologická vyšetrovacia metóda pre zobrazenie vnútra ľudského tela za pomoci využitia röntgenu
MRI	Magnetic Resonance Imaging – rádiologická vyšetrovacia metóda pre zobrazenie vnútra ľudského tela za pomoci využitia silného homogénneho magnetického poľa
OpenCV	Open Source Computer Vision – knižnica zoskupujúca funkcie a metódy na prácu s obrazmi v rámci počítačového videnia

1 Úvod

V dnešnej dobe každodenne pracujeme s množstvom informácií. Vznikajú rôzne systémy na ich spracovanie, od manuálnych až po plne automatické. Jedným z typov informácií sú obrazové informácie. Denne sa na celom svete vytvorí nespočetné množstvo fotografií a natočí sa veľa hodín videa. Pri takom množstve záznamov prirodzene vzniká potreba ich rozpoznávania, triedenia a ďalšieho spracovania.

Jednou z oblastí, kde je nutné spracúvať obrazové dáta, je medicína. Používajú sa tu viaceré zariadenia, ktoré produkujú rôzne typy výstupov. K typickým príkladom patria röntgenové snímky alebo snímky získané metódou magnetickej rezonancie (MRI). Už v súčasnosti existuje nesmierne množstvo systémov na ich spracovanie. Nedielnou súčasťou spracovania je segmentácia. Existujú rôzne metódy a dosahujú rozdielne výsledky.

Cieľom tejto práce je preskúmať možnosti použitia multi-agentových systémov. Následne získané poznatky pri bližšom štúdiu aplikovať pri výbere správnej metódy distribuovaných agentov na segmentáciu snímok z lekárskej oblasti. V ďalšom kroku navrhnuť a implementovať optimalizovanú metódu pre konkrétne použitie. Nakoniec overiť riešenie na vhodnom súbore testovacích obrazových dát a porovnať, či nové riešenie poskytuje zlepšenie oproti základným metódam segmentácie a voči iným multi-agentovým systémom.

Práca je rozdelená do niekoľkých kapitol. Zahŕňa opis segmentácie obrazu a jej základné metódy (kapitola 2), opis multi-agentových systémov (kapitola 3), venuje sa niektorým vybraným existujúcim riešeniam multi-agentových systémov z medicínskej oblasti (kapitola 4), špecifikáciu prototypu (kapitola 5), návrh riešenia (kapitola 6), implementáciu riešenia (kapitola 7) a overenie riešenia (kapitola 8).

2 Segmentácia obrazu

Segmentácia obrazu predstavuje v oblasti digitálneho spracovania obrazu dlhodobú výzvu. Jej cieľom je rozdeliť vstupný obraz na niekoľko disjunktných homogénnych oblastí. Jednotlivé oblasti obsahujú také body obrazu, ktoré majú podobnú intenzitu, farbu, textúru alebo iný aspekt v závislosti od toho, čo chceme segmentáciou dosiahnuť [1, 2, 3]. Takto získané oblasti možno ďalej spracúvať na úrovni rozpoznávania objektov.

Segmentácia obrazu predstavuje jeden zo základných komponentov analýzy digitálnych obrazov. Je to zložitý proces, ktorý ovplyvňuje mnoho faktorov, medzi inými kvalita vstupného obrazu, kvalita zariadenia, ktorým bol obraz zaznamenaný, svetelné podmienky v akých bol obraz zaznamenaný a aj prostredie, kde bol obraz zaznamenaný. [1]

Jedná sa o využívanú metódu spracovania obrazu, o čom svedčí široké spektrum uplatnenia. Využitie nachádza v rôznych oblastiach od filtrácie obrazov s vysokým šumom v medicíne, cez lokalizovanie objektov na satelitných obrazoch, systémy automatického riadenia dopravy až po oblasť počítačového videnia. [2]

Vďaka svojmu uplatneniu bolo časom vyvinutých niekoľko všeobecných techník, ako ju dosiahnuť. Medzi základné techniky segmentácie patria [2]:

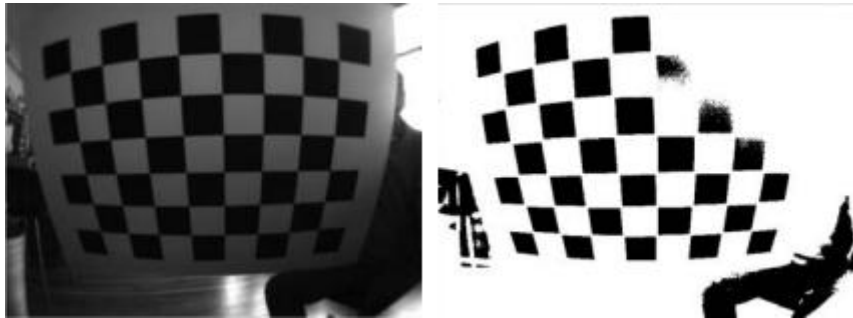
- prahové metódy,
- metódy založené na detekcii hrán,
- regionálne metódy,
- metódy zhľukovania.

Avšak segmentácia pre získanie polohy objektov v obraze je banálnym problémom, pretože je doménovo špecifická. To má za následok fakt, že neexistuje žiadne univerzálne riešenie, ktoré by bolo schopné pokryť uplatnenie segmentácie na všetky typy vstupných obrazov a pre všetky typy využitia [2, 3]. Preto je nutné techniky navzájom kombinovať s využitím doménových znalostí, aby bol dosiahnutý čo najlepší výsledok.

2.1 Prahové metódy

Prahové metódy (Thresholding) predstavujú úplne najjednoduchší spôsob segmentácie obrazu. Tieto metódy sú založené na delení bodov obrazu štandardne do dvoch skupín. Príslušnosť bodu do skupiny určuje stanovená prahová hodnota. V závislosti či je hodnota bodu menšia, alebo

väčšia ako prah sa bod zaradí do konkrétnej skupiny (Obr. 2.1). Ako deliaca hodnota sa môže použiť rôzne kritérium, najčastejšie sa však používa jas. [2, 4, 5]



Obr. 2.1 Segmentácia prahovaním [6]

Rozlišujeme dva druhy prahových metód [2, 4, 5]:

- *globálne prahovanie* – prahová hodnota sa určí globálne pre celý obraz,
- *lokálne (adaptívne) prahovanie* – prahová hodnota sa určuje zvlášť pre rôzne časti obrazu na základe hodnôt v blízkom okolí skúmaného bodu.

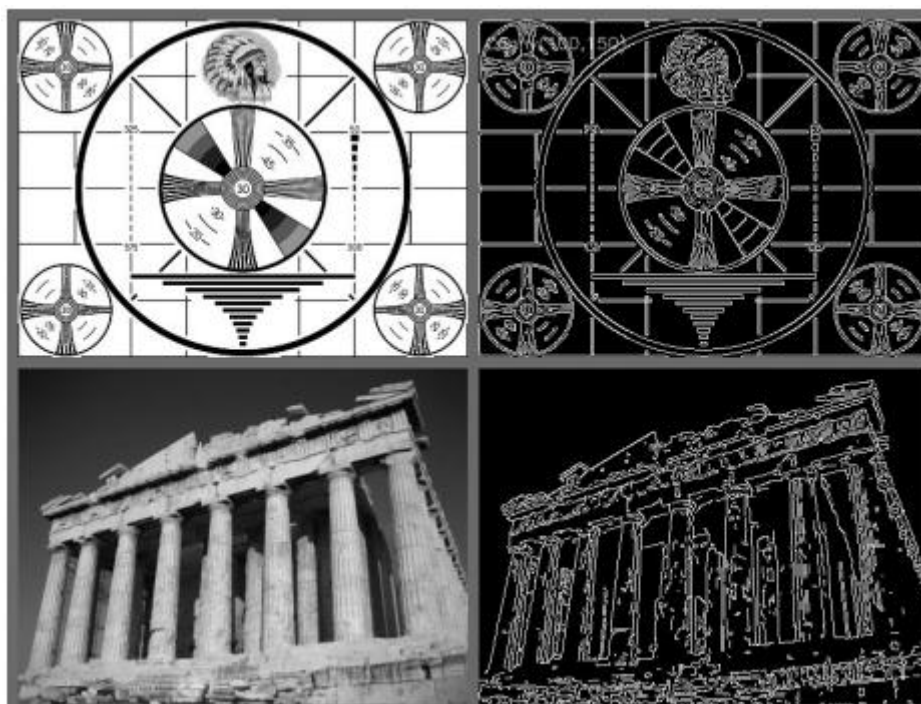
Určiť prahovú metódu je možné na základe hodnôt jasu v obraze alebo z jeho histogramu. Existujú viaceré metódy ako určiť prahovú hodnotu automaticky, medzi ne patrí metóda p-podielu, metóda analýzy tvaru histogramu a optimálne prahovanie – Otsuova metóda [4, 5].

Výhodou prahových metód je ich jednoduchosť, vďaka čomu sú výpočtovo nenáročné a rýchle. Avšak jednoduchosť je aj ich nevýhodou, pretože nimi dokážeme segmentovať len veľmi malú množinu obrazov a zložitejšie obrazy už pre ne predstavujú problém.

2.2 Metódy založené na detekcii hrán

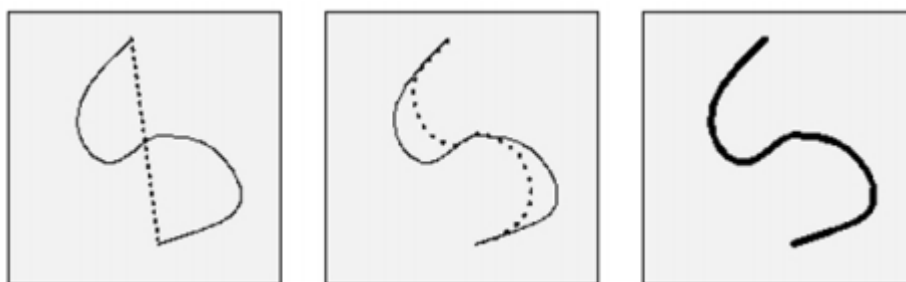
Metódy založené na detekcii hrán potrebujú na vstupe hranový obraz. Hrany v obraze predstavujú náhle zmeny hodnoty jasu. Hranový obraz teda chápeme ako množinu bodov miest prudkých zmien jasu. Pre získanie týchto bodov sa využíva prvá alebo druhá derivácia intenzity jasu. Na tomto princípe funguje Robertsov, Prewittov, Sobelov a Laplacov operátor. [2, 5]

Prvá metóda predstavuje jednoduché extrahovanie ohraničených objektov v obraze. Keby sme mali k dispozícii perfektný hranový obraz, tak by táto metóda fungovala skvele. Avšak v skutočnosti to nie je také jednoduché, pretože hrany vo vstupnom obraze často nie sú spojité, niektoré tam môžu úplne chýbať a niektoré tam naopak môžu byť navyše. Preto je často pri použití tejto metódy nevyhnutné predspracovanie a dodatočné spracovanie. Ukážku hranového obrazu je vidieť na Obr. 2.2.



Obr. 2.2 Hranový obraz získaný pomocou Canny detektora [6]

Druhou metódou je model aktívnych kontúr, tiež zvaná ako adaptívne hady (Snake algorithm). Hady predstavujú dvojdimenzionálne zovšeobecnenú splývajúcu krivku minimalizujúcu vynaloženú energiu [7]. Energiu hada môžeme chápať ako pohyb obrazom cez jeho definované vlastnosti, často sú nimi práve hrany. Energia závisí od tvaru a pozície, lokálne minimum vynaloženej energie predstavuje požadovanú vlastnosť obrazu. Postupné minimalizovanie energie zachytáva Obr. 2.3. Využitím tejto metódy je možné napríklad vylepšiť hranový obraz o uzatvorenie a spojenie hrán.



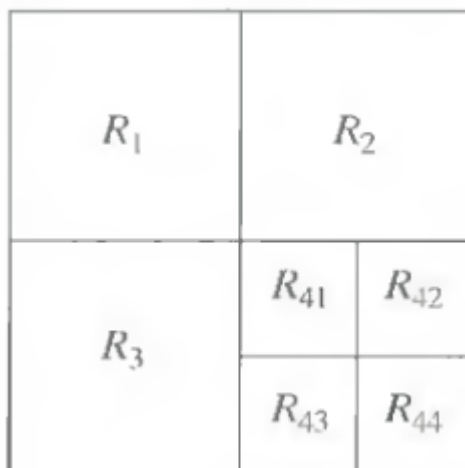
Obr. 2.3 Postupná minimalizácia energie hada pri segmentácii modelom aktívnych kontúr

2.3 Regionálne metódy

Regionálne metódy pracujú s obrazom ako s množinou oblastí, kde každá oblasť reprezentuje požadovanú vlastnosť alebo objekt.

Prvá metóda je metóda rozrastajúcich oblastí (Growing regions). Funguje na princípe, kde sa oblasti zväčšujú na základe definovaných kritérií. Na začiatku sú vybrané štartovacie body, ktoré spĺňajú požadované podmienky. V ďalšom kroku sa skúma okolie týchto bodov a oblasti sa rozrastajú pridávaním bodov, ktoré spĺňajú kritéria. Keď už nie je možné pridať žiaden ďalší bod, región je celý. Pre túto metódu je nesmierne dôležité určiť správne štartovacie body, v opačnom prípade výsledok bude s veľkou pravdepodobnosťou odlišný od očakávania. Výsledné oblasti môžu byť spojené aj nespojité v závislosti od definovaných kritérií. [4, 5]

Metóda rozdeľovania a spájania oblastí (Split & Merge) funguje na opačnej filozofii. Začína s celým obrazom, ktorý sa rekurzívne delí na podoblasti, najčastejšie na štyri (Obr. 2.4). Delenie oblasti sa zastaví v prípade ak oblasť spĺňa definované kritérium homogenity. Ako kritérium môže byť použitá napríklad stredná hodnota jasu v oblasti, rozptyl hodnôt jasu v oblasti alebo rozdiel vo farbe. Po dekompozícii nám vznikne množina regiónov rozdielnych veľkostí, z ktorých každý región je homogénny vzhľadom na definované kritérium. Avšak môže sa stať, že delením sa rozdelili aj homogénne regióny, preto v druhej fáze sa spájajú susedné homogénne oblasti dokopy. [4, 5]



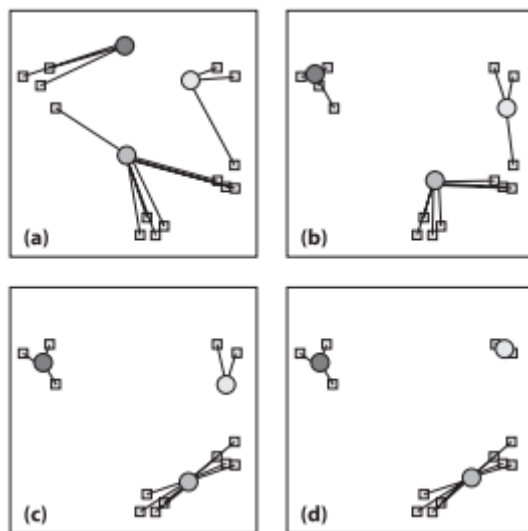
Obr. 2.4 Rozdelenie obrazu do viacerých regiónov [4]

2.4 Metódy zhlukovania

Metódy zhlukovania sa pozerajú na segmentáciu ako na klasifikačný problém. Každý obraz pozostáva z istého počtu zhlukov a každý bod obrazu patrí práve do jedného z nich. Každý zhluk bodov reprezentuje rozdielnu vlastnosť obrazu.

Prvou metódou je K-Means. Na začiatku sa zvolia centrá zhlukov, ktorých počet je zhodný s počtom výsledných zhlukov. Následne sa prechádza množina bodov obrazu, každý bod sa zaradi do zhluku, ktorý je z hľadiska definovaného kritéria najpodobnejší a prepočíta sa

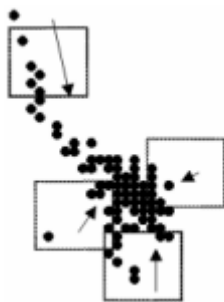
centrálna hodnota zhluku. Ako kritérium môže poslúžiť napríklad priemerná jas alebo stredná hodnota jas. V takomto prípade sa jedná len o jemnú modifikáciu prahovej metódy. Avšak ako kritérium môže poslúžiť aj kombinácia iných príznakov, napríklad farba, textúra, pozícia. Kroky tejto metódy segmentácie zachytáva Obr. 2.5, štvorčeky reprezentujú body obrazu a kruhy reprezentujú centrá zhlukov.



Obr. 2.5 Kroky segmentácie metódou K-Means [6]

Výhodou K-Means metódy je časová efektivita riešenia, pretože sa jedná o nenáročnú metódu. Avšak problémom je, že nedokáže zaručiť nájdenie najlepšieho riešenia, pretože je závislá od určenia počiatočných centier. Druhým problémom tejto metódy je, že zvyčajne nevieme dopredu určiť počet zhlukov, do ktorých chceme obraz segmentovať. [2, 6]

Druhou metódou je Mean-Shift. Opäť sa pracuje so zhlukmi bodov. Na začiatku sa zvolí veľkosť vyhľadávacích okien a ich štartovacie pozície v rámci obrazu. V každom z týchto okien sa určí stredná hodnota sledovanej veličiny (jasu, farby, ...) a centrum vyhľadávacieho okna sa presunie na jeho pozíciu (Obr. 2.6). Toto sa opakuje pre každé vyhľadávacie okno v obraze až pokiaľ ich pozícia neskonverguje. Následne sa spoja zhluky, ktorých konečná pozícia je rovnaká. Všetky časti obrazu, ktoré opísalo jedno vyhľadávacie okno cestou, patria do jedného zhluku, kde každý z týchto zhlukov opisuje sledovanú vlastnosť alebo objekt. [6, 7]

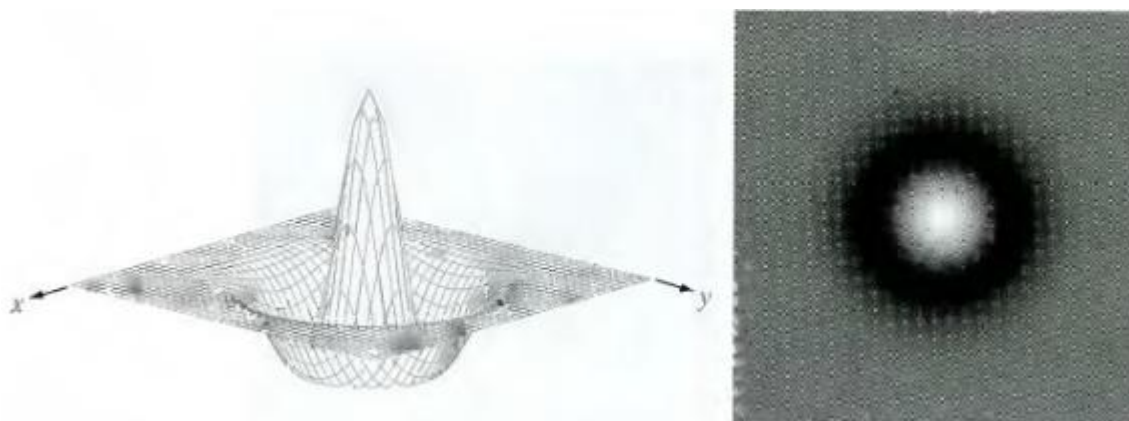


Obr. 2.6 Posuny vyhľadávacích okien pri segmentácii Mean-Shift

2.5 Iné metódy

Existujú aj iné typy metód segmentácie, ktoré sa nedajú zaradiť do žiadnej z predchádzajúcich skupín alebo kombinujú ich prístupy a vytvárajú hybridné metódy.

Prvou metódou je segmentácia rozvodnením (Watershed segmentation). Táto metóda chápe obraz ako množinu kopcov a údolí, kde hodnota jasů určuje nadmorskú výšku (Obr. 2.7). Následne sa sleduje pre zvolené body, ako by sa v nich rozliala voda a tým sa určujú segmentované oblasti. Táto metóda je náročná na aplikáciu ak požadujeme, aby podávala očakávané výsledky. Často dochádza k presegmentovaniu obrazu, keď je v obraze veľa pohorí alebo naopak ak sa spájajú aj rozdielne oblasti, keď sa voda rozlieva do viacerých údolí. Preto je často pri použití tejto metódy nevyhnutné predspracovanie obrazu. [4, 6]



Obr. 2.7 Segmentácia rozvodnením [4]

Druhou odlišnou metódou je segmentácia na základe šablóny (Template segmentation). Túto metódu možno použiť na hľadanie špecifických vzorov v obraze alebo lokalizáciu objektov so známym vzhľadom. Metóda spočíva vo vyhodnotení kritéria zhody pre každú polohu a otočenie vzoru v obraze. Miesta, kde kritérium zhody presahuje stanovený prah reprezentujú umiestnenie vzoru v obraze. Kritérium zhody môže byť definované rozdielne, najčastejšie sa používa korelácia medzi šablónou a obrazom na danej pozícii [2]. Avšak táto metóda je nesmierne časovo náročná ak sa má prehľadávať pre každé otočenie a škálu, preto je vhodné ju napríklad využiť len na overenie v pár pozíciách, ktoré sa určia inou metódou. [6]

3 Multi-agentové systémy

Multi-agentové systémy predstavujú spoločenstvá agentov, ktoré vykonávajú nejaké činnosti podľa definovaných pravidiel alebo svojej umelej inteligencie za účelom dosiahnutia spoločného cieľa. Jednotkou týchto systémov je agent.

„Agent je skutočný alebo virtuálny objekt, umiestnený v prostredí, v ktorom môže konať, vnímať ho a čiastočne si ho predstavovať, je schopný komunikovať s ostatnými agentmi a ktorý ma autonómne správanie, ktoré vychádza z jeho pozorovaní, jeho znalostí a interakcie s ostatnými agentmi“ [8].

Z uvedenej definície agenta vyplýva niekoľko princípov ich existencie [8]:

- *umiestnenie* – agent musí byť umiestnený v prostredí, ktoré ovplyvňuje jeho konanie,
- *telesnosť* – agent musí vedieť vnímať a ovplyvňovať svoje prostredie,
- *inteligencia* – inteligencia agenta je výsledkom jeho stretu s okolím a jeho vhodnými reakciami na podnety z prostredia,
- *súčinnosť* – agent ovplyvňuje ostatných agentov a reaguje na ich správanie,
- *autonómnosť* – agent vykonáva akcie na základe vlastného rozhodnutia.

V ponímaní problematiky segmentácie si pod agentmi predstavujeme akýchsi pracovníkov, ktorí pracujú nad obrazovými dátami. Ich vnímanie a predstava o prostredí vychádza z vlastností obrazu ako sú jas, farba, hrany a podobne. Pod umiestnením agenta v prostredí si môžeme predstaviť jeho pozíciu na určitej šírke a výške. Telesnosť agenta je zabezpečená tým, že agent bude mať dostupné vlastnosti o obraze a bude schopný ich modifikovať. Agent sa rozhoduje na základe dostupných pravidiel segmentácie, z ktorých si inteligentne vyberie najvhodnejšie pre danú situáciu buď na základe svojho rozhodnutia, alebo rozhodnutia z externého prostredia. Súčinnosť sa dá zabezpečiť prácou nad spoločnými dátami a tým, že agent zmení nejaké parametre obrazu, nové veličiny budú prístupné aj pre ostatných agentov.

Agentov môžeme klasifikovať z pohľadu prostredia v ktorom pracujú na [8]:

- *programové (softvérové)* – umiestnenie agenta je vo virtuálnom svete,
- *robotické (hardvérové)* – umiestnenie agenta je v reálnom svete.

Nás budú ďalej zaujímať iba programové agenty. Podľa mechanizmu, ktorý zabezpečuje ich inteligenciu ich môžeme ďalej deliť nasledovne [8]:

- *reaktívne agenty* – reagujú bezprostredne na vnemy z okolitého prostredia,

Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskych snímok

- *agenty bez pamäti* – ako vnem využíva komplexnú situáciu okolitého sveta,
- *agenty s pamäťou* – zvyčajne si pamätá aktuálny stav, v ktorom sa nachádza,
- *plánovo-uvažujúce agenty* – zvažujú možné akcie a vedia si vytvoriť plán činností,
 - *agent so zámerom* – má motiváciu dosiahnuť nejaký cieľ, tomu podriadiť svoju činnosť a plánuje si nasledovné akcie,
 - *sociálny agent* – rozširuje agenta so zámerom, má k dispozícii modely správania sa ostatných agentov a prispôsobuje svoje činnosti kontextu ostatných agentov.

Vlastnosti agentov sa nedajú definovať jednoznačne, avšak dajú sa definovať niektoré vlastnosti, ktoré by takéto agenty, ktoré sú reprezentované hardvérom alebo softvérom, mali spĺňať [3]:

- *autonómnosť* – schopnosť dosiahnuť svoj cieľ bez riadenia človekom,
- *sociálne schopnosti* – schopnosť komunikovať s inými agentmi alebo ľuďmi,
- *reaktívnosť* – schopnosť reagovať na zmeny v prostredí,
- *proaktívnosť* – schopnosť prevziať iniciatívu za účelom dosiahnutia cieľa,
- *robustnosť* – schopnosť zotaviť sa a učiť sa z chýb.

Riešenie segmentácie pomocou týchto nových metód umelej inteligencie prináša viacero výhod, ktoré sú viac-menej zrejmé. Poskytnú vyššiu efektívnosť a účinnosť vďaka faktu, že jednotlivým agentom dokážeme definovať pravidlá pre použitie viacerých základných metód segmentácie. Vďaka distribúcii spracovania medzi viacerých agentov sa dá využiť paralelné spracovanie a tým sa zvýši časová efektívnosť riešenia.

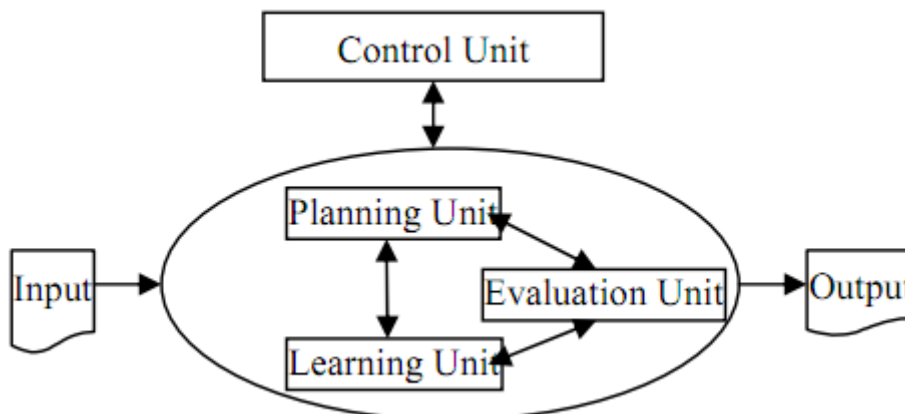
Ďalšie výhody takéhoto riešenia sú spomenuté v práci [10], avšak spomínam len tie, ktoré pokladám za najdôležitejšie:

- možnosť oddelenia algoritmu spracovania obrazu od riadiacej stratégie,
- spoľahlivosť – iné agenty v dobre organizovanom systéme dokážu opraviť agenta s nízkym výkonom a zlou znalosťou,
- schopnosť zaostrenia – niektoré úlohy nepotrebujú všetky znalosti,
- možnosť využiť paralelné a heterogénne štruktúry,
- možnosť distribuovanej a asynchrónnej implementácie, čo zníži čas vykonania.

Interná štruktúra agenta (Obr. 3.1) sa líši od spôsobu využitia multi-agentového systému a od zvolenej architektúry, avšak časťami agenta sú [3]:

- *vstupná jednotka* – prijíma vstupné dáta,
- *výstupná jednotka* – doručuje agentove výsledky,
- *plánovacia jednotka* – určuje stratégiu konania agenta,
- *výsledková jednotka* – kontroluje kvalitu vykonávaných akcií,

- *učiaca sa jednotka* – reprezentujúca znalosti pre adaptívne správanie sa,
- *riadiaca jednotka* – implementuje plán.



Obr. 3.1 Interná štruktúra agenta [3]

Pre multi-agentové systémy sú dôležité nielen vlastnosti jednotlivých agentov, ale aj vlastnosti systému ako celku. Systém sa môže skladať z rovnakých agentov. Takéto systémy sa využívajú na zvýšenie výkonu pomocou paralelného spracovania alebo na zvýšenie spoľahlivosti. Každý z agentov v takomto systéme je schopný riešiť všetky problémy. Druhou možnosťou sú systémy zložené z rozličných typov agentov. V takomto prípade dochádza k špecializácii, čo vedie k zvýšeniu efektivity práce. Analógiou je skutočný svet, kde každý človek pracuje v inej oblasti, je teda expertom v práci, ktorej sa venuje. Avšak, pre takéto multi-agentové systémy je dôležité zabezpečiť, aby každý problém vedel vyriešiť aspoň jeden typ agenta. [8]

Na základe toho, ako sú agenty koordinované a ako spolu kooperujú rozlišujeme dva prístupy [8]:

- *centralizovaný prístup* – koordinácia je preddefinovaná a riadi ju zhora zvláštny agent alebo skupina agentov,
- *distribuovaný prístup* – koordinácia je reaktívna a nie je riadená centrálnym agentom.

Oba prístupy majú svoje výhody a nevýhody. Centralizovaný spôsob je jednoduchší na implementáciu vďaka tomu, že riadenie je špeciálne oddelené v centrálnom agentovi alebo skupine agentov, ale dochádza k spomaleniu systému, pretože riadenie je nutné vykonávať na globálnej úrovni. Distribuovaný systém tento problém nemá, naopak je nutné vybaviť agentov spôsobom, ako sa navzájom budú koordinovať a komunikovať spolu, čo činí tento prístup zložitejší na implementáciu.

4 Existujúce riešenia

Segmentácia obrazu má stále priestor na zlepšenie. Dôvodom toho, že nie je stále na perfektnej úrovni, aj keď sa vyvíja už niekoľko desaťročí je viacero [3]. Najdôležitejším z nich je už spomínaný fakt, že riešenia pre segmentáciu obrazu sú problémovo závislé.

Obzvlášť badateľné je to v oblasti medicíny. Je rozdiel či segmentujeme prírodné snímky, snímky s dominujúcimi objektmi, alebo medicínske snímky. Pri analýze obrazov z medicínskej oblasti mnoho objektov (napr. ľudský mozog, srdce) nemá viditeľné hranice, obrazy sú prirodzene textúrované a nie sú jasné hrany medzi jednotlivými objektmi [9]. Ďalej medicínske snímky majú podobné odtiene šedi a podobné textúry medzi jednotlivými objektmi [3]. To spôsobuje menší rozptyl intenzity a použitých farieb v obraze, vďaka čomu je oveľa zložitejšie rozlíšiť obsiahnuté objekty v obraze.

V súčasnosti sa v medicíne aktívne využíva segmentácia obrazov napríklad v patológii, pri operácii mozgu [3, 10], pri štúdiu anatomických štruktúr, pri plánovaní liečenia alebo aj určovaní diagnózy [3]. Inde lekári spracúvajú snímky z röntgenu a MRI automaticky pomocou ľudskej inteligencie. Je snaha takéto snímky spracúvať automaticky pomocou počítača, aby napríklad dokázal samostatne určiť diagnózu pacienta. Vďaka tomu by lekár mohol konfrontovať svoj názor s vyhodnotením počítača. Aby to ale fungovalo, algoritmus použitý na segmentáciu takýchto snímok by musel dosahovať vyššiu spoľahlivosť. Problémom, ako to dosiahnuť, sa už zaoberajú viaceré práce, medzi inými aj [3, 9, 10, 11, 12]. Dosahujú pomerne slušné výsledky, avšak, stále je tu priestor pre zlepšenie.

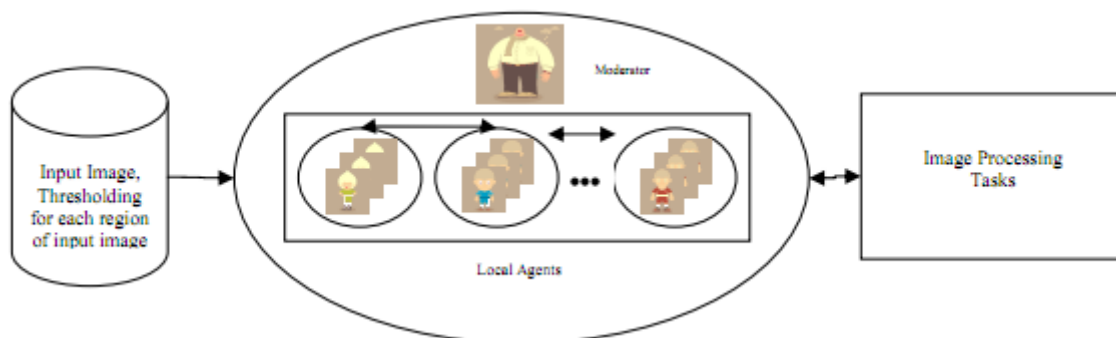
4.1 Multi-agentový systém autorov Chitsaz a Seng [3]

Prvým existujúcim riešením je multi-agentový systém na segmentáciu medicínskych snímok získaných pomocou počítačovej tomografie (CT). Tento systém je založený na centralizovanom prístupe. Existujú tu dva typy agentov:

- *agent moderátor* – iba jeden, dohliada a riadi celý proces segmentácie,
- *lokálne agenty* – je ich viacero, segmentujú lokálne časti obrazu.

Architektúra tohto systému (Obr. 4.1) je zložená z troch hlavných častí. Prvou sú vstupné materiály, teda vstupný obraz a prahové obrazy pre každú lokálnu časť obrazu. Druhou je samotné prostredie agentov tvorené spomínanými typmi agentov. Tretiu časť predstavujú procedúry na úpravu obrazu, ktoré agenty využívajú.

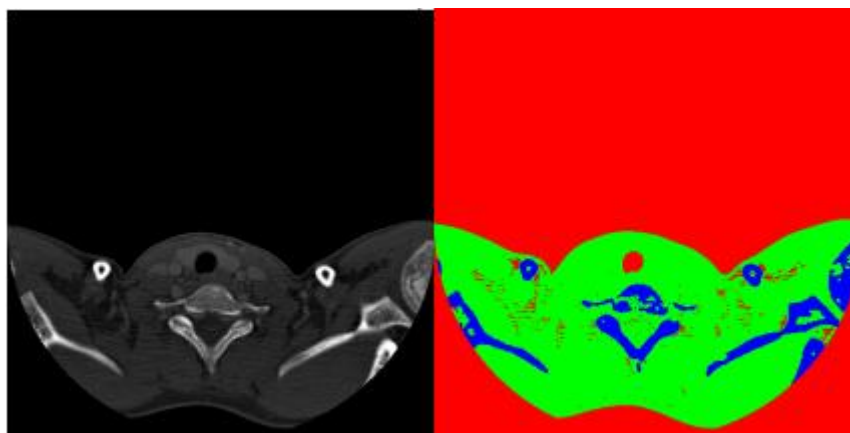
Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskeho snímku



Obr. 4.1 Architektúra multi-agentového systému [3]

Princíp fungovania systému nie je zložitý. Zakladá sa na zaradení každého obrazového bodu do určitej skupiny tak, ako to robia metódy zhlukovania. Najprv moderátor vytvorí a inicializuje lokálnych agentov v obraze. Každému z agentov je priradená určitá časť obrazu a jeho počiatočná poloha je v strede pridelenej časti. Na základe priori vedomostí (odhad prahovania danej oblasti), ktoré sú agentovi dodané označením každého obrazového bodu ich roztriedi do skupín. V prípade, že lokálny agent nevie rozhodnúť o príslušnosti obrazového bodu, ponechá ho neoznačený. Po istom čase moderátor skontroluje výsledky lokálnych agentov. Ak sa v obraze nachádzajú nezaradené body, vytvorí ďalšiu generáciu agentov, a tie umiestni práve do týchto bodov. Tento postup sa opakuje až pokiaľ nie sú všetky body roztriedené.

V práci sa uvádza, že presnosť segmentácie je na úrovni 90%. Ako porovnávacie kritérium bola zvolená odchýlka segmentácie pomocou multi-agentového systému oproti ľudskej segmentácii odborníkov. Výkonnosť dosahuje tento systém tiež dobré výsledky, celú dátovú sadu obrazov rozlíšenia 512x512 pixlov dokázal tento systém vysegmentovať v priemere za 15 sekúnd. Nanešťastie sa v práci neuvádza, koľko obrazov bolo v dátovej sade obsiahnutých. Obr. 4.2 zachytáva ukážku segmentácie tejto metódy, naľavo sa nachádza pôvodný obraz a napravo segmentovaný obraz.



Obr. 4.2 Ukážka segmentácie multi-agentového systému [3]

4.2 Multi-agentový systém autorov Liang, Xu, Mao a Cheng [9]

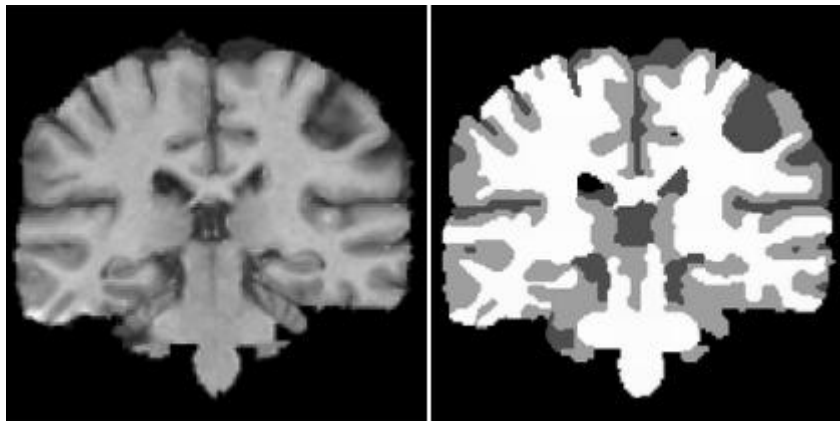
Druhým existujúcim riešením je multi-agentový systém na segmentáciu medicínskych snímok testovaný najmä na obrazoch MRI mozgu. Tento systém je založený na distribuovanom prístupe a existuje tu len jeden typ agenta.

Každý z agentov si pamätá spúšťacie podmienky agentových akcií. Tie sú ovplyvnené hodnotami ostatných agentov v okolí, v závislosti od vzájomnej vzdialenosti. Vlastnosti, ktoré sa pozorujú v rámci obrazu, sú vzájomný relatívny kontrast, priemerný kontrast v lokálnej oblasti a štandardná odchýlka kontrastu v lokálnej oblasti. Na základe týchto hodnôt agent posudzuje či je obrazový bod, na ktorom sa nachádza, zaujímavý (charakteristický) a podľa toho sa vykoná agentova akcia. V týchto bodoch sa nárazovo mení hodnota kontrastu a v podstate sa jedná o hrany v obraze.

Agent má k dispozícii štyri typy akcií:

- skopírovať sa,
- usadiť sa,
- pohnúť sa,
- zomrieť.

Princíp algoritmu opäť funguje na jednoduchej myšlienke. Na začiatku sú agenty náhodne inicializované vo vstupnom obraze. Agenty sa pohybujú v rámci obrazu, pokiaľ sa nedostanú na charakteristický bod obrazu. Vtedy sa agent skopíruje, jeden z nich sa usadí a ďalší pokračuje v hľadaní. Aby sa predišlo nekonečne trvajúcej hľadaniu, každý agent má definovanú dĺžku života, ktorá sa mu každým krokom skraca a kontroluje či agent nemá zomrieť. Keď zomrú všetky agenty, algoritmus končí a dostávame výsledný segmentovaný obraz.



Obr. 4.3 Ukážka segmentácie multi-agentového systému [9]

Výsledky tohto systému boli v práci porovnávané voči dvom ďalším metódam segmentácie. Segmentácia sa testovala na desiatich rozdielnych setoch s neznámym počtom obrazov menšieho rozmeru (181x217 pixlov). Ako kritérium si všímali citlivosť, špecifickosť a celkový výkon, pričom dosiahli lepšie výsledky ako zvyšné dve metódy. Obr. 4.3 zachytáva ukážku segmentácie tejto metódy, naľavo sa nachádza pôvodný obraz a napravo segmentovaný obraz.

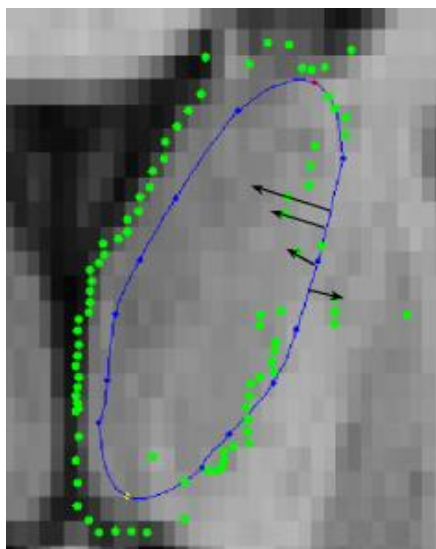
4.3 Multi-agentový systém autorov Shariatpanahi, Batmanghelich, Kermani, Ahmadabadi a Soltanian-Zadeh [10]

Tretím existujúcim riešením je multi-agentový systém na segmentáciu medicínskych snímok mozgu získaných pomocou MRI. Tento systém je založený na distribuovanom prístupe a existuje tu len jeden typ agenta.

Princíp tohto systému je odlišný od predchádzajúcich, pretože agenty pracujú na vyššej úrovni než na úrovni bodov obrazu. Konkrétne rozlišujú dve úrovne manipulácie:

- úroveň objektov,
- úroveň atómov.

Agenty hľadajú model objektu dodaný na vstupe v rámci obrazu a ak nájdu zhodu na úrovni objektov, prispôbujú okraje modelu na úrovni atómov objektu prítomnému v segmentovanom obraze. To je možné docieľiť vďaka tomu, že model je zložený z viacerých atómov. Agenty navzájom komunikujú, vymieňajú si potrebné informácie a tým sa koordinujú. Majú definovaný proces riešenia konfliktov v prípade, že sa nevedia dohodnúť. Proces prispôbovania zachytáva Obr. 4.4, kde modrá elipsa reprezentuje aktuálny tvar modelu objektu, malé modré kruhy reprezentujú jednotlivé atómy modelu a zelené kruhy reprezentujú ohraničenie objektu v obraze.

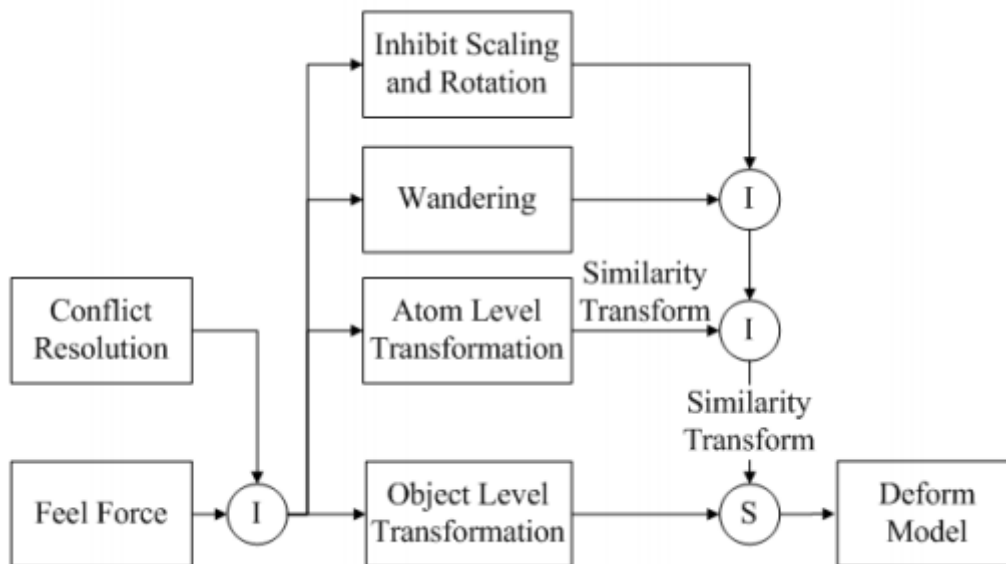


Obr. 4.4 Prispôbovanie modelu objektu [10]

Akcie agentov závisia od úrovne, na ktorej aktuálne pracujú:

- úroveň objektov:
 - pohyb modelu,
 - otočenie modelu,
 - škálovanie modelu,
- úroveň atómov:
 - pohyb atómu,
 - zmena uhlu atómu,
 - zmena vzdialenosti atómov,
- mimoúrovňové:
 - blokovanie otočenia,
 - blokovanie škálovania,
 - túlanie sa.

Mimoúrovňové akcie boli do systému doplnené kvôli zvýšeniu efektívnosti. Blokovanie otočenia a škálovania vychádza z priori vedomostí, že model každého objektu sa môže v obraze nachádzať iba v určitých otočeniach a škálach. Akcia túlania sa bola dodefinovaná ako únik z lokálneho extrému, v podstate slúži ako prechod z úrovne atómov naspäť na úroveň objektov. Celú architektúru agenta spolu s prioritami akcií zobrazuje Obr. 4.5.



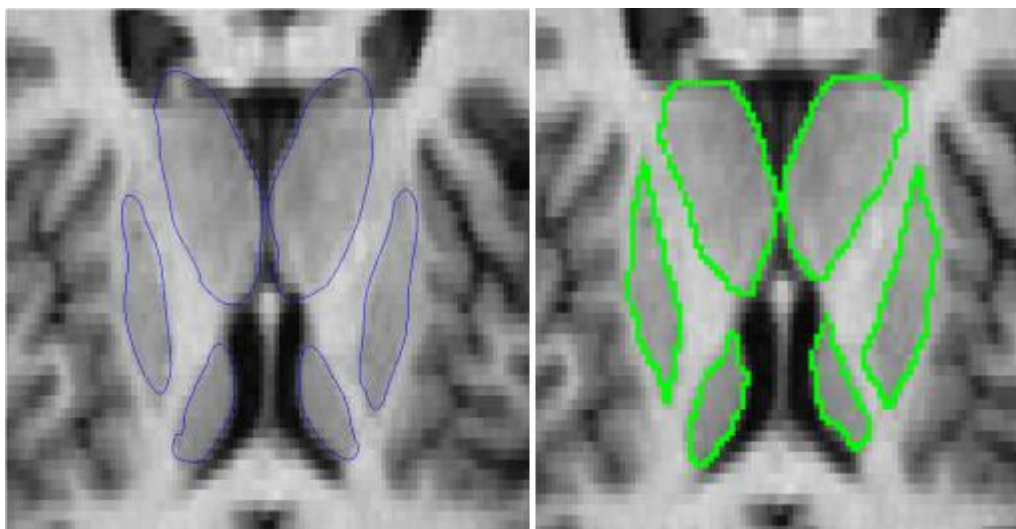
Obr. 4.5 Architektúra agenta v multi-agentovom systéme [10]

Tento multi-agentový systém využíva celú škálu priori a interpretovaných vedomostí:

- pravdepodobné pozície každého modelu,
- pravdepodobná veľkosť a otočenie modelu,
- vnútorné a okolité tkanivá modelu,
- symetria – mozog má dve približne rovnaké hemisféry,

- súhlasné hranice – niektoré modely majú spoločné hranice,
- vzájomná poloha modelov.

Ako porovnávacie kritérium bolo zvolená odchýlka segmentácie pomocou multi-agentového systému oproti ľudskej segmentácii odborníkov. Segmentácia bola otestovaná na dátovej sade, z ktorých každá obsahuje 128 obrazov rozmeru 256x256 pixlov. V práci sa uvádza, že tento systém dosahuje v priemere 83% úspešnosť. Obr. 4.6 zachytáva ukážku segmentácie tejto metódy, naľavo sa nachádza segmentovaný obraz pomocou multi-agentového systému a napravo segmentovaný obraz pomocou ľudskej segmentácie odborníka.



Obr. 4.6 Ukážka segmentácie multi-agentového systému [10]

4.4 Multi-agentový systém autorov Idir, Merouani a Tlili [11]

Štvrtým existujúcim riešením je multi-agentový systém na segmentáciu medicínskych snímok prsníkov za účelom odhalenia nádoru. Tento systém je založený na centralizovanom prístupe riadenom skupinou agentov. V systéme teda rozlišujeme:

- *radiace agenty* – monitorovací agent, triediaci agent, duálny agent a agent používateľského rozhrania,
- *výkonné agenty* – hranové a regionálne agenty,

Princípom tohto systému je využitie duálnej iregulárnej pyramídy. Jedná sa o zásobník grafov, ktoré opisujú obraz. Každá vrstva pyramídy reprezentuje graf spojení, kde vrcholy sú regióny a hrany reprezentujú vzťah susediacich oblastí. Najspodnejšia vrstva pyramídy sa vytvára zo vstupného obrazu. Každá ďalšia vrstva vzniká redukciou tej predchádzajúcej, pričom sa dodržia dve základné pravidlá:

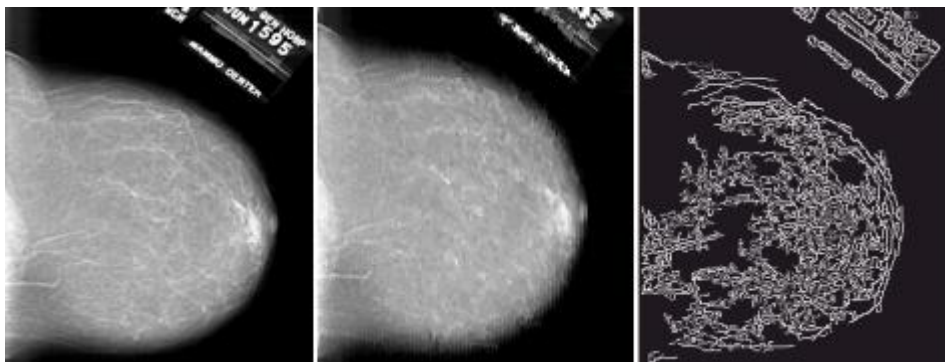
- dvaja susedia nemôžu prežiť do nasledujúcej vrstvy,
- pre každú neprežívajúcu bunku existuje v susedstve prežívajúca bunka na ďalšej úrovni.

Najspodnejšia vrstva regiónov je vytvorená pomocou segmentácie rozvodnením a vrstva hrán je vytvorená pomocou Canny detektora. Následne sa inicializujú príslušné agenty. Dôjde k spojeniu podobných regiónov, pričom ako kritéria homogenity slúžia priemer úrovne šedej v regióne a variancia regiónu. Riadiace agenty zabezpečujú výber výkonných agentov do ďalšej vrstvy. Ak sa vyberú, tak na novej vrstve vytvoria svoju kópiu. Duálny agent sa stará o mapovanie hrán k regiónom.

Každý z agentov má svoje vlastné akcie, ktoré sú pre príslušný typ agenta nutné. V systéme sa uvažovalo s nasledujúcimi akciami pre výkonných agentov:

- *označenie teritória a extrakcia vlastností* – slúži na označenie oblasti, do ktorej agent patrí a na získanie informácií o tejto oblasti (oblasť môže byť definovaná regionálnym agentom alebo aj hranovými agentmi),
- *skúmanie* – slúži na skúmanie okolia a odhalenie susediacich agentov,
- *plánovanie spájania* – agenty sa môžu spájať, ak reprezentujú podobné oblasti,
- *kooperácia* – agenty navzájom kooperujú, aby dosiahli najlepšie možné spájania,
- *decimovanie* – počet agentov každou úrovňou pyramídy klesá, preto je nevyhnutné mať možnosť počet agentov ovplyvňovať,
- *reprodukcia* – prežívajúci agent vytvára na novej úrovni svoju kópiu,
- *pripevnenie* – každý neprežívajúci agent hľadá v susedstve agenta, ktorý sa mu najviac podobá.

Výsledky tohto systému nie sú k dispozícii, systém bol v dobe písania práce len vo vývoji a hľadali sa najlepšie možnosti vyjadrenia kooperácie medzi výkonnými agentmi navzájom. Preto Obr. 4.7 zachytáva iba predspracovanie obrazu, zľava doprava je to pôvodný obraz, regionálny obraz a hranový obraz.



Obr. 4.7 Ukážka predspracovania multi-agentového systému [11]

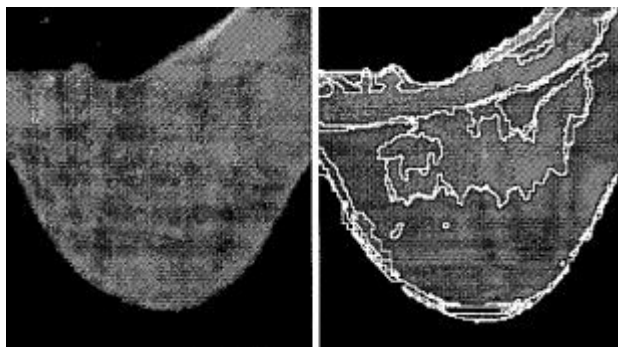
4.5 Multi-agentový systém autorov Duchesnay, Montois, Jacqueslet a Kinie [12]

Posledným existujúcim riešením je tiež multi-agentový systém na segmentáciu medicínskych snímok prsníkov za účelom odhalenia nádoru. Je veľmi podobný multi-agentovému systému [11]. Opäť sa jedná o centralizovaný prístup, ktorý je ale riadený agentovým manažérom, ktorý nemá vlastnosti agenta. Systém funguje na iregulárnej pyramíde s dvoma typmi agentov – regionálne a hranové agenty. Regionálne agenty sú na najspodnejšej úrovni vytvorené zo segmentovaného obrazu regionálnou metódou rozdeľovania. Hranové agenty sú vytvárané z hranového obrazu.

Akcie sú taktiež obdobné, akurát ich je menej:

- *skúmanie* – každý agent si počíta regionálne a hranové vlastnosti z obrazu a ukladá ich do lokálnej databázy,
- *plánovanie spájania* – každý agent hľadá v okolí susedov, z ktorými sa môže spojiť,
- *kooperácia* – každý agent kooperuje so susednými agentmi, aby videl vlastnosti obrazu z ich pohľadu, využívajú sa nato Petriho siete,
- *vyjednávanie* – každý agent vyjednáva so susedmi, aby určili najlepšieho prežívšieho agenta do ďalšej vrstvy.

Presné výsledky tohto systému nie sú tiež k dispozícii. Jedná sa o starší systém testovaný na súčasné pomery zaostalom hardvéri. V práci sa však uvádza, že tento systém dosahuje zhruba rovnakú kvalitu výstupu ako segmentácia pomocou modelu aktívnych kontúr – adaptívnych hadov. Obr. 4.8 zachytáva ukážku segmentácie tejto metódy, naľavo sa nachádza pôvodný obraz a napravo segmentovaný obraz.



Obr. 4.8 Ukážka segmentácie multi-agentového systému [12]

5 Špecifikácia prototypu

Pri špecifikácii prototypu sme vychádzali z doterajšej analýzy. Naším cieľom je vytvoriť základný náčrt multi-agentovej segmentácie tak, aby ju bolo možné rozširovať po zistení nových poznatkov. Agenty budú mať k dispozícii určité dáta o obraze a určité akcie, ktoré nad nimi môžu vykonávať. Použitím externých nastavení budeme schopní definovať ich správanie a správne ich vyladiť, aby vykonávali segmentáciu obrazu požadovaným spôsobom. V prípade, že nebudeme schopní docieľiť výsledok, vytvoríme ďalšie akcie a skúsime prekonfigurovať nastavenia.

Z dôvodu, aby bolo riešenie v rámci možností jednoduché, rozhodli sme sa pre centralizovaný prístup koordinácie agentov podobný ako v [3]. Riadené budú moderátorom, ktorý bude mať k dispozícii všetky aktuálne informácie o obraze a jeho vlastnostiach, a teda bude schopný segmentáciu koordinovať. Najprv si moderátor vytvorí dáta o obraze. Medzi najdôležitejšie budú patriť dáta o hranách a o regiónoch. Vďaka tomu bude možné kombinovať pri segmentácii metódy založené na detekcii hrán a regionálne metódy a tak využiť ich výhody. Ukážku predspracovania prototypu zachytáva Obr. 5.1, zľava doprava je to pôvodný obraz, regionálny obraz a hranový obraz.



Obr. 5.1 Ukážka predspracovania prototypu

Následne sa podľa získaných informácií inicializujú agenty. Každý z agentov bude mať prístup k potrebným globálnym dátam o obraze a bude si uchovávať informácie, čo vykonal. Tým pádom vytvárame reaktívne agenty s pamäťou. Typmi agentov, ktoré využijeme v našej práci sme sa inšpirovali v [11, 12] a jedná sa konkrétne o:

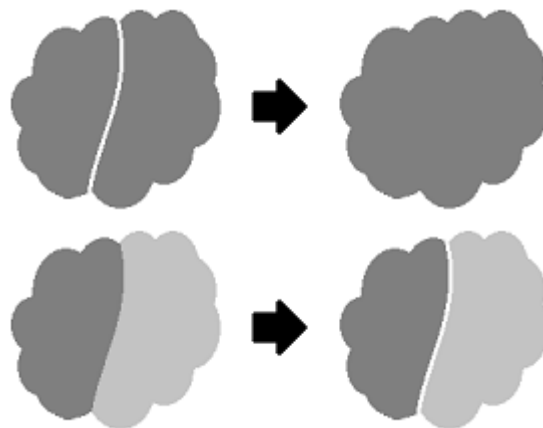
- hranové agenty,
- regionálne agenty.

Oba typy agentov budú odvodené od generického agenta. Vďaka tomu budeme schopní vytvoriť v prípade potreby nový typ agenta. Každý agent bude mať svoje identifikačné číslo, zoznam svojich detí, ktoré môže v procese spracovania vytvárať a aktuálnu pozíciu v obraze.

Hranové agenty budú využívať informácie o hranách a budú schopné tieto informácie modifikovať. Vďaka tomu budú mať k dispozícii dodatočné informácie ako sú vzdialenosti od najbližších hrán, spojitost týchto hrán alebo aké regióny sú v okolí. Okrem klasických akcií ako pohyb a vytvorenie ďalšieho agenta, budú schopné modifikovať dáta o hranách takým spôsobom, že budú vytvárať alebo ničiť hrany v obraze.

Regionálne agenty budú naopak využívať informácie o regiónoch a taktiež budú schopné tieto informácie modifikovať. Budú pracovať na akejsi vyššej úrovni ako hranové agenty, pretože budú k svojej činnosti využívať aj informácie o hranách v rámci samotného regiónu. Budú schopné regióny deliť alebo sa spájať s inými regionálnymi agentmi za vzniku jednej oblasti.

Po inicializácii agentov budú agenty vykonávať činnosť, na ktorú boli navrhnuté, a to pomocou definovaných akcií. Príkladom takých akcií môže byť likvidácia hrany v prípade, že sa vyskytuje medzi dvoma homogénnymi regiónmi alebo vytvorenie hrany medzi dvoma rozdielnymi regiónmi (Obr. 5.2). Každý z agentov sa bude riadiť distribuovaným spôsobom samostatne a až keď skončí svoju činnosť, odovzdá informácie moderátorovi.



Obr. 5.2 Príklady akcií prototypu

Moderátor nebude dohliadať na spracovanie samotného obrazu, ale bude zodpovedný najmä za signalizáciu ukončenia segmentácie. Keď všetky agenty vykonajú všetky činnosti, ktoré mohli vykonať, pošlú moderátorovi informáciu o ukončení svojej činnosti. Takto dostaneme výsledný segmentovaný obraz, ktorý môžeme zobrazit' alebo dodatočne ďalej spracovať.

6 Návrh riešenia

Pri návrhu riešenia sme vychádzali zo spracovanej analýzy domény a existujúcich riešení a taktiež zo špecifikácie prototypu. Návrh bol vytváraný s cieľom zachytiť najdôležitejšiu funkcionálnu a s ohľadom na jednoduché rozširovanie. Pri návrhu počítame s využitím agentov a vytvorením multi-agentového systému. Pretože segmentácia obrazov nie je plne deterministická úloha, potrebujeme využiť prostriedky, ktoré budú dostatočne flexibilné nato, aby nám zabezpečili kvalitné výsledky. Práve použitie agentov nám to dokáže zabezpečiť, pretože správanie agentov vieme ovplyvniť tak, ako budeme potrebovať.

6.1 Identifikácia hráčov systému

V prvom rade je nutné si zadať hráčov systému. Budeme rozlišovať dva typy hráčov:

- interní operátori:
 - moderátor,
 - agenty,
- externí používatelia.

Pod internými operátormi chápeme všetky objekty systému, ktoré sa podieľajú na samotnej segmentácii či už priamo (agenty rôznych typov), alebo nepriamo (moderátor).

Pod externými používateľmi rozumieme všetkých používateľov (ľudia, programy), ktorí dodávajú vstupné obrazy, spúšťajú segmentáciu a vyhodnocujú výsledky.

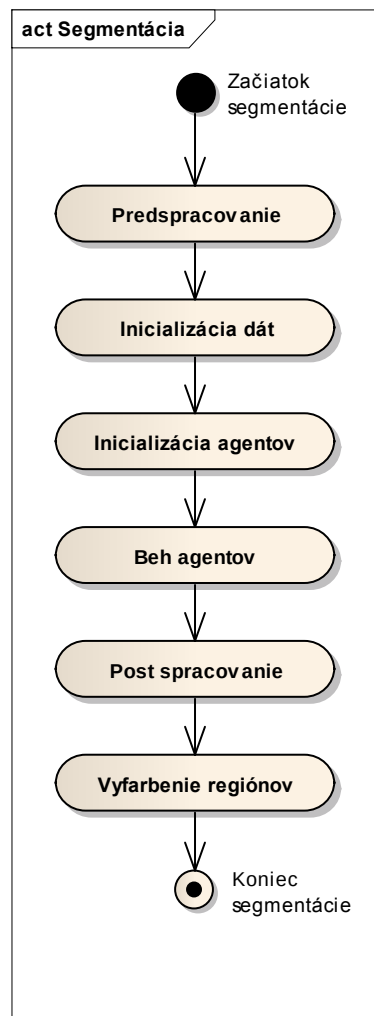
Každý typ hráča bude mať k dispozícii rôzne funkcie, pomocou ktorých bude schopný dosiahnuť svoj cieľ.

6.2 Interný operátor – moderátor

Moderátor je zodpovedný za vytvorenie segmentovaného obrazu pre každý vstupný obraz. Aby to docielil, pre každý z nich sa spúšťa rozsiahly proces. Najprv sa vykoná predspracovanie, kde sa pripraví hranový a regionálny obraz. Potom prebehne inicializácia dát, ktorá zahŕňa naplnenie špeciálnej dátovej štruktúry pre každý pixel zo vstupného obrazu. Následne sa inicializujú agenty na stanovené miesta v obraze. Potom moderátor spustí samotný beh agentov. Po ich skončení sa vykoná post spracovanie, ktoré skoriguje prípadné nedostatky. Posledným krokom

Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskych snímok

je vyfarbenie regiónov, kde sa na základe spracovaných dát vytvorí finálne regionálne rozdelenie obrazu. Postupnosť krokov segmentácie zachytáva Obr. 6.1.



Obr. 6.1 Diagram aktivít segmentácie obrazu

6.2.1 Predspracovanie

V kroku predspracovania je cieľom dostať vstupný hranový a regionálny obraz.

Hranový obraz získame postupnosťou nasledovných krokov:

- *roztiahnutie intervalu jasu na interval (0, 255)* – využívame tak celé spektrum 8-bitoých obrazov,
- *adaptívne prahovanie* – získame jednoduchú binárnu segmentáciu obrazu
- *Canny detektor* – vytvoríme hranový obraz.

Regionálny obraz získame postupnosťou nasledovných krokov:

- *roztiahnutie intervalu jasu na interval (0, 255)* – využívame tak celé spektrum 8-bitoých obrazov,

- *zhlukovanie do regiónov* – na základe hodnoty jasú sú jednotlivé pixle obrazu roztriedené do štyroch skupín, susediace pixle patriace do rovnakej skupiny vytvárajú región.

6.2.2 Inicializácia dát

Proces inicializácie dát naplňa špeciálnu dátovú štruktúru pre každý pixel zo vstupného obrazu. Vzniká tak komplexná informácia o obraze. Položkami tejto štruktúry sú nasledovné informácie:

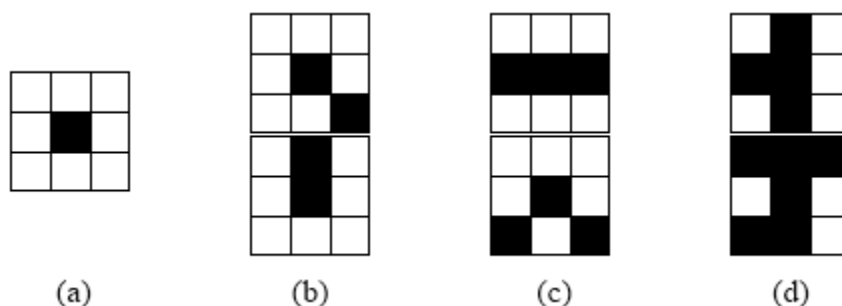
- pôvodná hodnota jasú vstupného šedo-tónového obrazu,
- prítomnosť hrany vo vstupnom obraze,
- prítomnosť hrany v štartovacom obraze pre agentov,
- prítomnosť hrany vo výslednom segmentovanom obraze po spracovaní agentmi,
- prítomnosť hrany vo výslednom segmentovanom obraze po post spracovaní,
- regionálne zaradenie vstupného obrazu predspracovaného regionálneho obrazu,
- regionálne zaradenie výsledného segmentovaného obrazu,
- počet pixlov, ktoré sú hranami v okolí aktuálneho pixlu vstupného obrazu,
- počet pixlov, ktoré sú hranami v okolí aktuálneho pixlu štartovacieho obrazu pre agentov,
- informácia či bol daný pixel už navštívený nejakým agentom,
- uhol,
- amplitúda.

Pod uhlom a amplitúdou rozumieme hodnoty získané pomocou Sobelovho operátora používaného aj na detekciu hrán. Amplitúda je sila odozvy či sa v danom bode nachádza hrana. Čím je hodnota vyššia, tým je pravdepodobnejšie, že sa tam hrana skutočne nachádza. Uhol je smer hrany.

6.2.3 Inicializácia agentov

Keď máme dostupné informácie o vstupnom obraze, môžeme inicializovať agentov. Agenty dostávajú k dispozícii štartovací obraz. Tento obraz vzniká prienikom hranovej informácie a regionálnej informácie. Vďaka tomu máme vstupnú informáciu potvrdenú dvomi nezávislými metódami, čo zvyšuje spoľahlivosť výsledku. Pre každý pixel je vypočítaný počet hrán v okolí. Podľa tohto počtu môžeme jednotlivé pixle rozdeliť do štyroch skupín:

- *osamostatnené pixle* (Obr. 6.2 a) – pixle, ktoré nemajú žiadnu hranu v okolí,
- *koncové pixle* (Obr. 6.2 b) – pixle, ktoré majú práve jednu hranu v okolí,
- *tranzitné pixle* (Obr. 6.2 c) – pixle, ktoré majú práve dve hrany v okolí,
- *rozdeľovacie pixle* (Obr. 6.2 d) – pixle, ktoré majú viac ako dve hrany v okolí.



Obr. 6.2 Kategorizácia pixlov podľa počtu hrán v okolí

Myšlienka inicializácie je taká, že agenti zaujmú všetky pozície koncov hrán (koncové pixle) a všetky pozície, kde je určitá pravdepodobnosť kríženia viacerých hrán (rozdeľovacie pixle). Do osamostatnených pixlov sa inicializujú z dôvodu, že pri procese segmentácie by sa tam inak nemuseli dostať. Agenty sa teda inicializujú na všetky pixle, ktoré sú hranou a zároveň kde počet hrán v okolí sa nerovná dvom (pixle nie sú tranzitné).

6.2.4 Beh agentov

Moderátor spustí agentov, ktorí operujú na základe obrazových dát dovtedy, pokiaľ majú definovanú akciu ako postupovať alebo pokiaľ nevykonajú stanovený maximálny počet akcií. Maximálny počet akcií slúži aj ako poistka voči nekonečnému behu agenta. Výsledky zapisujú priamo do globálnej štruktúry dát o obraze.

6.2.5 Post spracovanie

V tomto procese sa vykoná korekcia hranového obrazu. Korekcia je nutná z dôvodu, že agenti nám nevedia vždy zaručiť, že výstupný hranový obraz obsahuje iba kompletne ohraničené regióny, ktoré môžu prejsť do kroku vyfarbovanie regiónov. Preto sme navrhli nasledovnú postupnosť úloh:

- vytvorenie kostry,
- uzatvorenie vonkajšej hrany,
- uzatvorenie vnútorných hrán.

Kostra sa vytvára s cieľom zmenšiť hrúbku všetkých hrán na úroveň jedného pixlu (Obr. 6.3). Získanie kostry nemá knižnica OpenCV implementovanú, preto sme prevzali zdrojové kódy z [13]. Používa sa nato algoritmus Zhang-Suen [14]. Tento algoritmu je podľa [15] časovo najrýchlejší pre získanie kostry obrazu.

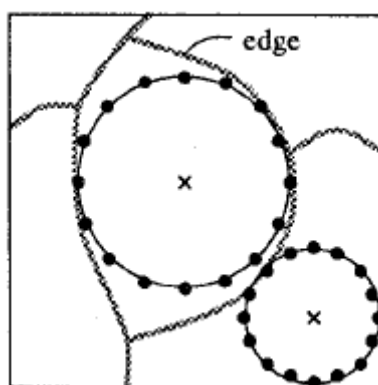


Obr. 6.3 Ukážka vytvorenia kostry [15]

Algoritmus na uzatváranie hrán, ktorý poskytuje uspokojivú spoľahlivosť sa nachádza v [16]. V tejto práci autori zavádzajú koncept rozrastajúcich sa kruhov. Každý z kruhov reprezentuje jeden ucelený región. Metóda sa vykonáva vo viacerých krokoch.

Prvým krokom je extrakcia hranovej informácie obrazu. V našom prípade podsúvame algoritmu hranovú informáciu, ktorú vytvorili agenti.

Druhým krokom je inicializácia kruhov a ich uzlov (Obr. 6.4). Pre tento krok je nutné dopredu vypočítať pre každý pixel obrazu najmenšiu vzdialenosť od existujúcej hrany. Kruhy sú inicializované na miesta lokálnych maxím. Na tento účel sme využili dištančnú transformáciu (distance transform), čím sme dostali masku vzdialeností od najbližšej hrany. Pomocou morfolologickej operácie rekonštrukcie obrazu sme vyfiltrovali lokálne maximá obrazu. Následne sme pre každú z oblastí našli jej globálne maximum a tam sme inicializovali kruhy. Počiatočná veľkosť polomeru je rovná práve hodnote získanej dištančnou transformáciou v iniciálnom bode obrazu. Inicializáciu uzlov kruhu sme vykonali tak, že každý pixel na obvodě kruhu sme vyhlásili za uzol.



Obr. 6.4 Inicializácia kruhov a ich uzlov [16]

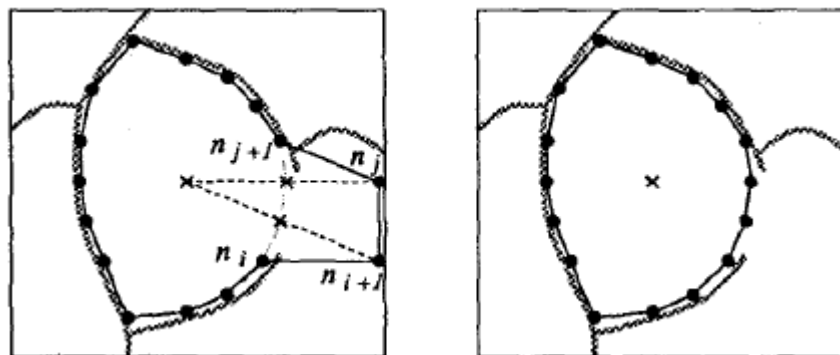
Tretím krokom je samotné rozrastanie kruhov (Obr. 6.5). Pre každý z uzlov kruhu sa vypočíta nová pozícia v obraze. Táto nová pozícia je na priamke spájajúcej stred kruhu a aktuálnu pozíciu uzla, avšak uzol je posunutý smerom od centra kruhu o vzdialenosť rovnajúcej sa

hodnote dištančnej transformácii v aktuálnej pozícii. Posúvanie uzlov sa vykonáva dovtedy, pokiaľ je vzdialenosť od najbližšej hrany väčšia, nanajvýš rovná 1,5.



Obr. 6.5 Proces rozrastania kruhov [16]

Následne sa práca venuje niekoľkým korekčným mechanizmom. Z nich sme implementovali korekciu vyčnievajúcich uzlov (Obr. 6.6).



Obr. 6.6 Korekcia vyčnievajúcich uzlov [16]

Identifikujú sa miesta výrazného nárastu a poklesu vzdialeností uzlov od centra kruhu. Na uzly nachádzajúce sa práve v oblasti zväčšenej vzdialenosti sa aplikuje rovnica interpolácie (Obr. 6.7), ktorá vypočíta novú vzdialenosť uzla od stredu kruhu.

$$r'_k = r_i + \frac{r_{j+1} - r_i}{j - i} * (k - i) \quad (i + 1 \leq k \leq j)$$

Obr. 6.7 Rovnica interpolácie [16]

Opísaný algoritmus sme použili na uzatvorenie vnútorných hrán. Pre potreby uzatvorenia vonkajšej hrany sme algoritmus modifikovali. Proces rozrastania sme zmenili na proces zmenšovania. Inicializujeme štvoruholník na hranu obrazu. Každý pixel štvoruholníka je jeho uzlom. Pozície uzlov sa menia tak, že sa presúvajú smerom k stredu obrazu. Ak na novej pozícii uzlu už je hrana, zmenšovanie daného uzla sa zastaví. Opäť sa použije interpolácia, teraz sa

však nová pozícia vzdáľuje od stredu obrazu. Modifikovaný algoritmus predpokladá, že objekt záujmu segmentácie sa nachádza približne v strede obrazu.

6.2.6 Vyfarbenie regiónov

Cieľom tohto kroku je transformovať hranový obraz na regionálny obraz. Aby sme to dosiahli, najprv použijeme metódu rozrastajúcich oblastí. Identifikujeme tak spojitý regióny. Pre každý z týchto regiónov urobíme prienik s pôvodným hranovým obrazom z predspracovania. Tým sa nám regióny rozbijú na viac menších častí. Každú takto vzniknutú časť ofarbíme farbou, ktorá je rovná priemernej hodnote z pôvodných hodnôt jasu v danej časti obrazu. Pomocou metódy zhľukovania K-Means zaradíme takto vzniknuté regióny do štyroch skupín. Nakoniec roztriedime ešte pixle reprezentujúce hrany. Pixel hrany sa zaradí do takej skupiny, ktorej početnosť v okolí daného pixlu je najväčšia.

6.3 Interný operátor – agent

Každý agent je samostatnou jednotkou, ktorá má definované akcie, ktoré má vykonať na základe globálnych dát o obraze a lokálnych dát agenta. Každý agent si musí pamätať viacero informácií, ktorými sú:

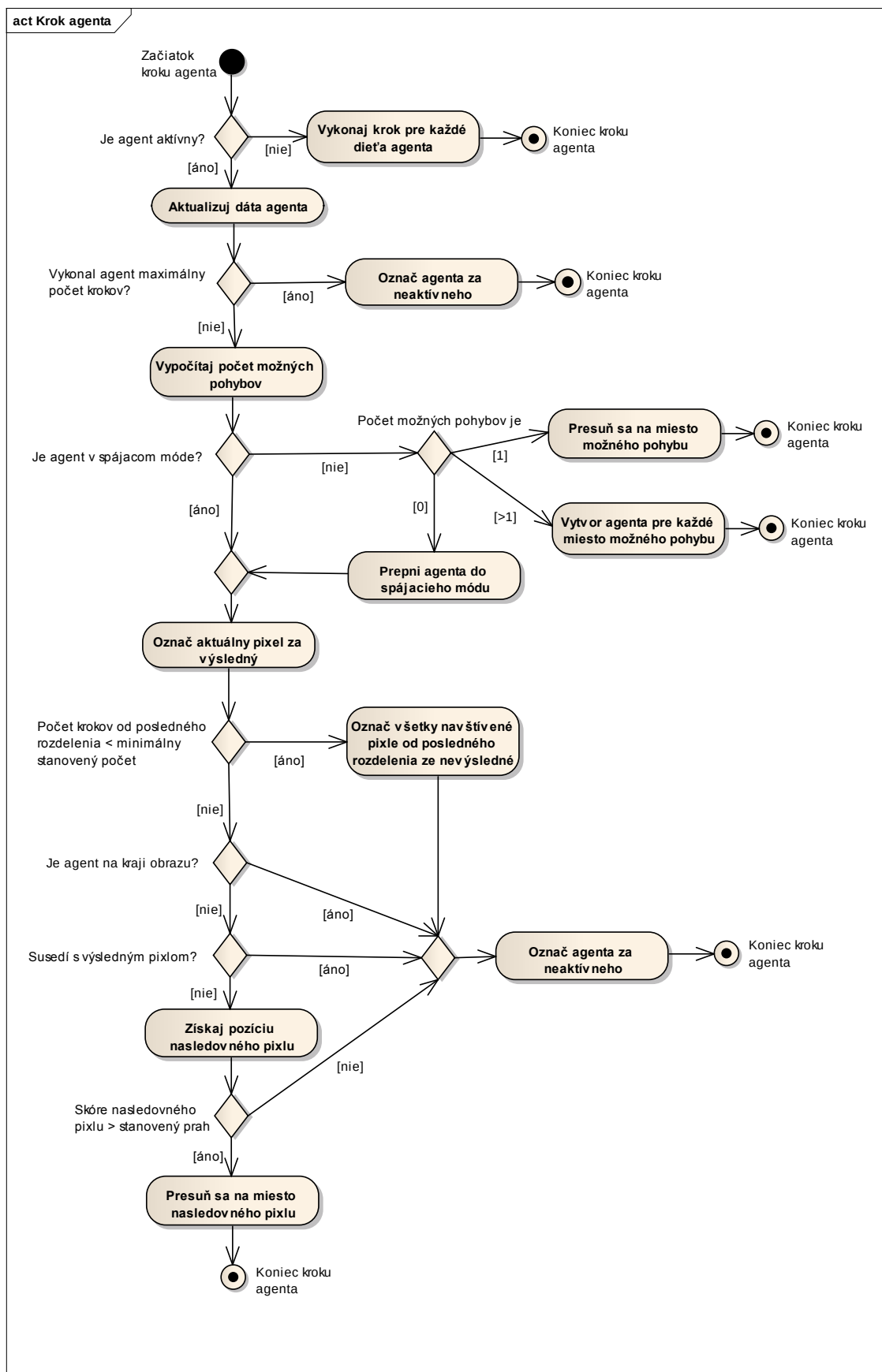
- Identifikátor,
- pozícia v obraze,
- počet vykonaných krokov,
- počet vykonaných krokov od posledného rozdelenia,
- potrebné príznaky (či je agent aktívny, či je v spájacom móde, ...),
- navštívené body obrazu,
- deti agenta.

Najdôležitejšie akcie agenta sú nasledovné štyri:

- pohyb po existujúcej hrane v obraze,
- rozmnoženie agenta do viacerých pixlov v okolí,
- hľadanie nasledovného pixla, keď už neexistuje hrana v obraze,
- zničenie agenta.

Celý proces fungovania agenta je dosť komplexný, je zachytený na Obr. 6.8. V nasledujúcich podkapitolách sa pokúsime bližšie vysvetliť ako agent uvažuje a ako fungujú jednotlivé akcie.

Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskeho snímku



Obr. 6.8 Diagram aktivít kroku agenta

6.3.1 Vykonaj krok pre každé dieťa agenta

Každý agent môže mať svoje deti, ktoré vznikajú v procese rozmnoženia agenta. Táto štruktúra je hierarchická, to znamená, že aj deti agenta môžu mať ďalšie deti. V procese vykonávania segmentácie obrazu môže prísť k tomu, že rodič ukončí svoje vykonávanie skôr ako jeho deti, preto je nutné zabezpečiť, aby jeho deti mohli nerušene pokračovať v práci, pokiaľ majú ešte akcie, ktoré môžu vykonať. Z toho nám vyplýva, že ak je agent neaktívny, treba aj tak skontrolovať či sú aj jeho deti neaktívne. Ak sú aktívne, rekurzívne si skontrolujú svoje deti a v prípade, že môžu vykonať nejakú akciu, vykonajú ju. Počet detí agenta je však obmedzený, aby neprichádzalo k nekontrolovanému rozmnožovaniu agentov.

6.3.2 Aktualizuj dáta agenta

Pre zaručenie správnej funkcionality agenta je nevyhnutné, aby mal agent k dispozícii aktuálne dáta. Preto si v každom kroku musí aktualizovať svoje lokálne dáta. Spravidla sa aktualizuje počet vykonaných krokov a zoznam navštívených bodov obrazu.

6.3.3 Označ agenta za neaktívneho

Pre správne fungovanie systému ako celku musí mať moderátor k dispozícii informáciu, kedy každý agent dokončil svoje vykonávanie a nevedel už pokračovať v segmentácii alebo už vykonal všetky akcie, ktoré mohol. K tomu slúži práve príznak aktivity agenta, ktorý si moderátor po každom kroku všetkých agentov vypýta a zisťuje si či ešte aspoň jeden agent má akciu, ktorú by chcel vykonať. Keď agent nemá žiadnu ďalšiu akciu, zničí sa alebo inak povedané označí sa za neaktívneho.

6.3.4 Vypočítaj počet možných pohybov

Pre každú pozíciu agenta neexistujú rovnaké podmienky, ako sa má agent zachovať. Odvíja sa to na základe počtu možných pohybov, kam sa agent môže pohnúť. Počet možných pohybov je v intervale (0, 8). Zistí sa tak, že pre každý z ôsmich okolitých pixlov sa zistí či nie je krajným pixlom obrazu, či ho agent už niekedy nenavštívil a či tam existuje hrana. Ak pixel splní všetky tri podmienky, je zaradený do možných pohybov agenta z aktuálnej pozície.

6.3.5 Presuň sa na miesto možného pohybu / Presuň sa na miesto nasledovného pixlu

Táto akcia presunie agenta na stanovené nové miesto v obraze. V ďalšom kroku budú jeho akcie závislé od tohto nového miesta.

6.3.6 Vytvor agenta pre každé miesto možného pohybu

Ak sa agent dostane na miesto, odkiaľ môže postupovať viacerými smermi, nemôže dopredu vedieť, ktorý smer je správny alebo či je správnych ciest viacero. Preto sa dokáže rozmnožovať. Sám sa vyberie jednou z ciest kam sa presunie. Na zvyšné miesta možného pohybu vytvorí svoje deti a usadí ich tam. Tie sa v ďalšom kroku budú chovať presne tak, ako každý iný agent, teda správanie nezávisí od toho, či je agent rodičom, alebo dieťaťom.

6.3.7 Prepni agenta do spájacieho módu

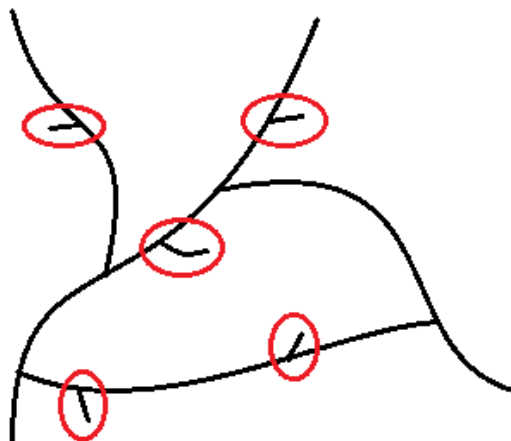
Skôr či neskôr sa každý agent dostane do situácie, že v okolí neexistujú žiadne body, ktoré sú hranou a ešte v nich nebol. V takomto prípade sa agent prepne do spájacieho módu, teda už sa nehýbe po existujúcich hranách v obraze, ale sám ich začína vytvárať alebo ničiť opäť v závislosti od situácie.

6.3.8 Označ aktuálny pixel za výsledný

Táto akcia predstavuje priamu modifikáciu globálnej štruktúry dát o obraze, kde sa zmení informácia, že na danej pozícii, kde sa agent nachádza, sa vyskytuje hrana.

6.3.9 Označ všetky navštívené pixle od posledného rozdelenia za nevýsledné

Agent má stanovený minimálny počet krokov koľko musí prejsť od posledného miesta rozdelenia (miesta, kde sa agent naposledy rozmnožil). Ak je dĺžka takejto hrany kratšia ako stanovený limit, hrana sa vymaže z globálnej štruktúry dát. Vďaka tejto akcii sa z obrazu vymažú malé hrany (Obr. 6.9) a zostávajú nám len tie väčšie, významnejšie. V prípade, že by sme toto chovanie chceli zrušiť, stačí stanoviť limit rovný nule.



Obr. 6.9 Vymazanie krátkych hrán z obrazu

6.3.10 Získaj pozíciu nasledovného pixlu

Keď agent ide vytvárať nové hrany, potrebuje mať určitú heuristiku, ktorá mu povie, ktorý z okolitých bodov je s najväčšou pravdepodobnosťou hrana. Za týmto účelom sme navrhli funkciu počítajúcu skóre každého pixlu v okolí (Obr. 6.10). Z okolitých pixlov sa vyberie bod s najväčším skóre a ak je toto skóre väčšie ako minimálny prah stanovený v nastaveniach, tak sa agent presunie na dané miesto, inak sa agent zničí. Cieľom tohto prístupu pomocou bodov je skombinovanie viacerých zdrojov (amplitúda, uhol, regionálny obraz, hranový obraz, originálny obraz, ...). Nastavením váh (násobiteľov) pre jednotlivé zložky vieme regulovať príspevky jednotlivých zložiek na výsledok. Doplnením tejto rovnice o ďalšie členy vieme systém bodovania okolitých pixlov v budúcnosti pozmeniť podľa potreby.

$$ps = am * ca + rm * cr + em * eTP + pm * ep$$

Obr. 6.10 Rovnica výpočtu skóre pixlu

Vysvetlivky k rovnici výpočtu skóre pixlu na Obr. 6.10:

- ps – výsledné skóre daného pixlu,
- am – násobiteľ kontrolovaného uhlu získaný z nastavení,
- ca – kontrolovaný uhol v intervale (0, 1), jeho výpočet zachytáva rovnica na Obr. 6.11,
- rm – násobiteľ kontrolovaného regiónu získaný z nastavení,
- cr – kontrolovaný región v intervale (0, 1), jeho výpočet zachytáva rovnica na Obr. 6.12,
- em – násobiteľ hrán získaný z nastavení,
- eTP – príznak či v skúmanom bode je, alebo nie je hrana (ak je, $eTP = 1$, inak $eTP = 0$),
- pm – násobiteľ penalty získaný z nastavení,

- ep – hodnota penalty rovná buď -2, -1, alebo 0, jej výpočet zachytáva rovnica na Obr. 6.13.

$$ca = \left(\frac{\min(ampTP, ma)}{ma} \right) * \left(- \left(\frac{\min(|angAP - angTP|, ||angAP - angTP| - 360|)}{360} - 1 \right) \right)$$

Obr. 6.11 Rovnica výpočtu kontrolovaného uhlu

Vysvetlivky k rovnici výpočtu kontrolovaného uhlu na Obr. 6.11:

- $ampTP$ – hodnota amplitúdy v skúmanom bode získaná z globálnych dát o obraze,
- ma – maximálna hodnota amplitúdy stanovená v nastaveniach,
- $angAP$ – uhol v stupňoch v aktuálnom bode získaný z globálnych dát o obraze,
- $angTP$ – uhol v stupňoch v skúmanom bode získaný z globálnych dát o obraze.

$$cr = \left(- \left(\left(\frac{|origAP - origTP|}{255} \right) - 1 \right) \right) * regTP$$

Obr. 6.12 Rovnica výpočtu kontrolovaného regiónu

Vysvetlivky k rovnici výpočtu kontrolovaného uhlu na Obr. 6.12:

- $origAP$ – hodnota jasu originálneho obrazu v aktuálnom bode,
- $origTP$ – hodnota jasu originálneho obrazu v skúmanom bode,
- $regTP$ – príznak či v skúmanom bode je hranica medzi dvomi regiónmi (ak je, $regTP = 1$, inak $regTP = 0$).

$$ep = -resTP - startTP$$

Obr. 6.13 Rovnica výpočtu penalty

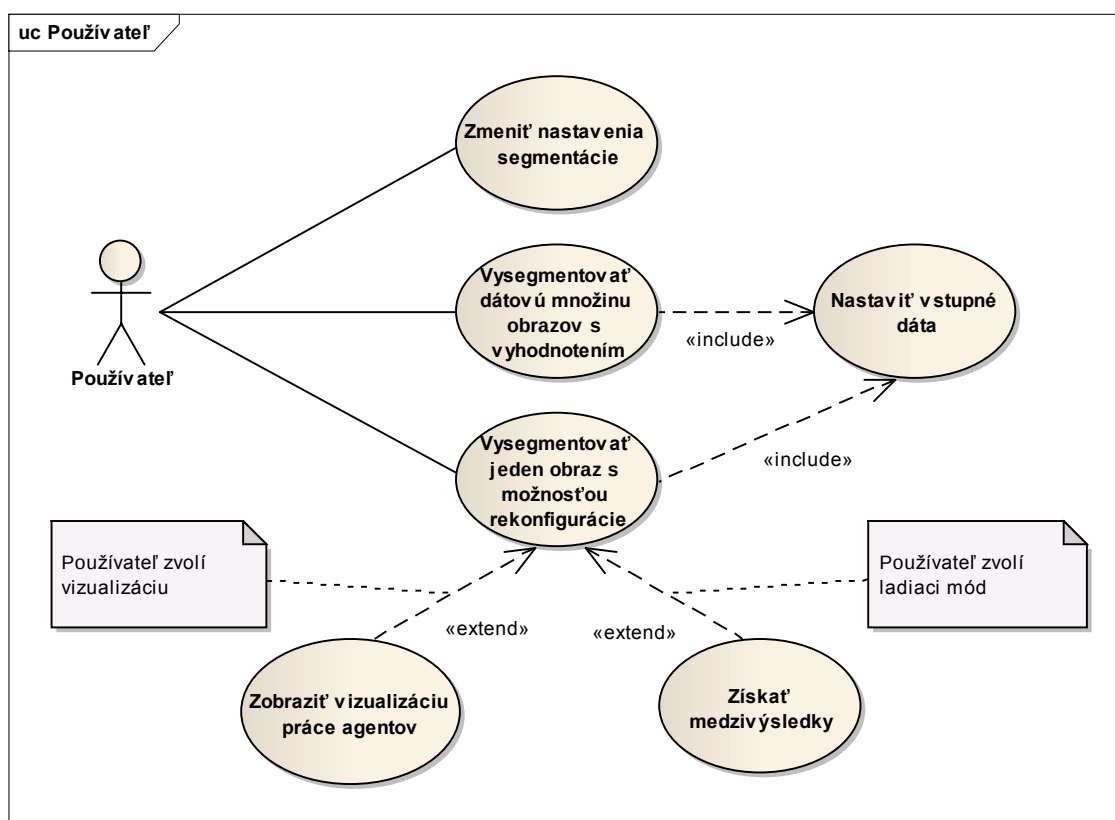
Vysvetlivky k rovnici výpočtu penalty na Obr. 6.13:

- $resTP$ – príznak, či v skúmanom bode je výsledná hrana (ak je, $resTP = 1$, inak $resTP = 0$),
- $startTP$ – príznak, či v skúmanom bode štartovacieho obrazu pre agentov je hrana (ak je, $startTP = 1$, inak $startTP = 0$).

Myšlienka použitia penalty je s cieľom prinútiť agentov hľadať alternatívne hrany. Vďaka tomu, že body s už existujúcimi hranami dostanú aj negatívne skóre, agent sa vyberie radšej na iné miesto, kde je skóre vyššie.

6.4 Interakcia používateľa so systémom

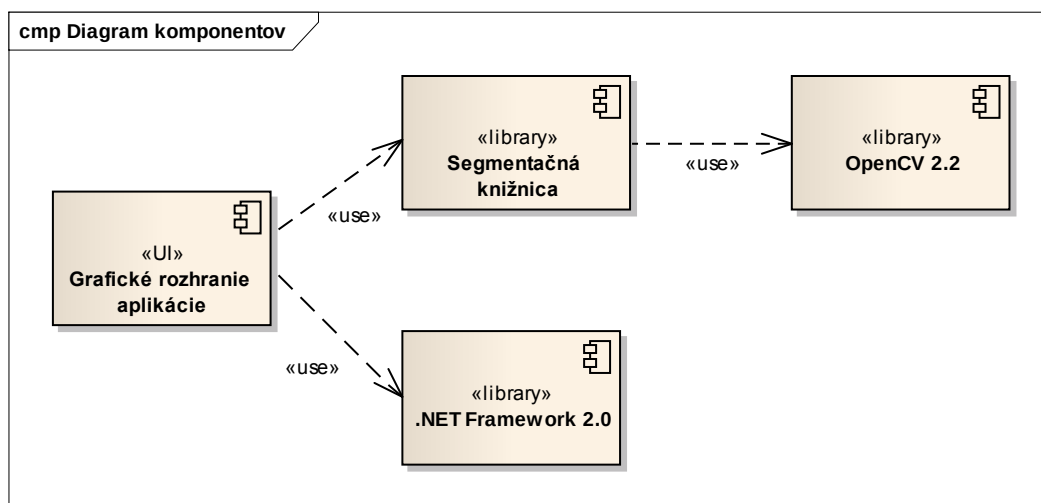
Pre externého používateľa je samozrejme najdôležitejšie získanie segmentovaného obrazu. Segmentovaný obraz môže získať dvomi spôsobmi. Prvým je segmentovanie jedného obrazu. V tomto prípade má externý používateľ možnosť dynamicky meniť nastavenia segmentácie. Keďže uvažujeme, že externými používateľmi môžu byť aj iné programy, zaujímavou funkciou je aj získanie medzivýsledkov, ktoré môžu iné systémy využívať na iné účely alebo na rozšírenie existujúcej segmentácie vlastnými metódami. Taktiež je možné získať medzivýsledky alebo vizualizáciu práce agentov na konkrétnom obraze vo forme videa. Druhým spôsobom je dávkové segmentovanie usporiadané pre vyhodnocovanie výsledkov. Aby bolo možné segmentovaný obraz získať, potrebuje systém vstupnú dátovú množinu obrazov. Preto je nutné, aby externý používateľ mal možnosť nastavenia vstupných dát. Kvôli možnosti nastavenia správania sa interných agentov pri samotnej segmentácii má používateľ možnosť zmeniť aj nastavenia segmentácie. Všetky možnosti interakcie externého používateľa so systémom zachytáva Obr. 6.14.



Obr. 6.14 Diagram prípadov použitia externého používateľa

7 Implementácia riešenia

Systém je rozdelený do dvoch častí. Prvou časťou je grafické rozhranie aplikácie implementované v jazyku C#. Druhou časťou je segmentačná knižnica implementovaná v jazyku C++. Ako implementačné prostredie sme použili Microsoft Visual C# 2010 Express a Microsoft Visual C++ 2010 Express. Architektúru systému zachytáva Obr. 7.1.



Obr. 7.1 Diagram komponentov

Používame aj externú knižnicu OpenCV vo verzii 2.2. Táto knižnica poskytuje viaceré metódy a algoritmy z oblasti počítačového videnia, ktoré nám uľahčujú prácu a manipuláciu s obrazovými dátami. Tie sú rozdelené do viacerých modulov [17]:

- *opencv_core* – základné dátové štruktúry a aritmetické funkcie,
- *opencv_imgproc* – hlavné funkcie spracúvajúce obraz,
- *opencv_highgui* – funkcie na čítanie a zápis obrazu a videa, funkcie používateľského rozhrania,
- *opencv_features2d* – lokálne detektory a deskriptory, funkcie na prácu s nimi,
- *opencv_calib3d* – funkcie na kalibráciu kamery, dvojstranný odhad geometrie, stereo funkcie,
- *opencv_video* – funkcie na odhad pohybu, sledovanie príznakov, extrakciu popredia,
- *opencv_objdetect* – funkcie na detekciu objektov,
- *opencv_ml* – funkcie strojového učenia,
- *opencv_flann* – algoritmy výpočtovej geometrie,
- *opencv_contrib* – funkcie prispievateľov,
- *opencv_legacy* – zastarané funkcie z predchádzajúcich verzií,
- *opencv_gpu* – funkcie pre podporu akcelerácie na grafickej karte.

Implementovanie systému bolo vykonané v súlade s návrhom. Pre viaceré funkcionality spomínaných hráčov systému sme hľadali rôzne spôsoby riešenia, z ktorých sme vybrali najlepšie riešenie. V nasledujúcich podkapitolách podrobnejšie rozoberieme spôsob implementácie jednotlivých funkcií spolu s testovanými alternatívami.

7.1 Funkcionalita moderátora

Moderátor zabezpečuje samotnú segmentáciu obrazu. V návrhu sme definovali postupnosť metód, ktoré sa vykonávajú:

- predspracovanie,
- inicializácia dát,
- inicializácia agentov,
- beh agentov,
- post spracovanie,
- vyfarbenie regiónov.

Moderátor je implementovaný v triede *Moderator*. Podľa fázy, v ktorej sa proces segmentácie nachádza, táto trieda využíva ďalšie triedy ako *Edge_Preprocessor*, *Edge_Processor*, *Edge_Postprocessor*, *Region_Preprocessor* a *Region_Processor*.

7.1.1 Predspracovanie

Predspracovanie sa vykonáva štandardnými metódami spracovania obrazu tak, ako bolo uvedené v návrhu.

7.1.2 Inicializácia dát

Špeciálnu dátovú štruktúru informácií o pixli sme implementovali podľa návrhu v štruktúre *Pixel_Data*:

```
struct Pixel_Data
{
    int original_value;
    bool original_edge;
    bool intersect_edge;
    bool result_edge;
    bool edited_edge;
    int original_region;
    int result_region;
    int surround_edge_count;
    int surround_intersect_count;
    int visited;
    double angle;
    double amplitude;
};
```

Táto štruktúra je vkomponovaná do triedy *Image_Data*, ktorá obsahuje informácie o všetkých pixloch obrazu a poskytuje metódy na ich zmenu a extrakciu.

Pôvodná hodnota vstupného obrazu (*original_value*) sa mapuje zo šedo-tónového obrazu (hodnoty medzi 0 a 255). Z tohto obrazu sa vytvára hranový obraz (*original_edge*). Najprv sa rozťahne škála hodnôt práve na celý interval (0, 255), následne prebehne adaptívne prahovanie a informácia o prítomnosti hrany sa vytiahne pomocou Canny detektora. Zo vstupného obrazu sa tiež vytvára regionálny obraz (*original_region*), kde sa tiež najprv rozťahne škála hodnôt na celý interval a potom sa vykoná zaradenie jednotlivých pixlov do kategórie regiónov. V súčasnej implementácii sa vytvárajú štyri kategórie. Poslednými dvoma informáciami, ktoré sa získavajú zo vstupného obrazu sú uhol (*angle*) a amplitúda (*amplitude*). Tieto hodnoty získame pomocou Sobelovho operátora, ktorý sa používa tiež na detekciu hrán.

Keď máme získané informácie zo vstupného obrazu, pripravíme si pre každý bod počet pixlov v okolí, ktoré sú hranami vo vstupnom obraze (*surround_edge_count*). Vďaka tejto informácii môžeme jednotlivé pixle kategorizovať do skupín, ako sme naznačili v návrhu. Teraz sa pripraví štartovací obraz pre agentov (*intersect_edge*), ktorý vznikne prienikom hranového obrazu a kontúrovej analýzy regionálneho obrazu a tiež sa vypočíta pre každý bod počet pixlov v okolí, ktoré sú hranami v takto vzniknutom obraze (*surround_intersect_count*).

Zvyšné položky štruktúry naplňajú samotné agenty. Ukladajú si body, ktoré navštívili (*visited*) a vytvárajú výsledný hranový obraz (*result_edge*). Vo fáze post spracovania sa naplňajú položky editovaného hranového obrazu (*edited_edge*) a výsledný regionálny obraz (*result_region*).

7.1.3 Inicializácia agentov

Tak, ako sme uviedli v návrhu, agenty sa inicializujú na všetky pixle, ktoré sú hranou v štartovacom obraze pre agentov a zároveň, kde počet hrán v okolí sa nerovná dvom (pixle nie sú tranzitné).

7.1.4 Beh agentov

Beh agentov je implementovaný taktiež v súlade s návrhom. V každom kroku cyklu sa kontroluje či existuje aspoň jeden agent, ktorý má akciu, ktorú vie vykonať. Ak neexistuje, proces segmentácie končí. Agenty si interne kontrolujú počet vykonaných krokov a ak prekročia maximálny stanovený limit, skončia svoje vykonávanie tým, že sa označia za neaktívne. Okrem

toho má moderátor definovaný globálny počet krokov. Keď sa prekročí, proces segmentácie taktiež končí.

7.1.5 Post spracovanie

V súčasnej implementácii sa z výsledného hranového obrazu vyrobí kostra, čím sa hrúbka všetkých hrán zmenší na úroveň jedného pixlu. Následne sa koriguje uzatvorenie vonkajšej hrany a vnútorných hrán podľa algoritmu opísaného v návrhu.

7.1.6 Vyfarbenie regiónov

Vyfarbenie regiónov prebieha presne podľa návrhu.

7.2 Funkcionalita agenta

Agentu sme implementovali podľa návrhu. K takémuto správaniu sme dospeli po viacerých experimentoch na skúšobných obrazoch. Ako kritéria hľadania hrán sme skúšali viacero alternatív. Cieľom týchto experimentov bolo nájsť riešenie, ktoré by poskytovalo čo najlepšie výsledky.

7.2.1 Spájanie na základe vzdialenosti

Prvým pokusom bolo nájdenie všetkých koncových pixlov v obraze a spájanie na základe vzdialenosti, kde sa spájali iba body vzdialené menej ako stanovený limit. Táto metóda sa ukázala ako úplne nedostatočná. Problémom bolo, že body spájala lineárne, čiže nevyužívala lokálne vlastnosti obrazu.

7.2.2 Model aktívnych kontúr

Druhým pokusom bolo vylepšenie za pomoci použitia modelu aktívnych kontúr – adaptívnych hadov, aby sa jednotlivé body spájali s využitím vlastností obrazu, avšak, správanie tohto algoritmu bolo nepredvídateľné a nedostali sme očakávané hrany.

7.2.3 Kombinácia vlastností obrazu

Tretím pokusom bolo nahradenie rozhodovania ako spájať body za použitia jednoduchých informácií, ako či sa na danom mieste nachádza hrana v predspracovanom hranovom obraze a aký región sa nachádza v predspracovanom regionálnom obraze. Dosiahli sme lepšie

výsledky, avšak, nevyriešili sme problém s počtom identifikovaných spojení. Problémom bol relatívne malý počet miest určených na spájanie.

7.2.4 Použitie rovnice pre rozhodovanie agentov

Pretože sme nedokázali identifikovať dostatočný počet miest na spájanie, prehodnotili sme rozmýšľanie agentov. Stanovili sme, že môžu hľadať hrany aj keď nevedia, kam sa v skutočnosti dostanú. Na ohodnotenie, ktorým smerom sa agenty budú pohybovať, sme využili už spomenuté lokálne vlastnosti o obraze a keďže nevieme zatiaľ stanoviť váhy dôležitosti jednotlivých príznakov, vznikla parametrická rovnica opisujúca ohodnotenie možných bodov pohybu agenta. Jednotlivé váhy nastavíme v nastaveniach systému. Celé správanie agenta je implementované v triede *Agent*.

7.3 Funkcionalita externého používateľa

Z návrhu pre externého používateľa boli zadefinované tieto prípady použitia:

- Zmeniť nastavenia segmentácie,
- Nastaviť vstupné dáta,
- Vysegmentovať jeden obraz s možnosťou rekonfigurácie,
- Vysegmentovať dátovú množinu obrazov s vyhodnotením,
- Zobrazit vizualizáciu práce agentov,
- Získať medzivýsledky.

7.3.1 Zmena nastavenia segmentácie

Pod zmenou nastavení segmentácie rozumieme menenie hodnôt parametrov ovplyvňujúcich segmentáciu uložených v konfiguračnom súbore. Konfiguračný súbor je v textovom formáte a má nasledovnú štruktúru:

```
minimum_moving_steps 1
termination_steps 500
max_children_count 10
max_actions_multiplier 30
agent_score_threshold 0
maximum_amplitude 255
angle_multiplier 0.5
region_multiplier 1
edge_multiplier 0.01
penalty_multiplier 0.1
```

Význam jednotlivých nastavení:

- *minimum_moving_steps* – určuje minimálny počet krokov agenta od posledného rozdelenia, ktorý musí prejsť, aby hrana bola označená za výslednú,

- *termination_steps* – určuje maximálny počet krokov agenta, po jeho vyčerpaní bude agent označený za neaktívneho,
- *max_children_count* – určuje maximálny počet detí agenta, po jeho prekročení sú všetky deti agenta označené za neaktívne,
- *max_actions_multiplier* – určuje násobiteľ globálneho počtu krokov, globálny počet krokov je teda daný súčinom tohto násobiteľa a počtom inicializovaných agentov,
- *agent_score_threshold* – minimálna hodnota skóre pre každý skúmaný bod, pre ktorú je agent ochotný pokračovať v hľadaní hrán,
- *maximum_amplitude* – maximálna hodnota amplitúdy, ak je amplitúda vyššia ako táto hodnota, nastaví sa práve na toto stanovené maximum, aby sa predchádzalo neúmernému zvýhodňovaniu bodov s vysokou amplitúdou pri počítaní skóre skúmaných bodov,
- *angle_multiplier* – násobiteľ kontrolovaného uhlu používaný pri výpočte skóre skúmaných bodov,
- *region_multiplier* – násobiteľ kontrolovaného regiónu používaný pri výpočte skóre skúmaných bodov,
- *edge_multiplier* – násobiteľ hrán používaný pri výpočte skóre skúmaných bodov,
- *penalty_multiplier* – násobiteľ penalty používaný pri výpočte skóre skúmaných bodov.

Metódy na inicializáciu nastavení a získanie jednotlivých položiek z nastavení sú implementované v triede *Settings*.

7.3.2 Nastavenie vstupných dát

Pod nastavením vstupných dát rozumieme získanie cesty k adresáru, ktorý obsahuje dátovú množinu obrazov alebo cesty k jednému obrazu v závislosti od spôsobu použitia. Ak máme k dispozícii dátovú množinu segmentovaných obrazov a chceme vykonať porovnanie úspešnosti segmentácie, tak potrebujeme aj cestu k adresáru s príslušnými segmentovanými obrazmi. V tomto prípade sa predpokladá rovnaké pomenovanie korešpondujúcich obrazov. Všetky cesty sú meniteľné cez používateľské rozhranie.

7.3.3 Segmentovanie jedného obrazu z možnosťou rekonfigurácie

Pod touto možnosťou rozumieme spustenie procesu segmentácie jedného obrazu, pričom používateľské rozhranie je usporiadané na testovanie parametrov segmentácie. Tie je možné jednoducho zmeniť a nechať si regenerovať obrazy.

7.3.4 Segmentovanie dátovej množiny obrazov s vyhodnotením

Pod touto možnosťou rozumieme spustenie procesu segmentácie všetkých obrazov z adresára zadaného na vstupe. Po dokončení procesu je možné sa pohybovať medzi výsledkami pre jednotlivé obrazy. Používateľské rozhranie je usporiadané na prezeranie výsledkov, môžeme získať štatistiky pre konkrétny jeden obraz údajovej množiny alebo štatistiky pre celú údajovú množinu obrazov.

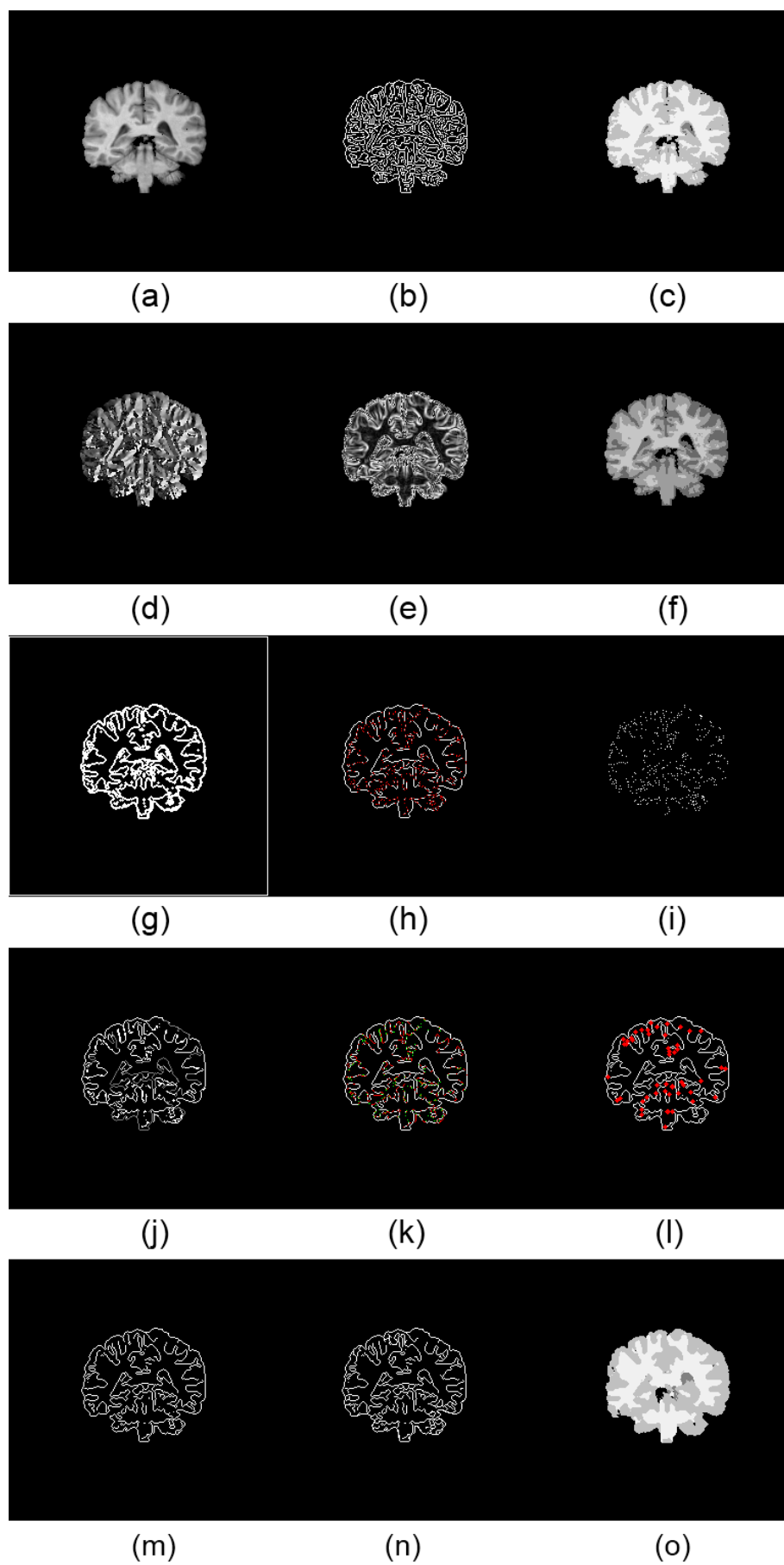
7.3.5 Zobrazenie vizualizácie práce pre agentov

Zobrazenie vizualizácie je implementované priamo v knižnici. Tá poskytuje možnosť generovať videosekvencie. Video, ktoré generuje systém zachytáva vizualizáciu ako agenty dopĺňali hrany a ako sa vyfarbujú jednotlivé regióny.

7.3.6 Získanie medzivýsledkov

Za účelom získania výsledného segmentovaného obrazu a k nemu prislúchajúcich medzivýsledkov sme implementovali triedu *Visualiser*, ktorá vráti obrazy v dátovej štruktúre `cv::Mat`. Túto štruktúru definuje knižnica `OpenCV`. Zoznam získateľných výstupov zahŕňa:

- originálny obraz, ktorý bol dodaný moderátorovi na vstupe (Obr. 7.2 a),
- predspracovaný hranový obraz (Obr. 7.2 b),
- predspracovaný regionálny obraz (Obr. 7.2 c),
- obraz zobrazujúci amplitúdu získanú pomocou Sobelovho operátora (Obr. 7.2 d),
- obraz zobrazujúci uhol získaný pomocou Sobelovho operátora (Obr. 7.2 e),
- segmentovaný obraz pomocou pyramídovej segmentácie implementovanej v knižnici `OpenCV` (Obr. 7.2 f),
- obraz zobrazujúci prechody medzi regiónmi získaný pomocou kontúrovej analýzy regionálneho obrazu (Obr. 7.2 g),
- štartovací obraz dostupný agentom, červenou farbou sú zobrazené štartovacie miesta agentov (Obr. 7.2 h),
- obraz zobrazujúci štartovacie miesta agentov (Obr. 7.2 i),
- obraz zobrazujúci miesta, kde sa agenty pohybovali (Obr. 7.2 j),
- obraz, ktorý obsahuje informáciu, ktoré hrany agenty doplnili (červená farba) alebo vymazali (zelená farba) v porovnaní s dodaným štartovacím obrazom (Obr. 7.2 k),
- obraz s rozpoznanými nespojenými hranami (Obr. 7.2 l),
- výsledný segmentovaný hranový obraz po spracovaní agentmi (Obr. 7.2 m),
- výsledný segmentovaný hranový obraz po post spracovaní (Obr. 7.2 n),
- výsledný segmentovaný regionálny obraz (Obr. 7.2 o).



Obr. 7.2 Ukážka získateľných výstupov

8 Overenie riešenia

Pre potreby overenia riešenia sa nám podarilo získať prístup do repozitára MRI obrazov mozgu [18]. Na takto získaných obrazoch sme vykonávali experimenty a hľadali najlepšie správanie agentov. Repozitár obsahoval mnoho dátových množín obrazov. K niektorým sú poskytované aj cieľové segmentované obrazy segmentované expertmi a referenčné výsledky. Práve tieto dátové množiny sme si zvolili pre vyhodnotenie nášho riešenia.

8.1 Opis dátovej množiny obrazov

Získané dátové množiny obrazov boli v medicínskom formáte, na konverziu do bitmapových súborov sme použili nástroj GWRawMRIView. Pre naše overenie sme vybrali 20 dátových množín, z ktorých každá obsahuje 51 až 61 snímok v rozlíšení 256x256. Každá dátová množina reprezentuje naskenovanie mozgu jedného pacienta. Nízke rozlíšenie snímok je spôsobené tým, že väčšina strojov na magnetickú rezonanciu pracuje buď s rozlíšením 128x128, alebo 256x256. Konkrétny počet snímok v jednotlivých dátových množinách uvádzame v Tab. 8.1.

Tab. 8.1 Počet snímok jednotlivých dátových množín

<i>Označenie dátovej množiny</i>	<i>Počet snímok</i>
1_24	55
2_4	56
4_8	55
5_8	57
6_10	61
7_8	56
8_4	55
11_3	58
12_3	59
13_3	57
15_3	55
16_3	51
17_3	57
100_23	53
110_3	55
111_2	58
112_2	55
191_3	56
202_3	58
205_3	58
Celkom	1125

8.2 Spôsob vyhodnotenia

V referenčných výsledkoch sa hodnotí správne zaradenie jednotlivých pixlov snímku do štyroch kategórií:

- pozadie,
- cerebrospinálny mok,
- sivá hmota,
- biela hmota.

Percentuálne ohodnotenie jednotlivých kategórií dostávame ako podiel počtu správne zaradených pixlov do kategórie a celkového počtu pixlov v danej kategórii. Čím vyššie je percentuálne hodnotenie, tým lepšia je metóda.

8.3 Nastavenie parametrov

Multi-agentový systém je závislý od nastavenia parametrov segmentácie. Aj keď by mali byť predvolené hodnoty dostačujúce, hľadali sme najlepšie parametre pre zvolenú dátovú množinu. Výsledky jednotlivých experimentov zachytáva Tab. 8.2.

Tab. 8.2 Vyhodnotenie segmentácie v závislosti od parametrov

<i>Zmeny parametrov voči predvoleným hodnotám</i>	<i>Pozadie</i>	<i>Cerebro-spinálny mok</i>	<i>Sivá hmota</i>	<i>Biela hmota</i>	<i>Celkom</i>
žiadne	99,525%	65,276%	32,333%	56,079%	93,473%
angle_multiplier 1	99,526%	65,475%	31,622%	55,990%	93,427%
angle_multiplier 0.25	99,525%	65,111%	32,379%	55,823%	93,487%
angle_multiplier 0.25 region_multiplier 0.5	99,525%	65,581%	31,572%	55,641%	93,406%
angle_multiplier 0.25 edge_multiplier 0.1	99,526%	64,974%	32,242%	54,984%	93,418%
angle_multiplier 0.25 penalty_multiplier 0.2	99,526%	65,036%	32,137%	55,702%	93,482%
angle_multiplier 0.25 penalty_multiplier 0.05	99,525%	65,100%	32,465%	55,802%	93,463%

Predvolené hodnoty parametrov sú nasledovné:

- *minimum_moving_steps* 1,
- *termination_steps* 500,
- *max_children_count* 10,
- *max_actions_multiplier* 30,
- *agent_score_threshold* 0,
- *maximum_amplitude* 255,

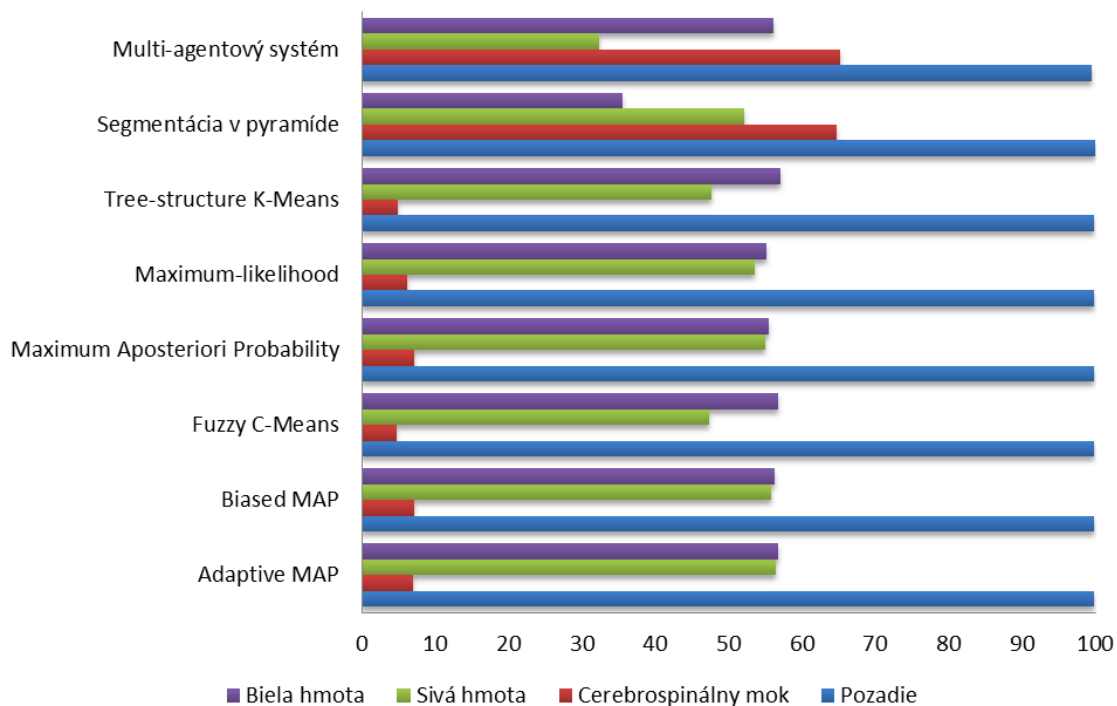
- *angle_multiplier* 0.5,
- *region_multiplier* 1,
- *edge_multiplier* 0.01,
- *penalty_multiplier* 0.1.

8.4 Porovnanie výsledkov voči iným metódam

Porovnanie multi-agentového systému voči iným metódam sme uskutočnili na kompletnej dátovej množine 1125 snímok za použitia predvolených nastavení. Výsledky zachytáva Tab. 8.3 a Obr. 8.1.

Tab. 8.3 Porovnanie výsledkov voči iným metódam

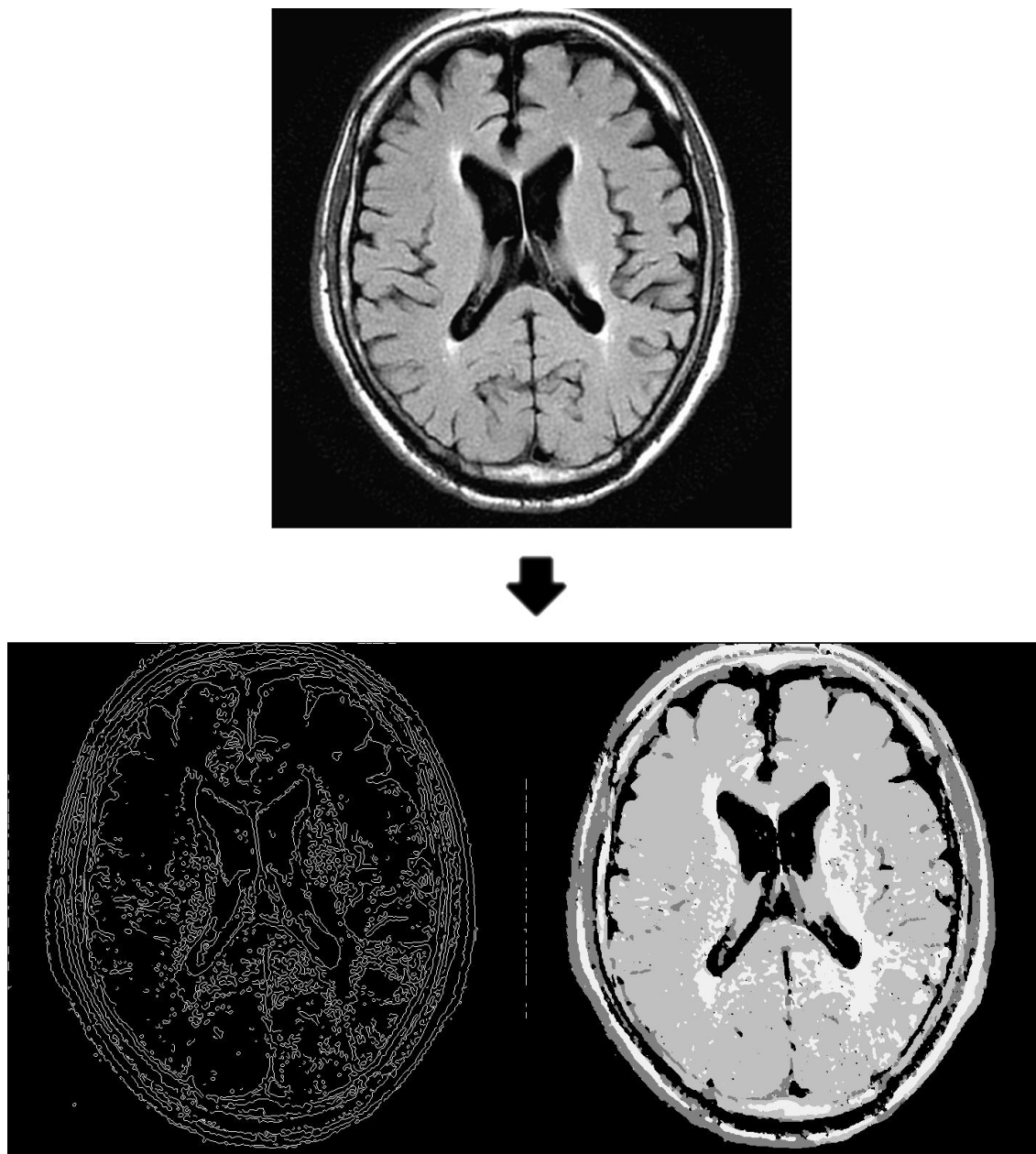
<i>Metóda</i>	<i>Pozadie</i>	<i>Cerebrospinálny mok</i>	<i>Sivá hmota</i>	<i>Biela hmota</i>
Multi-agentový systém	99,525%	65,276%	32,333%	56,079%
Segmentácia v pyramíde	99,999%	64,756%	52,058%	35,606%
Tree-structure K-Means	99,9%	4,9%	47,7%	57,1%
Maximum-likelihood	99,9%	6,2%	53,5%	55,1%
Maximum Aposteriori Probability	99,9%	7,1%	55,0%	55,4%
Fuzzy C-Means	99,9%	4,8%	47,3%	56,7%
Biased MAP	99,9%	7,1%	55,8%	56,2%
Adaptive MAP	99,9%	6,9%	56,4%	56,7%



Obr. 8.1 Graf výsledkov segmentácie rôznych metód

8.5 Výstupy systému

Metódu multi-agentového systému sme otestovali okrem dátovej množiny aj na obrazoch s vyšším rozlíšením, kde vizuálne dosahuje dobré výsledky. Obr. 8.2 zachytáva segmentáciu obrazu s rozlíšením 720x720 pixelov.



Obr. 8.2 Segmentácia obrazu s vyšším rozlíšením

8.6 Vyhodnotenie časovej náročnosti

Okrem kvality výstupu nás zaujíma aj rýchlosť riešenia. Z tohto dôvodu sme vyhodnotili dĺžku trvania segmentácie opäť na kompletnej dátovej množine 1125 snímok za použitia predvolených nastavení. Dosiahli sme priemerný čas 1155,722 ms na jednu snímku, pričom

Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z
lekárskych snímok

minimálny čas potrebný na segmentáciu jednej snímky činil 249 ms a maximálny čas činil 2879 ms. Parametre testovanej zostavy sú nasledovné:

- dvojjadrový procesor Intel Core i5 procesor s taktom 2,53 GHz,
- 4 GB RAM,
- pevný disk Intel SSD 520 Series.

9 Zhodnotenie

Cieľom tejto práce bolo získať informácie o používaných metódach segmentácie a o možnosti využitia multi-agentových systémov pre vylepšenie segmentácie lekárskych snímok. Dôraz bol kladený na preskúmanie niekoľkých existujúcich riešení, ich prípadných nedostatkov a ich možnosti vylepšenia aj vďaka kombinácii viacerých prístupov.

Začiatok práce bol venovaný analýze problémovej oblasti segmentácie. Pozreli sme sa, aké základné metódy sa používajú na segmentáciu obrazu. Zanalyzovali sme prahové metódy, metódy založené na detekcii hrán, regionálne metódy a metódy zhlukovania. Vyzdvihli sme ich prednosti a nedostatky. Ďalej sme sa zamerali na preskúmanie multi-agentových systémov, ako takéto systémy fungujú a čo nám prinášajú. Následne sme sa pozreli na viaceré práce venujúce sa tejto problematike a podali sme informácie o funkčnosti multi-agentových systémov v týchto prácach pre využitie segmentácie lekárskych snímok.

Získané informácie sme využili pri prvotnej špecifikácii prototypu. Snahou bolo vytvorenie špecifikácie, ktorá by umožňovala ľahké rozširovanie. Navrhli sme riešenie s použitím riadených agentov, ktorých správanie je ovplyvniteľné nastaveniami v konfiguračnom súbore. To nám zabezpečí ľahkú zmenu ich chovania len pomocou zmeny hodnôt jednotlivých nastavení. Pri návrhu riešenia sme sa inšpirovali najmä prácami [11] a [12]. Využili sme myšlienku použitia centralizovanej koordinácie agentov a rozdelenia činností na úrovni hrán a regiónov.

Taktiež sme sa oboznámili s funkciami knižnice OpenCV. Táto knižnica zoskupuje funkcie a metódy na prácu s obrazmi v rámci počítačového videnia, do ktorej patrí aj segmentácia obrazov. Rozbehali sme platformu na implementáciu programu s využitím tejto knižnice, zistili sme reálnosť implementácie a začali pracovať na riešení.

Správanie agentov nie je jednoznačne definovateľné, preto sme experimentovali s možnosťami ich správania. Vykonali sme viacero pokusov, ktoré nám poskytli rozdielne výsledky. Nakoniec sme sa rozhodli pre riadenie agentov za použitia funkcie počítajúcej skóre pixlov. Funkcia kombinuje viacero základných obrazových informácií. Miera príspevku jednotlivých zložiek je konfigurovateľná, vďaka čomu máme možnosť ovplyvňovať samotný proces segmentácie.

Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskych snímok

Následne sme sa venovali samotnej implementácii. Implementovali sme systém, ktorý dokáže načítať vstupný obraz, inicializovať potrebné dáta a vytvoriť agentov, ktorí dokážu segmentovať obraz a vrátiť nám výstupný segmentovaný hranový obraz.

Výstupy agentov neboli dokonalé a často sa stávalo, že po segmentácii zostávali v obraze nespojené hrany a tým pádom nám nevznikali ucelené regióny. To nám spôsobovalo problémy s ofarbením regiónov. Navrhli sme využiť kľúčovú metódu uzatvárania hrán z práce [16]. Vďaka použitiu tejto metódy a jej čiastočnej adaptácie sme vyriešili problém uzatvárania hrán a implementovali sme ofarbovanie regiónov.

Z dôvodu možnosti ľahšej rozšíriteľnosti sme oddelili prezentačnú a aplikačnú časť. Vzniklo tak grafické používateľské rozhranie, ktoré využíva funkcie segmentácie z oddelenej knižnice. Používanie aplikácie sa stalo používateľsky prívetivejším ako použitie konzoly. Doplnili sme aj možnosť získať vizualizáciu práce agentov ako videosekvenciu.

V poslednom kroku sme sa zaoberali vyhodnotením výsledkov. Použili sme dátovú množinu obrazov [18] a porovnali sme naše výsledky voči iným riešeniam. Porovnania sme prezentovali v časti overenia riešenia. Ako sa ukázalo, multi-agentový systém je konkurencie schopný, ale stále má potenciál ďalšieho zlepšovania. Hodnotili sa štyri kategórií segmentácie. Náš multi-agentový systém skončil v jednej z kategórií ako prvý a v dvoch ďalších kategóriách zaostával od najlepšieho len o 1%. Systém nebol optimalizovaný špeciálne pre použitie segmentácie mozgov, takže metóda je použiteľná aj v inom kontexte.

10 Zoznam použitej literatúry

1. Wei, B.C., Mandava, R.: Multi-objective nature-inspired clustering techniques for image segmentation. In: *Cybernetics and Intelligent Systems (CIS), 2010 IEEE Conference on*, Singapore (June 2010), 150-155.
2. Varshney, S.S., Rajpal, N., Purwar, R.: Comparative study of image segmentation techniques and object matching using segmentation. In: *Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*, Delhi (December 2009), 1-6.
3. Chitsaz, M., Seng, W.C.: A Multi-agent System Approach for Medical Image Segmentation. In: *Future Computer and Communication, 2009. ICFCC 2009. International Conference on*, Kuala Lumpur (April 2009), 408-411.
4. Gonzales, R.C., Woods, R.E.: Digital Image Processing. Second Edition. New Jersey: Prentice-Hall, Inc., 2002. ISBN 0-201-18075-8.
5. Dobeš, M.: Zpracování obrazu a algoritmy v C#. 1. vyd. Praha: BEN – technická literatura, 2008. ISBN-978-80-7300-233-6.
6. Bradski, G., Kaehler, A.: Learning OpenCV. First Edition. Sebastopol: O'Reilly Media, Inc., 2008. ISBN-978-0-596-51613-0.
7. Szeliski, R.: Computer Vision: Algorithms and Applications. 2010. http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf (2011-05-08)
8. Návrát, P. a i.: Umelá inteligencia. 2. vyd. Bratislava: Vydavateľstvo STU, 2007. ISBN 978-80-227-2629-0.
9. Liang, J., Xu, Z., Mao, D., Cheng, X.: Multi-agent based evolutionary algorithm in medicine image segmentation. In: *Journal of Communication and Computer*, vol. 6, June 2009, no. 6 (serial no. 55), ISSN 1548-7709, 31-35.
10. Shariatpanahi, H.F., Batmanghelich, N., Kermani, A.R.M., Ahmadabadi, M.N., Soltanian-Zadeh, H.: Distributed Behavior-based Multi-agent System for Automatic Segmentation of Brain MR Images. In: *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, Vancouver (October 2006), 4535-4542.
11. Idir, K., Merouani, H., Tlili Y.: Dual Pyramid of Agents for Image Segmentation. In: *World Academy of Science, Engineering and Technology*, vol. 6, June 2005, no. 47, 198-201.
12. Duchesnay, E., Montois, J.-J., Jacquelet, Y., Kinie, A.: An Agent-based Implementation of Irregular Pyramid for Distributed Image Segmentation. In: *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, Antibes-Juan les Pins (October 2001), 409-415.
13. Parker, J.R.: Algorithms for image processing and computer vision. First Edition. New York: John Wiley & Sons, 1997. ISBN 978-0471140566.
14. Zhang, T.Y., Suen, C.Y.: A Fast Parallel Algorithm for Thinning Digital Patterns. In: *Communication of the ACM*, vol. 27, March 1984, no. 3, ISSN 0001-0782, 236-239.
15. Widiarti, A.R.: Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendraprasad - Wang-Gupta Thinning. In: *VII. International University Conference 2011, Amsterdam* (July 2011), 146-150.

16. Shimbashi, T., Kobuko, Y., Shirota, N.: Region segmentation using edge based circle growing. In: *Image Processing, 1995. Proceedings., International Conference on*, Washington, D.C. (October 1995), 65-68.
17. Laganière, R.: *OpenCV 2 Computer Vision Application Programming Cookbook*. First Edition. Birmingham: Packt Publishing Ltd., 2011. ISBN 978-1-849513-24-1.
18. Internet Brain Segmentation Repository. <http://www.cma.mgh.harvard.edu/ibsr/> (2011-10-11)

A Inštalačná príručka

Cieľom inštalačnej príručky je poskytnúť návod, ako spustiť program a ako pripraviť prostredie na vývoj pre používateľov, ktorí plánujú rozširovať implementovaný systém. Táto príručka poskytuje návod pre implementačné prostredie Microsoft Visual C++ 2010 Express a Microsoft Visual C# 2010 pre operačný systém Microsoft Windows 7.

Spustenie programu bez implementačného prostredia:

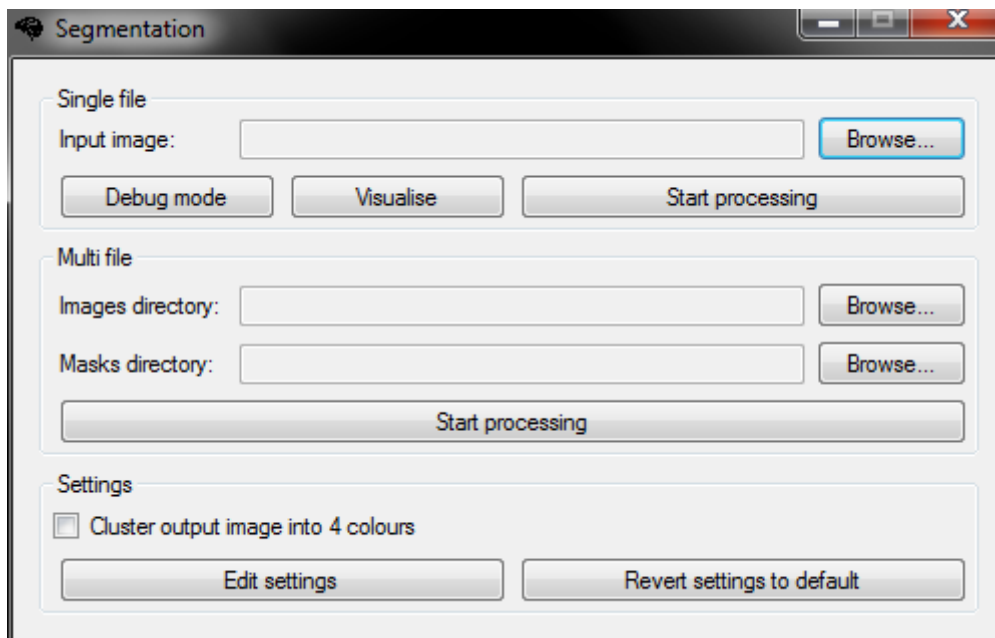
- nainštalujeme OpenCV 2.2 (*/install/OpenCV-2.2.0-win32-vs2010.exe*):
 - uistíme sa, že pri inštalácii zaškrtneme možnosť pridať OpenCV do premennej prostredia *PATH*,
- ak používame starší operačný systém, uistíme sa, že máme nainštalovaný .NET Framework 2.0 (*/install/dotnetfx.exe*),
- skopírujeme program z CD nosiča na disk (*/bin*):
 - tento krok je nutný z dôvodu, že program si vytvára priečinok s obrazmi bez ktorého nebude korektne fungovať,
- spustíme program SegmentationGUI (*/bin/SegmentationGUI.exe*),
- opis, ako používať program sa nachádza v používateľskej príručke.

Príprava prostredia na vývoj:

- nainštalujeme OpenCV 2.2 (*/install/OpenCV-2.2.0-win32-vs2010.exe*):
 - uistíme sa, že pri inštalácii zaškrtneme možnosť pridať OpenCV do premennej prostredia *PATH*,
- nainštalujeme Microsoft Visual C++ 2010 Express (*/install/vc_web.exe*),
- nainštalujeme Microsoft Visual C# 2010 Express (*/install/vcs_web.exe*),
- skopírujeme projekty z CD nosiča na disk (*/src*),
- spustíme projekt Segmentation (*/src/Segmentation/Segmentation.sln*),
- stlačíme F5, projekt sa skompiluje a vytvorí sa segmentačná knižnica (*Segmentation.dll*),
- spustíme projekt SegmentationGUI (*/src/SegmentationGUI/SegmentationGUI.sln*),
- zmeníme typ výstupu z *Debug* na *Release*:
 - zapneme pokročilé nastavenia (*Menu => Tools => Settings => Expert settings*),
 - zobrazíme nastavenia (*Menu => Tools => Options*),
 - zobrazíme všetky nastavenia (v otvorenom okne zaškrtneme *All settings*),
 - zapneme pokročilé nastavenia kompilácie (v otvorenom okne ideme do menu *Projects and solutions* a zaškrtneme *Show advanced build configurations*),
 - otvorené okno zavrieme a v menu prepneme typ výstupu z *Debug* na *Release*,
- stlačíme F5, projekt sa skompiluje a program sa spustí (*SegmentationGUI.exe*).

B Používateľská príručka

Po spustení programu sa zobrazí úvodné okno (Obr. B.1).



Obr. B.1 Úvodné okno programu

Úvodné okno je rozdelené do troch častí:

- spracovanie jedného obrazu,
- dávkové spracovanie viacerých obrazov,
- nastavenia segmentácie.

B.1 Spracovanie jedného obrazu

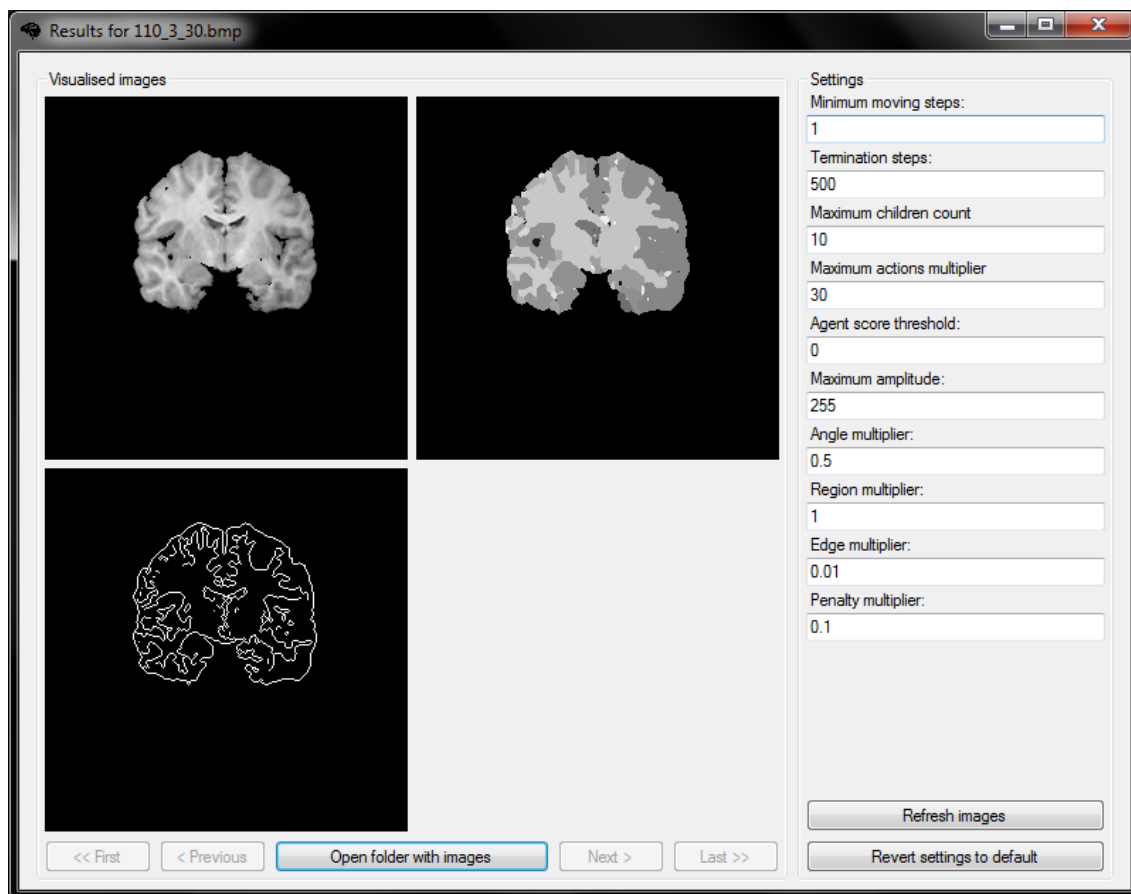
Táto časť programu je usporiadaná na prácu s jednotlivými obrazmi samostatne. Tlačidlom *Browse* vyberieme zdrojový obraz. Následne máme tri možnosti ako obraz spracovať:

- *zobrazenie všetkých výstupov segmentačnej knižnice* – stlačením tlačidla *Debug mode* segmentačná knižnica vráti všetky ponúkané obrazy vrátane všetkých medzivýsledkov a otvorí adresár, kde sa uložili,
- *zobrazenie vizualizácie práce agentov* – stlačením tlačidla *Visualise* sa vygeneruje video prezentujúce ako agenti dopĺňali hrany do obrazu a ako boli výsledné regióny ofarbené,
- *zobrazenie výsledkov segmentácie* – stlačením tlačidla *Start processing* sa spustí proces segmentácie a zobrazí sa nám okno s výsledkami (Obr. B.2).

Okno s výsledkami zobrazuje tri obrazy. Vľavo hore sa nachádza zdrojový obraz, vpravo hore výsledný segmentovaný obraz a vľavo dole hranový obraz po spracovaní agentmi. Po dvojkliku

Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskych snímok

na hociktorý z obrazov sa tento obraz otvorí v predvolenom prehliadači obrázkov. Tlačidlo *Open folder with images* otvorí adresár obsahujúci uložené obrazy.



Obr. B.2 Okno s výsledkami pri spracovaní jedného obrazu

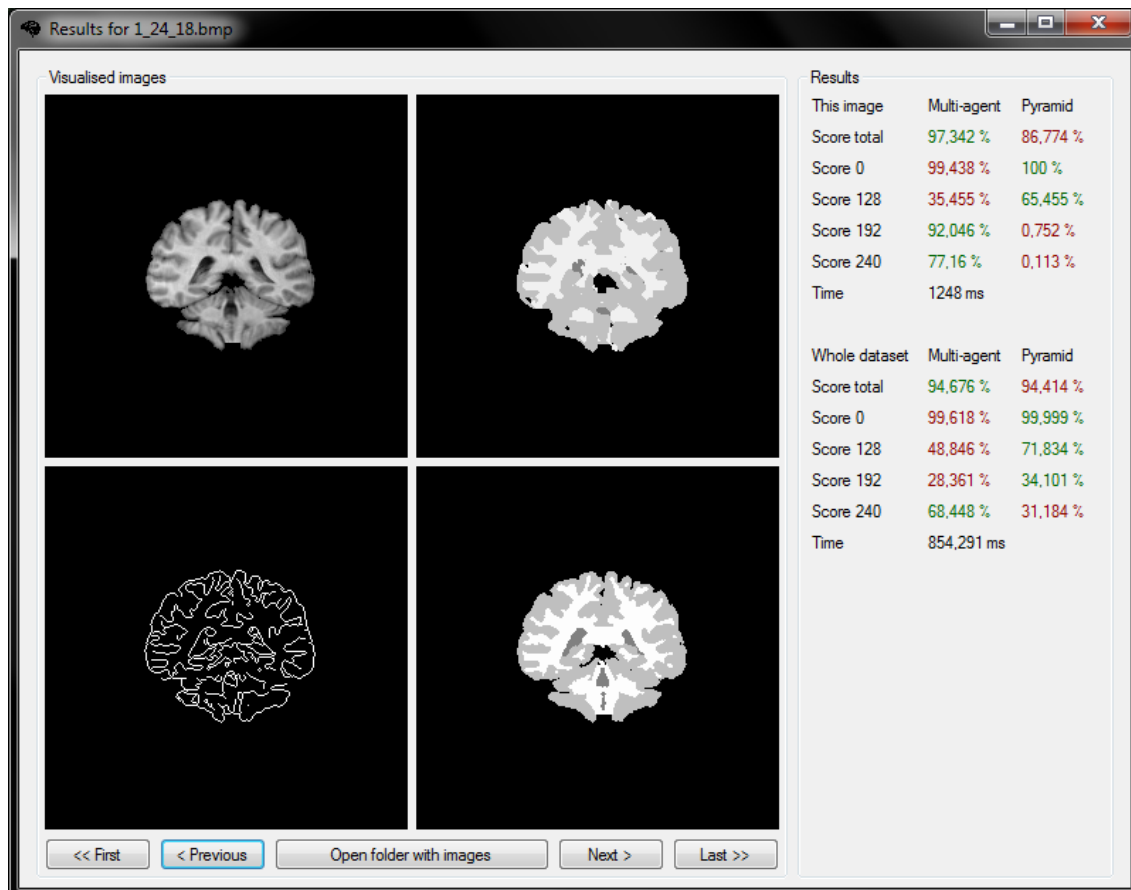
V pravej časti okna sa nachádza sekcia s nastaveniami. Máme tu možnosť meniť nastavenia segmentácie. Tlačidlom *Refresh images* sa zmenené nastavenia uložia do súboru, opätovne sa spustí segmentácia obrazu a zobrazia sa nám nové obrazy. Táto funkcia je veľmi užitočná najmä pre testovanie ako jednotlivé parametre ovplyvňujú výsledky. Tlačidlo *Revert settings to default* vráti zmeny v nastaveniach a obnoví pôvodné hodnoty.

B.2 Dávkové spracovanie viacerých obrazov

Táto časť programu je uspokojená na prácu s množinou obrazov s účelom vyhodnotenia. Pomocou horného tlačidla *Browse* vyberieme adresár obsahujúci množinu obrazov, ktoré chceme segmentovať. Dolným tlačidlom *Browse* vyberiem adresár obsahujúci množinu obrazov, s ktorými chceme porovnávať. Predpokladá sa zhodný názov korešpondujúcich obrazov. Tlačidlom *Start processing* spustíme segmentáciu. Zobrazí sa stav priebehu segmentácie a po dokončení procesu sa zobrazí okno s výsledkami (Obr. B.3).

Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskych snímkov

Okno s výsledkami zobrazuje štyri obrazy. Vľavo hore sa nachádza zdrojový obraz, vpravo hore výsledný segmentovaný obraz, vľavo dole hranový obraz po spracovaní agentmi a vpravo dole obraz, s ktorým porovnávame. Po dvojkliku na hociktorý z obrazov sa tento obraz otvorí v predvolenom prehliadači obrázkov. Tlačidlami *First*, *Previous*, *Next* a *Last* sa presúvame medzi jednotlivými obrazmi, ktoré boli v zvolenom adresári. Tlačidlo *Open folder with images* otvorí adresár obsahujúci uložené obrazy.



Obr. B.3 Okno s výsledkami pri dávkovom spracovaní viacerých obrazov

V pravej časti okna sa nachádza sekcia s vyhodnotením. Vyhodnotenie obsahuje percentuálnu úspešnosť segmentácie pre jednotlivé sledované farby a časové vyhodnotenie. V hornej časti sú výsledky pre aktuálne zobrazený obraz, v dolnej časti sú výsledky pre celý adresár. Všetky výsledky sú uvádzané pre náš multi-agentový systém a pre porovnanie tiež pre segmentáciu v pyramíde.

B.3 Nastavenia segmentácie

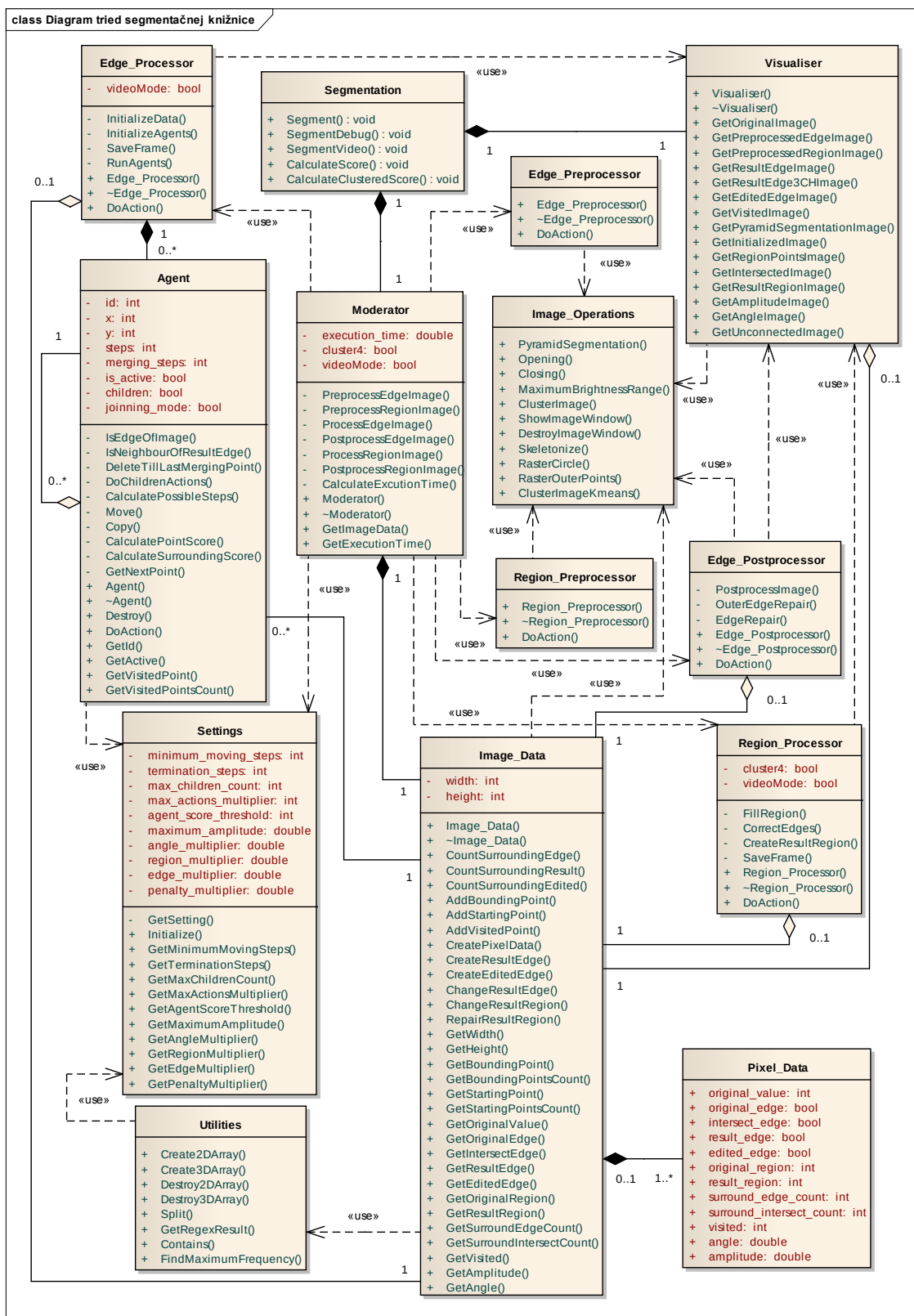
V tejto časti programu konfiguruje nastavenia segmentácie. Stlačením tlačidla *Edit settings* sa zobrazia nastavenia v predvolenom textovom editore. Tlačidlom *Revert settings to default* vrátime zmeny v nastaveniach na pôvodné hodnoty. Voľba *Cluster output image into 4 colours*

Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z

lekárskych snímok

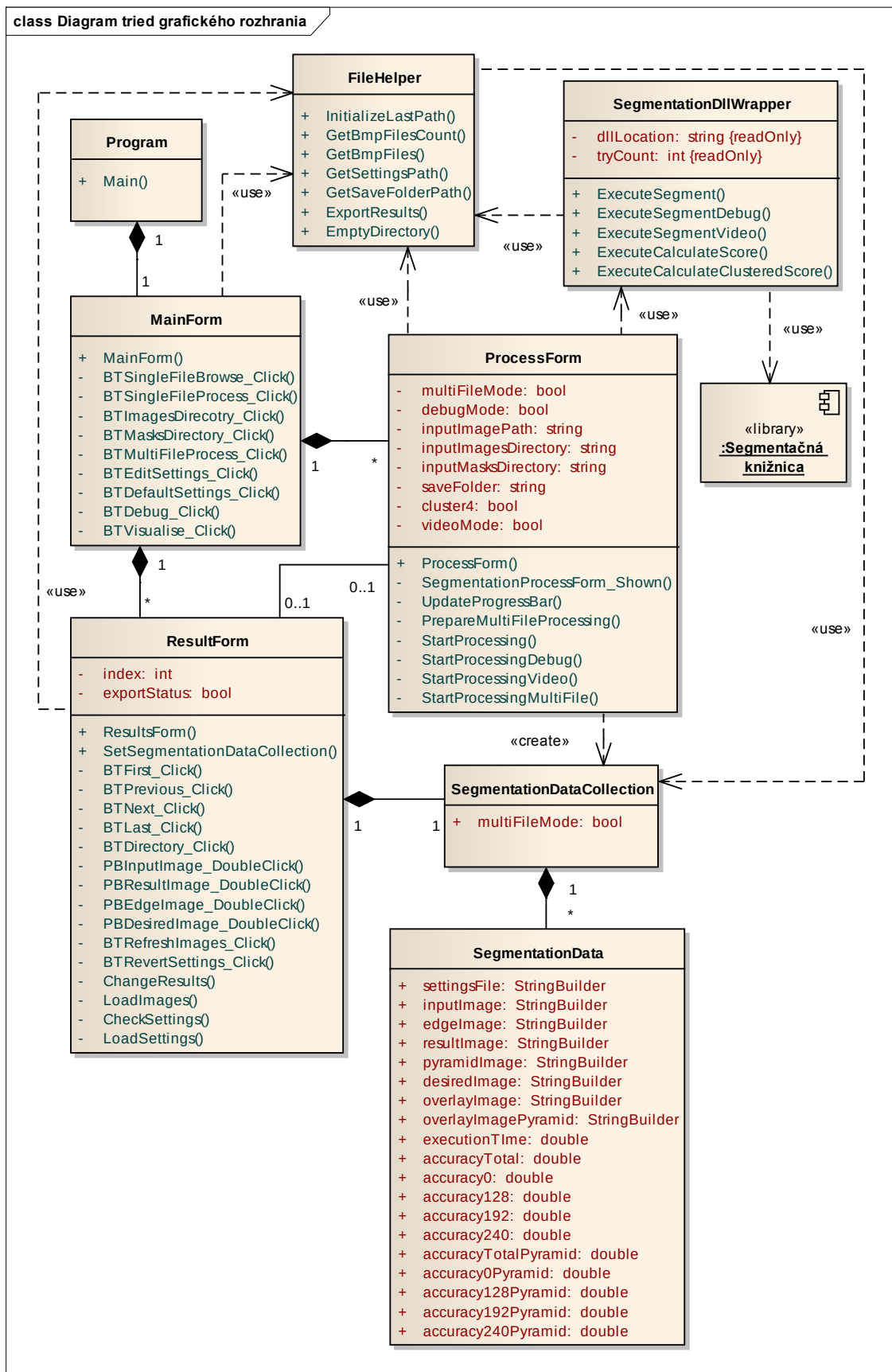
ovplyvňuje či sa má výsledný segmentovaný obraz rozdeliť iba do 4 rozdielnych typov regiónov, alebo nie. Táto vlastnosť je dôležitá pre porovnanie voči referenčným výsledkom, kde sa segmentovalo do štyroch farieb so zadefinovaným významom.

C Implementačná dokumentácia – diagramy tried



Obr. C.1 Diagram tried segmentačnej knižnice

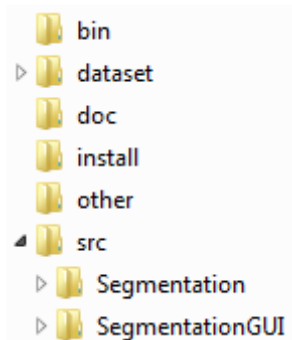
Segmentácia obrazu s použitím metód distribuovaných agentov pre získavanie informácie z lekárskych snímkov



Obrázok C.2 Diagram tried grafického rozhrania

D Obsah CD nosiča

Obsah CD nosiča je rozdelený do viacerých častí. V koreňovom adresári sa nachádza textový súbor (*Readme.txt*) opisujúci logické rozdelenie súborov (Obr. D.1).



Obr. D.1 Logická stromová štruktúra obsahu CD nosiča

V adresári *bin* sa nachádza spustiteľný program (*SegmentationGUI.exe*) vrátane potrebnej segmentačnej knižnice (*Segmentation.dll*), nastavení (*SegmentationGUI.exe.config*) a konfiguračného súboru segmentácie (*Settings.txt*).

V adresári *dataset* sa nachádzajú testovacie dátové množiny obrazov (*Dataset.zip*).

V adresári *doc* sa nachádza elektronická verzia dokumentácie diplomovej práce vo formáte pdf (*Diplomova_praca_Sobotka.pdf*) a návrhové diagramy v projekte pre Enterprise Architect (*Diplomova_praca_II_Sobotka.eap*).

V adresári *install* sa nachádza inštalátor OpenCV 2.2 (*OpenCV-2.2.0-win32-vs2010.exe*), inštalátor Microsoft Visual C++ 2010 Express (*vc_web.exe*), inštalátor Microsoft Visual C# 2010 Express (*vcs_web.exe*) a inštalátor .NET Framework 2.0 (*dotnetfx.exe*).

V adresári *other* sa nachádza program pre konverziu medicínskych snímok do formátu bmp (*GWRawMRIView.exe*) a skúšobná verzia programu Enterprise Architect (*easetup.exe*).

V adresári *src* sa nachádzajú zdrojové kódy segmentačnej knižnice (*adresár Segmentation*) a zdrojové kódy grafického rozhrania (*adresár SegmentationGUI*).

E Príspevok na konferenciu IIT.SRC 2012

Image Segmentation using Methods of Distributed Agents to Obtain Information from Medical Images

Marek SOBÔTKA*

*Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
melcor.sobotka@gmail.com*

Abstract. Nowadays, a large amount of information is worked with every day. Different systems for data processing have been created. These systems can be manual or fully automatic. Medicine is one area where it is necessary to process image data. Several devices are being used here that produce different types of output. Typical examples include X-rays or pictures obtained by the magnetic resonance imaging (MRI). An integral part of image processing is image segmentation. The aim of this work is to explore the possibilities of using a multi-agent system for image segmentation. Our system combines several types of basic image information to obtain the best results.

1 Introduction

Nowadays, a large amount of information is worked with every day. Different systems for data processing have been created. These systems can be manual or fully automatic. One type of data being processed is image data. Daily, countless numbers of photos are created and hours of videos are recorded around the world. With so many records there naturally arises the need for its recognition, sorting and further processing.

Medicine is one area where it is necessary to process image data. Several devices are being used here that produce different types of output. Typical examples include X-rays or pictures obtained by the magnetic resonance imaging (MRI). There is already a vast array of systems for image processing. An integral part of image processing is image segmentation.

Image segmentation is considered a long-term challenge in digital image processing. Its aim is to divide the input image into several disjoint homogeneous regions. These regions contain points of image that have a similar intensity, colour, texture or another aspect depending on what is going to be achieved by segmentation [1, 2, 3]. Thus obtained regions may be further processed at the level of recognizing objects. There are different methods by the usage of which different results are achieved.

The greatest problem of image segmentation is that it is problem-dependent. Segmentation of photos of nature containing dominating objects is rather different from that of medical images. Many objects, such as the human brain and heart, do not have visible borders in images from the medical field, images are naturally textured and there are no clear edges among objects [4]. Medical images show

* Master degree study programme in field: Software Engineering
Supervisor: Dr. Vanda Benešová, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

similar grey shades and similar textures among objects [3]. This causes smaller variance of intensity and colours in the image, making it more difficult to distinguish the objects contained in the image.

Image segmentation is used in medicine for example in the pathology in the brain surgery [3, 5], in the study of anatomical structures, in the planning of treatment or in the determination of a patient's diagnosis [3]. Elsewhere, doctors process images from X-ray and MRI automatically by human intelligence. There are efforts to process these images automatically by computer. As a result, physicians can confront their opinions with an assessment of a computer. Though to make it work, the algorithm used for segmentation of such images would have to achieve higher reliability. Several works are already addressing this problem, among others [3, 4, 5, 6, 7]. They have achieved relatively good results, but there is still room for improvement.

2 Proposed method

2.1 Multi-agent system for segmentation

Multi-agent system which we are researching combines several types of basic image information to obtain the best results. Centralized coordination of agents similar to that in [3] is used to make the solution simple. Agents will be controlled by a moderator, who will have all current information about the image and its properties, and thus will be able to coordinate image segmentation. First, the moderator creates the image data. The most important data include edges and regions.

The agents are initiated according to gathered information. Each agent will have access to the relevant global data about the image, as well as keep information about what he has carried out. So we are creating reactive agents with memory. The types of agents in our multi-agent system were inspired by [6, 7]:

- edge agents
- regional agents

The goal of the edge agents is to create a set of fully closed regions. Regional agents get this set and they are designed to colour each region correctly.

2.2 Moderator

Moderator is responsible for creating a segmented image for each input image. Every image starts an extensive process.

At first, data are initiated. This includes preparation of a special data structure for each pixel of the input image. The data structure contains following information:

- the original brightness value of the input grey-tone image
- the presence of the edge in the input image
- the presence of the edge in the starting image for agents
- the presence of the edge in the output segmented image
- regional classification of the pixel in the input image
- regional classification of the pixel in the output segmented image
- the number of pixels that are edges around the current pixel of the input image
- the number of pixels that are edges around the current pixel of the starting image for agents
- information if the pixel has been visited by the agent
- angle
- amplitude

By angle and amplitude we understand the values obtained using the Sobel operator, which is commonly used to detect edges. Amplitude is the strength of response if the current pixel is edge. The higher the value is, the more likely that the edge is present in that pixel. The angle is the direction of the edge.

Subsequently, agents are initialized to the specified locations in the image. Agents receive a starting image. This image is created by the intersection of edge information and regional information. Thus, we have input information confirmed by two independent methods, which increases the reliability of results. For each pixel we have calculated the number of edges in the neighbourhood. We can divide individual pixels according to this number into four groups:

1. *independent pixels* (see Figure 1 (a)) – pixels that have no edge in the neighbourhood
2. *end pixels* (see Figure 1 (b)) – pixels that have exactly one edge in the neighbourhood
3. *transit pixels* (see Figure 1 (c)) – pixels that have exactly two edges in the neighbourhood
4. *partition pixels* (see Figure 1 (d)) – pixels having more than two edges in the neighbourhood

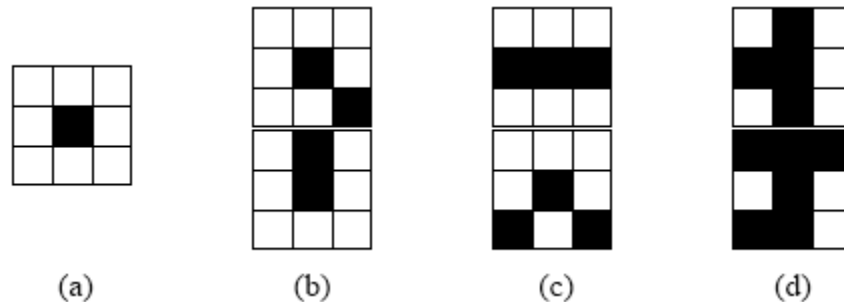


Figure 1. Categorisation of pixels according to the number of edges in the neighbourhood.

Agents take up positions of the ends of edges (end pixels) and all positions where there is a certain probability of crossing several edges (partition pixels). They are also initialized to the independent pixels because there is no guarantee that the agents will reach these positions in the process of image segmentation. Agents are, therefore, initiated to all pixels that are edges and where the number of edges in the neighbourhood is not equal to two (pixels are not transit).

Lastly, the moderator starts the agents. They operate using image data until agent has defined action or until agent has reached the maximum allowed amount of actions. The maximum amount of actions also serves as a safeguard against endless running of the agent. The results are entered directly into the global structure of image data.

2.3 Edge agents

Each agent is an autonomous entity that has defined actions to be taken using global data of the image and local data of the agent. Each agent has to remember some information, such as:

- identifier
- position in the image
- number of executed steps
- number of executed steps since the last partition
- essential flags (if agent is active, if agent is in joining mode, etc.)
- visited pixels of the image
- agent's children

Each agent can have their children which are created in the process of reproduction of the agent. In the process of image segmentation there may sometimes happen that the parent finishes its actions earlier than children, so it is necessary to ensure that agent's children could continue working uninterrupted, while they have actions that can be done. Therefore, if the agent is inactive, it is necessary to check if children are inactive.

Agent requires having actual data, to ensure correct functionality. Therefore, in each step agent must update its local data.

For a system as a whole moderator must have available information, when each agent completed its actions and can no longer continue in the image segmentation, or when each agent already completed all actions that were possible. That is why active flag is used. Moderator checks after each step, if there is at least one agent able to perform another action. When there is no further action for the agent to carry out, it marks itself as inactive.

Reactions of the agent can vary for different positions in the image. It depends on the number of possible movements, where the agent can move. This number is in the interval (0, 8). Agent checks whether the pixel is not on the boundary of the image, whether the pixel has never been visited by any agent and whether on the pixel there is edge for each of the eight neighbouring pixels. If the pixel meets all three conditions, it is included in the possible movements from the agent's current position.

Then the agent reacts according to the calculated number of possible steps for the current position of the agent:

1. Number of possible steps equals one. Then the agent moves to the specified point. The next action will depend on this new location.
2. The agent gets to the point from which it can go several directions (number of possible steps is more than one). Then the agent cannot determine which way is the best. Therefore, it has to reproduce itself. It chooses one way where to move. The agent creates children for each other way and settles them there. In the next step these children will behave just like any other agent. The behaviour is independent of the fact, whether the agent is parent or child.
3. Number of possible steps equals zero. Then the agent switches to joining mode. In this mode agent does not move on already existing edges in the image, but it starts creating new edges or destroying the existing edges, depending on the situation. This action directly modifies global structure of image data. Agent changes information on its current position.

Every agent has the limitation as to how many times it should move since it was reproduced. The edge is removed from the global data structure of the image if the number of moves is less than the limit. This action ensures that all small edges in the image are deleted and only the important ones remain.

When the agent is creating a new edge, we need to use some heuristics that tell which one of the neighbouring pixels is most likely an edge. For this purpose, we designed the function counting the scores of each pixel in the neighbourhood:

$$ps = am * ca + rm * cr + em * eTP + pm * ep \quad (1)$$

where:

$$ca = \left(\frac{\min(ampTP, ma)}{ma} \right) * \left(- \left(\frac{\min(|angAP - angTP|, |angAP - angTP| - 360|)}{360} - 1 \right) \right) \quad (2)$$

$$cr = \left(- \left(\left(\frac{|origAP - origTP|}{255} \right) - 1 \right) \right) * regTP \quad (3)$$

$$ep = -resTP - startTP \quad (4)$$

Legend to formulas (1), (2), (3), (4):

- ps – the final score of the pixel
- $am/rm/em/pm$ – *controlled angle/controlled region/edge/penalty* multiplier obtained from settings
- $eTP/regTP/resTP/startTP$ – flag if there is *edge/boundary between two regions/output edge/edge in the starting image for agents* at the target point (if yes flag = 1, if no flag = 0)
- $ampTP$ – amplitude value at the target point obtained from the global data of the image
- ma – the maximum amplitude obtained from settings
- $angAP/angTP$ – angle in degrees at the *current/target* point obtained from the global data
- $origAP/origTP$ – brightness value of the original image at the *current/target* point

From the surrounding pixels the point with the highest score is selected. Then the agent moves to that location if the score is greater than the minimum threshold set in the settings, otherwise the agent destroys itself. This method combines various sources (amplitude, angle, regional image, edge image, original image, etc.). We can regulate the contributions of individual components to the result by adjusting the weights (multipliers) for each component. The idea of using penalties is to compel agents to search for alternative edges. Thanks to the fact that points of existing edges get a negative score, the agent will most likely choose another position instead, where there is a higher score.

2.4 Regional agents

At the time of writing this paper the multi-agent system was still in development. Therefore, regional agents have not been implemented yet. Individual regions are currently coloured by calculating the average brightness of all pixels in the region.

3 Results

Implemented multi-agent system can create edge image and segmented image from the input medical image (see Figure 2). This is possible by using edge agents. These agents can successfully add significant edges and delete insignificant short edges, which are often just noise. However, it has turned out that our solution still has several problems, with which we will deal soon.



Figure 2. Example of multi-agent system output.

The first problem is that in the output segmented image there are not all the regions. The agents have not closed all regions which causes problems with colouring. We will address this problem by creating special version of edge agents that will ensure that all the regions are closed in all circumstances.

The second problem is that the resulting regions are not correctly classified into groups and, thus, not correctly coloured. This problem will be eliminated by creating regional agents. These agents will be responsible for colouring each region by multiple criteria such as size of the region, colour of neighbouring regions, the histogram of brightness of the region, average brightness of the points in the region, etc. We will experiment to get desirable results.

We are interested in the performance of our multi-agent system, as well. The solution was tested on a dataset containing 55 images at a resolution of 256x256 pixels. The test was run three times and we achieved the average value of image segmentation 525.28 ms for a single picture.

4 Conclusions

In this paper, an automatic medical image segmentation method has been presented. This method tries to combine several basic image information of an image, such as edge information and regional information, to obtain the desirable results.

The results suggest that the method currently faces a number of open problems. The output segmented image sometimes contains certain improperly coloured regions. This is caused by unclosed regions in edge image. We are aware of these problems and we are working on the solutions in order to eliminate them.

Final method will be evaluated on the reference dataset¹ and the segmentation results will be compared to other methods.

References

- [1] Wei, B.C., Mandava, R.: Multi-objective nature-inspired clustering techniques for image segmentation. In: *Cybernetics and Intelligent Systems (CIS), 2010 IEEE Conference on*, Singapore (June 2010), 150-155.
- [2] Varshney, S.S., Rajpal, N., Purwar, R.: Comparative study of image segmentation techniques and object matching using segmentation. In: *Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*, Delhi (December 2009), 1-6.
- [3] Chitsaz, M., Seng, W.C.: A Multi-agent System Approach for Medical Image Segmentation. In: *Future Computer and Communication, 2009. ICFCC 2009. International Conference on*, Kuala Lumpur (April 2009), 408-411.

¹ Internet Brain Segmentation Repository – <http://www.cma.mgh.harvard.edu/ibsr/>

- [4] Liang, J., Xu, Z., Mao, D., Cheng, X.: Multi-agent based evolutionary algorithm in medicine image segmentation. In: *Journal of Communication and Computer*, vol. 6, June 2009, no. 6 (serial no. 55), ISSN 1548-7709, 31-35.
- [5] Shariatpanahi, H.F., Batmanghelich, N., Kermani, A.R.M., Ahmadabadi, M.N., Soltanian-Zadeh, H.: Distributed Behavior-based Multi-agent System for Automatic Segmentation of Brain MR Images. In: *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, Vancouver (October 2006), 4535-4542.
- [6] Idir, K., Merouani, H., Tlili Y.: Dual Pyramid of Agents for Image Segmentation. In: *World Academy of Science, Engineering and Technology*, vol. 6, June 2005, no. 47, 198-201.
- [7] Duchesnay, E., Montois, J.-J., Jacquélet, Y., Kinie, A.: An Agent-based Implementation of Irregular Pyramid for Distributed Image Segmentation. In: *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, Antibes-Juan les Pins (October 2001), 409-415.