

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	6
2.2.3	Hybridné metódy	8
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	12
4.1	Qt	12
4.2	VTK	13
4.3	Rozdelenie projektu do tried	14
4.4	Grafické užívateľské rozhranie	14
4.5	Príklad použitia	18
5	Výsledky	22
6	Záver	31

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 dátá rôznych formátov a pomáha z nich získať užitočné informácie. Obrazové dátá je potrebné zobraziť, či už pred alebo po modifikácií.

Cieľom práce je vytvorenie softvéru, ktorý slúži na vizualizáciu a segmentáciu obrazov získaných konfokálnym laserovým mikroskopom, konkrétnie sa jedná o biologické dátá a to obrazy makrofágov. Takéto dátá 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 dátá 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 svietielkujúcim proteínom (kaede) pre lepšiu viditeľnosť pod mikroskopom. Pôvodné dátá makrofágov sú získané v časovom úseku 5 hodín a s časovým krokom 4 minúty. Následne dané trojdimenzionálne obrazové dátá sú premietnuté do roviny za použitia maximálnej intenzity približne zo 70 rezov, z ktorých vzniknú dvoj-dimenzionálne obrazové dátá.

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 funkcionálita programu 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 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 ľahko spracovávajú. Preto sme vybrali niekoľko druhov prahovacích metód, ktoré sme skombinovali spolu s metódou segmentácie subjektívnych plôch a aplikovali sme ich na testované dátu.

2.1 Globálne prahovanie

Úlohou globálnych prahovacích metód je nájsť jedinú 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[7] patrí medzi automatické prahovacie metódy, ktorá rozdeľuje obrazové dátu 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 tohto 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 dátu. 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 obrazu.

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 definícii 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 zadefinovať medzi-triednu varianciu ako sumu

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

Z hodnoty medzi-triednej variancie, 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 dátu s makrofágmi, v prípade že dátu 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" obsahujúcej stochasticky generované dátu. 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

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

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

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 pozícii (u, v) . Prahovú hodnotu dostaneme pomocou aritmetického priemeru najdeného 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 šírke 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 normálka 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 jednoduchej 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.

Sauvolova prahovacia metóda je vylepšením Niblackovej metódy. Prah $Q(u, v)$ je definovaný v tvare

$$Q(u, v) = \mu_R(u, v) \cdot [1 + \kappa \left(\frac{\sigma_R(u, v)}{\sigma_{max}} - 1 \right)], \quad (23)$$

kde parameter $\kappa \geq 0$, σ_{max} je dynamickým rozsahom pre štandardné odchýlky. Približne hodnoty priemeru intenzít $\mu_R(u, v)$ a smerodajnej odchýlky $\sigma_R(u, v)$ na fixne danej oblasti R sme dostali z rovnice vedenia tepla, tak ako je popísaná vyššie. Parametre metódy boli v našom prípade zvolené nasledovne $\kappa = 0.18$ a $\sigma_{max} = 128$.

2.2.3 Hybridné metódy

Pri hybridnej Niblackovej a Bernsenovej metóde, uvažujeme lokálny kontrast $c(u, v)$ definovaný v Bernsenovej metóde v sekcií 2.2.1 a na výpočet lokálneho prahu $Q(u, v)$ použijeme rovnicu (22), z Niblackovej metódy popísanú v sekcií 2.2.2. Lokálny kontrast $c(u, v)$ aj prah $Q(u, v)$ sú definované na oblasti R .

Pri druhej hybridnej metóde, použijeme kombináciu Bernsenovej metódy a Sauvolovej metódy, podobne ako pri predchádzajúcim prípade lokálny kontrast $c(u, v)$ vypočítame z Bernsenovej metódy a na výpočet lokálneho prahu $Q(u, v)$ použijeme rovnicu (23), zo Sauvolovej metódy. Lokálny kontrast $c(u, v)$ aj prah $Q(u, v)$ sú definované na oblasti R .

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 dátu, na ktoré bola aplikovaná niektorá z vyššie uvedených prahovacích metód.

Použitá metóda má tvar

$$u_t = \sqrt{\varepsilon^2 + |\nabla u|^2} \nabla \cdot (g \frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}), \quad (24)$$

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

$$g(s) = \frac{1}{1 + Ks^2}, K > 0, \quad (25)$$

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

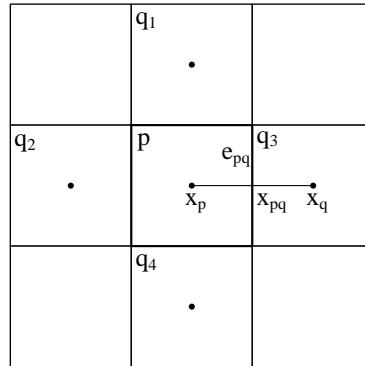
Pri vytváraní numerickej schémy sme pre časovú diskretizáciu použili implicitnú a semi–implicitnú schému a pri priestorovej diskretizácii bola použitá metóda konečných objemov.

3.1 Implicitná metóda pre rovniciu vedenia tepla

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

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

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 (26) integrujeme cez konečný objem p a dostaneme rovnicu v tvare

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \int_p \nabla \cdot (\nabla u^n) dx. \quad (27)$$

Na pravú stranu rovnice aplikujeme Greenovu vetu a dostaneme

$$\int_p \nabla \cdot (\nabla u^n) dx = \int_{\partial p} \nabla u^n \cdot \vec{n}_p dS, \quad (28)$$

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

$$\int_{\partial p} \nabla u^n \cdot \vec{n}_p dS = \sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \cdot \vec{n}_{pq} dS, \quad (29)$$

vieme napísť slabú konečno-objemovú formuláciu úlohy

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \cdot \vec{n}_{pq} dS = \sum_{q \in N(p)} \int_{e_{pq}} \frac{\partial u^n}{\partial \vec{n}_{pq}} dS, \quad (30)$$

Riešenie v MKO chápeme ako po častiach konštantnú funkciu na konečných objemoch p , označíme u_p^n . Preto vieme ľavú stranu rovnice napísť ako

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \frac{u_q^n - u_p^n}{\tau} m(p). \quad (31)$$

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

$$\nabla u^n \cdot \vec{n}_{pq} \approx \frac{u_q^n - u_p^n}{d_{pq}}. \quad (32)$$

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

$$\sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \cdot \vec{n}_{pq} dS \approx \sum_{q \in N(p)} m(e_{pq}) \frac{u_q^n - u_p^n}{d_{pq}}. \quad (33)$$

Po dosadení dostávame implicitnú schému

$$\frac{m(p)(u_p^n - u_p^{n-1})}{\tau} = \sum_{q \in N(p)} m(e_{pq}) \frac{(u_q^n - u_p^n)}{d_{pq}} \quad (34)$$

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

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

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

$$(1 + \frac{\tau}{h^2} \sum_{q \in N(p)} 1) u_p^n - \frac{\tau}{h^2} \sum_{q \in N(p)} u_q^n = u_p^{n-1}. \quad (36)$$

Rovnicu (36) 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 (24) vydelíme členom $\sqrt{\varepsilon^2 + |\nabla u|^2}$ a dostaneme

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

kde $g^0 = g(|\nabla G_\sigma * I^0|)$ a $|\nabla G_\sigma * I^0|$ predstavuje zhľadený gradient. Teraz spravíme priestorovú diskretizáciu pomocou metód konečných objemov, čo znamená, že zintegrujeme rovnicu (37) cez konečný objem p a dostaneme

$$\int_p \frac{1}{|\nabla u^{n-1}|_\varepsilon} \cdot \frac{u^n - u^{n-1}}{\tau} dx = \int_p \nabla \cdot (g^0 \frac{\nabla u^n}{|\nabla u^{n-1}|_\varepsilon}) dx, \quad (38)$$

Následne na pravú stranu rovnice aplikujeme Greenovu vetu

$$\int_p \frac{1}{|\nabla u^{n-1}|_\varepsilon} \cdot \frac{u_p^n - u_p^{n-1}}{\tau} dx = \int_{\partial p} g^0 \frac{\nabla u^n}{|\nabla u_p^{n-1}|_\varepsilon} \vec{n}_{pq} dS, \quad (39)$$

Derivácia v smere vonkajšej normály ku konečnému objemu p je v rovnici (39) reprezentovaná ako $\nabla u^n \cdot \vec{n}_{pq}$. Na ľavej strane rovnice (39) 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

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

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

$$\frac{m(p)}{|\nabla u_p^{n-1}|_\varepsilon} \frac{u_p^n - u_p^{n-1}}{\tau} = \sum_{q \in N(p)} g_{pq}^0 \frac{u_q^n - u_p^n}{d_{pq}} \cdot \frac{m(e_{pq})}{|\nabla u_{pq}^{n-1}|_\varepsilon}. \quad (41)$$

Ked'že ide o pixelovú sieť s hranu h , dostaneme tvar

$$\frac{h^2}{\tau |\nabla u_p^{n-1}|_\varepsilon} u_p^n + \sum_{q \in N(p)} \frac{g_{pq}^0}{|\nabla u_{pq}^{n-1}|_\varepsilon} (u_q^n - u_p^n) = \frac{h^2}{\tau |\nabla u_p^{n-1}|_\varepsilon} u_p^{n-1}. \quad (42)$$

Z tvaru (42), dostaneme rovnicu pre každý pixel

$$a_p^{n-1} u_p^n - \sum_{q \in N(p)} a_{pq}^{n-1} u_q^n = b_p^{n-1} u_p^{n-1}, \quad (43)$$

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

$$\begin{aligned} a_{pq}^{n-1} &= \frac{g_{pq}^0}{|\nabla u_{pq}^{n-1}|_\varepsilon}, \\ a_p^{n-1} &= \frac{h^2}{\tau |\nabla u_p^{n-1}|_\varepsilon} + \sum_{q \in N(p)} a_{pq}^{n-1}, \\ b_p^{n-1} &= \frac{h^2}{\tau |\nabla u_p^{n-1}|_\varepsilon}, \end{aligned} \quad (44)$$

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

$$g_{pq}^0 = \frac{1}{1 + k |\nabla \frac{u_o + u_t}{2}|^2}, \quad (45)$$

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 [2], ktoré obsahujú veľa na implementovaných tried a boli užitočným nástrojom pri vytváraní užívateľského prostredia a VTK [1] knižnicami, ktoré slúžia na zobrazovanie a manipuláciu s dátami.

4.1 Qt

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

Najčastejšie využívané Qt triedy 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 vykreslovanie 2D a 3D dát, s ktorými program pracuje. V programe je použité kaskádové usporiadanie takýchto podokien, čo znamená že vykreslovanie okná sa môžu navzájom prekrývať, dajú sa minimalizovať/maximalizovať vrámcí hlavného okna a zavrieť.

- **QScrollArea**

QScrollArea sa nachádza v každom podokne widgetu QMidiArea. Zabezpečuje možnosť priblížiť/oddialiť a posúvať vizualizované dátu. Tiež sa nachádza aj v častiach programu, kde bolo potrebné aplikovať ScrollBar.

- **QVTKOpenGLNativeWidget**

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

- **QDockWidget**

Tento widget obsahuje všetky informácie a nastavenia súvisiace s dátami. Každý logický celok má vlastný panel (*eng. dock*), ktorý sa dá vrámcí okna premiestňovať, ukotvovať buď na ľavej alebo na pravej strane okna a v prípade, že sa prekrývajú vytvorí sa z nich viacero záložiek. Taktiež sa dajú v prípade potreby minimalizovať.

- **QTreeWidget**

Všetky dátá, či už pôvodné, vyprahované alebo vysegmentované pomocou programu, sa nachádzajú v zozname, z ktorého sa dá vybrať ktoré dátá budú vykreslené. QTreeWidget bol použitý aby sme vykreslované dátá vedeli zadať do medzi 2D alebo 3D dát.

- **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 dátá mimo rozsahu alebo sa nikde nepoužívajú tak budú automaticky odstránené. Teda uľahčuje pracovať s dátami bez varovných hlášok.

- **vtkPoints**

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

- **vtkPolygon**

Uľahčuje vytváranie buniek n -stranného mnohouholníka v rovine. V našom prípade ide o štvoruholníky, znázorňujúce diskrétnu siet. 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 počtom 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 vykreslované dátá, či už sa jedná o 2D alebo 3D dátá. V tejto triede môžu byť uložené informácie o tom akým spôsobom budú dátá reprezentované - geometrické informácie o štruktúre vykreslovaných 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 počastiach spojítom priestore.

- **vtkPolyDataMapper**

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

- **vtkActor**

Táto trieda reprezentuje, geometriu a vlastnosti vykreslovaných dát na zobrazovanej scéne. Odkazuje na geometriu uloženú ako objekt triedy na vtkPolyDataMapper a dedí funkcie súvisiace s pozíciou a orientáciou vykreslovaných 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 vykreslovanie 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 vykreslovanie, viacerých alebo jedného rezu cez trojdimenzionálne dátá. Vo VTK na rezanie (eng. cutting) znamená zredukovanie dimenzie o jedna. V našom prípade išlo o 3D dátá, ktoré sme rezali pomocou roviny.

- **vtkCubeAxesActor**

Pomocou tejto triedy, sú zobrazované hranice vykreslovaných 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 dátá typu vtkPolydata do súboru. Štandardný formát vytvoreného súboru je *.vtlp*.

4.3 Rozdelenie projektu do tried

Projekt je rozdelený do viacerých súborov (tryed) dodržiavajúcej zásady objektovo-orientovaného programovania, z ktorých každá zabezpečuje funkcionality inej časti programu. Nasledovne:

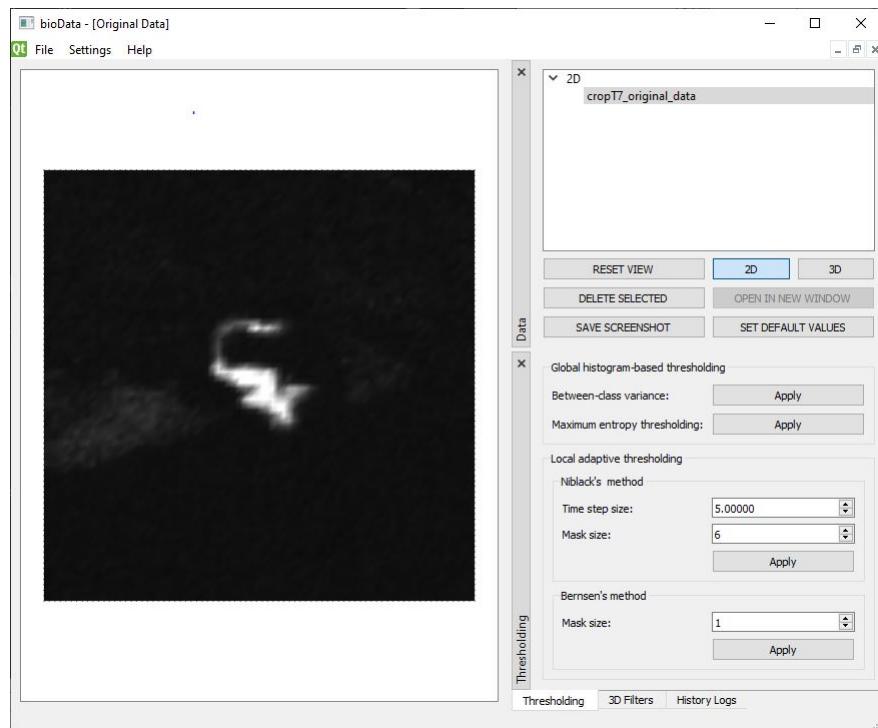
- Trieda **bioData**, spája všetky triedy a definuje užívateľské rozhranie pomocou Qt knižníck.
- V **source** sa načítavajú dátá zo súborov, vytvárajú sa dátá na následné zobrazenie spolu so zafarbením. Taktiež obsahuje funkcie na uloženie dát v rôznych formátoch.
- V triede **filters**, sú definované všetky prahovacie a segmentačné metódy.
- **ViewerWidget** zabezpečuje vykreslovanie samotných 2D aj 3D dát, spolu s osami a izočiarami.
- Trieda **subWin** uchováva, hodnoty nastavené v užívateľskom rozhraní pre každé otvorený súbor 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 knižníck.

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.

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 dátá. Na pravej strane,



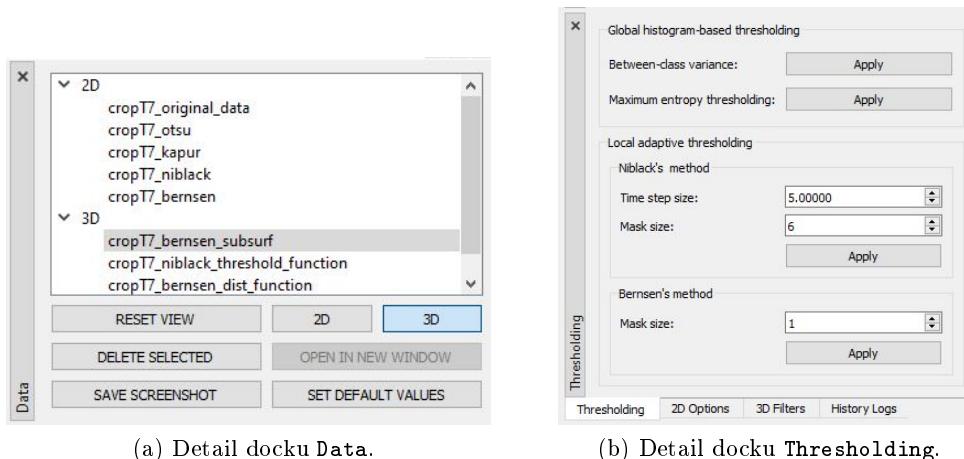
Obr. 2: Grafické rozhranie programu po načítaní vybraných dát.

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 filtrovami, 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 vysvetlené.
- **3D** - v prípade, že v zozname sú zvolené 3D dáta, tak sa s nimi dá adekvátnie manipulovať a tlačidlo je vysvetlené.
- **DELETE SELECTED** - vymaže zvolené dáta zo zoznamu.
- **OPEN IN NEW WINDOW** - zatiaľ neaktívne.
- **SAVE SCREENSHOT** - uloží práve zobrazené dáta z vykresľovacej plochy ako obrazové dátá vo formáte *.png*
- **SET DEFAULT VALUES** - prestaví hodnoty všetkých voliteľných hodnôt naspäť na predvolené hodnoty.

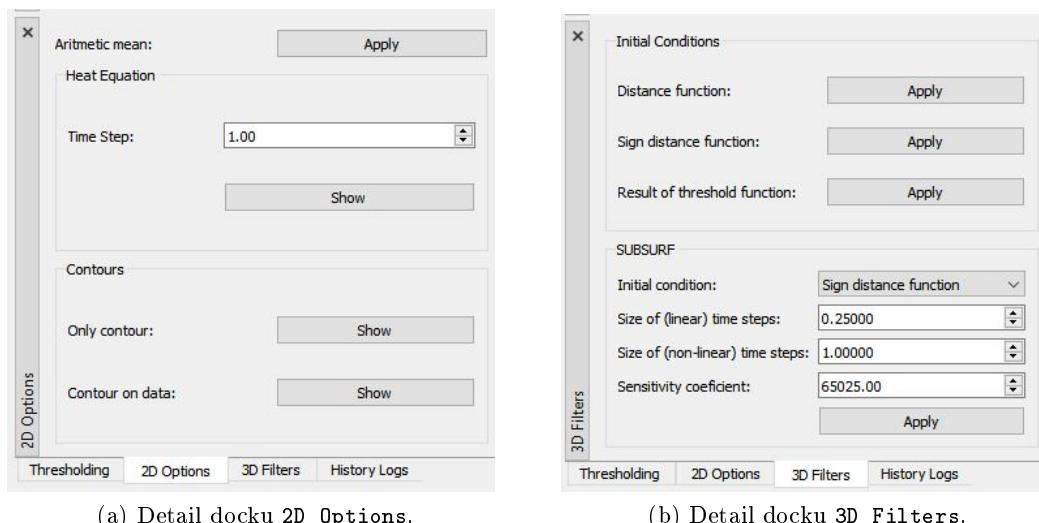
V záložke *Thresholding* obr. 3b, sa nachádzajú štyri implementované prahovacie algoritmy, ktoré vstupujú následne ako okrajová podmienka do segmentačnej metódy. Ide o dva globálne prahovacie algoritmy, ktoré závisia len na histograme a teda neberú na vstup žiadne parametre. A dva lokálne adaptívne prahovacie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



Obr. 3

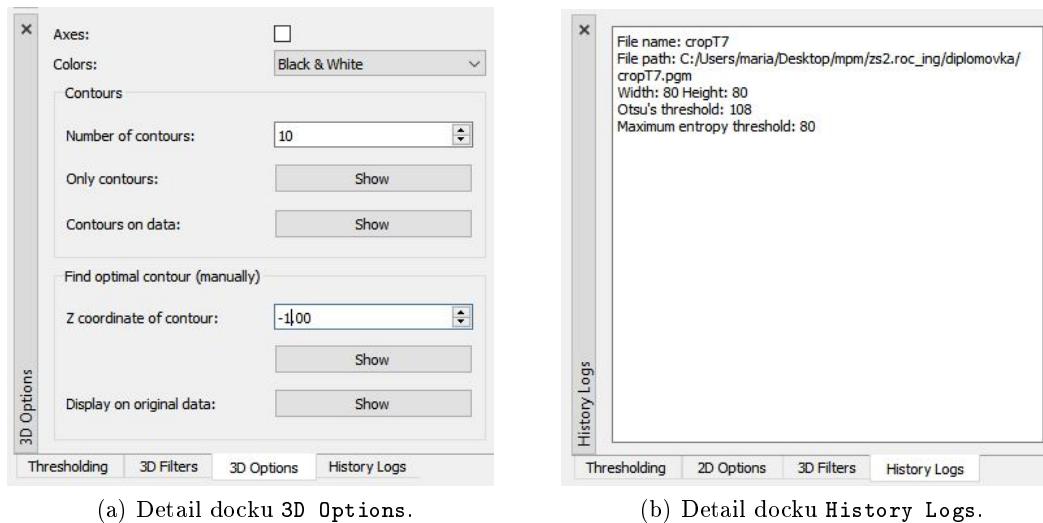
Záložka **2D Options** obr. 4a je zobrazená len v prípade, že sú zvolené 2D dátá. V tejto záložke sa nachádzajú možnosti súvisiace len s 2D dátami. Dájú sa tam zobraziť kontúry na dátach, jeden krok rovnice vedenia tepla a priemer originálnych a vyprahovaných obrazových dát.

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 zobraziť ešte pred vstupom do SUBSURFu.



Obr. 4

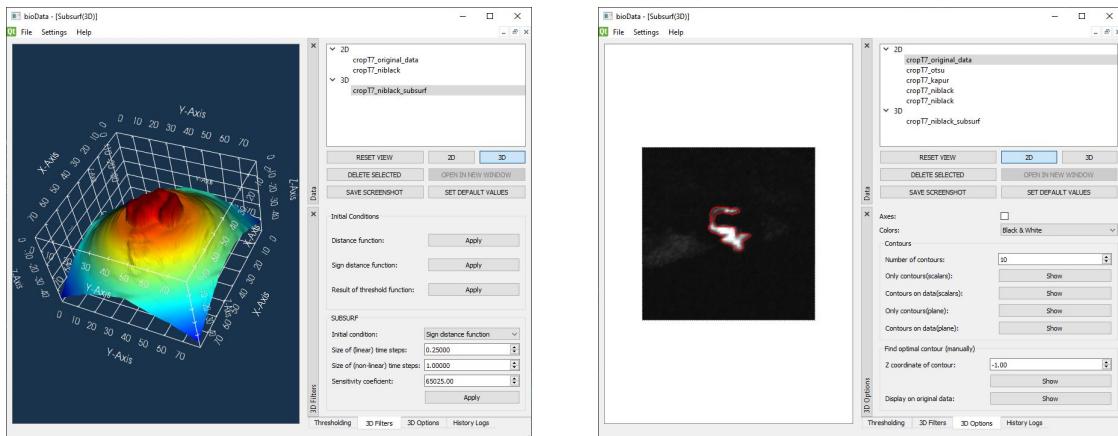
V **3D Options** obr. 5a je zobrazený len v prípade, že sú zvolené 3D dátá. V tejto záložke sa nachádzajú možnosti súvisiace len s 3D dátami. V týchto možnostiach sa dajú zapnúť osi, zmeniť farbu, vykresliť zvolený počet izočiar bud' na dátach alebo bez nich a dá sa zvoliť optimálna izočiara a následne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



Obr. 5

V záložke History Logs obr. 5b sa nachádza jednoduché textové pole, ktoré zaznamenáva väčšinu úkonov vykonalých 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 dátá s makrofágom, na ktoré bol aplikovaný SUBSURF, za počiatočné podmienky bola použitá dištančná funkcia.



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

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

Obr. 6

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

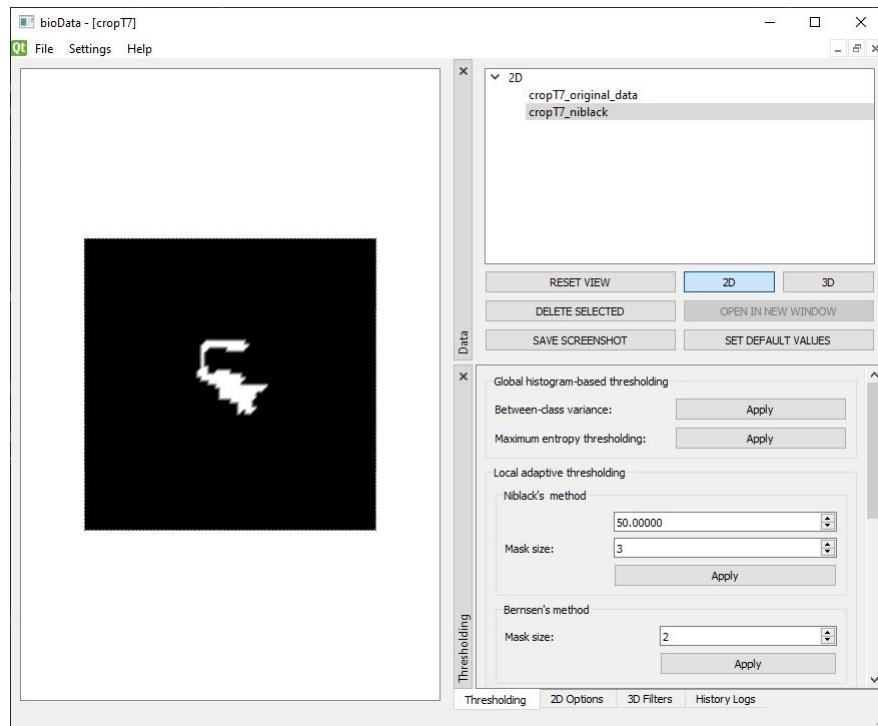
Vykresľovacia plocha má niekoľko, vstavaných funkcií, ktoré sa dajú ovládať cez klávesové skratky. Ide napríklad o klávesy W a S prepínajú zobrazenie dát medzi obrysom dát a zafarbeným modelom, klávesa R reštartuje pohľad na objekt na vykresľovanej ploche.

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 dát vie uložiť ako ascii portable gray map dátu .pgm. Ďalšou možnosťou je Close Files, kedy sa zatvoria všetky otvorené súbory. A možnosť 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.

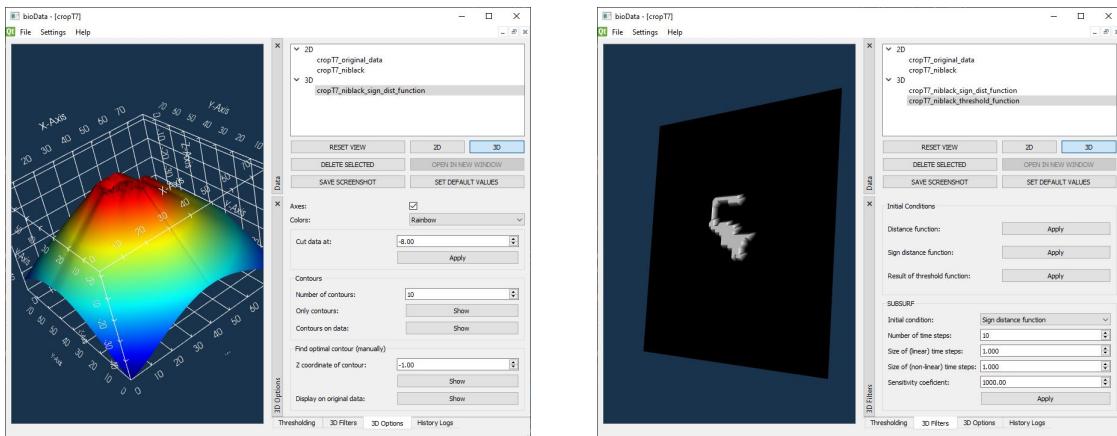
4.5 Príklad použitia

V tejto časti si ukážeme príklad na použitie implementovaného softvéru. Otvoríme súbor typu *.pgm*. Načíta sa užívateľské prostredie obr. 2. V záložke **Thresholding**, zvolíme niektorú z implementovaných prahovacích funkcií, v tomto príklade zvolíme Niblackovu metódu, s časovým krokom $\sigma = 50.0$ a veľkosťou masky $m = 3$. Výsledok prahovania v užívateľskom prostredí je zobrazený na obr. 7.



Obr. 7: Grafické rozhranie, s dátami na ktorých je aplikovaná Niblackova prahovacia metóda.

Vyprahované dátá, boli pridané do zoznamu. Tieto dátá použijeme na vstup do segmentačnej metódy SUBSURF. Zvolíme záložku **3D Filters** detail možností je zobrazený na obr. 4b, kde sa dajú zobraziť 2 funkcie, ktoré môžeme použiť ako počiatočnú podmienku do segmentačnej metódy. Prvou možnosťou počiatočnej podmienky je znamienková funkcia vzdialenosť obr. 8a. Druhá možnosť je výsledok prahovacej funkcie obr. 8b, čo znamená, že takéto dátá majú nastavené súradnice z na -1 , ak prahovacia funkcia zaradila pixel do pozadia a 1 , ak sa jedná o pixel makrofágu. Prahovacia funkcia idúca na vstup, musí byť v oboch prípadoch vyznačená v paneli **Data**.

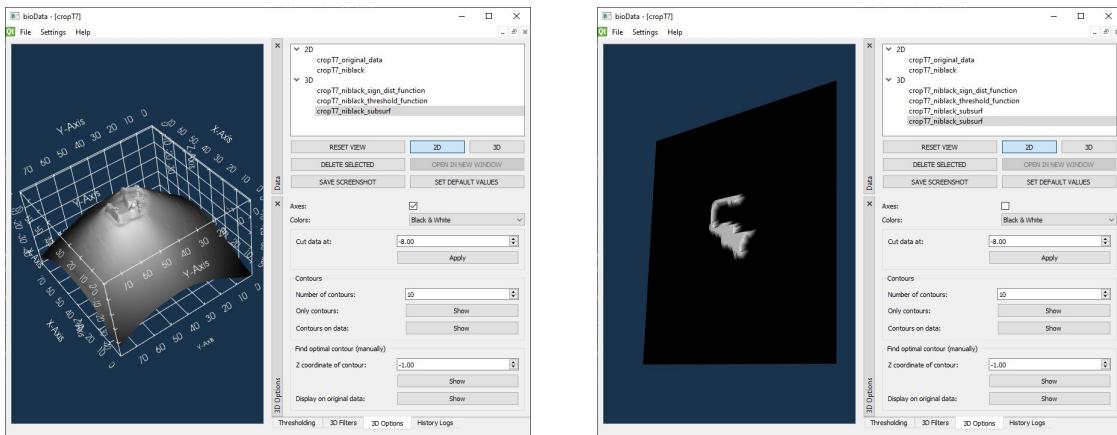


(a) Grafické rozhranie programu zobrazujúce znamienkovú funkciu vzdialenosť spolu s osami x,y,z.

(b) Grafické rozhranie programu zobrazujúce prahovaciu funkciu.

Obr. 8

Typ počiatočnej podmienky, vstupujúcej do segmentačnej metódy SUBSURF zvolíme zo zoznamu. Následne nastavíme požadované parametre metódy: počet časových krokov t , veľkosť lineárneho časového kroku σ , veľkosť nelineárneho časového kroku τ , koeficient sensitivity k . V ukážkovom príklade obr. 9 sú použité hodnoty: $t = 10$, $\sigma = 0.25$, $\tau = 1.0$, $k = 200$, pre oba druhy počiatočných podmienok.

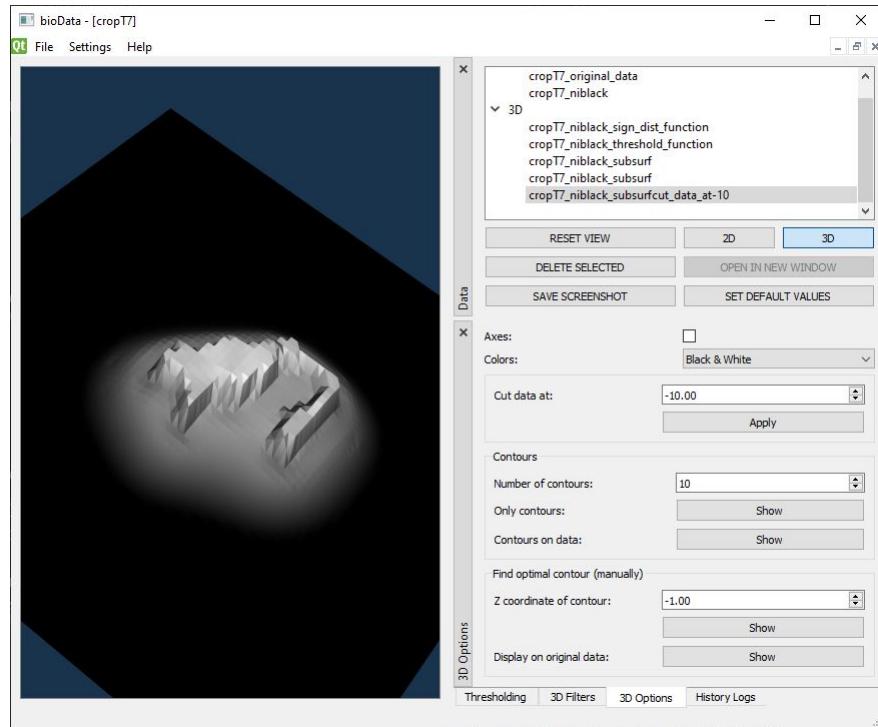


(a) Grafické rozhranie a výsledok segmentačnej funkcie pri použití znamienkovej funkcie vzdialenosť pre počiatočnú podmienku.

(b) Grafické rozhranie a výsledok segmentačnej funkcie pri použití prahovacej funkcie pre počiatočnú podmienku.

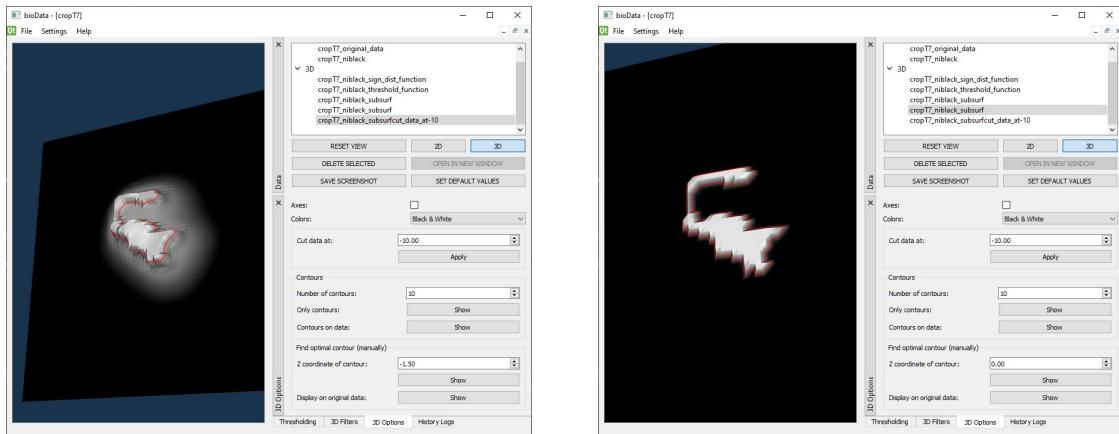
Obr. 9

Na obr. 9a je zobrazený výsledok s počiatočnou podmienkou znamienkovej funkcie vzdialenosť, kde si môžeme všimnúť, že približne od hodnoty -10 neovplyvní výsledok segmentácie, preto ju orežeme na zvolenej hodnote a všetky hodnoty sa nachádzajúce sa pod ňou budú mať hodnotu -10 . Možnosť na orezanie dát sa nachádza v záložke 3D Options obr. 5 a výsledok je zobrazený na obr. 10.



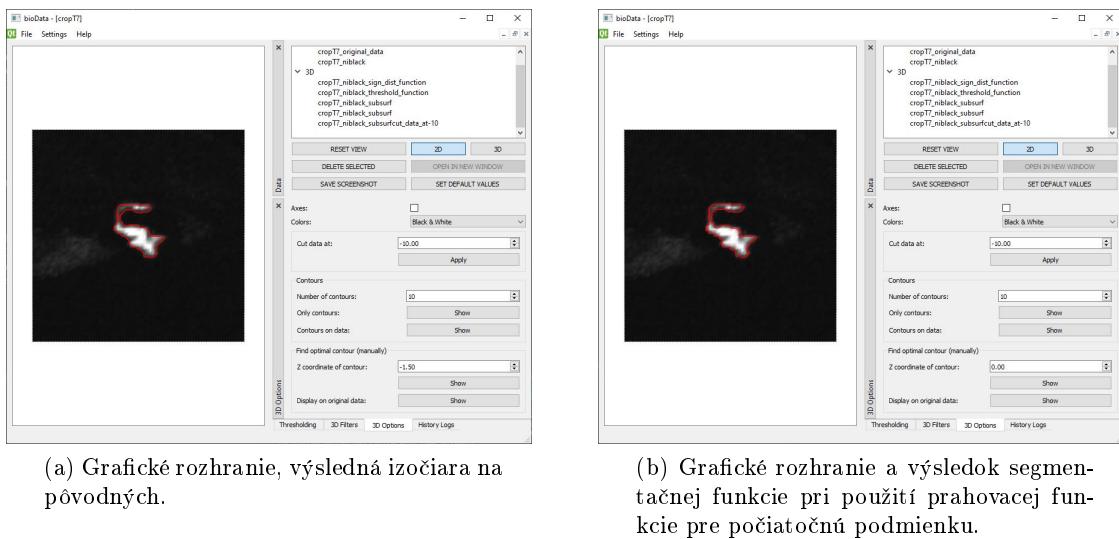
Obr. 10: Grafické rozhranie, s orezanými dátami.

Teraz zvolíme optimálnu izočiaru, v prípade počiatočnej podmienky znamienkovej funkcie vzdialenosť zvolíme izočiaru s hodnotou $z = -1.5$ obr. 11a. V prípade počiatočnej podmienky prahovacej funkcie bude mať izočiara hodnotu $z = 0$ obr. 11b.

(a) Grafické rozhranie, výsledok segmentačnej funkcie pri použití znamienkovej funkcie vzdialenosť pre počiatočnú podmienku a izočiara na hodnote $z = -1.5$.(b) Grafické rozhranie, výsledok segmentačnej funkcie pri použití prahovacej funkcie pre počiatočnú podmienku a izočiara na hodnote $z = -1.5$.

Obr. 11

Nakoniec optimálne zvolenú izočiaru vykreslíme obr. 12 spolu s pôvodnými dátami. Výsledné dátá vieme uložiť pomocou **SAVE SCREENSHOT** tlačidla vo formáte *.png*.

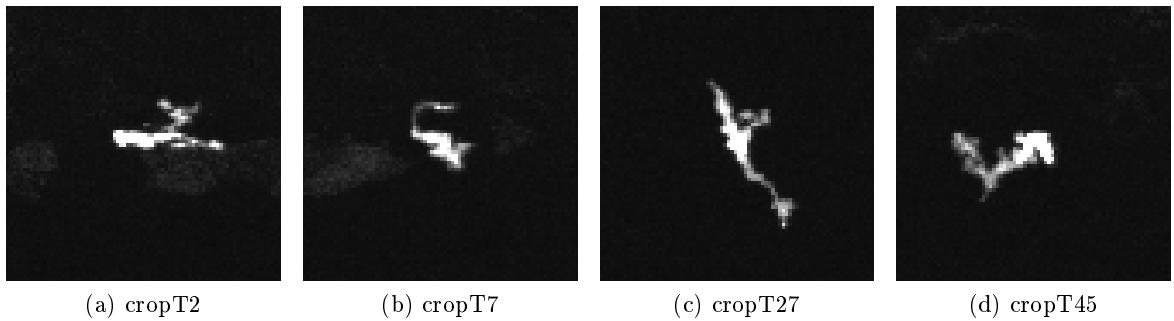


(b) Grafické rozhranie a výsledok segmentačnej funkcie pri použití prahovacej funkcie pre počiatočnú podmienku.

Obr. 12

5 Výsledky

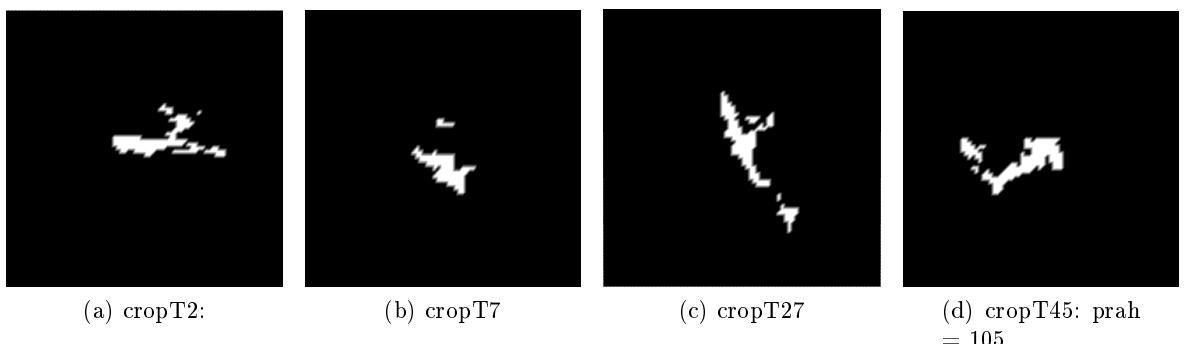
Na ukážku výsledkov sme vybrali niekoľko dát s makrofágmi so zložitými tvarami a šumom v pozadí. Vybrali sme nasledujúce dátá. Obr. 13a bol vybraný z dôvodu, že niektoré časti sú 'odtrhnuté' od zvyšku makrofágu a dátá obsahujú výrazný šum, viditeľný priamo za makrofágom. Dátá na Obr. 13b a Obr. 13c majú užšie časti, so slabšími intenzitami, ktoré je tiež potrebné zahrnúť ako časť makrofágu. Dátá 13d majú na ľavej časti výrazne slabšiu intenzitu.



Obr. 13: Zvolené originálne dátá.

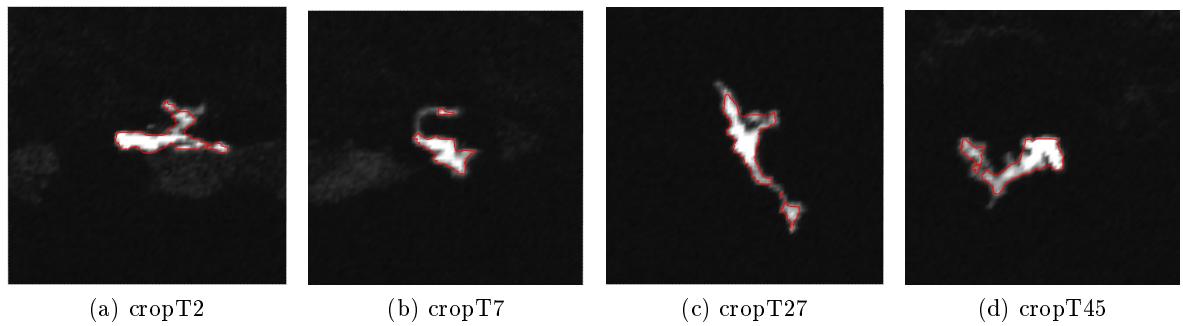
Na vstup do segmentačnej metódy sme použili dva druhy počiatočných podmienok a niekoľko prahovacích metód popísaných v sekcií 2. Úlohou bolo, nájsť počiatočnú podmienku spolu s prahovacou metódou, ktoré by čo najpresnejšie popisovali pôvodný tvar makrofágu. Parametre segmentačnej metódy SUBSURF boli zvolené nasledovne: $t = 15, \sigma = 0.25, \tau = 1.0, k = 200$ a hodnotu výslednej izočiary sme volili podľa toho aký typ počiatočnej podmienky bol zvolený. Pri počiatočnej podmienke danej znamienkovou funkciou vzdialenosť sme izočiaru nastavili na hodnotu $z = -1$ a pri počiatočnej podmienke danej prahovacou funkciou bola izočiara nastavená na $z = 0$, keďže vo väčšine prípadoch boli tieto voľby najvhodnejšími. Výsledky segmentácie sú vyznačené na pôvodných dátach červenou farbou.

Na obr. 14 je zobrazená globálna prahovacia funkcia popísaná v sekcií 2.1.1 používajúca Otsuho metódu. Dá sa všimnúť, že táto prahovacia metóda nezachytila celý tvar makrofágu, hlavne ak išlo o úzku časť s menšou intenzitou.



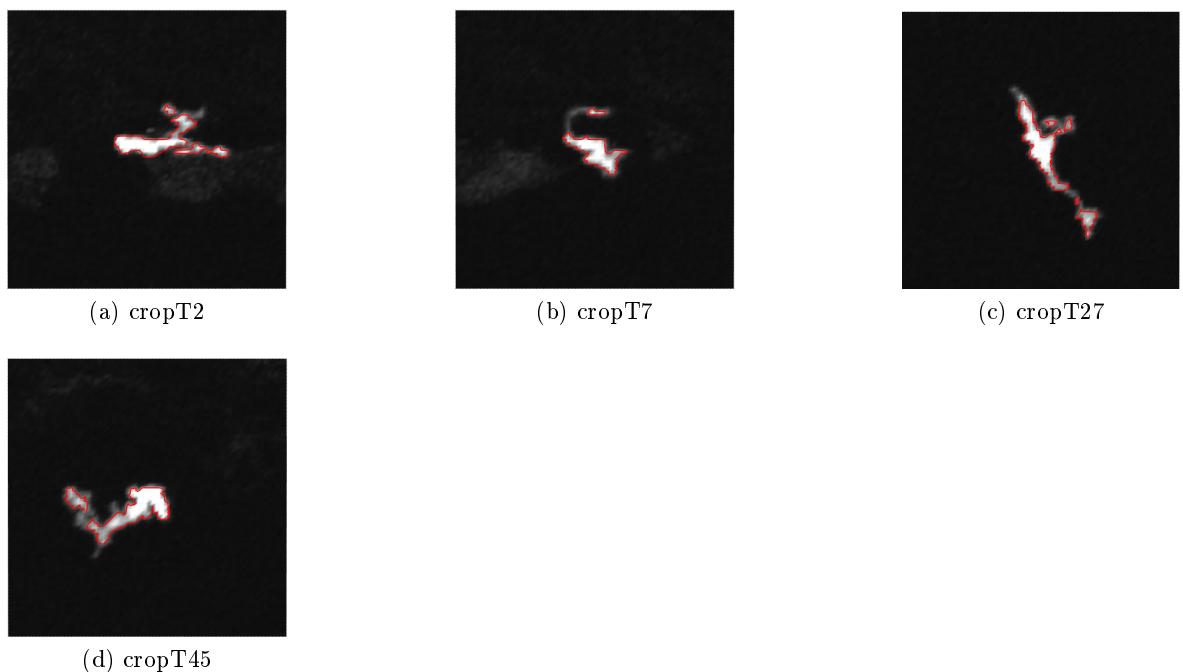
Obr. 14: Otsuho metóda

Izočiara na obr. 16 vyznačujúca výsledok segmentácie, síce Ale môžeme vidieť chýbajúce časti, hlavne na obr 16b a 16c. Ďalej si môžeme všimnúť, že niektoré časti makrofágov, boli vďaka segmentácii prepojené aj keď ich prahovacia metóda roztrhla.



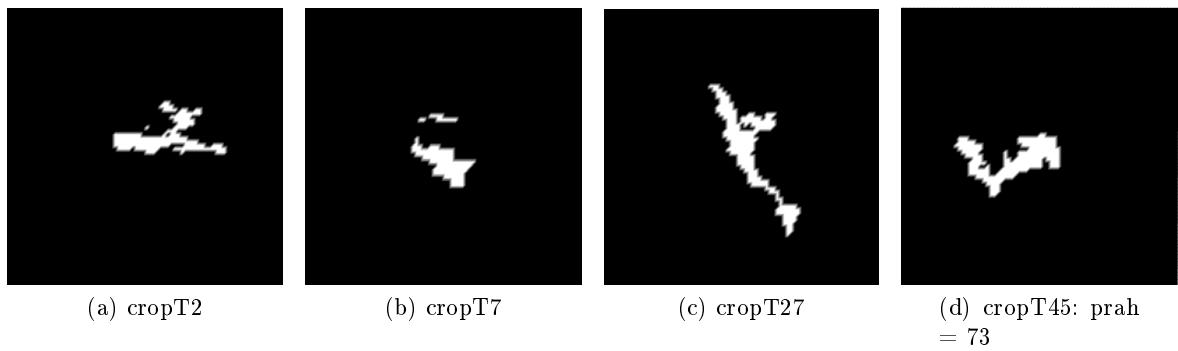
Obr. 15: Výsledok SUBSURFu za použitia Otsuho prahovacej metódy.

VÝSLEDKY - SDF



Obr. 16: Výsledok SUBSURFu za použitia Otsuho prahovacej metódy.

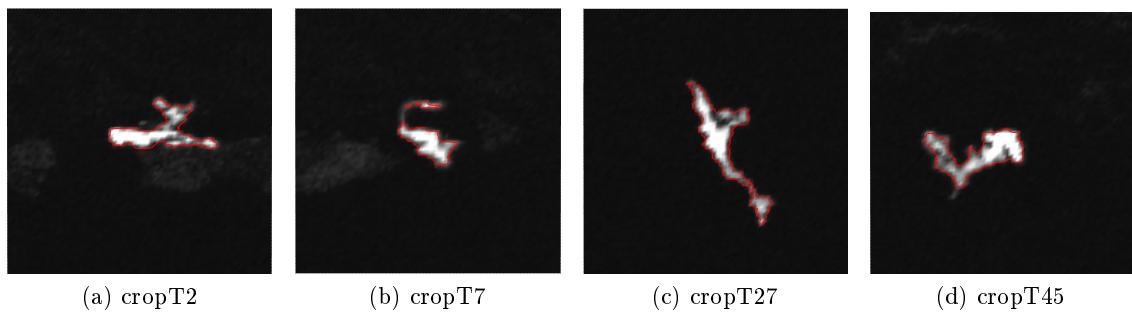
Prahovacia metóda popísaná v sekcií 2.1.2 získaná pomocou maximálnej entropie, je zoobrazená na obr. 17. Môžeme si všimnúť, že prah tejto metódy bol určený o niečo vhodnejšie ako predošlá metóda, napríklad v prípade 17c. Avšak pri tomto type prahovacej metódy sa môže stať, že objekt bude obsahovať aj časti, ktoré pôvodne nemal.



Obr. 17: Prahovanie pomocou maximálnej entropie.

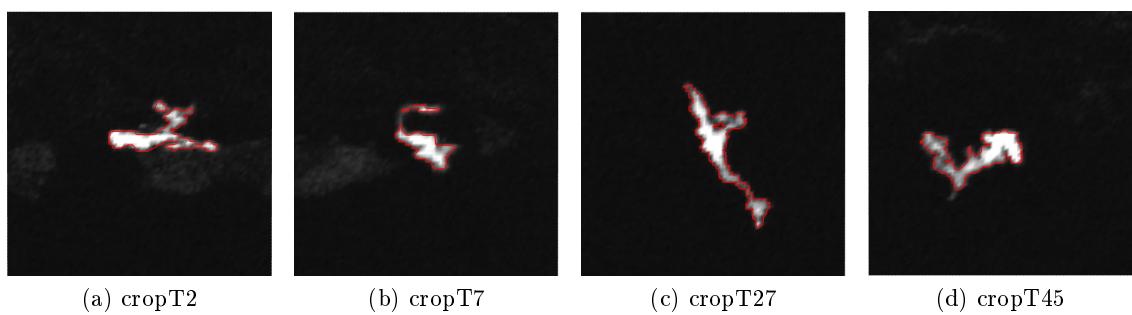
Vo výsledkoch 19 vidieť, že objekty obsahujú niektoré časti, ktoré sa na pôvodných dátach nenachádzajú a v niektorých prípadoch neobsahujú všetky časti.

VÝSLEDKY - SDF



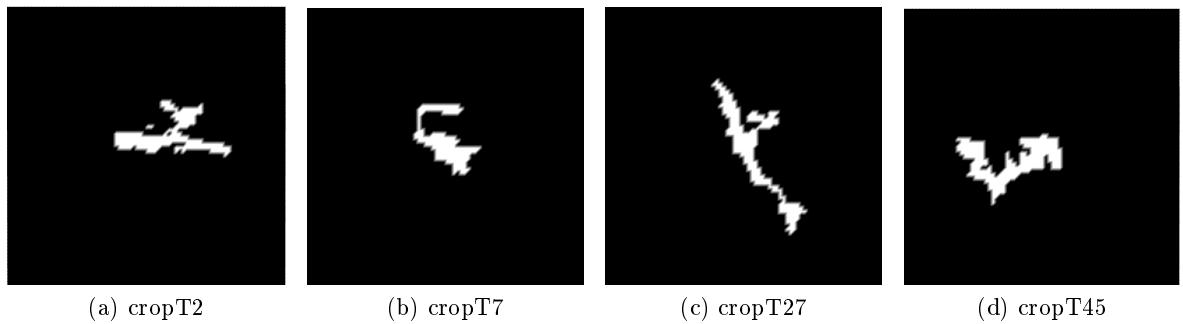
Obr. 18: Výsledok SUBSURFu pre prah nájdený pomocou maximálnej entropie.

VYSLEDKY - TF



Obr. 19: Výsledok SUBSURFu pre prah nájdený pomocou maximálnej entropie.

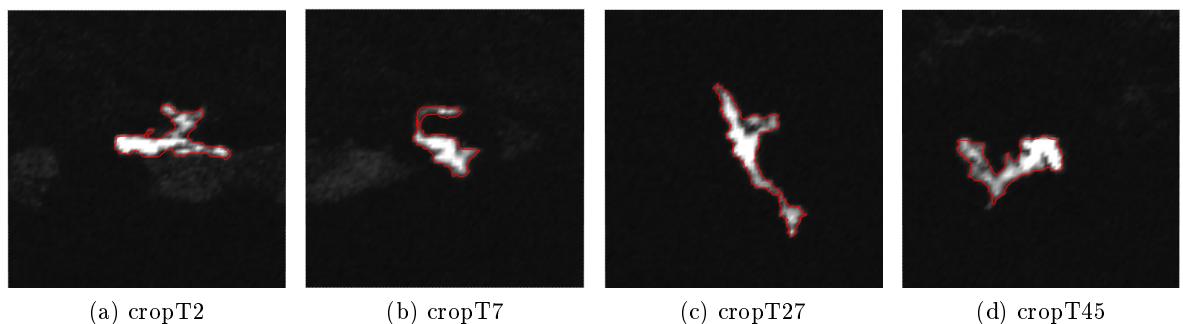
Pri Niblackovej prahovacej metóde popísanej v sekcii 2.2.2, s veľkosťou časového kroku $\sigma = 50.0$, a veľkosťou masky $m = 3$ sú zobrazené na obr. 20.



Obr. 20: Zvolené originálne dátá.

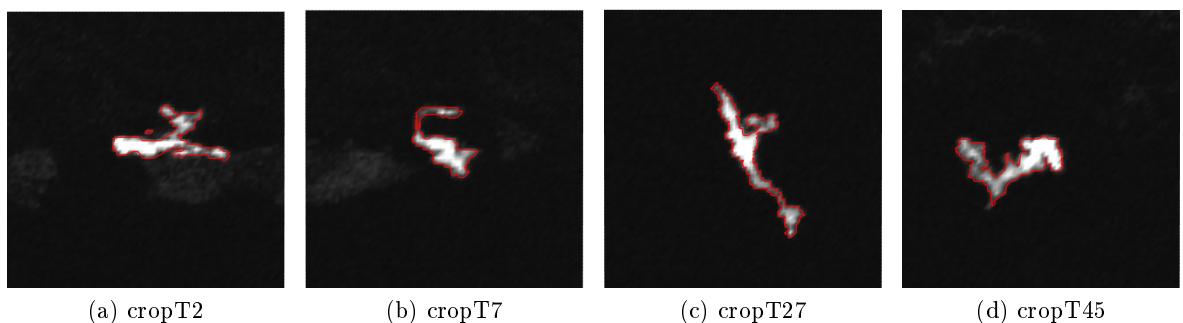
Na obr. 22 sú zobrazené výsledky, kde boli na vstup ako okrajová podmienka uvažované dátá z Niblackovej prahovacej metódy.

VÝSLEDKY - SDF



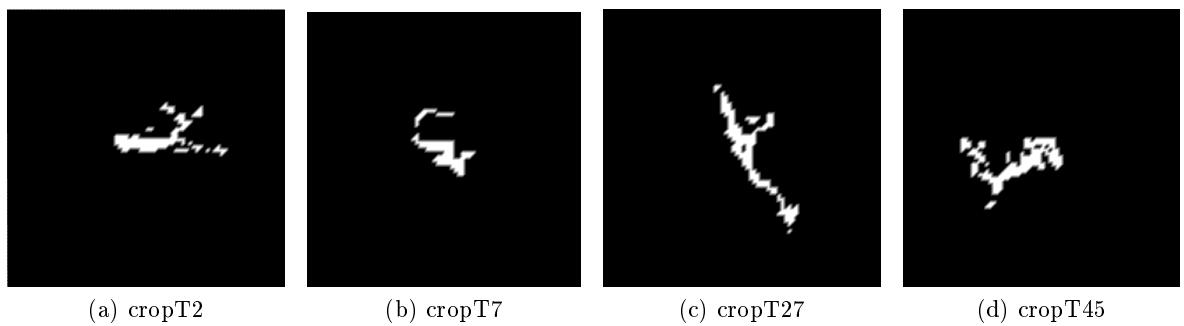
Obr. 21: Niblackova prahovacia metóda.

VÝSLEDKY - TF



Obr. 22: Niblackova prahovacia metóda.

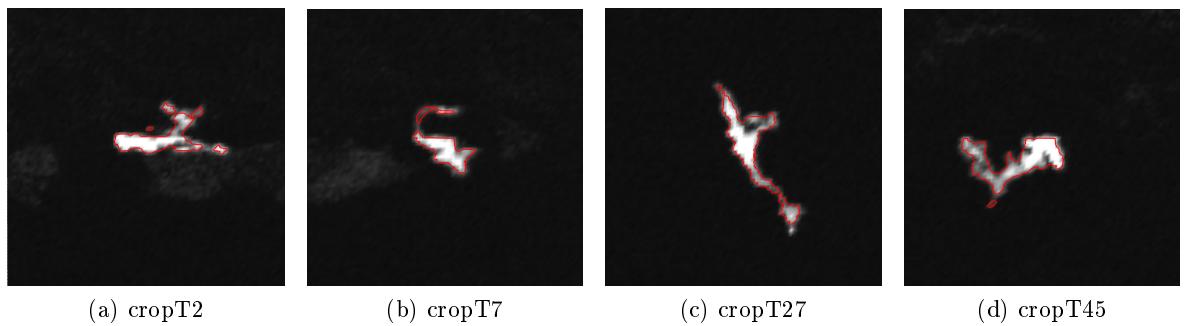
Bernsenova prahovacia funkcia popísaná v sekcií 2.2.1, pre veľkosť okna $m = 2$, je zobrazená na 23.



Obr. 23: Zvolené originálne dátá.

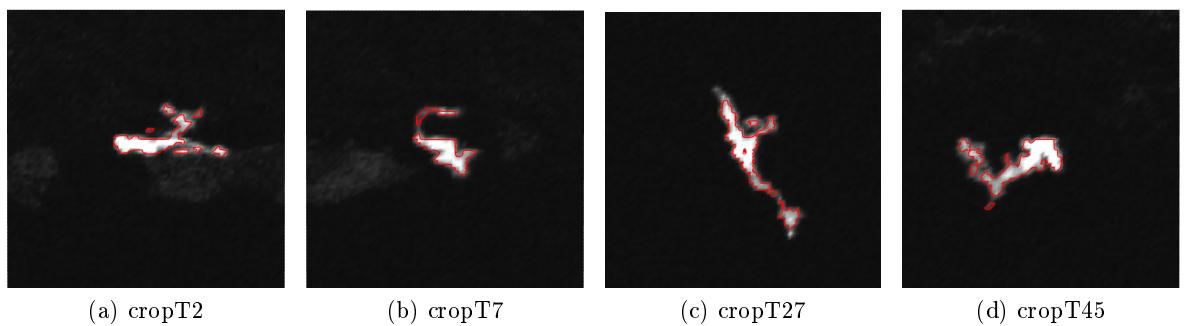
Výsledky segmentačnej metódy, s kde na vstup sme brali Bernsenove prahovanie na obr. 34.

VÝSLEDKY - SDF



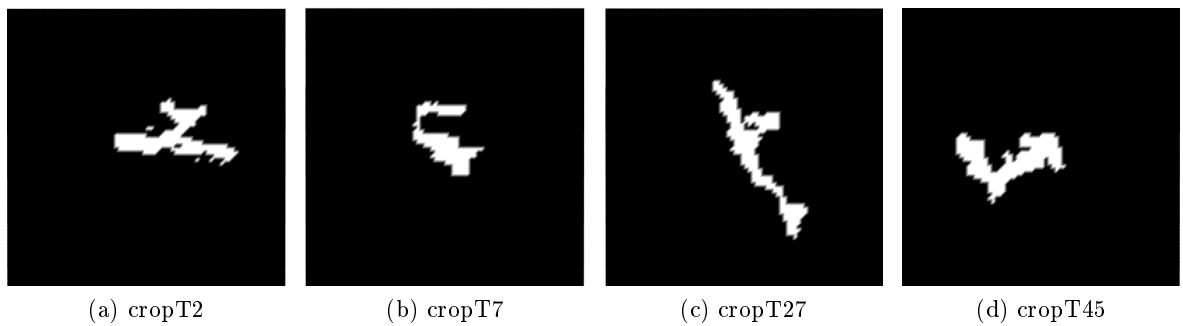
Obr. 24: Zvolené originálne dátá.

VÝSLEDKY - TF



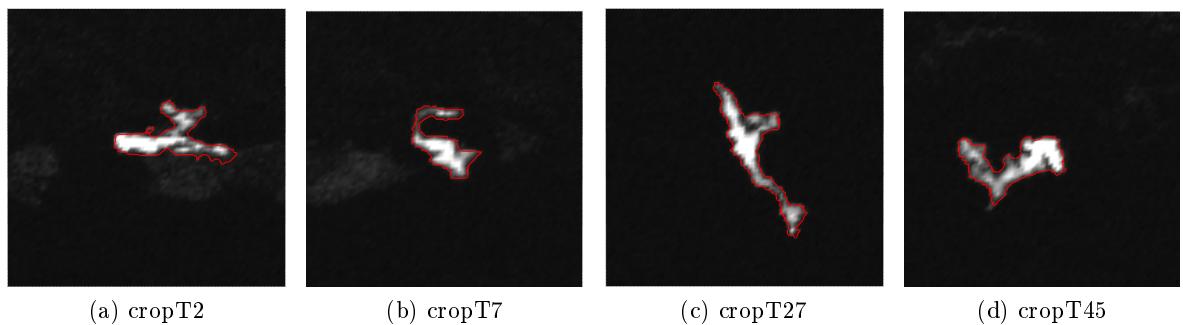
Obr. 25: Zvolené originálne dátá.

HYBRID 1



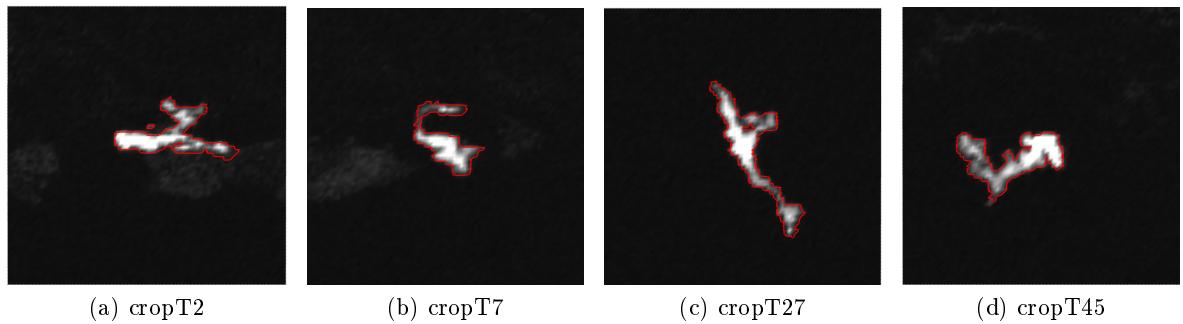
Obr. 26: Zvolené originálne dátá.

VYSLEDKY - SDF



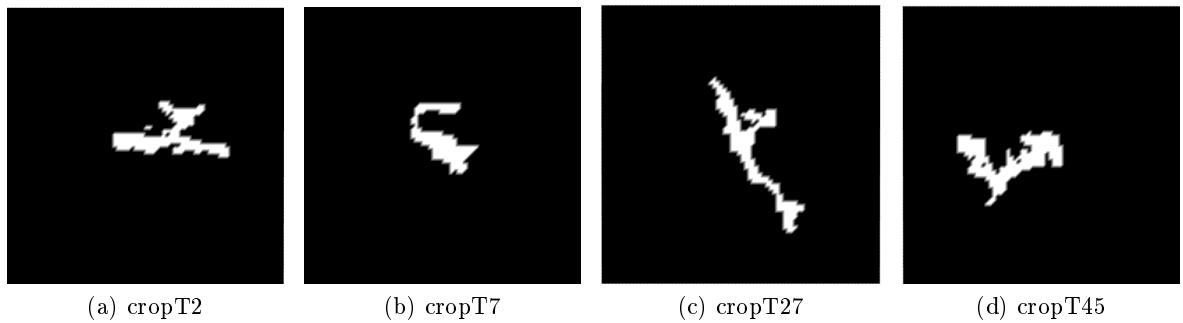
Obr. 27: Zvolené originálne dátá.

VYSLEDKY - TF



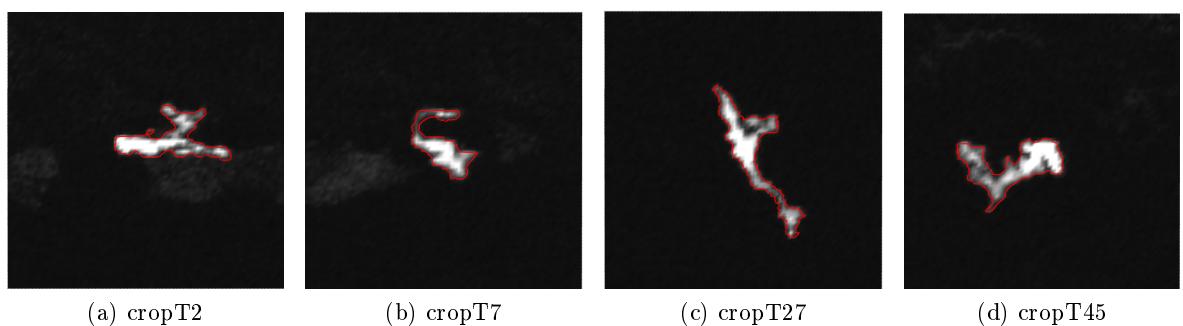
Obr. 28: Zvolené originálne dátá.

SAVOULA



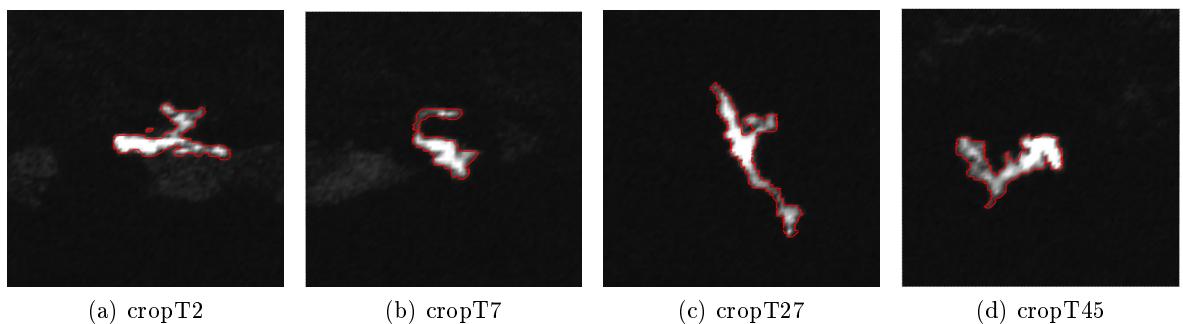
Obr. 29: Zvolené originálne dátá.

VYSLEDKY - SDF



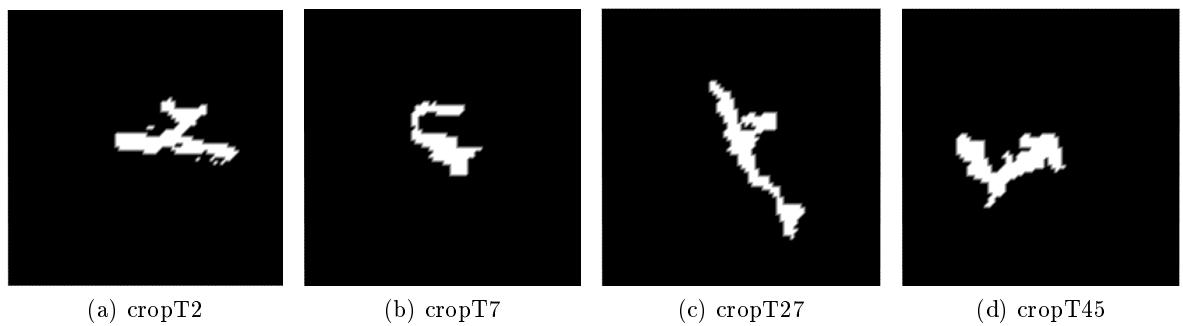
Obr. 30: Zvolené originálne dátá.

VYSLEDKY - TF



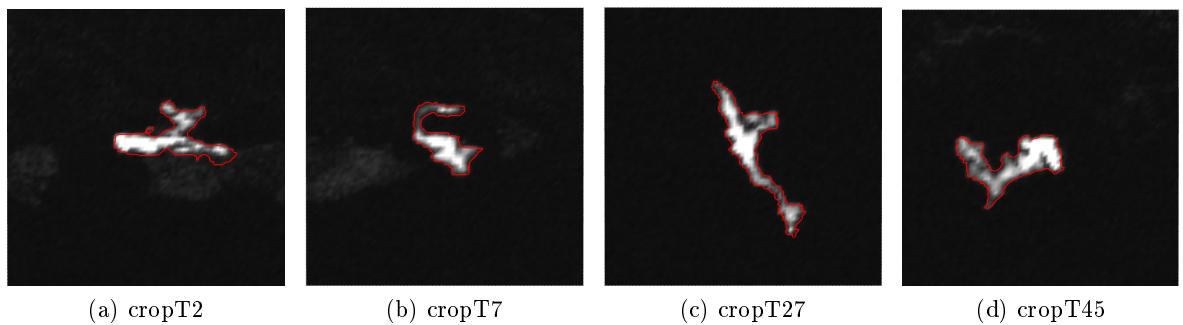
Obr. 31: Zvolené originálne dátá.

HYBRID 2



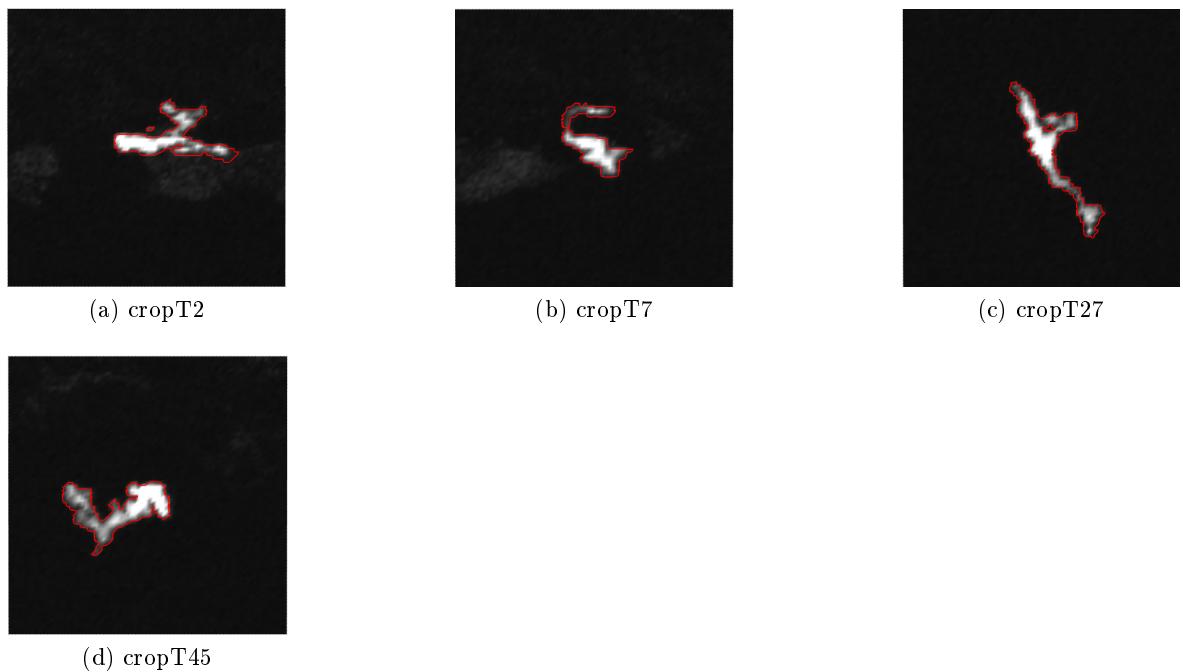
Obr. 32: Zvolené originálne dátá.

VYSLEDKY - SDF



Obr. 33: Zvolené originálne dátá.

VYSLEDKY - TF



Obr. 34: Zvolené originálne dátá.

Z výsledkov na obrázkoch 16, 19, 22 a 34 vidieť, že vhodnejšími prahovacími metódami vstupujúcimi do segmentácie ako okrajová podmienka sú lokálne adaptívne prahovacie metódy.

A Niblackova prahovacia metóda obr. obr [20](#) najlepšie popisuje skutočný tvar makrofágu.

Zo zobrazených výsledkov si môžeme všimnúť že s, že síce počiatočná podmienka prahovej funkcie zachytí drobné výbežky makrováfu ptresnejšie ako

6 Záver

Literatúra

- [1] **VTK dokumentácia:** <https://vtk.org/documentation/>,
- [2] **Qt dokumentácia:** <https://doc.qt.io/>,
- [3] Seol Ah Park, Tamara Sipka, Zuzana Krivá, Martin Ambroz, Michal Kollar, Balazs Kosa, Mai Nguyen-Chi, Georges Lutfalla, Karol Mikula, **Macrophage image segmentation by Thresholding and subjective surface Method:** https://www.researchgate.net/publication/337755994_Macrophage_image_segmentation_by_Thresholding_and_subjective_surface_Method,
- [4] Zuzana Krivá, Karol Mikula, Oľga Stašová, **Spracovanie obrazu vybrané kapitoly z prednášok**,
- [5] Wilhelm Burger, **Principles of Digital Image Processing: Core Algorithms**,
- [6] Anna Korzynska, Lukasz Roszkowiak, Carlos Lopez, Ramon Bosch, Lukasz Witkowski1, Marylene Lejeune **Validation of various adaptive threshold methods of segmentation applied to follicularlymphoma digital images stained with 3,3'-Diaminobenzidine&Haematoxylin:** <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3656801/>,
- [7] Nobuyuki Otsu, **A Threshold Selection Method from Gray-Level Histograms**,