

SEGMENTÁCIA MEDICÍNSKYCH DÁT POMOCOU POHYBUJÚCICH SA ROVINNÝCH KRIVIEK

Jozef Urbán, Karol Mikula

Úvod

Zdravotná starostlivosť predstavuje dôležitú súčasť kvality života ľudskej spoločnosti. Je s ňou konfrontovaný každý človek, preferujúci kvalitnú liečbu. Tá je spojená s diagnózou, ktorá často závisí od skúseností lekára a intenzity symptómov konkrétnej choroby. Dnes však pri liečbe a diagnostike už zohráva veľkú úlohu technika. Rozvoj priemyslu a počítačovej techniky dnes ponúka lekárom mnohé možnosti, ako zdokonaľiť diagnózu i liečbu pacientov.

Spoločnosť Tatramed spol. s r.o. so sídlom v Bratislave vyvíja od roku 1992 softvérové diagnostické a terapeutické systémy založené na trojrozmernej vizualizácii obrazových dát pacientov. Pod obrazovými dátami rozumejme snímky z počítačovej tomografie (computer tomography - CT), magnetickej rezonancie (MR - magnetic resonance), pozitronovej emisnej tomografie (PET - positron emission tomography), angiografie a skiografie.

Po vyšetrení pacienta niektorou z daných metód si lekár môže pomocou spomínaného softvéru prezrieť série jeho snímok. Takto jednoduchým spôsobom môže robiť diagnózu pacienta vyšetreného v jednej nemocnici odborník, ktorý si jeho snímky prezrie vo svojej ambulancii na počítači. Pomocou jednotlivých softvérových nástrojov môže snímky takmer dokonale diagnostikovať a označiť v nich dôležité miesta, napísať k nim správu a pripraviť tak materiál pre úspešnú ďalšiu liečbu. Softvér tiež ponúka možnosť segmentácie objektov v jednotlivých snímkach. Po detekcii hrán toho istého objektu vo viacerých snímkach nasledujúcich za sebou je možné z nich vytvoriť približný trojrozmerný model daného objektu.

Takto lekárom vytvorený model má viacero využití. Môže sa použiť pri plánovaní ožarovania pacientov pri rádioterapii, kde je potrebné mať okontúrované (vysegmentované) orgány, ktoré sú rizikové z hľadiska ožiarenia a taktiež nádor, na ktorý má byť celkové ožarovanie zamerané. Na základe týchto modelov fyzici navrhnu vhodné dávky a smery ožiarenia. Trojrozmerný model je tiež výbornou pomôckou z hľadiska vizualizácie. Lekár vďaka nemu môže napríklad pri operácii nádoru mozgu plánovať spôsob a smer najlepšieho prístupu k danému nádoru. V neposlednom rade sa tieto modely využívajú pri porovnávaní štúdií pacientov pred a po zákroku, alebo ožarovaní. Aplikácia, ktorú firma vyvinula dokáže totiž počítať objemy týchto modelov a teda sa dá s jej pomocou napríklad zisťovať, či sa vnútorné krvácanie zastavilo, či sa krv správne vstrebáva, alebo či nádor ešte rastie a pod.

V tejto oblasti potrebovala firma Tatramed spol. s r.o. pre svoj produkt TomoCon vyvinúť nový segmentačný nástroj. Pomocou neho by si lekár mohol jednoducho, semi-automaticky, detekovať hranice dôležitých objektov v medicínskych dátach.

Momentálne produkt firmy poskytuje lekárovi pre segmentáciu hrán objektov tieto tri jednoduché nástroje:

- **Freehand (voľná ruka)** V tomto prípade ide o najjednoduchší prístup z pohľadu

vývoja, ale zároveň asi o najpracnejší z pohľadu lekára. Používateľ v tomto prípade doslova voľnou rukou kreslí hranice zvoleného objektu. To je však pre presné vystihnutie hrán i pri použití tabletu časovo náročné a vyžaduje si to primeranú zručnosť pri segmentácii.

- **Polygon (mnohouholník)** Ide o vyznačenie okraja pomocou zadaných uzlových bodov, ktoré sú následne spájané úsečkami. Tento prístup je pre používateľa rýchlejší ale zároveň je nepresný, pretože segmentované hrany sú väčšinou zakrivené a ich aproximácia pomocou úsečiek prináša pomerne veľkú nepresnosť.
- **Magic Wand (kúzelnícka palička)** Posledný nástroj pracuje na báze zjednocovania susedných bodov snímky s podobnou hodnotou úrovne šedi. Využíva sa pri ňom zrejme istý druh region growing algoritmu. Ťahaním myši užívateľ určuje približné hranice vysegmentovaného objektu, pričom zároveň znižuje, resp. zväčšuje toleranciu intenzít šedi v susedných bodoch. Tento nástroj je rýchly a pohodlný, no je vhodný predovšetkým pre veľké a pekne členené tvary. Väčšinou je nutná ešte následná úprava objektu získaného segmentáciou.

Úlohy segmentácie objektov, prípadne automatickej detekcie hrán objektov v obraze, možno modelovať pohybujúcimi sa krivkami. Takýto prístup nazývaný aj aktívne kontúrovanie, bol do spracovania obrazu zavedený v [6]. Pohyb krivky, ktorá má automaticky nájsť hranicu objektu, môže byť riadený vhodne zvoleným rýchlostným polom [2, 7], regularizovaný krivosťou [4, 3, 1], elastickou energiou krivky [11] a stabilizovaný vhodnou voľbou tangenciálnych rýchlostí bodov krivky [5, 8, 9, 10, 11]. Vhodná numerická aproximácia polohy krivky v obraze je dôležitou súčasťou praktických segmentačných úloh a vyžaduje netriviálne matematické prístupy.

Hlavným cieľom tejto práce je vytvorenie matematického modelu a počítačového programu, použiteľného pre semi-automatizované vyhľadávanie hraníc objektov v medicínskych dátach. Vyvinutý nástroj má pracovať rýchlo a jeho použitie má byť jednoduché. Predpokladá sa použitie medicínskych obrazových údajov zo základných vyšetrovacích modalít typu CT, MRI, PET. Užívateľ bude zadávať pomocou kurzora vždy začiatkový a koncový bod krivky. Jej tvar bude dopyčovaný na základe obrazovej informácie vstupných dát. Výsledná hranica bude reprezentovaná bodmi, ležiacimi na výslednej krivke. Pokúsime sa docieľiť rôznymi metódami, aby distribúcia bodov krivky bola čo najrovnomernejšia. Výsledky dosiahnuté týmito metódami porovnáme.

1 Vstupné dáta

Predtým, ako sa začneme zaoberať samotnými krivkami a ich pohybom, predstavme si najskôr vstupné dáta, s ktorými budeme pracovať a na ktorých predovšetkým budú závisieť naše výsledky.

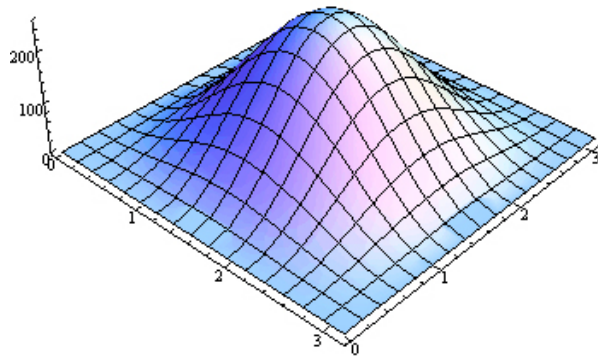
1.1 Reprezentácia vstupných dát

Vstupnými dátami budú v našom prípade obrázky v stupňoch šedi. Najskôr sme pracovali s obrázkami formátu *.PGM (portable graymap file format) ale neskôr po vytvorení

užívateľského rozhrania sme prešli na formát *.BMP (bitmap). Formát *.PGM, ako sám názov hovorí je určený pre obrazy v stupňoch šedi, čo naznačuje, že aj jeho štruktúra je jednoduchšia ako pri bitmapách typu *.BMP. Vo všeobecnosti si však oba formáty môžeme pre jednoduchosť predstaviť podobne. A to ako hlavičku, ktorá obsahuje dôležité údaje ako rozmery obrázka v bodoch (*.BMP obsahuje okrem iného aj údaj o veľkosti súboru, počet bitov na bod a tiež dôležitú paletu farieb, ktorá závisí od počtu farieb obrázka) a obrazové dáta. Tie sú pre nás najdôležitejšie. Vo všeobecnosti sú to čísla od 0 do 255 ktoré predstavujú hodnoty intenzít šedi pre jednotlivé body obrázka, pričom hodnota 0 predstavuje čiernu farbu a hodnota 255 bielu. Teda každému bodu vstupného obrázka prislúcha jedna intenzita. Pre úplnosť pri formáte *.BMP toto nemusí platiť. Záleží od spôsobu uloženia súboru. Ten totiž môže byť pre každý bod reprezentovaný tromi hodnotami intenzít a to červenej, zelenej a modrej farby, čo sme v našej aplikácii museli ošetriť. Keďže ale pracujeme s obrázkami v stupňoch šedi, v prípade *.BMP pôjde o tri rovnaké hodnoty, čo nám prácu zjednodušuje. Podstatným pre nás ostáva, že vo vstupných dátach dostávame rozmery obrázka a hodnoty celých čísiel dané pre každý jeho bod.

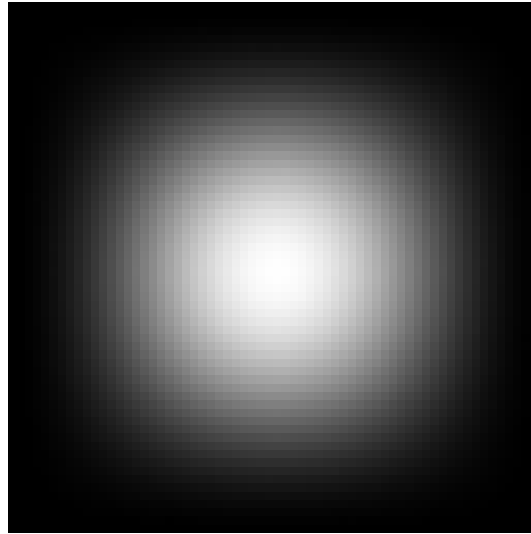
Vstupný obrázok si teda môžeme predstaviť ako trojrozmerný graf funkcie dvoch premenných. Kde hodnota funkcie je celočíselná intenzita šedi pre daný bod. Tu sa však stretávame s problémom, ktorý budeme musieť riešiť ako prvý. Obrázok sa nám totiž javí na prvý pohľad spojité, ale ak si ho vykreslíme v troch rozmeroch, tak zreteľne môžeme badať hrany alebo šum spôsobené celočíselnými hodnotami intenzít. Pre názornú ukážku sme vytvorili sami obrázok na základe funkcie danej analyticky.

$$u(x, y) = 255 \sin^2(x) \sin^2(y) \quad (1)$$



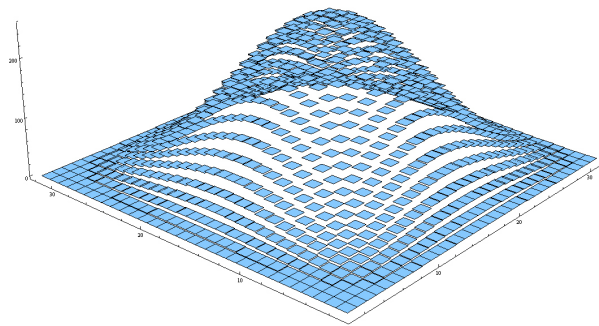
Obrázok 1.1: Graf funkcie (1) vykreslená v systéme Wolfram Mathematica 6, pomocou príkazu Plot3D

Ako bolo už spomenuté, funkcia (1) predstavuje hodnoty intenzít šedi pre body obrázka. Obrázok, ktorý sme vytvorili je reprezentovaný celočíselnými hodnotami funkcie v stredoch pixelov. Znova sme ho načítali a vykreslili jeho graf v troch rozmeroch. Na



Obrázok 1.2: Obrázok vytvorený na základe analytickej funkcie (1)

vykreslenie sme použili príkaz `ListPlot3D` systému Mathematica. Najskôr sme v tomto príkaze použili voľbu `InterpolationOrder -> 0`, ktorá nám poskytla pohľad, ako je reálne náš obrázok reprezentovaný po častiach konštantnou funkciou intenzity šedi. Parameter "data", v uvedenom príkaze predstavuje pole naplnené hodnotami intenzít šedi obrázka vytvoreného podľa funkcie (1) Potom sme použili pre rovnaký príkaz voľbu `InterpolationOrder -> 1`, ktorá pomocou interpolácie prvého stupňa pospájala hodnoty funkcie v jednotlivých bodoch, pričom vidíme stratu hladkosti oproti grafu na obrázku 1.1.



Obrázok 1.3: `ListPlot3D[data, Mesh -> False, Boxed -> False, InterpolationOrder -> 0]`

1.2 Predspracovanie vstupných dát

Ako je vidno na obr. 1.4, dáta, ktoré dostávame z obrázka vytvoreného z hladkej funkcie, nám vracajú síce podobné, ale veľmi "znečistené" hodnoty. Je zrejmé, že túto



Obrázok 1.4: `ListPlot3D[data, Mesh -> False, Boxed -> False, InterpolationOrder -> 1]`

nepresnosť spôsobuje celočíselná reprezentácia hodnôt (tzv. kvantizácia) v bodoch obrázka a samotný spôsob, akým je obrázok definovaný. Je totiž určený len v diskretných bodoch a nie spojitě v uvažovanej oblasti (tzv. vzorkovanie). Naše medicínske vstupné dáta budú mať vo všeobecnosti tento charakter. Hoci nemusí byť až takto výrazný, treba ho riešiť. Načítanú funkciu danú hodnotami v diskretných bodoch musíme nejakým spôsobom zhladiť. Jedným zo spôsobov je aplikovať na vstupné údaje rovnicu nestacionárneho vedenia tepla

$$\frac{\partial u}{\partial t} = \Delta u \quad (2)$$

začínajúc z počiatočnej podmienky danej originálnym obrazom. Keďže pracujeme s diskretnými dátami, k výpočtu pristúpime numericky a použijeme numerickú schému

$$\frac{u_{ij}^{m+1} - u_{ij}^m}{\sigma} = \frac{u_{i-1j}^m - 2u_{ij}^m + u_{i+1j}^m}{h^2} + \frac{u_{ij-1}^m - 2u_{ij}^m + u_{ij+1}^m}{h^2}, \quad (3)$$

kde u_{ij}^m predstavuje intenzitu v diskretnom bode danom súradnicami (i, j) v čase m , kde $m \in \{0, 1, \dots, M\}$. To znamená, že hodnoty u_{ij}^0 predstavujú pôvodný obrázok. $T = M\sigma$ je zvolený zhladzovací časový interval, σ je časový krok a M určuje, koľkokrát budeme aplikovať na dané dáta diskretné zhladenie. Konštanta h je vzdialenosť medzi susednými bodmi obrázka, t.j. veľkosť pixela. Pre jednoduchosť uvažujeme $h = 1$ a kvôli stabilite musí platiť $\sigma \leq h^2/4$. My sme v našej aplikácii zvolili $\sigma = 0.25$. Po úprave dostávame pre bod u_{ij}^{m+1} nasledovný vzťah:

$$u_{ij}^{m+1} = u_{ij}^m \left(1 - \frac{4\sigma}{h^2}\right) + \frac{\sigma}{h^2} (u_{i-1j}^m + u_{i+1j}^m + u_{ij+1}^m + u_{ij-1}^m). \quad (4)$$



Obrázok 1.5: Detail povrchu grafu funkcie po postupnom aplikovaní rovnice vedena tepla

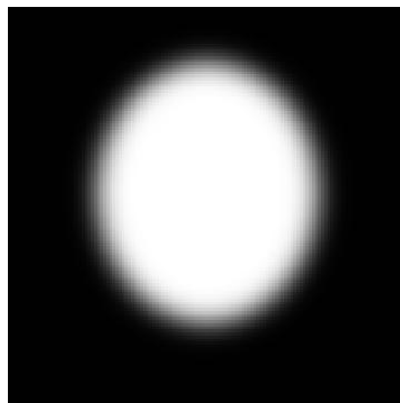
Na obr. 1.5 vidíme vplyv zhladenia na obrázok funkcie (1). Pre funkciu `ListPlot3D` sme použili parameter `ColorFunction->"CMYKColors"` aby bolo lepšie vidno porušenie pôvodného hladkého obrazu. Vidíme, že rovnica vedenia tepla nám pomáha docieľiť, požadovanú hladkosť vstupných dát.

Treba si však uvedomiť vplyv tohto modelu na dáta. Na jednej strane nám ich zhladzuje, ale na strane druhej zároveň deformuje a môže zhladiť aj hrany, alebo nerovnosti, ktoré sú dôležité a patria k pôvodným dátam. Preto treba vhodne voliť koeficient σ a zväčšiť koľkokrát aplikovať zhladenie na dané skalárne pole. Na základe charakteru vstupných medicínskych dát a našich experimentálnych skúseností nastavujeme v našej aplikácii ako preddefinovanú hodnotu $M = 2$.

2 Rýchlostné pole

V tejto časti sa zameriame na získanie rýchlostného poľa, ktoré bude hlavnou hybnou silou pohybu bodov krivky.

2.1 Konštrukcia vhodného rýchlostného poľa



Obrázok 2.1: Obrázok elipsy, ktorý budeme ďalej spracovávať

Ako bolo v predchádzajúcej kapitole ukázané, môžeme si vstupné dáta, dané skalárne pole, predstaviť ako graf funkcie dvoch premenných. Pri riešení nášho problému nám ide o nájdenie hrán objektov v obraze. V prvom rade hľadáme miesta, kde sa najviac a najrýchlejšie mení intenzita šedi. V trojrozmernej predstave, ide nám o nájdenie miest, kde najviac stúpa resp. klesá daná funkcia. To znamená, že pre každý bod obrázka hľadáme vektor gradientu.

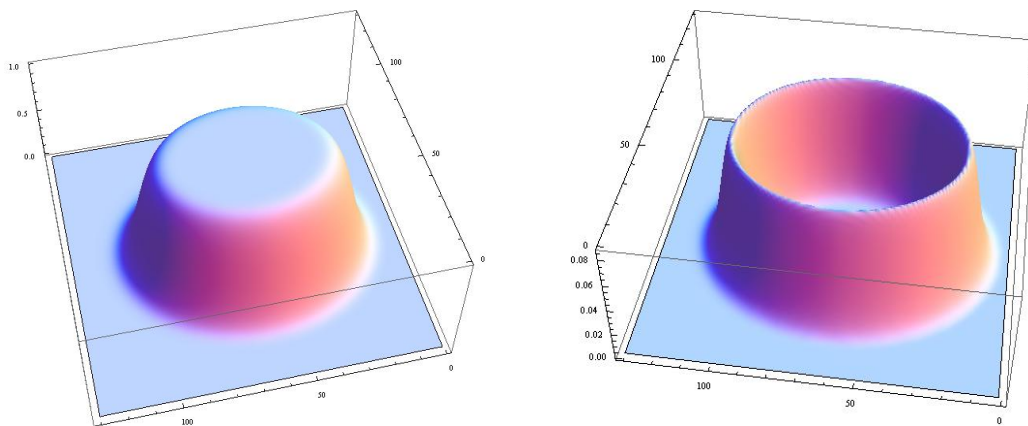
$$\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right)^T \quad (5)$$

Keďže vstupné dáta nám neposkytujú analytickú funkciu, ale len diskkrétne skalárne dáta, gradient počítame numerickou aproximáciou

$$\nabla u_{ij} = \left(\frac{u_{i+1j} - u_{i-1j}}{2h}, \frac{u_{ij+1} - u_{ij-1}}{2h} \right). \quad (6)$$

Keďže nás zaujímajú veľkosti gradientov v jednotlivých bodoch obrázka, počítame ich absolútne hodnoty vzťahom

$$|\nabla u_{ij}| = \sqrt{\frac{(u_{i+1j} - u_{i-1j})^2}{4h^2} + \frac{(u_{ij+1} - u_{ij-1})^2}{4h^2}} \quad (7)$$

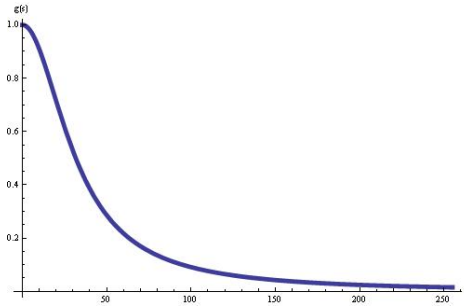


Obrázok 2.2: Trojrozmerné zobrazenie obrázka elipsy a jeho absolútnych hodnôt gradientov

Opísaný postup sme aplikovali na obrázok elipsy 2.1. Na obr. 2.2 vpravo vidíme, že získaná "funkcia" absolútnych hodnôt gradientu v bodoch obrázka nadobúda najväčšie hodnoty v miestach, kde je hrana objektu. Následne aplikujeme tzv. "škálovaciu" funkciu

$$g(s) = \frac{1}{1 + K s^2}, \quad (8)$$

na hodnoty obrázka, kde s je parameter predstavujúci absolútnu hodnotu gradientu v bode (x, y) , s koeficientom škálovania $k \in R, k > 0$, viď obr. 2.3 a 2.4. Na základe



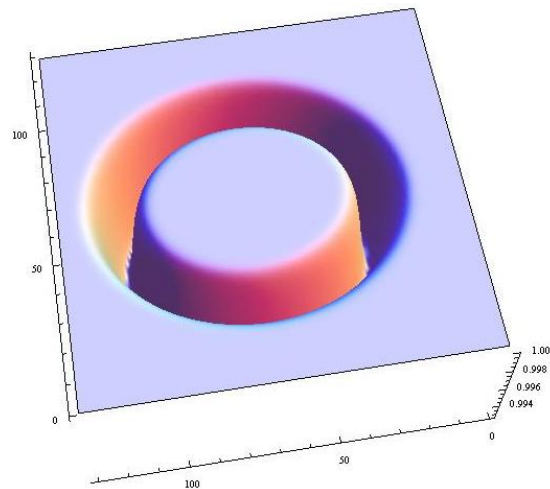
Obrázok 2.3: Graf funkcie (8)

dosahovaných výsledkov pre dané vstupné medicínske dáta v našej aplikácii ako preddefinovanú hodnotu volíme $K = 1000$.

Týmto spôsobom získame nové skalárne pole, pre ktoré opäť vypočítame gradienty v každom bode a tie vezmeme s opačným znamienkom, aby sme orientovali vektory ku hrane objektu. Takto získame rýchlostné pole

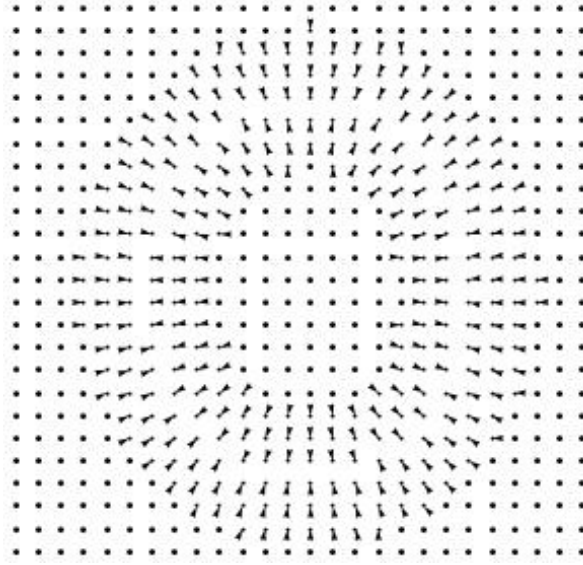
$$\mathbf{v}(x, y) = -\nabla g(|\nabla u(x, y)|), \quad (9)$$

ktoré je základom našej aplikácie a je najdôležitejšou informáciou, ktorú pri našom algoritme budeme využívať. Správnosť orientácie rýchlostného poľa si môžeme overiť i graficky.



Obrázok 2.4: Graf výsledných hodnôt po aplikácii funkcie (8)

Použijeme na to príkaz v systéme Mathematica `ListVectorFieldPlot`, ktorý vykreslí vektory $\mathbf{v}(x, y)$ ako šípky. Na obr. 2.5 vidíme, že jednotlivé vektory sú správne orientované a ak by sme do ich poľa vložili voľný bod, tak by mal byť nimi posúvaný až ku hrane objektu. Táto myšlienka je podstatou riešenia nášho problému.



Obrázok 2.5: Znázornenie vypočítaného vektorového poľa elipsy

2.2 Krivka v rýchlostnom poli

V tejto časti sa prvýkrát budeme venovať pohybu krivky. Počiatočnú krivku v našom prípade predstavuje úsečka. Tá bude reprezentovaná bodmi, ktoré ju rovnomerne delia. Ich vzdialenosť si nastavujeme v aplikácii. Užívateľ zadá prvý a posledný bod úsečky. Body medzi nimi sa dopočítajú na základe parametrickej rovnice priamky. Tieto body budeme označovať pomocou ich polohových vektorov \mathbf{r}_i^m . Takto označíme i -ty bod v m -tom časovom kroku pohybu. Samotný pohyb bude charakterizovaný vzťahom

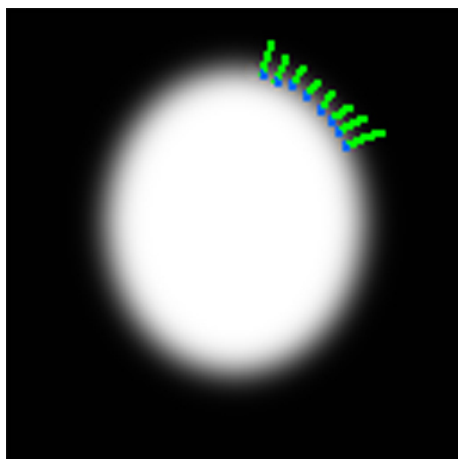
$$\partial_t \mathbf{r} = \mathbf{v} \quad (10)$$

Teda nová pozícia bodu sa bude počítat na základe vzťahu

$$\mathbf{r}_i^{m+1} = \mathbf{r}_i^m + \tau \mathbf{v} \quad (11)$$

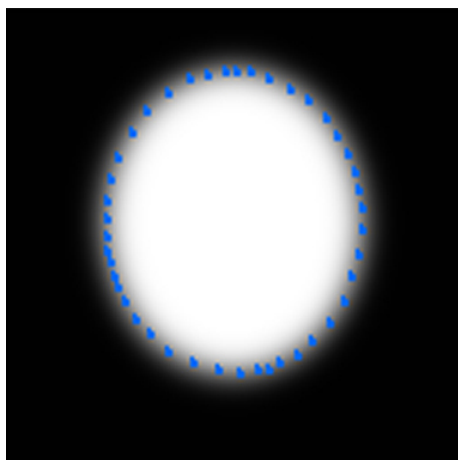
kde nová pozícia je vypočítaná ako súčet predchádzajúcej a posunutia spôsobeného rýchlostným poľom vypočítaným v pixeli, v ktorom sa nachádza bod \mathbf{r}_i^m , pričom τ je zvolený časový krok. Je vhodné ho voliť tak, aby sa pri jednom časovom kroku mohol bod krivky pohnúť maximálne o 1 pixel. V našej aplikácii je tento koeficient automaticky nastavovaný na základe vypočítaného rýchlostného poľa tak, aby spĺňal uvedenú podmienku. Napríklad pre obrázok elipsy je to hodnota $\tau = 5.99$. Dôležitý je i počet časových krokov, počas ktorých sa krivka hýbe. Voľbe zastavovacích podmienok pre tento pohyb sa venujeme v kapitole 3.3.

Implementácia tohto algoritmu do našej aplikácie bola pomerne jednoduchá a ponúkané riešenia boli nad očakávania potešujúce. Na obr. 2.6 vidíme funkčnosť aplikácie, s načítaným obrázkom elipsy. Je tu znázornená trajektória pohybu jednotlivých bodov smerom ku hrane elipsy. Ako vidíme, počiatočnú krivku tvorili body úsečky mimo elipsy, ktoré sa postupne posúvali smerom ku hrane objektu.



Obrázok 2.6: Ukážka aplikácie, znázornenie pohybu bodov počiatkovej úsečky ku hrane elipsy

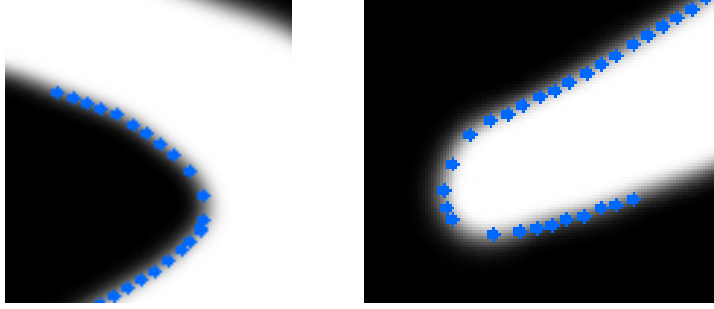
Ako vidíme na obr. 2.7, už v tejto fáze dokážeme pomocou našej aplikácie nájsť hrany jednoduchých objektov v obrázku. Toto však neplatí pre zložitejšie tvary. Tiež si môžeme všimnúť, že vzdialenosti susedných bodov nie sú rovnomerné. Určité riešenie tohto problému si ukážeme v nasledujúcej kapitole.



Obrázok 2.7: Hrana nájdená s využitím rýchlostného poľa (9)

3 Modifikácie rýchlostného poľa

Ako bolo vyššie spomenuté doposiaľ používaný algoritmus nemusí fungovať pre zložitejšie tvary. Takisto delenie výsledných kriviek je nerovnomerné. Pri jednoduchom tvare akým je elipsa sa nám tieto veci nejavia ako veľký problém. Ten sa však ukáže pri použití aplikácie na dáta, pre ktoré je určená. Vytvorili sme si pre tento účel obrázok, na ktorom bude dobre vidno spomínaný problém.

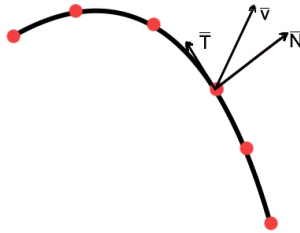


Obrázok 3.1: Nedostatky pohybu len pomocou rýchlostného poľa (9)

Jasne vidíme, že body sa k sebe približujú a od seba vzdalujú nerovnomerne. Je to pochopiteľné, keďže v aplikácii vystupujú ako jednotlivci v rýchlostnom poli, ktoré ich môže nasmerovať aj na rovnaké miesta. Tomuto sa však pokúsime zabrániť. Budeme sa snažiť nejakým spôsobom body zviazať a ich pohyb obmedziť v závislosti od susedných bodov krivky

3.1 Zanedbanie tangenciálnej zložky rýchlosti

Prvým prístupom k riešeniu uvedenej problematiky je rozklad vektora rýchlosti príslušajúceho danému bodu krivky na normálový a dotykový vektor ku krivke v danom bode. Ako vidíme z obr. 3.2, pri dostatočne jemnom delení krivky, nám tangenciálna zložka vektora rýchlosti krivku nemení, pohybuje len bodmi smermi k sebe a od seba. Ak ju zanedbáme, odstránime tento pohyb a pre rovnomerne rozdelenú počiatočnú úsečku zabezpečíme rovnomernejšiu distribúciu bodov počas pohybu.



Obrázok 3.2: Rozklad vektora rýchlosti

Pre modifikovaný pohyb pri ktorom zanedbáme tangenciálne zložky vektora rýchlosti dostávame vzťah

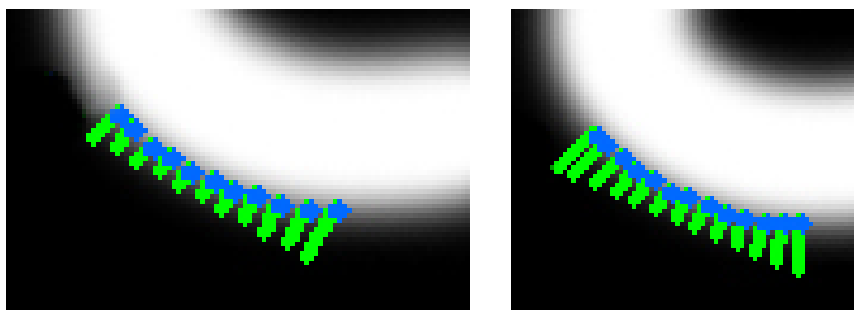
$$\partial_t \mathbf{r} = \mu v_{\mathbf{N}} \mathbf{N}, \quad (12)$$

kde μ je parameter a skalár $v_{\mathbf{N}}$ vypočítame ako projekciu vektora rýchlosti na smer normály ku krivke, čiže pôjde o nasledovný skalárny súčin

$$v_{\mathbf{N}} = \mathbf{v} \cdot \mathbf{N}. \quad (13)$$

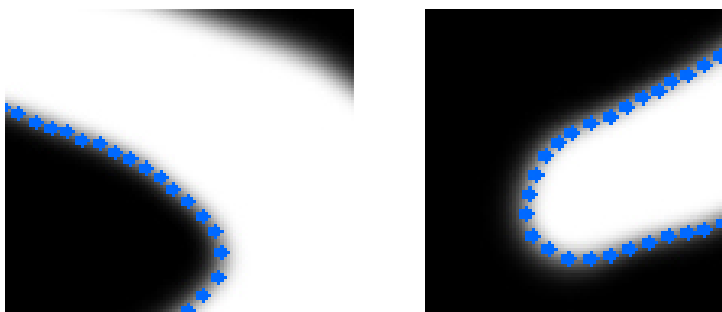
Normálové vektory \mathbf{N}_i pre jednotlivé body, kde \mathbf{N}_0 je normálový vektor prislúchajúci prvému bodu a \mathbf{N}_n poslednému bodu krivky, vypočítame nasledovne

$$\begin{aligned} \mathbf{N}_0 &= \left(\frac{\mathbf{r}_1 - \mathbf{r}_0}{|\mathbf{r}_1 - \mathbf{r}_0|} \right)^\perp \\ \mathbf{N}_n &= \left(\frac{\mathbf{r}_n - \mathbf{r}_{n-1}}{|\mathbf{r}_n - \mathbf{r}_{n-1}|} \right)^\perp \\ \mathbf{N}_i &= \left(\frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{|\mathbf{r}_{i+1} - \mathbf{r}_{i-1}|} \right)^\perp \quad \text{kde } i = 1, 2, \dots, n-1. \end{aligned} \tag{14}$$



Obrázok 3.3: Porovnanie pohybu bodov

To že táto projekcia funguje ako sme uviedli, znázorňuje obrázok 3.3. Vidíme na ňom porovnanie trajektórií pohybov bodov krivky. Vpravo sú to trajektórie pohybu spôsobené len rýchlostným poľom (pôvodný prístup) a vľavo so zanedbanými zložkami rýchlostí tangenciálnymi ku vyvíjajúcej sa krivke. Na prípade vľavo je pekne vidno, že ide o projekciu bodov pôvodnej úsečky v smere normálového vektora. Vidíme, že tieto trajektórie sú navzájom takmer rovnobežné. Dostávame takto rovnomernejšie delenie výslednú krivku, avšak pre zložitejšie tvary táto projekcia stačiť nemusí. Bude nutné ešte iným spôsobom zviazať susedné body, aby sa od seba nevzdalovali, resp. sa k sebe nepribližovali. Pomocou tohto prístupu sme dosiahli v našej aplikácii nasledujúce výsledky pre nami vytvorený obrázok, ktorý môžeme porovnať s obr. 3.1.



Obrázok 3.4: Hrany nájdené so zanedbanou tangenciálnou zložkou rýchlosti

3.2 Regularizácia pomocou krivosti

V predchádzajúcom postupe sme pri výpočte pohybu krivky upravovali vektor rýchlosti tak, že sme zanedbali jeho zložku tangenciálnu ku krivke. Teraz tento prístup vylepšíme. Do modelu zahrnieme vplyv krivosti modelovanej krivky.

Nami uvažovaný model pohybu krivky môžeme vyjadriť vo všeobecnom tvare

$$\partial_t \mathbf{r} = \beta \mathbf{N} + \alpha \mathbf{T}. \quad (15)$$

Člen α násobený jednotkovou tangentou \mathbf{T} považujeme v tejto časti za nulový, teda nebudeme uvažovať pohyb v smere tangenty (tomuto problému sa venujeme v kapitole 3.4). Člen β vo vzťahu (15) predstavuje normálovú zložku vektora rýchlosti doplnenú aj o vplyv krivosti krivky, t.j. uvažujeme

$$\beta = \mu v_{\mathbf{N}} + \epsilon k. \quad (16)$$

Môžeme teda napísať

$$\partial_t \mathbf{r} = \mu v_{\mathbf{N}} \mathbf{N} + \epsilon k \mathbf{N}. \quad (17)$$

Projekciu rýchlosti vektorového poľa do smeru normály $v_{\mathbf{N}}$ počítame pomocou už spomínaného vzťahu (13). Člen $k\mathbf{N}$ predstavuje vektor krivosti. Ako je uvedené napr. v [1], z Frenetových vzorcov pre vektor krivosti platí $k\mathbf{N} = \partial_s \mathbf{T} = \partial_{ss} \mathbf{r}$, pričom pozičný vektor \mathbf{r} je funkciou jednotkovej dĺžkovej parametrizácie s krivky a času t . Takto získavame nelineárnu parciálnu diferenciálnu rovnicu pre advekciu a difúziu krivky

$$\partial_t \mathbf{r} = \mu v_{\mathbf{N}} \mathbf{N} + \epsilon \partial_{ss} \mathbf{r}. \quad (18)$$

Ak použijeme na aproximáciu vektora krivosti metódu konečných prvkov a diagonalizáciu matice hmotnosti [3] alebo metódu plávajúcich konečných objemov [9], dostávame vzťah (viď napr. [1])

$$k_i^m \mathbf{N}_i^m \approx \frac{2}{h_{i+1}^m + h_i^m} \left(\frac{\mathbf{r}_{i+1}^m - \mathbf{r}_i^m}{h_{i+1}^m} - \frac{\mathbf{r}_i^m - \mathbf{r}_{i-1}^m}{h_i^m} \right) \quad i = 1, 2, \dots, n-1. \quad (19)$$

Konštanta n predstavuje počet bodov počítanej krivky. Vzdialenosti uzlov h_i v časovom okamihu m budeme počítat vzťahom

$$h_i^m = \sqrt{(x_i^m - x_{i-1}^m)^2 + (y_i^m - y_{i-1}^m)^2}, \quad \text{kde } \mathbf{r}_i^m = (x_i^m, y_i^m). \quad (20)$$

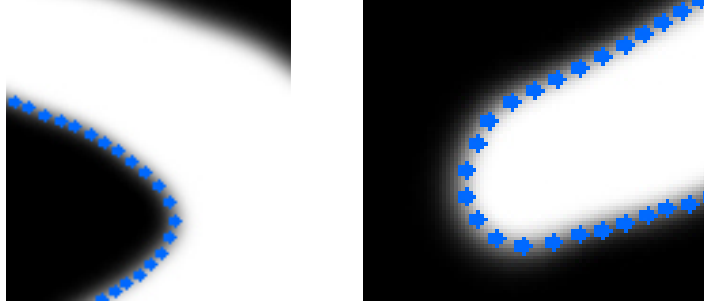
Vzťah (17) môžeme teda aproximovať v tvare

$$\frac{\mathbf{r}_i^{m+1} - \mathbf{r}_i^m}{\tau} = \mu (v_{\mathbf{N}})_i^m \mathbf{N}_i^m + \epsilon \frac{2}{h_{i+1}^m + h_i^m} \left(\frac{\mathbf{r}_{i+1}^m - \mathbf{r}_i^m}{h_{i+1}^m} - \frac{\mathbf{r}_i^m - \mathbf{r}_{i-1}^m}{h_i^m} \right). \quad (21)$$

Krivosť prispieva k citlivosti modelu na vzájomnú vzdialenosť susedných bodov, "zväzuje" body krivky. Ak by niektorý bod krivky bol v pôvodnom modeli sám mimo nájdenej hrany a takto by vytváral ostrý špic, krivostný člen spôsobí, že susedné body budú tento ťahať naspäť ku nájdenej hrane. Po úprave vzťahu (21) dostávame nasledovný výsledný model pre pozíciu bodu v novom čase

$$\mathbf{r}_i^{m+1} = \mathbf{r}_i^m + \tau\mu (v_{\mathbf{N}})_i^m \mathbf{N}_i^m + \tau\epsilon \frac{2}{h_{i+1}^m + h_i^m} \left(\frac{\mathbf{r}_{i+1}^m - \mathbf{r}_i^m}{h_{i+1}^m} - \frac{\mathbf{r}_i^m - \mathbf{r}_{i-1}^m}{h_i^m} \right) \quad (22)$$

Navrhnutý model sme implementovali do našej aplikácie, parametre μ , τ a ϵ volíme tak, aby sa žiaden bod krivky v jednom časovom kroku neposunul o viac ako 1 pixel a bola zaručená stabilita numerickej scémy. Výsledky, ktoré sme týmto modelom získali vidíme na obrázku 3.5.



Obrázok 3.5: Hrany nájdené modelom zahŕňajúcim krivosť

3.3 Zastavovacie podmienky

Ako sme uviedli v kapitole 2.2, pri pohybe krivky je dôležité správne zvoliť i počet jeho časových krokov. Ich počet musí byť dostatočný na to, aby sa príslušné body dokázali posunúť až po hranice objektu. Na druhej strane pri zbytočne veľkom počte časových krokov sa neefektívne plytvá časom výpočtu. Pre tento účel sme navrhli dva typy zastavovacích podmienok. Prvý sleduje zmeny celkovej dĺžky pohybujúcej sa krivky. Táto zmena súvisí aj s celkovou dĺžkou krivky, preto sme navrhli zastavovaciu podmienku ako pomer absolútnej hodnoty rozdielu dĺžky krivky v danom a predchádzajúcom časovom kroku a ich priemernej dĺžky

$$d^m = \frac{2|L^m - L^{m-1}|}{L^m + L^{m-1}}, \quad (23)$$

pričom podmienku pre zastavenie pohybu krivky sme po analýze dosahovaných výsledkov zvolili $d^m < 0.0001$.

Druhý návrh zastavovacej podmienky je zameraný na smer pohybu jednotlivých bodov krivky. Z predchádzajúcich častí je zrejmé, ako sa body hýbu. Rýchlostné pole ich buď posunie na hranu objektu, kde sa zastavia a už nehýbu, resp. ich pohyb je už zanedbateľný, alebo sa priblížia k hrane, prekročia ju a následne ich rýchlostné pole nasmeruje naspäť. Teda ich pohyb sa zmení na určité kmitanie okolo hrany. Pri každom obrátení pohybu v smere normály ku krivke sa zmení znamienko skaláru $v_{\mathbf{N}}$ na opačné. Táto zmena teda signalizuje, že daný bod už narazil na hranu objektu. Pre všetky body krivky budeme sledovať zmenu tohto znamienka, a zároveň testovať či hodnota skaláru

$v_{\mathbf{N}}$ nie je v absolútnej hodnote zanedbateľne malé číslo. Na základe analýzy dosiahnutých výsledkov sme zvolili zastavovaciu podmienku

$$|(v_{\mathbf{N}})_i^m| < 0.01 \quad \text{alebo} \quad (v_{\mathbf{N}})_i^m (v_{\mathbf{N}})_i^{m-1} < 0, \quad (24)$$

ktorá musí byť splnená pre všetky body krivky $i = 1, \dots, n$. Po porovnaní výsledkov sa ukázala ako lepšia druhá navrhnutá zastavovacia podmienka (24). Tú sme ešte doplnili o maximálnu hodnotu počtu časových krokov 200. Tá sa však uplatňuje len výnimočne, pretože pre nájdenie hrany stačí väčšinou oveľa menej časových krokov.

3.4 Asymptoticky rovnomerná redistribúcia

V predchádzajúcich častiach práce sme vytvárali krivku čo najlepšie vystihujúcu hľadanú hranu objektu. Ukázali sme, že modifikáciou pohybu krivky môžeme dostať rovnomernejšiu redistribúciu jej bodov, ako pri prvotnom modeli. V modifikovaných modeloch sme zanebávali zložku rýchlosti tangenciálnu ku krivke. V tejto časti ju navrhujeme podľa [11]. Model, ktorý vytvoríme bude v našej aplikácii slúžiť ako postprocessing krivky podľa voľby užívateľa. Predpokladajme, že máme vytvorenú uzavretú krivku použitím modelu (17) s využitím regularizácie pomocou krivosti. Krivku si označme Γ . Tá je parametrizovaná funkciou pozičného vektora. Majme veličinu určujúcu lokálnu dĺžku g definovanú na krivke, $g = \left| \frac{\partial \mathbf{r}}{\partial u} \right| \approx \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{h}$, pričom $h = \frac{1}{n}$ predstavuje rovnomerné delenie S^1 . Keďže predpokladáme uzavretú krivku, môžeme po diskretizácii v priestore a čase jednoducho nadefinovať periodické okrajové podmienky pridaním bodov $\mathbf{r}_0^m = \mathbf{r}_n^m, \mathbf{r}_{n+1}^m = \mathbf{r}_1^m$. Ďalej si označme vzdialenosti susedných bodov ako

$$h_i^m = |R_i|, \quad \text{kde } R_i = (R_{i1}, R_{i2})^T = \mathbf{r}_i^{m-1} - \mathbf{r}_{i-1}^{m-1}, \quad i = 1, \dots, n+1, \quad R_0 = R_n. \quad (25)$$

Pre časový vývoj lokálnej dĺžky a globálnej dĺžky krivky podľa [9] platí

$$\partial_t g = -gk\beta + g\partial_s \alpha \quad (26)$$

$$\frac{dL}{dt} = - \langle k\beta \rangle_{\Gamma} L, \quad (27)$$

kde L je celková dĺžka krivky a strednú hodnotu $\langle k\beta \rangle_{\Gamma} = \frac{1}{L} \int_{\Gamma} k\beta ds$ budeme pre časový krok m aproximovať vzťahom

$$\langle k\beta \rangle_{\Gamma} \approx \frac{1}{L^m} \sum_{l=1}^n h_l^m k_l^m \beta_l^m. \quad (28)$$

Krivosť k_i^m a normálovú zložku rýchlosti β_i^m budeme počítat pomocou vzťahov (viď [11])

$$k_i^m = \text{sgn}(R_{i-1} \wedge R_{i+1}) \frac{1}{2h_i^m} \arccos \left(\frac{R_{i+1} \cdot R_{i-1}}{h_{i+1}^m h_{i-1}^m} \right), \quad i = 1, \dots, n \quad (29)$$

$$\beta_i^m = \epsilon k_i^m + \frac{\mu(v_{\mathbf{N}})_i^{m-1} + \mu(v_{\mathbf{N}})_{i-1}^{m-1}}{2}, \quad i = 1, \dots, n. \quad (30)$$

Pre odvodenie asymptoticky rovnomernej redistribúcie bodov krivky je zaujímavý podiel

$$\frac{g}{L} \approx \frac{|\mathbf{r}_i - \mathbf{r}_{i-1}|}{Lh} = \frac{|\mathbf{r}_i - \mathbf{r}_{i-1}|}{\left(\frac{L}{n}\right)} = \frac{h_i}{\left(\frac{L}{n}\right)}, \quad (31)$$

ktorý je vlastne pomerom lokálnej dĺžky rozdelenej úsečky a priemernej dĺžky delenia. Pri rovnomernej redistribúcii chceme dosiahnuť, aby sa tento podiel rovnal 1. Označme

$$\theta = \ln \left(\frac{g}{L} \right), \quad (32)$$

a potom bude naším cieľom aby sa asymptoticky $\theta = 0$. Pretože platí

$$\partial_t \theta = \partial_t \left[\ln \left(\frac{g}{L} \right) \right] = \frac{L \partial_t g L - g \partial_t L}{L^2}, \quad (33)$$

po dosadení (26) a (27) do (33) dostávame

$$\partial_t \theta = \partial_s \alpha - k\beta + \langle k\beta \rangle_\Gamma. \quad (34)$$

Ako sme uviedli, chceme, aby $\theta \rightarrow 0$ pre $t \rightarrow \infty$. To sa dá dosiahnuť mnohými spôsobmi. Napríklad ako je uvedené v [11] tak, že θ spĺňa diferenciálnu rovnicu $\partial_t \theta = (e^{-\theta} - 1)\omega$, kde ω predstavuje rýchlosť priebehu redistribúcie. Pre naše θ dané vzťahom (32) sa pravá strana zjednoduší a dostaneme $e^{-\ln(\frac{g}{L})} - 1 = e^{\ln(\frac{L}{g})} - 1 = \frac{L}{g} - 1$ a teda z (34) dostaneme vzťah pre tangenciálnu rýchlosť α , ktorou dosiahneme asymptoticky rovnomernú redistribúciu bodov krivky.

$$\partial_s \alpha = k\beta - \langle k\beta \rangle_\Gamma + \left(\frac{L}{g} - 1 \right) \omega. \quad (35)$$

Keďže platí $\mathbf{T} = \partial_s \mathbf{r}$ a ako už bolo uvedené $k\mathbf{N} = \partial_s \mathbf{T} = \partial_{ss} \mathbf{r}$, môžeme napísať vzťah pre výsledný model aj s tangenciálnou zložkou rýchlosti ako

$$\partial_t \mathbf{r} = \mu v_{\mathbf{N}} \mathbf{N} + \epsilon \partial_{ss} \mathbf{r} + \alpha \partial_s \mathbf{r}. \quad (36)$$

Na aproximáciu rovníc (35) a (36) použijeme tzv. metódu plávajúcich konečných objemov [9, 10, 11]. Integráciou pozdĺž v čase sa meniaceho úseku krivky $[\mathbf{r}_{i-1}, \mathbf{r}_i]$ dostaneme z rovnice (35) vzťah

$$\int_{\mathbf{r}_{i-1}}^{\mathbf{r}_i} \partial_s \alpha ds = \int_{\mathbf{r}_{i-1}}^{\mathbf{r}_i} k\beta - \langle k\beta \rangle_\Gamma + \left(\frac{L}{g} - 1 \right) \omega ds, \quad (37)$$

odkiaľ

$$\alpha_i = \alpha_{i-1} + h_i k_i \beta_i - h_i \langle k\beta \rangle_\Gamma + \left(\frac{L}{n} - h_i \right) \omega, \quad i = 1, \dots, n, \quad \alpha_0 = 0, \quad (38)$$

čo predstavuje výslednú tangenciálnu zložku rýchlosti v jednotlivých diskretných bodoch krivky. Na získanie numerickej aproximácie rovnice (36) ju zintegrujeme cez duálny objem $[\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}_{i+1}]$

$$\int_{\tilde{\mathbf{r}}_i}^{\tilde{\mathbf{r}}_{i+1}} \partial_t \mathbf{r} ds = \int_{\tilde{\mathbf{r}}_i}^{\tilde{\mathbf{r}}_{i+1}} \mu v_{\mathbf{N}} \mathbf{N} + \epsilon \partial_{ss} \mathbf{r} + \alpha \partial_s \mathbf{r} ds. \quad (39)$$

kde $\tilde{\mathbf{r}}_i = (\mathbf{r}_{i-1} + \mathbf{r}_i)/2$. Dostávame tak

$$q_i \frac{d\mathbf{r}_i}{dt} = \mu (v_{\mathbf{N}})_i \mathbf{N}_i q_i + \epsilon [\partial_s \mathbf{r}]_{\tilde{\mathbf{r}}_i}^{\tilde{\mathbf{r}}_{i+1}} + \alpha_i (\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i), \quad (40)$$

kde $q_i = \frac{1}{2}(h_i + h_{i+1})$. Získané členy na pravej strane vzťahu (40) môžu byť prirodzene aproximované ako

$$\epsilon \partial_s \tilde{\mathbf{r}}_{i+1} - \epsilon \partial_s \tilde{\mathbf{r}}_i \approx \epsilon \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{h_{i+1}} - \epsilon \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{h_i} \quad (41)$$

$$\alpha_i (\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i) \approx \alpha_i \frac{\mathbf{r}_{i+1} + \mathbf{r}_i}{2} - \frac{\mathbf{r}_i + \mathbf{r}_{i-1}}{2} = \alpha_i \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2} \quad (42)$$

Pokiaľ aproximujeme časovú deriváciu spätnou diferenciou

$$q_i \frac{d\mathbf{r}_i}{dt} \approx q_i^m \frac{\mathbf{r}_i^m - \mathbf{r}_i^{m-1}}{\tau} \quad (43)$$

a použijeme semi-implicitnú aproximáciu členov v (38), (41) a (42) dostaneme rovnice

$$\alpha_i^m = \alpha_{i-1}^m + h_i^m k_i^m \beta_i^m - \frac{h_i^m}{L^m} \sum_{l=1}^n h_l^m k_l^m \beta_l^m + \omega \left(\frac{L^m}{n} - h_i^m \right), \alpha_0 = 0 \quad (44)$$

$$\frac{q_i^m}{\tau} \mathbf{r}_i^m + \frac{\mathbf{r}_i^m - \mathbf{r}_{i+1}^m}{h_{i+1}^m} + \frac{\mathbf{r}_i^m - \mathbf{r}_{i-1}^m}{h_i^m} + \frac{\alpha_i^m}{2} (\mathbf{r}_{i-1}^m - \mathbf{r}_{i+1}^m) = \frac{q_i^m}{\tau} \mathbf{r}_i^{m-1} + \mu (v_{\mathbf{N}})_i^m \mathbf{N}_i^m, \quad (45)$$

$i = 1, \dots, n, L^m = \sum_{l=1}^n h_l^m$. Tieto rovnice reprezentujú trojdiagonálny cyklický systém, ktorý môžeme zapísať v tvare

$$\mathcal{A}_i^m \mathbf{r}_{i-1}^m + \mathcal{B}_i^m \mathbf{r}_i^m + \mathcal{C}_i^m \mathbf{r}_{i+1}^m = \mathcal{F}_i^m \quad (46)$$

$$\mathcal{A}_i^m = -\frac{\epsilon}{h_i^m} + \frac{\alpha_i^m}{2} \quad (47)$$

$$\mathcal{C}_i^m = -\frac{\epsilon}{h_{i+1}^m} - \frac{\alpha_i^m}{2} \quad (48)$$

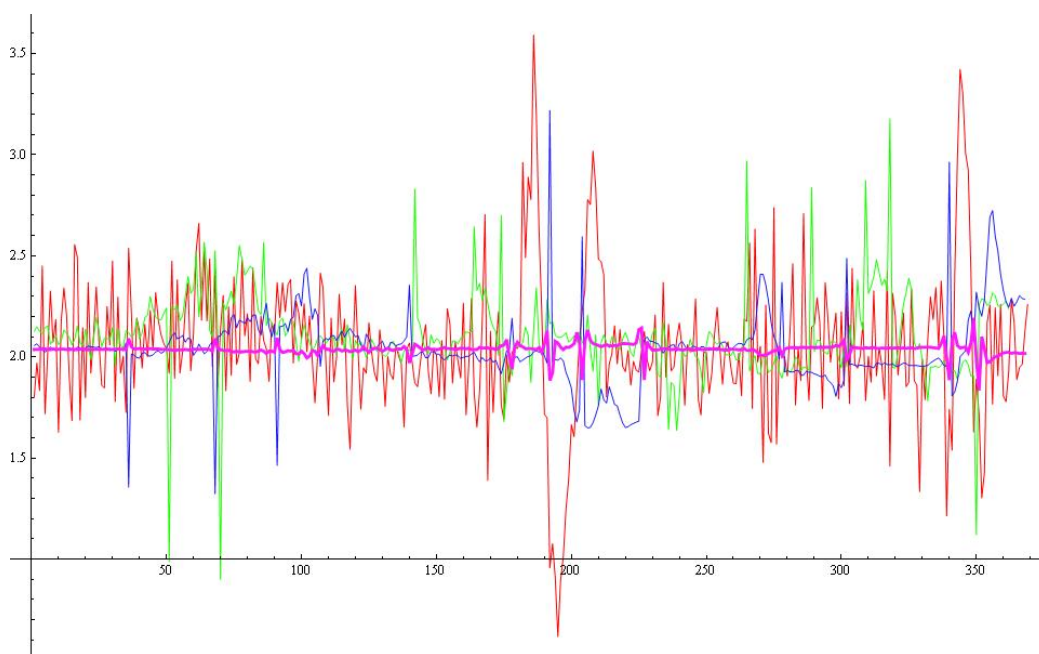
$$\mathcal{B}_i^m = \frac{q_i^m}{\tau} - \mathcal{A}_i^m - \mathcal{C}_i^m \quad (49)$$

$$\mathcal{F}_i^m = \frac{q_i^m}{\tau} \mathbf{r}_i^{m-1} + \mu (v_{\mathbf{N}})_i^m \mathbf{N}_i^m q_i^m \quad (50)$$

Dostali sme tak na ľavej strane trojdiagonálnu cyklickú maticu. V praktických výpočtoch volíme $\mu = 3$, $\epsilon = 0.05$, $\omega = 5$ a parameter τ volíme automaticky tak, aby výsledná matica bola diagonálne dominantná. Na riešenie systému lineárnych rovníc v takom prípade môžeme použiť Sherman-Morrisonovu folmulu a Thomasov algoritmus [12]. Uvedený model aplikujeme vo viacerých časových krokoch na uzavretú krivku. Dostávame tak stále rovnomernejšie delenie bodov na nej. Počet krokov volíme opäť automaticky, a to tak, že sledujeme pomer medzi najväčšou a najmenšou vzdialenosťou susedných bodov krivky a výpočet zastavíme ak je tento pomer blízky jednotke. Uvedený model sme implementovali do našej aplikácie, výsledky, ktoré sme ním dosiahli sú prezentované v nasledujúcej kapitole na obrázku 4.1 a v tabuľke 4.1.

4 Porovnanie dosiahnutých výsledkov

Ak porovnáваме dosiahnuté výsledky, už voľným okom vidíme, že pôvodný model pri ktorom pohyb krivky spôsobuje len rýchlostné, pole dáva ako výsledok najmenej rovnomerné delenie výslednej krivky. Iba na základe vizuálnej informácie vieme však len veľmi ťažko porovnať ostatné modely. Preto sme v aplikácii vytvorili funkciu, ktorá ako výstup dáva textový súbor so vzdialenosťami bodov vytvorenej krivky. Tie sme potom vykreslili v systéme Mathematica pomocou príkazu ListPlot. Dosiahnuté výsledky sme mohli lepšie porovnať. Uvádzame výsledky z tohto porovnania pre vstupné dáta nášho skúšobného obrázka. Implicitnú vzdialenosť bodov pôvodnej úsečky sme nastavili na 2 pixely pre všetky štyri metódy.



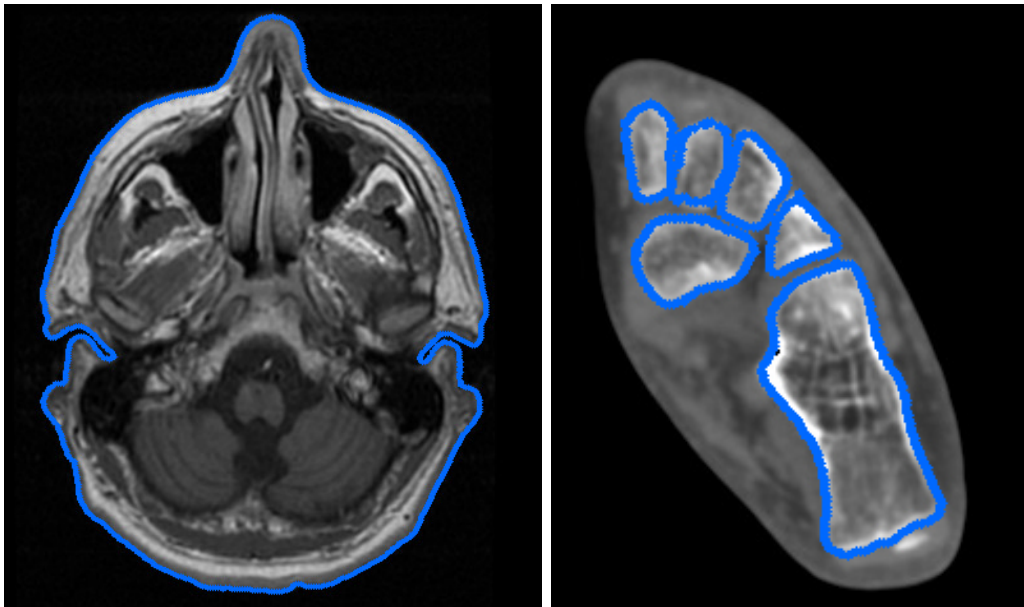
Obrázok 4.1: Grafické porovnanie rovnomernosti rozdelenia krivky dosiahnutej navrhnutými modelmi

Červenou farbou sú znázornené hodnoty dosiahnuté prvým modelom (10), zelenou druhým (12), modrou tretím (17) a fialovou štvrtým modelom (36). Ako vidíme, najlepšie rozdelenie nám poskytuje štvrtý model s tangenciálnou rýchlosťou α danou (35). Napriek výkyvom dosahujeme slušné výsledky i pri ostatných modeloch. Dosiahnuté výsledky reprezentuje aj výsledná tabuľka. Uvádzame v nej minimálne a maximálne hodnoty vzdialeností susedných bodov a takisto vypočítaný rozptyl týchto vzdialeností.

Na záver prezentujeme praktické výsledky pre medicínske dáta.

	Min	Max	Variance
model1	0.616815	3.59267	0.122167
model2	0.899491	3.17837	0.0420902
model3	1.3232	3.21941	0.0333361
model4	1.83412	2.19442	0.0009181

Tabuľka 1: Porovnanie modelov



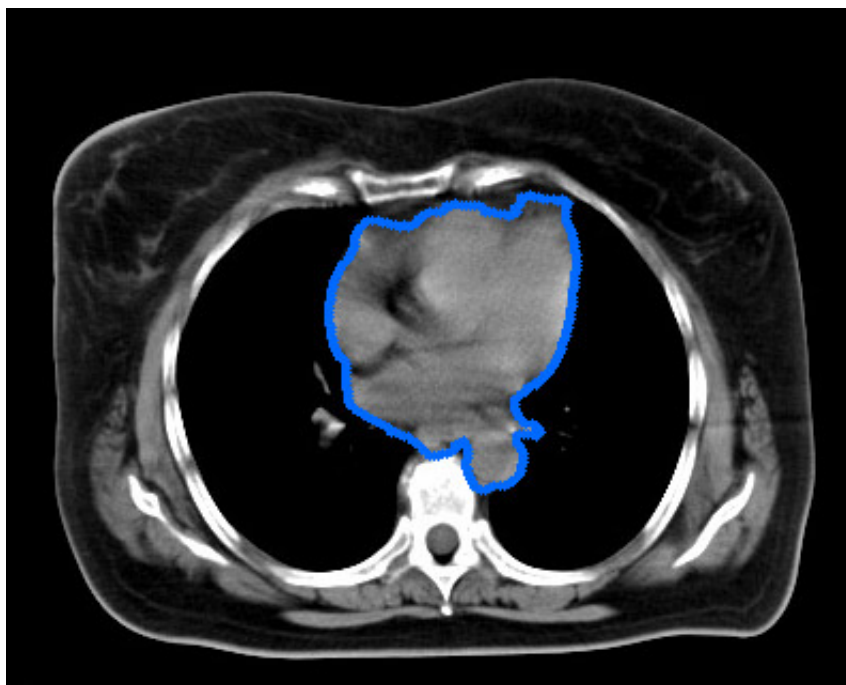
Obrázok 4.2: Praktické výsledky na medicínskych dátach (CT hlavy a chodidla)

5 Záver

Na základe dosiahnutých výsledkov, uvedených v predchádzajúcich kapitolách je zrejme, že sme dosiahli vytýčený cieľ. Vytvorili sme vhodný matematický model a počítačový program, použiteľný pre semi-automatické vyhľadávanie hraníc objektov v medicínskych dátach. Takisto sa nám podarilo docieľiť rôznymi metódami, aby distribúcia bodov krivky bola čo najrovnomernejšia. Vyvinutý nástroj pracuje rýchlo a jeho použitie je jednoduché. Výsledná hranica hľadaného objektu je reprezentovaná bodmi, ležiacimi na výslednej krivke.

PodĎakovanie

Táto práca bola podporovaná firmou Tatramed spol. s r.o. a projektami APVV-0351-07, APVV-LPP-0020-07, VEGA 1/0269/09.



Obrázok 4.3: Praktické výsledky na medicínskych dátach (CT hrudníka)

Referencie

- [1] M. Balažovjeh, K. Mikula, *A higher order scheme for the curve shortening flow of plane curves*, ALGORITMY 2009, Conference on Scientific Computing, Vysoké Tatry- Podbanske, Slovakia, March 15-20, 2009, pp. 165-176
- [2] V. Caselles, R. Kimmel, G. Sapiro, *Geodesic active contours*, International Journal of Computer Vision, 22 (1997), pp. 61–79.
- [3] G. Dziuk, *Convergence of a semi discrete scheme for the curve shortening flow*, Mathematical Models and Methods in Applied Sciences, 4 (1994), pp. 589–606.
- [4] M. Gage, R. S. Hamilton, *The heat equation shrinking convex plane curves*, J. Diff. Geom., 23 (1986), pp. 69–96.
- [5] T.Y. Hou, J. Lowengrub, M. Shelley, *Removing the stiffness from interfacial flows and surface tension*, J. Comput. Phys., 114 (1994), pp. 312–338.
- [6] M. Kass, A. Witkin, D. Terzopulos, *Snakes: active contour models*, International Journal of Computer Vision, 1 (1987), pp. 321–331.
- [7] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A. Yezzi, *Conformal curvature flows: from phase transitions to active vision*, Arch. Rational Mech. Anal., 134 (1996), pp. 275–301.
- [8] M. Kimura, *Numerical analysis for moving boundary problems using the boundary tracking method*, Japan J. Indust. Appl. Math., 14 (1997), pp. 373–398.

- [9] K. Mikula, D. Ševčovič, *Evolution of plane curves driven by a nonlinear function of curvature and anisotropy*, SIAM J. Appl. Math., 61 (2001), pp. 1473–1501.
- [10] K. Mikula, D. Ševčovič, *A direct method for solving an anisotropic mean curvature flow of planar curve with an external force*, Mathematical Methods in Applied Sciences, 27 (2004) pp. 1545-1565.
- [11] K. Mikula, D. Ševčovič, M. Balažovjeh *A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations*, akceptované do časopisu Communications in Computational Physics (2009)
- [12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flamery *Numeric Recipes - The Art of Scientific Computing (Third edition)*, Cambridge University Press, 2007, 1235 s., ISBN-13 978-0-511-33555-6.

Autori (Authors):

Bc. Jozef Urbán

Katedra matematiky a deskriptívnej geometrie, Stavebná fakulta STU

Radlinského 11,

813 68 Bratislava.

e-mail: jozo.urban@gmail.com

Prof. RNDr. Karol Mikula DrSc.

Katedra matematiky a deskriptívnej geometrie, Stavebná fakulta STU

Radlinského 11,

813 68 Bratislava.

e-mail: mikula@math.sk