

**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

1	Úvod	3
2	Segmentácie obrazov	4
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	7
2.3	Metóda subjektívnych plôch (SUBSURF)	8
3	Numerické schémy	9
3.1	Implicitná metóda pre rovnicu vedenia tepla	9
3.2	Numerická schéma pre SUBSURF	10
4	Softvér	13
4.1	Qt	13
4.2	VTK	14
4.3	Rozdelenie projektu do tried	15
4.4	Grafické užívateľské rozhranie	16
5	Výsledky	20
6	Záver	24

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 zobrazíť, 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 rozoznanie objektov na dátach.

Z biologického hľadiska, je makrofág typ bielej krvinky, ktorá hrá dôležitú úlohu pri ochrane imunitného systému a hemostázy. 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 dvoj-dimenzionálne 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 algoritmami, 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 knižníc, ktoré boli použité pri vytváraní užívateľského prostredia, VTK knižníc, 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 porovnanie 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 šedo-tónových 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 pixle nachádzajúce sa medzi hodnotami $(0, 1, \dots, q)$ a C_1 obsahuje všetky zvyšné pixle nachádzajúce sa na intervale $(q + 1, \dots, K - 1)$, teda

$$(u, v) \in \begin{cases} C_0 & \text{ak } I(u, v) \leq q \quad (\text{pozadie}) \\ C_1 & \text{ak } I(u, v) \geq q \quad (\text{objekt}) \end{cases}. \quad (1)$$

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 najužšie). 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 známu ako medzi–triedna variancia (between class variance), ktorá je výpočtovo menej náročná, keďže sú spracovávané len šedo-tónové 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

$$p_i = \frac{h_i}{N}, \quad \sum_{i=1}^N p_i = 1, \quad (2)$$

kde i je konkrétny level intenzity ($0 \leq i \leq K$), N je celkový počet pixlov 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 bi-modálnym.

Na definíciu tried C_0 a C_1 je potrebné vypočítať stredné hodnoty μ_0 a μ_1 definované ako

$$\mu_0 = \sum_{i=1}^q \frac{ip_i}{\omega_0(q)}, \quad \mu_1 = \sum_{i=q+1}^K \frac{ip_i}{\omega_1(q)}, \quad (3)$$

kde $\omega_0(q)$ a $\omega_1(q)$ sú sumy definované ako

$$\omega_0(q) = \sum_{i=1}^q p_i, \quad \omega_1(q) = \sum_{i=q+1}^K p_i, \quad (4)$$

rozdelenia pravdepodobnosti pre triedy C_0 a C_1 . Následne vypočítame smerodajné odchýlky σ_0 a σ_1 , pomocou ktorých vieme zdefinovať medzi-triednu varianciu ako sumu

$$\sigma^2 = \sigma_0 + \sigma_1. \quad (5)$$

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

$$\sigma^2(q^*) = \max_{1 \leq q < K} \sigma_1^2(q), \quad (6)$$

kde σ 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

$$H(I) = - \sum_{u,v} p(g) \log_b(p(g)), \quad (7)$$

kde g je intenzita v pixle 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 hodnota logaritmu bude vždy záporná.

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

$$H_0(q) = -\frac{1}{P_0(q)}S_0(q) + \log(P_0(q)) \quad (8)$$

$$H_1(q) = -\frac{1}{1 - P_0(q)}S_1(q) + \log(1 - P_0(q)), \quad (9)$$

kde P_0 predstavuje kumulatívnu pravdepodobnosť a S_0, S_1 sú vopred vyrátané sumy.

Celková entropia pre daný prah q je daná ako

$$H(q) = H_0(q) + H_1(q); \quad (10)$$

Kumulatívna pravdepodobnosť P_0 je definovaná ako

$$P_0(q) = \begin{cases} p(0) & \text{pre } q = 0 \\ P_0(q-1) + p(q) & \text{pre } 0 < q < K, \end{cases} \quad (11)$$

a sumačné podmienky S_0, S_1 sú predpočítané a definované ako

$$S_0(q) = \begin{cases} p(0) \cdot \log(p(0)) & \text{pre } q = 0 \\ S_0(q-1) + p(q) \log(p(q)) & \text{pre } 0 < q < K \end{cases} \quad (12)$$

$$S_1(q) = \begin{cases} 0 & \text{pre } q = L-1 \\ S_0(q+1) + p(q+1) \log(p(q+1)) & \text{pre } 0 \leq q < K-1 \end{cases} \quad (13)$$

Táto metóda je jednoduchá a účinná, pretože závisí len od histogarmu obrazu. Entropia ako kritérium na voľbu prahu v obrazových dátach má dlhú tradíciu a navrhnutých bolo viacero metód. Vyššie uvedená metóda je jednou zo starších metód a bola navrhnutá matematikom J. N. Kapurom.

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 pixlu $I(u, v)$ 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, ktorá určuje prah dynamicky pre každú polohu na obrazových dátach (u, v) , je založená na minimálnej a maximálnej intenzite nachádzajúcej sa v okolí $R(u, v)$. Ak

$$\begin{aligned} I_{min}(u, v) &= \min_{(i,j) \in R(u,v)} I(i, j), \\ I_{max}(u, v) &= \max_{(i,j) \in R(u,v)} I(i, j), \end{aligned} \quad (14)$$

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

$$Q(u, v) \leftarrow \frac{I_{\min}(u, v) + I_{\max}(u, v)}{2} \quad (15)$$

Táto operácia je vykonávaná tak dlho, až kým lokálny kontrast $c(u, v) = I_{\max}(u, v) - I_{\min}(u, v)$ nie je väčší ako preddefinovaný limit c_{\min} . Ak $c(u, v) < c_{\min}$, tak predpokladáme, že pixle zodpovedajúce jednej oblasti patria do tej istej triedy a sú automaticky priradené do pozadia.

2.2.2 Niblackova metóda

Prah $Q(u, v)$ pri Niblackovej metóde sa mení ako funkcia lokálneho priemeru intenzít $\mu_R(u, v)$ a smerodajnej odchýlky $\sigma_R(u, v)$, v tvare

$$Q(u, v) = \mu_R(u, v) + \kappa \sigma_R(u, v). \quad (16)$$

Lokálny prah je určený pridaním konštanty $\kappa \geq 0$ k smerodajnej odchýlke $\sigma_R(u, v)$ a lokálneho priemeru $\mu_R(u, v)$. Lokálne hodnoty smerodajnej odchýlky $\sigma_R(u, v)$ dostaneme ako

$$\sigma_R(u, v) = \frac{1}{N} \sum_{(i,j) \in R(u,v)} (I(i, j) + \bar{I}(i, j))^2, \quad (17)$$

kde R označuje fixne dané okolie, so stredom v (u, v) , N je počet prvkov nachádzajúcich sa v okolí R , $I(i, j)$ sú označené pixle obrazu nachádzajúce sa v okolí R a $\bar{I}(i, j)$ je priemer pixlov patriacich okoliu R . Po úprave dostaneme tvar

$$\sigma_R(u, v) = \overline{I(i, j)^2} - (\overline{I(i, j)})^2, \quad (18)$$

kde $\overline{I(i, j)}$ je priemerom intenzít z fixne daného okolia R a $\overline{I(i, j)^2}$ je priemerom druhých mocnín intenzít z fixne daného okolia R . Tieto priemery budeme približne počítať pomocou rovnice vedenia tepla, pričom šírka okna bude zodpovedať čas.

Budeme hľadať funkciu $u(x, t)$, kde $x \in \Omega$, $t \in [0, T]$ a PDR je definovaná v tvare

$$\frac{\partial u(x, t)}{\partial t} = \Delta u(x, t), \quad (19)$$

s Neumanovými okrajovými podmienkami na hranici $\partial\Omega$ v tvare

$$\frac{\partial u(x, t)}{\partial \vec{n}} = 0, \quad (20)$$

kde \vec{n} je jednotková vonkajšia normála ku hranici $\partial\Omega$ a počiatočnou podmienkou

$$u(x, 0) = u^0(x), \quad (21)$$

ktorá je určená počiatočnými obrazovými dátami.

Výpočet pre lokálny prah je definovaný na oblasti R so stredom v (u, v) . Polomer oblasti R by mala byť čo najväčšia, aspoň tak veľká ako štruktúra, ktorú sa vyprahovaním snažíme získať, ale dostatočne malá na zachytenie zmien (nerovností) pozadia.

Jeden z problémov, ktorý môže nastať pre malé hodnoty smerodajnej odchýlky $\sigma_R(u, v)$ (získané na oblastiach v obrazových dátach s takmer konštantnou intenzitou),

prah bude mať hodnotu blízku lokálnemu priemeru, čo spôsobí, že segmentácia je pomerne citlivá na nízku amplitúdu šumu ("ghosting"). Pomocou jednoduchšej modifikácie rovnice (16) pridaním konštanty d , ktorá zabezpečí minimálnu vzdialenosť od priemeru v tvare

$$Q(u, v) = \mu_R(u, v) + \kappa \sigma_R(u, v) + d, \quad (22)$$

kde $d \geq 0$. V našom prípade, sme parameter κ nastavili na $\kappa = 0.18$ a pre konštantu d sme použili $d = 20$. Takto nastavené parametre dávali v našom prípade veľmi dobré výsledky.

2.3 Metóda subjektívnych plôch (SUBSURF)

Metóda subjektívnych plôch (SUBSURF) je vysoko účinnou segmentačnou metódou, ktorá dokáže účinne nájsť chýbajúce hranice objektu alebo odstrániť šum z pozadia. Avšak kvalita výsledku segmentácie závisí od voľby počiatočnej podmienky. Keďže väčšina makrofágov má zložitý tvar, môžeme predpokladať, že metóda SUBSURF by nedávala správne výsledky ak by nebola vhodne zvolená počiatočná podmienka. Za počiatočnú podmienku zvolíme dáta, na ktoré bola aplikovaná niektorá z vyššie uvedených prahovacích metód.

Použitá metóda má tvar

$$u_t = \sqrt{\varepsilon^2 + |\nabla u|^2} \nabla \cdot \left(g \frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}} \right), \quad (23)$$

kde ε je modelovacím parametrom a u je vyvíjajúca sa level funkcia. Funkcia g reprezentuje takzvaný hranový detektor a má tvar

$$g(s) = \frac{1}{1 + K s^2}, K > 0, \quad (24)$$

kde $s = |\nabla G_\sigma * I^0|$. Zhľadený gradient $\nabla G_\sigma * I^0$ získame napríklad aplikovaním jedného kroku vedenia tepla. *DOPLNIT?*

3 Numerické schémy

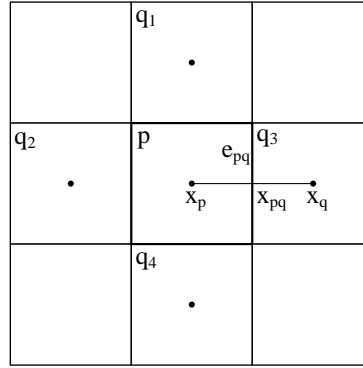
Pri vytváraní numerickej schémy sme pre časovú diskretizáciu použili implicitnú a semi-implicitnú schému a pri priestorovej diskretizácii bola použitá metóda konečných objemov.

3.1 Implicitná metóda pre rovnicu vedenia tepla

Pri implicitnej metóde, budeme časovú deriváciu aproximovať pomocou spätnej diferencie a pravú stranu rovnice berieme v novom časovom kroku n . Má tvar

$$\frac{u^n - u^{n-1}}{\tau} = \Delta u^n = \nabla \cdot (\nabla u^n) \quad (25)$$

a pre ľubovoľnú veľkosť časového kroku je bezpodmienečne stabilná.



Obr. 1: Detail siete spolu s označením. V strede sa nachádza pixel p , susedné pixle sú označené ako q_i , kde $q_i = 1, 2, 3, 4$, hrany medzi pixlami označíme ako e_{pq} , x_p a x_q sú body.

Pre pixelovú sieť budeme používať označenie uvedené na obr. 1. Rovnicu (25) integrujeme cez konečný objem p a dostaneme rovnicu v tvare

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \int_p \nabla \cdot (\nabla u^n) dx. \quad (26)$$

Na pravú stranu rovnice aplikujeme Greenovu vetu a dostaneme

$$\int_p \nabla \cdot (\nabla u^n) dx = \int_{\partial p} \nabla u^n \cdot \vec{n}_p dS, \quad (27)$$

kde \vec{n}_p je jednotkovou vonkajšou normálou ku hranici konečného objemu p . Pretože platí

$$\int_{\partial p} \nabla u^n \cdot \vec{n}_p dS = \sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \cdot \vec{n}_{pq} dS, \quad (28)$$

vieme napísať slabú konečno-objemovú formuláciu úlohy

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \cdot \vec{n}_{pq} dS = \sum_{q \in N(p)} \int_{e_{pq}} \frac{\partial u^n}{\partial \vec{n}_{pq}} dS, \quad (29)$$

Riešenie v MKO chápeme ako po častiach konštantnú funkciu na konečných obje-
moch p , označíme u_p^n . Preto vieme ľavú stranu rovnice napísať ako

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \frac{u_q^n - u_p^n}{\tau} m(p). \quad (30)$$

Pravá strana reprezentuje tok (flux) cez hranicu e_{pq} v smere normály \vec{n}_{pq} . Člen ľavej
strany $\nabla u^n \cdot \vec{n}_{pq}$ aproximujeme na hrane e_{pq} konečnou diferenciou v bode x_{pq} nasledovne

$$\nabla u^n \cdot \vec{n}_{pq} \approx \frac{u_q^n - u_p^n}{d_{pq}}. \quad (31)$$

V časovom kroku n aproximáciu x_{pq} použijeme na celej hrane e_{qp} a dostaneme

$$\sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \cdot \vec{n}_{pq} dS \approx \sum_{q \in N(p)} m(e_{pq}) \frac{u_q^n - u_p^n}{d_{pq}}. \quad (32)$$

Po dosadení dostávame implicitnú schému

$$\frac{m(p)(u_p^n - u_p^{n-1})}{\tau} = \sum_{q \in N(p)} m(e_{pq}) \frac{(u_q^n - u_p^n)}{d_{pq}} \quad (33)$$

kde platia vzťahy $m(p) = h^2$, $m(e_{pq}) = h$, $d_{pq} = h$. Po dosadení dostaneme lineárny
rovnícový systém

$$u_p^n + \frac{\tau}{h^2} \sum_{q \in N(p)} (u_q^n - u_p^n) = u_p^{n-1}. \quad (34)$$

Pri implicitnej schéme platí bezpodmienečná stabilita (diskrétny princíp minima a
maxima), čo znamená, že ak pre ľubovoľnú voľbu priestorového kroku h a časového
kroku τ , platí $u_{min} \leq u_p^{n-1} \leq u_{max}$ potom platí aj pre $u_{min} \leq u_p^n \leq u_{max}$. Riešenie
lineárneho systému upravíme do nasledovného tvaru

$$(1 + \frac{\tau}{h^2} \sum_{q \in N(p)} 1) u_p^n - \frac{\tau}{h^2} \sum_{q \in N(p)} u_q^n = u_p^{n-1}. \quad (35)$$

Rovnicu (35) dostaneme na každom konečnom objeme p . Následne schému riešime
pomocou super-relaxačnej metódy (eng. Successive Over-Relaxation method - SOR).

3.2 Numerická schéma pre SUBSURF

V prvom rade urobíme úpravu do divergentného tvaru, tak že rovnicu (23) vydelíme
členom $\sqrt{\varepsilon^2 + |\nabla u|^2}$ a dostaneme

$$\frac{1}{|\nabla u^{n-1}|_\varepsilon} \cdot \frac{u^n - u^{n-1}}{\tau} = \nabla \cdot (g^0 \frac{\nabla u^n}{|\nabla u^{n-1}|_\varepsilon}), \quad (36)$$

kde $g^0 = g(|\nabla G_\sigma * I^0|)$ a $|\nabla G_\sigma * I^0|$ predstavuje zhladený gradient. Teraz správime priestorovú diskretizáciu pomocou metód konečných objemov, čo znamená, že integrujeme rovnicu (36) cez konečný objem p a dostaneme

$$\int_p \frac{1}{|\nabla u^{n-1}|_\varepsilon} \cdot \frac{u^n - u^{n-1}}{\tau} dx = \int_p \nabla \cdot (g^0 \frac{\nabla u^n}{|\nabla u_p^{n-1}|_\varepsilon}) dx, \quad (37)$$

Následne na pravú stranu rovnice aplikujeme Greenovu vetu

$$\int_p \frac{1}{|\nabla u^{n-1}|_\varepsilon} \cdot \frac{u_p^n - u_p^{n-1}}{\tau} dx = \int_{\partial p} g^0 \frac{\nabla u^n}{|\nabla u_p^{n-1}|_\varepsilon} n_{pq} dS, \quad (38)$$

Derivácia v smere vonkajšej normály ku konečnému objemu p je v rovnici (38) reprezentovaná ako $\nabla u^n \cdot n_{pq}$. Na ľavej strane rovnice (38) budeme uvažovať konštantnú reprezentáciu riešenia a jeho gradientu. Normálovú deriváciu z pravej strany rovnice nahradíme konečnou diferenciou hodnotami, ktoré reprezentujú hodnoty na hranách konečného objemu p . Nasledovne

$$\frac{m(p)}{|\nabla u_p^{n-1}|_\varepsilon} \frac{u_p^n - u_p^{n-1}}{\tau} = \sum_{q \in N(p)} \int_{e_{pq}} g_{pq}^0 \frac{u_q^n - u_p^n}{d_{pq}} \cdot \frac{1}{|\nabla u_p^{n-1}|_\varepsilon} ds. \quad (39)$$

Výčíslením integrálu v časovom kroku n aproximáciu x_{pq} použijeme na celej hrane e_{qp} a dostaneme

$$\frac{m(p)}{|\nabla u_p^{n-1}|_\varepsilon} \frac{u_p^n - u_p^{n-1}}{\tau} = \sum_{q \in N(p)} g_{pq}^0 \frac{u_q^n - u_p^n}{d_{pq}} \cdot \frac{m(e_{pq})}{|\nabla u_p^{n-1}|_\varepsilon}. \quad (40)$$

Keďže ide o pixelovú sieť s hranu h , dostaneme tvar

$$\frac{h^2}{\tau |\nabla u_p^{n-1}|_\varepsilon} u_p^n + \sum_{q \in N(p)} \frac{g_{pq}^0}{|\nabla u_p^{n-1}|_\varepsilon} (u_q^n - u_p^n) = \frac{h^2}{\tau |\nabla u_p^{n-1}|_\varepsilon} u_p^{n-1}. \quad (41)$$

Z tvaru (41), dostaneme rovnicu pre každý pixel

$$a_p^{n-1} u_p^n - \sum_{q \in N(p)} a_{pq}^{n-1} u_q^n = b_p^{n-1} u_p^{n-1}, \quad (42)$$

kde $a_{pq}^{n-1}, a_p^{n-1}, b_p^{n-1}$ označujú

$$\begin{aligned} a_{pq}^{n-1} &= \frac{g_{pq}^0}{|\nabla u_p^{n-1}|_\varepsilon}, \\ a_p^{n-1} &= \frac{h^2}{\tau |\nabla u_p^{n-1}|_\varepsilon} + \sum_{q \in N(p)} a_{pq}^{n-1}, \\ b_p^{n-1} &= \frac{h^2}{\tau |\nabla u_p^{n-1}|_\varepsilon}, \end{aligned} \quad (43)$$

a po dosadení okrajových podmienok dostávame sústavu lineárnych rovníc riešenú v novom časovom kroku u^n .

V našom prípade na získanie hranového detektora g_{pq}^0 použijeme priemer z originálnych dát a dát, na ktoré sme použili niektorú z globálnych alebo lokálnych adaptívnych prahovacích metód nasledovne

$$g_{pq}^0 = \frac{1}{1 + k|\nabla \frac{u_o + u_t}{2}|^2}, \quad (44)$$

kde u_o označujú pôvodné dáta a u_t sú vyprahované dáta.

4 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.

4.1 Qt

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

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 prekryvať, dajú sa minimalizovať/maximalizovať vrámci hlavného okna a zavrieť.

- **QScrollArea**

QScrollArea sa nachádza v každom podokne widgetu QMdiArea. 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 panel (*eng. dock*), ktorý sa dá vrámci okna premiestňovať, ukotvovať buď na ľavej alebo na pravej strane okna a v prípade, že sa prekryvajú vytvorí sa z nich viacero záložiek. Taktiež sa dajú v prípade potreby minimalizovať.

- **QTreeWidget**

Všetky dáta, či už pôvodné alebo vysegmentované pomocou programe, 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.

4.2 VTK

Visualization Toolkit (VTK) sú voľne dostupnými knižnicami, ktoré v programe slúžia na zobrazovanie a interakciu s 2D aj 3D dátami. Spomenieme niektoré knižnice, ktoré hrajú dôležitú úlohu pri našej implementácii.

- **vtkSmartPointer**

Táto šablónová trieda, slúži ako pointer pre VTK triedy. Jeho 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.

- **vtkPoints**

Je triedou reprezentujúcou zoskupenie trojíc (x, y, z) 3D bodov a manipuláciou s uloženými bodmi.

- **vtkPolygon**

Uľahčuje vytváranie buniek n -stranného mnohoúhelníka v rovine. V našom prípade ide o štvoruholníky, znázorňujúce diskretnú sieť. Každý štvoruholník reprezentuje jeden pixel načítaných obrazových dát.

- **vtkTriangles**

Umožňuje vytvorenie trojuholníkových buniek v priestore.

- **vtkCellArray**

Objekty triedy vtkCellArray zabezpečujú prepojenie samostatných buniek rôznych typov, v našej implementácii sa jedná o bunky trojuholníkové (vtkTriangles) a štvorcové (vtkPolygon). Štruktúra tejto triedy je reprezentovaná celočíselným poľom so štruktúrou v tvare: $(n, id_1, id_2, \dots, id_n, n, id_1, id_2, \dots, id_n, \dots)$, kde n je počet bodov, nachádzajúcich sa v bunke a id je index z pridruženého zoznamu bodov.

- **vtkPolyData**

V objektoch tejto triedy sú zadefinované 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.

- **vtkColorTransferFunction**

Je to trieda, ktorá nám slúži na definíciu farebných prechodov cez farebné modely RGB alebo HSV v po častiach spojitom priestore.

- **vtkPolyDataMapper**

Je triedou, zabezpečujúcou tzv. 'namapovanie' čo znamená, že zadefinuje, vlastnosti polydát definovaných ako objekt triedy vtkPolyData, ktoré sú potrebné pri následnom vykreslení.

- **vtkActor**

Táto trieda reprezentuje, geometriu a vlastnosti vykresľovaných dát na zobrazovanej scéne. Odkazuje na geometriu uloženú ako objekt triedy na vtkPolyDataMapper a dedí funkcie súvisiace s pozíciou a orientáciou vykresľovaných dát. Tieto informácie sú uložené v transformačnej matici o veľkosti 4×4 , ktorá zabezpečuje rotácie vo všetkých smeroch (x, y, z) , škálovanie objektov atď..

- **vtkRenderer**

Trieda zabezpečujúca samotné vykresľovanie dát. Stará sa o prevod geometrie, špecifikuje svetelné podmienky a orientáciu kamery. Tiež vykonáva transformáciu súradníc medzi svetovými súradnicami, súradnicami zobrazenia (t.j. súradnicový systém počítačovej grafiky) a súradnicami displeja (t.j. súradnice obrazovky displeja).

- **vtkCutter**

Trieda vtkCutter, je filtrovacou triedou, ktorá pomocou implicitnej funkcie zabezpečuje vykresľovanie, viacerých alebo jedného rezu cez trojdimenzionálne dáta. Vo VTK na rezanie (eng. cutting) znamená zredukovanie dimenzie o jedna. V našom prípade išlo o 3D dáta, ktoré sme rezali pomocou roviny.

- **vtkCubeAxesActor**

Pomocou tejto triedy, sú zobrazované hranice vykresľovaných dát spolu s označeniami osí a hodnotami nachádzajúcimi sa v smeroch x, y, z - výška, šírka a hĺbka.

- **vtkInteractorStyleJoystickCamera, vtkInteractorStyleImage**

Tieto dve triedy umožňujú prepínanie interakcie objektu s prostredím, v našom prípade išlo o odobranie možnosti rotácie pri 2D obrazových dátach.

- **vtkXMLPolyDataWriter**

Extrahuje dáta typu vtkPolydata do súboru. Štandardný formát vytvoreného súboru je *.vtp*.

4.3 Rozdelenie projektu do tried

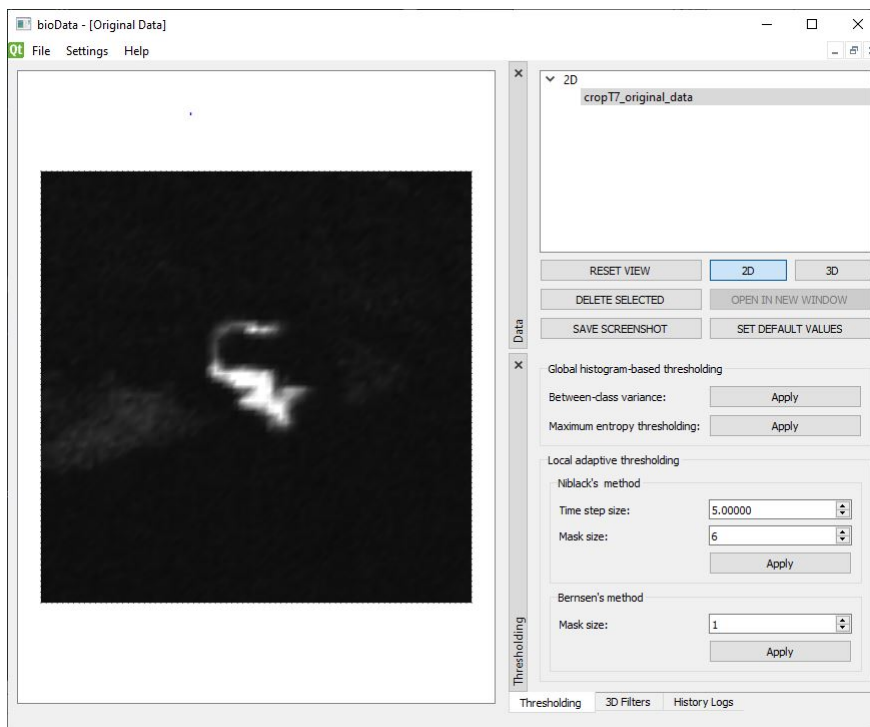
Projekt je rozdelený do viacerých súborov(tried) dodržiavajúcej zásady objektovo-orientovaného programovania, z ktorých každá zabezpečuje funkcionálnu časť programu. Nasledovne:

- Trieda **bioData**, spája všetky triedy a definuje užívateľské rozhranie.
- V **source** sa načítavajú dáta zo súborov, vytvárajú sa dáta na následné zobrazenie spolu so zafarbením.
- V triede **filters**, sú definované všetky filtrovacie funkcie spolu s počiatočnými a okrajovými podmienkami vstupujúcimi do SUBSURFu.
- **ViewerWidget** zabezpečuje vykresľovanie samotných 2D aj 3D dát.
- Trieda **window** uchováva, hodnoty nastavené v užívateľskom rozhraní pre každé okno osobitne.

4.4 Grafické užívateľské rozhranie

V tejto sekcii sa oboznámime s vizuálnou stránkou vytvoreného softvéru a popíšeme funkcionality. Pri tvorbe programu sme sa zamerali na to aby bol čo najjednoduchší a vedel ho ovládať aj niekto, kto sa do danej problematiky až tak do hĺbky nevyzná. Grafické rozhranie(GUI) je vytvorené pomocou Qt knižníc.

Pri otvorení programu sa zobrazí okno, v ktorom sa nachádza horná lišta s položkami File, Settings a Help zvyšok bude načítaný až po otvorení. V každej z týchto položiek sa nachádzajú ďalšie možnosti. Položka File obsahuje možnosti: Open, Save, Close Files a Close.



Obr. 2: Grafické rozhranie programu po načítaní vybraných dát.

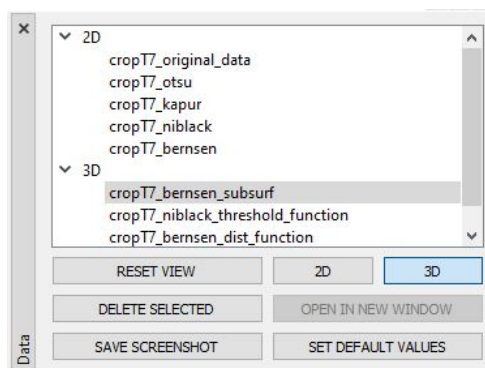
Po zvolení možnosti Open, sa otvorí dialógové okno, z ktorého sa dajú vybrať obrazové súbory typu .pgm (portable gray map). Po zvolení súboru sa zobrazí celé užívateľské prostredie aj so zvolenými dátami obr. 2. Na ľavej strane sú zobrazené načítané dáta. Na pravej strane, môžeme vidieť bočný panel rozdelený na dve časti, v ktorých sa nachádza zoznam spracovávaných dát v programe pre 2D aj 3D dáta, možnosti pre 2D a 3D dáta spolu so všetkými implementovanými filtrami, prahovacími metódami a možnosťami pre zobrazenie uložených vo viacerých záložkách.

Prvá časť panelu má názov *Data* obr. 3a obsahuje zoznam 2D a 3D dát, s ktorými sa doteraz pracovalo. Umožňuje spätné načítanie dát nachádzajúcich sa v zozname. Tiež obsahuje tlačidlá:

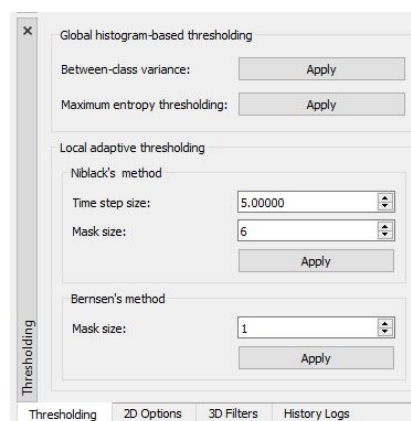
- **RESET VIEW** - vykresľovaný objekt, prekreslí na stred vykresľovacej plochy.
- **2D** - v prípade, že v zozname sú zvolené 2D dáta, odoberie možnosť rotácie a je vysvietené.
- **3D** - v prípade, že v zozname sú zvolené 3D dáta, tak sa s nimi dá adekvátne manipulovať a tlačidlo je vysvietené.

- DELETE SELECTED - vymaže zvolené dáta zo zoznamu.
- OPEN IN NEW WINDOW - zatiaľ neaktívne.
- SAVE SCREENSHOT - uloží práve zobrazené dáta z vykresľovacej plochy ako obrazové dáta vo formáte *.png*
- SET DEFAULT VALUES - prestaví hodnoty všetkých voliteľných hodnôt naspäť na predvolené hodnoty.

V záložke **Thresholding** obr. 3b, sa nachádzajú štyri implementované prahovacie algoritmy, ktoré vstupujú následne ako okrajová podmienka do segmentačnej metódy. Ide o dva globálne prahovacie algoritmy, ktoré závisia len na histograme a teda neberú na vstup žiadne parametre. A dva lokálne adaptívne prahovacie algoritmy, ktoré berú na vstup veľkosť masky a jeden z nich veľkosť časového kroku.



(a) Detail docku Data.

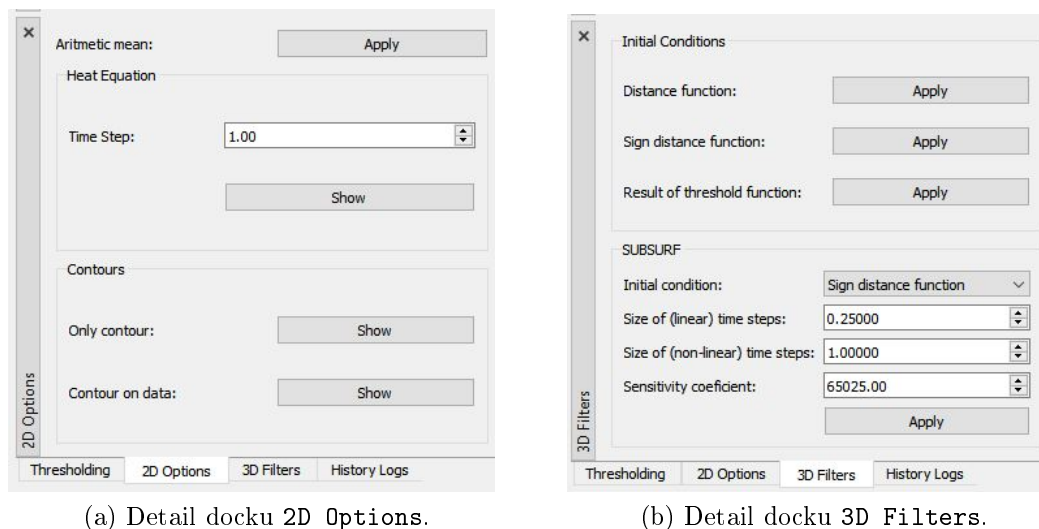


(b) Detail docku Thresholding.

Obr. 3

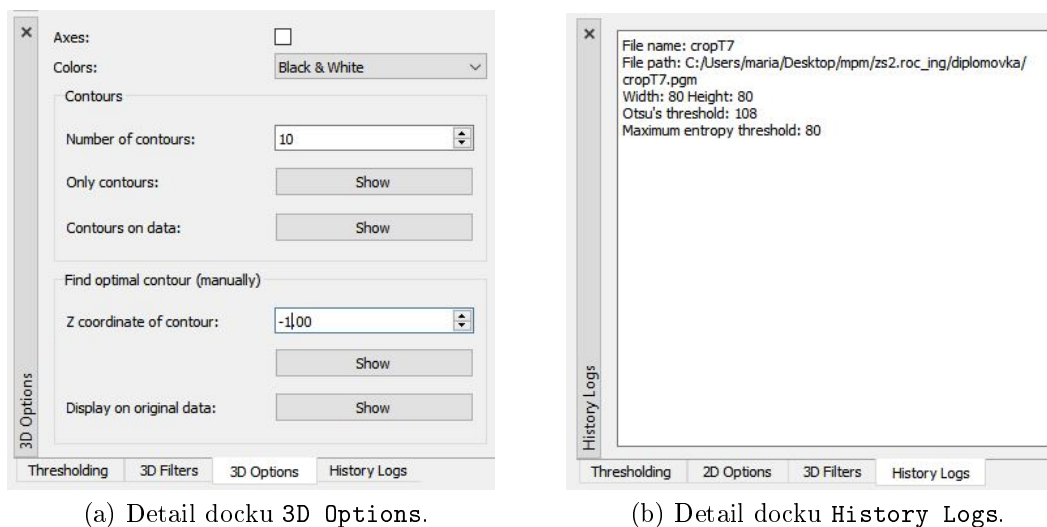
Záložka **2D Options** obr. 4a je zobrazená len v prípade, že sú zvolené 2D dáta. V tejto záložke sa nachádzajú možnosti súvisiace len s 2D dátami. Dajú sa tam zobrazíť kontúry na dátach, jeden krok rovnice vedenia tepla a priemer originálnych a vyprahovaných obrazových dát.

3D Filters obr. 4b obsahuje možnosti pre samotný SUBSURF spolu s jeho parametrami. Taktiež sa tam nachádzajú počiatočné podmienky znamienkovej dištančnej a prahovacej funkcie, ktoré sa dajú zobrazíť ešte pred vstupom do SUBSURFu.



Obr. 4

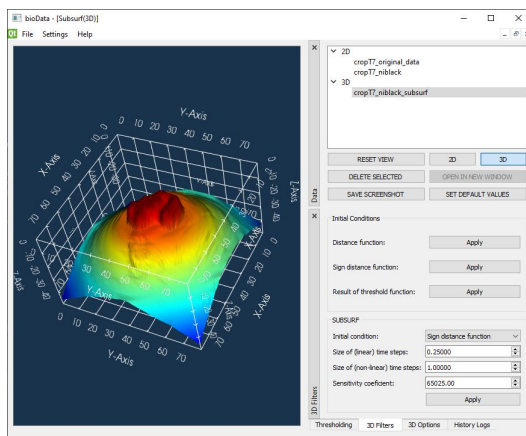
V 3D Options obr. 5a je zobrazený len v prípade, že sú zvolené 3D dáta. V tejto záložke sa nachádzajú možnosti súvisiace len s 3D dátami. V týchto možnostiach sa dajú zapnúť osi, zmeniť farba, vykresliť zvolený počet izočiar buď na dátach alebo bez nich a dá sa zvoliť optimálna izočiara a následne ju je možné vykresliť na pôvodných dátach.



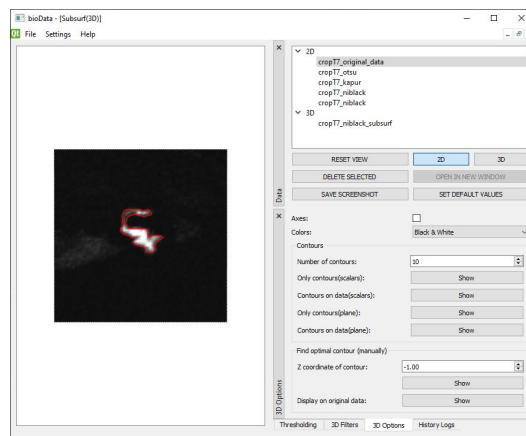
Obr. 5

V záložke History Logs obr. 5b sa nachádza jednoduché textové pole, ktoré zaznamenáva väčšinu úkonov vykonaných v programe. Taktiež sa tam nachádzajú informácie o načítanom súbore, ako napríklad aj názov súboru, cesta k súboru atď... Tento výstup sa dá uložiť do jednoduchého textového súboru.

Na obr. 6a sú zobrazené dáta s makrofágom, na ktoré bol aplikovaný SUBSURF, za počiatočnú podmienku bola použitá dištančná funkcia.



(a) Grafické rozhranie programu zobrazujúce 3D dáta spolu s osami x,y,z.



(b) Grafické rozhranie programu zobrazujúce výslednú izočiaru na pôvodných dátach

Obr. 6

Na obr. 6b sú vidieť pôvodné dáta, na ktorých je zobrazená výsledná izočiara.

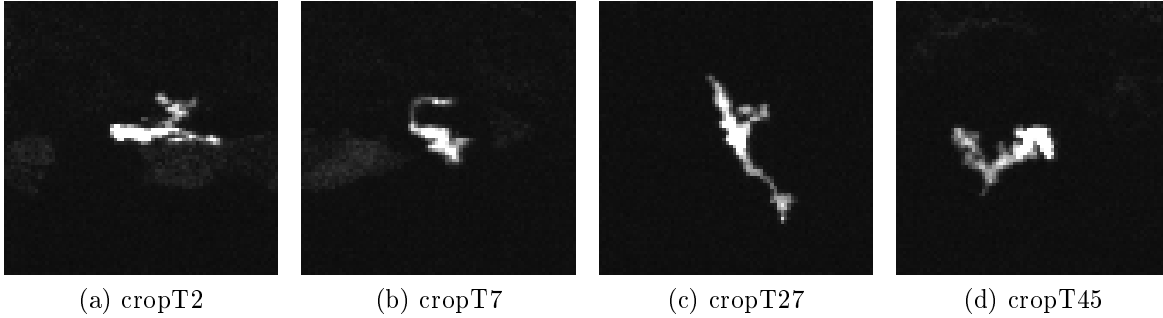
Vykresľovacia plocha má niekoľko, vstavaných funkcií, ktoré sa dajú ovládať cez klávesové skratky. Ide napríklad o klávesy W a S prepínajú zobrazenie dát medzi obrysom dát a zafarbeným modelom, klávesa R reštartuje pohľad na objekt na vykresľovanej ploche.

Zvyšné možnosti, v hornej lište v položke File sa ešte nachádza možnosť Save, ktorá ponúka buď exportovanie 2D aj 3D dát vo VTK formáte .vtp (typ súboru špecifický pre vtkPolyData) alebo 2D dáta vie uložiť ako ascii portable gray map dáta .pgm. Ďalšou možnosťou je Close Files, kedy sa zatvoria všetky otvorené súbory. A možnosti Close, ktorá zatvorí celý program.

V hornej lište sa ešte nachádzajú položky Settings a Help. V Settings vie užívateľ skryť/zobraziť všetky položky nachádzajúce sa vpravo. A Help obsahuje informácie o programe a jednoduchú dokumentáciu.

5 Výsledky

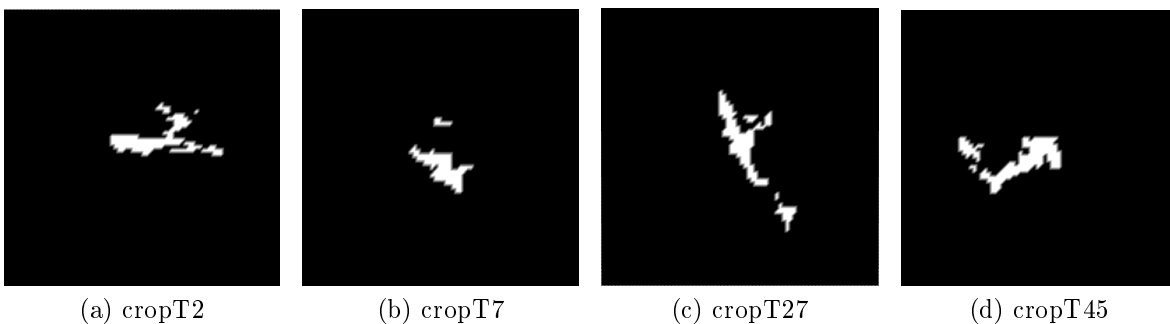
Na ukážku výsledkov sme vybrali niekoľko dát s makrofágmi so zložitými tvarmi a šumom v pozadí. Vybrali sme nasledujúce dáta.



Obr. 7: Zvolené originálne dáta.

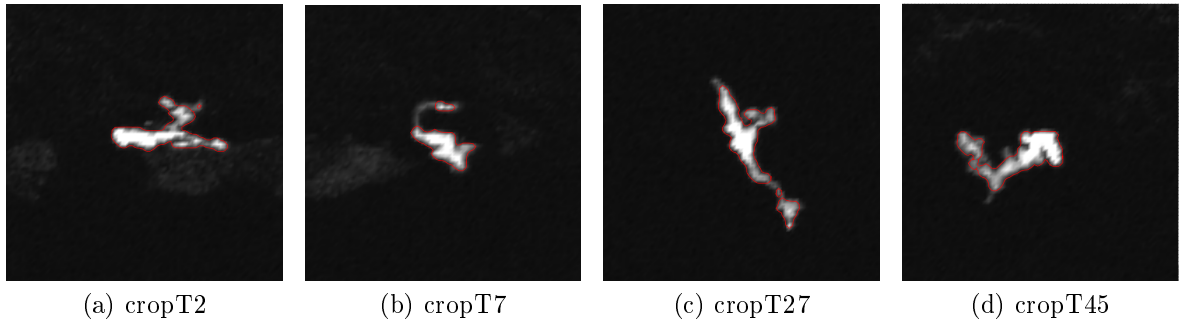
V projekte sú implementované štyri rôzne prahovacie metódy, ktoré by mohli slúžiť ako okrajová podmienka na vstup do segmentačnej metódy. Vyprahované dáta by mali čo najviac popisovať tvar makrofágu bez toho aby sa roztrhli a nemali by obsahovať šum z pôvodných dát. Následne sme na vyprahované dáta aplikovali SUBSURF, vo všetkých prípadoch boli použité rovnaké vstupné parametre: $\sigma = 1.0$, $\tau = 1.0$, $k = 255^2$ a hodnotu výslednej izočiary sme brali na hodnote $z = -1$, keďže vo väčšine prípadoch bola najvhodnejšou.

Na obr. 8 je zobrazená prahovacia funkcia popísaná v sekcii 2.1.1 používajúca medzitriednu varianciu. Dá sa všimnúť, že táto prahovacia metóda nezachytila celý tvar makrofágu, hlavne ak išlo o úzku časť s menšou intenzitou.



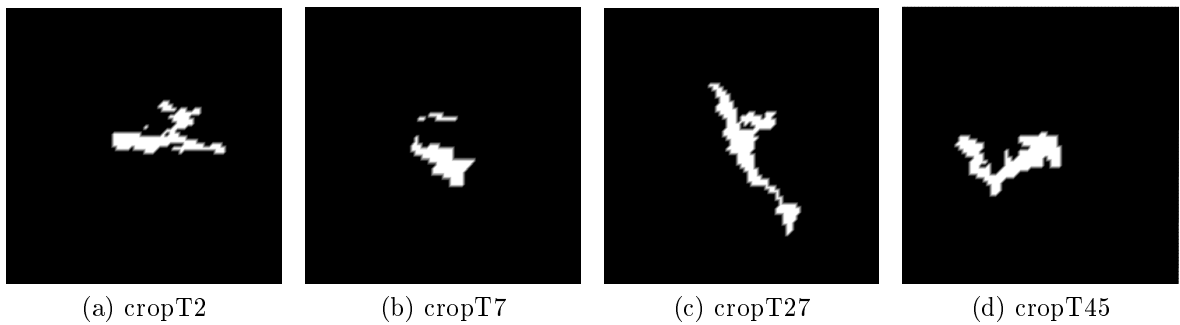
Obr. 8: Otsuho metóda

Červenou izočiary je na obr. 9 vyznačený výsledok segmentácie. Ale môžeme vidieť chýbajúce časti, hlavne na obr 9b a 9c. Ďalej si môžeme všimnúť, že niektoré časti makrofágov, boli vďaka segmentácii prepojené aj keď ich prahovacia metóda roztrhla.



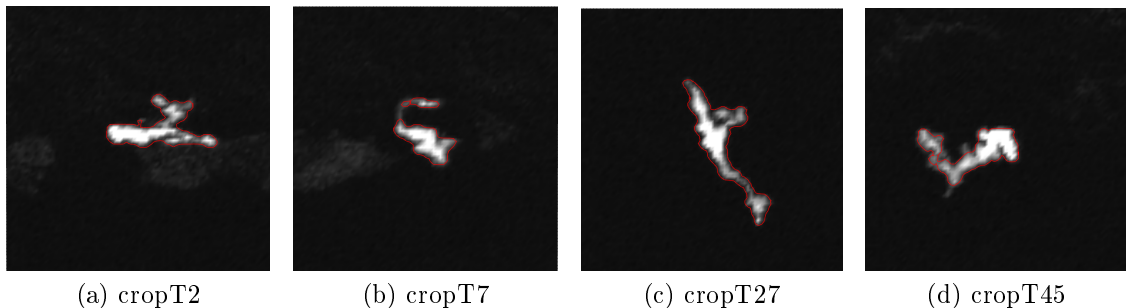
Obr. 9: Výsledok SUBSURFu za použitia Otsuho prahovacej metódy.

Prahovacia metóda popísaná v sekcii 2.1.2 získaná pomocou maximálnej entropie, je zobrazená na obr. 10. Môžeme si všimnúť, že prah tejto metódy bol určený o niečo vhodnejšie ako predošlá metóda, napríklad v prípade 10c. Avšak pri tomto type prahovacej metódy sa môže stať, že objekt bude obsahovať aj časti, ktoré pôvodne nemal.



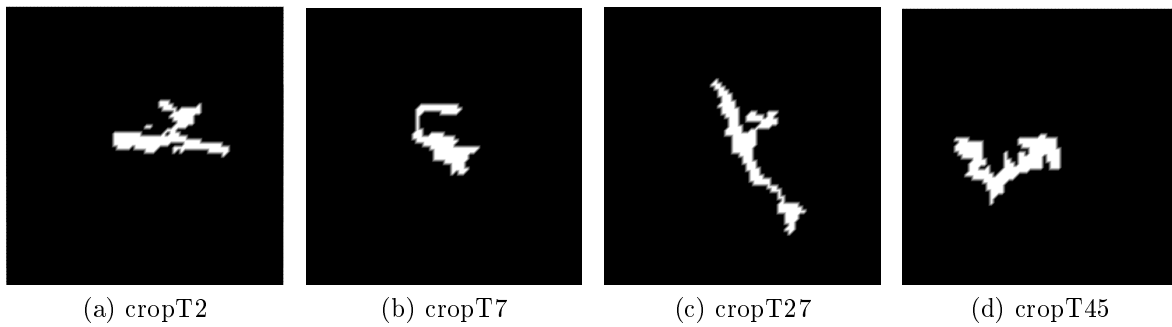
Obr. 10: Prahovanie pomocou maximálnej entropie.

Vo výsledkoch 11 vidieť, že objekty obsahujú niektoré časti, ktoré sa na pôvodných dátach nenachádzajú a v niektorých prípadoch neobsahujú všetky časti.



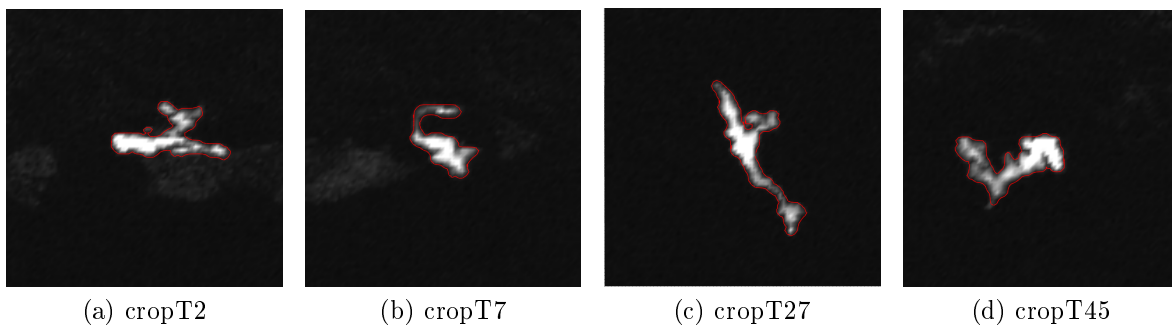
Obr. 11: Výsledok SUBSURFu pre prah nájdený pomocou maximálnej entropie.

Pri Niblackovej prahovacej metóde popísanej v sekcii 2.2.2, s veľkosťou časového kroku $\sigma = 50.0$, a veľkosťou masky $m = 3$ sú zobrazené na obr. 12.



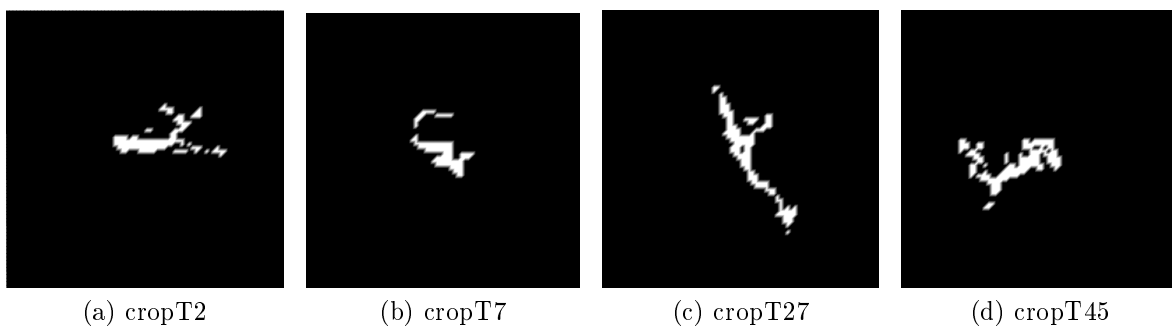
Obr. 12: Zvolené originálne dáta.

Na obr. 13 sú zobrazené výsledky, kde boli na vstup ako okrajová podmienka uvažované dáta z Niblackovej prahovacej metódy.



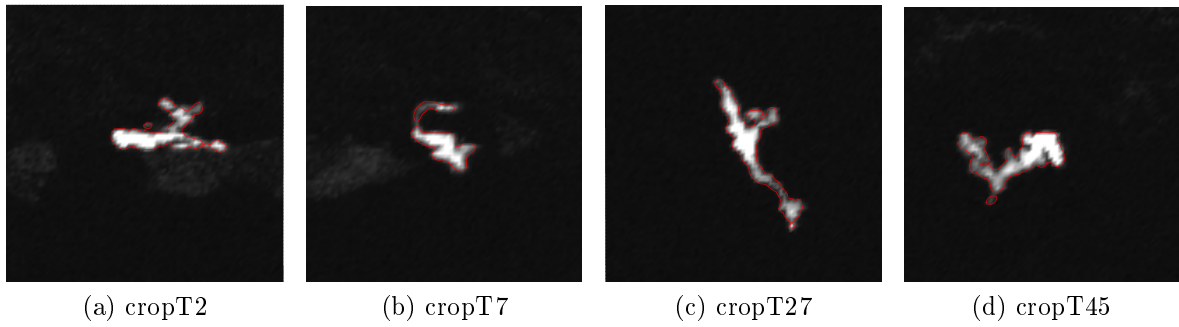
Obr. 13: Niblackova prahovacia metóda.

Bernsenova prahovacia funkcia popísaná v sekcii 2.2.1, pre veľkosť okna $m = 2$, je zobrazená na obr. 14.



Obr. 14: Zvolené originálne dáta.

Výsledky segmentačnej metódy, s kde na vstup sme brali Bernsenove prahovanie na obr. 15.



Obr. 15: Zvolené originálne dáta.

Z výsledkov na obrázkoch 9, 11, 13 a 15 vidieť, že vhodnejšími prahovacími metódami vstupujúcimi do segmentácie ako okrajová podmienka sú lokálne adaptívne prahovacie metódy. A Niblackova prahovacia metóda obr. obr 12 najlepšie popisuje skutočný tvar makrofágu.

6 Záver

Literatúra