SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STAVEBNÁ FAKULTA

Automatické segmentačné metódy biologických dát

Diplomová práca

Študijny program:	Matematicko-počítačové modelovanie
Študijny 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

V rôznych vedných disciplínach ale aj v bežnom živote sa používajú rôzne aplikácie spracovania obrazov. Spracovanie obrazov je metóda, ktorá pomocou rôznych matematických operácii a algoritmov upravuje obrazové dáta rôznych formátov a pomáha z nich získavať užitočné informácie. Obrazové dáta 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, 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 maktofág 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 funkcionalita 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 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 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 $(0, 1, \ldots, q)$ a C_1 obsahuje všetky zvyšné pixle nachádzajúce sa na intervale $(q + 1, \ldots, K - 1)$, teda

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

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 $(q_1, \ldots, q_n)$, kde n reprezentuje počet intenzít v danom obraze.

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

$$p_i = \frac{h_i}{N}, \qquad \sum_{i=1}^{N} p_i = 1, \tag{2}$$

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

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

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

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

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

$$\sigma^2 = \sigma_0 + \sigma_1. \tag{5}$$

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

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

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

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

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

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

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

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

Celková entropia pre daný prah q je daná ako

$$H(q) = H_0(q) + H_1(q); \tag{10}$$

Kumulatívna pravdepodobnosť P_0 je definovaná ako

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

a sumačné podmienky S_0, S_1 sú predpočítané a definované ako

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

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

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

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

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

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

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 limitom c_{min}. Ak $c(u, v) < c_{min}$, tak sa predpokladá, že pixle zodpovedajúce jednej oblasti patria do jednej 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 smerodajnú odchýlku $\sigma_R(u, v)$, v tvare

$$Q(u, v) = \mu_R(u, v) + \kappa\sigma_R(u, v). \tag{16}$$

Lokálny prah je určený pridaním konštanty $\kappa \geq 0$ k smerodajnej odchýlke $\sigma_R(u, v)$ a lokálneho priemeru $\mu_R(u, v)$.Na získanie $\sigma_R(u, v)$ a $\mu_R(u, v)$ sme použili rovnicu vedenia tepla, tiež známu ako lineárno-difúznu rovnicu. Táto rovnica je považovaná za najstaršiu filtračnú metódu v spracovaní obrazu, založenú na PDR.

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

$$\frac{\partial u(x,t)}{\partial t} = \Delta u(x,t), \tag{17}$$

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

$$\frac{\partial u(x,t)}{\partial \vec{n}} = 0, \tag{18}$$

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

$$u(x,0) = u^0(x), \tag{19}$$

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

$$Q(u,v) = \mu_R(u,v) + \kappa\sigma_R(u,v) + d, \tag{20}$$

kde $d \geq 0$.

PRIDAT AKE SU OPTIMALNE HODNOTY κ a d

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 nefungovala poriadne a 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

$$u_t = \sqrt{\varepsilon^2 + |\nabla u|^2}\nabla.(g\frac{\nabla u}{\sqrt{\varepsilon^2 + |\nabla u|^2}}), \tag{21}$$

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

$$g(s) = \frac{1}{1+Ks^2}, K > 0, \tag{22}$$

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

3 Numerické schémy

Vo všeobecnosti môže byť numerická schéma vysvetlená ako algoritmický popis, ktorý nájde správne riešenie matematického problému. Pri implementácii boli použité nasledujúce numerické schémy:

- Implicitná metóda pre rovnicu vedenia tepla,
- Semi−implicitná metóda pre SUBSURF.

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

$$\frac{u^n - u^{n-1}}{\tau} = \Delta u^n = \nabla.(\nabla u^n) \tag{23}$$

a pre ľubovoľnú veľkosť časového kroku je bezpodmienečne stabilná.
OBRAZOK S OZNACENIM HRAN
Rovnicu (23) integrujeme cez konečný objem p a dostaneme rovnicu v tvare

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \int_p \nabla.(\nabla u^n) dx. \tag{24}$$

Na pravú stranu rovnice aplikujeme Greenovu vetu a dostaneme

$$\int_p \nabla.(\nabla u^n) dx = \int_{\partial p} \nabla u^n \vec{n}_p dS, \tag{25}$$

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

$$\int_{\partial p} \nabla u^n \vec{n}_p dS = \sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \vec{n}_{pq} dS, \tag{26}$$

vieme napísať slabú konečno-objemovú formuláciu úlohy

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \vec{n}_{pq} dS = \sum_{q \in N(p)} \int_{e_{pq}} \frac{\partial u^n}{\partial \vec{n}_{pq}} dS, \tag{27}$$

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

$$\int_p \frac{u^n - u^{n-1}}{\tau} dx = \frac{u_q^n - u_p^n}{\tau} m(p). \tag{28}$$

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

$$\nabla u^n \vec{n}_{pq} \approx \frac{u_p^n - u_p^{n-1}}{d_{pq}}. \tag{29}$$

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

$$\sum_{q \in N(p)} \int_{e_{pq}} \nabla u^n \vec{n}_{pq} dS \approx \sum_{q \in N(p)} m(e_{pq}) \frac{u_p^n - u_p^{n-1}}{d_{pq}}. \tag{30}$$

Po dosadení dostávame implicitnú schému

$$\frac{m(p)(u_p^n - u_p^{n-1})}{\tau} = \sum_{q \in N(p)} m(e_{pq})(u_q^n - u_p^n) = \nabla.(\nabla u^n) \tag{31}$$

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

$$u_p^n + \frac{\tau}{m(p)} \sum_{q \in N(p)} (u_q^n - u_p^n) = u_p^{n-1}. \tag{32}$$

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

$$(1 + \frac{\tau}{h^2} \sum_{q \in N(p)} 1) u_p^n - \frac{\tau}{h^2} \sum_{q \in N(p)} u_q^n = u_p^{n-1}. \tag{33}$$

Rovnicu (33) dostaneme na každom konečnom objeme p. Následne schému riešime pomocou super-relaxačnej metódy (SOR).

3.2 Semi−implicitná metóda pre SUBSURF

V prvom rade urobíme semi−implicitnú časovú diskretizáciu, tak že rovnicu (21) vydelíme členom $\sqrt{\varepsilon^2 + |\nabla u|^2}$ a dostaneme

$$\frac{1}{|\nabla u^{n-1}|_\varepsilon} . \frac{u^n - u^{n-1}}{\tau} = \nabla.(g^0 \frac{\nabla u^n}{|\nabla u^{n-1}|_\varepsilon}), \tag{34}$$

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 (34) cez konečný objem p a dostaneme

$$\int_p \frac{1}{|\nabla u^{n-1}|_\varepsilon} . \frac{u^n - u^{n-1}}{\tau} dx = \int_p \nabla.(g^0 \frac{\nabla u^n}{|\nabla u_p^{n-1}|_\varepsilon}) dx, \tag{35}$$

Následne na pravú stranu rovnice aplikujeme Greenovu vetu

$$\int_p \frac{1}{|\nabla u^{n-1}|_\varepsilon} . \frac{u_p^n - u_p^{n-1}}{\tau} dx = \int_{\partial p} g^0 \frac{\nabla u^n}{|\nabla u_p^{n-1}|_\varepsilon} \vec{n_{pq}}_p dS, \tag{36}$$

Derivácia v smere vonkajšej normály ku konečnému objemu p je v rovnici (36) vyjadrená ako $\nabla u^n . \vec{n_{pq}} = \frac{\partial u^n}{\partial \vec{n_p}}$. Na ľavej strane rovnice (36) budeme uvažovať konštantnú reprezentáciu riešenia a jeho gradientu. Normálovú deriváciu z pravej strany rovnice nahradíme konečnou diferenciou

$$\frac{m(p)}{|\nabla u_p^{n-1}|_\varepsilon \tau} (u_p^n - u_p^{n-1}) = \sum_{q \in N(p)} \int_{e_{pq}} g_{pq}^0 \frac{\nabla u^n}{|\nabla u_{pq}^{n-1}|_\varepsilon} \vec{n}_{pq} ds, \tag{37}$$

$$\frac{m(p)}{|\nabla u_p^{n-1}|_\varepsilon \tau}(u_p^n - u_p^{n-1}) = \sum_{q\in N(p)} \int_{e_{pq}} g_{pq}^0 \frac{u_q^n - u_p^n}{d_{pq}} . \frac{1}{|\nabla u_{pq}^{n-1}|_\varepsilon} ds, \tag{38}$$

$$\frac{m(p)}{|\nabla u_p^{n-1}|_\varepsilon \tau}(u_p^n - u_p^{n-1}) = \sum_{q\in N(p)} g_{pq}^0 T_{pq} \frac{1}{|\nabla u_{pq}^{n-1}|_\varepsilon}(u_p^n - u_p^{n-1}), \tag{39}$$

$$u_p^n - u_p^{n-1} = \frac{|\nabla u_p^{n-1}|_\varepsilon \tau}{m(p)} \sum_{q\in N(p)} g_{pq}^0 T_{pq} \frac{1}{|\nabla u_{pq}^{n-1}|_\varepsilon}(u_p^n - u_p^{n-1}), \tag{40}$$

$$u + p^{n-1} = u_p^n + \sum_{q\in N(p)} \frac{|\nabla u_p^{n-1}|_\varepsilon \tau}{m(p)} \frac{g_{pq}^0 T_{pq}}{|\nabla u_{pq}^{n-1}|_\varepsilon}(u_p^n - u_p^{n-1}), \tag{41}$$

$$u + p^{n-1} = (1 + \sum a_{pq}) u_p^n - \sum a_{pq} u_q^n, \tag{42}$$

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++.

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 vrámci 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 QScrollAree 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 vrámci 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 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.

4.1.1 VTK

Visualization Toolkit (VTK) sú voľne 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 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 hlášok.
- **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.
- **vtkPolyDataMapper**
- **vtkActor**
- **vtkRenderer**

4.2 Grafické užívateľské rozhranie

V tejto sekcii sa oboznámime s vizuálnou stránkou vytvoreného softvéru a popíšeme funkcionalitu. 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. V každej z týchto položiek sa nachádzajú možnosti

5 Výsledky

6 Záver

Literatúra