SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE Stavebná fakulta

HĽADANIE IDEÁLNEJ CESTY PRE KAMERU VIRTUÁLNEJ KOLONOSKOPIE

Diplomová práca

SvF-5343-38875

Študijný program:Matematicko-počítačové modelovaniePracovisko:Katedra matematiky a deskriptívnej geometrieVedúci diplomovej práce:prof. RNDr. Karol Mikula DrSc.

Bratislava 2011

Bc. Jozef Urbán

Čestné prehlásenie

Vyhlasujem, že som túto prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Bratislava 16. 5. 2011

Vlastnomučný nodnie

Vlastnoručný podpis

Poďakovanie

Touto cestou by som sa chcel poďakovať prof. RNDr. Karolovi Mikulovi DrSc. za cenné pripomienky a odborné konzultácie a firme Tatramed spol. s r.o. za zastrešenie počas vypracovania tejto práce.

Abstrakt

Práca sa zaoberá problematikou hľadania cesty objektami v 3D obrazových dátach. Je vypracovaná v spolupráci s firmou TatraMed spol. s r.o. so sídlom v Bratislave. Jej výsledkom má byť vytvorenie vhodného matematického modelu a počítačového programu na hľadanie ideálnej cesty ľudským hrubým črevom, reprezentovaným obrazovou informáciou z 3D počítačovej tomografie. Táto cesta má určovať trajektóriu kamery pre virtuálnu kolonoskopiu, technológiu odboru medicíny, ktorá slúži na vyšetrenie hrubého čreva pomocou počítača. Lekár jej pomocou vyhľadáva v čreve polypy resp. nádory. Prvým krokom nami zvoleného postupu je segmentácia dát, pomocou ktorej určíme objemy, v ktorých budeme hľadať cestu. Na segmentáciu využívame škálovanie a tzv. vyplňovací algoritmus. V nasledujúcom kroku v každom vysegmentovanom objeme zostrojíme počiatočný odhad krivky, ktorá bude reprezentovať hľadanú cestu. Na to využívame Dijkstrov algoritmus pomocou ktorého vypočítame vzdialenosti ku zvolenému bodu podobjemu. Počiatočnú krivku zostrojíme ako cestu z miesta s maximálnou vzdialenosťou do počiatočného bodu v smere najväčšieho klesania. V ďalšom kroku počítame vzdialenostnú funkciu ku hranici vysegmentovaného objemu. Gradient tejto funkcie určuje rýchlostné pole, do ktorého vložíme počiatočnú krivku. Projekciou rýchlostného poľa do normálovej roviny ku krivke, regularizáciou pohybu krivky vzhľadom na jej krivosť a pridaním vhodne zvolenej tangenciálnej rýchlosti dostaneme ako výsledok hladkú, asymptoticky rovnomerne rozdelenú krivku reprezentujúcu optimálnu trajektóriu pre kameru virtuálnej kolonoskopie.

Kľúčové slová: virtuálna kolonoskopia, evolúcia 3D kriviek, tangenciálna rýchlosť, asymptoticky rovnomerná redistribúcia, vzdialenostná funkcia, segmentácia

Abstract

This work deals with finding a path inside objects in 3D image data and is performed in cooperation with the company TatraMed Bratislava. The goal is to develop a suitable mathematical model and computer program for finding an ideal path in human colon represented by a visual information given by 3D computer tomography. This path will determine a trajectory of camera in virtual colonoscopy, medical technology dealing with colon diagnoses by computer. Physicians use this technology for searching polyps and tumours in colon. The first step in our approach is segmentation of subvolumes in which we find virtual camera path. In order to determine colon subvolumes, we use thresholding and region growing algorithms. In the second step we create an initial guess for the curve which will represent the searched path. We construct it by computing of a distance from a point source by Dijkstra algorithm followed by backtracking. In the next step we calculate distance function to the boundary of segmented volume. The gradient of this function determines the velocity vector field in which we insert the initial curve. Using projection of the vector field to the plane normal to evolving curve, a regularization of the motion by curvature and suitable tangential velocity, we end up with the smooth, asymptotically uniformly discretized curve representing optimal trajectory for the camera in virtual colonoscopy.

Keywords: virtual colonoscopy, evolving 3D curves, tangential velocity, asymptotically uniform redistribution, distance function, segmentation

Obsah

	Úvod		4
		Virtuálna kolonoskopia	5
		Ciele práce	7
1	Segmentácia		
	1.1	Vstupné dáta	8
	1.2	Prahovanie	9
	1.3	Vyplňovací algoritmus	10
2	Poč	iatočná krivka	14
3	Vho	odné rýchlostné pole	17
	3.1	Vzdialenostná funkcia	17
	3.2	Konštrukcia rýchlostného poľa	18
4	Optimálna 3D trajektória		20
	4.1	Krivka v rýchlostnom poli	20
	4.2	Normálová rýchlosť a krivosť	21
	4.3	Asymptoticky rovnomerná redistribúcia	23
5	Dos	siahnuté výsledky	27
Li	Literatúra		

Úvod

Kvalita života každého človeka súvisí s jeho zdravotným stavom. Ten nemusí byť vždy ideálny, ba dokonca nás často môže veľmi obmedzovať. Preto vyhľadávame nemocnice a zdravotné centrá, kde očakávame kvalitné vyšetrenie a zdravotnú starostlivosť. Po prehliadke lekár určuje diagnózu. Samotné vyšetrenia často trvajú dlho a môžu byť pre pacienta nepríjemné ba dokonca bolestivé. Na druhej strane konvenčné metódy vyšetrenia nemusia dať lekárovi dostatok informácií na presnú diagnózu. Rozvoj priemyslu a počítačovej techniky v dnešnej dobe však ponúka nové možnosti. Vďaka nim pacient absolvuje kratšie a často minimálne nepríjemné vyšetrenia a lekár ľahšie a presnejšie definuje príčinu jeho zdravotných ťažkostí.

Táto práca vznikla v spolupráci s firmou Tatramed s r.o. so sídlom v Bratislave. Firma od roku 1992 vyvíja softvérové a terapeutické systémy založené na trojrozmernej vizualizácii obrazových dát pacientov. Tie sa získajú z vyšetrení pomocou diagnostických staníc počítačovej tomografie (CT - computed tomography), magnetickej rezonancie (MR - magnetic resonance), pozitrónovej emisnej tomografie (PET - positron emission tomography), angiografie a skiagrafie. Softvér ponúka lekárovi rôzne možnosti zobrazenia takto získaných sérií pacienta. A to od najjednoduchších ako je prezeranie jednotlivých snímok až po zložitejšie ako je volume rendering. Ďalej lekár môže pomocou jednoduchých diagnostických nástrojov v pacientových snímkach označiť dôležité miesta, napísať, či dokonca nahrať správu k danému pacientovi a takto pripraviť materiál pre jeho úspešnú liečbu. Tieto systémy sa dajú tiež využiť na segmentáciu objektov v pacientových dátach a vytváranie ich relatívne presných 3D modelov. Tie sa využívajú napríklad pri plánovaní ožarovania pacientov pri rádioterapii (určia sa objemy ktoré majú byť ožiarené a tie ktoré ožiarené nesmú byť a podľa toho sa plánujú dávky a smery ožiarenia), tiež môžu pomôcť pre vizualizáciu a pre sledovanie rastu nádorov alebo priebehu vnútorných krvácaní. Navyše pomocou systémov firmy Tatramed môže lekár diagnostikovať pacienta na počítači na úplne inom mieste napríklad aj v inej nemocnici. Keďže firma nechce zaostávať za konkurenciou, snaží sa venovať svoju pozornosť i vývoju v oblasti najnovších technológií. Preto sa rozhodla vyvinúť nástroj slúžiaci pre virtuálnu kolonoskopiu (viď nasledujúcu časť). Táto práca sa zaoberá prvou časťou tohto vývoja, t.j. hľadaním ideálnej cesty pre kameru virtuálnej kolonoskopie.

Hlavným prínosom práce má byť vytvorenie programu, ktorý toho bude schopný. Vzhľadom na predchádzajúce práce zaoberajúce sa podobnou problematikou je tu použitých viacero nových prístupov. V prvej kapitole sa budeme zaoberať charakterizovaním vstupných dát a segmentáciou čreva v nich. V druhej časti sa zameriame na vytvorenie počiatočnej krivky. Na to využijeme Dijkstrov algoritmus, pomocou ktorého vypočítame vzdialenosti ku zvolenému bodu vysegmentovaného objemu. Počiatočný odhad krivky dostaneme ako cestu z miesta s maximálnou hodnotou vzdialenosti do počiatočného bodu v smere najväčšieho spádu. Získanú počiatočnú krivku vložíme do vektorového poľa. To vytvoríme opäť pomocou vzdialenostnej funkcie, ktorú budeme počítať od hranice vysegmentovaného objemu. Tá je pre 2D spracovaná v [2, 15]. V tomto prípade pôjde o nový prístup, pretože sa bude jednať o výpočet vzdialenostnej funkcie 3D relaxáciou Eikonalovej rovnice. Pomocou gradientu tejto funkcie definujeme rýchlostné pole, do ktorého vložíme počiatočnú krivku. Pohybom 2D kriviek vo vhodne zvolenom rýchlostnom poli sa zaoberajú napríklad autori v [3, 10]. V tejto práci sa budeme venovať pohybu 3D kriviek, ktorý má oproti 2D určité špecifiká. Jadrom práce je kapitola nazvaná Optimálna 3D krivka. Prezentujeme v nej podobné modifikácie rýchlostného poľa ako sú uvedené v [5, 11, 12, 13, 17], kde sa pracuje len v 2D oblasti. Rýchlostné pole projektujeme do roviny normálovej ku krivke, krivku regularizujeme pomocou krivosti a navrhujeme tangenciálnu redistribúciu bodov diskrétnej krivky. Pri návrhu vhodnej tangenciálnej rýchlosti sme potrebovali zostrojiť novú ortonormálnu bázu. Pri jej konštrukcii sme sa nechali inšpirovať článkom [8]. Prístupy, prezentované v tejto časti sú nové a originálne. Na záver zhodnotíme dosiahnuté výsledky. Navrhnutý matematický model a výpočtový algoritmus sme publikovali v článku [14].

Virtuálna kolonoskopia

Rakovina hrubého čreva je podľa oficiálnych údajov z roku 2007 v štátoch začlenených do Svetovej zdravotníckej organizácie po rakovine prsníka a pľúc treťou najčastejšou. Najrizikovejšou skupinou sú ľudia starší ako 50 rokov, diabetici, obézni ľudia, ľudia trpiaci nezhubnými ochoreniami hrubého čreva, ženy, ktoré prekonali alebo majú rakovinu prsníka, vaječníkov alebo maternice resp. ľudia, ktorým v rodine diagnostikovali kolorektálny karcinóm. Ak je však táto rakovina včas objavená, dá sa úspešne liečiť.

Kolonoskopia je vyšetrenie hrubého čreva, ktoré v čreve dokáže úspešne odhaliť polypy a kolorektálne karcinómy. Samotné vyšetrenie trvá od 15 minút do hodiny. Pri tomto vyšetrení sa používa kolonoskop (prístroj, ktorý sa skladá z hadice, má zabudovanú miniatúrnu kameru, prípadne nástroj na odoberanie tkaniva). Pacient pred vyšetrením musí dodržať predpísanú diétu (zníži príjem potravy, musí prijímať dostatočné množstvo tekutín, tri dni pred zákrokom prijíma len kašovitú stravu, deň pred vyšetrením len tekutiny a preháňadlá), aby mal hrubé črevo v čase vyšetrenia prázdne. Počas vyšetrenia sa zavádza kolonoskop cez konečník do hrubého čreva. Črevom postupuje až po jeho začiatok. Lekár na obrazovke vidí vďaka kamere kolonoskopu situáciu v samotnom čreve. Keďže je toto vyšetrenie nepríjemné a bolestivé, pacient pred jeho začiatkom dostane lieky tlmiace bolesť. Niektorí pacienti absolvujú toto vyšetrenie pod celkovou anestéziou. Podľa skúseností lekárov, len malé percento pacientov je ochotné opakovane dobrovoľne podstúpiť toto vyšetrenie a radšej sa vystavia riziku neliečenej rakoviny a pod.

Pri virtuálnej kolonoskopii ide o neinvazívne vyšetrenie použitím počítačovej tomografie. Príprava pacienta je podobná ako pri klasickom vyšetrení. Večer pred ním ale musí vypiť kontrastný roztok. Tým sa obalia všetky zvyšky stolice v čreve, čo bude vidno na snímkach z CT resp. dajú sa jednoducho odfiltrovať. Pred vyšetrením sa pacientovi nafúkne (vzduchom alebo CO_2) hrubé črevo. Potom sa pacient oskenuje v dvoch polohách (na bruchu a na chrbte) pomocou počítačovej tomografie. Pacientova úloha v tomto okamihu končí. Na rad prichádza špeciálny program, ktorý zo snímok zostaví 3D model čreva, ktorý sa dá prezerať podobne ako obraz pri klasickej kolonoskopii. Kameru virtuálnej kolonoskopie môže lekár sám ovládať, natáčať ako aj spustiť automatickú prehliadku čreva, kde kamera putuje po vypočítanej ideálnej trajektórii. Výsledky dosiahnuté virtuálnou kolonoskopiou sú porovnateľné s klasickým prístupom [9]. Lekár si snímky môže prezerať kedykoľvek a tak dlho ako potrebuje a navyše vďaka snímkam z CT vidí situáciu nie len v čreve, ale i v celej brušnej dutine. Niektoré softvéry poskytujúce virtuálnu kolonoskopiu dávajú možnosť prezerania panorámy čreva. Ide o virtuálne rozrezanie a rozbalenie čreva. V takomto zobrazení má lekár možnosť iným spôsobom sledovať štruktúru čreva. Okrem týchto prínosov, ponúka virtuálna kolonoskopia oproti klasickej i možnosť vyšetrenia pacientov, ktorí nemôžu absolvovať vyšetrenie bežným prístupom, napríklad z dôvodu, že majú pre kolonoskop nepriechodné črevo. Ďalšou výhodou oproti klasickému prístupu je, že pacient je minimálne zaťažený samotným vyšetrením. Nevýhodou je ožiarenie, ktoré pacient počas vyšetrenia na CT príjme. Jeho dávka záleží od kvality a rýchlosti CT prístroja. Vo všeobecnosti je však neškodná. Pacient tiež môže byť alergický na podávanú kontrastnú látku, čo sa však dá vopred zistiť. Hlavnou nevýhodou ostáva fakt, že pokiaľ lekár v čreve nájde polyp alebo nádor, nemôže ho týmto virtuálnym prístupom odobrať.

Cieľom tejto práce je vytvorenie matematického modelu a počítačového programu na automatické nájdenie ideálnej cesty pre kameru virtuálnej kolonoskopie. S tým je spojené osvojenie si problematiky pohybujúcich sa 3D kriviek z matematického ako aj numerického hľadiska. Predpokladá sa použitie medicínskych obrazových dát z CT. Užívateľ zadá vstupné dáta a program v nich na základe obrazovej informácie vysegmentuje podobjemy prislúchajúce črevu a vypočíta ideálnu cestu. Pod ideálnou cestou rozumieme krivku, ktorá prechádza z jedného konca čreva na druhý, ide približne stredom, je hladká a diskrétne body, ktoré ju tvoria sú na nej rovnomerne rozložené. Pokúsime sa to docieliť viacerými spôsobmi, pričom výsledky na záver porovnáme. Pri práci a implementácii budeme využívať programovací jazyk C++ vo vývojovom prostredí MS Visual Studio 2008. Na analýzu výsledkov použijeme softvéry Mathematica a ParaView.

Kapitola 1

Segmentácia

V tejto kapitole budeme hovoriť o prvom kroku nášho snaženia, o segmentácii. V prvom rade od nej závisí kvalita dosiahnutého výsledku. Tá je úzko spojená so vstupnými dátami, preto ich v úvode tejto kapitoly stručne opíšeme.

1.1 Vstupné dáta

Vstupom pre vytváranú aplikáciu budú trojrozmerné binárne zapísané *.raw súbory. Presne také súbory aké sa získavajú z CT modalít. Tieto modality určujú na základe absorpcie röntgenového žiarenia látkou, ktorou prechádzajú, hustotu tkanív v jednotlivých rezoch pacienta. Táto hustota má jednotku Hounsfield (HU). Jej rozsah je od - 1024 (vzduch) do 3071 (kosti). Tieto hodnoty sú však vo vstupných súboroch lineárne roztiahnuté do rozsahu pre dátový typ unsigned short (t.j. od 0 do 65535). Zapísané sú hneď za hlavičkou súboru, v ktorej sa určujú okrem iného i rozmery dát. Vo všeobecnosti sú dáta tvorené rezmi s rozmermi 512 x 512 pixelov, s typickou hrúbkou rezu okolo 0,75 mm. Počet rezov sa pohybuje od 600 do 850. Dáta dosahujú celkovú veľkosť približne 400MB. Kvôli ďalšej práci preškálujeme dáta na polovičné rozmery v smere x, y, aj z. Takto dostávame dáta o rozmeroch cca 256x256x400 voxelov, pričom ich veľkosť je už len okolo 50 MB. Keďže bolo žiaduce tieto dáta nejakým rozumným spôsobom zobraziť užívateľovi, museli sme ich pre vykreslenie transformovať. Dáta, ktoré sme potrebovali vidieť sú v HU v rozsahu od -1000 do 1000. Preto sme patrične orezali originálne dáta a lineárne sme ich pretransformovali do rozsahu pre bitmapy (0 až 255). Túto transformáciu sme robili len kvôli zobrazeniu. Originálne dáta sme zachovali nezmenené, pretože by sme z nich takto mohli stratiť potrebné informácie pre segmentáciu. V predloženej práci budeme naše priebežné výsledky a postupy prezentovať predovšetkým na testovacích obrázkoch.

Pre názornosť a jednoduchšiu predstavu si jednotlivé prípady ukážeme najskôr v 2D a potom analogicky v 3D. Pre 2D prípad sme vytvorili testovací obrázok pripomínajúci zjednodušený tvar čreva. Pre 3D to je zas útvar pripomínajúci otočené písmeno U.



Obr. 1.1: Testovacie dáta

1.2 Prahovanie

Ako prvý krok segmentácie čreva v medicínskych dátach sme zvolili prahovanie. Využili sme ho na odlíšenie podobjemov vyplnených plynom od ostatných objemov. Prahovanie je najjednoduchšou metódou segmentácie obrazu. Nech T predstavuje prah, f(i, j, k) hodnotu vo voxeli vstupných dát so súradnicami i, j, k a g(i, j, k) hodnotu vo výstupnom voxeli. Prah T zvolíme tak, aby zodpovedal hodnote plynu vo vstupných dátach. Je to približne -1000 HU. Ak následne použijeme predpis

$$g(i,j,k) = \begin{cases} 1, & ak f(i,j,k) \le T \\ 0, & ak f(i,j,k) > T \end{cases}$$

dostaneme ako výstup dáta, kde v každom podobjeme vyplnenom plynom bude hodnota nastavená na 1 a inde na 0. Príklad segmentácie prahovaním v jednom reze vidíme na nasledujúcom obrázku.



Obr. 1.2: Rez originálnymi dátami a ich prahovanie

1.3 Vyplňovací algoritmus

V druhom kroku segmentácie sme potrebovali navzájom odlíšiť jednotlivé podobjemy nastavené na hodnotu 1 v predchádzajúcom kroku. Na to sme použili mierne modifikované riadkové semienkové vyplňovanie (scan-line seed fill). Všetky body oblasti okrem svojej súradnice vlastnia 0 alebo 1, podľa toho, či sa v nich nachádza plyn alebo nie. Na rozbehnutie algoritmu potrebujeme ešte tri informácie. Jednak je to počiatočný vnútorný bod segmentovanej oblasti tzv. semienko, od ktorého sa začne oblasť vypĺňať. Ďalej je to podmienka, podľa ktorej sa rozhoduje, či sa hodnota v skúmanom bode má zmeniť. Môže to byť napríklad rozsah hodnôt, do ktorého musí spadať. V našom prípade to bude podmienka, či skúmaný bod je vyplnený plynom, teda, či je jeho hodnota 1. Nakoniec potrebujeme hodnotu, na ktorú sa má nastaviť vyplnený voxel. V našom prípade ju bude predstavovať poradie vysegmentovaného objemu + 1, keďže na hodnotu 1 sú nastavené všetky podobjemy vyplnené plynom. Pre jednoduchosť opisu popíšeme vyplňovací algoritmus v 2D. Uvažujme 4-smerovo súvislú oblasť, to znamená, že zo semienka sa môžeme pohybovať len vo vodorovnom a zvislom smere. V prvom kroku otestujeme suseda nad a pod semienkom. Ak niektorý z nich spĺňa určené kritérium (jeho hodnota je 1), pridáme ho do zoznamu semienok. Tiež sa nastaví prepínač up, resp down na hodnotu TRUE, že bol do zoznamu pridaný horný resp. dolný sused. Ďalej pokračujeme v testovaní susedov smerom doľava. Ak testovaný sused vyhovuje kritériu, nastavíme v ňom príslušnú hodnotu (poradie + 1). Zároveň testujeme, jeho susedov zhora a zdola. Ak horný sused nesplňa podmienku, znamená to, že v tomto bode je hrana a prepínač up nastavíme na FALSE. Ak však spĺňa podmienku, a prepínač up je nastavený na TRUE, budeme ho ignorovať (nepridáme ho do zoznamu, len v ňom nastavíme príslušnú hodnotu), pretože v zozname

semienok je už semienko z riadku tohto suseda a medzi nimi nie je hrana. Ak je up nastavené FALSE, nastavíme ho na TRUE a suseda pridáme do zoznamu semienok rovnako ako v prvom kroku. Podobne postupujeme v prípade dolného suseda. Testovanie smerom doľava pokračuje kým nenarazíme na ľavého suseda, ktorý nespĺňa požadované kritérium. Vtedy nastavíme up aj down na hodnotu FALSE a rovnaký postup vykonáme aj smerom doprava od počiatočného semienka. Keď sa práca na počiatočnom riadku ukončí, ďalšie semienka vyberáme postupne zo zoznamu semienok a podľa už opísaného postupu doň ďalšie dopĺňame. Algoritmus beží až kým sa nevyprázdni zoznam semienok. Pre 3D prípad je postup podobný, len uvažujeme 6-smerovo súvislú oblasť, kde nám pribudne sledovanie susedov aj smerom dopredu a dozadu. Samotný algoritmus budeme spúšťať pre voxely nastavené na jednotku. Najskôr si však musíme uvedomiť, že najväčší podobjem vyplnený vzduchom obklopuje telo pacienta. V tomto priestore, však nebudeme chcieť hľadať cestu. Preto ho odstránime. Využijeme popísaný algoritmus, len danú oblasť nastavíme na hodnotu 0. Keďže vo vstupných dátach sa okrem pacienta nachádza i stôl na ktorom leží, prípadne jeho oblečenie, môže sa stať, že ak by sme zvolili pre odstránenie oblasti vonkajšieho vzduchu niektorý okrajový voxel, vypĺňanie by zastalo na hranici stola. Preto najskôr v dátach vytvoríme okolo pacienta úzky pás, ktorý nastavíme na 1 a následne vložíme semienko do niektorého okrajového voxela, pričom príslušný objem nastavíme na hodnotu 0. Tento postup je pre 2D rez znázornený na nasledujúcich obrázkoch.



Obr. 1.3: Odstránenie podobjemu so vzduchom okolo pacienta

Po odstránení vonkajšieho podobjemu budeme hľadať v celom objeme voxely s hodnotou 1. Postupovať môžeme napríklad odspodu. Keď nájdeme vyhovujúci voxel, spustíme v ňom vyplňovací algoritmus. Hodnoty v príslušnom objeme nastavíme na hodnotu 2 (t. j. poradie + 1). Keď sa celý podobjem vyplní, postupujeme ďalej v hľadaní voxela s hodnotou 1 tam, kde sme naposledy zastali. Takto pokračujeme až kým neprídeme na koniec dát a teda neostal už žiadny voxel nastavený na 1. Počas vypĺňania sme tiež počítali počet vyplnených voxelov v jednotlivých podobjemoch, a tak sme zisťovali ich približný objem. Ak bol veľmi malý (menší ako zvolená minimálna hodnota), tak sme ich ignorovali (nastavili sme ich hodnotu na 0). Takto sme odstránili malé objemy vyplnené plynom (napríklad pľúca, atď.)

V poslednom kroku segmentácie sme odstránili príliš tenké objemy vyplnené vzduchom. Také vznikajú napríklad medzi pacientom a stolom CT, na ktorom leží. Na to sme využili vzdialenostnú funkciu podrobne opísanú v kapitole 3.1. Pomocou nej sme vypočítali v jednotlivých podobjemoch vzdialenosti k ich hraniciam. Pre každý podobjem sme následne určili maximálnu vypočítanú vzdialenosť. Ak bola príliš malá (objem bol veľmi tenký), tak sme ho ignorovali (jeho hodnotu sme nastavili na 0).

Na obr. 1.3 vidíme výsledok segmentácie po odstránení vonkajšieho vzduchu a následne po odstránení malých a tenkých podobjemov znázornený v programe ParaView. Vidíme, že sa nám podarilo vysegmentovať hrubé a dokonca i časti tenkého čreva. Značná nespojitosť častí tenkého čreva je spôsobená tým, že nie je úplne vyprázdnené a ani celkom nafúknuté. To nám však nevadí, keďže cieľom je vyšetrovať hrubé črevo.



Obr. 1.4: Výsledok segmentácie po odstránení vonkajšieho podobjemu vzduchu (vľavo) a malých a tenkých podobjemov (vpravo)

Podmienkou presnej segmentácie je okrem kvality vstupných dát vhodne zvolená to-

lerancia pri prahovaní a to, či je črevo úplne vyprázdnené. Existujú však algoritmy, ktoré dokážu z čreva odstrániť i nechcenú tekutinu ba dokonca i zvyšky stolice.

Kapitola 2

Počiatočná krivka

V predchádzajúcej časti sme úspešne vysegmentovali oblasť, cez ktorú má viesť cesta kamery pre virtuálnu kolonoskopiu. Skôr ako začneme tvarovať hľadanú cestu, musíme navrhnúť počiatočný tvar krivky, ktorá ju bude reprezentovať. Pri konštrukcii počiatočnej krivky sme sa nechali inšpirovať článkom [16]. Zvolili sme ľubovoľný vnútorný bod vysegmentovaného objemu a vo vnútri objemu sme vypočítali vzdialenosti k tomuto bodu. Na to sme použili rýchly Dijkstrov algoritmus [4] (hrany grafu spájajúce susedné voxely majú hodnotu 1). Následne sme zistili, kde sa nachádza maximálna hodnota vzdialenost. Keďže črevo je podlhovastý orgán (jeho dĺžka je výrazne väčšia ako jeho hrúbka), maximum sa bude nachádzať na jednom z koncov vysegmentovaného podobjemu. Z tohto bodu opäť vypočítame vzdialenosti. Nové maximum dosiahneme na druhom konci čreva. Ak sa potom vrátime z tohto maxima cestou cez najväčší spád, t.j. najväčšej absolútnej hodnoty gradientu, dostaneme sa opäť na začiatok podobjemu. Voxelové súradnice cesty, ktorá bola týmto spôsobom prekonaná, budú predstavovať parametrickú 3D krivku, počiatočný odhad cesty vo vysegmentovanom podobjeme. Na nasledujúcich obrázkoch vidíme vypočítané vzdialenosti pomocou Dijkstrovho algoritmu ako i získané počiatočné krivky.

Na obrázkoch je vidno, že počiatočná krivka prechádza celým črevom. Je však zrejmé, že nespĺňa požiadavky na ideálnu cestu. Nie je ani hladká, ani nejde stredom čreva. Naopak, je viackrát zalomená a ťahá sa často po stene vysegmentovaného objemu.



Obr. 2.1: Graf vypočítaných vzdielností pre 2D dáta zobrazený v 3D a v 2D spolu s počiatočnou krivkou



Obr. 2.2: Graf vypočítaných vzdielností pre 3D dáta zobrazený v 2D reze a 3D počiatočná krivka vo vysegmentovanom objeme



Obr. 2.3: Vysegmentované črevo s počiatočnou cestou

Kapitola 3

Vhodné rýchlostné pole

Vo štvrtej kapitole sa budeme venovať pohybu krivky v rýchlostnom poli. V tejto časti toto vektorové pole navrhneme ako gradient vzdialenostnej funkcie počítanej k hranici vysegmentovaného objemu. Navrhnuté rýchlostné pole bude hlavnou hybnou silou krivky a bude prispievať k tomu, aby výsledná cesta ležala približne v strede čreva.

3.1 Vzdialenostná funkcia

Ako bolo spomenuté, v každom vysegmentovanom podobjeme potrebujeme vypočítať vzdialenostnú funkciu určujúcu vzdialenosť od hranice podobjemu. Jej návrh a výpočet je podrobne opísaný pre 2D prípad v článku [2, 15]. My uvedieme jej výpočet pre 3D prípad.

Označme hľadané riešenie vzdialenostnej funkcie ako d. To získame riešením Eikonalovej diferenciálnej rovnice

$$d_t + |\nabla d| = 1. \tag{3.1}$$

Výpočtovou oblasťou je vnútro vysegmentovaného objemu. Pred samotným výpočtom určíme okraj oblasti. Sú to tie vysegmentované voxely, ktoré susedia s voxelmi s hodnotou 0. Úlohu budeme riešiť s nulovými Dirichletovými okrajovými podmienkami na hranici vysegmentovanej oblasti. Teda na hranici bude hodnota vzdialenostnej funkcie fixovaná počas celého výpočtu na 0.

Rovnica (3.1) môže byť diskretizovaná podľa Rouy-Tourinovej schémy. Označme d_{ijk}^n približnú hodnotu riešenia d v časovom kroku n v strede voxela s priestorovou súradnicou $(i, j, k), \tau_D$ dĺžku časového kroku a h_D dĺžku hrany voxela. Ďalej definujme funkciu M_{ijk}^{pqr} , kde $p,q,r \in \{-1,0,1\}, |p|+|q|+|r|=1$ ako

$$M_{ijk}^{pqr} = (\min(d_{i+p,j+q,k+r}^n - d_{ijk}^n, 0))^2.$$
(3.2)

Rouy-Tourinovej schému pre (3.1) potom môžeme napísať ako

$$d_{ijk}^{n+1} = d_{ijk}^{n} + \tau_D - \frac{\tau_D}{h_D} \sqrt{\max(M_{ijk}^{-1,0,0}, M_{ijk}^{1,0,0}) + \max(M_{ijk}^{0,-1,0}, M_{ijk}^{0,1,0}) + \max(M_{ijk}^{0,0,-1}, M_{ijk}^{0,0,1})}.$$

Ako vidno z numerickej schémy, na hodnotu vzdialenostnej funkcie vplývajú hodnoty v susedných bodoch. Algoritmus funguje tak, že na začiatku zafixujeme na nulu (resp. nastavíme vhodne okrajové podmienky) hodnoty v bodoch, ku ktorým chceme počítať vzdialenosť a iteračne ju počítame vo všetkých ostatných bodoch vysegmentovaného objemu dovtedy, kým model neskonverguje k vzdialenosti približne rovnej skutočnej. Algoritmus sa dá urýchliť rôznymi spôsobmi. Napríklad tak, že na začiatku v celom objeme namiesto nuly nastavíme najmenšiu vzdialenosť medzi stredmi dvoch susedných voxelov. Tiež je možné fixovať priebežne hodnoty v tých bodoch, ktoré sa už nemenia. O urýchlení algoritmu sa viac píše v [2].

Na nasledujúcich obrázkoch vidíme vypočítané vzdialenostné funkcie pre testovacie obrázky.



Obr. 3.1: Graf vzdialenostnej funkcie počítanej od hranice pre 2D dáta zobrazený v 2D a detail zobrazenia v 3D

3.2 Konštrukcia rýchlostného poľa

Ako už bolo spomenuté, samotné vektorové pole dostaneme ako gradient vzdialenostnej funkcie (3.1). Teda



Obr. 3.2: Graf vzdialenostnej funkcie počítanej od hranice pre 3D dáta zobrazený v 2D reze

$$\mathbf{v}(x,y,z) = \nabla d(x,y,z) = \left(\frac{\partial d}{\partial x}, \frac{\partial d}{\partial y}, \frac{\partial d}{\partial z}\right)^T.$$
(3.3)

Gradient budeme počítať numerickou aproximáciou centrálnou diferenciou

$$\mathbf{v}_{ijk} = \left(\frac{d_{i+1jk} - d_{i-1jk}}{2h}, \frac{d_{ij+1k} - d_{ij-1k}}{2h}, \frac{d_{ijk+1} - d_{ijk-1}}{2h}\right).$$
(3.4)



Obr. 3.3: Znázornenie detailu vektorového poľa vypočítaného na 2D dátach

Týmto spôsobom získame rýchlostné pole smerujúce ku "hrebeňu" vzdialenostnej funkcie. Práve naň chceme dostať našu počiatočnú krivku. Toto pole je najďôležitelšou informáciou pre modely uvedené v nasledujúcej kapitole. Na základe neho nájdeme ideálnu trajektóriu pre kameru viruálnej kolonoskopie.

Kapitola 4

Optimálna 3D trajektória

V druhej kapitole sme navrhli a vytvorili počiatočnú krivku. V tejto časti z nej vymodelujeme ideálnu cestu vysegmentovaným objemom. Na tento účel využijeme predovšetkým navrhnuté rýchlostné pole. To bude hlavnou hybnou silou krivky. V tejto kapitole budeme okrem iného prezentovať rôzne modifikácie tohto rýchlostného poľa a ich vplyv na výslednú krivku. Pohybom 2D kriviek riadeným vhodne zvoleným rýchlostným poľom sa zaoberajú [3, 10, 17].

4.1 Krivka v rýchlostnom poli

Krivka bude reprezentovaná bodmi. Tie budeme označovať ich polohovými vektormi \mathbf{r}_{i}^{n} , kde *i* označuje poradie bodu krivky a *n* časový krok pohybu, pričom nultý a *m*-tý bod považujeme za fixovaný. Najjednoduchší pohyb krivky môžeme popísať nasledovne

$$\partial_t \mathbf{r} = \mathbf{v}.\tag{4.1}$$

Ide vlastne o pohyb bodov zapríčinený len rýchlostným poľom. Novú pozíciu bodu teda dostaneme pomocou nasledujúceho vzťahu

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \tau \mathbf{v},\tag{4.2}$$

kde τ predstavuje zvolený časový krok. Na obr. 4.1 vidíme výsledky dosiahnuté týmto prístupom. Jasne je vidno vplyv rýchlostného poľa na pohybujúce sa body. Krivky sa pohli ku stredu vysegmentovaného objemu. Body krivky nie sú žiadnym spôsobom zviazané ani regularizované. Najviac je to vidno v 2D prípade, kde vektorové pole zhlukuje body do skupín v stredoch jednotlivých krúžkov. Z tohto dôvodu krivky nie sú hladké. Ak pospájame body takto vytvorenej krivky môže sa stať, že ich spojnice pretnú hranicu vysegmentovaného podobjemu. Toto riziko je pri tak členitej oblasti ako je črevo veľmi vysoké. Tento problém bude treba riešiť.

Treba poznamenať, že existujú štandardné prístupy virtuálnej kolonoskopie [6, 16, 18], využívajúce kombináciu (tzv. váženú sumu) dvoch vzdialenostných funkcií. Jednej vzdialenostnej funkcie počítanej od štartovacieho bodu a druhej, ktorá je počítaná od hranice objemu. Krivka sa vytvorí ako cesta z globálneho maxima do minima v smere najväčšieho klesania. Tento prístup však môže viesť k závažným problémom pri vytváraní trajektórie, kedy vytváraná cesta môže uviaznuť v miestach s lokálnym minimom. Iné metódy zas nezabezpečia, aby výsledná krivka ležala v strede objemu, ale len v určitom páse okolo stredu. Navyše potrebujú poznať minimálnu hrúbku čreva.



Obr. 4.1: Výsledky dosiahnuté len vplyvom rýchlostného poľa pre 2D a 3D dáta

4.2 Normálová rýchlosť a krivosť

Ako sme spomenuli v predchádzajúcej časti, chceme nejakým spôsobom docieliť, aby body pohybovanej krivky boli nejakým spôsobom zviazané, aby sa nezhlukovali do skupín a aby bola krivka hladká. Momentálny model totiž tieto požiadavky nespĺňa a preto je nepoužiteľný pre zložitejšie štruktúry akou je napríklad hrubé črevo. Medzi spomenutými skupinami bodov môže byť veľká vzdialenosť a teda nie je zaručené, že medzi nimi nie je stena čreva, na ktorú by sme tým pádom mohli naraziť. Pozitívom prvotného modelu však je, že rýchlostné pole detekuje stred vysegmentovaného objemu.

Pohyb krivky vo všeobecnosti môžeme rozložiť na pohyb v smere tangenty a normály. Ako je uvedené aj v [17], ak je diskrétna krivka dostatočne jemne delená, jej tvar nezáleží od tangenciálnej zložky rýchlosti. Tá má vplyv len na rozloženie bodov na krivke, ktoré je momentálne nevhodné. Tvar krivky určuje normálová zložka rýchlosti. V tejto časti nevhodnú tangenciálnu zložku pohybu zanedbáme. Pohyb krivky v normálovej rovine ku krivke v 3D definujeme ako

$$\partial_t \mathbf{r} = \mu \, \mathbf{N}_{\mathbf{v}} + \epsilon k \mathbf{N} \,, \tag{4.3}$$

kde N_v predstavuje rýchlosť projektovanú do normálovej roviny, N predstavuje jednotkový normálový vektor 3D krivky, μ a ϵ sú parametre. Vektor T na obr. 4.2 pred-



Obr. 4.2: Projekcie rýchlostného poľa do roviny normálovej ku krivke

stavuje jednotkový tangenciálny vektor ku krivke a **v** vektor rýchlostného poľa. Projekciou rýchlosti do vektora **T** násobenou **T** dostaneme $\mathbf{T}_{\mathbf{v}} = (\mathbf{T}.\mathbf{v})\mathbf{T}$. Vektor projektovaný do normálovej roviny ku krivke následne dostaneme ako rozdiel $\mathbf{N}_{\mathbf{v}} = \mathbf{v} - \mathbf{T}_{\mathbf{v}}$. Člen $k\mathbf{N}$ uvedený v (4.3) predstavuje vektor krivosti počítaný podľa schémy z [17]

$$k_i^n \mathbf{N}_i^n \approx \frac{2}{h_{i+1}^n + h_i^n} \left(\frac{\mathbf{r}_{i+1}^n - \mathbf{r}_i^n}{h_{i+1}^n} - \frac{\mathbf{r}_i^n - \mathbf{r}_{i-1}^n}{h_i^n} \right) \quad i = 1, 2, \dots m - 1.$$
(4.4)

Celé číslo m označuje počet bodov počítanej krivky
ai je poradové číslo bodu. Na výpočet vzdialeností susedných uzlov
 h_i v časovom okamihu n použijeme vzťah

$$h_i^n = \sqrt{(x_i^n - x_{i-1}^n)^2 + (y_i^n - y_{i-1}^n)^2 + (z_i^n - z_{i-1}^n)^2}, \quad \text{kde} \quad \mathbf{r}_i^n = (x_i^n, y_i^n, z_i^n).$$
(4.5)

Rovnicu (4.3) potom môžeme prepísať ako

$$\frac{\mathbf{r}_{i}^{n+1} - \mathbf{r}_{i}^{n}}{\tau} = \mu \left(\mathbf{N}_{\mathbf{v}} \right)_{i}^{n} + \epsilon \frac{2}{h_{i+1}^{n} + h_{i}^{n}} \left(\frac{\mathbf{r}_{i+1}^{n} - \mathbf{r}_{i}^{n}}{h_{i+1}^{n}} - \frac{\mathbf{r}_{i}^{n} - \mathbf{r}_{i-1}^{n}}{h_{i}^{n}} \right) \,. \tag{4.6}$$

Týmto modelom odstránime nevhodný tangenciálny pohyb bodov krivky. Krivostný člen navyše prináša citlivosť pohybu bodov na vzdialenosť od susedov a tak ich navzájom



Obr. 4.3: Výsledky dosiahnuté projekciou do normálovej roviny a krivosťou na 2D a 3D dátach

zväzuje. Krivka získaná týmto modelom má regulárnejšie rozloženie bodov ako pri predchádzajúcom postupe a je takisto umiestnená približne v strede vysegmentovaného objemu. Výsledky dosiahnuté na testovacích dátach sú zobrazené na obr. 4.2. Ako vidíme, výsledky sú lepšie ako po použití prvého modelu. Vidíme však aj voľným okom, že body krivky sú stále nerovnomerne rozmiestnené. To môže spôsobiť pri zložitejších štruktúrach nesprávny pohyb krivky v rýchlostnom poli. Ak chceme, aby tvar krivky čo najlepšie vystihoval hrebeň vzdialenostnej funkcie, musí byť táto krivka čo najrovnomernejšie rozdelená. To dosiahneme pridaním vhodne zvolenej tangenciálnej rýchlosti do modelu. Tá zabezpečí rovnomernú redistribúciu bodov krivky.

4.3 Asymptoticky rovnomerná redistribúcia

V 3D prípade nemôžeme priamočiaro zovšeobecniť 2D prístup pre tangenciálnu redistribúciu zavedenú v [11],[12]. Dôvodom je fakt, že nevieme definovať krivosť ako skalár so znamienkom. Takisto Frenetov trojhran má singulárne správanie v bodoch, kde je vektor krivosti $k\mathbf{N}$ rovný nule. Preto zavedieme inú ortogonálnu bázu, ktorej správanie bude pozdĺž krivky hladké. Takýto postup bol realizovaný v [8]. Tangenciálna redistribúcia tam bola vytvorená pre tzv. $k_1 - k_2 - \omega - L$ formuláciu pohybu 3D krivky. My túto tangenciálnu redistribúciu zovšeobecníme na asymptoticky rovnomernú, pričom uvažujeme štandardnú formuláciu pre evolúciu pozičného vektora **r** 3D krivky.

Našu novú bázu bude tvoriť T, t.j. jednotkový tangenciálny vektor krivky, a dva vektory ležiace v normálovej rovine ku krivke definované ako $N_1 = \frac{N_v}{|N_v|}$ a $N_2 = N_1 \times T$.

$$k\mathbf{N} = k_1\mathbf{N}_1 + k_2\mathbf{N}_2. \tag{4.7}$$

 ${\rm V}$ [8] autori uvádzajú rozklad pohybu krivky do ortonormálnej bázy ako

$$\partial_t \mathbf{r} = U \mathbf{N_1} + V \mathbf{N_2} + \alpha \mathbf{T},\tag{4.8}$$

kde α je tangenciálna komponenta pohybu
aUresp. Vsú normálové komponenty, vypočítané vzťahmi

$$U = k_1 + \mu |\mathbf{N}_{\mathbf{v}}| \tag{4.9}$$

$$V = k_2. \tag{4.10}$$

Nech krivka Γ je parametrizovaná funkciou pozičného vektora **r**. Definujme veličinu určujúcu lokálnu dĺžku g na krivke, $g = \left|\frac{\partial \mathbf{r}}{\partial u}\right|$, kde $u \in [0, 1]$. V diskrétnom prípade je aproximovaná ako $g \approx \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{h}$, pričom $h = \frac{1}{m}$ predstavuje rovnomerné delenie rozsahu parametra u, t.j. intervalu [0,1]. Z článku [8] vyplýva, že pre časovú zmenu lokálnej dĺžky 3D krivky platí

$$\partial_t g = g \partial_s \alpha - g(Uk_1 + Vk_2), \tag{4.11}$$

kdes je dĺžková parametrizácia. Z toho dostaneme vzťah pre časový vývoj globálnej dĺžky L krivky Γ

$$\frac{dL}{dt} = - \langle Uk_1 + Vk_2 \rangle_{\Gamma} L, \qquad (4.12)$$

kde výraz
 $< Uk_1 + Vk_2 >_{\Gamma}$ označuje strednú hodnotu pozdĺž krivky. Budeme ho aproximovať ako

$$\langle Uk_1 + Vk_2 \rangle_{\Gamma}^n = \frac{1}{L^n} \sum_{l=1}^m h_l^n (U_l^n k_{1l}^n + V_l^n k_{2l}^n)$$
$$L^n = \sum_{l=1}^m h_l^n,$$

kde $U_l^n, k_{1l}^n, V_l^n, k_{2l}^n$ sú aproximácie príslušných kvantít na diskrétnom segmente krivky. Aby sme odvodili asymptoticky rovnomernú redistribúciu bodov krivky, zaujímavý je pre nás podiel

$$\frac{g}{L} \approx \frac{|\mathbf{r}_i - \mathbf{r}_{i-1}|}{Lh} = \frac{|\mathbf{r}_i - \mathbf{r}_{i-1}|}{\left(\frac{L}{m}\right)} = \frac{h_i}{\left(\frac{L}{m}\right)}.$$
(4.13)

Ten predstavuje pomer lokálnej dĺžky delenia krivky a priemernej dĺžky jej delenia. Ak sa podiel rovná 1, dosiahli sme rovnomernú redistribúciu. Označme

$$\theta = \ln\left(\frac{g}{L}\right) \,. \tag{4.14}$$

Cieľom je, aby sa asymptoticky $\theta=0.$ Keďže platí

$$\partial_t \theta = \partial_t \left[\ln \left(\frac{g}{L} \right) \right] = \frac{L}{g} \frac{\partial_t g L - g \partial_t L}{L^2}, \tag{4.15}$$

po dosadení (4.11) a (4.12) do (4.15) dostaneme

$$\partial_t \theta = \partial_s \alpha - (Uk_1 + Vk_2) + \langle Uk_1 + Vk_2 \rangle_{\Gamma} .$$

$$(4.16)$$

Keďže chceme, aby $\theta \to 0$ pre $t \to \infty$, zvolíme ju napríklad takú, že spĺňa diferenciálnu rovnicu $\partial_t \theta = (e^{-\theta} - 1)\omega_r$, kde ω_r je rýchlosť priebehu redistribúcie. Pravá strana vzťahu (4.14) sa zjednoduší a dostaneme $e^{-\ln(\frac{q}{L})} - 1 = e^{\ln(\frac{L}{g})} - 1 = \frac{L}{g} - 1$ a teda z (4.16) dostaneme vzťah pre tangenciálnu rýchlosť α . Tou dosiahneme asymptoticky rovnomernú redistribúciu bodov krivky.

$$\partial_s \alpha = Uk_1 + Vk_2 - \langle Uk_1 + Vk_2 \rangle_{\Gamma} + \left(\frac{L}{g} - 1\right)\omega_r.$$

$$(4.17)$$

Keďže platí $\mathbf{T} = \partial_s \mathbf{r}$ a $k\mathbf{N} = \partial_{ss}\mathbf{r}$, môžeme výsledný model aj s tangenciálnou zložkou rýchlosti zapísať v tvare intrinsickej advekčno-difúznej parciálnej diferenciálnej rovnice s riadiacou silou

$$\partial_t \mathbf{r} = \mu \, \mathbf{N}_{\mathbf{v}} + \epsilon \, \partial_{ss} \mathbf{r} + \alpha \partial_s \mathbf{r}, \tag{4.18}$$

s α danou vzťahom (4.17) a splnenými Dirichletovými okrajovými podmienkami (fixované koncové body krivky).

Naším posledným krokom je numerická diskretizácia rovníc (4.17) a (4.18). Podobne ako v prácach [11, 13], dostaneme diskrétny tvar pre výpočet tangenciálnej rýchlosti ako

$$\alpha_i^n = \alpha_{i-1}^n + h_i^n (U_i^n k_{1i}^n + V_i^n k_{2i}^n) - h_i^n < Uk_1 + Vk_2 >_{\Gamma}^n + \left(\frac{L^n}{m} - h_i^n\right) \omega_r , \qquad (4.19)$$

pre $i = 1, ..., m - 1, \alpha_0^n = 0$. V poslednom bode bude vychádzať tiež $\alpha_m^n = 0$. Diskretizáciu rovnice (4.18) spravíme použitím semi-implicitnej schémy. Dostaneme

$$\frac{h_{i+1}^{n} + h_{i}^{n}}{2} \frac{\mathbf{r}_{i}^{n+1} - \mathbf{r}_{i}^{n}}{\tau} = \mu \frac{h_{i+1}^{n} + h_{i}^{n}}{2} (\mathbf{N}_{\mathbf{v}})_{i}^{n} +$$

$$\epsilon \left(\frac{\mathbf{r}_{i+1}^{n+1} - \mathbf{r}_{i}^{n+1}}{h_{i+1}^{n}} - \frac{\mathbf{r}_{i}^{n+1} - \mathbf{r}_{i-1}^{n+1}}{h_{i}^{n}} \right) + \frac{\alpha_{i}^{n}}{2} (\mathbf{r}_{i+1}^{n+1} - \mathbf{r}_{i-1}^{n+1}),$$
(4.20)

pre $i = 1, \ldots, m - 1$, s predpísanými \mathbf{r}_0^{n+1} a \mathbf{r}_m^{n+1} .

Schéma (4.20) reprezentuje tri lineárne troj-diagonálne systémy pre súradnice x, y, zbodov krivky. Rovnice môžu byť prepísané ako

$$\mathcal{A}_i^n \mathbf{r}_{i-1}^{n+1} + \mathcal{B}_i^n \mathbf{r}_i^{n+1} + \mathcal{C}_i^n \mathbf{r}_{i+1}^{n+1} = \mathcal{F}_i^n$$
(4.21)

s koeficientami

$$\mathcal{A}_{i}^{n} = -\frac{\epsilon}{h_{i}^{n}} + \frac{\alpha_{i}^{n}}{2}, \quad \mathcal{C}_{i}^{n} = -\frac{\epsilon}{h_{i+1}^{n}} - \frac{\alpha_{i}^{n}}{2}, \quad \mathcal{B}_{i}^{n} = \frac{h_{i}^{n} + h_{i+1}^{n}}{2\tau} - \mathcal{A}_{i}^{n} - \mathcal{C}_{i}^{n},$$
$$\mathcal{F}_{i}^{n} = \frac{h_{i}^{n} + h_{i+1}^{n}}{2\tau} \mathbf{r}_{i}^{n} + \mu \left(\mathbf{N}_{\mathbf{v}}\right)_{i}^{n} \frac{h_{i}^{n} + h_{i+1}^{n}}{2}.$$

Troj-diagonálne systémy riešime Thomasovým algoritmom, ktorý je veľmi rýchly. Jeho numerická stabilita je podmienená striktnou diagonálnou dominantnosťou matice systému. Táto vlastnosť môže byť ľahko splnená vhodnou voľbou časového kroku. V skutočnosti (vďaka takmer rovnomernému deleniu krivky) môže byť τ proporcionálne kroku priestorovej diskretizácie, a preto numerické riešenie je realizované rýchlym a stabilným spôsobom.

Výsledky dosiahnuté týmto prístupom na testovacích dátach sú zobrazené na obr. 4.4. Vidíme, že body krivky sú rovnomerne rozdelené a krivka spĺňa všetky stanovené požiadavky. Našli sme teda ideálnu cestu pre kameru virtuálnej kolonoskopie.



Obr. 4.4: Výsledky dosiahnuté výsledným modelom na 2D a 3D dátach

V prezentovaných výpočtoch sme používali parametre $\mu=\epsilon=\tau=\omega_r=1$. Metóda je robustná s ohľadom na ich voľbu.

Kapitola 5

Dosiahnuté výsledky

V predchádzajúcich kapitolách sme hľadali krivku, ktorá by mohla predstavovať ideálnu cestu pre kameru virtuálnej kolonoskopie. Videli sme, ako sa naše výsledky postupne zlepšovali. Mohli sme pozorovať, aký vplyv na krivku má rýchlostné pole a prečo je dobré ho modifikovať. Záverečným modelom (4.18) sme dosiahli vizuálne najlepšie výsledky. Nielen čo sa týka tvaru ale hlavne rozloženia bodov na krivke. Číselné porovnanie redistribúcie bodov krivky poskytuje nasledujúci obrázok. Na grafe vidíme vplyv modelov na redistribúciu bodov. Porovnávame model (4.1) (červená farba), model (4.3) (modrá farba) a model (4.18) (fialová farba). Na grafoch sú znázornené vzdialenosti susedných bodov pre 3D testovacie dáta.



Obr. 5.1: Porovnanie redistribúcie bodov v jednotlivých modeloch pre testovacie 3D dáta

Krivky dosiahnuté posledným modelom majú vhodné vlastnosti aby tvorili trajektóriu kamery pre virtuálnu kolonoskopiu. Na nasledújúcich obrázkoch sú znázornené výsledky dosiahnuté na ďalších testovacích dátach (špirála) a na dátach reálneho pacienta.

Na obr. 5.4 vidíme výstup komerčnej aplikácie, do ktorej bol implementovaný nami



Obr. 5.2: Počiatočná a výsledná krivka nájdená v 3D špirále.

navrhnutý výpočtový model.

Výbornou vlastnosťou celého výpočtu je jeho rýchlosť. Na pracovnej stanici s parametrami 4GB RAM, 1.6 GHz Core 2 Duo, trval celý náš algoritmus cca 8 sekúnd a pamäťová náročnosť celého procesu bola cca 180 MB.



Obr. 5.3: Výsledky dosiahnuté na reálnych dátach



Obr. 5.4: Ukážky z výstupu komerčnej aplikácie do ktorej bol implementovaný navrhnutý algoritmus

Literatúra

- [1] Bovik, A.: Handbook of Image and Video Processing, Academic Press, 2000.
- [2] Bourgine, P., Frolkovič, P., Mikula, K., Peyrieras, N., Remešíková, M.: Extraction of the intercellular skeleton from 2D microscope images of early embryogenesis, in Lecture Notes in Computer Science 5567, Springer, 2009, pp.39-49.
- [3] Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours, International Journal of Computer Vision, 22 (1997), pp. 61–79.
- [4] Dijkstra, E. W.: A note on two problems in connection with graphs, Numerische Mathematik 1 (1959) pp. 269271.
- [5] Dziuk, G.: Convergence of a semi discrete scheme for the curve shortening flow, Mathematical Models and Methods in Applied Sciences, 4 (1994) pp. 589-606.
- [6] Hong, L., Kaufman, A. E., Wei, Y., Viswambharan, A., Wax, M., Liang, Z.: 3D Virtual Coloscopy, IEEE Symposium on Biomedical Visualization (1995) pp. 26-32.
- [7] Hong, L., Muraki, S., Kaufman, A. E., Bartz, D., He, T.: Virtual voyage: Interactive navigation in the human colon, ACM SIGGRAPH 97 (1997) pp. 27-34.
- [8] Hou, T. Y., Klapper, I., Si, H.: Removing the stiffness of curvature in computing 3-D filaments, J. Comput. Physics , 143 (1998) 628-664.
- [9] Lefere, P., Gryspeerdt, S. (Eds.): Virtual colonoscopy: A practical guide, Springer-Verlag Berlin Heidelberg, 2010, ISBN 978-3-540-79879-8
- [10] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Conformal curvature flows: from phase transitions to active vision, Arch. Rational Mech. Anal., 134 (1996), pp. 275–301.
- [11] Mikula, K., Ševčovič, D.: Evolution of plane curves driven by a nonlinear function of curvature and anisotropy, SIAM J. Appl. Math., 61 (2001), pp. 1473-1501.

- [12] Mikula, K., Ševčovič, D.: 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.
- [13] Mikula, K., Ševčovič, D., Balažovjech, M.: A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations, Comm. Comp. Physics, Vol. 7, No. 1 (2010) pp. 195-211.
- [14] Mikula, K., Urbán, J.: 3D curve evolution algorithm with tangential redistribution for a fully automatic finding of an ideal camera path in virtual colonoscopy, Lecture Notes in Computer Science 6667, Springer, 2011
- [15] Rouy, E., Tourin, A.: Viscosity solutions approach to shape-from-shading, SIAM Journal on Numerical Analysis 29 No. 3 (1992) pp. 867–884.
- [16] Truyen, R., Deschamps, T., Cohen, L. D.: Clinical evaluation of an automatic path tracker for virtual colonoscopy, Medical Image Computing and Computer-Assisted Intervention (MICCAI), Utrecht, Netherlands, October 2001.
- [17] Urbán, J.: Evolúcia rovinných kriviek v spracovaní obrazu, Slovenská Technická Univerzita v Bratislave, 2009.
- [18] Wan, M., Dachille, F., Kaufman, A. E.: Distance-field based skeletons for virtual navigation, Proceedings of IEEE Visualization conference 2001, pp 239- Â-245.