

UNIVERZITA KARLOVA V PRAZE
MATEMATICKO-FYZIKÁLNÍ FAKULTA

DIPLOMOVÁ PRÁCE



Václav Krajíček

Měření objemu v 3D datech

Duben 2007

Kabinet software a výuky informatiky
Vedoucí diplomové práce: **RNDr. Josef Pelikán**
Studijní program: **Informatika**

Děkuji RNDr. Pelikánovi za cenné rady a připomínky k této práci. MUDr. Horákovi za poskytnutí lékařských znalostí a dat. Mgr. Maršálkovi za pomoc a podporu.

Prohlašuji, že jsem svou diplomovou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 20. Dubna 2007

Václav Krajíček
vlastnoruční podpis

Obsah

1	Úvod	11
1.1	Cíle	12
1.2	Struktura textu	13
2	Zobrazovací metody v lékařství	14
2.1	Počítačová tomografie	15
2.2	Zobrazení tomografických snímků	16
2.3	Zpracování a analýza dat	17
2.4	Měření objemu	17
3	Metody segmentace	18
3.1	Prahování	18
3.2	Narůstání oblastí	19
3.3	Deformovatelné modely	20
3.4	Obecné vlastnosti algoritmů	20
4	Deformovatelné modely	22
4.1	Active Contour Models - Snakes	22
4.1.1	Diferenciální rovnice a jejich řešení	22
4.1.2	Odvození Active Contours	23
4.1.3	Optimalizace	24
4.1.4	Vlastnosti	26
4.2	Balónky	27
4.3	Další vylepšení pro Aktivní kontury	28
4.3.1	Modely zohledňující topologii - T-Snakes	28
4.3.2	Segmentace série řezů	29
4.3.3	Aktivní plochy	29
4.4	Geodesic contours	29
4.4.1	Optimalizace - Snižování gradientu	30
4.5	Level-sets	31
4.5.1	Narrow-band	34
4.5.2	Fast Marching Level-sets	35
4.6	Modely řízené statistikou oblasti	35
4.6.1	Využití Greenovy věty	36
4.7	Apriorní znalosti	38

4.7.1	Statistická analýza - PCA	38
4.7.2	Point Distribution Model	38
4.7.3	Active Shape Model	41
4.7.4	Active Appearance Model	42
4.8	Kombinace všeho - Shape Based Approach	42
4.9	B-Spline a Free-Form Deformace	45
4.10	Shrnutí	46
5	Zvolený postup rekonstrukce	47
5.1	Rekonstrukce postupnou segmentací řezů	48
5.1.1	Reprezentace kontury	48
5.1.2	Energetický model	54
5.1.3	Optimalizace	58
5.1.4	Heuristika	64
5.1.5	Algoritmus	65
5.1.6	Měření objemu počítáním voxelů	65
5.2	Rekonstrukce modelováním B-Spline plochy	65
5.2.1	Definice B-Spline plochy	67
5.2.2	Optimalizace	68
5.2.3	Algoritmus a konvergence	68
5.2.4	Měření objemu z ohraničující plochy	69
5.2.5	Převod hraniční reprezentace na objemovou	71
5.3	Porovnání	71
6	Měření a výsledky	73
6.1	Použitá data	74
6.2	Měření ledvin	74
6.3	Měření sleziny	76
6.4	Výsledky segmentace a rekonstrukce	76
6.5	Rychlost, přesnost a konvergence	77
6.6	Diskuse	80
7	Závěr	85
A	Implementace	87
A.1	Reprezentace dat	87
A.2	Lékařská data	88
A.3	Uživatelské rozhraní	88
A.4	Vizualizace	88
A.5	Paralelizace	89
B	Obsah DVD	92

Seznam obrázků

1.1	Anatomický atlas - horní část břicha zepředu	13
2.1	Ukázka dat počítačové tomografie	15
2.2	Nehomogenita voxelů a interpolace	16
3.1	Prahování	19
3.2	Deformovatelné modely	20
4.1	Chyby a problémy aktivních kontur	26
4.2	Balónkový model	27
4.3	Ukázka nadrovin v reprezentaci level-sety	32
4.4	Třénovací množina ASM	39
4.5	Diagram znázorňující průběh algoritmu	44
5.1	Schéma procedury rekonstrukce orgánu postupnou segmentací řezů	49
5.2	B-Spline báze	50
5.3	Příklady B-Spline křivek	52
5.4	Vkládání uzlových bodů do B-Spline křivky	53
5.5	Křížení B-Spline křivky	54
5.6	Hranová energie zohledňující směr gradientu	55
5.7	Vliv polohy křivky na energii oblasti	56
5.8	Vliv délky úseků B-Spline křivky na celkovou hladkost	57
5.9	Snižování gradientu	58
5.10	Konvergence optimalizační metody	63
5.11	Alternativní konstrukce optimalizačního kroku	64
5.12	B-Spline plocha	67
5.13	Inicializace B-Spline plochy	69
5.14	Řez plochy rovinou	72
6.1	Statistika oblasti	75
6.2	Fáze nasycení kontrastní látkou	76
6.3	Výsledek rekonstrukce ledviny	77
6.4	Postupná segmentace ledviny	78
6.5	Rekonstrukce sleziny modelováním B-Spline plochy	78
6.6	Konvergence aktivních kontur s konstantním krokem	78
6.7	Konvergence aktivních kontur s adaptivním krokem	79

6.8	Dopad propagace v algoritmu postupné segmentace řezů	83
7.1	Rekonstrukce orgánů břišní dutiny	86
A.1	Laboratorní rozhraní - segmentation	89
A.2	Srovnání rychlostí paralelní a sériové verze algoritmu	90
A.3	Srovnání rychlosti podle OS a CPU	91

Seznam tabulek

6.1	Výsledky měření - pravá ledvina	80
6.2	Výsledky měření - slezina	81
6.3	Porovnání výsledků při adaptivním a konstantním kroku	81
6.4	Výsledky konvergence s různým prahem	82
6.5	Porovnání měření kontrastní a nativní série	82
6.6	Invariance měření vůči statistice	83
6.7	Invariance měření vůči vstupu	84

Seznam algoritmů

1	Level-sets - segmentace	34
2	PDM - zarovnání trénovacích tvarů	40
3	Active Shape Model - algoritmus rekonstrukce tvaru	41
4	Rekonstrukce orgánu postupnou segmentací řezů.	66
5	Rekonstrukce orgánu modelováním plochy.	70
6	Získání objemové reprezentace z hranice reprezentované plochou.	72

Název práce: *Měření objemu v 3D datech*

Autor: *Václav Krajíček*

Katedra (ústav): *Kabinet software a výuky informatiky*

Vedoucí diplomové práce: *RNDr. Josef Pelikán*

e-mail vedoucího: *Josef.Pelikan@mff.cuni.cz*

Abstrakt:

Diagnostika prostřednictvím počítačové tomografie, nebo magnetické rezonance je už delší dobu běžnou součástí komplexního lékařského vyšetření. Tato zařízení chrlí velké objemy dat, které musí být následně prostudovány lékařem odborníkem. Zde se nabízí velký prostor pro využití počítačů k podpoře diagnostiky, nebo vizualizaci.

Tato práce si bere za úkol změřit objem specifických orgánů v lidském těle jako jsou ledviny, nebo slezina. K měření objemu orgánu je nutné provést rekonstrukci objektu z trojrozměrných dat. Rekonstrukce je založena na segmentaci obrazových dat, která odděluje rekonstruované orgány od pozadí. Úloha segmentace je speciálně přizpůsobena pro tento typ dat, která jsou, exemplář od exempláře, charakteristická velkou podobností v obecném tvaru, ale také velkou různorodostí v drobných detailech. Byly zvoleny deformovatelné modely jakožto primární nástroj pro segmentaci trojrozměrného objemu. Deformovatelné modely jsou hlavním tématem této práce.

Programy měřící objem v lékařských datech existují, ale díky vzrůstajícímu výkonu počítačů bylo možné implementovat algoritmy, které zatím tento software neobsahuje.

Výsledkem této práce je systém, který segmentuje a počítá objem v lékařských datech. Zároveň byla provedena srovnávací měření odhadující chybu vzhledem ke skutečným hodnotám.

Klíčová slova: *segmentace, měření objemu, deformovatelné modely*

Title: *Volume measurement in 3D data*

Author: *Václav Krajíček*

Department: *Department of Software and Computer Science Education*

Supervisor: *RNDr. Josef Pelikán*

Supervisor's e-mail address: *Josef.Pelikan@mff.cuni.cz*

Abstract:

Medical diagnostics using computed tomography or magnetic resonance is a common part of complex medical checkup for a long time. These devices produce vast volumes of data, which must be analyzed by an expert physician. There is a big space for utilising computers to aid medical diagnosis, or visualization.

This work aims to measure volume of a specific organ in the human body, like kidneys or spleen. To measure volume of the organ it is necessary to reconstruct the object from three-dimensional data. Reconstruction is based on segmentation of the image data, which separates reconstructed organs from the background. The task of segmentation is specially adapted to this kind of data, which are, from specimen to specimen, characterized by big similarity in common shape, but also by big heterogeneity in tiny details. Deformable models were chosen as a primary tool for the segmentation of three-dimensional volumes. They are also the main topic of this work.

Programs measuring volume in medical data exist, but thanks to increasing performance of modern computers it is possible to implement new algorithms, which are not present in today's software tools.

The result of this work is a system which segments and measures volume in medical data. We also perform benchmarks, which expose an error of the measurement with respect to ground truth.

Keywords: *segmentation, volume measurement, deformable models*

Kapitola 1

Úvod

Přesná a rychlá diagnostika je důležitou a nepostradatelnou součástí práce lékaře. Dlouhou dobu patří radiologické metody mezi základní běžně dostupné diagnostické prostředky. Samotný rentgen oslavil svojí stovku už dávno. Rentgen a počítačová tomografie jsou však jen pouzí dva členové větší skupiny zobrazovacích metod v lékařství, které nám umožní nahlédnout do útroh lidského těla. Velikou výhodou těchto metod je, že jsou nedestruktivní a neinvazivní a téměř vůbec neohrožují život ani pohodlí pacienta.

Na druhou stranu, informace takto získané mají horší kvalitu než některé destruktivní metody. Rentgenové snímky jsou pro laika nečitelnou změť černobílých skvrn. K užitku mohou tedy být jen trénovanému oku odborníka. I znalosti a zkušenosti specialisty mají však své hranice. Možnosti a diagnostické postupy lze dále vylepšit a rozšířit s použitím moderních počítačů. Již mnoho let slouží výpočetní technika k vizualizaci dat všech typů zobrazovacích metod v lékařství. Obzvlášť poslední vývoj v oblasti grafických akceleratorů umožnil proměnit ve vyšetřovací stanici každý osobní počítač. Vizualizační prostředky slouží jak k výuce, k diagnostice a plánování chirurgických zákroků, tak i k prezentacím pro širokou laickou veřejnost.

V posledních letech se dostávají do povědomí odborné veřejnosti pojmy jako Počítačem podporovaná diagnóza (*CAD - Computer Aided Diagnosis*), nebo Počítačem podporovaná chirurgie (*CAS - Computer Aided Surgery*). Endoskopické operace jsou běžnou záležitostí, dokonce byly realizovány první operace pacientů z části řízené počítačem. Počítač měl v takových případech za úkol s využitím snímků ze zobrazovacích metod optimálně vybrat a asistovat při umístění kloubní náhrady tak, aby pacientova rehabilitace trvala co nejkratší dobu a měl co nejméně obtíží.

Vizualizace je nezbytným prostředkem procesu diagnostiky, protože člověk přijímá osmdesát procent všech informací ve formě viditelného světla. Lidský mozek je velmi dobrý ve vizuálním hodnocení obrazu. Je dokonce mnohem lepší než počítač. Dokáže těžit ze znalostí a zkušeností, i z oborově úplně jiné oblasti. Mozek analyzuje obraz nesystematicky, díky tomu může dojít k přehlédnutí. Navíc člověk podléhá únavě. V hodnocení každého obrazu je tedy výrazně subjektivní. Dnešní

běžně rozšířené zobrazovací metody generují ohromné množství dat. Není pochyb o tom, že jich bude víc, než by člověk s největším úsilím zvládl bez použití počítače analyzovat. Systémy CAD tvoří podpůrnou sílu a jaksi druhý názor, s jehož pomocí se daří zvyšovat spolehlivost a efektivitu vyšetření prostřednictvím některé ze zobrazovacích metod. Příkladem použití CAD může být podpora analýzy mamografických snímků. Systém zde funguje jako hloupý "spell checker", který označí všechny podezřelé případy a předloží je vyšetřujícímu lékaři k pozornému posouzení.

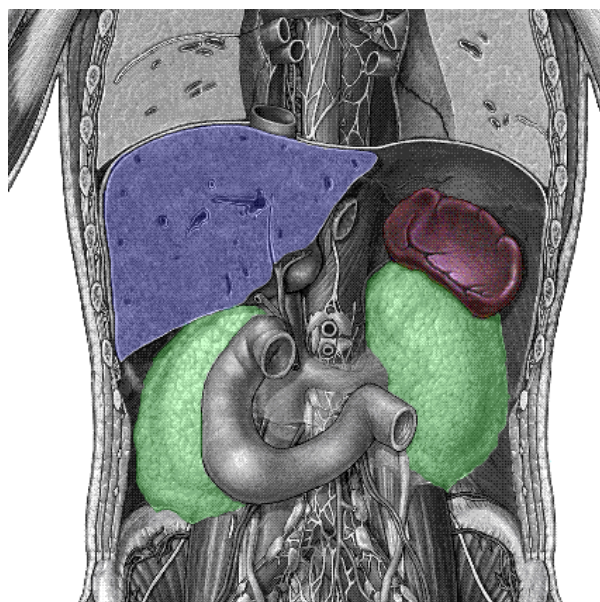
Podpora diagnostiky nemusí spočívat jen ve sledování optických (kvalitativních) vlastností snímku. K zjištění toho, co se s organismem v dané chvíli děje, mohou přispět i kvantitativní měření. Například měřením teploty lze detekovat, zda pacientův organismus zápasí s virovou nákazou. Měřením hustoty kostní dřene se dá odhalit stupeň osteoporózy a náchylnosti ke zlomeninám dlouhých kostí. Zvětšené uzliny signalizují probíhající onemocnění. Z množství protilátek v krvi, sedimentace, lze odhalit nabuzení imunitního systému při boji s konkrétní nemocí. Měření je i tématem této práce.

1.1 Cíle

Cílem této práce je implementovat algoritmy měřící objem orgánů v lidském těle z dat 3D zobrazovacích metod. Primárně budeme pracovat se snímky z počítačové tomografie. Výsledky však mohou být aplikovatelné i například na data z magnetické rezonance. Naši pozornost konkrétně zaostříme na měření objemu ledvin a případně dalších orgánů v břišní dutině, jako jsou slezina, játra nebo slinivka. Obtížnost úkolu stoupá v pořadí vyjmenovaných. Zmíněné orgány jsou zobrazeny na Obrázku 1.1.

Měření objemu je realizováno prostřednictvím rekonstrukce orgánu v objemových datech. Hlavním prostředkem rekonstrukce jsou segmentační algoritmy založené na deformovatelných modelech. Kvalita výsledků této segmentace a rekonstrukce, jejich přesnost, je posuzována při porovnání se segmentací provedenou člověkem. Na druhou stranu, důležitým požadavkem na celou proceduru je i rychlost. Pokud má být tedy algoritmus v praxi používán, musí být kompromisem mezi rychlostí, přesností a nezávislostí na vstupu uživatele. Proto se budeme snažit námi implementované algoritmy otestovat na reálné množině testovacích dat a určit relativní chybu měření.

Analýza lékařských dat a jejich segmentace je obtížná úloha. Je tomu tak, protože data jsou nepřesná díky nedokonalým, přesto stále přesnějším zobrazovacím metodám. Vlastnosti těchto metod také nezabrání vzniku různých artefaktů, které komplikují správnou interpretaci obrazové informace. Ani lidské vnitřnosti nejsou snadno rozpoznatelné objekty. I zdraví lidé vykazují velkou variabilitu tvarů svých orgánů a pacienti, kteří jsou nemocní, mohou mít nepředvídatelně deformované a poškozené vnitřnosti. Proto neexistuje univerzální algoritmus a každý vyzkoušený postup má své meze. Aby se v praxi dosáhlo lepších výsledků, tak si každá úloha žádá speciální algoritmus.



Obrázek 1.1: Anatomický atlas - horní část břicha zepředu. Jsou zde dobře vidět ledviny (zeleně). Pravá je částečně překrytá dvanácterníkem. Slezina (červeně) leží vlevo zhruba v úrovni ledvin a játra (modře) leží vpravo vepředu o něco výše. Na obrázku je vidět jen jejich část.

1.2 Struktura textu

V první části této práce (kapitola 2) bude vysvětlena situace okolo zobrazovacích metod v medicíně. Stručně budou představeny a charakterizovány tomografické metody. Seznámení se s touto látkou je klíčové pro prvotní směřování práce. Budou také blíže probrány specifika řešené úlohy.

Druhá část práce (kapitola 3 a 4) se zabývá segmentací obrazových dat. Blíže se zaměřuje na jednu skupinu algoritmů - algoritmy založené na deformovatelných modelech. Budou zde probrány nejdůležitější myšlenky, principy a z nich vyplývající výhody a nevýhody. Jejich popisu je věnováno dost prostoru, protože z něho čerpá zbytek práce.

Ve třetí části (kapitola 5) této práce jsou popsány implementované algoritmy segmentace a rekonstrukce objektů z lékařských dat postavené na základě deformovatelných modelů. Bude vysvětleno, že rekonstrukce je přirozenou cestou ke změření objemu. K rekonstrukci používáme dva principiálně odlišné algoritmy.

Kapitola 6 obsahuje výsledky testů implementovaných algoritmů. Popisujeme zde testovací množinu a specifika jednotlivých případů. Jsou zde představeny jak výsledky rekonstrukce, tak výsledky měření objemu porovnané mezi sebou a se správným výsledkem ruční segmentace. Z těchto pozorování jsou učiněny závěry.

V kapitole 7 je pak shrnuta práce jako celek. Nakonec jsou uvedeny náměty k vylepšení a dalšímu směřování výzkumu.

V dodatku jsou pak shrnuty detaily popisované implementace. Je zde popsáno technické řešení, použité knihovny a rozhraní implementovaného software.

Kapitola 2

Zobrazovací metody v lékařství

Zobrazovací metody v lékařství jsou široký obor zahrnující techniky založené na měření různých forem fyzikálního vlnění (akustické, různá spektra elektromagnetického vlnění) a polí (magnetické, elektrické). Metody můžeme rozdělit na ty, které mapují:

1. funkci tkáně
2. strukturu tkáně

Mezi zobrazovací techniky mapující funkci tkáně patří metody nukleární medicíny jako jsou *pozitronová emisní tomografie* (Positron emission tomography, zkráceně PET), nebo *jednofotonová emisní tomografie* (Single photon emission computed tomography, zkráceně SPECT), které měří záření kontrastní radioaktivní látky aplikované pacientovi při vyšetření, které se vstřebávají v biologicky aktivních tkáních, nádorech, srůstajících kostech, nebo hojících se ranách.

Mezi základní metody mapující strukturu tkáně patří rentgen. Funguje na principu záznamu průchodu rentgenových paprsků tkáněmi lidského těla ze zářiče, takzvané rentgenky, na citlivý film. Výsledek je dvojrozměrný obraz, který je projekcí celého objemu mezi rentgenkou a filmem. Ztratíme tak prostorovou informaci.

Zaznamenat prostorovou informaci dokáží až *tomografické metody*. Zachycují 3D obraz snímaného objemu jako sérii rovnoběžných řezů, viz Obrázek 2.2a. Podle toho je nakonec odvozen i jejich název z řečtiny (tomos = plátek, graphia = psát). Vzdálenost mezi každými dvěma sousedními řezy je konstantní. Čím menší tato vzdálenost je a čím větší je rozlišení samotných řezů, tím přesnější je zachycená informace.

Mezi hlavní zástupce tomografických metod patří *axiální výpočetní tomografie* (Computed axial tomography, zkráceně CAT, nebo CT) a nebo také *magnetická rezonance* (MR). Běžně používané lékařské CT vytváří série řezů, které jsou kolmé k ose lidského těla. Běžná MR dokáže zachytit řezy v libovolném směru. Podstatný rozdíl je také ve fyzikálním principu, na kterém CT a MR pracují. CT snímá rentgenové paprsky po průchodu lidským tělem a MR měří magnetické pole vybuzených vodíkových atomů.



Obrázek 2.1: Ukázka dat počítačové tomografie. Vlevo je snímek jednoho řezu. Uprostřed je v detailu vidět hvězdicový artefakt. Vpravo jsou hrany nalezené hranovým detektorem. Pro detekci hran se jako nejlepší z jednoduchých metod ukázala konvoluce s vertikálním a horizontálním Sobelovým filtrem velikosti 3.

2.1 Počítačová tomografie

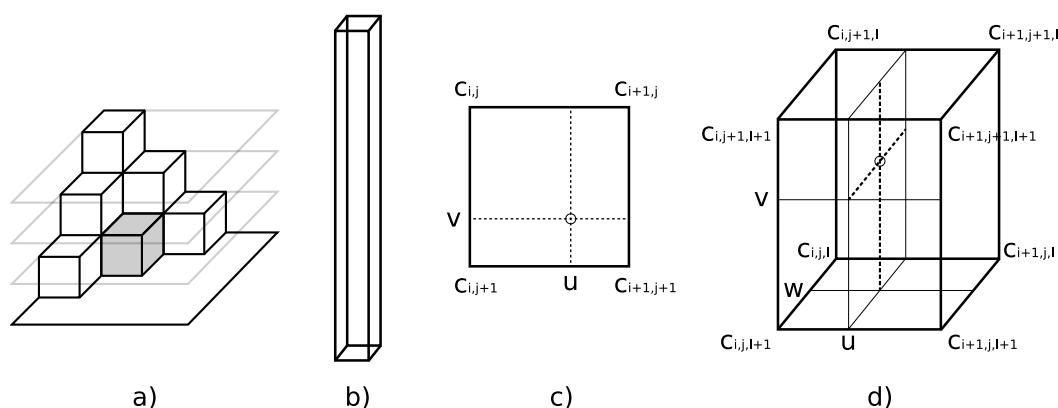
V této práci budou použity hlavně snímky z CT. Příklad je vidět na Obrázku 2.1. Z tohoto důvodu se krátce zmíníme o jejich vlastnostech. Na rozdíl od jednoduchého rentgenu jsou data z CT prostorově normovaná. Jsou přesně známé vzdálenosti bez ohledu na polohu pacienta uvnitř zařízení. Normované jsou i hodnoty výsledné obrazové funkce, které přesně odrážejí hustotu tkáně. Takovou vlastnost nemá žádná jiná zobrazovací metoda. Pro hodnoty obrazu existuje takzvaná Hounsfieldova stupnice, pojmenovaná podle konstruktéra prvního počítačového tomografu Godfrey Newbold Hounsfielda. Obrazová funkce nabývá na této stupnici hodnot -1000 až 3095 HU (Hounsfield unit) a je tedy uložena ve 12 bitech. Voda má hodnotu 0 HU, vzduch -1000 HU a kostní tkáň 600 až 1000 HU. Podle měření hustoty vody a vzduchu se počítačový tomograf kalibruje.

CT snímkování není dokonalé a obraz vykazuje některé chyby, se kterými je třeba počítat při dalším zpracování. Chyby, které se zde vyskytují, jsou

1. Partial volume efekt
2. Hvězdicový artefakt
3. Šum
4. Rozmazání pohybem

Partial volume efekt je způsoben zachycením obrazu ve větších objemových elementech, než jaké jsou nejmenší detaily v obraze. Například malé struktury jako žilky a okolí dotyku dvou orgánů se zprůměrují do jediné hodnoty. Kvůli tomu nemají snímky ostré hrany, jak ukazují obrázek 2.1.

Hvězdicový artefakt je způsoben nepřesností metody zpětné projekce výpočtu příčného řezu. Vzniká v místech s velkým rozdílem intenzity sousedních oblastí. Tato chyba poškozují velkou část okolního obrazu.



Obrázek 2.2: Nehomogenita voxelů a interpolace. **a)** Pravidelná mřížka voxelů. **b)** Ukázka poměru stran voxelu v datech z CT. V některých případech je poměr základny a výšky až 1:10. **c)** Bi-lineární interpolace na 2D obrázku počítá hodnotu ze čtyř nejbližších sousedů. **d)** Trilineární interpolace v 3D datech počítá hodnotu až z osmi sousedů.

Šum je klasický problém ve zpracování obrazu a platí pro něj známá pravidla. Brání v detekci hran. Na druhou stranu jeho redukce vede ke ztrátě hranové informace. Větší rozlišení zvýší subjektivně podíl šumu a malé rozlišení vede k výše popsanému partial volume efektu.

Rozmazání pohybem je způsobeno tím, že snímací procedura trvá nějakou dobu. I když je to jen několik vteřin, není možné úplně zastavit veškerý pohyb v lidském těle. Pacient může zadržet na krátkou dobu dech, ale nezastaví tlukot srdce, nebo peristaltické pohyby. Samozřejmě, díky technickému vývoji se průměrná doba snímání tomografu pořád snižuje.

U tomografie je většinou vzdálenost sousedních řezů jiná, a často mnohem větší, než rozměr obrazového elementu řezu. Série takových řezů představuje značně nehomogenní síť voxelů. Proto se nedají například zkonstruovat stejně kvalitní řezy v libovolném směru. Síť voxelů zachycuje původní obrazovou funkci v diskrétních krocích. Pokud chceme znát hodnotu funkce i mezi těmito kroky, musíme ji interpolovat z hodnot okolních bodů mřížky. Nejjednodušší interpolace hodnotou nejbližšího souseda vytvoří schodovitou aproximaci obrazové funkce. Lineární interpolací z hodnot několika sousedů získáme o něco lepší střechovitou aproximaci. Situace je znázorněna na Obrázku 2.2.

2.2 Zobrazení tomografických snímků

Stupnice 4096 hodnot intenzity se nedá zobrazit na běžných zobrazovacích zařízeních. Člověk sám není schopen rozlišit mezi tolika úrovněmi šedi. Proto je nutné mapovat tuto stupnici na tradiční 8bitovou, 256hodnotovou stupnici šedi. Mapování se realizuje prostřednictvím takzvané *Look-Up-Table* (LUT). Nejjednodušší LUT lineárně mapuje jeden souvislý úsek, takzvané okénko, Hounsfieldovy stup-

nice na stupnici šedi. Hodnoty před tímto okénkem se zobrazí na černou barvu a hodnoty za tímto okénkem na bílou. V praxi okénko definují pouhé dvě hodnoty, střed a poloměr.

Série snímků z CT představuje objemová data. Je mnoho způsobů, jak zobrazit objemová data, ale v současné době většina grafických systémů pracuje efektivně s hraniční reprezentací a s objekty reprezentovanými sítí trojúhelníků. Často se proto používá převod objemových dat na hraniční reprezentaci algoritmem *marching cubes*, nebo vytvářením hraniční reprezentace *napojováním kontur* objektů v jednotlivých řezech. Na druhé straně spektra zobrazovacích metod stojí raytracing. Možné jsou i různé hybridní přístupy k zachycení průhlednosti a vrstevnatosti objektů, takzvané metody přímého renderování objemu (*Direct volume rendering*).

2.3 Zpracování a analýza dat

Jediná série tomografických dat obsahuje velké množství informace. Její prozkoumání zabere člověku řádově více času než prohlédnutí snímku z rentgenu. Cílem počítačového zpracování je extrahovat a vizualizovat důležité informace z mnoha snímků, a usnadnit tak člověku zdlouhavou analýzu. Složitý úkol se dá rozdělit na několik obecných a nezávislých úloh.

Registrace je úloha nalezení mapování mezi dvěma i více množinami dat, kdy každá množina obsahuje různé informace a jejich fúzí získáme lepší a na informaci bohatší obraz. Obvyklé jsou například registrace CT a MR, nebo CT a SPECT.

Segmentace je rozdělením obrazu na logické celky. Může jít o nejjednodušší členění obrazu na popředí a pozadí. Segmentace je nejčastější úloha prováděná na lékařských snímcích, protože tvoří první krok při řešení složitějších úloh, jako je klasifikace, nebo rekonstrukce.

Klasifikace pojmenovává a určuje nalezené objekty v obraze. Klasifikaci můžeme provádět na různé úrovni, od pixelů přes tvary až po celé obrazy.

Rekonstrukce lékařských objemových dat vytváří zpětně přesnou geometrickou reprezentaci objektů v obraze.

2.4 Měření objemu

Pravě rekonstrukce je klíčová pro měření objemu orgánů. Pokud se podaří rekonstruovat měřený orgán O a určit, které voxely jsou jeho součástí (patří do množiny M_O), pak jeho objem je dán vztahem

$$\text{Volume}(O) = |M_f| \times \text{objem_voxelu}$$

Kapitola 3

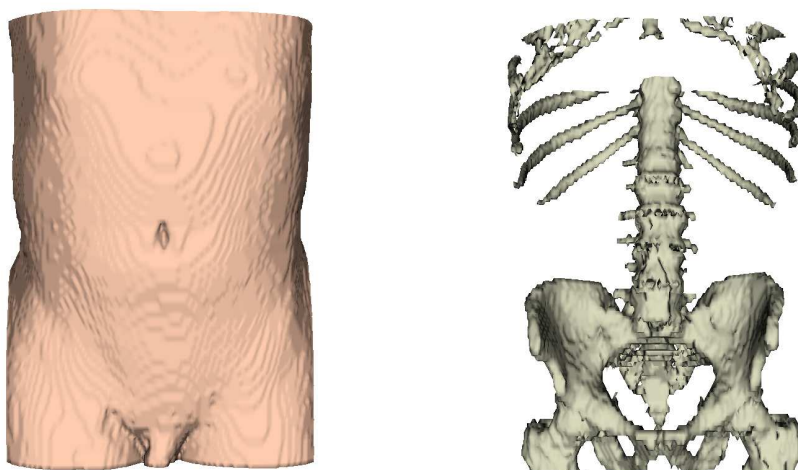
Metody segmentace

Segmentace lékařských dat je specifická úloha v oblasti počítačového vidění, proto jsou pro ní některé algoritmy vhodnější než jiné. V posledních dvaceti letech bylo k segmentaci dat z počítačové tomografie, nebo magnetické rezonance, použito mnoho různých metod. V podstatě každou metodu segmentace obrazu lze upravit a s jistým úspěchem použít pro segmentaci CT, MR, sonografických snímků, nebo digitalizovaných rentgenových snímků (RTG). Stručný přehled těchto metod nabízí například [27], nebo [31]. Pro každou specifickou úlohu je potřeba jiný, případně jinak parametrizovaný algoritmus. Univerzální algoritmus pro segmentaci neexistuje. V praxi se dá vyzorovat používání tří hlavních přístupů: Prahování, Narůstání oblastí a Deformovatelné modely.

3.1 Prahování

Prvním přístupem jsou metody založené na prahování (thresholding). Prahování je nejstarší a nejjednodušší metoda segmentace. Tyto metody se často používají při ruční segmentaci nebo vizuální analýze snímků. Zde se spoléhá na vlastnosti zobrazovací metody, která přiřadí každé tkáni takovou intenzitu, jakou má tato tkáň hustotu. Proto se také ve spojitosti s lékařskými snímky z CT hovoří o hodnotě obrazové funkce jako o denzitě (z angl. density - hustota). Prahování staví na jednoduché myšlence, že segmentovaný objekt má celý stejnou hustotu odlišnou od hustoty pozadí, tedy v obraze se projevuje jako oblast o stejné intenzitě odlišné od intenzity pozadí. Výběrem elementů, jejichž hodnota intenzity přesahuje určitý práh, nebo spadá do intervalu mezi dvěma prahy, se získá množina elementů spadajících do segmentované oblasti. Ukázkou výsledku získaného jednoduchým prahováním je možné vidět na Obrázku 3.1.

Samotné prahování neprodukuje příliš kvalitní výsledky, zvláště pokud je obraz hodně zašuměný. Proto se prahování často kombinuje například s morfologickými operacemi, jako jsou dilatace a eroze, nebo bývá součástí nějakého složitějšího algoritmu. Metody automatického hledání prahu si berou za úkol nalézt práh analýzy distribuce intenzity v segmentovaném obraze. Prahování je globální metoda a jeden práh se použije pro všechny body obrazu. Existují i lokální, nebo hierarchické



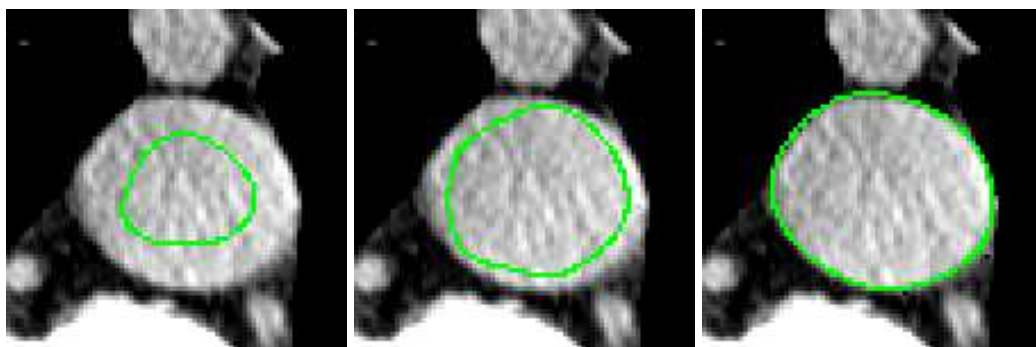
Obrázek 3.1: Ukázka prahování. Na obrázku vlevo jsou zobrazeny pouze voxely mající hodnotu odpovídající denzitě kůže a měkkých tkání, tj. -150-0 HU (Hounsfield Unit). Vpravo jsou voxely 1000-1500 HU, což zhruba odpovídá kostem.

metody prahování, které pro každou část obrazu použijí jiný práh. Nástroje pro prahování obsahuje většina systémů pro práci s lékařskými snímky, protože jejich používání je jednoduché a intuitivní. Vyčerpávající přehled prahovacích algoritmů skýtá práce [37]. Prahování v kombinaci s morfologickými operacemi použil i Pavel Campr ve své diplomové práci [5] na segmentaci kostry a těla z CT snímků.

3.2 Narůstání oblastí

Druhou skupinou jsou metody založené na narůstání oblastí (region growing). Základem těchto metod je záplavové procházení z počátečních zárodečných bodů, určených například uživatelem, nebo jiným způsobem v předchozích fázích analýzy. Elementy obrazu se procházejí jako graf, do šířky, přičemž sousední vrcholy jsou dány 4-,8- souvislostí v 2D případě, nebo 6-,18-,26- souvislostí v 3D případě. Největším problémem jsou zde zašuměná data. To je často případ CT snímků s větším rozlišením. Algoritmus si z principu nedovede poradit s přerušenými hranami, nebo proměnlivou intenzitou objektu, který má vysegmentovat. V jádru předpokládá, že součástí segmentovaného objektu je všechno, co leží blízko a má stejnou, či podobnou intenzitu. Během průchodu přidává do segmentované množiny navštívené vrcholy splňující jisté kritérium a přechází na jejich sousedy. V případě nesplnění kritéria se přes vrchol nešíří. Toto rozhodovací pravidlo se nazývá *kritérium homogeneity*.

Existují modifikace základního algoritmu. Například adaptivní narůstání oblastí [32] si dokáže poradit s nerovnoměrnou intenzitou. V porovnání s prahováním, kde je hlavním jmenovatelem pouze intenzita, je u narůstání oblastí brána v potaz i poloha. Narůstání oblastí má lokální charakter. Na podobném základu, rozhod-



Obrázek 3.2: Deformovatelné modely. Postupný vývoj z výchozího tvaru až po výslednou konturu.

vacím kritériu, jsou postaveny i metody slučování a dělení oblastí, nebo metody využívající transformaci rozvodí (watershed).

3.3 Deformovatelné modely

Poslední důležitou a zároveň nejzajímavější skupinou jsou metody založené na deformovatelných modelech (deformable models). Pro tyto algoritmy je charakteristický přístup *shora-dolu*, čímž se podstatně odlišují od tradičních přístupů k počítačovému vidění. Častějším postupem je nejprve extrahovat primitiva a poté se je pokusit slepit ve vyšší celky. Deformovatelné modely začnou prvotním odhadem celého tvaru, který pak porovnávají a pod vlivem fyzikálně definovaných sil upravují tak, aby lépe odpovídal segmentovanému obrazu. Princip segmentace je realizován například tak, že deformovatelný model popisuje tvar uzavřenou křivkou, nebo plochou, která odděluje segmentovanou oblast uvnitř od pozadí vně, jak je vidět na Obrázku 3.2.

Obecně lze deformovatelné modely použít i k hledání hran, rozpoznávání vzorů, registraci a jiným úlohám. Oproti prahování a narůstání oblastí pracují deformovatelné modely explicitně i s tvarem segmentované oblasti. Tyto metody jsou výpočetně náročné, ale mají potenciál vytvořit plně automatické systémy pro segmentaci lékařských dat. Jejich velkou výhodou je, že v sobě dokáží integrovat různé množství apriorních informací. Typickým příkladem je segmentace pomocí šablon (template matching), kdy algoritmus začne výpočet se znalostí reálného exempláře, nebo exemplářů, třídy objektu, který se snaží v předloženém obraze nalézt. Další text bude zaměřen především na vysvětlení deformovatelných modelů, zkoumání jejich vlastností a použitelnosti k řešení úloh z kapitoly 1.1.

3.4 Obecné vlastnosti algoritmů

Úloha měření objemu se realizuje na objemových, tedy třírozměrných datech. Důležité hledisko, podle kterého se dají všechny algoritmy rozdělit, je jejich

použitelnost k segmentování 3D dat, nebo dokonce univerzálnost, kdy na dimenzionalitě dat vůbec nezáleží. S rostoucí dimenzí dat roste vždy výpočetní náročnost každého algoritmu. Univerzální algoritmy mohou být užitečné při segmentaci a trackování záběrů ze sonografu, kdy k prostorovým dimenzím přibude ještě čas. Lékařské snímky z CT a MR dávají trojrozměrnou informaci, jsou to však série dvourozměrných obrázků. Takto získaná trojrozměrná mřížka není většinou úplně homogenní a to může ovlivnit i běh algoritmu. Algoritmy v kapitole 4 jsou vysvětleny převážně na 2D obraze. Jejich použitelnost pro 3D data bude explicitně zmíněna.

Mnoho algoritmů je často závislých na nastavení parametrů, na přesnosti uživatelského vstupu, nebo na uživatelské interakci v průběhu segmentace. Segmentace se dá podle stupně interakce dělit na: manuální, poloautomatickou, plně automatickou, nebo interaktivní. Od poloautomatických a automatických metod segmentace se očekává chytrý algoritmus, který má dost času na to, aby všechno spočítal. Od metod manuální a interaktivní segmentace se naopak očekává pohodlné uživatelské rozhraní, intuitivní význam parametrů použitých algoritmů a rychlá odezva při běhu výpočtu. Interaktivní algoritmy by se měly snažit redukovat míru subjektivity typickou pro manuální segmentaci. To znamená, že by měly maximálně snížit rozdíl mezi výsledky segmentace provedené různými uživateli, nebo dokonce mezi dvěma pokusy jednoho uživatele.

Kapitola 4

Deformovatelné modely

4.1 Active Contour Models - Snakes

Aktivní kontury, označované též jako hadi, patří do skupiny deformovatelných modelů. Původní algoritmus aktivních kontur spatřil světlo světa v roce 1988 v práci [25]. Stal se základem pro mnoho dalších děl o deformovatelných modelech. Aktivní kontury našly uplatnění v aplikacích trackování videa, stereovize, ale především v segmentaci lékařských dat. Následující text shrnuje jejich původní matematické odvození.

4.1.1 Diferenciální rovnice a jejich řešení

Aktivní kontury, stejně jako ostatní algoritmy deformovatelných modelů, jsou velmi precizně matematicky formulované. Základní myšlenka algoritmu je převedena na diferenciální rovnice. Algoritmus samotný pak v podstatě numericky řeší soustavu diferenciálních rovnic. Proto je užitečné uvést některé výchozí pojmy, na které se bude následující text odkazovat. Obyčejná diferenciální rovnice prvního řádu s počáteční podmínkou

$$F(y, x) = y', \quad y(x_0) = y_0$$

Návod k řešení těchto rovnic poskytuje diferenciální počet. Někdy je ale i řešení obyčejné diferenciální rovnice velice složité. Na počítači se navíc nedají analytické metody řešení diferenciálních rovnic efektivně implementovat. Většinou ani není možné splnit některé základní podmínky, jako je třeba spojitost obrazové funkce. Proto nám nabízí numerická matematika nejjednodušší algoritmus pro přibližné řešení diferenciálních rovnic, takzvanou Eulerovu metodu.

$$y(x+h) = y(x) + h \cdot F(y(x), x) \tag{4.1}$$

Metoda se snadno implementuje, má však některé nevýhody související s přesností a numerickou stabilitou. O něco lepší je implicitní Eulerova metoda

$$y(x+h) = y(x) + h \cdot F(y(x+h), x+h) \tag{4.2}$$

4.1.2 Odvození Active Contours

Idea Aktivních kontur je založena na minimalizaci energetického funkcionálu E_{snake}^* . Funkcionál E_{snake}^* je zobrazení z prostoru parametrických křivek do \mathbf{R} . Tento funkcionál popisuje "věrohodnost" výskytu konkrétní kontury v obraze. Aktivní kontura je tedy reprezentována jako parametrická křivka $\mathbf{v}(t)$. Funkcionál integruje věrohodnost každého bodu t kontury $\mathbf{v}(t)$, která je dána

$$E_{snake} = E_{int} + E_{image} + E_{con}$$

Člen E_{int} je takzvaná "vnitřní" energie formující konturu. Jeho cílem je zajistit hladkost ve smyslu nízkých hodnot prvních a druhých derivací $\mathbf{v}(t)$.

$$E_{int} = \alpha(t) \frac{\partial \mathbf{v}(t)}{\partial t} + \beta(t) \frac{\partial^2 \mathbf{v}(t)}{\partial^2 t} \quad (4.3)$$

Váhové funkce $\alpha(t)$ a $\beta(t)$ definují, jak má být aktivní kontura hladká a pružná. Jinak řečeno, tyto parametry mají vliv na tvar aktivní kontury v prudkých zatáčkách. Například je tak možné regulovat, zda se bude kontura "lepít" na hrany v obraze, nebo spíše zachová hladký tvar. Vnější energie E_{image} se skládá z více složek.

$$E_{image} = w_{line} E_{line} + w_{edge} E_{edge} + w_{term} E_{term}$$

Složky E_{line} , E_{edge} a E_{term} kvantifikují matematicky snadno popsatelné detaily obrazu. Původní E_{line} je nejjednodušší a definuje přitažlivost kontury k tmavším, nebo světlejším oblastem obrazu v závislosti na váhovém faktoru w_{line} . To je efektivní v případě tenkých hran mezi stejně barevnými oblastmi.

$$E_{line} = f(\mathbf{v}(t))$$

Zajímavějším členem je E_{edge} , který přitahuje kontury k hranám, tzn. k místům s vysokou hodnotou velikosti gradientu. Energetické minimum (největší gradient) může být méně výrazné, může se vyskytovat ve větší vzdálenosti od výchozí polohy kontury, nebo může být zastíněno šumem. Kvůli snazší konvergenci je dobré před výpočtem gradientu obraz rozmazat, nebo použít nějaký speciální hranový filtr.

$$E_{edge} = -|\nabla f(\mathbf{v}(t))|^2$$

Nejsložitějším termem je E_{term} . Jeho úkolem je detekovat ostré rohy, nebo konce hran. Toho se dosahuje zkoumáním křivosti kontury. Křivost se kvůli předpokládanému šumu počítá na rozmazaném obraze C . Obraz lze rozmazat konvolucí gausiánu G_σ s f , tzn. $C = G_\sigma(x, y) * f(x, y)$. Obraz C je převeden na obraz úhlů gradientu C , tedy $\theta = \tan^{-1}(C_x/C_y)$, kde $C_x = \frac{\partial C}{\partial x}$ a $C_y = \frac{\partial C}{\partial y}$. Vektor ve směru gradientu je pak definován $\mathbf{n} = (\cos\theta, \sin\theta)$. Normála gradientu, která vlastně odpovídá směru hrany, je $\mathbf{n}_\perp = (-\sin\theta, \cos\theta)$. Derivací θ podle normály \mathbf{n}_\perp zjistíme změnu směru gradientu, ve směru normály. Počítáním hodnoty této derivace na křivce $\mathbf{v}(t)$ se zjistí změna směru hrany v obraze podél kontury. Pokud kontura kopíruje hladkou, rovnou hranu je celková hodnota E_{term} malá.

$$\begin{aligned}
E_{term} &= \frac{\partial \theta}{\partial \mathbf{n}_\perp} = \frac{\partial^2 C / \partial \mathbf{n}_\perp^2}{\partial C / \partial \mathbf{n}} \\
&= \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{3/2}}
\end{aligned}$$

Teoreticky by bylo možné do E_{image} zahrnout i jiné termy podle libosti a požadavků řešené úlohy. Právě na pružnosti toho rámce staví mnoho dalších prací. Pokud je v literatuře na původní algoritmus aktivních kontur odkazováno, má se na mysli pouze term E_{int} s první a druhou derivací křivky a hranový term E_{edge} .

4.1.3 Optimalizace

Pro úplnost zbývá uvést optimalizační schéma z původního článku [25], podle kterého lze aktivní kontury implementovat. Úkolem je nalézt takovou křivku \mathbf{v} , aktivní konturu, která přes celou svou délku minimalizuje hodnotu E_{snake} . Provede se tedy minimalizace funkcionálu

$$E_{snake}^*(\mathbf{v}) = \int_0^1 E_{snake}(\mathbf{v}(t)) dt \quad (4.4)$$

Toto je úloha pro variační počet. Diferenciální počet se zabývá hledáním extrémů reálných funkcí, hodnot v definičním oboru reálných čísel. Variační počet se oproti tomu zabývá hledáním extrémů funkcionálů na jejich definičním prostoru, kterým jsou funkce. Pro nalezení extrému funkcionálu E_{snake}^* je nutnou podmínkou rovnost jeho variace (obdoba derivace reálné funkce) nule. Tato rovnost je ekvivalentní s takzvanou Euler-Lagrangeovou rovnicí, pro jejíž řešení jsou postačující následující podmínky, za předpokladu konstantních faktorů α a β .

$$\begin{aligned}
\alpha x_{tt} + \beta x_{tttt} + \frac{\partial E_{ext}}{\partial x} &= 0 \\
\alpha y_{tt} + \beta y_{tttt} + \frac{\partial E_{ext}}{\partial y} &= 0
\end{aligned}$$

Kde x_{tt} , y_{tt} , x_{tttt} a y_{tttt} jsou druhé a čtvrté derivace složek $\mathbf{v}(t)$ podle parametru křivky. Řešení nemusí, kvůli pouhé nutné podmínce, dát minimum. Věty variačního počtu hovoří o stacionárním bodu. Přesto je to dobrý výchozí předpoklad pro implementaci. Aby bylo možné toto schéma prakticky implementovat, převede se (4.4) do diskrétního tvaru.

$$E_{snake}^* = \sum_{i=1}^n E_{int} + E_{ext}$$

Derivace $x(t)$ a $y(t)$ v Eulerových rovnicích se aproximují konečnými diferencemi. Pro křivku $\mathbf{v}(t) = (x(t), y(t))$ rozdělenou na n bodů $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$ platí n

rovnice pro každou ze souřadnic x a y . Rovnice pro jeden bod \mathbf{v}_i , tedy dvě skalární rovnice, má tvar

$$\begin{aligned} \alpha_i(\mathbf{v}_i - \mathbf{v}_{i-1}) &- \alpha_{i+1}(\mathbf{v}_{i+1} - \mathbf{v}_i) \\ &+ \beta_{i-1}(\mathbf{v}_{i-2} - 2\mathbf{v}_{i-1} - \mathbf{v}_i) \\ &- 2\beta_i(\mathbf{v}_{i-1} - 2\mathbf{v}_i - \mathbf{v}_{i+1}) \\ &+ \beta_{i+1}(\mathbf{v}_i - 2\mathbf{v}_{i+1} - \mathbf{v}_{i+2}) \\ &+ (f_x(i), f_y(i)) = 0 \end{aligned} \quad (4.5)$$

Derivace $f_x(i)$ a $f_y(i)$ se spočítá buď analyticky, nebo numericky. Křivka je uzavřená a platí $\mathbf{v}_{n+i} = \mathbf{v}_i, i \in \{-1, 0, 1, 2\}$. Rovnice (4.5) se dá napsat v maticovém tvaru.

$$\begin{aligned} \mathbf{A}\mathbf{x} + \mathbf{f}_x(\mathbf{x}, \mathbf{y}) &= 0 \\ \mathbf{A}\mathbf{y} + \mathbf{f}_y(\mathbf{x}, \mathbf{y}) &= 0 \end{aligned}$$

Tato diferenciální rovnice se vyřeší implicitně-explicitní Eulerovou metodou. Dosazením do (4.1) a (4.2) a následným sečtením získáme

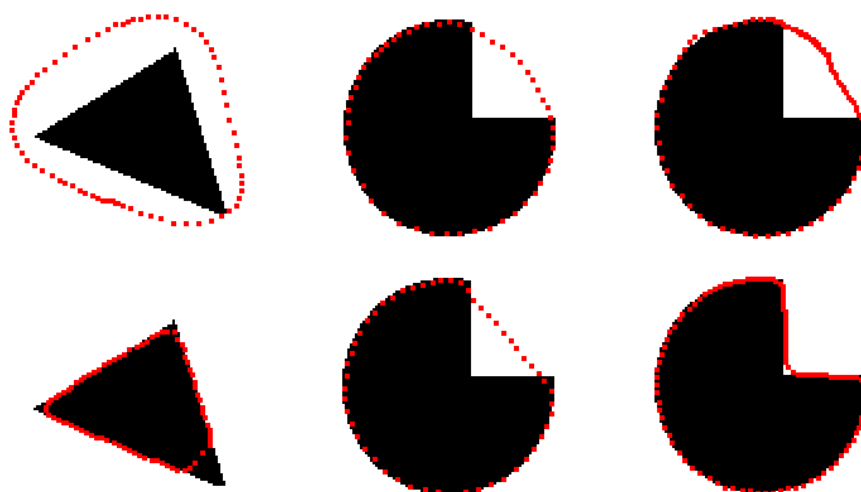
$$\begin{aligned} \mathbf{A}\mathbf{x}_t + \mathbf{f}_x(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) &= -\gamma(\mathbf{x}_t - \mathbf{x}_{t-1}) \\ \mathbf{A}\mathbf{y}_t + \mathbf{f}_y(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) &= -\gamma(\mathbf{y}_t - \mathbf{y}_{t-1}) \end{aligned}$$

Interní energie se řeší dle schématu implicitní Eulerovy metody, kdy je derivace vyjádřena násobením aktuálního kroku maticí A . Externí energie se počítá explicitní Eulerovou metodou pouhým přičtením derivace f . Úpravou se získá řešení této lineární soustavy rovnic

$$\begin{aligned} \mathbf{x}_t &= (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\mathbf{x}_{t-1} - \mathbf{f}_x(x_{t-1}, y_{t-1})) \\ \mathbf{y}_t &= (\mathbf{A} + \gamma\mathbf{I})^{-1}(\gamma\mathbf{y}_{t-1} - \mathbf{f}_y(x_{t-1}, y_{t-1})) \end{aligned} \quad (4.6)$$

Matice $\mathbf{Q} = (\mathbf{A} + \gamma\mathbf{I})$ je 5-diagonální a její inverze také. Inverzní matici \mathbf{Q}^{-1} lze snadno získat například LU rozkladem. Stačí \mathbf{Q}^{-1} spočítat jednou na začátku běhu algoritmu.

V přehledu nebyl úmyslně zmíněn poslední term E_{con} . Umožňuje uživatelský vstup do výpočtu algoritmu, kdy uživatel zadá klíčové body, na které se mají kontury uchytit, nebo oblasti, kterým se mají vyhnout. Mohou být takto definovány omezující podmínky dané apriorní znalostí konkrétní úlohy, okraje obrazu, začátek a konec křivky, atd. Algoritmus aktivních kontur je jednou ze základních metod použití deformovatelných modelů. Výklad ukazuje dvě oddělené fáze návrhu, které jsou společné pro většinu deformovatelných modelů. První je formulace kritérií, požadavků a parametrů a druhá je optimalizační výpočet. Hledání optimálních křivek ve smyslu minimální energie je úloha známá z fyziky, například Optiky, nebo Obecné teorie relativity. Eulerovy rovnice a variační počet patří mezi základní prostředky k jejich řešení.



Obrázek 4.1: Ukázka chyb při běhu základního algoritmu aktivních kontur. Vlevo je vidět, že se úseky s řídkým rozložením bodů neudrží na ostrých výčnělcích. Uprostřed se dá pozorovat, jak úsek nepřilne dokonale k hranám díky 'vyhlazujícímu' vlivu funkcionálu E_{int} . Vpravo je výsledek lepší z důvodu hustšího rozložení bodů ve vykrojené oblasti.

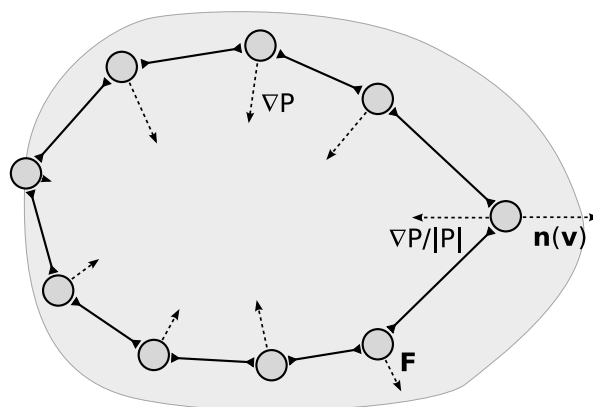
4.1.4 Vlastnosti

Implementace dané metody a její použití na umělých datech ukazuje na některé podstatné nevýhody. První z nich způsobuje, pro počítače charakteristická, diskretizace. Na počátku musí být znám počet bodů, pro které se bude řešit soustava rovnic. Pro body na konturách se částečně vžilo označení 'snaxel' (snake pixel). Snaxelů musí být tolik, aby byl výpočet dostatečně přesný, ale ne mnoho, aby nebyl příliš časově náročný. Asymptotická složitost algoritmu je přitom řádu $O(\text{pocet_kroku} \cdot n^2)$. Pokud se tvar křivky v průběhu iterace příliš změní a dojde například k natažení a v nataženém úseku dojde k zřídnutí bodů, dochází k chybám, jak ukazuje Obrázek 4.1. K chybám dojde i při zhuštění bodů. Částečně se dá těmto problémům vyhnout, pokud se křivka čas od času reparametrizuje. To je nepříjemné, protože si to vynutí znovu spočítání matice Q^{-1} .

Dalším problémem je stanovení vhodných hodnot parametrů α , β , γ . K tomu všemu je možné definovat zvlášť α_i a β_i , pro každý bod na křivce. Většinou se to nedělá. Správná hodnota parametrů je závislá na typu úlohy a často se dá určit pouze experimentálně. Taková je realita u většiny deformovatelných modelů.

Důležitá je také počáteční poloha křivky. Výsledek je jiný pro každou počáteční polohu. I pro dvě podobné počáteční polohy se může podstatně lišit. Většina algoritmů deformovatelných modelů odvozených od aktivních kontur se proto snaží co nejvíce omezit závislost na počáteční poloze.

Posledním tématem je podmínka pro ukončení výpočtu a možnost uvíznutí v lokálním minimu. Výpočet se dá ukončit ve chvíli, kdy jsou změny poloh snaxelů



Obrázek 4.2: Balónkový model. Pohled na konturu jako na posloupnost hmotných bodů spojených pružinami.

dostatečně malé. Například ve sledovaném okénku posledních iterací je výchylka poloh menší než dané ε . Ale i z principu algoritmu se může stát, že řešení uvízne v lokálním minimu a ukončovací podmínka výpočet přeruší dřív, než se povede algoritmu z minima vyváznout.

4.2 Balónky

Balónkový model je prvním vylepšením základních aktivních kontur. V práci [10], kde je představen, se pohlíží na kontury jako na řetězce hmotných bodů spojených pružinami nulové délky. Na vyhlazující vliv funkcionálu E_{int}^* lze pohlížet jako na silové působení těchto pružin, viz Obrázek 4.2. Tuhost pružin je dána parametry, které zhruba odpovídají parametrům α, β z rovnice (4.3).

Autor také odhalil, že v maticovém vyjádření 4.6 se skrývá konvoluce posloupnosti sousedních bodů s filtrem, který má omezený support. Stačí si uvědomit, že i -tý řádek matice Q^{-1} obsahuje nenulové hodnoty ve sloupcích $i-2$ až $i+2$. V případě, že platí $\beta_i = \beta, \alpha_i = \alpha \forall i$, pak má filtr největší hodnotu v i -tém sloupci postupně klesající v $(i \pm 1)$ -tém a $(i \pm 2)$ -tém sloupci. V jednom kroku výpočtu tak dochází k vyhlazení váženým součtem sousedních bodů. To se dá pozorovat na tom, že samotný vyhlazující vliv E_{int} kontrahuje všechny body křivky do jednoho středu.

Pokud bychom se snažili detekovat nějaký tvar v obrázku pomocí uzavřené aktivní kontury, musíme konturu inicializovat okolo předpokládaného tvaru a očekávat, že se při smršťování zachytí na hranách tvaru. Balónkový model přišel s ideou, že pro většinu aplikací by byl lepší opačný směr pohybu kontury v průběhu výpočtu. Místo smršťování použít rozpínání. Odtud také název balónkový model. Definuje "nafukovací" sílu zahrnující složku, která působí kolmo na křivku.

$$\mathbf{F} = k_1 \cdot \mathbf{n}(\mathbf{v}) - k \frac{\nabla P}{\|\nabla P\|}$$

Kde ∇P je vektor pohybu bodu podle původního algoritmu. Tato síla je nor-

malizovaná, takže pohyb dál nezávisí na velikosti gradientu, ale pouze na zvolené konstantě k . Vektor \mathbf{n} je kolmice na konturu \mathbf{v} v daném bodě. Poměrem a velikostí k_1 a k se reguluje rychlost a směr pohybu (nafukování, splaskávání). Vektor \mathbf{F} pak určuje změnu polohy bodu křivky. Je nutné průběžně reparametrizovat křivku. Buď pravidelně po každých h krocích, nebo až vzdálenost nějakých dvou sousedních bodů překoná dané δ .

Standardní hranový operátor daný velikostí gradientu je nahrazen Cannyho-Derichého hranovým detektorem. Cannyho detektor má schopnost non-maximální suprese, která zvýrazní gradient tvořící souvislé hrany a potlačí šum (suprese). Umožňuje získat kvalitnější hrany, na kterých se rozpínající kontury modelu snáze zastaví.

Aby bylo dosaženo rovnovážného stavu, je nutné vhodně nastavit konstanty k_1 a k . Během výpočtu hrají roli časového kroku Eulerovy metody. Jelikož vektory, které násobí, mají jednotkovou velikost, samy by neměly přesáhnout velikost strany jednoho pixelu. Tak se předejde přeskočení některých tenkých hran, nebo vyskočení z nalezeného minima.

V navazujícím článku [11] jsou uvedeny i příklady segmentace 3D objektů v objemových datech. Navíc je zde prezentována metoda konečných prvků pro řešení variační rovnice původních aktivních kontur a balónkového modelu. Ta pomáhá bojovat s velkou výpočetní složitostí, která se naplno projeví až při počítání v 3D datech.

Balónkový model byl už přímo navržen pro segmentaci lékařských dat. Počítá s vývojem uzavřené křivky, která je v prvním kroku umístěna dovnitř segmentovaného objektu. Díky tomu není výpočet tolik závislý na počáteční poloze. Výchozí křivka může být umístěna, s téměř stejným výsledkem, na libovolném místě uvnitř objektu. Křivku je však nutné často reparametrizovat. Výsledek výpočtu také stále hodně závisí na konstantách k_1 a k . Navíc algoritmus dobře segmentuje pouze objekty s jednoduchou souvislou topologií, víceméně konvexního tvaru.

4.3 Další vylepšení pro Aktivní kontury

Základní algoritmus aktivních kontur je možné vylepšit v každé jeho problematické oblasti a přizpůsobit každé konkrétní aplikaci, jak ukazují následující příklady.

4.3.1 Modely zohledňující topologii - T-Snakes

Problém segmentace složité topologie deformovatelnými modely je řešen v práci [30]. Autor si rozdělil obrazový prostor na čtvercovou síť a dělením každého čtverce diagonálou v jednom směru získá simplexovou síť. Simplexová síť má mnoho výhod, protože existuje její dobrá algebraická formalizace. Mezi její zajímavé vlastnosti patří, že pozice bodu v simplexové buňce lze vyjádřit jednoznačným způsobem pomocí vrcholů simplexu. Tím pádem je i jednoznačná reprezentace kontur, ploch a složitějších struktur. Jeden simplex je nejmenší jednotkou prostoru. Jeho velikost odpovídá velikosti nejmenších detailů, které algoritmus umí zachytit.

Konturu definuje posloupnost simplexů, kterými prochází. Přitom je jednoznačně určeno, kterou stěnou simplexu vstupuje dovnitř a kterou stěnou vystupuje ven. Jedním simplexem může procházet maximálně jedna kontura. Iterativní algoritmus v každém kroku pohne s konturou a vytvoří novou posloupnost simplexů, přičemž je schopen detekovat změny topologie. Například kolize, nebo dělení. Simplexovou síť je možné přirozeně sestavit v i prostoru vyšší dimenze, aniž by se zkomplikovaly algoritmy pro procházení hranice a detekci topologických změn.

4.3.2 Segmentace série řezů

V případě segmentace série snímků, které dohromady tvoří trojrozměrnou mřížku, je možné použít předchozí algoritmy pro segmentaci každého snímku v sérii. Z vysegmentovaných řezů se zrekonstruuje 3D objekt. Omezením segmentace pouze na řezy se bohužel ztrácí mnoho informace, která je obsažena napříč snímky, například v hranách vedoucích mimo rovinu snímků. Jedno schéma, jak počítat v řezech odděleně a zároveň neignorovat informaci obsaženou v jiných rovinách, je popsáno v práci [9]. Autor nechává postupně zkonvergovat uzavřené křivky v jednotlivých řezech série, poté změni orientaci osy a vygeneruje novou sérii řezů. V této sérii zrekonstruuje z původní množiny uzavřených kontur nové kontury ležící v rovinách snímků nové série. Při rekonstrukci se musí řešit situace, kdy vznikne více uzavřených kontur, nebo i kontury uvnitř kontur (díry). Opět se spustí algoritmus aktivních kontur na každém snímku zvlášť. Změna orientace, přepočítání a opětovná konvergence aktivních kontur se opakují tak dlouho, dokud se celkový tvar získaný spojením oblastí ze všech snímků významně mění.

4.3.3 Aktivní plochy

Kontura je dána posloupností vrcholů spojených úsečkami, kdy má každý vrchol uzavřené kontury dva své sousedy. V případě 3D plochy je to mírně složitější. Plochu lze aproximovat trojúhelníkovou sítí. Takovou dostatečně jemnou trojúhelníkovou síť je pak možné použít při segmentaci v 3D datech. Musí se navíc vyřešit zjemňování a hrubnutí plochy během reparametrizace. Jedna metoda aktivních ploch včetně výpočetního schématu je popsána například v [29].

4.4 Geodesic contours

Drobných vylepšení původního algoritmu aktivních kontur, jako jsou ty z předchozí kapitoly, se dá nalézt ještě mnohem více. Nemění nic na základní myšlence minimalizace energetického funkcionálu. Zajímavý ideový přerod přinesla práce [6]. Autor nejprve zdůraznil důležitost parametrizace křivky v původním modelu. Velice totiž záleží, jak rychle je křivka v integrálu (4.4) probíhána. Pro různé způsoby probíhání je možné, že vyjdou jiné výsledky. Souvisí to s nedostatkem zmíněným v popisu implementace základního modelu, který se týkal hustoty bodů. Problém se dá matematicky vyřešit tak, že se křivkový integrál parametrizuje "podle oblouku".

Reparametrizace je tímto zahrnuta přímo do modelu a není uměle řešena až během výpočtu.

$$\begin{aligned}\ell(T) &= \int_0^T \sqrt{x'(t)^2 + y'(t)^2} dt \\ L(l) &= \ell^{-1}(l), \quad l \in [0, length] \\ E_{edge}^* &= - \int_0^{length} |\nabla f(\mathbf{v}(L(l)))| dl\end{aligned}$$

Nejdůležitější poznatek je, že teoretický problém minimalizace funkcionalu se dá převést na nalezení geodetických křivek. *Geodetická křivka* je definovaná jako nejkratší (lokálně) cesta mezi dvěma body. Tento pojem je převzat z geodezie, vědy, která se zabývá měřením zemského povrchu. Navržený algoritmus tedy staví na hledání geodetických křivek. Alternativně lze na geodetickou křivku pohlížet jako na spojnicu dvou bodů, která minimalizuje váženou vzdálenost. Tato úvaha umožnila spojit termy vnitřních a vnějších sil, E_{int} a E_{image} , do jednoho, který bere v úvahu jak vlastnosti obrazu (hrany), tak vlastnosti křivky (hladkost, délka).

$$E_{geodesic}^* = - \int_0^1 \underbrace{|\nabla f(\mathbf{v}(t))|}_{\text{váha hrany}} \underbrace{\sqrt{x'(t)^2 + y'(t)^2}}_{\text{váha vzdálenosti}} dt \quad (4.7)$$

4.4.1 Optimalizace - Snižování gradientu

Matematická optimalizace je třída algoritmů, pomocí kterých se hledá minimum dané funkce $f : \mathbf{A} \rightarrow \mathbf{R}$. Jde o systematický výběr hodnot vedoucí co nejrychleji ke globálnímu minimu (obecně extrému). Tento výběr lze provádět mnoha způsoby, které odpovídají různým optimalizačním algoritmům. Jedním z takových algoritmů je například *snižování gradientu* (gradient descent). Snižování gradientu provádí výběr od počátečního prvku ve směru gradientu optimalizované funkce f .

$$x_{n+1} = x_n + \delta \frac{\partial f}{\partial x}$$

Snižování gradientu je použito k nalezení geodetických křivek. Počáteční prvek tvoří výchozí křivka v čase $q = 0$, $\mathbf{v}_0(t) = \mathbf{v}(0, t)$. K optimalizaci potřebný gradient je v [6] odvozen derivací (4.7).

$$\frac{\partial E_{geodesic}^*}{\partial \mathbf{v}} \approx \frac{\partial \mathbf{v}(q, t)}{\partial q} = g(f) \kappa \mathbf{n} - (\nabla g \cdot \mathbf{n}) \mathbf{n}$$

Funkce g je metrika vlivu gradientu v obraze. Platí pro ni, že $g(r) \rightarrow 0$ pro $r \rightarrow \infty$. Vektor \mathbf{n} je normála jednotkové velikosti a κ je křivost křivky \mathbf{v} . Tento model formalizuje přechod od minimalizace energie k hledání geodetických křivek. Zajímavější je reprezentace prostřednictvím takzvaných *level-setů*, ve které je tento model počítán, protože nabízí elegantní řešení mnoha problémů aktivních kontur.

4.5 Level-sets

Level-sets jsou numerické výpočetní schéma, které nalézají aplikace v mnoha oblastech. Používá se například v simulacích dynamiky kapalin, výpočtu animací, nebo ve vizuálních efektech. Metody level-setů jsou výsledkem mnohaleté práce a bádání v oblasti propagace rozhraní. Zde jde uvedena implementace level-setů pro hledání tvarů v obraze podle [1] a [28]. Hlavní výhodou level-setů je nezávislost výsledku na počáteční poloze a tvaru křivky. Level-sets umí segmentovat i velmi složité tvary, které obsahují díry, nebo mají hodně členitou hranici. Není problém segmentovat více objektů najednou. Díky těmto vlastnostem se dá této metody využít při segmentování videa, kde objekty v obraze mění topologii mezi jednotlivými snímky. Stejně výhodná je aplikace na lékařská data z CT a MRI. Princip segmentace level-sets se nejlépe ilustruje na dvojrozměrném obraze. Ale sám výpočetní rámec není dimenzí dat nijak omezen. Proto lze level-sets použít i pro segmentaci objemových dat. V textu, kde to je možné, se bude za obrazový prostor brát obecně prostor dimenze N . Zpracování data vyšší dimenze je možné samozřejmě výměnou za delší dobu výpočtu. Výpočetní složitost je jednou z nevýhod level-setových metod.

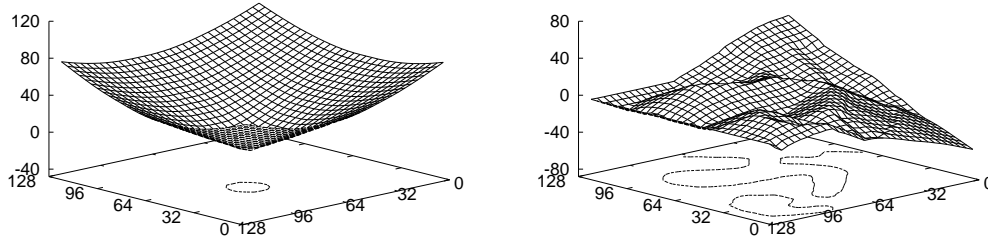
Jak tedy level-sets fungují? Na začátku je výchozí hladká uzavřená křivka $\mathbf{v}(0)$ v Eukleidovském prostoru. Funkce $\mathbf{v}(q)$ dává množinu křivek tak, jak se vyvíjejí v čase q z počáteční $\mathbf{v}(0)$ pod vlivem sil F . Body na křivce $\mathbf{v}(q)$ jsou dány funkcí $\mathbf{v}(q, t)$ pro $t \in (0, S)$. Až sem je situace stejná jako v předchozích algoritmech. Křivka může být reprezentována body a s těmi lze pohybovat v drobných diskretních krocích. To vede k řadě problémů popsanych v předchozím odstavci. Druhou možností je chápat $\mathbf{v}(q)$ pro pevné q jako podmnožinu nadroviny $\psi(\mathbf{x}, q) : R^N \rightarrow R$ danou rovnicí $\psi(\mathbf{v}, q) = 0$. Počáteční křivka je $\mathbf{v}(q = 0) = \{\mathbf{x} \mid \psi(\mathbf{x}, q = 0) = 0\}$. Takže se místo počítání samotné křivky \mathbf{v} v obraze počítá nadrovina ψ a křivka \mathbf{v} se z ní odečte až v momentě, kdy je jí potřeba zobrazit. Hranice objektu, křivka $\mathbf{v}(q)$, je *implicitně zadaná* výrazem $\psi(\mathbf{v}, q) = 0$. Bude na ni v textu odkazováno jako na *vrstevnici nulté hladiny* ψ . Stejně jako u původním aktivních kontur jde o vývoj funkce pod vlivem sil odvozených z analyzovaného obrazu a případně vnitřních vlastností, jako je křivost, nebo derivace. Hlavní otázkou je, jak reprezentovat nadrovinu $\psi(\mathbf{v}, q)$ a jak definovat síly, které s ní budou hýbat.

Na začátku se zavede $\psi(\mathbf{x}, q = 0)$ jako nejkratší vzdálenost bodu \mathbf{x} od křivky $\mathbf{v}(0)$. Pro $x \in R^N$ tedy platí

$$\psi(x, t = 0) = \pm d$$

Znaménko d značí, nachází-li se bod \mathbf{x} uvnitř (mínus), nebo vně (plus) uzavřené křivky ve vzdálenosti d . Ukázka nadroviny pro obrazová data dimenze $N = 2$ je vidět na Obrázku 4.3.

Odvození rovnice pro výpočet nadroviny ψ začne u definice $\mathbf{x}(q)$, $q \in [0, \infty)$ jako polohy bodu, na vrstevnici nulté hladiny funkce ψ závislé na čase. $\mathbf{x}(0)$ je bod výchozí křivky. Tato funkce se dosadí do rovnice nadroviny ψ a vyjde následující podmínka



Obrázek 4.3: Vlevo je nadrovina ψ na začátku segmentace. Vpravo je nadrovina ψ na konci běhu algoritmu. Je jasně vidět, že vnitřek objektu leží pod nulovou hladinou a hranice objektu je definována jako vrstevnice na hladině nula.

$$\psi(\mathbf{x}(q), q) = 0 \quad (4.8)$$

Jak je vidět, hodnota ψ se v $(\mathbf{x}(q), q)$ se v čase nemění. Derivací (4.8) za použití řetízkového pravidla dostáváme

$$\frac{\partial \psi}{\partial q} + \frac{\partial \psi}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial q} = \frac{\partial \psi}{\partial q} + \sum_{i=1}^N \frac{\partial \psi}{\partial x_i} \frac{\partial x_i}{\partial q} = 0 \quad (4.9)$$

Do této rovnice se dosadí výraz odvozený z obrazových dat. Nejlépe se vyjádří velikost rychlosti bodu nadroviny, tzn. změna jeho polohy v čase, ve směru normály. Označme ji F . Pro implicitně definovanou funkci, ve tvaru $f(x, y) = 0$, platí, že její normála (kolmice v bodě je) je dána gradientem ∇f . Tedy pokud je $\partial \mathbf{x} / \partial q$ změna polohy bodu \mathbf{x} v čase a $\nabla \psi$ normála ψ v bodě \mathbf{x} , pak velikost rychlosti ve směru normály je

$$F = |\partial \mathbf{x} / \partial q| \left(\frac{\nabla \psi}{|\nabla \psi|} \cdot \frac{\partial \mathbf{x} / \partial q}{|\partial \mathbf{x} / \partial q|} \right) = \frac{\nabla \psi}{|\nabla \psi|} \cdot \partial \mathbf{x} / \partial q$$

Ve skalárním součinu dvou jednotkových vektorů se skrývá kosinus, který říká, jak hodně se směr pohybu bodu \mathbf{x} v rovině podílí na pohybu ve směru normály. Tato velikost rychlosti se vyskytuje i ve vztahu (4.9).

$$\frac{\partial \psi}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial q} = \left| \frac{\partial \psi}{\partial \mathbf{x}} \right| \frac{\partial \psi / \partial \mathbf{x}}{|\partial \psi / \partial \mathbf{x}|} \frac{\partial \mathbf{x}}{\partial q} = \left| \frac{\partial \psi}{\partial \mathbf{x}} \right| F \quad (4.10)$$

Dosazením (4.10) do (4.9) se dostane

$$\frac{\partial \psi}{\partial q} + \left| \frac{\partial \psi}{\partial \mathbf{x}} \right| F = 0 \quad (4.11)$$

Výraz (4.11) je parciální-diferenciální rovnice Hamilton-Jacobiho typu. Z toho vyplývají i vlastnosti řešení (ψ je hladká funkce) a zároveň i vlastnosti včleněné

funkce \mathbf{v} . Teorie okolo řešení parciálních-diferenciálních rovnic je za rámec této práce, proto je zde zmíněn jen numerický algoritmus z [28]. Hodnoty funkce ψ se aproximují v diskrétním rastru děleném uniformě v x a y krokem velikosti h a v q krokem Δq . Hledají se tedy hodnoty $\psi(ih, jh, n\Delta q)$. Nahrazením derivací diferencemi v rovnici (4.11) dostáváme

$$\frac{\Psi_{ij(n+1)} - \Psi_{ijn}}{\Delta q} + |\nabla \Psi_{ijn}| F(\mathbf{x}(n\Delta q)) = 0 \quad (4.12)$$

$$\nabla \Psi_{ijn} = \begin{pmatrix} \frac{\Psi_{(i+1)jn} - \Psi_{ijn}}{h} \\ \frac{\Psi_{i(j+1)n} - \Psi_{ijn}}{h} \end{pmatrix}$$

Zbývá tedy formulovat F . Hodnota F se rozdělí na dvě komponenty F_A a F_G . F_A je velikost síly, která zapříčiňuje neustálý pohyb hranice. Odpovídá vlastně nafukovací síle u balónků. F_G je velikost síly, která vychází z geometrie hranice. Stará se o její vyhlazení. Má stejný význam jako E_{int} u aktivních kontur.

$$\frac{\partial \psi}{\partial q} + \left| \frac{\partial \psi}{\partial \mathbf{x}} \right| F_A(\mathbf{x}(q)) + \left| \frac{\partial \psi}{\partial \mathbf{x}} \right| F_G(\mathbf{x}(q)) = 0$$

Dále jsou dvě možnosti, jak nové termy použít. V prvním případě nebude do výpočtu zahrnuta vyhlazující F_G , tzn. $F_G = 0$. Místo F_A se definuje nová síla F_I , která působí proti F_A a zároveň je řízena obsahem obrazu.

$$F_I = \frac{-F_A}{M_1 - M_2} (|\nabla G_\sigma * f(x, y)| - M_2)$$

Kde M_1 a M_2 jsou maximum a minimum z gradientu vyhlazeného obrázku $\nabla G_\sigma * f(x, y)$. Síla F_I teď může nabývat hodnot $(-F_A, 0)$. V takovém případě dosahuje F_I minima tzn. $-F_A$ v místě, kde je nejsilnější hrana, platí $F = 0$ a hranice se na takovém místě zastaví. Kde je gradient maximální, tedy M_1 je $F_I = 0$ a hranice se posunuje ve směru normály silou F_A .

V případě, že se použije i vyhlazovací síly F_G , je přístup poněkud odlišný. Obě dvě nenulové síly F_G a F_A se vynásobí hodnotou k_I .

$$k_I(x, y) = \frac{1}{1 + |\nabla G_\sigma * f(x, y)|}$$

Výraz k_I nabývá hodnot blízkých nule v místech, kde je vysoký gradient. A naopak blíží se k jedničce v místech, kde je intenzita obrazu téměř konstantní. Výrazněji k nule padá k_I ve tvaru

$$k_I(x, y) = e^{|\nabla G_\sigma * f(x, y)|}$$

Předchozí definice sil pro vývoj nadroviny ψ se vztahovaly pouze na nultou hladinu. Protože se hovořilo o silách odvozených z obrazu, který se vyskytuje v rovině nulté hladiny, musí se síly dodefinovat na celém ψ . Proto termy F_I a k_I odvozené

z obrazu rozšíříme na celou ψ a vzniknou tak rozšířené \hat{F}_I a \hat{k}_I . Rovnice pro plochu v obou diskutovaných případech tedy budou vypadat takto.

$$\begin{aligned} \frac{\partial \psi}{\partial q} + \hat{F}_I \left| \frac{\partial \psi}{\partial \mathbf{x}} \right| &= 0 \\ \frac{\partial \psi}{\partial q} + \hat{k}_I (F_A + F_G) \left| \frac{\partial \psi}{\partial \mathbf{x}} \right| &= 0 \end{aligned}$$

Pro rozšířenou funkci \hat{F}_I musí platit rovnost s původní funkcí F_I na nulté hladině. Mimo nultou hladinu lze funkci dodefinovat například hodnotou síly F_I nejbližšího bodu na vrstevnici nulté hladiny.

Iterativní algoritmus z předchozího výkladu je stručně shrnut na výpisu algoritmu 1.

Algoritmus 1 Level-sets - segmentace

1. Vstupní křivka $\mathbf{v}(q = 0)$ tvořící vrstevnici nulté hladiny.
 2. Konstrukce mřížky hodnot funkce $\psi(x, q = 0)$: hodnoty na křivce \mathbf{v} jsou nastaveny na 0. Hodnoty mimo křivku jsou inicializovány vzdáleností nejbližšího bodu křivky. Navíc vzdálenost se počítá záporná pro body uvnitř křivky a kladná pro body vně.
 3. Konstrukce rozšířené \hat{F} pro všechny body mřížky.
 4. Udatování mřížky podle rovnice (4.12).
 5. Konstrukce nové kontury $\mathbf{v}(q)$ z mřížky ψ .
 6. Pokud není splněno konvergenční kritérium, pokračuje se $(n + 1)$. krokem od bodu 3 pro $q = (n + 1)\Delta q$.
-

Je dost pravděpodobné, že přímo nulové hodnoty se po kroku 4 algoritmu v mřížce vyskytovat nebudou. Rekonstrukce kontury se provede během průchodu mřížky nalezením bodů, ve kterých se mění znaménko oproti sousedům. Za hraniční bod lze tedy prohlásit například ten, jehož soused ve 4-okolí, nebo 8-okolí je opačného znaménka. Přesnější výsledek lze získat bilineární interpolací. Jinou možností je trackování hranice záporné oblasti funkce ψ .

4.5.1 Narrow-band

Jak je vidět, jeden krok algoritmu zahrnuje konstrukci funkce rychlosti \hat{F} tak, že hodnota v (x, y) je rovna hodnotě v nejbližším bodě vrstevnice nulté hladiny. To znamená, že pro každý bod diskrétní mřížky se hledá nejbližší bod na křivce $\mathbf{v}(q)$. To způsobuje v praxi téměř neúnosnou výpočetní složitost. Proto se hledalo řešení,

jak celý algoritmus urychlit. Prvním trikem, který uvádí [28], je definovat si úzký pásek (narrow-band) okolo hladiny nulté úrovně a updatovat ψ a \hat{F} během každého kroku pouze v něm. Jednou za čas, například po 20 krocích v závislosti na šířce pásku, reinitializovat ψ na celém obraze.

4.5.2 Fast Marching Level-sets

Algoritmus segmentace pomocí level-setů je i s použitím narrow-bandu stále poměrně dost pomalý. Proto byla odvozena slabší varianta [36], co do schopností segmentace, která je však ještě o něco rychlejší. V první řadě předpokládá, že hodnota F má na celém oboru stejné znaménko. Buď je všude kladná, nebo všude záporná. Chování algoritmu je možné si představit tak, že hranice oblastí se šíří jako hranice rozlévajících se kapalin. Nikdy se nevrací zpět, neustupuje z již zaplaveného území. Díky tomu lze zúžit narrow-band na šířku jednoho pixelu ve směru šíření. Během šíření vlny se pak zároveň mění i narrow-band. Pokaždé, když je pixel narrow-bandu zaplaven, jsou jeho sousedé do narrow-bandu přidáni. Rozhodnutí, který pixel zaplavit, je dáno nejmenší hodnotou funkce $T(x, y)$. Funkce $T(x, y)$ má význam času, ve kterém se vlna dostane nad bod (x, y) . Tato funkce splňuje

$$|\nabla T|F = 1$$

Funkce T je inicializována na počátku pouze v narrow-bandu výchozí křivky. V průběhu algoritmu se pak dopočítává pro všechny nové body narrow-bandu. Tímto zjednodušením se v jednom kroku místo procházení plochy prochází pouze body křivky. Platí, že během celého výpočtu je každý bod obrazu navštíven maximálně jednou.

4.6 Modely řízené statistikou oblasti

Všechny předchozí algoritmy deformovatelných modelů měly jednu charakteristickou a zcela zásadní vlastnost. Jejich výpočet byl řízen hranami v obraze. To však předpokládá, že obraz bude mít kvalitní hrany a málo šumu. Segmentované objekty a tvary budou relativně ostře ohraničeny. To právě většinou není případ lékařských snímků, jak je vidět v úvodu na Obrázku 2.1. Data z CT jsou více zašuměná při větším rozlišení a tenčích řezech. Na druhou stranu data s nízkým rozlišením jsou málo přesná. Objekty na nich mohou splývat a hrany mohou být rozmazané. Proto není dobré spoléhat jen na schopnost hran zastavit vyvíjející se aktivní konturu. Hrany mohou být podél okrajů objektů velmi nerovnoměrné a nevýrazné. I to je dobře vidět na Obrázku 2.1. Právě proto deformovatelné modely řízené hranami často selhávají na lékařských datech, nebo jsou hodně závislé na parametrech vnitřních vyhlazujících sil, zásahu uživatele, či apriorní informaci.

V práci [8] byl předveden model postavený čistě na statistice rozložení intenzity v segmentované oblasti a mimo ni. Celou oblast obrazu označíme ω . Pokud máme uzavřenou křivku \mathbf{v} , pak výraz $int(\mathbf{v})$ značí oblast uvnitř křivky a výraz $ext(\mathbf{v}) =$

$\omega \setminus \text{int}(\mathbf{v})$ oblast vně křivky. Energetický funkcionál modelu, který počítá s tím, že se obraz skládá ze dvou oblastí definovaných relativně konstantní intenzitou, je

$$E_{region}^*(\mathbf{v}) = \alpha \int_{\text{int}(\mathbf{v})} (f(x,y) - \mu_{\text{int}(\mathbf{v})})^2 dx dy + \beta \int_{\text{ext}(\mathbf{v})} (f(x,y) - \mu_{\text{ext}(\mathbf{v})})^2 dx dy \quad (4.13)$$

Kde \mathbf{v} je křivka ohraničující segmentovaný objekt. $f(x,y)$ je hodnota intenzity obrazu v bodě (x,y) . Termy μ_{int} a μ_{ext} jsou střední hodnoty intenzit oblastí int a ext . Parametry α a β je možné řídit váhu obou oblastí v celkovém součtu energie. Úloha je opět stejná jako u aktivních kontur, minimalizovat energetický funkcionál (4.13). Minimalizací hodnoty energie dochází k tomu, že se oblasti formují tak, aby obsahovaly pouze ty body s intenzitou, která je nejbliž středním hodnotám. Hranice \mathbf{v} expanduje a pohlcuje body s podobnou intenzitou, zachovávajíc homogenitu celé oblasti int . Trochu to připomíná narůstání oblastí z kapitoly 3.2. Algoritmus zde má však plnou kontrolu nad tvarem prostřednictvím jeho hranice.

Hodnoty $\mu_{\text{int}}(\mathbf{v})$ a $\mu_{\text{ext}}(\mathbf{v})$ může zadat uživatel, například díky hlubším znalostem o segmentovaném objektu. Nebo mohou být počítány v průběhu algoritmu z aktuálního stavu vývoje křivky $\mathbf{v}(t)$. Střední hodnota intenzity v oblasti je dána vztahem

$$\mu_R = \frac{\int_R f(x,y) dx dy}{\int_R 1 dx dy}$$

Model je možné obohatit přidáním dalších termů, jakými jsou délka hrany a obsah oblastí. Tím ovšem přibudou váhové parametry, na jejichž nastavení bude výpočet citlivý. Volba takových parametrů se podobá příslovečnému sčítání jablek s hruškami.

$$E_{CV}^*(\gamma) = E_{region}^*(\mathbf{v}, \alpha, \beta) + \nu \text{Length}(\mathbf{v}) + \lambda \text{Area}(\mathbf{v})$$

Praktická implementace naráží na velkou nevýhodu tohoto modelu. Je jí výpočetní náročnost při počítání plošných, nebo objemových integrálů. V [8], kde je tento model použit, jsou kontury reprezentovány prostřednictvím level-setů. Díky tomu se výpočet plošného integrálu snadno ztratí při vyhodnocování ψ . Algoritmus navíc zadarmo získá mnohé výhody level-setů, jako je libovolná změna topologie v průběhu výpočtu, nebo nezávislost na poloze výchozí křivky.

4.6.1 Využití Greenovy věty

Problém časové náročnosti při výpočtu plošného integrálu se dá elegantně vyřešit využitím matematické teorie, konkrétně Greenovy věty.

Věta 1 (Greenova) Je-li $G \subset \mathbf{R}^2$ horizontálně a vertikálně jednoduchá (G lze v horizontálním a vertikálním směru ohraničit funkcemi f_1, f_2 , tak, že $f_1 < f_2$) a otevřená. $C = \bar{G}/G$ je po částech hladká, jednoduchá, uzavřená cesta. Nechť $F : G \rightarrow \mathbf{R}^n$ je spojitě a má spojitě parciální derivace 1. řádu na \bar{G} . Pak platí

$$\oint_C F_1 dx + F_2 dy = \iint_G \frac{\partial F_2}{\partial x} - \frac{\partial F_1}{\partial y} dx dy \quad (4.14)$$

Navíc platí

$$\oint_C F_1 dx = \iint_G -\frac{\partial F_1}{\partial y} dx dy, \quad \oint_C F_2 dy = \iint_G \frac{\partial F_2}{\partial x} dx dy \quad (4.15)$$

Greenova věta je speciální verzí obecnější Stokesovy věty. Jde vlastně o převod výpočtu rotací na množině G na výpočet cirkulace kolem množiny G . Greenova věta je klasický prostředek pro výpočet křivkových a plošných integrálů. V praxi se lze setkat se zařízením jménem planimetr, který funguje na principu Greenovy věty. Dokáže měřit obsah oblasti trackováním jejího obvodu. Matematicky se to dá vyjádřit takto

$$\begin{aligned} \iint_G 1 dx dy &= \frac{1}{2} \iint_G 1 - (-1) dx dy \\ &\stackrel{Green}{=} \frac{1}{2} \oint_C \left(\int -1 dy \right) dx + \left(\int 1 dx \right) dy \\ &= \frac{1}{2} \oint_C -y dx + x dy \end{aligned}$$

Greenova věta se dá využít k výpočtu modelu podobného (4.13). Nejprve si uvědomíme, co je vlastně třeba spočítat. Minimalizace funkcionálu E_{model}^* nějakého modelu s integrálem přes množinu se dosáhne metodou snižování gradientu. K tomu jsou potřeba derivace E_{model}^* podle souřadnic bodů křivky.

$$\begin{aligned} E_{model}^* = \int_G F dx dy &\stackrel{Green}{=} \oint_C \left(\int F dy \right) dx \\ &= - \oint_C \left(\int F dx \right) dy \end{aligned} \quad (4.16)$$

$$\frac{\partial E_{model}^*}{\partial x} = - \oint_C F dy \quad (4.17)$$

Při derivování pravé strany (4.17) byla použita věta o derivaci integrálu s parametrem. Derivace podle y se spočítá analogicky. Díky užití Greenovy věty je dosaženo snížení výpočetní složitosti při vyhodnocení parciální derivace podle jedné souřadnice ze zhruba $O(\text{vyska_obrazu} \times \text{sirka_obrazu})$ na $O(\text{delka_krivky})$.

4.7 Apriorní znalosti

Doposud se pracovalo pouze s intenzitou a s gradientem daného segmentovaného obrazu. Nepočítalo se žádnými jinými vstupními informacemi a tvořené modely hledaly a rekonstruovaly tvary bez povědomí, o jaké objekty se zhruba jedná. Informace o tom, jaký mají přibližně tvar, nebo jaké je předpokládané rozložení intenzity by mohlo pomoci v jejich nalezení a přesné segmentaci. Tato data "navíc" se nazývají apriorní znalosti. Typů apriorních znalostí a způsobů, jak s nimi pracovat při analýze obrazu, je mnoho, viz [38]. Pro účely segmentace obrazu a rekonstrukce objektů se hodí hlavně statistické informace o tvaru a intenzitě.

4.7.1 Statistická analýza - PCA

Analýza hlavních komponent (Principal Component Analysis, PCA) je nástrojem statistické matematiky v boji s množinami mnohodimenzionálních dat s cílem je co nejvíc zjednodušit. Stručně řečeno, cílem PCA je zjistit korelace různých složek vzorků dat napříč celou množinou X . PCA je lineární transformace, stejně tak jako Fourierova transformace. Převádí hodnoty z původního systému souřadnic do takového, kde projekce analyzovaných vektorů s největší variabilitou bude zobrazena do první souřadnice. Ty, co mají druhou největší variabilitu se zobrazí do druhé souřadnice, atd. Z toho vyplývá, že na rozdíl od ostatních lineárních transformací nemá PCA explicitně definovanou bázi. Její báze se odvodí z dat. Pro první bázový vektor \mathbf{w}_1 odpovídající první nejdůležitější souřadnici platí

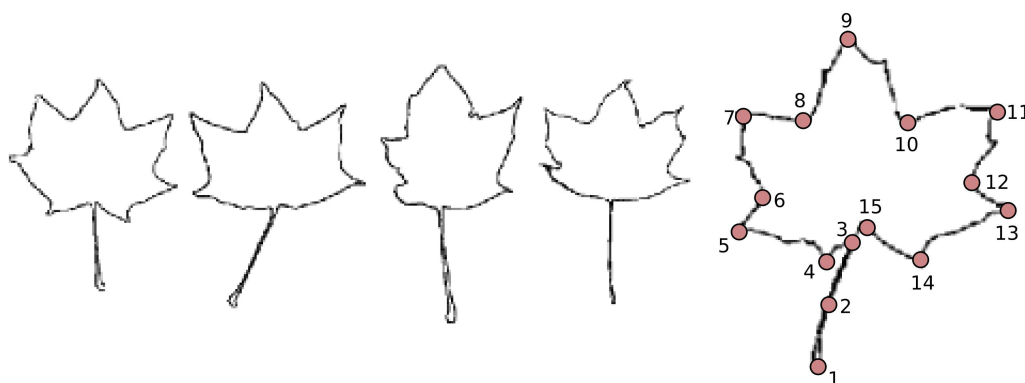
$$\mathbf{w}_1 = \operatorname{argmax}_{|\mathbf{w}|=1} E\{(\mathbf{w}^T(\mathbf{x} - E(X)))^2\}$$

Poslední souřadnice odpovídají směřům s nejmenší variabilitou, které jsou pro zachování informace nejméně důležité. Redukce dimenzionality lze dosáhnout zahazením těch několika posledních, méně významných, souřadnic. V praxi se PCA báze množiny X získá spočtením vlastních vektorů a vlastních čísel kovarianční matice množiny X .

PCA se kromě jiného používá například k redukci počtu charakteristických vlastností v teorii rozpoznání vzorů, nebo k nalezení optimálních obálkových těles v počítačové geometrii.

4.7.2 Point Distribution Model

Model distribuce bodů je jedna možnost, jak zachytit tvar 2D nebo i 3D objektů. Perfektně se hodí k integraci do deformovatelných modelů. Nepředpokládá, že objekty mají pevný, nedeformovatelný tvar. Základem modelu je rozložení sledovaných bodů (landmarks) na hranici, nebo povrchu objektů, případně na jeho vnitřních strukturách. V základní verzi jsou to navíc taková místa, která v obraze tvoří hrany. Rozložením těchto bodů je možné zdůraznit důležitost některých částí objektu, nebo zvýraznit detaily. Položení bodů se v horším případě provádí manuálně, nebo v lepším případě pomocí nějakého poloautomatického [17], či



Obrázek 4.4: Ukázka označování trénovací množiny

plně automatického [21] algoritmu. Snahou je označovat takto větší množinu exemplářů jedné třídy objektů. Aby se tyto exempláře daly mezi sebou srovnávat, musí si body na různých exemplářích odpovídat. Přirozeně, značek musí být na každém exempláři stejný počet. Pokud by byly značky očíslovány přirozenými čísly, tak značky konkrétního čísla musí být na dvou exemplářích umístěny na "stejném" místě. Na Obrázku 4.4 je vidět několik neoznačovaných exemplářů javorového listu a jeden označovaný. Značka číslo 9 označuje vrchol prostředního výběžku listu na každém exempláři.

Variabilita tvaru exemplářů se předpokládá a bude následně podrobena statistické analýze. Co však vadí, je jejich různá poloha, natočení a uniformní zvětšení. Nejprve musí být množina exemplářů zarovnána. Uvedený postup je z [14]. Tedy hledají se takové hodnoty parametrů transformace tuhého tělesa, kterými jsou: úhel otočení, vektor posunutí a faktor zvětšení, aby se minimalizovala míra T daná součtem čtverců rozdílů souřadnic. Je-li tvar exemplářů popsán N body, pak mohou být souřadnice značek naskládány do jednoho vektoru délky $2 \times N$. Pro exemplář i tedy $\mathbf{v}_i = (x_1, y_1, \dots, x_N, y_N)$. Zarovnání dvou exemplářů i a j se dosáhne minimalizací výrazu

$$T = (\mathbf{v}_i - M_j(\mathbf{v}_j))^T W (\mathbf{v}_i - M_j(\mathbf{v}_j))$$

Matice W je volitelná diagonální matice parametrů, která umožňuje dát při zarovnávání různým značkám jinou váhu. Například by bylo vhodné, aby při zarovnávání listů z Obrázku 4.4 byl kladen důraz na překryv ústí řapíku a vrcholu prostředního článku (číslo 9 a 3). Nižší váhy mají většinou značky, které exemplář od exempláře nejvíce mění svoji polohu. Vyšší váhu oproti tomu mají takové, které jsou polohou stabilní. V [14] je popsán iterativní algoritmus, jak zarovnat postupně všechny exempláře. Algoritmus je shrnut ve výpisu algoritmu 2.

Nyní jsou exempláře zarovnány a dají se spočítat například střední hodnoty souřadnic jednotlivých značek. Tak se získá *průměrný tvar* $\bar{\mathbf{v}}$. Spočítáním rozptylů souřadnic značek se dostanou meze, ve kterých se dané značky vyskytovaly napříč trénovací množinou. Také se dá spočítat rozložení pravděpodobnosti jejich poloh.

Algoritmus 2 PDM - zarovnaní trénovacích tvarů

1. Zarovnej každý exemplář vůči prvnímu.
2. Spočítej průměrný tvar z celé množiny prostým průměrováním souřadnic značek.
3. Zarovnej průměrný tvar na nějakou pevně danou velikost, orientaci a polohu. Tento krok se provádí, aby se v průběhu algoritmu průměrný tvar například neustále nezmenšoval. Nedošlo by ke konvergenci.
4. Zarovnej každý exemplář vůči průměrnému tvaru.
5. Dokud se průměrný tvar mění, opakuj od kroku 2.

Tyto meze a pravděpodobnosti pak mohou sloužit k omezení pohybu aktivních kontur. Omezení se vztahuje vždy jen na jednu značku v rámci průměrného tvaru nezávisle na ostatních. Lepší by bylo nalézt vztahy mezi více značkami tak, aby pohyb jedné značky byl omezen pohybem a polohou ostatních.

K získání těchto vztahů poslouží PCA. Analýza množiny exemplářů započne sestrojením kovarianční matice (dimenze $2N \times 2N$) odchylek exemplářů \mathbf{v}_i od průměrného tvaru $\bar{\mathbf{v}}$

$$S = \frac{1}{N} \sum_{i=1}^N (\mathbf{v}_i - \bar{\mathbf{v}})(\mathbf{v}_i - \bar{\mathbf{v}})^T$$

Spočítají se vlastní vektory \mathbf{p}_i a vlastní čísla λ_i této matice, pro něž platí

$$S \mathbf{p}_i = \lambda_i \mathbf{p}_i, \quad \lambda_i \geq \lambda_{i+1}$$

Dá se ukázat, že vlastní vektor \mathbf{p}_i s větším vlastním číslem λ_i odpovídá směru s větší variabilitou souřadnic. Tyto směry se nazývají *módy variace*. Vektor \mathbf{p}_i reprezentuje odchylku množiny značek. Jakoby svazuje množinu značek dohromady. Protože módy variace odpovídají odchylkám od průměrného tvaru, je možné pomocí nich a vektoru koeficientů \mathbf{b} nakombinovat libovolný tvar \mathbf{v} v množině exemplářů.

$$\mathbf{v}_i = \bar{\mathbf{v}} + \mathbf{b}P, \quad P = (\mathbf{p}_1, \dots, \mathbf{p}_N)$$

Zvětšováním a zmenšováním hodnot b_i se mění jednotlivé charakteristické rysy analyzovaného objektu. V případě listů z Obrázku 4.4 by to mohla být velikost nebo poloha řapíku, velikost výčnělků listu, či velikost mezer mezi výčnělky. Tyto charakteristické rysy jsou dány souhrou poloh několika značek. Přípustné hodnoty parametrů b_i se pohybují v intervalu $\langle -3\sqrt{\lambda_i}, 3\sqrt{\lambda_i} \rangle$. Díky tomu, že jsou vektory \mathbf{p}_i seřazené podle důležitosti, je zajištěno, že při vynechání posledních vektorů dojde k nejmenší chybě. Použije se tedy jen prvních několik vektorů a vlastních čísel,

řekněme k . Právě vysvětlená fáze odpovídá trénování modelu a množina použitých exemplářů se nazývá *trénovací množina*.

4.7.3 Active Shape Model

Active Shape Model (ASM) z [12] [15] je model vycházející z aktivních kontur a PDM. Pohyb křivek probíhá, pod vlivem gradientu, podobně jako u aktivních kontur, je však omezený přípustnými parametry modelu PDM. Samotný postup při hledání tvaru je stručně shrnut v algoritmu 3.

Algoritmus 3 Active Shape Model - algoritmus rekonstrukce tvaru

1. Inicializace \mathbf{x} průměrným tvarem $\bar{\mathbf{v}}$.
 2. Hledání hrany na kolmici ke křivce v místě značek. Získáme posunutí $d\mathbf{x}$
 3. Nalezení parametrů transformace tuhého tělesa M , která minimalizuje součet čtverců souřadnic rozdílu $M(\mathbf{x})$ a $(\mathbf{x} + d\mathbf{x})$.
 4. Zbytek $M(\mathbf{x}) - (\mathbf{x} + d\mathbf{x})$ je deformací tvaru.
 5. Nalezení transformace vektorů $M(\mathbf{x})$ a $(\mathbf{x} + d\mathbf{x})$ do prostoru modelu. Tedy $T(M(\mathbf{x}))$ a $T(\mathbf{x} + d\mathbf{x})$.
 6. Určí se \mathbf{b} a $\mathbf{b} + d\mathbf{b}$. Kdy platí $\bar{\mathbf{v}} + \mathbf{b}P = T(M(\mathbf{x}))$ a $\bar{\mathbf{v}} + (\mathbf{b} + d\mathbf{b})P = T(\mathbf{x} + d\mathbf{x})$.
 7. Update tvarového vektoru, $\mathbf{b} \rightarrow \mathbf{b} + d\mathbf{b}$
 8. Omezí se vektor parametrů \mathbf{b} , aby nepřesahoval přípustné hodnoty. Tedy splnění $D_m = \sum_{i=1}^t (b_i^2 / \lambda_i) \leq D_{max}(= 3)$ se dosáhne naškálváním $b_i \rightarrow b_i \cdot \frac{D_{max}}{D_m}$
 9. Rekonstrukce tvaru \mathbf{x} s novým \mathbf{b} v obrazovém prostoru. Získá se tak přípustný tvar v blízkosti hran.
 10. Dokud se tvar \mathbf{x} mění opakuje se od kroku 2
-

ASM v sobě elegantním způsobem integruje apriorní znalosti. S tím je spojená větší složitost algoritmu jak konstrukčně, tak výpočetně. Algoritmus se skládá z typické fáze učení, která zpracuje trénovací data a vytváří z nich statistickou reprezentaci tvaru, použitou následně k segmentaci objektů. Nevýhodou je nutnost mít dostatek dat použitelných k naučení PDM. Data musí splňovat kritéria statistického vzorku. Nepříjemná práce je i s rozložením značek, obzvlášť když se musí dělat manuálně. Skutečným problémem se však stává v případě rozšíření algoritmu pro segmentaci 3D dat. Pokud jsou trojrozměrná data prezentována v řezech, jak je to typické pro tomografické zobrazovací metody, křivka probíhající v jednom řezu u jednoho exempláře může u jiného probíhat napříč několika řezy, což uživatel nemá

šanci sledovat. Na trojrozměrném modelu je řádově víc značek, proto jsou potřeba řádově větší trénovací množiny. Ručně označkovat takové množiny je téměř nadlidský úkol. Výsledky segmentace 3D dat pomocí ASM byly publikovány v práci [17], nebo [22].

4.7.4 Active Appearance Model

Active Appearance Model (AAM) popsáný v [13] rozšiřuje statistický popis tvaru z PDM použitý v ASM o statistický popis intenzity v obraze. Algoritmus segmentace opět předchází tvorba modelu. Model se vytváří analýzou trénovací množiny a skládá se z vektorů popisujících módy variace, středního tvaru, intenzity a rozsahu koeficientů. První část tvorby modelu je totožná s tvorbou modelu PDM. Druhá část vyžaduje zdeformování (warping) všech trénovacích exemplářů tak, aby značky ležely na pozicích středního tvaru \bar{x} . Zdeformované exempláře se navzorkují. Tyto vzorky se normalizují, aby se odstranil vliv různého jasu. Normalizace vlastně odpovídá zarovnání vzorků v PDM. Na normalizované vzorky se provede PCA. Vzniknou tak dva modely. První je totožný s PDM a popisuje tvar, druhý pak popisuje distribuci intenzity.

$$\mathbf{v}_i = \bar{\mathbf{v}} + \mathbf{b}_v P_v, \quad \mathbf{g}_i = \bar{\mathbf{g}} + \mathbf{b}_g P_g$$

Aplikací PCA na parametry exemplářů (\mathbf{b}_x , \mathbf{b}_g) v těchto modelech složených dohromady se získá jeden sjednocený (appearance) model popisující i vztahy mezi intenzitou a tvarem napříč trénovací množinou. Například při naučení na množině fotografií lidského obličeje se odhalí korelace typu, delší nos vrhá větší stín a podobně.

Rekonstrukce objektu v obraze následně probíhá hledáním parametrů modelu tak, aby model co nejlépe odpovídal předloženému obrazu. Přitom nesmí parametry překročit povolené rozptyly zjištěné ve fázi učení.

4.8 Kombinace všeho - Shape Based Approach

V článku [39] byl uveden algoritmus segmentace založený na deformovatelných modelech, zahrnující většinu doposud uvedených přístupů. Základ tvoří opět minimalizace energetického funkcionálu. Tentokrát je energie počítaná ze statistiky intenzity v oblasti. Tvar je reprezentován level-sety a výpočet probíhá podobně jako v kapitole 4.5. Výpočet tvaru je omezen a řízen apriorní znalostí stejným způsobem, jako tomu bylo v kapitole 4.7 o ASM. Metoda je demonstrována na 2D i 3D umělých a reálných lékařských datech. Výsledky, se kterými je metoda prezentována jsou působivé.

Algoritmus je rozdělen na dvě fáze. První je fáze učení, která proběhne nad vysegmentovanou trénovací množinou. Její smysl a průběh odpovídá trénovací fázi PDM. Liší se především způsobem reprezentace trénovacích tvarů a metodou jejich zarovnání. PDM používalo množinu značek. Zde je tvar i reprezentován bitovou

mapou I^i , kde každý obrazový element odpovídá bitu. Bity pixelů tvořících pozadí mají hodnotu nula a bity pixelů, které odpovídají objektu, mají hodnotu jedna. Zarovnání je se provádí minimalizací snižováním gradientu následujícího výrazu.

$$T = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{\iint_{\omega} (I^j - I^i)^2 dA}{\iint_{\omega} (I^j + I^i)^2 dA} \right) \quad (4.18)$$

Výraz (4.18) tvoří vlastně součty čtverců. Normalizace v sumě má zajistit, aby se tvary během minimalizace nezmenšovaly. Gradient z (4.18) se počítá podle parametrů transformací tuhého tělesa, neboli posunutí, otočení a uniformní zvětšení každé bitové mapy. Proces zarovnání je výpočetně náročný, protože pomalu konverguje, navíc se může stát, že předčasně zkonverguje k lokálnímu minimu. Zlepšení v obou případech je možné dosáhnout hierarchickou optimalizací. Nejprve se zarovnávají tvary v redukováném rozlišení a následně se dorovnají v originálním rozlišení pomocí výrazně menšího počtu kroků.

Sledováním analogie s algoritmem učení PDM bude sestrojen průměrný tvar a množina nejdůležitějších módů variací, představující společné charakteristické znaky tvarů v trénovací množině. Obojí však musí být ve formě level-setů, tedy jako nadroviny procházející nultou hladinou. Tato nadrovina v prostoru dimenze $N + 1$ se dá reprezentovat hodnotami $(N + 1)$. souřadnice v diskrétním N -dimenzionálním rastru. V kapitole 4.5 byla hotnota této souřadnice určena nejkratší vzdáleností od vrstevnice v N -dimenzionálním prostoru nulté hladiny. Hodnota má záporné znaménko, pokud je uvnitř tvaru. Zarovnané tvary I^i se převedou na tuto reprezentaci Ψ^i a vypočte se průměrný tvar $\bar{\Phi}$.

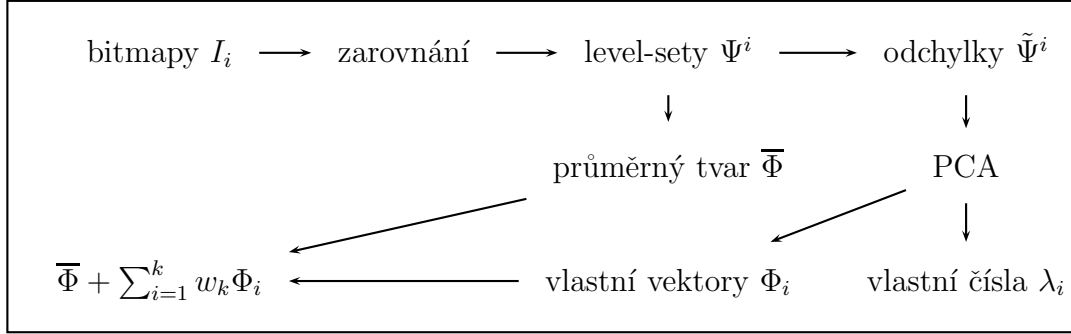
$$\bar{\Phi} = \frac{1}{N} \sum_{i=1}^N \Psi^i$$

Módy variace se získají provedením PCA. Pro každé Ψ^i se spočítá rozdílová hodnota $\tilde{\Psi}^i = \Psi^i - \bar{\Phi}$. Všechny $\tilde{\Psi}^i$ se podle souřadnic lexikograficky uspořádají do vektorů ψ^i . Vytvoří se matice $S = [\psi^1 \dots \psi^n]$. Získají se vlastní čísla λ_i a vlastní vektory \mathbf{p}_i kovarianční matice $\frac{1}{N} S S^T$. Převedením vlastních vektorů zpět na diskrétní rastr se dostanou příslušné módy variace Φ_i , připravené k nakombinování s průměrným tvarem $\bar{\Phi}$. Vlastní čísla zároveň určují podíl vlivu daného módu na charakteristických vlastnostech tvaru. Pro účely modelu se zachová k módů variace s největší hodnotou vlastního čísla. Ty se pak použijí k nakombinování tvarů tak, že pokryjí převážnou většinu přípustných odchylek. Do výpočtu se dají zahrnout i parametry p transformace tuhého tělesa, tím se dostane model pro zachycení objektu v libovolné pozici a libovolného tvaru v přípustných mezích.

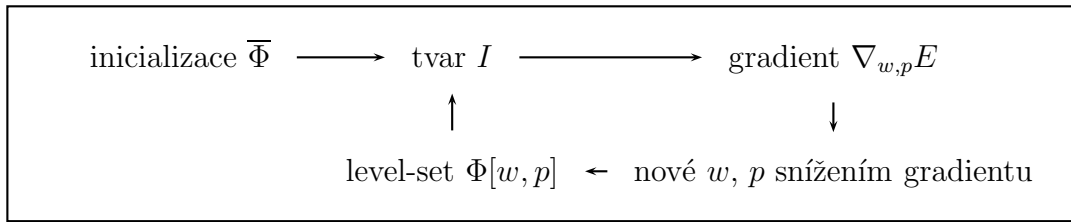
$$\Phi[w, p] = \bar{\Phi}(T[p](x, y)) + \sum_{i=1}^k w_k \Phi_i(T[p](x, y))$$

Během samotné segmentace se minimalizuje funkcionál popisující "věrohodnost" výskytu tvaru snižováním gradientu podle parametrů w a p .

Fáze učení



Fáze segmentace



Obrázek 4.5: Diagram znázorňující průběh algoritmu

Minimalizovaný funkcionál bude zahrnovat statistiku oblasti, podobně jako metoda popsaná v části 4.6. Autoři této metody přišli s dalšími typy funkcionálů

$$\begin{aligned}
 E_{binary}^* &= -\frac{1}{2}(\mu_I - \mu_{\omega/I})^2 \\
 E_{variance}^* &= -\frac{1}{2} \left(\left(\frac{\iint_I f^2 dA}{\iint_I 1 dA} - \frac{\iint_{\omega/I} f^2 dA}{\iint_{\omega/I} 1 dA} \right) - (\mu_I^2 - \mu_{\omega/I}^2) \right)^2 \quad (4.19)
 \end{aligned}$$

Během každého kroku optimalizace se opakovaně musí spočítat gradient. K tomu je nutné získat tvar I z implicitní reprezentace Φ a pomocí něj pak spočítat gradient zvoleného funkcionálu (4.19). Během výpočtu nedochází k žádnému omezení parametrů w_k tak, jako tomu bylo u ASM. Zde metoda těžší pouze z vývoje tvaru povolenými směry módů variace. Všechny kroky algoritmu jsou přehledně znázorněny diagramem 4.5.

Výhody tohoto algoritmu vyplývají z vlastností dílčích metod, jsou shrnuty v následujícím výčtu:

1. Apriorní znalost nedovolí, aby se výsledek segmentace významně lišil od exemplářů z trénovací množiny.
2. Díky apriorní znalosti a samotnému principu deformovatelných modelů se metoda snadno vypořádává se zašuměným obrazem.
3. Reprezentace tvaru objektu level-sety dovoluje libovolnou změnu topologie v průběhu výpočtu.

4. Použití level-setů také vede k menší závislosti na poloze výchozího tvaru.
5. Díky učení modelu z trénovací množiny reprezentované pomocí bitmap odpadá problém se značkováním.
6. Snadné rozšíření algoritmu pro práci s daty libovolné dimenze.
7. Algoritmus je výpočetně náročný.

4.9 B-Spline a Free-Form Deformace

V původním algoritmu aktivních kontur byla hranice reprezentována jako posloupnost bodů. Implicitní reprezentace vnímala konturu tvaru jako nultou vrstevnici jisté nadroviny. Alternativním řešením k těmto dvěma extrémům je použití aproximačních spline křivek, nebo ploch známých z geometrie a počítačem podporovaného návrhu (CAD - *Computer Aided Design*). Nejvhodnějším nástrojem jsou B-Spline křivky.

B-Spline křivka je určena stupněm, několika málo řídicími vrcholy a uzlovými body, které definují báze funkce. B-Spline nabízí volitelně vysokou úroveň kontroly tvaru křivky, který je při popisu hranice různých objektů potřeba. Velkou výhodou při výpočtu aktivních kontur v této reprezentaci je malý počet optimalizovaných parametrů. Výpočet se redukuje z optimalizace souřadnic každého bodu křivky, snaxelu, pouze na optimalizaci souřadnic řídicích bodů. Díky tomu je běh algoritmu s B-Spline rychlejší. Jak je ukázáno v práci [4], dovoluje použití B-Spline křivek úplně vynechat vnitřní vyhlazující term E_{int} zavedený rovnicí (4.3). Samotné použití B-Spline neřeší bohužel segmentaci tvarů se složitější topologií. Dokonce i výpočet průsečíků B-Spline může být problém. Efektivní výpočetní schéma pro B-Spline zahrnující statistiku oblastí uvádí [24].

Myšlenku B-Spline křivek, kdy je tvar ovládán jen několika málo řídicími body sledují i algoritmy rekonstrukce založené na využití Free Form Deformací (FFD). FFD je nástroj známý z CAD, hojně využívaný pro své snadné ovládní a pěkné modelovací výsledky. Do prostoru obrazu je umístěn objekt předpokládaného tvaru a topologie, například z atlasu. Následně se pak optimalizují řídicí body FFD sítě okolo tohoto objektu. Původní objekt se tak zdeformuje, aby nejlépe odpovídal předloženému obrazu. Výhodou je, že FFD síť nemůže libovolně změnit topologii a pohyb řídicích bodů samotných se dá uměle omezit, aby se zamezilo křížení a kontrakcím. Exmplář z atlasu může obsahovat kromě hranice i hodnoty intenzit obrazu uvnitř a v okolí objektu, které lze také zahrnout do optimalizovaného kritéria. Jedno použití FFD lze nalézt v článku [2].

4.10 Shrnutí

Třída algoritmů segmentace založených na deformovatelných modelech je velmi rozsáhlá a není možné uvést všechny existující příklady a modifikace. Dobrým

zdrojem informací může být například kniha o aktivních konturách [3], nebo seznam všech prací [40] citujících původní článek o aktivních konturách.

Předchozí text mapuje nejdůležitější myšlenky okolo deformovatelných modelů. Zmiňuje podrobněji původní článek a uvádí některá vylepšení pokoušející se řešit problematiku otázky. Probírá se i významná implicitní reprezentace aktivních kontur. V druhé polovině je pak představena revoluční změna kritéria ze sledování hran, na sledování oblastí se specifickou intenzitou. Poté je detailně vysvětleno, jak lze vytvořit model tvaru zahrnující apriorní znalost a použít ho k rekonstrukci objektu ve smyslu deformovatelných modelů. Výklad končí syntézou všech představených myšlenek v konkrétní metodě segmentace. Na úplný závěr jsou uvedeny další zajímavé alternativy reprezentace tvaru.

Jediná teze popisující deformovatelné modely by mohla znít takto: *Deformovatelné modely jsou založeny na optimalizaci vztahu mezi abstrakcí a reálnými daty*. Vztahem jsou zde energetické funkcionály. Abstrakcí pak způsob reprezentace tvarů. Optimalizací mohou být různé metody od řešení diferenciálních rovnic, po numerickou nelineární optimalizaci, či různé heuristiky. Daty pak můžeme chápat jak segmentovaný obraz, tak apriorní znalosti.

Kapitola 5

Zvolený postup rekonstrukce

Celou množinu představených algoritmů pro deformovatelné modely se pokusíme posoudit i z praktického úhlu pohledu. Podle rychlosti a složitosti ji můžeme seřadit a rozdělit na dvě kategorie. První z nich jsou algoritmy rychlé a jednoduché. Takové se hodí pro implementaci v rozhraní s požadavkem na rychlou uživatelskou odezvu. Mezi ně patří především různé implementace aktivních kontur bez apriorní znalosti v základním explicitním tvaru.

Druhou kategorií tvoří algoritmy složitější, výpočetně náročnější, ovšem slibující lepší výsledky. K takovým algoritmům můžeme počítat ASM, AAM a jiné algoritmy silně využívající apriorní znalosti, jako je reprezentace v PDM. Extrémně pomalé jsou algoritmy, které používají k reprezentaci tvaru implicitní funkce, level-setsy. Ale i z těch pomalejších jsou vhodnější ty, které nám dovolují rozdělit výpočet na nezávislé fáze, kdy je možné některé výpočty provést dopředu jen jednou a poté vícekrát použít jejich výsledku. Trénování statistického modelu je toho dobrým příkladem. Rychlost algoritmu ovlivňují především dvě věci.

První z nich je počet parametrů modelu. Tento počet přímo určuje dobu jednoho kroku optimalizačního procesu. Počet parametrů je přirozeně větší při zpracování trojrozměrných dat.

Druhý aspekt ovlivňující rychlost optimalizačního kroku je množství práce věnované optimalizaci jednoho parametru. Zejména může jít o spočítání derivace, vyhodnocení funkcionálu, nebo nějakého kritéria.

Do běhu výpočtu jsou často zahrnuty i různé kontrolní a konsolidující operace. Příkladem může být přepočítání narrowbandu u algoritmu level-setů, nebo reparametrizace u balonkového modelu aktivních kontur. Tyto operace se většinou neprovádějí v každém kroku optimalizace. Čas pro jejich vykonání může nastat buď po pevně daném počtu kroků, nebo v případě, že je splněna nějaká snadno vyhodnotitelná podmínka.

Po této úvaze a s ohledem na zadanou úlohu a charakter vstupních dat, které jsou popsány v kapitole 1.1, jsme zvolili vhodné algoritmy segmentace. Vytvořili jsme s jejich pomocí dvě rekonstrukční procedury. Jsou si vzájemně příbuzné, ale konečného cíle, měření objemu, dosahují rozdílným způsobem.

5.1 Rekonstrukce postupnou segmentací řezů

První procedura rekonstrukce objektu v CT datech podle [7] je motivována faktem, že poskytnutá trojrozměrná data jsou silně nehomogenní. Jednotlivé řezy jsou od sebe příliš vzdálené, než aby algoritmus mohl s výhodou těžit stejnou měrou ze všech tří rozměrů. Proto jsme se rozhodli rozdělit výpočet algoritmu na fáze, ve kterých probíhá 2D segmentace jednoho řezu. Po ukončení fáze se práce přesune na sousední řez. Přitom se provedou korigující výpočty, které mají za úkol co nejpřesněji inicializovat segmentaci v následujícím řezu.

Segmentace samotná je postavená na implementaci aktivních kontur podle [24]. Aktivní kontury jsou reprezentovány B-Spline křivkami namísto pouhé posloupnosti bodů. Optimalizace je řízena několika typy energetických funkcionalů, z kterých je nejdůležitější funkcional zahrnující statistiku rozložení intenzity v oblasti.

Procedura je zahájena vstupem uživatele, který označí takzvané póly segmentovaného orgánu. Jde vlastně o označení orgánu v prvním a posledním řezu, ve kterém se vyskytuje. Poté už pokračuje rekonstrukce sama bez vnějšího zásahu. Tuto míru uživatelského vstupu nepovažujeme za přehnanou. Praxe často ukazuje, že uživatel spíše vynaloží úsilí, než aby musel delší dobu čekat, což by byla cena za plně automatickou inicializaci.

Obrázek 5.1 schématicky ukazuje průběh algoritmu od uživatelského vstupu přes posloupnost fází, až k výsledné rekonstrukci. Vedlejším produktem rekonstrukce je i vypočítaná hodnota objemu, která je naším cílem. V následujících odstavcích budou podrobně probrány detaily jednotlivých kroků. Nejprve se zaměříme na algoritmus segmentace v jednom řezu.

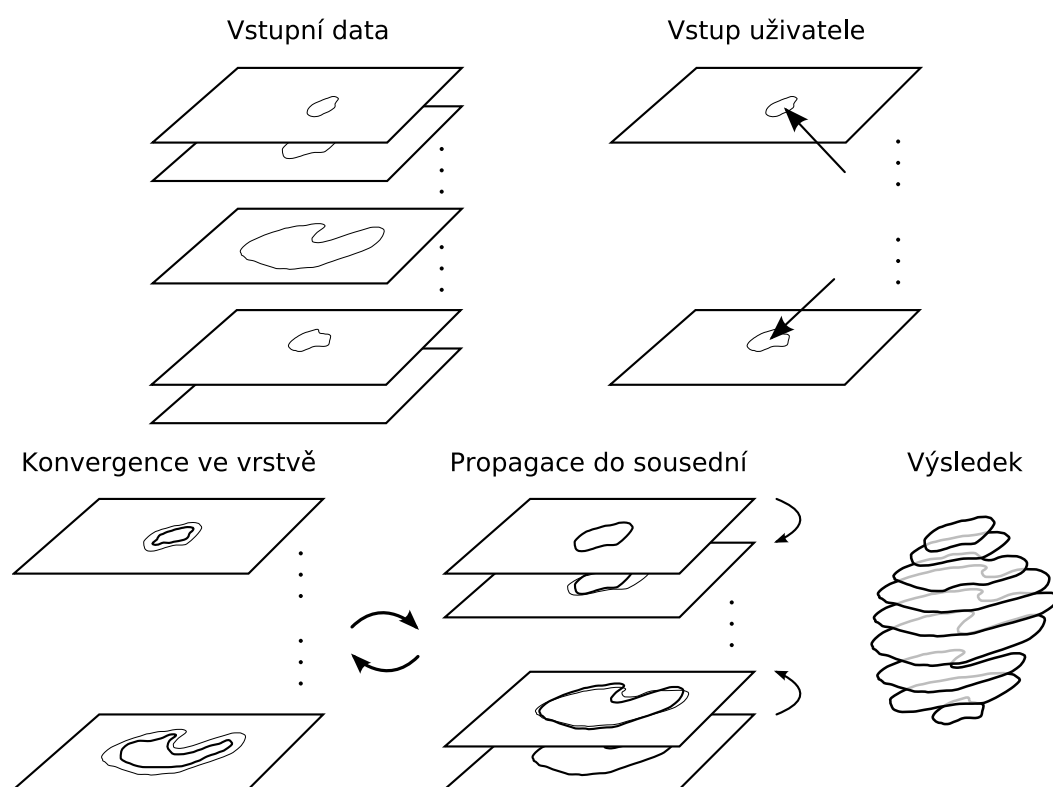
5.1.1 Reprezentace kontury

Jak již bylo ukázáno v průběhu kapitoly 4, deformovatelné modely se skládají ze tří hlavních komponent: reprezentace, modelu a optimalizace. Reprezentace je klíčová jak pro vlastnosti algoritmu, tak pro rychlost výpočtu. Například reprezentace prostřednictvím level-setů přináší pro segmentaci neocenitelné výhody, jako je libovolná změna topologie v průběhu algoritmu, nebo rozšiřitelnost do libovolné dimenze.

Parametrické křivky

Několik prací z pozdější doby, například [24] [4], ukazují, že právě použití B-Spline křivek jako reprezentace v algoritmech aktivních kontur, je dobrou alternativní variantou explicitní reprezentace. B-Spline křivky jsou hojně používány v geometrickém modelování a mají řadu dobrých vlastností, které se skvěle hodí i k našemu účelu. Budeme se držet původního značení a považujeme od teď $\mathbf{v}(t) = (x(t), y(t))$ za B-Spline křivku.

Obecně nalézají parametrické spline křivky využití v mnoha oblastech, nejen v počítačové grafice. Nabízejí jednoduchý způsob, jak vnést spojitost a hladkost ve



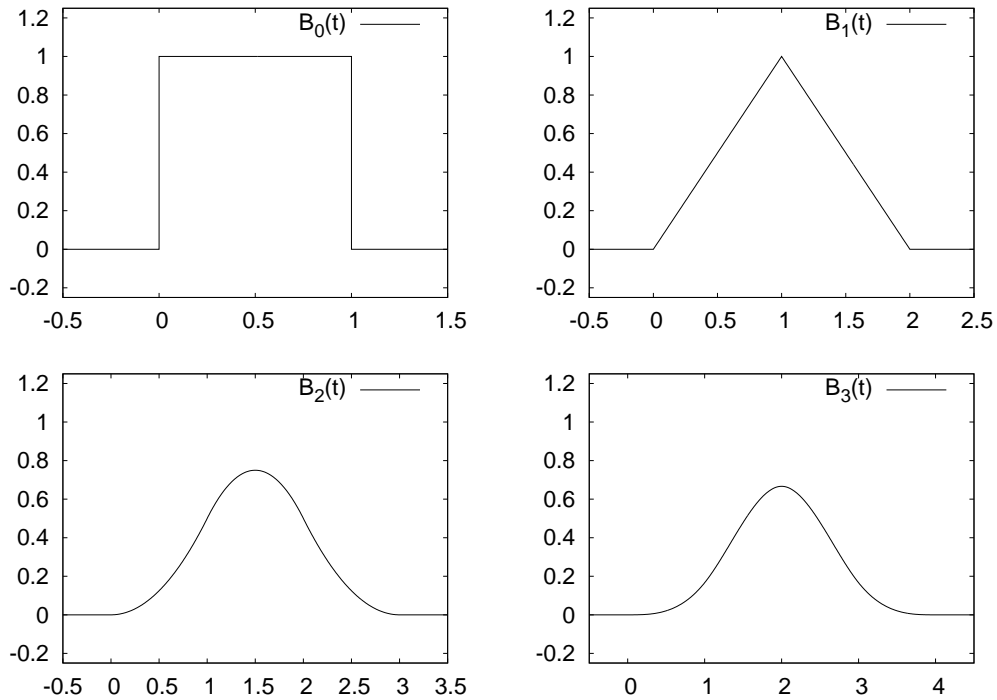
Obrázek 5.1: Schéma procedury rekonstrukce orgánu postupnou segmentací řezů.

smyslu derivací do diskretních dat. Takový byl cíl především *interpolačních křivek*, které měly za úkol poskytnout odhad hodnoty mezi dvěma kontrolními měřeními. Interpolační křivky procházejí těmito řídicími body a propojují je jednou spojitou a hladkou funkcí. V případě, že nejde tolik o přesnou hodnotu funkce, ale hlavně o její tvar, hladkost a spojitost, mohou se hodit *aproximační křivky*. Aproximační křivky neprocházejí všemi svými řídicími body. Pro mnohé interpolační a aproximační křivky však platí, že se dají vyjádřit pomocí řídicích bodů \mathbf{c}_k a takzvaných *bázových funkcí*.

$$\mathbf{v}(t) = \sum_{k=0}^{M-1} \mathbf{c}_k \text{Base}_k(t)$$

B-Spline křivka

Obecnou teorií okolo spline křivek se nebudeme dále zabývat a zaměříme se pouze na námi používané B-Spline křivky a jejich vlastnosti. B-Spline křivka \mathbf{v} stupně N je aproximační křivka definovaná na M bodech, kdy $M > N > 0$. Křivka \mathbf{v} má spojitě derivace do řádu $N - 1$. Bázové funkce B-Spline křivky jsou nenulové na omezeném souvislém úseku jejich definičního oboru. Tyto úseky jsou dány takzvaným *vektorem uzlů* $\mathbf{t} = (t_0, \dots, t_M)$. Posloupnost t_n je neklesající, tzn. $t_0 \leq t_1 \leq \dots \leq t_M$. Podle hodnot t_n se B-Spline křivky dělí na uniformní, kdy $t_{i+1} - t_i = h, \forall i, 0 \leq i \leq M - 1$,



Obrázek 5.2: První báze uniformních B-Spline řádu křivek $N = 1, \dots, 3$ s vektorem uzlů $\mathbf{t} = (0, 1, 2, 3, 4, \dots)$. B-Spline báze řádu N je lineární kombinací posunutých bází řádu $N - 1$.

a neuniformní v ostatních případech. Bázi lze zavést mnoha způsoby, záleží na použití. Nejobecnější je rekurzivní forma.

$$B_{n,k}(t) = \frac{t - t_k}{t_{k+n} - t_k} B_{n-1,k}(t) + \frac{t_{k+n+1} - t}{t_{k+n+1} - t_{n+1}} B_{n-1,k+1}(t)$$

$$B_{0,k}(t) = \begin{cases} 1 & t_k \leq t < t_{k+1} \\ 0 & \text{jinak} \end{cases}$$

Obrázek 5.2 ukazuje grafy bází uniformních B-Spline. Díky tomu, že je každá bazová funkce nenulová jen na omezeném intervalu definičního oboru, má každý řídicí bod vliv pouze na omezený kus křivky. Vlivy řídicích bodů podél křivky se vzájemně překrývají. Na jeden bod křivky má najednou vliv $N + 1$ řídicích bodů. Proto je možné optimalizovat vztah pro výpočet bodu křivky následujícím způsobem.

$$\mathbf{v}(\mathbf{c}, t) = \mathbf{c} \cdot \mathbf{B}_N(t) = \sum_{k=0}^{M-1} \mathbf{c}_k B_{N,k}(t) = \sum_{k=j}^{j+N} \mathbf{c}_k B_{N,k}(t), \quad t \in (t_{j+N-1}, t_{j+N}) \quad (5.1)$$

Součet nenulových bází lze použít pokud známe j . To lze ale zjistit v případě uniformních B-Spline křivek snadno, $j = \lfloor (t - t_1)/h \rfloor$. V takové situaci je smysl parametru k v posunutí bazové funkce. Platí tedy $B_{n,k}(t) = B_{n,0}(t - kh)$. Pro uniformní

B-Spline lze nahradit rekurzivní výpočet báze $B_{n,i}$ jednoduchým posunutím nerekurzivní báze B_n . Můžeme také využít symetrie a toho, že každá báze je prolutím dvou totožných, částečně se překrývajících bází nižšího řádu. Uvádíme vzorec pro bázi kubického B-Spline, který je použit v naší implementaci.

$$\bar{B}_3(t) = \begin{cases} 2/3 + |t|^3/2 - t^2, & 0 \leq |t| < 1 \\ (2 - |t|)^3/6, & 1 \leq |t| < 2 \\ 0 & \text{jinak} \end{cases}$$

Tento vzorec dává bázi B_3 posunutou doleva tak, že je symetrická kolem středu. S tím se musí počítat při vyhodnocování B-Spline. Parametrické křivky jsou většinou uváděny v maticovém tvaru. I pro B-Spline křivku se dá takový tvar odvodit. Často se vyskytujícím příkladem je kubický B-Spline.

$$\mathbf{v}_i(t) = [t^3, t^2, t, 1] \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix}$$

Derivace křivky

K další práci s B-Spline křivkami v našich algoritmech budeme potřebovat jejich derivace podle parametru t . Protože kontrolní body samy nejsou na parametru t závislé, dojde pouze na derivování bazových funkcí. Derivací rekursivního vzorce za použití matematické indukce se dá dokázat [34], že

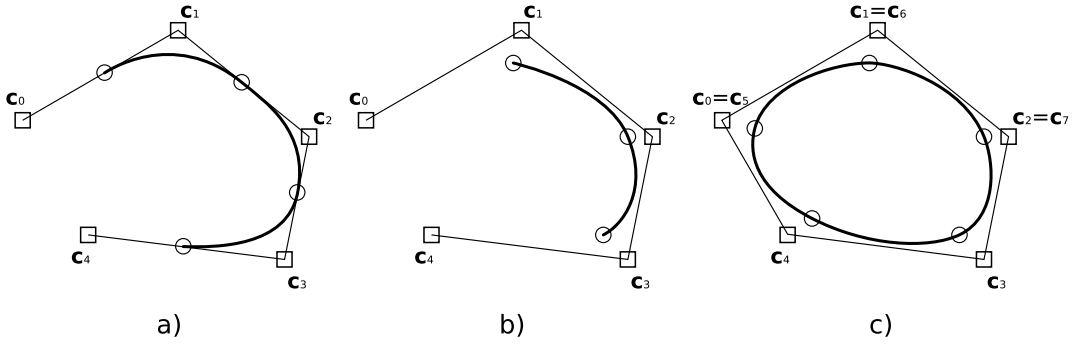
$$B'_{n,k}(t) = \frac{n}{t_{k+n} - t_k} B_{n-1,k}(t) - \frac{n}{t_{k+n+1} - t_{k+1}} B_{n-1,k+1}(t)$$

Uzavřená křivka

Naše úloha vyžaduje použití uzavřených B-Spline křivek. Hladké spojení začátku a konce křivky se realizuje ztotožněním N posledních bodů s N prvními, tedy $c_i = c_{M-N+i}$, $0 \leq i \leq N-1$. Pokud se navíc jedná o neuniformní B-Spline křivku, musí se shodovat rozteč prvních a posledních uzlů, $t_{i+1} - t_i = t_{M-N+i+1} - t_{M-N+i}$, $0 \leq i \leq N-1$. Tak se zajistí, že prvních N bazových funkcí se bude shodovat tvarem s poslední N -ticí bazových funkcí. B-Spline křivka a uzavřená B-Spline křivka je vidět na Obrázku 5.3.

Dělení křivky

Další důležitou operací, kterou budeme u B-Spline křivek potřebovat, je jejich zjemňování přidáváním nových řídicích bodů a uzlů. Vkládání nových bodů se dá provést při zachování původního tvaru. Na druhou stranu, při odebrání řídicích bodů může ke změně tvaru dojít. Zjemnění se provede vložením uzlu t mezi dva existující uzly $t_k < t < t_{k+1}$. Poté dojde k nahrazení řídicích bodů majících vliv na dělenou část křivky. Nové body se vypočítají podle následujícího schématu.



Obrázek 5.3: Příklady B-Spline křivek. **a)** Kvadratický B-Spline. **b)** Kubický B-Spline. **c)** Uzavřený kubický B-Spline. Čtverečky označují řídicí body a kolečka označují takzvané uzlové body, místa na křivce kdy $t = t_i$ pro nějaké i .

$$(t_0, \dots, t_k, t_{k+1}, \dots, t_M) \longrightarrow (t_0, \dots, t_k, t, t_{k+1}, \dots, t_M)$$

$$\mathbf{c}_{k-N}, \underbrace{\mathbf{c}_{k-N+1}, \dots, \mathbf{c}_{k-1}}_{N-1 \text{ starých vrcholů}}, \mathbf{c}_k \longrightarrow \mathbf{c}_{k-N}, \underbrace{\mathbf{c}'_{k-N+1}, \dots, \mathbf{c}'_{k-1}}_{N \text{ nových vrcholů}}, \mathbf{c}_k$$

$$\mathbf{c}'_i = (1 - a_{i-1})\mathbf{c}_i + a_i\mathbf{c}_{i+1}, \quad k - N + 1 \leq i \leq k$$

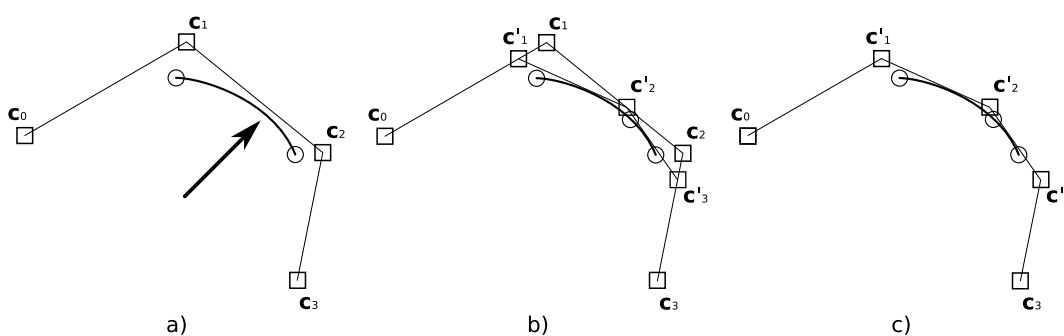
$$a_i = \frac{t - t_i}{t_{i+1} - t_i}, \quad k - N + 1 \leq i \leq k$$

Schéma vkládání jednoho nového uzlového bodu je znázorněno na Obrázku 5.4. Pokud je třeba vytvořit B-Spline křivku z uzlových bodů, tedy vstupem jsou vrcholy, kterými má vytvořená křivka procházet, musíme spočítat polohu všech řídicích bodů a vyrobit i vektor uzlů. Pro spočítání řídicích bodů uniformní, uzavřené B-Spline máme dostatek informací. V ostatních případech bychom museli přijít s dalšími podmínkami. Pokud je křivka uniformní, bude vektor uzlů odpovídat posloupnosti přirozených čísel. Společně se zadanými uzlovými body \mathbf{u}_i máme rovnice

$$t_i = i, \quad \mathbf{u}_i = \mathbf{v}(t_i), \quad 0 \leq i \leq M - 1$$

Dosažením každého uzlového bodu do (5.1) získáme dvě soustavy lineárních rovnic. Jednu pro souřadnici x a druhou pro y .

$$\begin{bmatrix} B(1) & \dots & B(N) & \dots & 0 & 0 & 0 \\ 0 & B(1) & \dots & B(N) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ B(3) & \dots & B(N) & \dots & 0 & B(1) & B(2) \\ B(2) & B(3) & \dots & B(N) & \dots & 0 & B(1) \end{bmatrix} \begin{bmatrix} c_{0,x} \\ c_{1,x} \\ \vdots \\ c_{M-2,x} \\ c_{M-1,x} \end{bmatrix} = \begin{bmatrix} u_{0,x} \\ u_{1,x} \\ \vdots \\ u_{M-2,x} \\ u_{M-1,x} \end{bmatrix}$$



Obrázek 5.4: Vkládání uzlových bodů do B-Spline křivky. **a)** Na místě dělení úseku B-Spline vznikne nový uzlový bod. **b)** Nové řídicí body vzniknou na spojnicích původních. **c)** Výsledná křivka má stejný tvar jako výchozí.

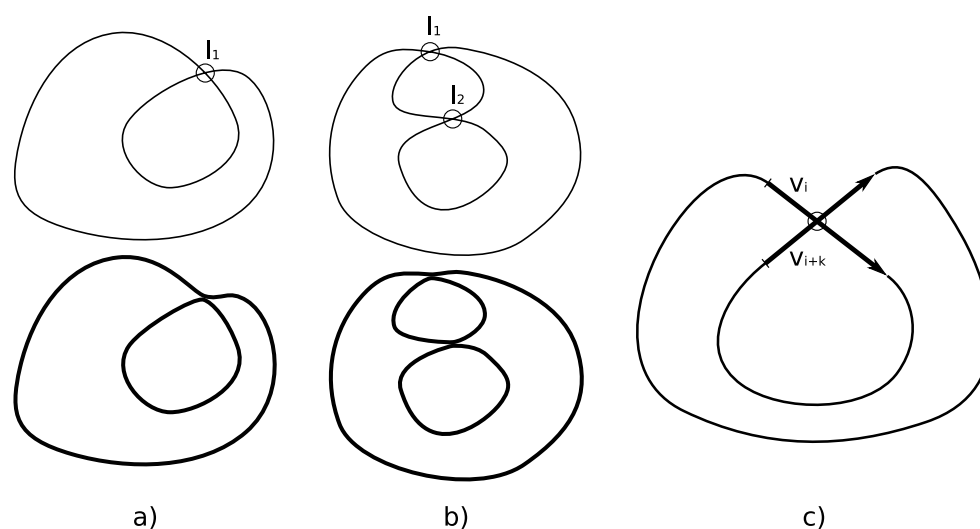
K řešení této soustavy rovnic můžeme použít některou z existujících metod numerické matematiky. V naší implementaci byl použit LU rozklad z [33]. Počítání křivky z uzlových bodů je také výhodný nástroj použitelný při interaktivní editaci křivky. Místo manipulace řídicími body můžeme hýbat přímo uzlovými body a umístit je přímo v místě, kde bude křivka procházet. V naší implementaci se použije metoda dopočítání řídicích bodů pokaždé, když se mění počet uzlů. Řešení soustavy při dělení křivky je sice výpočetně náročnější než vkládání jednotlivých uzlů, ale tento postup je univerzální a použitelný při více operacích.

Kolize křivky

Poslední oblast týkající se použití B-Spline křivek v našem algoritmu je detekce křížení křivky samé se sebou a způsob, jak takovou situaci vyřešit. Úloha skrývá dvě podotázky, a to zdali se křivka někde kříží sama se sebou a kde přesně se křivka kříží. Ideální by bylo, kdyby se dala otázka pouhé existence řešit samostatně a efektivněji než hledání křížení. Při pozitivním výsledku by se pak zjistilo přesné místo křížení časově náročnějším postupem. Při častém opakování s negativním výsledkem by jednoduchý test ušetřil mnoho času.

Autoři [24] se spokojili se zjištěním, jestli křivka obsahuje smyčky, viz Obrázek 5.5a. Smyčky se dají detekovat integrací úhlu směrnice podél křivky. U uzavřené křivky nám vždy vyjde násobek 2π . Pokud je výsledek jiný, než $\pm 2\pi$, je na křivce někde smyčka, tudíž i křížení. Postup selže v případě dvou smyček opačně orientovaných. Navíc, jak ukazuje Obrázek 5.5b, ne každé křížení je zapříčiněno smyčkou.

Rozhodli jsme se detekovat křížení náročnější, ale o něco spolehlivější metodou. Prvně navzorkujeme křivku krokem $\delta t = 1/R$. Tím získáme uzavřenou lomenou čáru. Každou úsečku této lomené čáry pak testujeme oproti všem ostatním. Metoda se dá urychlit využitím vlastností konvexního obalu B-Spline křivky. Platí totiž, že každý úsek B-Spline křivky je uvnitř konvexního obalu řídicích bodů, které tento úsek ovlivňují. Pokud by se vzorkovaly a kontrolovaly pouze ty úseky, jejichž konvexní obaly kolidují, došlo by k dalšímu urychlení. Nutné je poznamenat, že obaly N sousedících úseků vždy kolidují.



Obrázek 5.5: Křížení B-Spline křivky. **a)** Jednoduchá smyčka. **b)** Křížení, které není smyčkou. **c)** Jedno křížení navzorkovaných úseček podél křivky se při testu každý s každým vyskytne dvakrát.

Průsečíky uzavřené křivky se sebou samou ji rozdělí na několik samostatných uzavřených křivek. Tyto křivky získáme dopočítáním řídicích bodů z uzlových bodů uvedeným algoritmem. Jako vstup použijeme posloupnost uzlových bodů, které leží na souvislé části rozdělené křivky a průsečíky, které ji ohraničují.

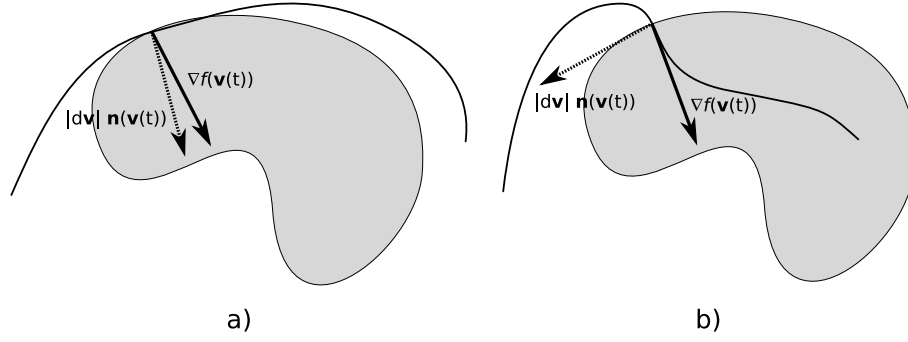
Už několikrát bylo v předchozích odstavcích zmíněno, že použití uniformních B-Spline je rychlejší a vede k jednodušší implementaci. V [4] je dokázáno, že uniformní kubický B-Spline je ekvivalentní klasicky reprezentovaným aktivním konturám. Proto tedy v naší implementaci použijeme uniformní kubické B-Spline křivky s hodnotami uzlů danými uniformní posloupností celých čísel.

5.1.2 Energetický model

Vztah, který vyhodnocuje míru "věrohodnosti" křivky jakožto kontury tvarů v obraze, je zformulován podobně jako u původních aktivních kontur. Zahrnuje citlivost na přítomnost hran v obraze. Důležitější však je, že zohledňuje distribuci intenzity oblastí uvnitř a vně křivky. Vnitřní energie křivky se podstatně liší a má zde trochu jinou interpretaci, než v původních aktivních konturách. V poslední řadě je zahrnut i term započítávající vliv vnějších podmínek. Tento term nalezneme své využití až při použití v rekonstrukční proceduře. Shrnutím do jednoho vztahu získáme energetický model, který je kompozicí všech těchto vlivů.

$$E_{B-Snake}^* = E_{edge}^* + E_{region}^* + E_{int}^* + E_c^*$$

Nejprve se zaměříme na hrany. Pro volbu hranového termu máme dvě možnosti. První z nich je totožná s termem v základních aktivních konturách. Tento term počítá pouze velikost gradientů podél křivky.



Obrázek 5.6: Hranová energie zohledňující směr gradientu. **a)** Křivka prochází ve směru hrany a skalární součin roste. **b)** Křivka prochází kolmo na hranu a skalární součin klesá.

$$E_{edge_mag}^* = - \int |\nabla f(\mathbf{v}(t))|^2 dt$$

Druhou zajímavou možností je term, který využívá i směrovou informaci obsaženou v gradientu. Toho je dosaženo tak, že křivka procházející bodem snižuje hodnotu termu tím víc, čím menší úhel svírá gradient s normálou křivky v daném bodě. Jinými slovy, term se snižuje, když křivka prochází ve směru hrany. Hranová energie řízená směrem gradientu je znázorněna na Obrázku 5.6.

$$E_{edge_dir}^* = - \int \nabla f(\mathbf{v}(t)) \cdot |d\mathbf{v}| \cdot \mathbf{n}(\mathbf{v}(t)) dt$$

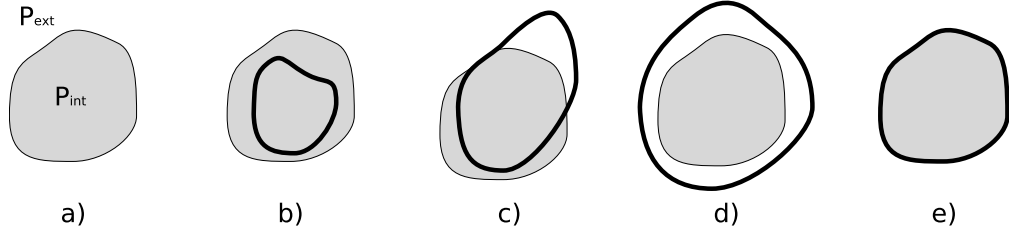
Nejvýznamnější je term pracující se statistikou intenzity objektů a pozadí obrazu. Důvody a výhody použití tohoto termu jsou z části vysvětleny už v kapitole 4.6. Použijeme podobnou energii, jako je ta daná výrazem (4.13). Na rozdíl od ní nebudeme pracovat s odchylkou od střední hodnoty, ale přímo s hodnotou pravděpodobnosti.

$$E_{region}^* = - \int_{int} \log(P(f(s)|s \in int(\mathbf{v}))) ds - \int_{ext} \log(P(f(s)|s \in ext(\mathbf{v}))) ds \quad (5.2)$$

Hustota pravděpodobnosti P bude jeden ze vstupních parametrů algoritmu. Můžeme ji aproximovat pomocí normálního rozdělení, které je dané rozptylem σ a střední hodnotou μ . Dosazením do známého vzorce pro normální rozdělení získáme hodnotu P .

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Druhou možností je definovat funkci P tabulkou pro všechny hodnoty obrazové funkce. V implementaci jsme zvolili právě tuto cestu jako nejspolehlivější způsob, jak poskytnout algoritmu apriorní znalost o segmentovaném objektu. Logaritmus ve vzorci (5.2) slouží jako míra, která dilatuje hodnotu pravděpodobnosti



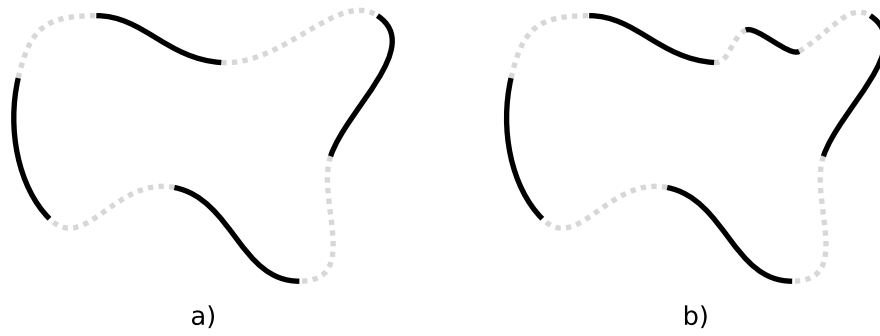
Obrázek 5.7: Vliv polohy křivky na energii oblasti. **a)** Předpokládá se obrázek obsahující oblast s rozložením pravděpodobnosti odpovídající P_{int} a doplněk této oblasti s rozložením P_{ext} . **b)** Oblast vně křivky obsahuje hodně prvků s malou pravděpodobností. **c)** Oblast uvnitř i vně obsahuje prvky s malou pravděpodobností, tudíž je možné zvětšovat energii. **d)** Oblast uvnitř křivky obsahuje nepravděpodobné prvky. **e)** Oblast uvnitř i v ně obsahuje jen pravděpodobné prvky. Energie je maximální.

na neomezený interval $(-\infty, 0)$, čímž se dosahuje lepší separace oblastí. Model je díky tomu stabilnější vůči šumu. Zároveň logaritmus usnadňuje výpočet při použití normálního rozložení. Výraz integrující dvě funkce přes dvě oblasti celého obrazu lze zjednodušit následujícím způsobem.

$$\begin{aligned}
 E_{region}^* &= - \int_{int} \log(P_{int}(f(s))) ds - \int_{ext} \log(P_{ext}(f(s))) ds \\
 &\quad + \left(\int_{int} \log(P_{ext}(f(s))) ds - \int_{int} \log(P_{ext}(f(s))) ds \right) \\
 &= - \int_{int} \log(P_{int}(f(s))) - \log(P_{ext}(f(s))) ds - \int_{int \cup ext} \log(P_{ext}(f(s))) ds \\
 &= - \int_{int} \log \left(\frac{P_{int}(f(s))}{P_{ext}(f(s))} \right) ds + C
 \end{aligned}$$

Hodnota $C = - \int_{int \cup ext} \log(P_{ext})$ není závislá na křivce \mathbf{v} , nebude se tedy měnit během vývoje křivky, proto ji můžeme z termu E_{region}^* vynechat. Při změně křivky definující oblast int a ext se hodnota termu zmenšuje, pokud se zvětšuje pravděpodobnost intenzity v oblasti int , daná hustotou P_{int} , a pravděpodobnost intenzity v oblasti ext , daná hustotou P_{ext} . Oba integrály pracují jako spojité nádoby. Prvky, které při změně křivky odejdou z int , se přesunou do ext . Chování hodnoty tohoto energetického termu demonstruje Obrázek 5.7.

Term vnitřní energie křivky se podstatně mění oproti tomu popsanému v kapitole 4.1.2. V algoritmu původních aktivních kontur byla snaha minimalizovat první a druhou derivaci, tím vyhladit křivku a snížit křivost podél celé její délky. Zde však pracujeme s B-Spline křivkou, která má spojité derivace. S ohledem na parametrizaci je křivka vždy dostatečně hladká. Co ovlivňuje křivost, tedy i hladkost bez ohledu na parametrizaci, je délka křivky mezi uzlovými body. Jinými slovy, hodně malých B-Spline úseků vede k členitosti křivky a velké celkové křivosti. Na Obrázku 5.8 se dá pozorovat, že nejmenší křivost podél celé křivky lze zajistit stejnou délkou všech B-Spline úseků.



Obrázek 5.8: Vliv délky úseků B-Spline křivky na celkovou hladkost. **a)** Pokud jsou všechny úseky stejně dlouhé, křivka má nejmenší celkovou křivost. **b)** Krátké úseky způsobují členitost křivky.

V [4] je dokonce term pro vnitřní energii úplně vynechán a hladkost je regulována rozestupem uzlů. My se budeme podle [24] snažit udržovat stejně dlouhé B-Spline úseky. Nejprve musíme zjistit délku celé křivky

$$Length(\mathbf{v}) = \int_0^M |\mathbf{v}'(t)| dt = \int_0^M \sqrt{x'(t)^2 + y'(t)^2} dt$$

Délka jednoho úseku křivky z celkového počtu M je zachycena konstantou c .

$$c = \left(\frac{Length(\mathbf{v})}{M} \right)^2$$

Term, jenž postihuje části křivky delší, nebo kratší než kolik je průměrná délka takové části, je vlastně součtem čtverců chyb, které se snažíme minimalizovat.

$$E_{int}^* = \int_0^M (|\mathbf{v}'(t)|^2 - c)^2 dt$$

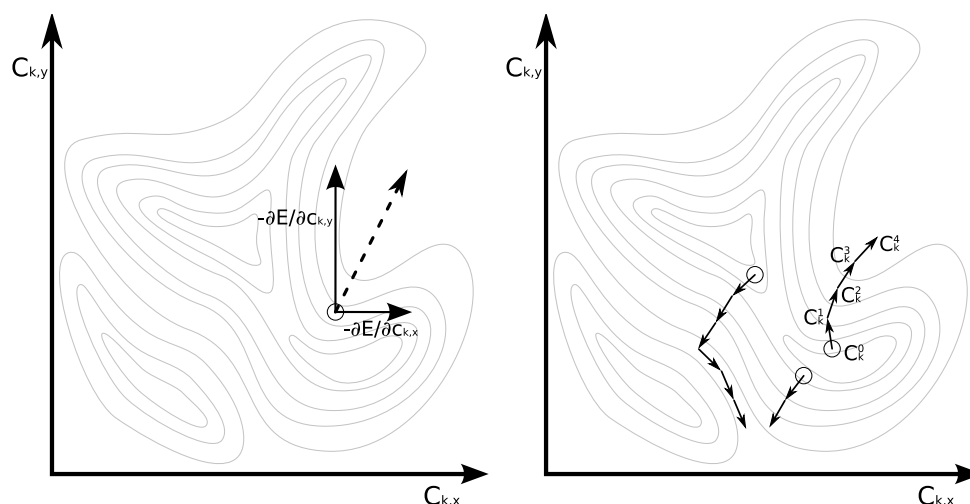
Poslední term, který zbývá definovat, je term vyhodnocující vliv vnějších podmínek na vývoj křivky. Prostřednictvím tohoto termu se ovlivňuje vývoj křivky v průběhu rekonstrukce po řezech tak, že se při propagaci křivky do sousedního řezu vytvoří šablona, od které se křivka během svého vývoje nesmí příliš vzdálit. Minimalizace tohoto termu odpovídá minimalizaci skalární funkce podél křivky.

$$E_c^* = \int_0^M dist\ field(\mathbf{v}(t)) dt$$

Tato skalární funkce je v našem případě předpočítaná nejkratší vzdálenost k dané šabloně $temp$.

$$dist\ field(\mathbf{x}, temp) = \min_t |\mathbf{x} - temp(t)|$$

Přitom nepočítáme jen šablonu jednoho sousedního řezu, ale zahrneme do výpočtu i řezy vzdálenější, samozřejmě s menší vahou. To všechno pro případ, že



Obrázek 5.9: Snižování gradientu. Vlevo: Výběr je řízen záporným gradientem optimalizované funkce. Vpravo: Příklad procesu optimalizace. Vývoj silně závisí na inicializaci.

segmentace předchozího sousedního řezu selhala a šablona z něj samotného by byla špatná.

$$dist\ field(\mathbf{x}) = \sum_{\mathbf{v} \in neighbours} weight(\mathbf{v}) dist\ field(\mathbf{x}, \mathbf{v})$$

5.1.3 Optimalizace

K optimalizaci definovaného modelu jsme použili snižování gradientu, které bylo popsáno v kapitole 4.4.1. Pro zopakování, snižováním gradientu iterativně minimalizujeme funkci $F(x)$ tak, že systematicky vybíráme její parametr z definičního oboru proti směru gradientu. Na Obrázku 5.9 je znázorněn tento proces.

$$x_{n+1} = x_n - \delta \nabla F_n$$

V limitě x_n tak získáme lokální minimum funkce. Existuje však nebezpečí, že při běhu algoritmu uvízneme v lokálním minimu. Proto je celý proces závislý na dobré inicializaci x_0 . Velikost kroku se dá řídit parametrem δ . Menší hodnota způsobí pomalou konvergenci a zvýší šanci, že proces někde uvízne. Větší hodnota zase vede k nepřesnostem.

V našem případě zastupuje funkci F energetický funkcionál $E_{B-Snake}^*$. Parametry, které optimalizujeme, jsou řídicí body B-Spline křivky \mathbf{v} .

$$F(x) = E_{B-Snake}^*(\mathbf{v}(\mathbf{c}, t)) = E_{B-Snake, \mathbf{v}}^*(\mathbf{c})$$

Parciální derivace

Klíčové jsou parciální derivace energetického funkcionálu, tedy všech termů v jeho součtu. Protože hodnota funkcionálu není v proporcionálním vztahu k prostoru parametrů, je lepší gradient normalizovat. Hodnota parametru δ pak odpovídá velikosti změny vektoru parametrů. Tímto způsobem se lépe kontroluje optimalizační proces. Na druhou stranu ztratíme informaci o velikosti gradientu.

$$\begin{aligned} \mathbf{c}^{n+1} &= \mathbf{c}^n - \delta \cdot \text{step}(\mathbf{c}^n) = \mathbf{c}^n - \delta \frac{\nabla_{\mathbf{c}} E_{B-Snake, \mathbf{v}}^*}{|\nabla_{\mathbf{c}} E_{snake, \mathbf{v}}^*|} \\ &= \mathbf{c}^n - \delta \frac{\nabla_{\mathbf{c}} E_{edge, \mathbf{v}}^* + \nabla_{\mathbf{c}} E_{region, \mathbf{v}}^* + \nabla_{\mathbf{c}} E_{int, \mathbf{v}}^* + \nabla_{\mathbf{c}} E_{c, \mathbf{v}}^*}{|\nabla_{\mathbf{c}} E_{edge, \mathbf{v}}^* + \nabla_{\mathbf{c}} E_{region, \mathbf{v}}^* + \nabla_{\mathbf{c}} E_{int, \mathbf{v}}^* + \nabla_{\mathbf{c}} E_{c, \mathbf{v}}^*|} \end{aligned}$$

V následujících odstavcích odvodíme gradienty jednotlivých termů. Začneme určením gradientu termu E_{edge_mag} , který se skládá z parciálních derivací.

$$\nabla_{\mathbf{c}} E_{edge_mag, \mathbf{v}}^* = \left[\frac{\partial E_{edge_mag, \mathbf{v}}^*}{\partial c_{1,x}}, \frac{\partial E_{edge_mag, \mathbf{v}}^*}{\partial c_{1,y}}, \dots, \frac{\partial E_{edge_mag, \mathbf{v}}^*}{\partial c_{n,y}} \right]$$

Odvodíme parciální derivaci podle souřadnice x řídicího bodu c_k . Derivace podle druhé souřadnice se bude řídit symetrickým schématem.

$$\begin{aligned} \frac{\partial E_{edge_mag, \mathbf{v}}^*}{\partial c_{k,x}} &\stackrel{VoDISP}{=} \int \frac{\partial (|\nabla f(\mathbf{v}(t))|^2)}{\partial c_{k,x}} dt \\ &\stackrel{RP}{=} \int \frac{\partial (|\nabla f(\mathbf{v}(t))|^2)}{\partial x} \frac{\partial x(t)}{\partial c_{k,x}} + \frac{\partial (|\nabla f(\mathbf{v}(t))|^2)}{\partial y} \frac{\partial y(t)}{\partial c_{k,x}} dt \\ &= \int \frac{\partial (|\nabla f(\mathbf{v}(t))|^2)}{\partial x} B(t-k) dt \end{aligned}$$

První rovnost je použitím "Věty o derivaci integrálu s parametrem" (předpoklad konvergence integrálu, diferencovatelnosti a majorizovatelnosti funkce je splněn). Druhá rovnost je použití řetězového pravidla pro derivování složené funkce. Druhý sčítanec se derivováním nuluje. Použití zmíněné věty a pravidla už nebude při dalším odvozování pokaždé zdůrazňováno. Abychom mohli výpočet daného integrálu naprogramovat, musíme ho převést na sumu. Vzorkujeme s frekvencí R vzorků na B-Spline úsek a využíváme toho, že $B(t)(= B_n(t))$ koeficient je nenulový pouze na intervalu $(0, n+1)$.

$$\frac{\partial E_{edge_mag, \mathbf{v}}^*}{\partial c_{k,x}} = \frac{1}{R} \sum_{i=0}^{NR} \frac{\partial f_x^2}{\partial x} \left(\mathbf{v} \left(\frac{i}{R} + k \right) \right) B \left(\frac{i}{R} \right)$$

Nejzajímavější a nejdůležitější je derivace termu počítající pravděpodobnost oblastí. Jde totiž o integrál přes větší množinu, než jsou body křivky. Jeho výpočet při

každé iteraci by byl časově náročný. Využijeme tedy Greenovu větu a převedeme integrál přes oblast na integrál přes hranici této oblasti.

$$E_{region,\mathbf{v}}^* = - \int_{int} \underbrace{\log \left(\frac{P_{int}(f(s))}{P_{ext}(f(s))} \right)}_F ds$$

$$\stackrel{Green}{=} \oint_{\partial int} \left(\int F dx \right) \cdot dy dt = \int_{\mathbf{v}} \left(\int F dx \right) \cdot y'(t) dt$$

Derivace tohoto integrálu podle hranice, kterou je B-Spline křivka \mathbf{v} , je už téměř stejně snadná jako derivace termu E_{edge_mag} .

$$\begin{aligned} \frac{\partial E_{region,\mathbf{v}}^*}{\partial c_{k,x}} &\stackrel{VoDISP}{=} \int_{\mathbf{v}} \frac{\partial}{\partial c_{k,x}} \left(\int F dx \right) \cdot y'(t) dt \\ &\stackrel{RP}{=} \int_{\mathbf{v}} \frac{\partial}{\partial x} \left(\int F dx \right) \cdot \frac{\partial \mathbf{v}(t)}{\partial c_{k,x}} \cdot y'(t) dt + \left(\int F dx \right) \cdot \frac{\partial y'(t)}{\partial c_{k,x}} dt \\ &= \int_{\mathbf{v}} F \cdot \frac{\partial \mathbf{v}(t)}{\partial c_{k,x}} \cdot y'(t) dt \tag{5.3} \\ &= \int_{\mathbf{v}} F \cdot B(t-k) \cdot \sum_{l=0}^M c_{l,y} B'(t-l) dt \\ &= \sum_{l=0}^M c_{l,y} \cdot \int_{\mathbf{v}} F \cdot B(t-k) \cdot B'(t-l) dt \\ &\approx \sum_{l=0}^M c_{l,y} \cdot \underbrace{\frac{1}{R} \sum_{i=0}^{NR} F \left(\frac{i+kR}{R} \right) \cdot \underbrace{B(t-k) \cdot B'(t-l)}_{b_{kli}}}_{Q_{kl}} \end{aligned}$$

Pro urychlení výpočtu si termy b_{kli} předpočítáme. Jejich hodnota se v podstatě nemění v průběhu celého algoritmu. Metoda předpočítání některých hodnot ve výrazech, například navzorkováním aktuální křivky, nebo hodnot obrazu, kudy křivka prochází, velice urychlí celý výpočet.

Zdaleka nejsložitěji vypadá derivace termu vnitřní energie. Přesto to není tak složité, jak se zdá. Nejprve se začne roznásobením výrazu uvnitř integrálu E_{int} .

$$\begin{aligned} E_{int,\mathbf{v}}^* &= \int_0^M (x'(t)^2 + y'(t)^2 - c)^2 dt \\ &= \int_0^M x'(t)^4 + y'(t)^4 + 2x'(t)^2 y'(t)^2 dt - \\ &\quad 2c \int_0^M x'(t)^2 + y'(t)^2 dt + \int_0^M c^2 dt \tag{5.4} \end{aligned}$$

Je dobré připomenout, jak vypadá derivace B-Spline křivky podle parametru t .

$$x'(t) = \sum_{i=0}^M c_{i,x} B'(t-i) \quad (5.5)$$

$$y'(t) = \sum_{i=0}^M c_{i,y} B'(t-i) \quad (5.6)$$

Dosazením derivace komponent křivky (5.5) a (5.6) do výrazu (5.4), zderivováním celého výrazu podle $c_{k,x}$ a další úpravou získáme

$$\begin{aligned} \frac{\partial E_{int,v}^*}{\partial c_{k,x}} &= \int_0^M (4x'(t)^3 + 4x'(t)y'(t)^2) \frac{\partial x'(t)}{\partial c_{k,x}} dt - 4c \int_0^M (x'(t)) \frac{\partial x'(t)}{\partial c_{k,x}} dt \\ &= 4 \int_0^M x'(t)^3 \frac{\partial x'(t)}{\partial c_{k,x}} dt + 4 \int_0^M x'(t)y'(t)^2 \frac{\partial x'(t)}{\partial c_{k,x}} dt - 4c \int_0^M (x'(t)) \frac{\partial x'(t)}{\partial c_{k,x}} dt \\ &= 4 \int_0^M \left(\sum_{i=0}^M c_{i,x} B'(t-i) \right)^3 B'(t-k) dt \\ &\quad + 4 \int_0^M \left(\sum_{i=0}^M c_{i,x} B'(t-i) \right) \left(\sum_{i=0}^M c_{i,y} B'(t-i) \right)^2 B'(t-k) dt \\ &\quad - 4c \int_0^M \left(\sum_{i=0}^M c_{i,x} B'(t-i) \right) B'(t-k) dt \\ &= 4 \sum_{l,m,n} c_{l,x} c_{m,x} c_{n,x} \int_0^M B'(t-l) B'(t-m) B'(t-n) B'(t-k) dt \\ &\quad + 4 \sum_{l,m,n} c_{l,x} c_{m,y} c_{n,y} \int_0^M B'(t-l) B'(t-m) B'(t-n) B'(t-k) dt \\ &\quad - 4c \sum_l c_{l,x} \int_0^M B'(t-l) B'(t-k) dt \end{aligned}$$

Zde je jedinou důležitou úpravou převedení součinu sum na trojitou sumu, neboli sumu všech kombinací prvků sum. Opět můžeme výrazy, které nejsou závislé na křivce, předpočítat.

$$h_1(l,m,n) = \int_0^M B'(t-l) B'(t-m) B'(t-n) dt$$

$$h_2(l) = \int_0^M B'(t-l) B'(t-k) dt$$

Na závěr ještě zbývá doplnit derivaci termu vnějších podmínek E_c . Postup je opět úplně stejný jako u E_{edge_mag} . Jeho převedení na sumu, zde kvůli tomu ani neuvádíme.

$$\frac{\partial E_{c,v}^*}{\partial c_{k,x}} = \int_0^M \frac{\partial dist\ field(\mathbf{v}(t))}{\partial x} B(t-k) dt$$

Kritérium zastavení

Existuje celá teorie zabývající se optimalizačními algoritmy. Mnoho z nich je modifikací založenou na snižování gradientu, například metoda konjugovaného gradientu, nebo BFGS. Základní popis algoritmu většinou nezahrnuje podmínku zastavení. Podmínka zastavení algoritmu často může záviset na charakteru řešené úlohy. Nejobvyklejší podmínky zastavení jsou takové, které algoritmus ukončí po:

1. pevném počtu kroků.
2. pevném počtu vyhodnocení optimalizované funkce.
3. splnění podmínky na hodnotu optimalizované funkce.
4. splnění podmínky na hodnotu parametru.
5. splnění jiného kritéria.

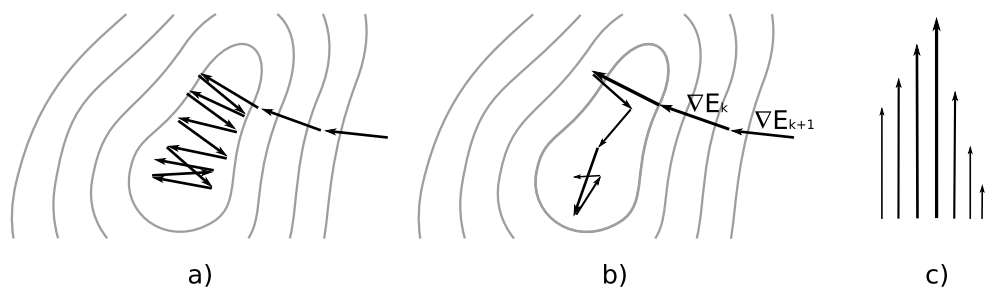
Není třeba zdůrazňovat, že ve správný čas ukončený algoritmus může podstatně urychlit běh celé rekonstrukce. Nejjednodušší je empiricky zjistit, jak dlouho přibližně trvá běh algoritmu na konkrétní třídě úloh a stanovit podle toho maximální počet kroků s jistou rezervou. Algoritmus by měl vždy obsahovat podmínku zastavení po pevném počtu kroků. Zabrání to nekonečné oscilaci, nebo divergenci procesu.

Sledováním hodnoty energetického funkcionálu vede k nárůstu výpočtů v průběhu každého iteračního cyklu. Jednou se provede výpočet derivací a podruhé výpočet hodnoty funkce. Je vhodnější rozhodovat o zastavení podle nějaké relativizované, statistické veličiny, proto se sleduje průměrný rozptyl v malém okénku posledních měření. Sníží se tím závislost kritéria na dané úloze a také náchylnost ke zmatení náhodným šumem. Poklesem hodnoty sledovaného kritéria pod nějakou mez se algoritmus ukončí.

Stejně se dá postupovat i u sledování parametrů c křivky v . Parametry B-Spline křivky jsou poměrně dost nestabilní, protože více výrazně odlišných konfigurací může definovat podobné křivky. Pokud je detekovaná hranice hladká, algoritmus má tendenci, hlavně pod vlivem termu E_{int} , klouzat řídicími body po těchto konfiguracích.

Nejvhodnějším kritériem je odvozený nepřímý ukazatel změn, například obvod, nebo obsah kontury, případně kombinace obou. Sledováním rozptylu v okénku posledních naměřených hodnot můžeme detekovat jejich poklesnutí pod určitou hranici. Naší ukončující podmínkou bude tedy podíl obsahu a obvodu.

Můžeme však pozorovat, a je to logické, že hodnota odvozená z obvodu a obsahu více kolísá, má větší rozptyl, při segmentaci menších objektů. Příčinou je konstantně velký krok δ . Podíl změny při jednom kroku velikosti δ je větší u objektů s méně parametry. Může se tak stát, že algoritmus zkonverguje snáze na větších a složitějších tvarech a na malých nezkonverguje vůbec.



Obrázek 5.10: Konvergence optimalizační metody. **a)** Konvergence s konstantně dlouhým krokem. **b)** Konvergence s adaptivním krokem. **c)** Nerovnoměrné zvětšování a zmenšování kroku ($\alpha = 0.95, \beta = 0.1$). Pohyb ve stejném směru je častější než oscilace, proto prodlužujeme pomaleji, než zkracujeme.

Krok optimalizace

Odpovědí na tento problém je adaptivní krok při snižování gradientu. Modifikujeme vývoj vektoru parametrů \mathbf{c} následujícím způsobem

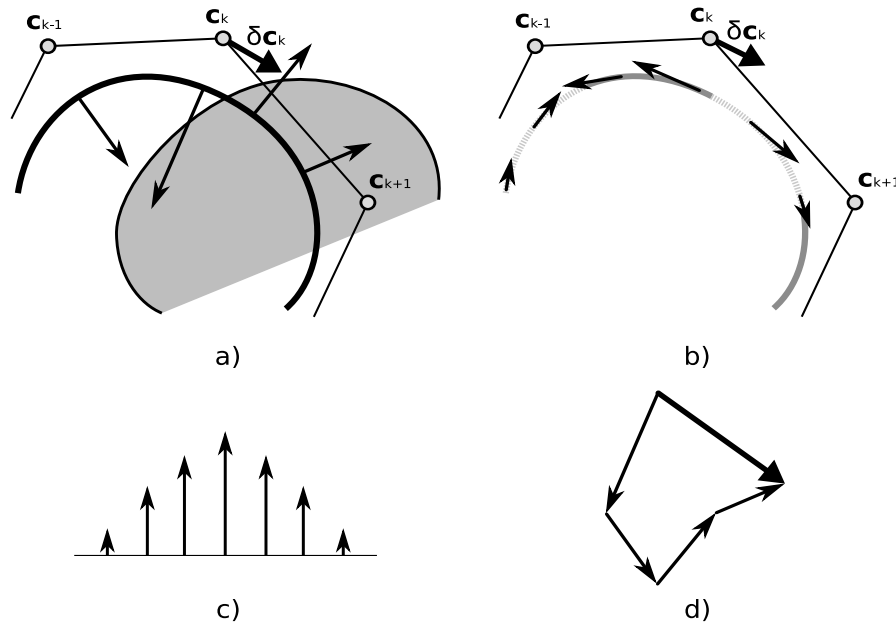
$$\mathbf{c}^{n+1} = \mathbf{c}^n - \delta_n \cdot \text{step}(\mathbf{c}^n)$$

Obrázek 5.10a ukazuje chování algoritmu v krajině parametrů při konstantním kroku. V blízkosti řešení gradient změnil směr v každém kroku téměř o 180° . Zajímavá myšlenka je snižovat nebo zvyšovat omezeně krok podle toho, jaký úhel svírají dva poslední gradienty. Úhel gradientů odpovídá skalárnímu součinu. Pokud je skalární součin záporný, dojde ke snížení, pokud kladný, tak ke zvýšení. Když se úhel gradientu změní například o 90° , zůstane krok zachován. Takto formulovaný adaptivní krok zachytíme hodnotou δ_n následovně

$$\delta_n = \alpha + \beta \cdot \text{step}(\mathbf{c}^{n-1}) \cdot \text{step}(\mathbf{c}^n)$$

Konvergence s adaptivním krokem je vidět na Obrázku 5.10b. Parametr α nastavený blízko 1 ovlivňuje poměr rychlosti zkracování a prodlužování kroku a parametr β zachycuje míru zkracování a prodlužování. Vliv parametrů na zkracování a prodlužování je vidět na Obrázku 5.10c. Pokud se krok snižuje během oscilace gradientu na konci optimalizačního procesu, ke splnění podmínky pro ukončení s jistotou dojde.

V průběhu algoritmu může křivka podstatně změnit tvar, hlavně se může výrazně zkrátit, nebo prodloužit. Musíme proto přidávat nové uzlové body do nejdelších úseků, pokud jejich délka překročí určitou mez. Naopak musíme i odebrat uzlové body mezi nejkratšími úseky, jejichž společná délka klesne pod jistou hranici. K vytvoření nové série parametrů změněné křivky použijeme metodu pro dopočítání řídicích bodů z kapitoly 5.1.1.



Obrázek 5.11: Alternativní konstrukce optimalizačního kroku. **a)** Příspěvky podél vlivu intenzity. **b)** Příspěvky podél vlivu parametrizace. **c)** Vliv koeficientu příspěvku řídicího bodu. **d)** Superpozice příspěvků.

5.1.4 Heuristika

Inovativním jádrem této práce je návrh alternativní konstrukce optimalizačního procesu pro řízení pohybu křivky. Z experimentů se dá vypořádat, že největší vliv na vývoj křivky má energie řízená statistikou oblasti, E_{region}^* . Při studiu chování křivky pod vlivem tohoto termu s různou výchozí polohou si všimneme, že body křivky se pohybují dovnitř a ven ve směru normály podle toho, jestli se nachází v oblasti s malou, nebo velkou pravděpodobností P_{int} . Pokud bychom tedy umístili normálové vektory podél celé křivky, škálovali a orientovali je tak, jak říká apriorní pravděpodobnost a sčítali jejich příspěvky v rámci řídicích bodů, získali bychom vektory posunu pro řídicí body.

Situace je nastíněna na Obrázku 5.11a. Jsou zde vektory naškálované podle velikosti příspěvku. Velikost příspěvku je z části dána B-Spline koeficientem vlivu daného řídicího bodu při výpočtu souřadnice křivky a z části hodnotami P_{int} a P_{ext} . Jak B-Spline koeficient ovlivňuje velikost vektoru, je znázorněno na Obrázku 5.11c. Za předpokladu nehomogenní parametrizace křivky bychom mohli do příspěvku započítat i velikost první derivace. Příspěvky se složí dle principu superpozice a tím získáme výsledný posun řídicího bodu, jak je vidět na Obrázku 5.11d. Evoluční schéma pro řídicí body tedy vypadá

$$\mathbf{c}_i^{n+1} = \mathbf{c}_i^n + \lambda \frac{\delta \mathbf{c}_i^n}{|\delta \mathbf{c}_i^n|}$$

Kde $\delta \mathbf{c}_i^n$ je posun jednoho řídicího bodu \mathbf{c}_i v kroku n . Tento posun se dle

předchozího popisu spočítá

$$\delta \mathbf{c}_i^n = \int \log \left(\frac{P_{int}(f(\mathbf{v}))}{P_{ext}(f(\mathbf{v}))} \right) \cdot B(t-i) \cdot \mathbf{n}(\mathbf{v}(t)) dt$$

Pokud si připomeneme, jak vypadala derivace E_{region}^* potřebná pro optimalizaci snižováním gradientu po použití Greenovy věty, měl integrál (5.3) tvar, který se přesně rovná naší optimalizaci, kdy je pohyb bodů dán superpozicí příspěvků normál.

Zjišťujeme, že i interní energie E_{int}^* může mít podobně jednoduchou interpretaci. Obrázek 5.11b ukazuje, že v případě interní energie může být posun řídicího bodu dán součtem příspěvků ve směru křivky vážený jak B-Spline koeficientem, tak rozdílem skutečné a průměrné délky vzorku. Tento rozdíl kolísá kolem nuly, takže optimálně dlouhý vzorek nepřispívá nijak. Z toho se dá odhadnout chování úseků, které se smršťují a natahují ve směru probíhání křivky.

$$\delta \mathbf{c}_i^n = \int (|\mathbf{v}'(t)|^2 - c) \cdot B(t-i) \cdot \mathbf{v}'(t) dt$$

5.1.5 Algoritmus

Nyní je čas shrnout celý algoritmus rekonstrukce. V předchozím textu jsme prezentovali všechny komponenty, ze kterých se algoritmus složí. Je popsáno jedno schéma postupné segmentace směrem z pólů, které se zdá být nejvhodnější pro segmentaci ledvin. Pro jiné orgány a úlohy mohou být výhodnější jiná schémata. Algoritmus se snaží ušetřit čas a využít výsledku z předchozího výpočtu. Zároveň tu zůstává možnost dalšího urychlení paralelizací dvou nezávislých výpočtů. Výpis 4 uvádí všechny kroky algoritmu.

5.1.6 Měření objemu počítáním voxelů

Měření objemu se realizuje počítáním pixelů uvnitř uzavřených křivek v každém řezu. Pixelu uvnitř křivky získáme algoritmem řádkového vyplňování pro kreslení polygonů (scanline polygon fill). Polygon je tvořen uzavřenou lomenou čarou vzniklou navzorkováním B-Spline křivky.

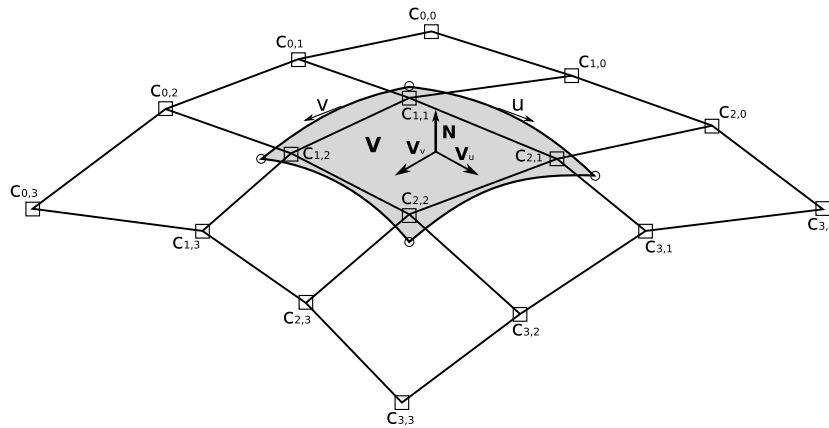
$$Volume(O) = objem_voxelu \sum_{rezy} |pixelu_tvaru_v_rezu|$$

5.2 Rekonstrukce modelováním B-Spline plochy

Algoritmus rekonstrukce postupnou segmentací vrstev je syntézou segmentace dvojrozměrného obrázku a několika vylepšení. Měření objemu se zde provádí jednoduchým počítáním voxelů. Na základě předchozího algoritmu můžeme navrhnout nové schéma, které pracuje plně s trojrozměrnou informací, neodděluje zpracování v jednotlivých řezech. Zároveň přicházíme s úplně jiným způsobem

Algoritmus 4 Rekonstrukce orgánu postupnou segmentací řezů.

1. Volba profilu pro danou třídu orgánů obsahující parametry: meze pro konvergenci, hustotu pravděpodobnosti P_{ext} , P_{int} , váhy energií, mezní velikosti B-Spline úseků, počáteční velikost kroku.
 2. Vstup od uživatele označující póly orgánu: bod p_1 v řezu $start_0$ a bod p_2 v řezu end_0 . Platí $start_0 < end_0$.
 3. Nastavení $i = 0$. Inicializace výchozí křivky v řezu $start_0$ a end_0 tvarem malé kružnice.
 4. Konvergence algoritmu aktivních kontur v řezu $start_i$ a end_i
 - (a) Výpočet $\nabla E_{B-snake}$ podle vztahů popsaných v kapitole 5.1.3.
 - (b) Úprava vektoru parametrů $\mathbf{c}^{n+1} = \mathbf{c}^n - \delta_n \cdot step(\mathbf{c}^n)$.
 - (c) Kontrola průsečíků. V případě křížení, zachová nejdelší uzavřený úsek.
 - (d) Kontrola délky úseků. V případě překročení povolených mezí rozděl/sluč nejdelší/nejkratší úsek/y. Na křivce se nově navzorkují uzlové body a dopočítají se všechny řídicí body (viz 5.1.1).
 - (e) Pokud není splněno konvergenční kritérium zvyš n o 1 a opakuj od 4a.
 - (f) Předpočítání šablony pro počítání vzdáleností $dist_{field}(\mathbf{x}, \mathbf{v})$ od zkonvergované křivky \mathbf{v} .
 5. Propagace z řezu $start_i$ do řezu $start_{i+1}$ a z řezu do end_i do řezu end_{i+1} .
 - (a) Koregistrace - provede se jednoduchá registrace spočívající v posunu a rotaci tvaru (křivky) do polohy s největší apriorní pravděpodobností.
 - (b) Přepočítání statistiky - přepočítá se pravděpodobnost P_{ext} , protože se mění během průchodu orgánu tělem (viz Obrázek 6.4).
 - (c) Inicializace křivky v řezu $start_{i+1}$ křivkou z řezu $start_i$ a křivky v řezu end_{i+1} křivkou z řezu end_i .
 6. Pokud se $start_i \neq end_i$, zvyš i o 1 a opakuj od kroku 4
-



Obrázek 5.12: B-Spline plocha. Ukázka jednoho bikubického B-Spline plátu, který je definován celkově 16 řídicími body.

měření, než je počítání voxelů. Původní segmentaci pomocí B-Spline aktivních kontur rozšíříme na B-Spline aktivní plochy.

5.2.1 Definice B-Spline plochy

B-Spline plocha je definována podobně jako B-Spline křivka. Používá stejné báze funkce. Místo vektoru řídicích bodů používá rovnou celou matici řídicích bodů $\mathbf{c}_{i,j}$. Budeme pracovat s uniformními bikubickými B-Spline plochami, proto ani nemá smysl hovořit o uzlech a matici uzlů. Jak vypadá jeden úsek B-Spline plochy, B-Spline plát, je vidět na Obrázku 5.12. B-Spline plocha má obdélníkovou topologii a bod na ploše v \mathbf{R}^3 je parametrizován dvěma souřadnicemi u a v podle výšky a šířky tohoto obdélníku.

$$\mathbf{V}(u, v) = (x(u, v), y(u, v), z(u, v))$$

Plocha je definována

$$\mathbf{V}(u, v) = \sum_{i=0}^N \sum_{j=0}^N \mathbf{c}_{i,j} B_{n,i}(u) B_{n,j}(v)$$

Budeme také potřebovat derivace plochy podle obou jejích parametrů. Představují tečné vektory ve směru daného parametru.

$$\mathbf{V}_u(u, v) = \sum_{i=0}^N \sum_{j=0}^N \mathbf{c}_{i,j} B'_{n,i}(u) B_{n,j}(v)$$

$$\mathbf{V}_v(u, v) = \sum_{i=0}^N \sum_{j=0}^N \mathbf{c}_{i,j} B_{n,i}(u) B'_{n,j}(v)$$

Vektorovým součinem těchto tečných vektorů získáme vektor \mathbf{N} kolmý na plochu. Normalizací pak normálu \mathbf{n} .

$$\mathbf{N}(u, v) = \mathbf{V}_u(u, v) \times \mathbf{V}_v(u, v)$$

$$\mathbf{n}(u, v) = \mathbf{N}(u, v) / |\mathbf{N}(u, v)|$$

5.2.2 Optimalizace

S využitím heuristiky odvozené interpretací derivace energie řízené statistikou oblasti, jak je popsána v kapitole 5.1.4 odvodíme jednoduchou evoluční rovnici pro výpočet plochy.

$$\mathbf{c}_{i,j}^{n+1} = \mathbf{c}_{i,j}^n + \lambda \frac{\delta \mathbf{c}_{i,j}^n}{|\delta \mathbf{c}_{i,j}^n|}$$

Posun $\delta \mathbf{c}_{i,j}^n$ řídicího bodu $\mathbf{c}_{i,j}^n$ plochy je dán přírůstkou kolmými na plochu vážené součinem B-Spline koeficientů a koeficientem odvozeným z pravěpodobností intenzity bodu, kterým křivka prochází. Součin B-Spline koeficientů je nenulový na omezené podmnožině definičního oboru parametrů u a v .

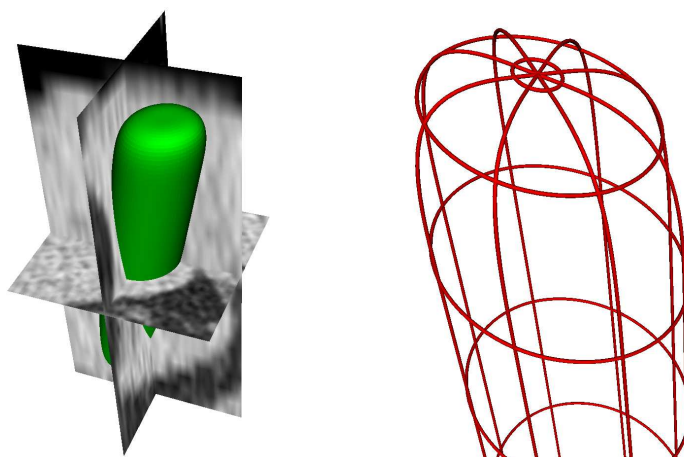
$$\delta \mathbf{c}_{i,j}^n = \iint_{\mathbf{V}} \log \left(\frac{P_{int}(f(\mathbf{V}))}{P_{ext}(f(\mathbf{V}))} \right) \cdot B(i-u) \cdot B(j-v) \cdot \mathbf{N}(u, v) \, dudv$$

5.2.3 Algoritmus a konvergence

Pouze s přírůstkou odvozenými z intenzity oblasti můžeme postavit funkční algoritmus, který zrekonstruuje tvar orgánu podobně jako v algoritmu rekonstrukce segmentací řezů. Inicializace spočívá, stejně jako u předchozího algoritmu, v umístění řídicích bodů ve vzdálených oblastech objektu. Mezi těmito body se vytvoří uzavřená výchozí B-Spline plocha. Topologie tohoto útvaru se během algoritmu měnit nebude, proto musí uživatel předem určit počet řídicích bodů. Poté už běží algoritmus sám.

K vytvoření uzavřené B-Spline plochy, zhruba tvaru protáhlé koule, kapsle, nebo uzavřeného válce, jsou potřeba dvě operace definující její topologii. Nejprve zabalíme plochu v jednom směru podobně, jako jsme to provedli s křivkou. Tedy ztotožníme posledních N sloupců řídicích bodů s prvními N sloupci řídicích bodů, $c_{i, M-N+k} = c_{i,k}$, $0 \leq k \leq N$, $0 < i < M_u$. Vznikne tak válec bez podstav. Zabalněním plochy v obou dvou směrech nevznikne koule ale torus. Pro uzavření válce musíme ztotožnit N prvních řad řídicích bodů do jednoho vrcholu, pólu. Stejně to provedeme i s poslední N -ticí řad a získáme druhý pól. Programově se to realizuje odděleným jednorozměrným polem pro souřadnice bodů a dvourozměrným polem ukazatelů na tyto souřadnice. Výsledek popsání modelování je vidět na Obrázku 5.13.

V průběhu algoritmu plocha nemění topologii, ani počet parametrů. Nekontrolují se křížení plochy samé se sebou. Ani se neoptimalizují velikosti jednotlivých plátů. Algoritmus zkonverguje po pevně daném počtu kroků, nebo když roz-



Obrázek 5.13: Inicializace B-Spline plochy. Umístění výchozí plochy do objemových dat (vlevo). Problematická topologie uzavřené B-Spline plochy v pólech (vpravo).

ptyl objemu v okénku několika posledních měření klesne pod stanovenou hranici. Princip měření objemu je popsán v následující kapitole. Celý algoritmus je shrnut v přehledu 5

5.2.4 Měření objemu z ohraničující plochy

Měření objemu se neprovádí sčítáním voxelů, jako tomu bylo u předchozího algoritmu. Místo toho využíváme poznatků z teorie plošného integrálu. Konkrétně jde o větu o divergenci. Nejprve zopakujeme několik definic.

Definice 1 (Plošný integrál skalární funkce) Pro $f : \mathbf{R}^3 \rightarrow \mathbf{R}$ na ploše S parametrizované \mathbf{V} platí

$$\iint_S f da = \iint_{\mathbf{V}} f(\mathbf{V}(u,v)) \cdot |\mathbf{V}_u \times \mathbf{V}_v| dudv$$

Definice 2 (Plošný integrál vektorové funkce) Pro vektorovou funkci $F = (F_1, F_2, F_3) : \mathbf{R}^3 \rightarrow \mathbf{R}^3$ na ploše S parametrizované \mathbf{V} platí

$$\begin{aligned} \iint_S F da &= \iint_S F \cdot \mathbf{n} da \\ &= \iint_{\mathbf{V}} F(\mathbf{V}(u,v)) \cdot \frac{\mathbf{V}_u \times \mathbf{V}_v}{|\mathbf{V}_u \times \mathbf{V}_v|} \cdot |\mathbf{V}_u \times \mathbf{V}_v| dudv \end{aligned}$$

Definice 3 (Divergence) Pro vektorovou funkci $F = (F_1, F_2, F_3) : G \subseteq \mathbf{R}^3 \rightarrow \mathbf{R}^3$ je divergence definována

$$\nabla \cdot F = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial z}$$

Algoritmus 5 Rekonstrukce orgánu modelováním plochy.

1. Uživatel označí póly orgánu.
2. Uživatel vybere profil s nastavením parametrů, topologie plochy a hustotami pravděpodobností.
3. Nastaví se $k = 0$. Vytvoření výchozí uzavřené B-Spline plochy (Obrázek 5.13).
4. Spočítá se vektor posunutí řídicích bodů $\delta c_{i,j}^k$.
5. Udatují se řídicí body $c_{i,j}^k \rightarrow c_{i,j}^{k+1}$.
6. Proveďte se konsolidace plochy. Spočívá v dopočítání řídicích bodů v pólech průměrováním okolních. To částečně kompenzuje problematickou topologii a zajistí, že plocha nebude mít špičaté póly.
7. Pokud nebylo splněno konvergenční kritérium a neproběhl stanovený počet kroků, zvýš k o 1 a opakuj od kroku 4.

Věta 2 (O Divergenci) Množina G je otevřená mezi grafy funkcí ve všech třech směrech. Plocha $\Phi = \bar{G} \setminus G$ je po částech jednoduchá, regulární, kladně orientovaná (tj. normály směřují vně). Množina $H \supset \bar{G}$ je otevřená. F má spojité derivace prvního řádu a $F : H \rightarrow \mathbf{R}^3$. Pak

$$\iiint_G (\nabla \cdot F) dG = \iint_{\Phi} F dS \quad (5.7)$$

Pomocí věty o divergenci můžeme snadno spočítat objem množiny ohraničené uzavřenou B-Spline plochou. Stačí pouze dosadit vhodnou vektorovou funkci F , aby na levé straně vyšel integrál konstanty přes objem množiny. Taková funkce je

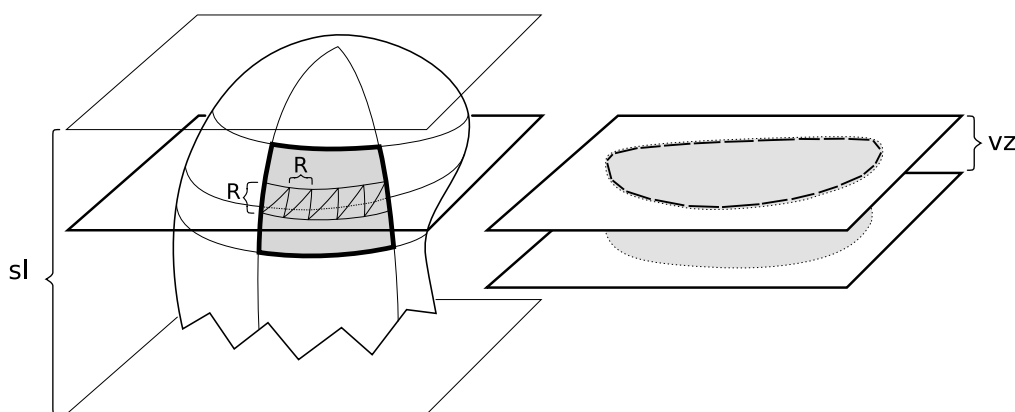
$$F = \left(\frac{x}{3}, \frac{y}{3}, \frac{z}{3} \right)$$

Divergence této funkce vychází

$$\nabla \cdot F = \left(\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \right) = 1$$

Dosazením do věty o divergenci (5.7) dostaneme na levé straně objem množiny G a na pravé straně integrál vektorové funkce F přes plochu ohraničující G .

$$\begin{aligned} \text{Volume}(V) &= \iiint_V 1 dV = \iiint_V (\nabla \cdot F) dV \stackrel{\text{VoD}}{=} \iint_{\partial V} F dS \\ &= \iint_{\partial V} F(T(u, v))(Tu \times Tv) dudv \\ &= \iint_{\partial V} (T_x(u, v), T_y(u, v), T_z(u, v))(Tu \times Tv) dudv \end{aligned}$$



Obrázek 5.14: Řez plochy rovinou. Rovina řeže některé trojúhelníky aproximace.

Diskretizací tohoto výrazu získáme sumu přes navzorkované hodnoty pokrývající tuto plochu. Jako vzorky mohou sloužit například trojúhelníky z trojúhelníkové aproximace plochy.

$$\begin{aligned} \text{Volume}(V) &= \sum_{i=0}^N T_x^i(u, v)N_x^i + T_y^i(u, v)N_y^i + T_z^i(u, v)N_z^i \\ &= \sum_{i=0}^N \nabla T^i \cdot N^i \end{aligned}$$

5.2.5 Převod hraniční reprezentace na objemovou

Na závěr této části o modelování B-Spline plochy ukážeme algoritmus převodu hraniční reprezentace na množinu voxelů, které se nalézají uvnitř uzavřené plochy. Použitím tohoto algoritmu získáme reprezentaci výsledku rekonstrukce, kterou můžeme srovnat s výsledkem rekonstrukce postupnou segmentací řezů. Samozřejmě můžeme i z této reprezentace změřit objem počítáním voxelů.

Algoritmus převede uzavřenou plochu $\mathbf{V}(u, v)$ na síť trojúhelníků. Poté se provede řez rovinami totožnými s rovinami řezů tomografických snímků. Získáme tak kontury ležící v těchto řezech. Vnitřek kontur spočítáme řádkovým vyplňováním, jako při měření objemu u algoritmu postupné segmentace řezů. Výsledkem sjednocení vnitřku všech řezů bude množina voxelů spadající dovnitř uzavřené plochy. Algoritmus je názorně demonstrován na Obrázku 5.14. Přesný popis algoritmu je v přehledu 6.

5.3 Porovnání

V této kapitole jsme představili dva implementované algoritmy, které rekonstruuji tvar orgánu z lékařských dat a umožňují spočítat jeho objem. Oba dva jsou založené na stejném principu deformovatelných modelů. Rozdílné jsou například ve způsobu

Algoritmus 6 Získání objemové reprezentace z hranice reprezentované plochou.

1. Zjistí se nejbližší a nejvzdálenější řídicí bod \mathbf{c}_n a \mathbf{c}_f k rovině 0. Předpokládá se, že objekt leží nad rovinou 0.
2. Vytvoř pole seznamů pro úsečky *LinesVects* předpokládané velikosti $sl = (\mathbf{c}_{f,z} - \mathbf{c}_{n,z})/vz$.
3. Projdi plochu \mathbf{V} se vzorkovací periodou R odpovídající požadované přesnosti.
 - (a) Vzorek (i, j) tvoří dva trojúhelníky $t_1 = \{(i, j), (i + R, j), (i + R, j + R)\}$ a $t_2 = \{(i, j), (i, j + R), (i + R, j + R)\}$.
 - (b) Zjisti vzdálenosti vrcholů t_1 od roviny 0.
 - (c) Do seznamu *LinesVects*[i] vložíme průsečnice trojúhelníku t_1 s rovinami i pro $\forall i$ "rovina nad nejbližším vrcholem" $\leq i \leq$ "rovina pod nejvzdálenějším vrcholem". Průsečnice je úsečka spojující průsečky dvou stran trojúhelníku s rovinou.
 - (d) Zopakuje 3c pro t_2 .
4. Každý neprázdný seznam úseček tvoří uzavřenou lomenou čáru. Proveďte se vyplňování polygonu po řádcích. Každý takto získaný pixel vlož do množiny voxelů.

práce s daty, kdy první pracuje v oddělených řezech a druhý se pohybuje najednou ve všech třech dimenzích. Nejzajímavější je však odlišnost v počítání objemu z různých reprezentací. První metoda počítá přímo voxely objemových dat. Druhá na rozdíl od toho integruje hranici orgánu reprezentovanou B-Spline plochou. Porovnání výsledků měření vlastností, přesnosti a rychlosti algoritmů jsou shrnuty v následující kapitole.

Kapitola 6

Měření a výsledky

V případě nasazení popsaných metod v lékařské praxi musíme dbát na několik důležitých kritérií. Jsou jimi především rychlost, robustnost a přesnost. Rychlost je důležitá, protože například lékař pracující se softwarem, který je založen na našich algoritmech, si nemůže dovolit čekat příliš dlouho na standardní vyšetření. Jiná situace může nastat při hromadném zpracování dat, kdy se každá ušetřená sekunda při zpracování jedné úlohy násobí v ušetřené minuty při zpracování všech dat.

Robustnost se dá posuzovat ze dvou hledisek. Především by měl algoritmus uspokojivě fungovat na co nejširší třídě vstupních dat dané úlohy. Na druhou stranu je algoritmus stále závislý na vstupu od uživatele, proto by měl správně fungovat i s málo přesným uživatelským vstupem. Měl by dávat stejné výsledky na stejných datech pro různé vstupy.

Protože jde o měření objemu, rekonstrukce je pouze prostředkem k jeho zjištění. V testech se budeme soustředit na chybu měření a ne na správnost rekonstrukce. V určitých situacích není důležitá ani přesná hodnota měření. Můžeme pracovat i s pouhou změnou této hodnoty na různých snímcích jednoho sledovaného subjektu. Pokud by se nám podařilo odhalit relativní chybu našeho algoritmu na větší množině subjektů, můžeme správnou hodnotu objemu odvodit i z chybného měření.

Výsledky rekonstrukce a měření algoritmy budeme srovnávat jednak mezi sebou a jednak s měřením manuální segmentací provedenou člověkem. Ani měření provedené člověkem nemusí být úplně přesné. Vždy záleží na zkušenostech, znalostech, kvalitě snímků a kondici uživatele, který měření provádí. Jiné výsledky bude generovat každý jednotlivý uživatel. Dokonce i jeden uživatel bude generovat různé výsledky v jinou denní dobu, v jiný pracovní den, nebo například i v závislosti na tom, jestli uživatel už obědval, nebo ještě ne. Z tohoto pohledu se do měření zavádí subjektivní faktor a to je věc, kterou by náš systém pro automatické měření objemu orgánů měl eliminovat. Bohužel jsme srovnání této subjektivity manuálního měření do našich testů nezahrnuli. Vyžádalo by si to účast větší skupiny školených uživatelů.

Pro manuální segmentaci využijeme tu hlavní výhodu algoritmu postupné segmentace řezů, že uživatel může snadno kontrolovat tvar křivky v každém řezu pomocí uzlových bodů. Nejprve tedy necháme zkonvergovat běžný algoritmus a

výsledek pak ručně domodelujeme. Měření objemu z manuální segmentace se provádí stejně jako u automatického algoritmu.

Pro každou specifickou úlohu měření je třeba jinak nastavit parametry algoritmů. Během měření celé množiny testovacích dat se parametry dál nemění.

6.1 Použitá data

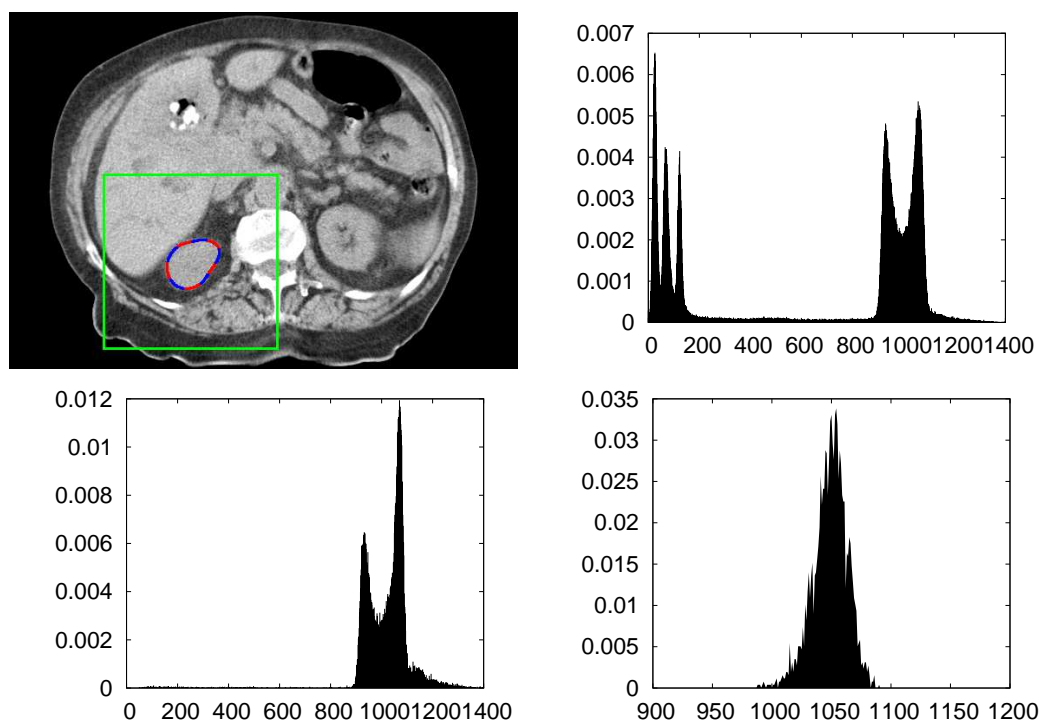
Doktorem Horákem nám byla poskytnuta data anonymních pacientů různé kvality a vlastností. U CT snímků je rozhodujícím atributem tloušťka řezu. Při práci s tomografem je možné nastavit ještě mnoho jiných parametrů (kolimace, filtr, expozice, ...) My jsme se omezili na řezy o tloušťce 5 milimetrů, které se v současné době používají při standardním vyšetření břišní dutiny a hrudníku. Ve speciálních případech je možné pořídit i jemnější data. Není pochyb o tom, že s technickým pokrokem se bude snižovat šířka řezů pro běžné standardní vyšetření. Je jasné, že přesnost výsledků závisí na rozlišení těchto dat.

Pro testování jsme vybrali datové série dvanácti pacientů z ne o mnoho větší množiny dostupných dat. Většinou se jedná o snímky starších a nemocných lidí. To se na datech projevuje různými vadami, od deformací, srůstů až po chybějící orgány, které bychom rádi měřili. Vzhledem k tomu, že vyšetření podstupují převážně nemocní lidé, je náš test v podstatě velice reálný. Data navíc často obsahují další vady typické pro CT snímkování. Projevuje se zde silně partial volume efekt. V některých případech můžeme pozorovat hvězdicový artefakt, způsobený například kontrastní látkou v sousedních orgánech, ve střevech a žaludku. Možná bychom mohli počítat s lepšími daty, kdyby operátor tomografu předem věděl, k jakému účelu je chceme použít.

Jednotlivé série označíme *pacient 01* až *pacient 12*. U všech pacientů se pokusíme změřit pravou ledvinu a slezinu. Měření provedeme jak metodou postupné segmentace řezů, tak modelováním B-Spline plochy.

6.2 Měření ledvin

Nejprve se pokusíme změřit objem ledvin všech dvanácti pacientů. Naštěstí všichni mají pravou ledvinu na svém místě. Ledvina má z vnějšího pohledu hezký a jednoduchý fazolovitý tvar, takže se zdá, že měření bude bez problémů. Navíc se nalézá v části těla obalená kontrastním tukem, orientovaná ve směru osy. Zajímavou výjimku tvoří *pacient 10*, jehož pravá ledvina je lehce vytočená dolním koncem dopředu pod úhlem 45° . V centrální části ledviny se nachází takzvaná pánvička (*renal pelvis*), na kterou navazuje močovod. V prostoru pánvičky se napojuje také žíla a tepna. Tento prostor má hustotu okolního tuku a někdy je na snímcích výrazně vidět a někdy téměř ne. Naše algoritmy si se složitější topologií poradit neumí. V řezu, kde je pánvička obklopena tkání ledvin, ji algoritmus zahrne jako součást orgánu. V řezech, kde je pánvička spojena s okolním prostorem, ji algoritmus správně "vykrojí", viz Obrázek 6.4.

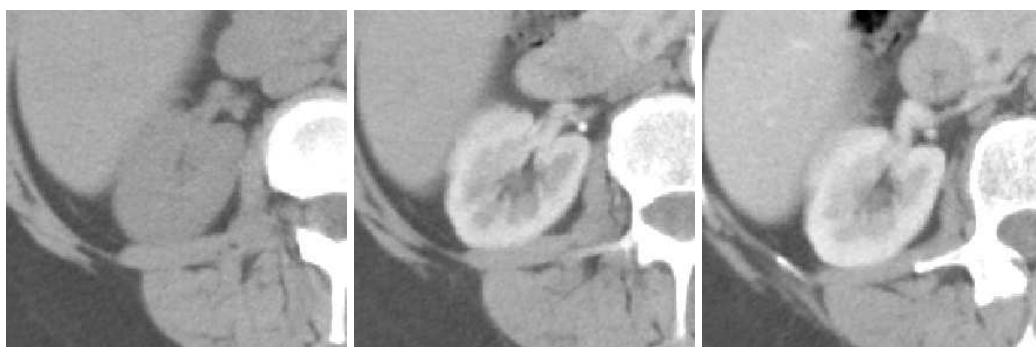


Obrázek 6.1: Statistika oblasti. První graf (vpravo nahoře) zachycuje rozložení pravděpodobnosti v celém obrázku vně ohraničeného objektu. Druhý graf (vlevo dole) ukazuje rozložení, pokud se omezíme pouze na blízké okolí objektu znázorněné obdélníkem v obrázku. Poslední graf (vpravo dole) je předpokládané rozložení intenzity uvnitř objektu.

Z chování popsaném v předchozím textu je vidět, jak se algoritmus řídí očekávanou intenzitou oblasti. Informace o očekávané intenzitě musí být předem známa a slouží jako vstupní parametr pro oba diskutované algoritmy. Nejlepších výsledků dosáhneme, když provedeme jednoduché přeměření intenzity v jednom řezu série. Měření intenzity spočívá v konstrukci histogramu intenzity v oblasti uvnitř a vně orgánu. Ze závislosti na intenzitě celé oblasti vyplývá důležité omezení. Algoritmy takto řízené dokáží dobře segmentovat objekty s homogenní intenzitou.

Pro účel rekonstrukce musí být rozložení intenzity uvnitř objektu stejné ve všech řezech. U rozložení intenzity vně objektu se řídíme jinými pravidly. Lepší výsledky při segmentaci dostaneme, když statistiku počítáme pouze z blízkého okolí objektu, takzvané oblasti zájmu (ROI - region of interest). V opačném případě se pravděpodobnost rozmělní na širším intervalu stupnice intenzity. Tím se sníží citlivost rozhraní při pohybu přes méně pravděpodobné oblasti. Na Obrázku 6.1 je vidět, jak vypadá statistika celého snímku a jak se liší při omezení na oblast zájmu.

Poslední užitečnou vlastností snímků ledvin je použití kontrastní látky během standardního vyšetření. Kontrastní látka na bázi jódu je pacientovi aplikována injekcí do krevního oběhu. Do ledviny se dostává z aorty, ledvina ji postupně filtruje a poté vyloučí v moči. Proces nasycení trvá 1 až 2 minuty. Vyšetření většinou za-



Obrázek 6.2: Fáze nasycení kontrastní látkou. Nativní fáze (vlevo) demonstruje homogenní intenzitu orgánu. Během arteriální fáze (uprostřed) je ledvina silně prokreslená, což znesnadňuje práci popsaným algoritmům. V průběhu venózní fáze (vpravo) je orgán homogenně nasycen kontrastní látkou.

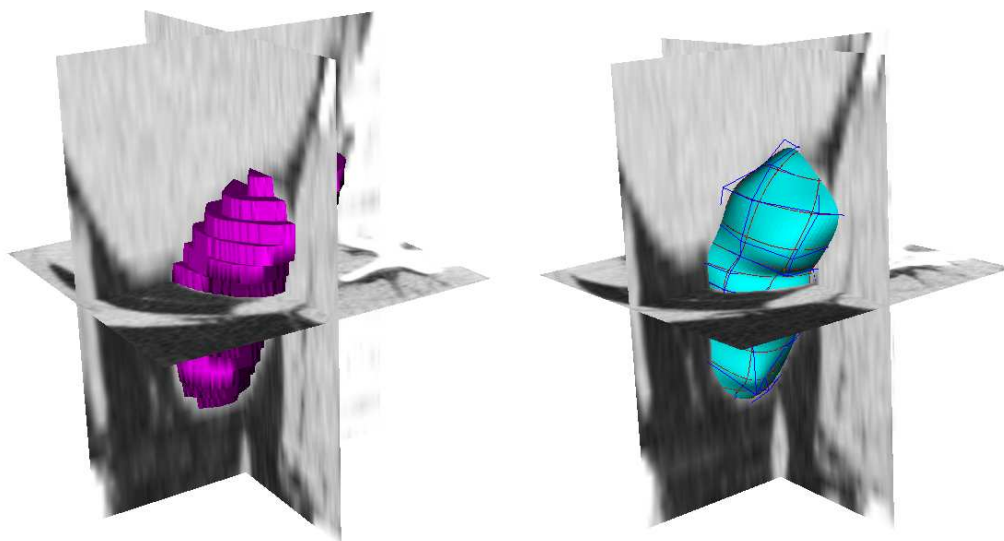
hrnuje tři skenování tomografem v různých fázích nasycení kontrastní látkou. Kontrastní látka způsobí zvýšení kontrastu ledviny na snímku. Díky tomu se objeví i struktury, které na snímku bez kontrastní látky vidět nejsou, jako například cysty, nádory, nebo rozdělení tkáně ledviny na dřev a kůru. Bohužel to naruší základní předpoklad pro fungování našich algoritmů, že segmentovaný orgán musí mít homogenní rozložení intenzity. Vnáší to také další faktor do procesu měření, kterým je správné načasování skenů po aplikaci kontrastní látky. Měření na snímcích zvýrazněných kontrastní látkou je realizovatelné a může dávat i lepší výsledky. My se v testech soustředíme především na měření v datech bez kontrastní látky. Pro lepší představu ukazuje Obrázek 6.2 jednotlivé fáze nasycení.

6.3 Měření sleziny

Slezina je orgán, který má z vnějšího pohledu složitější tvar než ledvina. Je o něco větší a mění svůj tvar poměrně výrazně řez od řezu. Navíc intenzita tkáně sleziny je velice blízká intenzitě tkáně okolních orgánů, srdce, střev nebo svalů. Na druhou stranu na snímcích není vidět, že by měla nějakou vnitřní strukturu. Význam měření sleziny je dost sporný, protože je známo, že slezina mění svoji velikost v závislosti na činnosti organismu. Tvoří jakousi vyrovnávací nádrž na červené a bílé krvinky a zároveň je to místo jejich zániku. Velikost je ovlivněna zánětem či infekcí, věkem, nemocí krve. Například i intenzivní práce svalů způsobí její odkrvení. *Pacient 12* z testovací množiny nemá slezinu vůbec a snímek *pacienta 10* byl tak poškozený, že nebylo možné orgán změřit.

6.4 Výsledky segmentace a rekonstrukce

Výslednou rekonstrukci, na které je založen náš postup pro měření objemu je možné vizualizovat. Oba dva prezentované postupy měření rekonstrukci provádějí. Na



Obrázek 6.3: Výsledek rekonstrukce ledviny. Rekonstrukce postupnou segmentací řezů (vlevo). Rekonstrukce modelováním B-Spline plochy (vpravo).

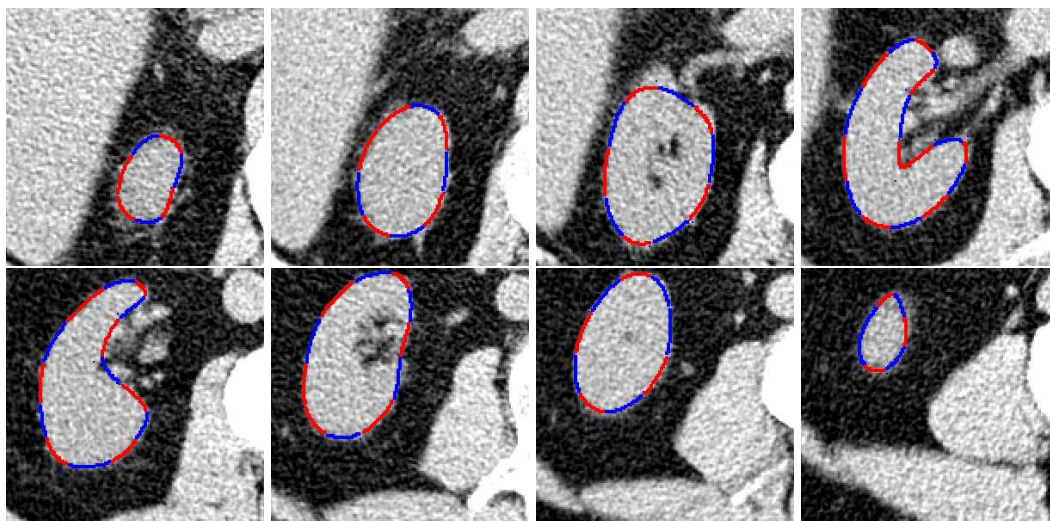
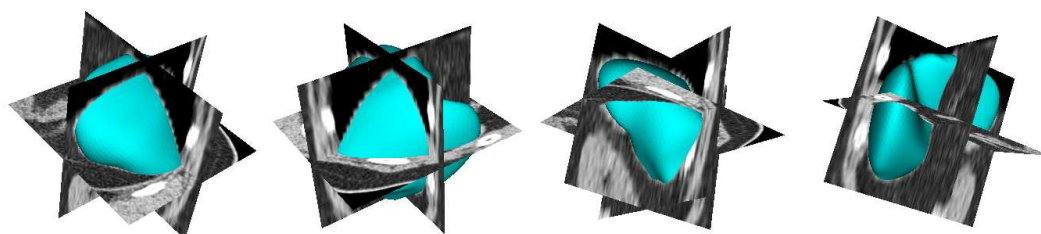
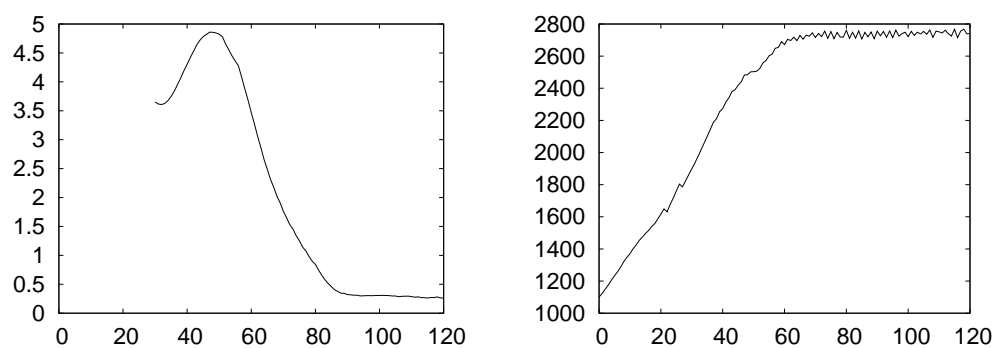
Obrázku 6.3 je vidět rekonstrukce ledviny pomocí obou těchto metod. K zobrazení dat používá náš program knihovnu VTK. Tato knihovna implementuje marching cubes pro zobrazení objemových dat, které jsou výstupem algoritmu postupné segmentace řezů. Poskytuje také prostředky pro zobrazení hraniční reprezentace, sítě trojúhelníků, které jsou výstupem algoritmu modelování B-Spline plochy.

Na Obrázku 6.4 je vidět, jak vypadá rekonstrukce ledviny postupnou segmentací v několika vybraných řezech. Na závěr ještě uvádíme rekonstrukci sleziny modelováním plochy, která je zobrazena na Obrázku 6.5.

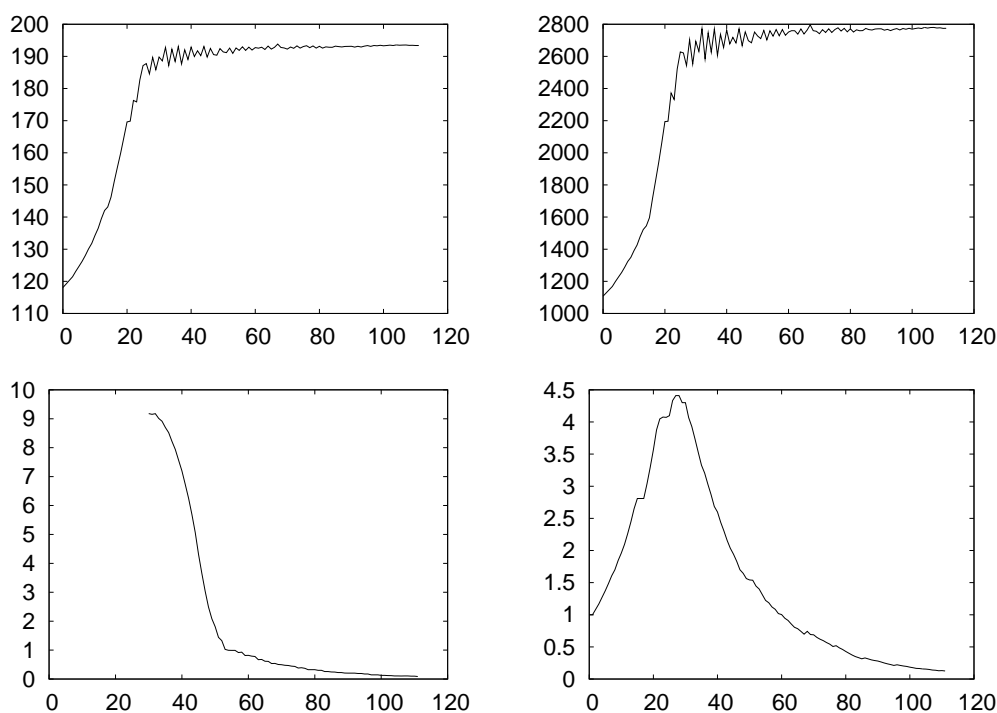
6.5 Rychlost, přesnost a konvergence

Testy byly prováděny na počítači Athlon 2600+ s 1GB RAM v OS Linux. Už v kapitole 5.1.3 bylo řečeno, že rychlost algoritmu je odvozena od rychlosti konvergence. Bylo vysvětleno, podle jakých kritérií lze konvergenci detekovat. Obrázek 6.6 ukazuje, jak vypadá průběh konvergence procesu optimalizace v prvním algoritmu, kde je použit konstantní krok ve směru gradientu. Práh, kolem kterého se ustálí konvergenční kritérium, se liší podle velikosti úlohy. Proto je nutné spolehnout se na zastavení algoritmu po pevném počtu kroků. Na Obrázku 6.7 je vidět jak vypadá proces v případě použití adaptivního kroku. Kritérium zde klesá až k nule. Díky tomu je možné odhalit konvergenci pomocí téměř libovolně malého prahu a ukončit algoritmus dřív. Můžeme navíc pozorovat, že algoritmus s adaptivním krokem konverguje rychleji. Už po 60. iteraci je konvergenční kritérium téměř ploché, kdežto u konstantního kroku se graf zlomí až po 80. iteraci.

Tabulka 6.1 ukazuje výsledky testu měření pravé ledviny. Měření metodou postupné segmentace řezů dopadlo celkem uspokojivě. Ošklivá a poškozená data

Obrázek 6.4: Postupná segmentace řezů pravé ledviny (*ren dexter*).Obrázek 6.5: Rekonstrukce sleziny (*splen*) modelováním B-Spline plochy

Obrázek 6.6: Konvergence aktivních kontur s konstantním krokem. První graf (vlevo) ukazuje, že stejné konvergenční kritérium jako u adaptivního kroku se ustálí na jedné hladině. Tato hladina je navíc závislá na aktuální velikosti křivky. Je to způsobeno oscilací hodnoty obsahu oblasti ke konci běhu algoritmu, jak ukazuje druhý graf (vpravo).



Obrázek 6.7: Konvergence aktivních kontur s adaptivním krokem. První graf (vlevo nahoře) ukazuje změnu délky během vývoje kontury. Druhý graf (vpravo nahoře) zachycuje obsah. Třetí graf (vlevo dole) demonstruje, jak konvergenční kritérium klesá, když se obsah a obvod ustálí. Na posledním grafu (vpravo dole) je vidět velikost adaptivního kroku v průběhu algoritmu.

dávají horší výsledky. Chyba se tak pohybuje mezi 1-4%. Mnohem hůř dopadla metoda modelování B-Spline plochy. V některých případech se nepovedlo orgán změřit vůbec. B-Spline plocha zdeformovala nepřípustným způsobem, přičemž křížila sama sebe. Je to zapříčiněno hlavně větší složitostí vnitřní struktury ledviny. Na metodě modelování B-Spline plochy je ještě mnohé co zlepšovat. Při měření jsme se snažili, aby oba algoritmy měly stejné startovní podmínky, tedy stejná data a vstupní parametry. Ukázalo se, že v některých případech bylo dosaženo lepších výsledků na snímcích s kontrastní látkou, viz níže.

Měření sleziny shrnuté v tabulce 6.2 dopadlo celkově hůř. Ukazuje se na něm, že modelování B-Spline plochy dává lepší výsledky než první metoda. Mohou za to větší rozdíly ve tvaru mezi jednotlivými řezy, kterým nevyhovuje naše schéma pro postupnou segmentaci. Modelování plochy používá ke stanovení hranice vhodnějším způsobem informaci napříč řezy. Na druhou stranu, pokud nebudeme v prvním algoritmu inicializovat nové křivky křivkami z propagovaného řezu, tedy zbavíme se dobrovolně informace ze sousední řezu, dosáhneme lepších výsledků. Naměřená chyba je i tak za hranicí únosnosti. K obhajobě špatných výsledků je třeba dodat, že i ruční segmentace sleziny není úplně jednoznačná. Na snímcích je špatně vidět její hranice, splývá s ostatními orgány a je rozmazaná kvůli

Pacient	manuál	slicealg			surfacefit			Rozdíl metod (%)
	Objem (cm^3)	Objem (cm^3)	Chyba (%)	Čas (s)	Objem (cm^3)	Chyba (%)	Čas (s)	
01	145,75	143,25	-1,71	27,31	144,13	-1,11	26,38	+0,61
02	178,60	181,27	+1,50	28,60	algoritmus selhal			
03	193,61	193,55	-0,03	26,67	190,47	-1,62	35,51	-1,59
04	132,14	129,64	-1,89	21,38	123,93	-6,21	25,08	-4,40
05	126,76	124,05	-2,14	23,21	111,67	-11,91	42,68	-9,98
06	195,79	189,88	-3,02	23,34	197,12	0,68	23,41	+3,82
07	101,81	97,99	-3,75	19,81	96,42	-5,30	17,86	-1,61
08	250,46	250,94	+0,19	29,44	264,66	+5,67	18,42	+5,47
09	150,62	156,56	+3,94	25,23	157,75	+4,73	40,68	+0,76
10	159,99	159,28	-0,44	27,64	150,66	-5,83	64,13	-5,42
11	207,31	203,61	-1,79	29,06	200,61	-3,23	59,85	-1,47
12	157,12	158,63	+0,96	21,32	162,00	+3,11	24,50	+2,13
Střední hodnota (%)			-0,68			-1,91		-1,38
Směrodatná odchylka (%)			2,15			5,30		4,50

Tabulka 6.1: Výsledky měření pravé ledviny (*ren dexter*) u dvanácti subjektů.

partial volume efektu.

Výsledná rychlost algoritmů není optimální pro praktické použití. V případě postupné segmentace řezů je přímo závislá na velikosti úlohy. Neboli jak velký orgán se měří, tak dlouho rekonstrukce trvá. V naší implementaci výpočet nejvíce zatěžuje měření vzdáleností k šabloně *distfield*, potřebné při vyhodnocení ∇E_c^* . Vynecháním tohoto termu se doba výpočtu zkrátí skoro na polovinu. Modelování B-Spline plochy vyžaduje znát na počátku měření topologii plochy a počet řídicích bodů. Při pevně daném vzorkování je délka jedné iterace konstantní. Celková doba výpočtu je tak závislá hlavně na rychlosti konvergence. Jeden krok konvergence se počítá poměrně dlouho. Při této konfiguraci je ve většině případů modelování plochy pomalejší než postupná segmentace řezů.

6.6 Diskuse

Dále jsme provedli několik testů, při kterých jsme zkoušeli robustnost a stabilitu, především u algoritmu postupné segmentace. První test se soustředil na vyhodnocení rozdílu při konvergenci algoritmu s adaptivním a s konstantním krokem. Test jsme provedli na menší skupině pacientů a výsledky shrnuje Tabulka 6.3. Přes omezený rozsah testování jsme vyvodili dva možné závěry. V prvním případě dává výpočet s konstantním krokem stejné výsledky, jen konverguje delší dobu (pacient 03 a pacient 11). Je totiž zastaven až po vykonání maximálního počtu kroků. V druhém případě uvízne algoritmus v lokálním minimu a dá horší výsledek (pacient 05 a pacient 08). Přitom se může stát (pacient 08), že zkonverguje dříve než

Pacient	manuál	slicealg			surfacefit			Rozdíl metod (%)
	Objem (cm^3)	Objem (cm^3)	Chyba (%)	Čas (s)	Objem (cm^3)	Chyba (%)	Čas (s)	
01	312.40	309,23	-1,01	10,44	299,69	-4,07	96,83	-3,09
02	132.45	143,69	+8,49	9,19	135,09	+1,99	41,03	-5,99
03	234.41	220,97	-5,73	11,43	219,29	-6,45	80,18	-0,76
04	193.09	198,61	+2,86	14,00	193,63	+0,28	56,70	-2,51
05	316.19	331,84	+4,95	14,40	263,87	+0,28	41,34	-20,48
06	280.51	272,00	-3,04	12,77	256,23	-8,66	44,76	-5,80
07	116.09	120,93	+4,17	6,17	117,40	+1,12	47,78	-2,93
08	1027.26	1019,19	-0,79	14,66	1016,64	-1,03	70,51	-0,25
09	118.40	111,70	-5,66	18,28	111,63	-5,72	58,30	-0,06
10	poškozená data							
11	442.50	444,68	+0,49	16,66	448,78	+1,42	73,30	+0,92
12	slezina nenalezena, pacient po splenektomii							
Střední hodnota (%)			+0,47			-2,08		-4,09
Směrodatná odchylka (%)			4,66			3,81		6,21

Tabulka 6.2: Výsledky měření sleziny (*splen*) u dvanácti subjektů.

v případě adaptivního kroku, což je způsobeno tím, že během měření se optimalizuje menší počet parametrů. Výsledný objem je pak zákonitě menší. Adaptivní krok tedy urychluje výpočet a částečně brání uvíznutí v lokálním minimu.

Hodnota prahu konvergenčního kritéria se také významně podílí na rychlosti a přesnosti výpočtu. Provedli jsme opět test konvergence s adaptivním krokem a různými prahy na menší vybrané množině pacientů. Výsledky jsou shrnuty v Tabulce 6.4. Je zde vidět, jaký vliv má rostoucí práh na přesnost výsledku a na rychlost konvergence. Jako rozumná hranice se nám jeví práh 0.1. Výsledkem prahů větších než 1.0 je už viditelně nepřesná rekonstrukce.

Jak už bylo řečeno, některé snímky jsou buď poškozené, nebo na nich jsou orgány nevýrazně separované meziorgánovým tukem. V takových případech se

Pacient	Adaptivní krok		Konstantní krok		Rozdíl	
	Objem (cm^3)	Čas (s)	Objem (cm^3)	Čas (s)	Objem (%)	Čas (%)
03	193,55	26,85	193,62	29,56	+0,04	+10,09
05	124,05	23,25	117,20	29,51	-5,52	+26,92
08	250,94	29,63	246,06	27,63	-1,95	-6,75
11	203,61	29,16	201,64	31,45	-0,97	+7,85

Tabulka 6.3: Porovnání výsledků algoritmu postupné segmentace pravé ledviny při použití adaptivního kroku optimalizace. Práh konvergence je nastaven na 0.1, parametry adaptivního kroku jsou $\alpha = 0.99$, $\beta = 0.1$ a maximální počet kroků je 250.

Pacient	Veličina	Práh					
		0	0.05	0.1	0.15	0.4	1.0
01	Objem	144,17	142,41	143,25	143,02	142,31	137,63
	Čas	32,78	30,4	27,31	25,45	19,33	15,27
02	Objem	183,26	184,62	181,27	179,68	177,69	177,38
	Čas	37,09	33,98	28,60	26,65	21,71	19,96
06	Objem	192,84	192,57	189,88	183,78	182,61	176,65
	Čas	27,29	26,72	23,34	21,68	16,07	13,45
12	Objem	159,27	158,87	158,63	158,20	153,31	153,70
	Čas	24,69	22,39	21,32	19,66	14,14	11,51

Tabulka 6.4: Výsledky konvergence postupné segmentace pravé ledviny s různým prahem konvergenčního kritéria za použití adaptivního kroku.

Pacient	Nativní série		Kontrastní série		Rozdíl objemu (%)
	Objem	Čas	Objem	Čas	
02	181,27	28,60	165,71	30,74	-8,58
04	129,64	21,38	120,09	18,80	-7,37
09	156,56	25,23	150,86	23,08	-3,64
10	159,28	27,64	157,95	27,14	-0,84

Tabulka 6.5: Měření pravé ledviny v kontrastní a nativní sérii na výběru pacientů.

můžeme u ledviny pokusit změřit objem na snímcích ve venózní fázi nasycení kontrastní látkou. Jaké jsou výsledky ve srovnání s měřením v nativních datech? Kontrastní látka zvýší kontrast orgánu na snímcích a umožní tak algoritmu odlišit ho od okolních tkání, nezvýrazní však každou část stejně. Tabulka 6.5 ukazuje, že výsledek měření na nativních snímcích je o 3-6% menší, než nativní případ. Pacient 10 je výjimka, zřejmě kvůli nestandardní orientaci orgánu. Tento rozdíl by se dal vysvětlit tím, že kontrastní látka prokreslí vnitřní strukturu ledviny a oddělí v ní tkáň, která se ale v nativní sérii měří jako součást orgánu.

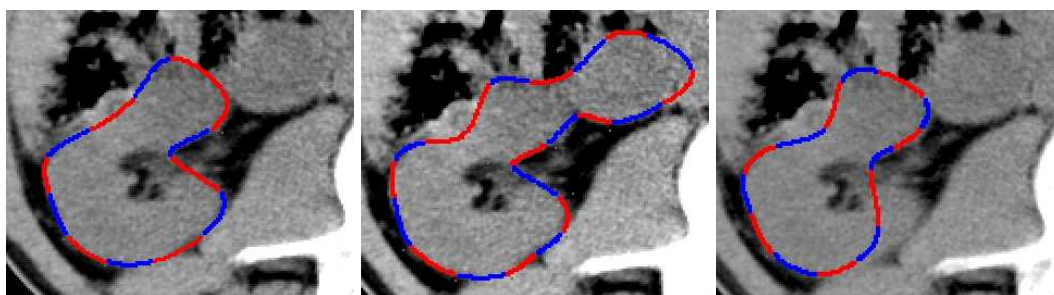
Nakonec jsme zkusili test invariance. Protože metoda je závislá na počátečním vstupu od uživatele, je užitečné provést více měření jednoho orgánu a zjistit, jak se výsledky liší. Odhalíme tím, jak je metoda invariantní k počátečnímu uživatelskému vstupu, volbě pólů. Tabulka 6.7 ukazuje výsledky sedmi měření metodou postupné segmentace na všech pacientech. Výsledky prokazují, že metoda je docela stabilní. Všechna měření u jednoho pacienta se prováděla s jednou předměřenou statistikou rozložení intenzity.

Zkusili jsme ještě vliv předměření statistiky, která je také součástí rekonstrukční procedury. U vybraných pacientů jsme provedli pět měření s různou statistikou jak na seriích v nativní, tak kontrastní fázi. Výsledky v Tabulce 6.6 ukazují, že metoda je na předměření statistiky citlivější, než na volbu pólů. U nativní série výrazněji než u kontrastní. Míra invariance se u kontrastní série blíží invarianci k volbě pólů.

Výsledky měření ukazují, že algoritmus modelování B-Spline plochy není do-

Pacient	Série	Měření objemu (cm^3)					Odchylka (%)
		1	2	3	4	5	
02	Nativní	173.73	173.24	184.38	172.37	186.36	3.80
	Kontrastní	163.22	165.06	169.74	162.67	160.39	2.14
04	Nativní	133.45	130.54	126.02	126.42	128.44	2.39
	Kontrastní	119.83	119.19	119.44	119.93	119.98	0.29
09	Nativní	158.30	146.41	157.62	159.85	155.27	3.43
	Kontrastní	151.08	152.17	155.71	151.20	150.88	1.33
10	Nativní	153.01	141.52	149.32	148.98	151.67	2.99
	Kontrastní	156.31	154.56	157.12	155.05	157.67	0.85

Tabulka 6.6: Invariance měření pravé ledviny vůči statistice předměřené uživatelem na vybraných kontrastních i nativních sériích.



Obrázek 6.8: Dopad propagace v algoritmu postupné segmentace řezů. Správně segmentovaný řez ledvinou (vlevo). Segmentace bez vlivu termu E_c^* (uprostřed). Výsledek segmentace bez přepočítání hustoty pravděpodobnosti P_{ext} (vpravo).

statečně zralý, aby se mohl rovnat postupné segmentaci. Jde však o zajímavý, alternativní přístup z pohledu měření objemu. Postupná segmentace řezů je značně složitější. Pro rekonstrukci je klíčový krok propagace, který je vlastně originálním přínosem této práce. Na závěr bychom podtrhli vliv akcí probíhajících v rámci propagace výsledků mezi řezy. Přepočítání hustoty pravděpodobnosti, se dá přirovnat metodám hierarchického prahování. Jeho vliv na výslednou segmentaci jediného řezu je vidět na Obrázku 6.8 spolu s vlivem vazby na sousední řezy.

Implementované algoritmy jsme vyzkoušeli na množině reálných testovacích dat. Především výsledky rekonstrukce ukazují na použitelnost daných algoritmů. Měření odhalila předpokládanou chybu pro případy dvou orgánů, ledviny a sleziny. Další měření těchto orgánů tak mohou být zpřesněny díky tomuto údaji. Pokud by měly být algoritmy použity pro změření jiných orgánů, bude třeba provést nejprve sérii měření, jako v našem experimentu, s konkrétními parametry na testovací množině a určit chybu. Teprve pak je možné vyhodnotit přesnost dané metody.

Výsledkem experimentů je zjištění, že algoritmus postupné segmentace řezů se dá použít celkem uspokojivě k měření ledvin. Na rozdíl od toho modelování ploch ve své momentální podobě není příliš spolehlivé. V závěru této práce jsou shrnuty některé myšlenky, které by mohly vést k lepším výsledkům.

Pacient	Měření							Průměr (cm^3)	Odchylnka (cm^3)	Odchylnka (%)
	1	2	3	4	5	6	7			
01	149.80	148.76	149.90	150.12	149.05	147.55	149.57	149.25	0.89	0.60
02	176.35	173.54	172.28	175.61	175.74	177.81	174.71	175.15	1.83	1.04
03	193.47	193.35	193.50	193.31	193.73	192.94	193.44	193.39	0.24	0.12
04	135.20	136.57	135.31	136.41	135.71	136.38	136.96	136.08	0.67	0.49
05	121.76	121.68	121.76	122.19	121.82	121.87	121.61	121.81	0.19	0.16
06	191.54	193.53	192.31	191.53	191.77	191.80	191.83	192.04	0.70	0.36
07	97.80	98.13	98.15	97.26	98.19	97.89	98.06	97.90	0.33	0.34
08	245.96	246.39	247.10	245.60	245.90	245.44	247.57	246.28	0.79	0.32
09	155.17	156.16	154.10	155.33	153.67	155.85	156.40	155.24	1.03	0.66
10	152.11	152.72	154.02	152.50	152.19	153.02	153.05	152.80	0.65	0.43
11	212.82	212.49	213.04	212.44	212.92	212.48	213.06	212.75	0.27	0.13
12	154.29	154.76	156.19	154.15	154.04	157.85	155.83	155.30	1.4	0.90

Tabulka 6.7: Invariance měření vůči uživatelskému vstupu v algoritmu postupné segmentace pravé ledviny. Předměřená statistika oblasti se nemění. Měření probíhá pouze na nativní sérii.

Kapitola 7

Závěr

V této práci byly představeny deformovatelné modely jakožto nástroj k segmentaci lékařských dat. Tyto metody dokáží úspěšně modelovat nelineárně deformovatelné objekty. Deformovatelné modely jsou ty obtížnější z technik uvedených v kapitole 3. Množství existujícího výzkumu v této oblasti značí, že skýtají mnoho otevřených problémů a velký potenciál pro praktické využití, obzvláště v lékařských aplikacích. Pro objekty sledované na lékařských snímcích jsou totiž nelineární deformace přirozenou vlastností.

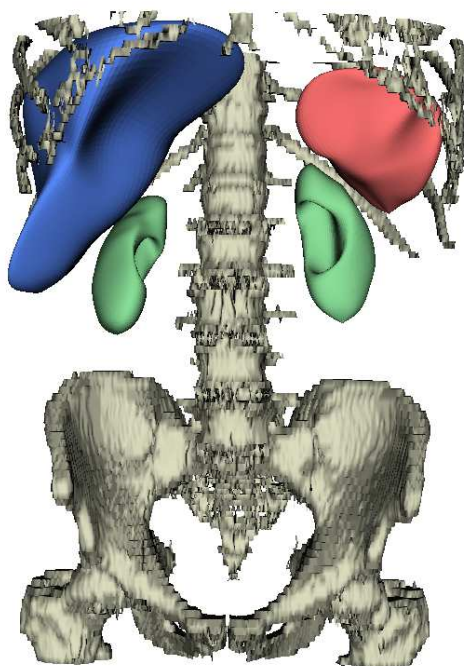
Algoritmy založené na deformovatelných modelech v praxi provádějí *optimalizaci vztahu abstrakce a reálných dat*. V první části práce byly ukázány různé kombinace optimalizace, vztahů a abstrakcí. Zjistili jsme, že B-Spline křivky a plochy jsou zajímavým médiem pro reprezentaci segmentovaných objektů v lidském těle. B-Spline křivky je možné vyjádřit malým počtem parametrů, a proto je jejich optimalizace rychlá. Ukázali jsme, že algoritmy založené na optimalizaci B-Spline ploch a křivek dokáží s jistou přesností segmentovat orgány lidského těla z poměrně hrubých dat počítačové tomografie.

Výsledky měření, která jsme provedli, ukazují, že konkrétní algoritmus se lépe hodí k segmentování orgánu, pro který je navržen. Je to v souladu s tezí v úvodní kapitole. Jediný univerzální algoritmus pro segmentaci zatím neexistuje. Ve své práci jsem se však snažil ukázat, prostřednictvím obsáhlé rešerše, že deformovatelné modely jsou univerzální třídou algoritmů pro segmentaci lékařských dat.

Vybudovaná rekonstrukční schémata na takto založené segmentaci jsou použitelná k měření objemu orgánu v břišní dutině popsaných v kapitole 1 a ukázaných na Obrázku 1.1 z lékařského atlasu. Nabízí se srovnání s výsledkem rekonstrukce na Obrázku 7.1.

Bohužel metody nejsou stoprocentně spolehlivé. Existují případy poškozených obrazových dat, či obrazů pacientů, kteří jsou nemocní, nebo poranění, proto na nich algoritmy většinou selhávají. Takové případy si vyžádají opakovaná měření, větší pozornost uživatele a případnou manuální korekci.

V průběhu zkoumání deformovatelných modelů a při jejich implementaci bylo odkryto mnoho zajímavých problémů, jejichž řešení by se mohlo stát tématem dalšího výzkumu. Především model v algoritmu modelování plochy by se dal



Obrázek 7.1: Rekonstrukce orgánů břišní dutiny. Ledviny (zelené) a slezina (červená) odpovídají tvarům na CT snímcích s vysokou přesností. Játra (modře) mají složitější strukturu a na obrázku chybí část menšího levého laloku.

rozšířit o další funkcionály. Například podobně, jako tomu bylo u B-Spline křivek, by mohla být přidána energie, která přinutí jednotlivé pláty plochy, aby měly stejnou velikost. Analogicky s B-Spline křivkami to může vypadat následovně.

$$E_{int} = \int \left(|\mathbf{N}(u,v)|^2 - \frac{area^2(\mathbf{V})}{(N \times N)^2} \right)^2 dudv$$

Bohužel tomuto rozšíření brání pevná rigidní válcová topologie plochy zdegenerovaná v pólech. Aby bylo možné zachytit složitější topologii orgánů, mohla by se jednoduchá čtvercová mříž řídicích bodů plochy nahradit jiným systémem. Libovolnou topologii bychom mohli reprezentovat například trojúhelníkovou sítí řídicích bodů. Zároveň bychom mohli klást i menší požadavky na počáteční polohu a tvar, pokud bychom do rekonstrukčního schématu zahrnuli adaptivní dělení (subdivision).

Algoritmus postupné segmentace na druhou stranu pracuje poměrně spolehlivě a více lokálně bez potřeby najednou přistupovat k větší oblasti dat. Je tedy možné uvažovat o tom, jestli by se dala implementovat schémata segmentace přizpůsobená pro paralelní počítání na víceprocesorových strojích a moderních strojích s vícejádrovými procesory. Paralelizaci si lze snadno představit v případě algoritmů založených na reprezentaci křivek level-sety z kapitoly 4.5. V takové situaci se více výpočetních jednotek podílí na vyhodnocení jednoho kroku optimalizace. Existují implementace level-setů využívající výpočetní výkon GPU [20].

Dodatek A

Implementace

Naše implementace zahrnuje oba algoritmy popsané v kapitole 5. Program je rozdělen na následující části

1. jádro obsahující klíčové datové struktury a algoritmy (*core*)
2. uživatelské rozhraní (*gui*)
3. rozhraní pro načítání dat (*readers*).
4. rozhraní pro vizualizaci (*vtk_iface*).
5. data v externích souborech obsahující parametry (*config*).

Software byl napsán v jazyce C++ za použití několika knihoven a externích nástrojů.

A.1 Reprezentace dat

Program pracuje s několika základními datovými typy, jako jsou *TArray**, *TVector*, *P*F* pro uložení polí a třísloužkových vektorů. *TMatrix* se používá pro uložení polí a velkých matic pro filtry a soustavy rovnic tak, aby se daly použít v numerických algoritmech z knihy [33]. Pro implementaci transformačních matic slouží *TTransformationMatrix* a *TVector4*.

Obrazová data můžeme reprezentovat mnoha způsoby. Buď jako 2D rastrové mapy v *TRasterData*, nebo jako množiny obrazových elementů reprezentující 2D tvar v *P2IVect*. K reprezentaci série snímků DICOM je určen *TDataSet*. Množinu voxelů můžeme uložit buď po řezech v *TVoxelsSliceMap*, a nebo jako seznam prostorových souřadnic v *TVoxelsVect*.

K uchování složitějších objektů jako jsou křivky a plochy, slouží *TSurface*, *TBSpline* a *TContour*.

A.2 Lékařská data

K ukládání lékařských dat se používalo mnoho proprietárních formátů, většinou vázaných na společnosti vyrábějící diagnostické přístroje. Nakonec zvítězil univerzální formát DICOM (Digital Imaging and Communications in Medicine), který je výsledkem standardizačního procesu severoamerické asociace výrobců elektrických zařízení (NEMA) a radiologické společnosti (ACR). Jedna jeho část popisuje ukládání obrazových dat z CT, MRI, nebo rentgenu. Trojrozměrná data z CT jsou ukládána po axiálních 2D řezech. Každý do samostatného souboru, většinou s koncovkou *.dcm*. Každý soubor kromě obrazových dat obsahuje kompletní paletu metainformací o pacientovi, vyšetření a popis snímku.

Práci s daty tohoto standardu implementuje multiplatformní C++ knihovna DICOM Toolkit [16]. Používáme jí v naší implementaci zabalenou do modulu *dcmReader*. Kromě toho na testovací, umělé obrázky ve formátu TGA používáme *tgaReader*.

A.3 Uživatelské rozhraní

Ačkoliv byla úloha motivována praktickými potřebami doktora Horáka, nebylo cílem této práce implementovat uživatelské rozhraní použitelné v praxi. Přesto však z důvodu snadného měření, prezentace výsledků a testování bylo třeba vytvořit minimální prostředí, které nám všechny tyto operace umožní. Toto experimentální rozhraní je tvořeno množstvím oken s ovládacími prvky, které umožňují nastavovat vstupní parametry a spouštět různé části algoritmů modifikující interní datové struktury. Výsledky se zobrazují ve formě 2D a 3D grafiky a textového výstupu. Nazvali jsme toto rozhraní *segmentation* a je zobrazeno na Obrázku A.1.

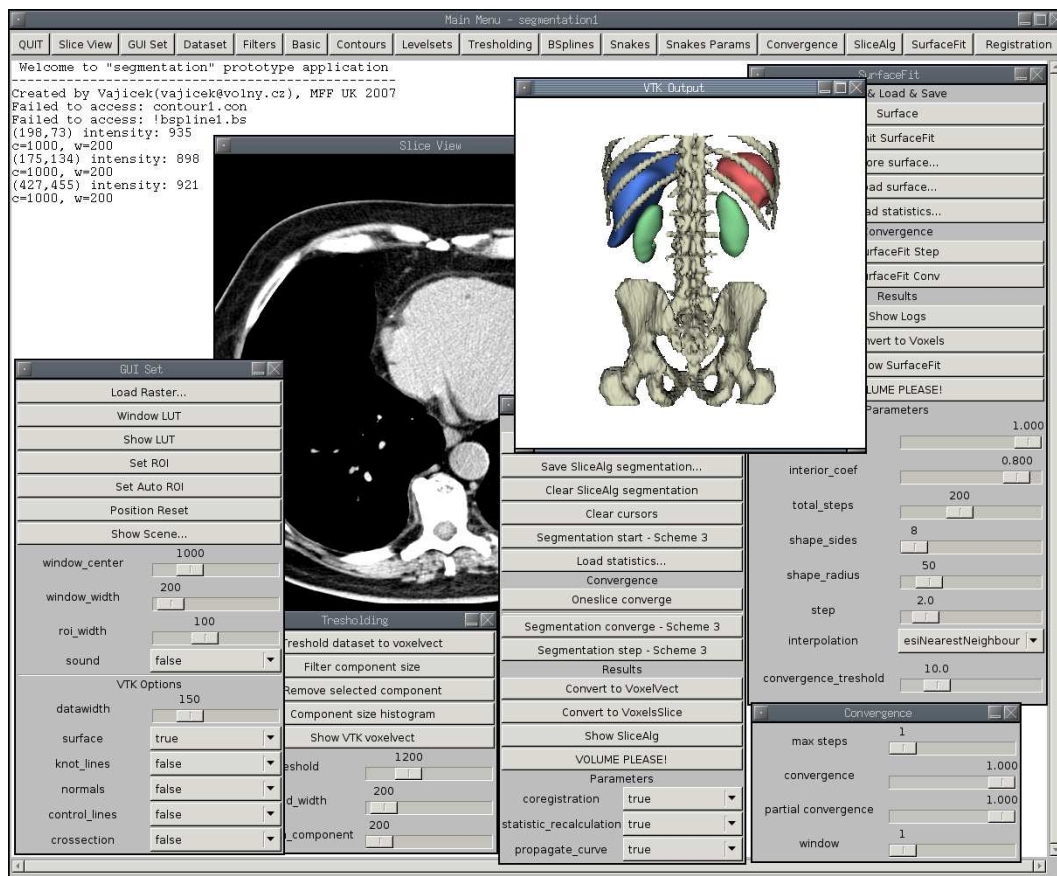
Rozhraní je naprogramované s použitím multiplatformního toolkitu GTK [19], což je knihovna pro vytváření uživatelského rozhraní s okénky a tlačítky. Díky tomu program, který byl vyvíjen na systému Linux, běží i v OS Windows. Na DVD přiloženém k této práci je binární spustitelná verze tohoto rozhraní. Více informací je také možné najít v uživatelské příručce na DVD.

Pro praktické používání by bylo vhodné vytvořit uživatelské rozhraní, nebo rozšíření, plugin, některého existujícího nástroje. Zajímavým vzorem může být program ITK-SNAP [23] určený pro interaktivní segmentaci lékařských dat pomocí deformovatelných modelů.

A.4 Vizualizace

Na výstup jednoduchých 2D obrazů, například jednotlivých řezů CT dat, postačí pixbuffer knihovny GTK, do kterého kreslíme prostřednictvím vlastních kreslicích rutin implementovaných v modulu *RasterOp*.

K zobrazení výsledků rekonstrukce ve 3D používáme knihovnu VTK (Visualization Toolkit) [26], [35]. Tato rozsáhlá knihovna je určena právě pro vi-



Obrázek A.1: Laboratorní rozhraní - segmentation

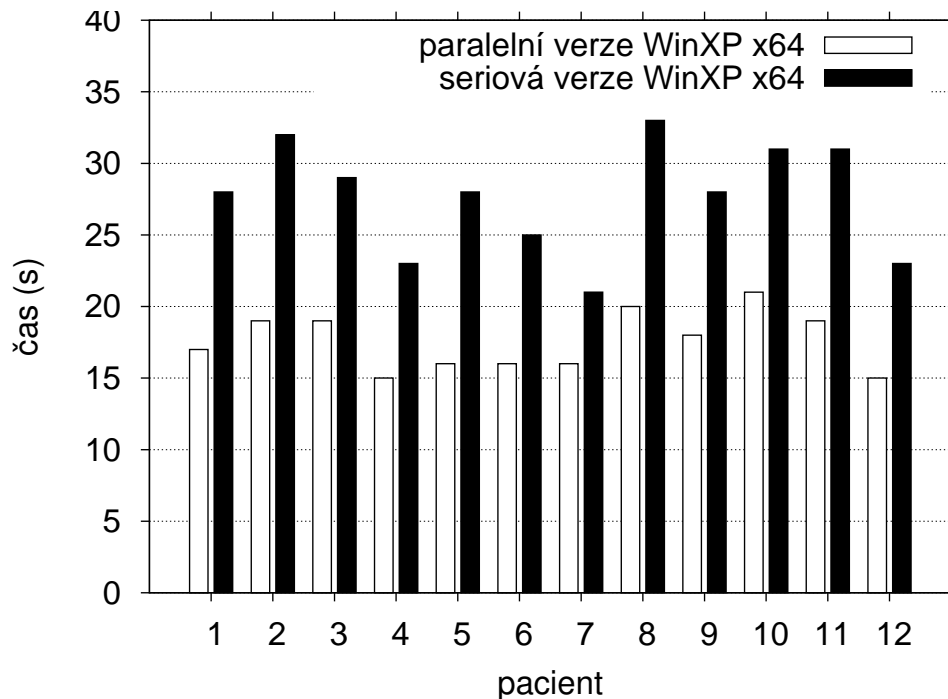
vizualizaci vědeckých dat. Některé obrázky v této práci jsou výstupem VTK. Rozhraní k VTK je implementováno v modulu *vtk_iface*. Z VTK využíváme hlavně *vtkMarchingCubes* k vizualizaci objemových dat a také přímé vykreslování polygonálních dat *vtkPolyData*.

V některých případech k zobrazení jednorozměrných, nebo i dvourozměrných dat bohatě poslouží program *gnuplot* [18]. Používáme ho například ke sledování hodnot konvergenčních kritérií v průběhu algoritmů, nebo k prohlížení histogramů.

A.5 Paralelizace

Dnes běžně dostupné vícejádrové procesory dovolují urychlit výpočet algoritmů, které se povede rozbít do oddělených vláken a minimalizovat režii spojenou s jejich synchronizací a vzájemnou komunikací. O rozložení zátěže mezi jádra se pak starají plánovací algoritmy operačního systému. Důležitým předpokladem je právě možnost nechat algoritmus pracovat na oddělených částech dat bez potřeby si vyměňovat aktuální informace.

Naše schéma postupné segmentace řezů je pro tyto účely téměř ideální. Můžeme algoritmus rozdělit na dva nezávislé běhy, každý postupující z jednoho pólu. Každý



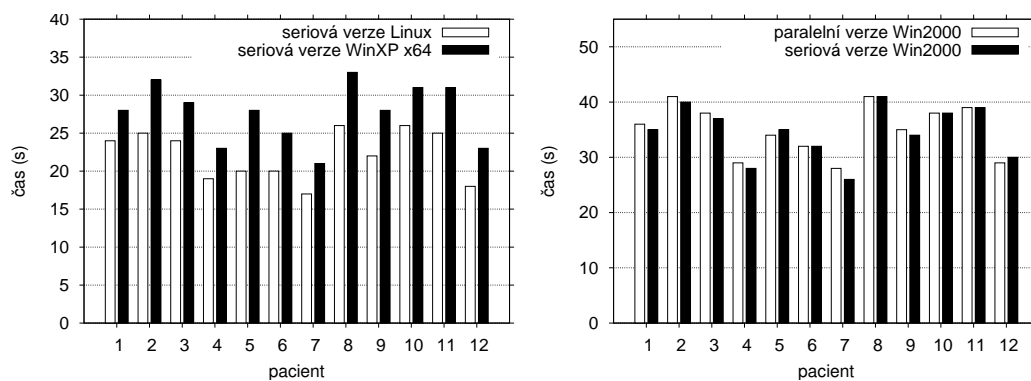
Obrázek A.2: Srovnání rychlostí paralelní a sériové verze algoritmu postupné segmentace řezů na testovací množině pacientů.

z těchto běhů vykonává samostatné, takzvané "pracovní", vlákno. Když se běhy dostanou při propagaci v řezech blízko k sobě, vlákna se ukončí a pokračuje algoritmus podle původního sériového schématu střídáním běhů. Předpokládáme, že je systém k vláknům spravedlivý a přiděluje jim čas rovnoměrně. Pak by měl být výsledek algoritmu téměř totožný jako v sériové verzi, protože oba běhy jdou ke středu stejně rychle jako při sériovém střídání.

Paralelní verzi jsme otestovali v OS Windows XP x64 na počítači s procesorem AMD Athlon 64 X2 Dual Core 3800+ na 2,01GHz a s 4GB RAM. Program byl zkompileován v MS Visual Studiu 8.0 s plnou optimalizací (-Ox). Optimalizace a použití ostrých verzí run-time knihoven má obzvlášť velký význam. Bez nich by byl paralelní algoritmus dokonce dvakrát pomalejší než ekvivalentní sériová verze. Graf na Obrázku A.2 ukazuje rozdíl rychlostí paralelní a sériové verze algoritmu na testovací množině pacientů. Průměrné zrychlení se pohybuje okolo 55,5%.

Testování na OS Windows nám poskytlo příležitost srovnání výkonu algoritmu na různých operačních systémech. Porovnali jsme sériové verze algoritmu z OS Windows a OS Linux na stejném stroji. Graf 1 na Obrázku A.3 ukazuje výrazný rozdíl v každém testovacím případě. Linuxová verze byla přeložena v gcc 4.1 s maximální optimalizací na rychlost (-O3). Aniž bychom chtěli činit nějaké radikální závěry, naše testy ukazují, že algoritmus je pod OS Windows v průměru o 24,9% pomalejší.

Paralelní algoritmus je na vícejádrovém počítači rychlejší. Jak se chová na jednojádrovém stroji AMD Athlon XP 2600+, ukazuje, pro zajímavost, Graf 2 na



Obrázek A.3: Srovnání rychlostí sériové verze algoritmu postupné segmentace podle operačního systému (vlevo). Srovnání rychlostí sériové a paralelní verze algoritmu na jednojádrovém stroji (vpravo).

Obrázku A.3. Rozdíly jsou neznatelné, průměrně v řádu 1% v neprospěch paralelní verze. Z toho se dá vyvodit, že nasazení verze optimalizované pro moderní procesory do výsledného produktu v tomto případě nezpůsobí výrazné zhoršení výkonu na starších počítačích.

Dodatek B

Obsah DVD

<i>src\</i>	Obsahuje zdrojové kódy aplikace
<i>core\</i>	Jádro implementace. Klíčové algoritmy a datové struktury.
<i>gui\</i>	Uživatelské rozhraní.
<i>dcmReader\</i> , <i>tgaReader\</i>	Načítání souborů TGA a DCM.
<i>win\</i>	Projekt pro MSVC 8.0.
<i>vtkgtk\</i>	Rozhraní pro VTK.
<i>makefile</i>	Projekt pro linux a gcc 4.1.
<i>data\</i>	Anonymizovaná testovací data dvanácti pacientů ve formátu DICOM.
<i>patient01\</i> , ..., <i>patient12\</i>	
<i>doc\</i>	Text diplomové práce, programátorská (doxygen) a uživatelská dokumentace.
<i>lib\</i>	Knihovny třetích stran nutné k přeložení a spuštění projektu. Především knihovny GTK, VTK a DCMTK.
<i>bin\</i>	Přeložený projekt a binární verze knihoven.
<i>README.txt</i>	Obsahuje popis obsahu a jiné aktuální informace.

Literatura

- [1] D. Adalsteinsson and J. Sethian. A fast level set method for propagating interfaces, 1995.
- [2] Eric Bardinet, Laurent D. Cohen, and Nicholas Ayache. A parametric deformable model to fit unstructured 3D data. *Computer Vision and Image Understanding: CVIU*, 71(1):39–54, 1998.
- [3] Andrew Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.
- [4] P. Brigger, J. Hoeg, and M. Unser. B-Spline snakes: A flexible tool for parametric contour detection. *IEEE Transactions on Image Processing*, 9(9):1484–1496, September 2000.
- [5] Pavel Campr. Získávání 3D modelu lidských tkání z obrazových dat CT. Master's thesis, ZČU, Plzeň, 2005.
- [6] Vincent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. In *ICCV*, pages 694–699, 1995.
- [7] Václav Krajiček, Josef Pelikán, and Martin Horák. Measuring and segmentation in CT data using deformable models. *WSCG*, 15:149, 2007.
- [8] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277, 2001.
- [9] R. Chung and K-K. Ho. Using 2D active contour models for 3D reconstruction from serial sections. *icpr*, 01:849, 1996.
- [10] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 53(2):211–218, 1991.
- [11] L.D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *PAMI*, 15(11):1131–1147, November 1993.
- [12] T. Cootes and C. Taylor. Active shape models - Smart Snakes, 1992.

- [13] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Lecture Notes in Computer Science*, 1407:484–??, 1998.
- [14] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 266–275, Berlin, 1992. Springer.
- [15] Timothy F. Cootes, A. Hill, Christopher J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. In *IPMI*, pages 33–47, 1993.
- [16] DCMTK. DCMTK - DICOM Toolkit. <http://dicom.offis.de/>, 2007.
- [17] Molly M. Dickens, Shaun S. Gleason, and Hamed Sari-Sarraf. Volumetric segmentation via 3D active shape models. In *SSIAI*, pages 248–, 2002.
- [18] Gnuplot. Gnuplot homepage. <http://www.gnuplot.info/>, May 2006.
- [19] GTK. GTK - Gimp Toolkit. <http://www.gtk.org/>, 2007.
- [20] Markus Hadwiger, Caroline Langer, Henning Scharsach, and Katja Buhler. State of the art report 2004 on GPU-based segmentation, 2004.
- [21] Y. Hicks. Automatic landmarking for building biological shape models, 2002.
- [22] A. Hill, A. Thornham, and C. Taylor. Model-based interpretation of 3D medical images, 1992.
- [23] ITK-SNAP. ITK-SNAP: Software tool for interactive segmentation. <http://www.itksnap.org/>, 2007.
- [24] Mathews Jacob, Thierry Blu, and Michael Unser. Efficient energies and algorithms for parametric snakes. *IEEE Transactions on Image Processing*, 13(9):1231–1244, 2004.
- [25] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [26] Kitware Inc. VTK Home Page. <http://public.kitware.com/VTK/>, 2007.
- [27] Sarang Lakare. 3D Segmentation Techniques for Medical Volumes, 2000.
- [28] Ravi Malladi, James A. Sethian, and Baba C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.
- [29] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *Proc. Fourth International Conf. on Computer Vision (ICCV'93)*, Berlin, Germany, 1993.

- [30] Tim McInerney and Demetri Terzopoulos. Topologically adaptable snakes. In *ICCV*, pages 840–845, 1995.
- [31] D.L. Pham, C. Xu, and J.L. Prince. A survey of current methods in medical image segmentation, 1998.
- [32] R. Pohle and K. Toennies. Segmentation of medical images using adaptive region growing, 2001.
- [33] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.
- [34] Jana Prochazkova. Derivative of B-Spline function. In *25th Conference on Geometry and Graphics, Czech*, 2005.
- [35] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit, 4th Edition*. Kitware Inc., 2006.
- [36] J. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci.*, volume 93, pages 1591–1595, 1996.
- [37] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation, 2004.
- [38] Milan Šonka, Václav Hlaváč, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. International Thomson Publishing, 2002.
- [39] Andy Tsai, Anthony Yezzi, William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, Eric Grimson, and Alan Willsky. A shape-based approach to the segmentation of medical imagery using level sets, 2003.
- [40] Chenyang Xu and Jerry L. Prince. "Snakes - Active Contour Models" Citations. http://iacl.ece.jhu.edu/projects/gvf/gvf_cite/snake_cite_citation.html, 2005.