

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**  
**STAVEBNÁ FAKULTA**

**Automatické segmentačné metódy biologických dát**

**Diplomová práca**

Študijný program:	Matematicko-počítačové modelovanie
Študijný odbor:	Aplikovaná matematika
Školiace pracovisko:	Katedra matematiky a deskriptívnej geometrie
Vedúci diplomovej práce:	doc. RNDr. Zuzana Krivá, PhD.

**BRATISLAVA 2020**  
**Bc. Mária Somorovská**

## Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Segmentácie obrazov</b>	<b>3</b>
2.1	Globálne prahovanie . . . . .	4
2.1.1	Otsuho metóda . . . . .	4
2.1.2	Prahovanie pomocou maximálnej entropie . . . . .	5
2.2	Lokálne adaptívne prahovanie . . . . .	6
2.2.1	Bernsenova metóda . . . . .	6
2.2.2	Niblackova metóda . . . . .	6
2.3	Metóda subjektívnych plôch (SUBSURF) . . . . .	6
<b>3</b>	<b>Softvér</b>	<b>7</b>
3.1	Qt . . . . .	7
3.1.1	VTK . . . . .	8
<b>4</b>	<b>Výsledky</b>	<b>8</b>
<b>5</b>	<b>Záver</b>	<b>8</b>

# 1 Úvod

V rôznych vedných disciplínach ale aj v bežnom živote sa používajú rôzne aplikácie spracovania obrazov. Spracovanie obrazov je metóda, ktorá pomocou rôznych matematických operácií a algoritmov upravuje obrazové dáta rôznych formátov a pomáha z nich získavať užitočné informácie. Obrazové dáta je potrebné zobrazit', či už pred alebo po modifikácií.

Cieľom práce je vytvorenie softvéru, ktorý slúži na vizualizáciu a segmentáciu obrazov získaných konfokálnym laserovým mikroskopom, konkrétne sa jedná o biologické dáta a to obrazy makrofágov. Takéto dáta môžu obsahovať šum, ktorý je potrebné odstrániť pre lepšie rozpoznanie objektov na dátach.

Z biologického hľadiska, je makrofág typ bielej krvinky, ktorá hrá dôležitú úlohu pri ochrane imunitného systému a homeostázy. Avšak disfunkčné makrofágy menia svoje účinky a u ľudí môžu spôsobovať závažné ochorenia ako sú napríklad zápalové ochorenia, ktoré vedú k častým infekciám alebo sa môžu podieľať na postupe rakoviny. Makrofág mení svoj tvar keď sa približuje smerom k rane. Táto zmena tvaru je spôsobená objektami, ktoré sa nachádzajú v blízkosti makrofágu, ako napríklad tkanivovými bunkami alebo medzibunkovou hmotou. Segmentácia obrazov môže byť užitočným nástrojom na porozumenie spôsobu interakcie medzi makrofágmi a bunkami, ktoré ho obklopujú, avšak takáto segmentácia môže byť náročnou úlohou, kvôli ich nepravidelnému tvaru.

Mikroskopové dáta makrofágov, s ktorými pracujeme v tejto práci, sú makrofágy priesvitného embrya zebričky pruhovanej (*lat. danio rerio*). Táto larva bola poranená a cytoplazmy makrofágov sú zafarbené zeleným svetielkujúcim proteínom (kaede) pre lepšiu viditeľnosť pod mikroskopom. Pôvodné dáta makrofágov sú získané v časovom úseku 5 hodín a s časovým krokom 4 minúty. Následne dané trojdimenzionálne obrazové dáta sú premietnuté do roviny za použitia maximálnej intenzity približne zo 70 rezov, z ktorých vzniknú dvojdimenzionálne obrazové dáta.

V tejto práci sa zaoberáme časťami/výrezmi takýchto dát, na ktorých sa nachádza jeden makrofág a používame rôzne metódy na segmentáciu, buď automatické alebo semi-automatické. Softvér by mal byť intuitívny, a teda určený aj užívateľom, ktorí implementovaným algoritmom nemusia rozumieť.

Práca je rozdelená do viacerých častí, v ktorých je podrobnejšie popísaná funkcionálna programová časť spolu s užívateľským prostredím, použitými algoritmi, knižnicami.

Prvá časť je teoretická a venuje sa matematickým algoritmom využitých pri implementácii, založených na poznatkoch zo spracovania obrazov. Jedná sa o niekoľko automatických a semi-automatických segmentačných metód, ktoré sú kombináciou prahovacích metód a segmentačnej metódy subjektívnych plôch (SUBSURF).

Druhá časť sa zameriava na technológie a knižnice využité pri implementácii. Nachádza sa tam popis Qt knižnic, ktoré boli použité pri vytváraní užívateľského prostredia, VTK knižnic, ktoré boli využité na zobrazenie a manipuláciu s dátami. Popísané sú tu aj triedy, ktoré boli v programe najviac využité.

Ďalšia časť sa zaoberá popisom grafického rozhrania programu, ktorá by mohla slúžiť aj ako manuál slúžiaci užívateľovi pri používaní.

V poslednej časti sa nachádzajú výsledky, ku ktorým sme v práci dospeli a porovnania medzi rôznymi automatickými a semi-automatickými segmentačnými metódami.

## 2 Segmentácie obrazov

Hlavnou úlohou pri segmentácii makrofágov, je rozlíšenie pozadia a objektu na obrazových dátach. Táto úloha môže byť sťažená kvôli strate intenzity na hranách alebo šumu, ktorý sa môže na dátach nachádzať.

Segmentácia takýchto dát je náročnou úlohou pretože makrofágy majú nepravidelné tvary s meniacou sa intenzitou, ktoré sa ťažko spracovávajú. Preto sme vybrali niekoľko druhov prahovacích metód, ktoré sme skombinovali spolu s metódou segmentácie subjektívnych plôch a aplikovali sme ich na testované dáta.

## 2.1 Globálne prahovanie

Úlohou globálnych prahovacích metód je nájsť jedinou optimálnu prahovú hodnotu  $q$ , v našom prípade šedotónových obrazových dát, ktorá zadefinuje každý pixel obrazu buď do popredia (ako objekt na dátach) alebo ako pozadie. Mnohé z týchto metód sú založené na histograme. Teda všetky pixely sú zatriedené do dvoch disjunktných množín  $C_0$  a  $C_1$ , kde množina  $C_0$  obsahuje všetky pixele nachádzajúce sa medzi hodnotami  $(0, 1, \dots, q)$  a  $C_1$  obsahuje všetky zvyšné pixele nachádzajúce sa na intervale  $(q + 1, \dots, K - 1)$ , teda

$$(u, v) \in \begin{cases} C_0 & \text{ak } I(u, v) \leq q \quad (\text{pozadie}) \\ C_1 & \text{ak } I(u, v) \geq q \quad (\text{objekt}) \end{cases}.$$

Treba si uvedomiť, že tieto hodnoty závisia od toho, či je pozadie bledé a objekt tmavý alebo naopak.

Prahovacie metódy založené na histograme sú zvyčajne jednoduché a účinné, pretože pracujú s malým množstvom dát. V našom prípade sa jedná o 256 odtieňov sivej/šede. Dajú sa rozdeliť na 2 hlavné kategórie: štatistické metódy a také, ktoré sú založené na tvare.

### 2.1.1 Otsuho metóda

Otsuho metóda patrí medzi automatické prahovacie metódy, ktorá rozdeľuje obrazové dáta na 2 rôzne triedy pomocou prahu  $q$  – na objekt a pozadie. Hlavnou myšlienkou tejto metódy je nájsť prah  $q$  taký, že výsledné distribúcie tried sú čo najlepšie oddelené, čo znamená, že príslušné histogramy majú čo najmenší rozptyl (sú čo njužšie). Na výpočet prahu  $q$  sa používa metóda známa ako vnútro–triedna variancia (within class variance), kde sa pomocou tohoto výpočtu hľadá minimum. V tomto prípade sa dá ukázať, že táto úloha môže byť zmenená na maximalizačnú úlohu, tiež známou ako medzi–triedna variancia (between class variance), ktorá je výpočtovo menej náročná, keďže sú spracovávané len šedotónové obrazové dáta. Ak by išlo o farebný RGB obraz, musel by byť rozdelený na jednotlivé intenzity a výsledkom by bolo viac prahových hodnôt  $(q_1, \dots, q_n)$ , kde  $n$  reprezentuje počet intenzít v danom obraze.

Nech  $K$  je maximálna intenzita obrazu a hodnoty normalizovaného histogramu, budú vypočítané ako

$$p_i = \frac{h_i}{N}, \quad \sum_{i=1}^N p_i = 1,$$

kde  $i$  je konkrétny level intenzity ( $0 \leq i \leq K$ ),  $N$  je celkový počet pixelov na obraze,  $h_i$  predstavuje histogram. Keďže v našom prípade máme len 2 triedy  $C_0$  a  $C_1$  a vzniknutý histogram sa tiež nazýva bimodálnym.

Na definíciu tried  $C_0$  a  $C_1$  je potrebné vypočítať stredné hodnoty  $\mu_0$  a  $\mu_1$  definované ako

$$\mu_0 = \sum_{i=1}^q \frac{ip_i}{\omega_0(q)}, \quad \mu_1 = \sum_{i=q+1}^K \frac{ip_i}{\omega_1(q)},$$

kde  $\omega_0(q)$  a  $\omega_1(q)$  sú rozdelenia pravdepodobnosti pre triedy  $C_0$  a  $C_1$ . Teraz môžeme zadefinovať rozptyl pre obe triedy

$$\sigma_0 = \omega_0(\mu_0 + \mu_T), \quad \sigma_1 = \omega_1(\mu_1 + \mu_T),$$

kde  $\mu_T = \omega_0\mu_0 + \omega_1\mu_1$ . Následne zadefinujeme Otsuho medzi-triednu varianciu ako

$$\sigma^2 = \sigma_0 + \sigma_1.$$

Z hodnoty medzi-triednej variance, pomocou maximalizácie rozptylu medzi pozadím a objektom v histograme, nájdeme optimálny prah  $q^*$ , ktorý je definovaný ako

$$\sigma^2(q^*) = \max_{1 \leq q < K} \sigma_1^2(q),$$

kde  $\sigma$  označuje rozptyl a  $K$  je maximálna intenzita obrazu.

Otsuho metóda dokáže dobre rozlíšiť dáta s makrofágmi, v prípade že dáta neobsahujú výrazný šum aj v prípade keď sa na dátach nachádzajú tenké časti alebo sú zložito tvarované. Avšak ak je intenzita šumu pozadia porovnateľná s intenzitou, táto metóda môže spôsobiť rozdelenie objektu a stratiť niektoré časti objektu, keďže je do úvahy braná len intenzita obrazu.

### 2.1.2 Prahovanie pomocou maximálnej entropie

Entropia je dôležitým pojmom v teórii informácií a najmä pri kompresii dát. Je to štatistická miera, ktorá kvantifikuje priemerné množstvo informácií obsiahnutých v "správe" obsahujúce stochasticky generované dáta. Entropia je definovaná ako

$$H(I) = - \sum_{u,v} p(g) \log_b(p(g)),$$

kde  $g$  je intenzita v pixli  $u,v$ ,  $p(g)$  je pravdepodobnosť intenzity v normalizovanom histograme,  $b$  je logaritmický základ, ktorý zvolíme buď  $b = 10$  alebo  $b = e$  aby boli dosiahnuté čo najlepšie výsledky. Hodnota entropie  $H$  bude vždy nadobúdať kladnú hodnotu, pretože argument logaritmu, sú pravdepodobnosti, ktoré patria intervalu  $(0,1)$  z čoho vyplýva že hodnotaa logaritmu bude vždy záporná.

Z dôvodu hľadania maximálnej hodnoty entropie, potrebujeme definovať entropie pre každú triedu

$$H_0(q) = - \frac{1}{P_0(q)} S_0(q) + \log(P_0(q))$$

$$H_1(q) = - \frac{1}{1 - P_0(q)} S_1(q) + \log(1 - P_0(q)),$$

kde  $P_0$  predstavuje kumulatívnu pravdepodobnosť a  $S_0, S_1$  sú sumačné podmienky. Kumulatívna pravdepodobnosť  $P_0$  je definovaná ako

$$P_0(q) = \begin{cases} p(0) & \text{pre } q = 0 \\ P_0(q-1) + p(q) & \text{pre } 0 < q < K, \end{cases}$$

a sumačné podmienky  $S_0, S_1$  sú predpočítané a definované ako

$$S_0(q) = \begin{cases} p(0) \cdot \log(p(0)) & \text{pre } q = 0 \\ S_0(q-1) + p(q) \log(p(q)) & \text{pre } 0 < q < K \end{cases}$$

$$S_1(q) = \begin{cases} 0 & \text{pre } q = L - 1 \\ S_0(q + 1) + p(q + 1)\log(p(q + 1)) & \text{pre } 0 \leq q < K - 1 \end{cases}$$

Táto metóda je jednoduchá a účinná, pretože závisí len od histogarmu obrazu. Tiež znama ako Kapurova (strana 20 je tam poznámka) \*PRIDAT AKO SA RATA CELKOVA ENTROPIA + \*

## 2.2 Lokálne adaptívne prahovanie

Lokálne adaptívne prahovanie namiesto jednej prahovej hodnoty pre celý obraz, používa adaptívne prahovanie, ktoré určuje meniacu sa prahovú hodnotu  $Q(u, v)$  pre každú polohu obrazu. Tieto hodnoty zodpovedajú každému pixelu  $I(u, v)$  zodpovedajúcemu danému obrazu. Nasledujúce metódy sa líšia iba s ohľadom na to, akým spôsobom sú získané prahy  $Q$  zo vstupného obrázku.

### 2.2.1 Bernsenova metóda

Táto metóda, určuje prah dynamicky pre každú polohu na obraze  $(u, v)$ , založená na minimálnej a maximálnej intenzite nachádzajúcej sa v okolí  $R(u, v)$ . Ak

$$\begin{aligned} I_{min}(u, v) &= \min_{(i,j) \in R(u,v)} I(i, j), \\ I_{max}(u, v) &= \max_{(i,j) \in R(u,v)} I(i, j), \end{aligned} \quad (1)$$

sú minimálnou a maximálnou hodnotou intenzity, na nejakom fixne danom okolí  $R$  so stredom na pozícií  $(u, v)$ . Prahovú hodnotu dostaneme pomocou aritmetického priemeru nájdeného minima a maxima daného okolia

$$Q(u, v) \leftarrow \frac{I_{min}(u, v) + I_{max}(u, v)}{2} \quad (2)$$

Táto operácia sa vykonáva tak dlho, až pokým lokálny kontrast  $c(u, v) = I_{max}(u, v) - I_{min}(u, v)$  sa nachádza nad preddefinovaným limtom  $c_{min}$ . Ak  $c(u, v) < c_{min}$ , tak pixle..

\*DOPIŠAT\*

### 2.2.2 Niblackova metóda

\*rovnica vedenia tepla + diskretizácia\*

## 2.3 Metóda subjektívnych plôch (SUBSURF)

\*POPIŠAT\*

Klasická metóda subjektívnych plôch je definovaná ako

$$u_t = \sqrt{\varepsilon^2 + |\nabla u|^2} \nabla \cdot (g \frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}), \quad (3)$$

$$\frac{1}{|\nabla u^{n-1}|_\varepsilon} \cdot \frac{u^n - u^{n-1}}{\tau} = \nabla \cdot (g^0 \frac{\nabla u^n}{|\nabla u^{n-1}|_\varepsilon}), \quad (4)$$

$$\int_p \frac{1}{|\nabla u^{n-1}|_\varepsilon} \cdot \frac{u^n - u^{n-1}}{\tau} dx = \int_p \nabla \cdot (g^0 \frac{\nabla u^n}{|\nabla u_p^{n-1}|_\varepsilon}) dx, \quad (5)$$

Green ->

$$\int_p \frac{1}{|\nabla u^{n-1}|_\varepsilon} \cdot \frac{u_p^n - u_p^{n-1}}{\tau} dx = \int_{\partial p} g^0 \frac{\nabla u^n}{|\nabla u_p^{n-1}|_\varepsilon} \vec{n}_p ds, \quad (6)$$

$$\frac{m(p)}{|\nabla u_p^{n-1}|_{\varepsilon\tau}}(u_p^n - u_p^{n-1}) = \sum_{q \in N(p)} \int_{e_{pq}} g_{pq}^0 \frac{\nabla u^n}{|\nabla u_{pq}^{n-1}|_{\varepsilon}} \vec{n}_{pq} ds, \quad (7)$$

$$\frac{m(p)}{|\nabla u_p^{n-1}|_{\varepsilon\tau}}(u_p^n - u_p^{n-1}) = \sum_{q \in N(p)} \int_{e_{pq}} g_{pq}^0 \frac{u_q^n - u_p^n}{d_{pq}} \cdot \frac{1}{|\nabla u_{pq}^{n-1}|_{\varepsilon}} ds, \quad (8)$$

$$\frac{m(p)}{|\nabla u_p^{n-1}|_{\varepsilon\tau}}(u_p^n - u_p^{n-1}) = \sum_{q \in N(p)} g_{pq}^0 T_{pq} \frac{1}{|\nabla u_{pq}^{n-1}|_{\varepsilon}} (u_p^n - u_p^{n-1}), \quad (9)$$

$$u_p^n - u_p^{n-1} = \frac{|\nabla u_p^{n-1}|_{\varepsilon\tau}}{m(p)} \sum_{q \in N(p)} g_{pq}^0 T_{pq} \frac{1}{|\nabla u_{pq}^{n-1}|_{\varepsilon}} (u_p^n - u_p^{n-1}), \quad (10)$$

$$u + p^{n-1} = u_p^n + \sum_{q \in N(p)} \frac{|\nabla u_p^{n-1}|_{\varepsilon\tau}}{m(p)} \frac{g_{pq}^0 T_{pq}}{|\nabla u_{pq}^{n-1}|_{\varepsilon}} (u_p^n - u_p^{n-1}), \quad (11)$$

$$u + p^{n-1} = (1 + \sum a_{pq}) u_p^n - \sum a_{pq} u_q^n, \quad (12)$$

### 3 Softvér

Na implementáciu a vytvorenie prostredia, bol zvolený objektovo orientovaný prístup jazyka C++, spolu s knižnicami Qt, ktoré obsahujú veľa užitočných tried a boli užitočným nástrojom pri vytváraní užívateľského prostredia a VTK knižnicami, ktoré slúžia na zobrazovanie a manipuláciu s dátami.

#### 3.1 Qt

Užívateľské rozhranie je vytvorené pomocou Qt knižníc, ktoré sú jedným z najpoužívanějších cross-platformových frameworkov na vytváranie užívateľského prostredia (GUI). Majú aj veľa predprogramovaných knižníc ktoré programátorovi uľahčia prácu. Sú naimplementované v jazyku C++.

\*doplnit ako funguju signaly a sloty\*

Najčastejšie využívané Qt knižnice v projekte:

- **QMdiArea**

Táto trieda zohráva jednu z najdôležitejších funkcií v programe. Funkcie tejto triedy fungujú v podstate ako správca okien pre MDI okná, čo v našom prípade znamená, že umožňuje vytvárať podokná pomocou triedy, v ktorých sa v programe nachádzajú ďalšie Qt triedy slúžiace na vykresľovanie 2D a 3D dát, s ktorými program pracuje. V programe je použité kaskádové usporiadanie takýchto podokien, čo znamená že vykresľovacie okná sa môžu navzájom prekrývať, dajú sa minimalizovať/maximalizovať v rámci hlavného okna a zavrieť.

- **QScrollArea**

QScrollArea sa nachádza v každom podokne widgetu QMidiArea. Zabezpečuje možnosť priblížiť/oddialiť a posúvať vizualizované dáta.

- **QVTKOpenGLNativeWidget**

Widget tejto triedy sa nachádza v každej QScrollArea a umožňuje samotné vykreslenie 2D/3D modelov za pomoci VTK knižníc.

- **QDockWidget**

Tento widget obsahuje všetky informácie a nastavenia súvisiace s dátami. Každý logický celok má vlastný 'dock', ktorý sa dá v rámci okna premiestňovať a ukotvovať buď na ľavej alebo na pravej strane okna. Taktiež sa dajú v prípade potreby minimalizovať.

- **QTreeWidget**

Všetky dáta, či už pôvodné alebo vysegmentované pomocou programu, sa nachádzajú v zozname, z ktorého sa dá vybrať ktoré dáta budú vykreslené. QTreeWidget bol použitý aby sme vykresľované dáta vedeli zadeliť do logických celkov, napríklad či ide o 2D alebo 3D dáta.

- **QVector**

Trieda QVector definuje dynamické polia, je šablónovou triedou. Ukladá premenné do susedných miesta v pamäti a poskytuje rýchly indexový prístup. Je použitá v prípadoch keď nie je potrebné odstraňovať prvky zvnútra QVectora.

- **QFile, QFileDialog**

Tieto triedy slúžia v programe na otváranie, načítavanie, ukladanie a manipuláciu s dátami.

### 3.1.1 VTK

Visualization Toolkit (VTK) sú vôľne dostupnými knižnicami, ktoré v programe slúžia na zobrazovanie a interakciu s 2D aj 3D dátami. Najviac využité knižnice:

- **vtkSmartPointer**

Táto šablónová trieda, slúži ako pointer nre VTK triedy. Jehou úlohou je zlepšiť manažment s pamäťou, čo znamená, že v prípade ak sú dáta mimo rozsahu alebo sa nikde nepoužívajú tak budú automaticky odstránené. Teda uľahčuje pracovať s dátami bez varovných hlások.

- **vtkPolyData**

V objektoch tejto triedy sú zadané vykresľované dáta, či už sa jedná o 2D alebo 3D dáta. V tejto triede môžu byť uložené informácie o tom akým spôsobom budú dáta reprezentované - geometrické informácie o štruktúre vykresľovaných dát. Takýmito informáciami môžu byť napríklad body, bunky, vektory, čiary, polygonálne alebo trojuholníkové pásy.

- **vtkPolyDataMapper**

- **vtkActor**

- **vtkRenderer**

## 4 Výsledky

## 5 Záver

## Literatúra