

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

INTERAKTIVNÍ 3D MODELOVÁNÍ LIDSKÝCH TKÁNÍ NA ZÁKLADĚ CT/MR DAT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL BARTOŠ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

INTERAKTIVNÍ 3D MODELOVÁNÍ LIDSKÝCH TKÁNÍ NA ZÁKLADĚ CT/MR DAT

INTERACTIVE 3D MODELING OF HUMAN TISSUES BY CT/MRI DATA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL BARTOŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. PŘEMYSL KRŠEK, Ph.D.

BRNO 2011

Abstrakt

Tato bakalářská práce se zabývá problematikou separace artefaktů v medicínských obrazových dat ze zdrojů magnetické rezonance a počítačové tomografie. Dále popisuje návrh a implementaci aplikace, demonstrující zvolené řešení, využívající vektorovou Subdivision plochu, pro potřeby ruční segmentace.

Abstract

This thesis deals with the separation of artifacts in medical image data from sources of magnetic resonance imaging and computerized tomography. It also describes the design and implementation of applications demonstrating the use of the chosen solution, using a vector Subdivision surface, the need for manual segmentation.

Klíčová slova

modelování tkání, segmentace medicínských dat, hodnotové artefakty, subdivision plocha, magnetická rezonance, počítačová tomografie

Keywords

tissue modeling, segmentation of medical data, value artifacts, subdivision surface, magnetic resonance imaging, computed tomography

Citace

Pavel Bartoš: Interaktivní 3D modelování lidských tkání na základě CT/MR dat, bakalářská práce, Brno, FIT VUT v Brně, 2011

Interaktivní 3D modelování lidských tkání na základě CT/MR dat

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Přemysla Krška, Ph.D.

.....
Pavel Bartoš
17. května 2011

Poděkování

Tímto bych velmi rád poděkoval vedoucímu mé bakalářské práce doc. Ing. Přemyslu Krškovi, Ph.D. za poskytnuté odborné vedení a rady během tvorby této práce.

© Pavel Bartoš, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
1.1	Cíl práce	2
1.2	Struktura dokumentu	2
2	Rozbor problematiky	4
2.1	Tvorba 3D modelů tkání	4
2.2	Degradace CT/MR dat	8
2.3	Současný stav řešení	10
3	Návrh řešení	13
3.1	Specifikace požadavků	13
3.2	Návrh systému	14
3.3	Použité technologie	16
4	Implementace	18
4.1	Jádro aplikace	18
4.2	Zobrazení	19
4.3	Aplikační pozadí	21
5	Výsledky	25
5.1	Přehled řešení	25
5.2	Uživatelská podpora	25
5.3	Výkonnostní testy	26
6	Závěr	28
A	Obsah CD	31

Kapitola 1

Úvod

V 80. letech 20. století dospěla medicína do bodu, odkdy lékaři mohou detailně zkoumat vnitřní orgány pacienta bez potřeby invazivního zákroku. Za tento pokrok vděčíme především metodám snímání pomocí počítačové tomografie (CT) a magnetické rezonance (MR). Plný potenciál těchto technologií se však projevil až ve spojení s výpočetní technikou. Ta v současnosti umožňuje získané snímky převést do počítačového modelu a pohodlně je tak zobrazovat ve virtuální prostředí. Na jejich základě je pak možné vytvářet reálné modely tkání, které nacházejí uplatnění v podobě přesných implantátů, nebo jako užitečné pomůcky při přípravě lékařů na operaci.

Medicínská obrazová CT/MR data však nejsou dokonalá a často nesou známky deformace, vycházející z použité technologie. Příkladem mohou být různé typy hodnotových artefaktů, způsobených vlivem anorganických látek na proces snímání. Veškeré tyto anomálie mají negativní vliv na další zpracování a využití dat. Jejich odstranění je však v současné době velmi technologicky a časově náročné.

1.1 Cíl práce

Cílem této práce je v první řadě prostudovat proces tvorby modelů lidských tkání, se zaměřením na oblast problematiky segmentace dat. Na základě získaných poznatků pak dále navrhnout a realizovat účinné řešení separace hodnotových artefaktů v CT/MR datech, s využitím vektorové Subdivision plochy, proložené 3D volume daty¹, pro potřeby editace segmentace.

Výstupem této práce by měla být samostatná aplikace demonstrující použití navrženého řešení. K tomu bude zapotřebí implementovat i podpůrné funkce, jako například načítání a zobrazování obrazových dat.

1.2 Struktura dokumentu

V kapitole č. 2 uvádím čtenáře do základů problematiky 3D modelování lidských tkání na základě medicínských obrazových dat. Dále je zde také popsáno, jakými problémy jsou tyto postupy zatíženy, a to především na úrovni segmentace. Na konci kapitoly uvádím metody, jakými je tato problematika řešena v praxi a proč je současné řešení nevyhovující.

Kapitola č. 3 se zabývá návrhem řešení, vycházejícím z poznatků z druhé kapitoly. Na začátku se věnuji analýze a specifikaci požadavků na systém. Druhá část kapitoly popisuje

¹Volume data - značení objemových diskrétních dat, popisující strukturu tkáně.

funkcionální návrh systému. V poslední části uvádím použité technologie a postupy, mezi které mimo jiné patří i zvolená Subdivision metoda, určená pro vyhlazování plochy.

V následující kapitole č. 4 popisuji implementaci aplikace s členěním do jednotlivých částí, odpovídajících logickému rozdělení výsledného systému. Cílem kapitoly je poskytnutí informací použitelných jako inspirace pro další vývoj aplikací v této oblasti.

Kapitola č. 5 se zabývá prezentací výsledného řešení a ověřením jeho použitelnosti.

Závěrečná kapitola č. 6 obsahuje shrnutí této práce.

Kapitola 2

Rozbor problematiky

Základním předpokladem pro úspěšný návrh řešení problematiky separace hodnotových artefaktů v medicínských datech je pochopení základního principu práce a vizualizace těchto dat. V první části této kapitoly se proto zabývám kompletním procesem vytváření 3D geometrických modelů lidských tkání. Díky získaným znalostem bude následně možné identifikovat a popsat hlavní problémy, se kterými se tento postup potýká, čemuž je věnována druhá část kapitoly. V třetí a poslední části je diskutováno současné řešení dané problematiky v praxi, za účelem identifikace nedostatků, které budou sloužit jako podklad pro návrh vlastního řešení.

2.1 Tvorba 3D modelů tkání

Tvorba virtuálních modelů lidských tkání je proces, kdy z diskretních medicínských obrazových dat, získaných snímáním těla pacienta, vytváříme 3D geometrický model cílové tkáně. Tento proces lze rozdělit do několika hlavních částí:

- Předzpracování CT/MR zdrojových dat
- Segmentace tkání
- Vektorizace diskretních dat
- Dodatečná optimalizace modelu

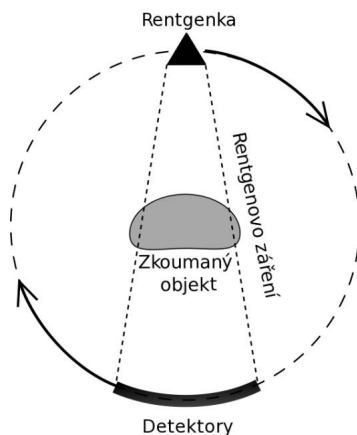
Veškeré tyto části, i se začleněním oblasti zdrojů dat, jsou popsány dále.

2.1.1 Získávání obrazových dat

V současné době existuje několik metod pro získání diskretních 3D volume dat potřebných k vizualizaci cílového objektu. Mezi ně patří například magnetická rezonance (MR), počítačová tomografie (CT), rentgen či ultrasonografie¹. V této práci se však budu zabývat pouze daty, získanými ze zdrojů CT/MR, které jsou v dané oblasti nejpoužívanější.

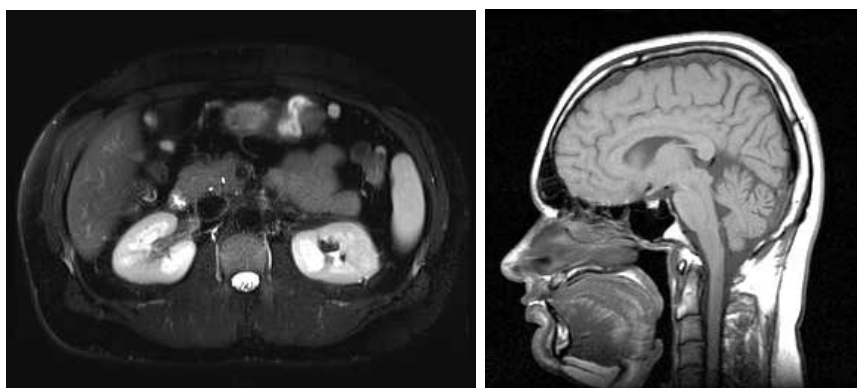
¹Ultrasonografie - metoda zobrazování tkáně s využitím ultrazvuku.

Počítačová tomografie (Computed Tomography) - princip této metody spočívá v ozařování částí pacientova těla z různých úhlů (obr. 2.1), pomocí úzkého paprsku rentgenového záření. Podle poměru vstupní a výstupní intenzity záření se určuje přibližná hustota tkáně a následně i jas odpovídajícího pixelu ve výsledném zobrazení. Tento proces probíhá v jednotlivých řezech, proto každý snímek je průřezem aktuální části pacientova těla. Rentgenové záření má však nižší rozlišovací schopnost a proto je vhodné především pro snímání kostí.



Obrázek 2.1: Princip snímání pomocí počítačové tomografie [13].

Magnetická rezonance (Magnetic resonance imaging) - stejně jako počítačová tomografie využívá principu rotačního snímání po jednotlivých řezech. Rozdíl spočívá především v použití silného homogenního magnetického pole místo rentgenového záření, díky němuž je dosaženo vyššího rozlišení hustoty tkání. Tyto vlastnosti se uplatňují při zobrazování měkkých tkání, jako například mozku či jiných vnitřních orgánů, kde jednotlivé tkáně jsou z pohledu hustoty velmi podobné. Nevýhodou je častější výskyt artefaktů a delší doba potřebná pro provedení vyšetření oproti počítačové tomografii.



Obrázek 2.2: Snímky pořízené pomocí magnetické rezonance[12].

Formát dat

Získaná CT/MR data představují sled 2D snímků jednotlivých řezů v různých stupních šedi, kde úroveň jasu každého pixelu udává hustotu snímané tkáně. Pevné tkáně mají zpravidla oproti měkkým tkáním vyšší jas. Strukturu uložených objemových dat lze popsat jako 3D matici, skládající se z jednotlivých voxelů².

Rozlišení snímků a přesnost snímání jsou hlavním faktorem určujícím kvalitu výsledného modelu. Běžně se používá rozlišení 512×512 pixelů s přesností 0,1 mm až 1 mm na pixel. Počet snímků se pak pohybuje okolo 100 - 200 řezů.

V současné době je standardem pro uložení 3D rastrových dat formát DICOM[2] (Digital Imaging and Communications in Medicine), který uchovává zejména informace o formátu obrazových dat a souborech, použitých protokolech či metadatech s údaji pacienta.

2.1.2 Předzpracování CT/MR zdrojových dat

Vstupní data ze zdrojů CT/MR zpravidla bývají v surovém formátu, tzn. nebyla nijak optimalizována pro následné zpracování. Předzpracováním dat dosáhneme zvýšení jejich kvality a následně i lepších výsledků při dalším postupu. Mezi nejpoužívanější metody patří:

- úprava kontrastu tkání za účelem získání jejich vyššího detailu;
- filtrace obrazu pro potlačení šumu.

2.1.3 Segmentace tkání

Pro vytvoření modelu specifické tkáně je potřeba určit její prostorové informace, které nejsou ve vstupních diskretních obrazových datech přímo obsaženy. Touto problematikou se zabývá segmentace dat. Princip spočívá v kategorizaci jednotlivých pixelů k příslušným typům tkáně. Na základě úrovně jasu pixelů z jednotlivých snímků určujeme jejich příslušnost k odpovídající tkáni. Tento proces lze provádět buď přímo na aktuálním řezu (2D segmentace) nebo na prostorové matici pixelů (3D segmentace). Rozlišujeme několik základních metod segmentace:

- **Manuální segmentace** - spočívá v ruční selekci požadované oblasti, často za pomoci podpůrných algoritmů. Tato metoda je však při vyšším počtu snímků velice časově náročná, jelikož se zpravidla jedná o 2D segmentaci, a proto každý snímek, do kterého požadovaná oblast zasahuje, musí být zpracovávána samostatně.

Naopak výhodou je kvalitnější výsledek, což je dáno interakcí uživatele, který při rozhodování využívá více vstupních informací a především svých zkušeností, oproti algoritmům automatické segmentace. Z tohoto důvodu se manuální segmentace většinou používá pro korekci výstupů automatické segmentace.

- **Automatická segmentace** - proces kategorizace tkání se provádí automaticky na základě specifických algoritmů uvedených dále. Pro zvýšení přesnosti se tyto algoritmy často kombinují. Výhodou je vyšší rychlost zpracování. Vhodné jsou především pro segmentaci dat s vysokým kontrastem hustoty tkání. Nevýhodou je pak absence zkušeností uživatele, což má často za následek například zahrnutí cizích objektů či artefaktů podobné hustoty, jako je aktuálně kategorizovaná tkáň.

²Voxel - element prostorových diskretních dat, podobně jako pixel v 2D grafice.

- **Poloautomatická segmentace** - kombinace dvou výše uvedených metod. Automaticky prováděný proces ovlivněný parametry zadanými uživatelem.

Dále jsou uvedeny jednotlivé algoritmy, využívané během segmentace, s rozdělením dle jejich přístupu:

Strukturální metody, jak jejich název napovídá, využívají při rozhodování informace o struktuře jejich okolí. Patří mezi ně:

- Detekce hran
- Morfologické metody
- Prohledávání grafu
- Deformovatelné modely
- Izoplochy a křivky úrovní
- Level-sets
- Aktivní kontury

Stochastické metody ignorují strukturální informace. Jejich princip spočívá v rozhodování na základě statické analýzy:

- Prahování
- Klasifikační metody
- Shlukování
- Adaptivní prahování
- Amplitudová projekce
- Kohonenovy mapy
- Markovské pole

Hybridní metody využívají částečně přístupy obou výše uvedených metod a proto není možné zařadit je do jedné z nich:

- Růst oblastí
- Rozdělení a sloučení
- Atlas-Guided
- Umělé neuronové sítě
- LEGION segmentace

2.1.4 Vektorizace

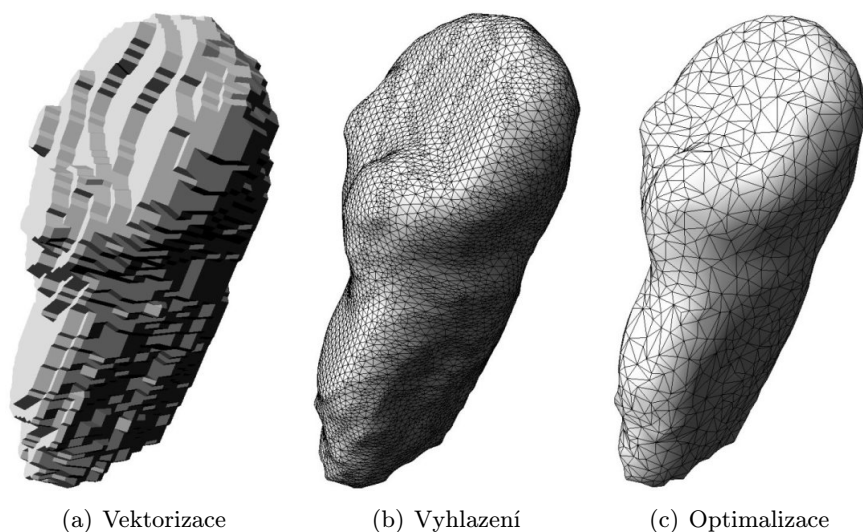
V současnosti nejčastější reprezentací počítačových 3D modelů je trojúhelníkový polygonální model. Jedná se vektorovou virtuální síť skládající se výhradně z trojúhelníků. Proces převodu diskretních dat získaných segmentací na požadovaný polygonální model označujeme jako vektorizace.

Marching cubes - jedná se o nejvyžívanější algoritmus při procesu vektorizace. Princip spočívá v procházení jednotlivých řezů segmentovaných dat a výběrem krychlí s vrcholy, odpovídajícími bodům vstupní mřížky. Krychle s minimálně jednou rozdílnou hodnotou svého vrcholu indikuje hranici modelovaného objektu. Těmi jsou následně prokládány trojúhelníky, které ve výsledku tvoří cílový polygonální model.

2.1.5 Optimalizace modelu

Po procesu vektorizace máme k dispozici použitelný geometrický model, který se však zpravidla následně dále upravuje. Prvním důvodem pro úpravu modelu je neplynulá povrchová struktura, způsobená vektorizací diskretních dat. K řešení tohoto defektu se v současné době obvykle používá vyhlazování s využitím Laplaceovy metody.

Druhý důvod pro úpravu modelu je čistě výkonostního charakteru. Výsledný polygon je často definován zbytečně velkým počtem trojúhelníků, což má za následek zpomalení následných výpočetních operací na modelu. Pro redukci počtu trojúhelníků bez ztráty detailů objektu využíváme triangulaci rovinných ploch. Dalším způsobem je ignorování nepodstatných detailů, což ve většině případů nemá zásadní vliv na výslednou geometrii modelu.



Obrázek 2.3: Vliv jednotlivých procesů na výsledný model[7].

2.2 Degradace CT/MR dat

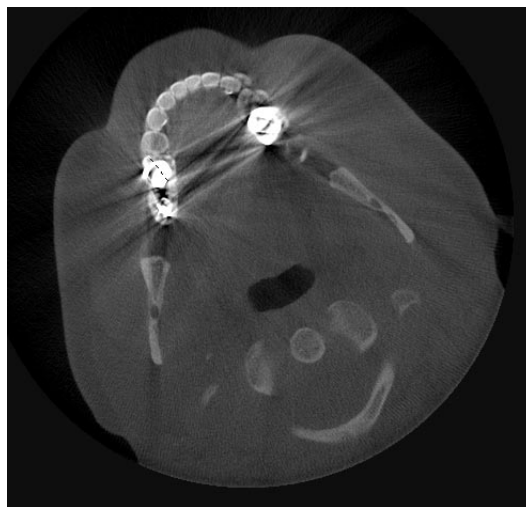
Pořízená obrazová data ze zdrojů CT/MR často bývají zatížena chybami vzniklými při snímání pacienta. Mezi nejčastější zdroje způsobující snížení kvality vstupních dat, v závislosti na použité technologii[7], patří:

- Rozlišení objemových dat
- Prostorová deformace
- Šum v obraze
- Pohybové artefakty
- Hodnotové artefakty

Veškeré výše uvedené degradace medicínských dat mají značný vliv na výslednou kvalitu počítačových modelů tkání. Tato práce se však zabývá především eliminací deformací modelů způsobených hodnotovými artefakty a cizími tělesy, které se nejčastěji odstraňují pomocí ruční segmentace. Následující bližší popis pomůže lépe navrhnout cílové řešení uvedené problematiky.

2.2.1 Hodnotové artefakty

Tento typ deformace vstupních dat je zpravidla způsoben přítomností metalických prvků ve snímaném objektu při využití metody počítačové tomografie. Kovové části, jako například zubní výplně či výztuže kostí, rapidně snižují intenzitu rentgenových paprsků, což způsobí nepřiměřené zvýšení jasu v postižené oblasti snímku. Takto deformované snímky se projevují ztrátou detailu tkání v přilehlém okolí. Následně, při použití automatické segmentace, jsou tyto artefakty chybně identifikovány jako lidská tkáň. Výsledkem je pak nereálný model zasažené oblasti.



Obrázek 2.4: Vliv metalické plomby na snímek pořízený počítačovou tomografií[1].

2.2.2 Cizí tělesa

Lékařské vybavení je často nedílnou součástí snímaného subjektu. Pomineme-li metalické prvky, kterým je věnována předešlá sekce, dalším zdrojem deformace vstupních obrazových dat jsou i nekovové objekty. Drenáž s podobnou hustotou jako má lidská lebka, může při dostatečné blízkosti deformovat vykreslený model lebeční kosti, na který byla použita automatická segmentace. Výsledkem je pak společná kategorizace obou objektů.

Nejúčinnější metodou pro odstranění tohoto defektu je, stejně jako v případě hodnotových artefaktů, ruční segmentace s využitím znalostí anatomie lidského těla a zkušeností uživatele, který je schopen odlišit anorganickou složku.

2.3 Současný stav řešení

V tuto chvíli by měl mít čtenář dostatek základních informací o problematice modelování lidských tkání na základě medicínských dat a také o teoretických možnostech odstraňování nežádoucích artefaktů během segmentace. Proto nyní mohu uvést způsoby, jakými je tato problematika řešena v praxi, a to konkrétně pomocí aplikace *3DimViewer*, u které se předpokládá využití poznatků této práce při dalším vývoji.

2.3.1 3DimViewer

3DimViewer [8] je freeware program od společnosti *3Dim Laboratory* pro tvorbu virtuálních modelů lidských tkání na základě CT/MR obrazových dat. Mezi jeho základní funkcionalitu mimo jiné patří:

- 3D zobrazování objemových dat
- Volume rendering za pomoci grafické akcelerace
- Segmentace tkání na základě prahování
- Zobrazování a práce ve více řezech najednou
- Import dat ve formátu DICOM
- 3D vizualizace povrchu modelu

Pomocí zásuvných modulů je možné jeho funkcionalitu dále rozšířit. *Segmentation Plugin* zahrnuje podporu manuální a poloautomatické segmentace, která umožňuje odstraňování nežádoucích artefaktů. Jeho bližší použití je popsáno v následující části.

2.3.2 Segmentace v praxi

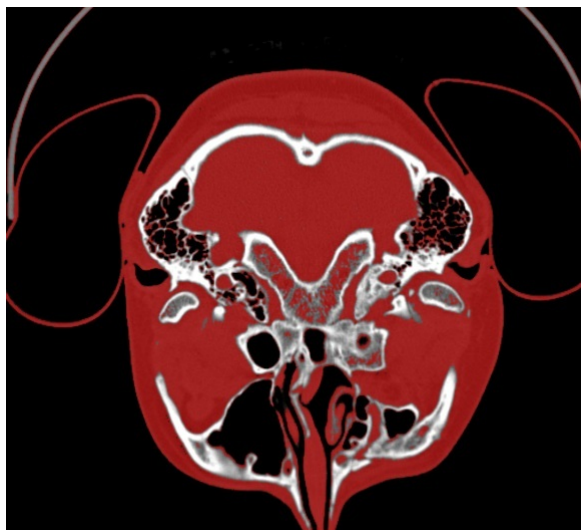
Program *3DimViewer* poskytuje spolu s doinstalovaným příslušným zásuvným modulem několik metod ruční segmentace. Zde uvedu příklady jejich použití na reálných datech, spolu s výhodami a nevýhodami daného postupu. Uvedené názvy nejsou oficiální a slouží pouze pro lepší orientaci.

Jako demonstrační vstupní data budou použita CT snímky části hlavy s rozlišením $512 \times 512 \times 11$ voxelů. Cílem je segmentace měkkých tkání. Komplikaci zde představuje přítomnost hlavové opěrky s podobnou hustotou.

Prahování

Jedná se o poloautomatickou segmentaci, kdy je výběr požadované oblasti prováděn na základě úrovně jasu pixelů, která je porovnávána s uživatelsky nastavenou hodnotou, tzv. prahem.

Výhody: Vysoká rychlost segmentace.

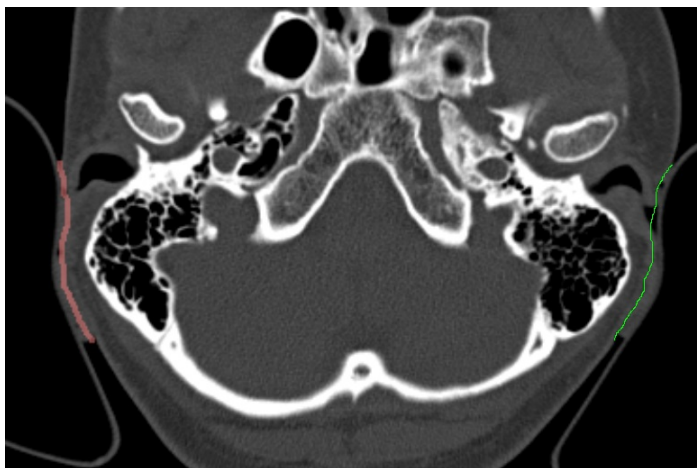


Obrázek 2.5: Výsledek segmentace pomocí metody Prahování.

Nevýhody: V případě dotyku a prakticky stejné hustoty obou objektů, ani uživatelské nastavení hodnoty prahu nedokáže provést segmentaci účinně, jak je patrné na obrázku 2.5.

Tvorba hranic

Princip metody spočívá v manuálním vytvoření hranice mezi cizím objektem a požadovanou tkání v každém řezu. Tato hranice je definována uživatelsky nakreslenou křivkou. Následně lze provést automatické 2D nebo 3D vyplnění cílové oblasti s nastavením příslušné hustoty tkáně.



Obrázek 2.6: Segmentace pomocí metody Tvorby hranic

Výhody: Kvalitní výstupní segmentace.

Nevýhody: Hranice je potřeba přesně definovat v každém řezu, což může být při vysokém počtu řezů velmi pracné.

Selekce oblastí

Poslední zvolenou možností je selekce požadovaných oblastí pomocí polygonu. Uživatel takzvaným „lasem“ ohraničí oblast vlivu fragmentu, nejlépe s podporou automatického výběru dle hustoty tkání. Takto označený objekt spadá do samostatné kategorie a v následujících fázích modelace není zahrnut.



Obrázek 2.7: Segmentace pomocí metody Selekce oblastí

Výhody: Opět kvalitní výstupní segmentace.

Nevýhody: I zde se jedná o manuální 2D segmentaci a proto je často značně pracná.

2.3.3 Motivace k řešení

Z výše uvedených příkladů vyplývá, že v současné době není v programu *3DimViewer* implementována vhodná efektivní metoda pro odstranění artefaktů v obrazových datech. První uvedený způsob - *Prahování*, lze použít pouze v situacích, kdy se cizí těleso nedotýká cílové tkáně. Naproti tomu, *Selekce oblastí* a *Tvorba hranic* mohou být teoreticky použity v jakékoliv situaci, pokud opomeneme jejich pracnost. V praxi je však čas uživatelů s dostatečnými znalostmi a zkušenostmi velmi drahý, a proto jsou tyto postupy často nepoužitelné. Využití naleznou pouze při segmentaci menšího počtu snímků nebo korekci automatické segmentace.

Možnost ruční segmentace více řezů najednou by přinesla nejen značné usnadnění a zrychlení opravy vizuálních dat, ale také by v některých případech zvýšila kvalitu výsledných modelů, která je v současnosti obětována za účelem snížení času a nákladů při zpracování medicínských dat.

Kapitola 3

Návrh řešení

Druhá kapitola se zabývá specifikací a návrhem řešení výše uvedené problematiky. V první části této kapitoly pro přehlednost shrnu hlavní cíle projektu, které byly naznačeny výše. Dále pak vydefinuji požadavky, které musí systém splňovat, a na jejichž základě bude možné provést návrh řešení. Tomu je věnována následující druhá část, kde popisuji především funkcionální návrh systému. Do poslední části jsem zařadil přehled použité technologie, kam patří i zvolená metoda *Subdivision*, pro vyhlazování separační plochy.

3.1 Specifikace požadavků

3.1.1 Cíle práce

Hlavním cílem této práce je navrhnout a reálně prezentovat účinné řešení podpory manuální segmentace obrazových dat, za účelem optimalizace selekce nežádoucích artefaktů, a to formou demonstračního programu. K tomu, dle zadání, bude využita vektorová separační plocha, proložená obrazovými daty. Takovéto řešení by nám mělo poskytnout možnost ruční 3D segmentace a editovat tak hranici selekce ve více vrstvách najednou. Použití *Subdivision* vyhlazování separační plochy nám navíc usnadní modelování hranic tkání se spojitým zakřivením.

3.1.2 Funkcionální požadavky

Nyní, na základě dosažených znalostí můžeme určit, jakou funkcionalitou musí výsledná aplikace disponovat, aby splňovala hlavní cíl práce.

Zobrazení medicínských dat

Jak již bylo uvedeno výše, výsledné řešení bude samostatnou aplikací. Proto je potřeba implementovat načítání a zobrazování medicínských dat formátu DICOM.

Návrh zobrazení by měl být inspirován současnou implementací programu *3DimViewer*, kde se jednotlivé řezy zobrazují ve třech různých osách. Jedinou výjimkou bude absence multiplanárního zobrazení, pomocí kterého v programu *3DimViewer* nelze manuální segmentaci provádět, a proto není pro toto řešení potřebné.

Editace a zobrazení separační plochy

Pro plnohodnotnou realizaci požadovaných hranic, použitých při segmentaci, musí mít uživatel možnost separační plochu plně editovat ve všech vrstvách, kterými prochází. Řešením je vytvoření pravidelné čtyřúhelníkové řídicí sítě, na jejímž základě bude výsledná separační plocha vyhlazována. Pomocí zobrazení křivky v aktuální vrstvě, reprezentující průsečík daného řezu a řídicí sítě, bude uživatel schopen definovat požadovaný tvar v dané oblasti. Deformací této křivky bude odcházet k odpovídajícímu zakřivení separační plochy.

Dalším požadavkem na separační polygon je, že musí přesně procházet řídicími body, zadanými uživatelem. Z tohoto důvodu bude výsledný polygon vyhlazován jednou z interpolačních metod *Subdivision*.

3.2 Návrh systému

Zde se věnuji bližšímu popisu plánovaného chování systému, rozděleného do příslušných částí, na jejichž základě budou následně vyvinuty jednotlivé komponenty.

3.2.1 Obecný popis řešení

Program bude zobrazovat zadaná medicínská data pomocí řezu v jedné ze tří na sobě kolmých osách, s možností přepínání mezi nimi a posunu po snímcích. V těchto řezech se bude také provádět editace řídicí sítě pomocí jejího průřezu v dané ose. Na základě této virtuální sítě bude vykreslována vyhlazená plocha procházející jejími řídicími body. To uživateli umožní snadné a rychlé definování hranic, představujících požadovanou oblast segmentace. Plánované chování systému je přehledně znázorněno na obrázku 3.1.

3.2.2 Načítání a zobrazování medicínských dat

Kompletní realizace načítání a formátování obrazových dat není cílem této práce a jeho implementace by odvedla pozornost od hlavního řešení. Z toho důvodu bude využit modul `mdsSliceView` z knihovny `MDSTk`, který dokáže převést data typu `Volume`¹ do dvojrozměrné textury odpovídajícího řezu, sloužící pro zobrazení aktuálního snímku v aplikaci. Modul `mdsSliceView` načítá data buď pomocí `roury` v příkazové řádce nebo pomocí souboru. Takto bude realizován i vstup programu.

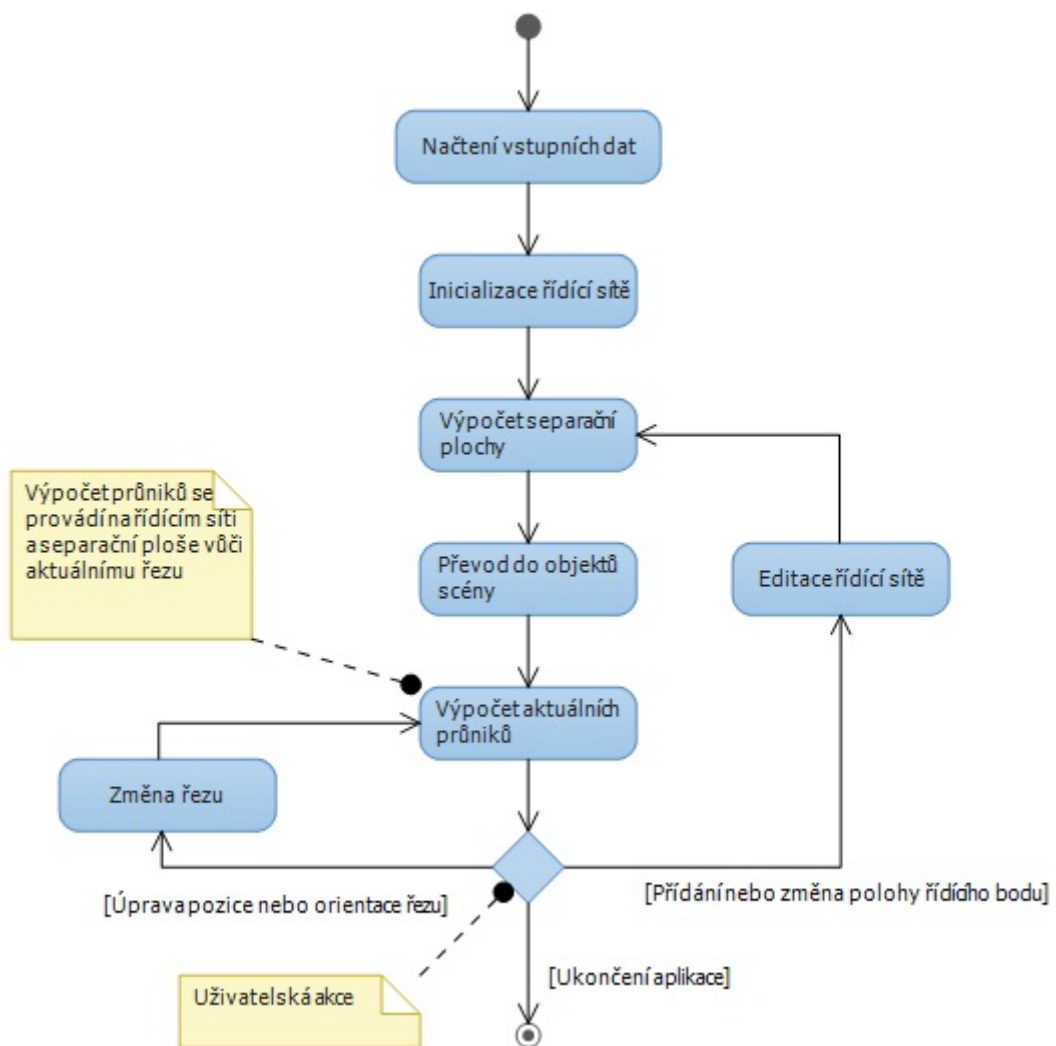
Načtená medicínská data se uživateli prezentují pomocí řezu v jedné ze tří na sobě kolmých osách, s možností cyklického přepínání mezi nimi. Posun v osách bude realizován diskrétně po jednotlivých snímcích a to v reakci na stisk klávesy.

3.2.3 Editace řídicí sítě

Pohyb s řídicími body sítě bude probíhat ve dvou osách vůči rovině aktuálního řezu. Obdobně bude řešeno i jejich vkládání.

Předchozí specifikace je definována pro řez obsahující řídicí body, neboli prochází-li přímo jednou ze stran řídicí sítě. Častěji se však řez bude nacházet mezi jednotlivými řádky či sloupci polygonu. Stejně jako v předchozím případě bude zobrazena křivka průniku řezu a editační sítě, avšak bez řídicích bodů. Ty musí být, na požadavek uživatele iniciovaný začátkem editace křivky, automaticky vloženy do aktuálního řezu za účelem korekce sítě v daném místě.

¹`Volume` - interní typ knihovny `MDSTk` pro uložení zpracovaných dat formátu DICOM.



Obrázek 3.1: Diagram aktivit popisující chování systému.

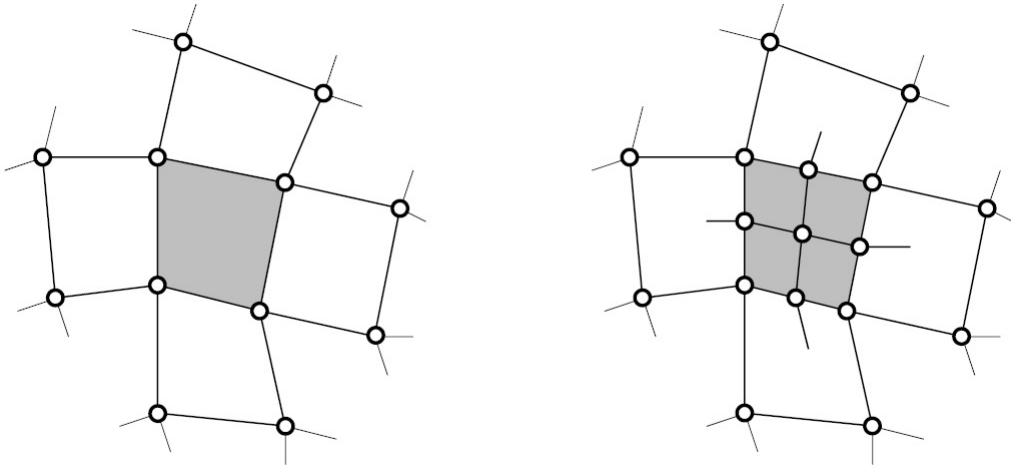
3.2.4 Vyhlazování

Stejně jako v případě řídicí sítě, i vyhlazená plocha bude zobrazována v řezu pomocí křivky. Proces vyhlazování separačního polygonu bude probíhat automaticky po každé změně řídicí sítě. Uživatel tak bude moci v reálném čase kontrolovat výsledek své práce a provádět efektivněji požadované úpravy. V případě vysokého rozlišení řídicího polygonu však může docházet k nežádoucímu zpoždění při vyhlazování a následnému „zamrznutí“ aplikace. Řešením je implementace uživatelské volby kvality vyhlazení plochy, jejímž snížením se výrazně zrychlí operace vyhlazování.

3.3 Použité technologie

3.3.1 Kobbelt schéma

Pro vyhlazení separační plochy musí být použita jedna z interpolačních metod *Subdivision* s možností aplikace na pravidelnou čtyřúhelníkovou síť. Takto zadaným požadavkům odpovídá metoda představená Leifem Kobbeltem[11]. Princip metody spočívá ve výpočtu hranových bodů, které jsou následně analogicky použity na získání cílového středového bodu.



Obrázek 3.2: Proces určení pozice středového bodu pomocí metody Kobbelt[11].

Výpočet nových bodů je realizován pomocí vztahu 3.1, kde koeficient ω udává konvergenci křivky.

$$\mathbf{p}'_{2i+1} := \frac{8 + \omega}{16}(\mathbf{p}_i + \mathbf{p}_{i+1}) - \frac{\omega}{16}(\mathbf{p}_{i-1} + \mathbf{p}_{i+2}) \quad (3.1)$$

Kobbelt při návrhu stejnojmenné metody vychází z použití čtyřbodové interpolační metody aplikované na pravidelnou čtyřúhelníkovou síť. Metoda lze po úpravě použít i na nepravidelnou čtyřúhelníkovou síť, což však v našem případě nebude potřeba.

3.3.2 MDSTk

Medical Data Segmentation Toolkit je sada nástrojů, napsaná v jazyce C++ s využitím *OpenGL*, pro podporu práce s vizuálními medicínskými daty. Umožňuje mimo jiné provádět

3D filtraci, detekci hran, segmentaci či multiplanární zobrazování. Současné řešení využívá jeden z modulů knihovny, konkrétně `mdsSliceView`, pro načítání a převod DICOM dat.

3.3.3 OpenSceneGraph

Jedná se o open source 3D grafický toolkit, používaný v oblasti virtuální reality, her, vizuálních simulací a modelování, napsaný v jazyce C++ s podporou *OpenGL*. V tomto projektu bude využit především pro zobrazení scény a realizaci základní interakce s uživatelem.

Kapitola 4

Implementace

V této kapitole se zabývám výslednou implementací systému, která vychází z návrhu řešení. Jednotlivě vytvořené třídy odpovídají oblastem definovaným v části návrhu systému. Kromě popisu implementovaných tříd uvedu i komplikace, se kterými jsem se v průběhu vývoje setkal a zmíním, jak byly řešeny. Tyto poznatky mohou být následně použity při vývoji implementace 3D manuální segmentace, například formou pluginu do programu *3DimViewer*, popřípadě jiných aplikací, zabývajících se touto problematikou.

V první části kapitoly je popsán proces spuštění programu a jeho následné řízení pomocí uživatelských vstupů. Druhá část se zabývá architekturou scény a způsobem reprezentace v ní obsažených objektů. Poslední část, nazvaná aplikační pozadí, popisuje výpočty, operace ale i datovou strukturu, poskytující potřebnou logickou podporu pro zobrazování objektů.

4.1 Jádru aplikace

4.1.1 Zpracování obrazových dat

Základ řešení, pracovně pojmenovaného *Virtual Surface Editor*, tvoří třída *VSE*, která je zodpovědná za spuštění programu, načtení dat a vytvoření hlavní scény. Implementace je inspirována modulem *mdsSliceView* z knihovny *MDSTk*. Stejně jako tento modul i *VSE* dědí ze třídy *mdsView*, čímž získává podporu následujících funkcí:

- Parsování vstupních argumentů
- Logování do souboru
- Čtení ze vstupního kanálu

První z nich je využita pro zpracování informací o vstupu zdrojových dat. Ty jsou načítána buď pomocí příslušně zformátovaného souboru nebo vstupní roury. Parametry spuštění jsou definovány následovně:

Parametr	Funkce
< file:name	Čtení dat z stdin
-i file:name	Načtení dat ze souboru

Tabulka 4.1: Parametry spuštění programu.

Takto spuštěná aplikace načte a zpracuje uvedené zdroje obrazových dat do formátu třídídimenzionální matice.

4.1.2 Interakce s uživatelem

V případě úspěšně načtených vstupních dat probíhá vytvoření základní scény pomocí třídy `osgViewer` z knihovny *OSG*. Realizace scény je blíže popsána v části 4.2.

Jejím úkolem je mimo jiné i zpracování uživatelských vstupů, což je zajištěno použitím třídy `VSEKeyEventHandler`. `VSEKeyEventHandler` kromě základní práce se scénou, jako je přiblížení, rotace, posun objektu atd., implementuje i zpracování či iniciaci následujících rozšířených uživatelských požadavků:

Změna pozice aktuálního řezu, zajištěné voláním metody `moveBack/moveFront` třídy `Plane` a následným požadavkem o aktualizaci průsečíků separační plochy a řídicí sítě s aktuálním řezem.

Cyklické přepínání orientace řezu ve všech třech osách. Realizace požadavku probíhá voláním metody `changeAxis` třídy `Plane` a stejně jako u změny pozice, i požadavkem na aktualizaci průsečíků.

Přidání bodu do řídicí sítě se skládá z několika částí. První je reakce na stisk tlačítka myši, kdy dochází k uložení informace o aktuální pozici kurzoru. Díky tomu, je při uvolnění stisku tlačítka možné určit, zda šlo o vložení bodu nebo jeho posun.

Následně je funkcí `checkInsertPoint` ověřeno, zda byl požadavek o vložení realizován v oblasti řezu, a to pomocí projekce kurzoru vůči scéně. V případě průniku objektem scény reprezentujícím řez je volán proces realizující vkládání bodu. Nakonec je provedena aktualizace zobrazení průsečíků řídicí sítě a nově vyhlazené separační plochy.

Pohyb s bodem řídicí sítě je podobný jako metoda vkládání, s několika rozdíly. Při stisku tlačítka se provádí vyhledání řídicích bodů ve specifické vzdálenosti od pozice kurzoru. Tím získáme informaci o bodu, se kterým bude manipulováno. Ta je následně využita při tahu myši, kdy se křivka, definující průsečík řídicí sítě a řezu, mění v aktuálním čase, v závislosti na pozici taženého řídicího bodu.

Stejně jako v případě vkládání, i zde se po uvolnění stisku tlačítka v případě potřeby realizuje vložení řídicích bodů na pozice průsečíků řídicí sítě a řezu. Poté je opět provedeno vyhlazení separační plochy a aktualizace zobrazení průsečíků.

Změna typu zobrazení objektů, která je blíže popsána v části 4.2.1, probíhá na požadavek uživatele postupně, formou stisknutí příslušné klávesy. Logickou podporu této funkcionality, jako je cyklická změna stavu či vyhledávání příslušných objektů, zajišťuje metoda `changeStyle`.

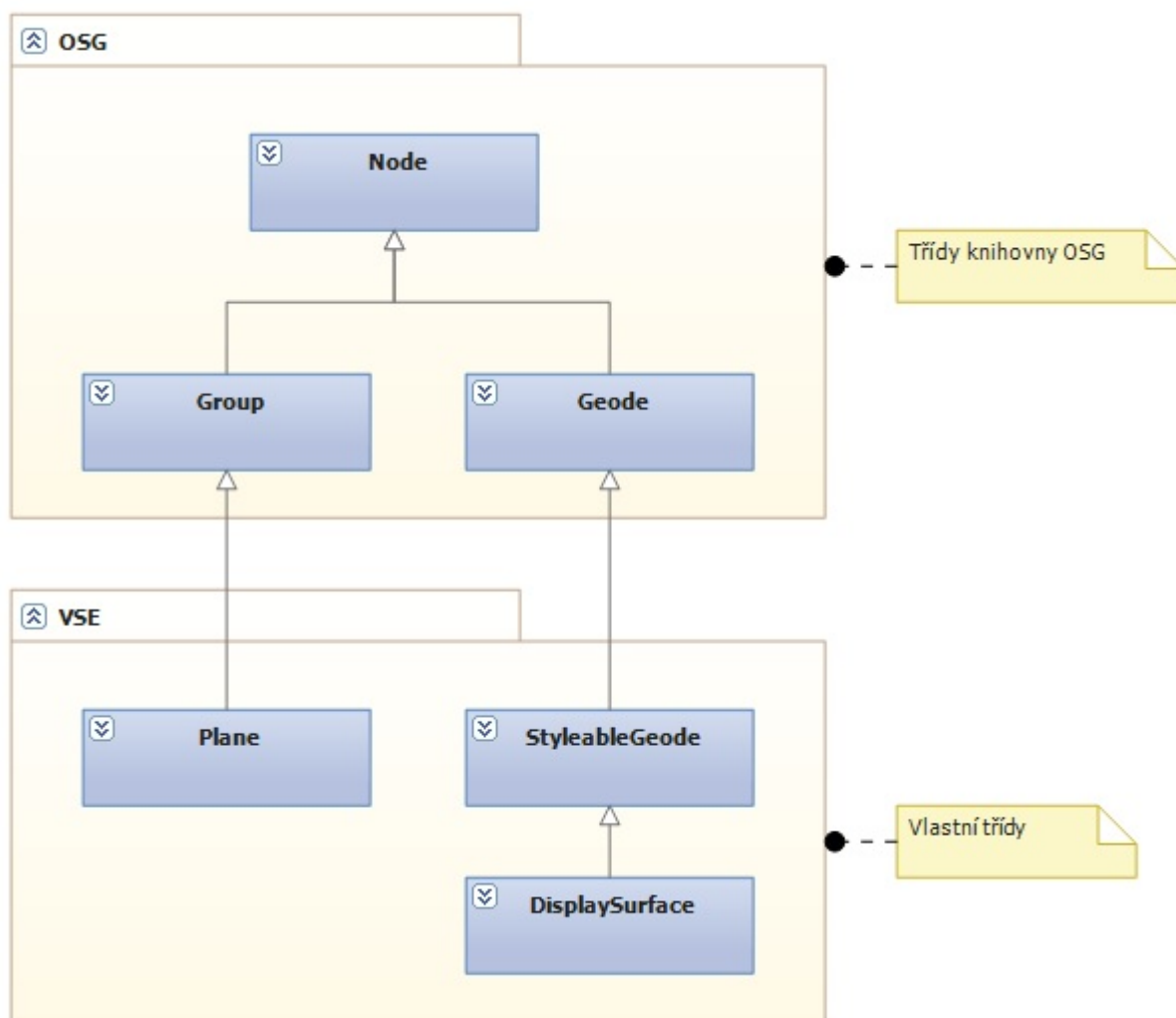
Volba stupně vyhlazení je poslední uvedenou funkcionalitou. Jedná se o nastavení požadovaného stupně vyhlazení a jeho následného provedení funkcí `smooth`.

4.2 Zobrazení

Pro pochopení způsobu implementace scény je potřeba znát architekturu zobrazení pomocí knihovny *OSG*. Ta definuje scénu jako stromovou strukturu hierarchicky řazených uzlů, tzv. *Nodes*, kde každý z nich představuje reálný objekt modelované scény. Toho bylo využito i při implementaci současného řešení, kde scéna obsahuje tři uzly:

- Řídící síť
- Separační plocha
- Zobrazení řezu

Obrázek 4.1 popisuje konstrukci scény z pohledu tříd. Z něj je patrné, že jako hlavní bázevá třída byla použita `Node`, umožňující zobrazování objektů ve scéně.



Obrázek 4.1: Vyrovnání sítě při vložení bodu na její hranu.

4.2.1 Separační plocha a řídicí síť

Instance třídy `DisplaySurface` představuje objekt reprezentující definici zobrazení pravidelného čtyřúhelníkového polygonu. V projektu je tak realizována vizualizace separační plochy a řídicí sítě. Použitím bázevé třídy `StyleableGeode`, získáme možnost volby jednoho ze tří následujících zobrazení:

- *Hide* - Označuje skrytí objektu ve scéně

- *Wire* - Definuje tzv. drátové zobrazení, neboli vykreslení objektu pomocí úseček
- *Full* - Značí vykreslení objektu plně dle jeho definice

Všechny tři stavy však podporuje pouze instance reprezentující separační plochu. V případě řídicí sítě není použití stylu *Full* žádoucí a proto tato volba není umožněna.

4.2.2 Rovina řezu

Stejně jako v případě `DisplaySurface`, i třída `Plane` realizuje zobrazení objektu ve scéně. Konkrétně se jedná o vizualizaci řezu diskrétních dat, reprezentující jednotlivé snímky, získanou odkazem na příslušnou datovou strukturu při konstrukci objektu. Takto načtené snímky jsou zobrazeny jako textura jednoho ze tří řezů, čímž je umožněn pohled na CT/MR data ve všech třech osách.

Změna polohy řezu je implementována diskrétně po jednotlivých snímcích. Při každém požadavku na posun řezu je vypočítána nová poloha geometrického modelu s použitím definovaného kroku. Dále je provedeno načtení následujícího snímku obrazových dat a také příslušná aktualizace textury. Obdobný proces probíhá i při změně osy řezu.

Dalším úkolem této třídy je realizace průsečíků řídicí sítě a separační plochy s aktuálním řezem. Oba tyto průsečíky jsou definovány křivkou odpovídající průřezu polygonu v dané pozici. Knihovna *OSG* však nepodporuje výpočet průniků dvou objektů, a proto bylo zapotřebí tuto funkcionalitu naimplementovat.

Výpočet průsečíku aktuálního řezu a separační plochy či řídicí sítě je realizován výběrem jejich čtyřúhelníkových primitiv, protínajících aktuální pozici řezu. Princip výběru spočívá v postupném ověřování průniku jednou z hran primitiva s aktuálním řezem. To je zajištěno funkcí `calculationIntersectionPoint`, která v případě kolize vrací pozici těchto bodů. Aplikováním procesu na všechny primitiva polygonu získáme seznam pozic průniků. Ty jsou následně použity jako vrcholy křivek definující tvar průsečíků.

V případech, kdy řez prochází přímo vrcholy primitiv řídicího polygonu, musí být uloženy dodatečné informace, protože pouze právě v této situaci je možné body řídicí sítě editovat. K tomu účelu slouží třída `CutPoint`, která poskytuje informaci o pozici bodu v prostoru a také o umístění v datové struktuře.

4.3 Aplikační pozadí

4.3.1 Datová reprezentace plochy

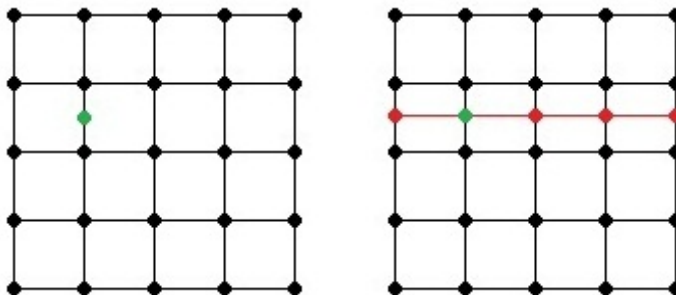
Pro realizace výpočtů a zobrazení separační plochy či řídicího polygonu je potřeba uchovávat informace o počtu vrcholů, jejich logickém rozmístění a také pozici v prostoru. Veškeré tyto informace jsou uloženy v dvojrozměrném poli, které si můžeme představit jako matematickou reprezentaci pravidelného čtyřúhelníkového polygonu. Tato struktura je definována ve třídě `MathSurface` s řadou podpůrných funkcí.

Nejzajímavější z nich je vkládání bodů, které je komplikováno požadavkem na zachování pravidelné čtyřúhelníkové sítě. Třída identifikuje dvě různé možnosti vkládání bodů:

Na hranu sítě

V tomto případě je bod vkládán přímo mezi dva již existující vrcholy. S takovouto situací se setkáme například v případě požadavku o přidání bodu v řezu, kde již jsou ostatní

body realizovány. Pro zachování pravoúhlé sítě je potřeba po vložení bodu síť vyrovnat. Obrázek 4.2 ukazuje situaci před a po vložení bodu na hranu sítě.



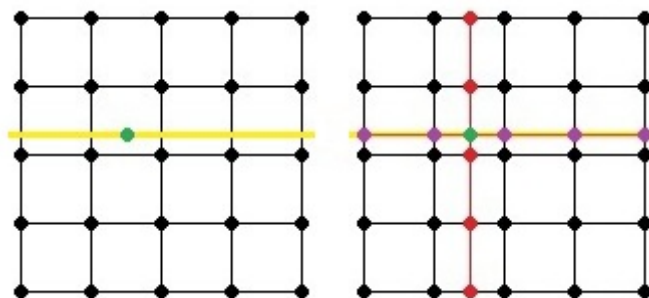
Obrázek 4.2: Vyrovnání sítě při vložení bodu na její hranu.

Realizaci této problematiky lze nalézt v metodě `insertPoint` třídy `MathSurface`. Podle informací, získaných z instance třídy `CutPoint`, jejíž odkaz je předán parametrem metody, lze určit požadovanou pozici pro vložení bodu. Před samotným vložení je však potřeba vytvořit nový vektor bodů na dané ose. To je obstaráno metodou `createEmptyLine`. Vytváření nových bodů však nemůže probíhat jednoduchým výpočtem středu hran čtyřúhelníků, jelikož by docházelo k deformaci požadovaného tvaru separační plochy. Z tohoto důvodu metoda `createEmptyLine` využívá pro výpočet pozice nových bodů třídu `FourPointMethod`, čímž je dosaženo zachování geometrie sítě. Posledním krokem je pak vložení požadovaného bodu na nově vytvořenou pozici.

Vložení bodu do elementu sítě

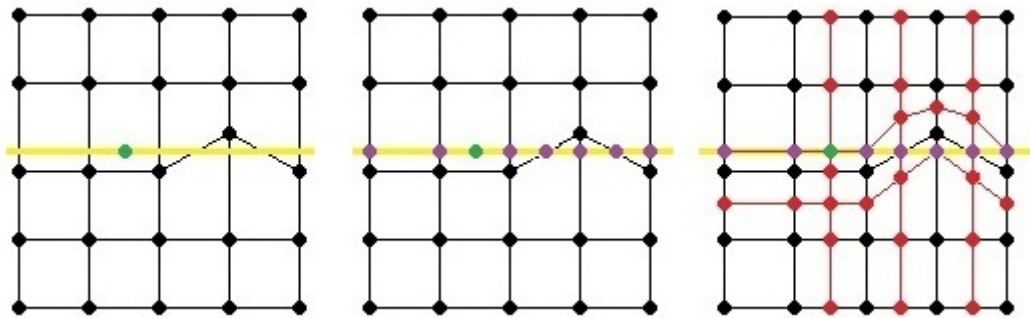
Během tohoto procesu je nový bod vkládán mimo hranu sítě. Algoritmus vložení je podobný jako u vkládání bodu na hranu, pouze se aplikuje dvakrát a to z důvodu potřeby provést vyrovnání sítě v obou osách. Tato metoda je použita při požadavku o vložení bodu do řezu, kde zatím žádné další řídicí body neexistují.

Implementace této metody však není tak jednoduchá, jak teorie naznačuje. Se značnou komplikací se lze setkat při vkládání bodů na výrazně deformované síti. To je dáno požadavkem na realizaci vložení řídicích bodů na pozici průsečíků řídicí sítě a řezu. Pro přehlednost uvádím příklad vložení řídicích bodů dle řezu, na základní a deformované síti.



Obrázek 4.3: Vložení řídicích bodů dle pozice řezu.

Jak je vidět z obrázku 4.3, prvním případě nenastává žádný problém - body se vkládají na průniky hran řídicí sítě a řezu. Pro vyrovnání na pravidelnou síť je potřeba vložit pouze



Obrázek 4.4: Vložení řídicích bodů dle pozice řezu na deformovanou síť.

jednu řadu nových bodů.

Jiná situace je v případě, kdy řez prochází současně oběma osami řídicí sítě. Po vložení bodů na pozice průsečíků, je následná úprava do pravidelné sítě výrazně složitější a zbytečně komplikuje výsledný polygon. Celá tato situace však nastává, pouze pokud je řídicí síť editována ve více řezech, což ve většině případů však není potřeba.

Výše popsanou problematiku vkládání bodů do elementů sítě řeší v první řadě metoda `realizeLine`, která provede vytvoření potřebných řádků a sloupců pomocí funkce `pushPointsOnYLine`, popřípadě `pushPointsOnXLine`, která navíc následně na definované pozice vloží body průsečíků. Poté je, v případě potřeby, volána metoda `insertPoint` a postup je nadále analogický, jako v případě vložení bodu na hranu sítě.

4.3.2 Vyhlazování

Vytvoření vyhlazené plochy zajišťuje metoda `createSmooth` třídy `Subdivision`. Jedinými vstupy této metody jsou požadovaný stupeň vyhlazení a odkaz na instanci třídy `MathSurface`, definující řídicí síť, na jejímž základě bude plocha vytvořena.

Dle návrhu řešení a specifikace algoritmu v části 3.3.1, je pro vyhlazení použito *Subdivision* schéma *Kobbelt*, které jako základ využívá čtyřbodovou interpolační metodu. Ta je reprezentována třídou `FourPointMethod` a podporuje výpočet středního bodu v závislosti na čtyřech okolních, a to pomocí funkce `calculate`.

V okrajových částech sítě však nejsou všechny potřebné body k dispozici. Proto je potřeba vytvořit virtuální body v závislosti na jejich okolí. To je realizováno metodou `createVirtualPoint` třídy `FourPointMethod` a to následujícím vztahem:

$$p_v = 2 * p_1 - p_0 \quad (4.1)$$

Jak ze vzorce vyplývá, virtuální bod p_v je obrazem bodu p_1 dle p_0 .

Následný proces vyhlazování křížově aplikuje funkci `fourPointMethod` na všechny vrcholy vyhlazované sítě. Výsledkem je zahuštěná síť o příslušný počet bodů, reprezentovaná třídou `MathSurface`. Tento proces je definován jako jedna úroveň vyhlazení a v případě potřeby může být opakován, dokud se nedosáhne požadovaného detailu vyhlazované plochy.

4.3.3 Transformace pro zobrazení

`MathSurface` reprezentuje pouze datovou strukturu pro uložení řídicí sítě či separační plochy. Proto je potřeba implementovat převod do formátu vhodného pro zobrazení - jedná se

o instanci třídy `DisplaySurface`.

Tento převod obstarává třída `SurfaceConvector` zahrnující metodu `toOSGGeometry`. Ta ze vstupní datové reprezentace sítě, v podobě třídy `MathSurface`, vytváří příslušný geometrický model reprezentovaný instancí třídy `Geode`, jejímž potomkem je třída `DisplaySurface`. Tím je zajištěna tvorba vizuálního modelu datové struktury plochy.

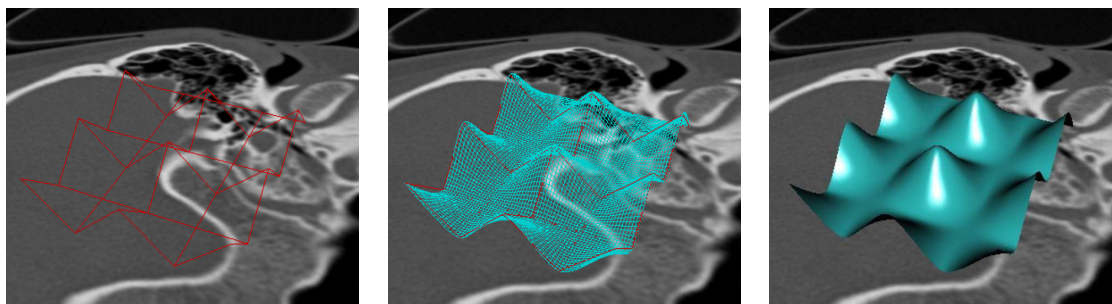
Kapitola 5

Výsledky

Cílem této kapitoly je prezentovat výsledky řešení a zároveň ověřit, zda byly definované požadavky splněny. První část kapitoly uvádí přehled výsledné implementace řešení. Druhá část je zaměřena na ověření hlavního cíle této práce, kterým je realizace usnadnění separace artefaktů během ruční segmentace. I když hodnocení uživatelského přínosu je značně subjektivní, můžeme výsledky konfrontovat se současným řešením v praxi, uvedeným v podkapitole 2.3. Poslední část je věnována přehledu výsledků výkonnostních testů procesu vyhlazování plochy. To je důležité především pro vymezení možností použití aplikace.

5.1 Přehled řešení

Výstupní řešení, dle zadání, poskytuje možnost editace vektorové *Subdivision* plochy pomocí pravidelné čtyřúhelníkové řídicí sítě. V následující části demonstrativně uvádím výsledky implementace řídicí sítě a výsledné vyhlazené plochy v plném a drátovém zobrazení.



(a) Řídící síť

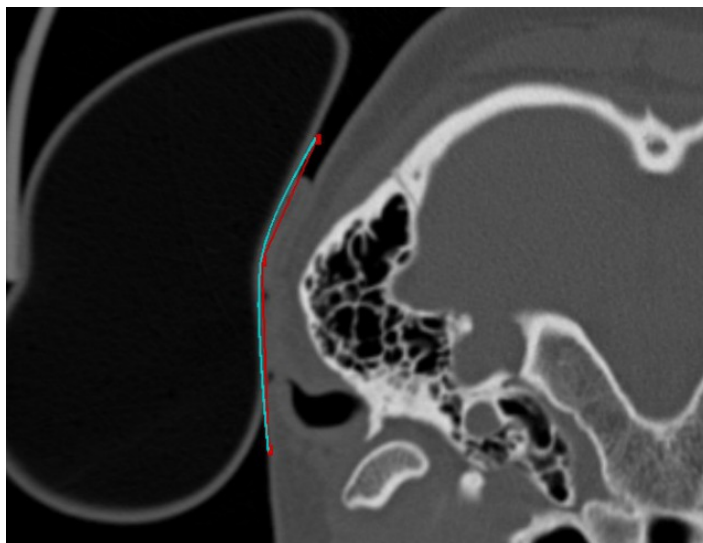
(b) Separáčn plocha v mdu *Wire*

(c) Separáčn plocha v mdu *Full*

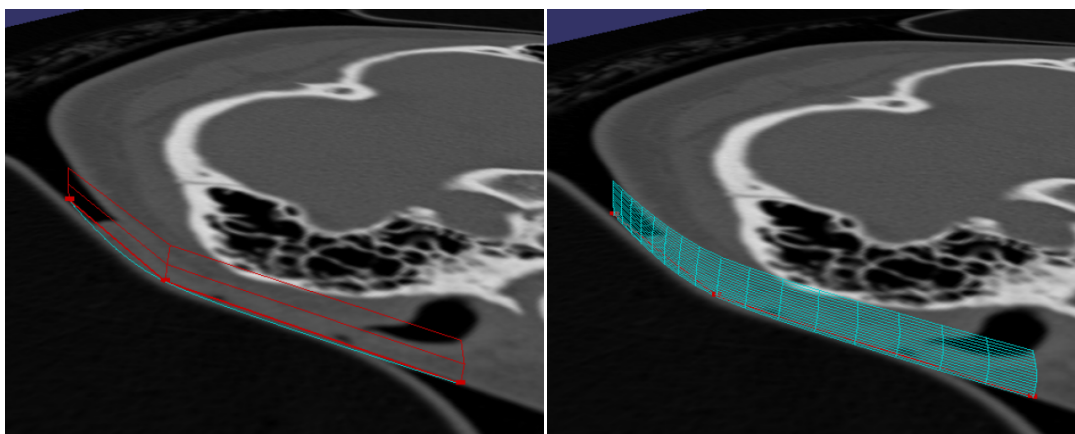
Obrzek 5.1: Ukzka implementace řidcí ste a vyhlazen plochy v rznch mdech zobrazen.

5.2 Uivatelsk podpora

Jak jz bylo uvedeno vše, hodnocen zrychlen a zjednodušen manuln separace bylo provedeno vči současnmu řešen, kter je popsno v odstavci 2.3, kde byl pouit reln pklad deformovanch vstupnch dat. Proces separace pomoc nov vyvinutho řešen mžeme vidt na nze doloench pkladech.



Obrázek 5.2: Editace hranice separace pomocí VSE.



Obrázek 5.3: Výsledná řídící síť a separační plocha.

Pro vytvoření separační hranice bylo postačující definovat 9 řídicích bodů rozložených ve třech různých vrstvách. Zbývající části jsou dopočítány automaticky, a proto není potřeba uživatelského zásahu ve všech vrstvách. Tím se významně snížil čas potřebný k realizaci hranice separace.

5.3 Výkonnostní testy

Výpočet separační plochy v reálném čase může značně snížit použitelnost aplikace. Implementovaná uživatelská volba stupně vyhlazení sice dokáže tento nedostatek značně omezit, ale v případech vysoké složitosti řídicí sítě by mohlo docházet ke ztrátě potřebných detailů. Z tohoto důvodu byly provedeny testy vyhlazování separační plochy. Jejich hlavním cílem je poskytnout informace o době potřebné pro vyhlazení plochy v různých situacích.

Z výsledků testů, uvedených v tabulce 5.1, je patrné, že aplikace může být bez problémů použita do realizace 20-30 řídicích bodů, se 4. stupněm vyhlazení. Při použití více řídicích bodů je potřeba tento stupeň příslušně snížit. I nejsložitější použitou řídicí sít s obsahem

Stupeň vyhlazení	Rozměr řídicí sítě			
	3×3	5×5	10×10	20×20
1	0.002	0.004	0.031	0.275
2	0.005	0.019	0.211	1.682
3	0.021	0.165	1.415	4.336
4	0.197	1.174	2.573	9.728

Tabulka 5.1: Přehled doby potřebné na vytvoření separační plochy v sekundách.

400 bodů však lze v rozumném čase vykreslit. V praxi se ale s touto situací pravděpodobně nesetkáme, jelikož editace tak velkého počtu řídicích bodů by byla značně časově náročná.

Kapitola 6

Závěr

Cílem této práce bylo proložit vektorovou Subdivision plochu 3D objemovými daty za účelem usnadnění separace nežádoucích artefaktů pomocí ruční segmentace. Jak vyplývá z výsledků, tento požadavek byl splněn. Použití separační plochy přineslo kýžené zkrácení doby procesu separace ve více vrstvách. Výstupní aplikace dokáže pracovat na reálných datech, a proto je vhodná pro demonstrační účel, za kterým byla vyvíjena.

Komplikace, se kterými jsem se během vývoje setkal, jsou dostatečně popsány, stejně jako implementace jejich řešení. Díky tomu je možné získané informace využít za účelem vývoje dalších projektů v této oblasti.

Během uživatelského testování aplikace byly odhaleny nedostatky řešení spojené s použitím na členitém povrchu. V těchto případech je pro definování detailu potřeba použít více řídicích bodů. Bohužel následně dochází k rychlému zahušťování řídicí sítě vlivem snahy o udržení její pravidelnosti. Zde se nabízí potenciální směr dalšího výzkumu, a to především v použití nepravidelné řídicí sítě.

Neméně důležitým přínosem této práce bylo získání osobních zkušeností v oblasti výzkumu dané problematiky a také návrhu a implementace výstupního řešení. Díky tomu jsem si prohloubil dosavadní zkušenosti s vývojem v jazyce C++ a především se seznámil s postupy práce ve virtuálním prostoru, která je z mého pohledu komplikovaná, avšak velmi zajímavá, a doufám, že budu mít možnost se této problematice věnovat i nadále.

Literatura

- [1] Corporation, E. C.: Compensating severe non-linearities caused by high-density objects. 2010, [Online].
URL http://www.exxim-cc.com/metal_artifact_reduction.html
- [2] DICOM: DICOM Standard. 2010, [Online].
URL ftp://medical.nema.org/medical/dicom/2009/09_01pu.pdf
- [3] Eric, W.; Grimson, L.; Smith, C.: Semi-Automatic Medical Image Segmentation. 2001, [Online].
URL http://people.csail.mit.edu/lauren/publications/odonnell_ms_thesis.pdf
- [4] Haacke, E. M.; Cheng, N. Y.; House, M. J.; aj.: Imaging iron stores in the brain using magnetic resonance imaging. *Magnetic Resonance Imaging*, ročník 23, č. 1, 2005: s. 1 – 25, ISSN 0730-725X.
URL <http://www.sciencedirect.com/science/article/B6T9D-4FJ880N-3/2/b767dfb6cb7d3b4549c946af691f6e9b>
- [5] Kršek, P.; Černochová, P.; Kaňovská, K.; aj.: Možnosti aplikace prostorových geometrických modelů lidských tkání v dentální chirurgii. In *VII Brněnské implantologické dny*, Medical Faculty of Masaryk University, 2004, ISBN 80-210-3334-7, s. 12–12.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=7403
- [6] Kršek, P.; Krupa, P.: Human tissue geometrical modelling. In *Applied Simulation and Modeling*, The International Association of Science and Technology for Development, 2003, ISBN 0-88986-384-9, s. 357–362.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=7333
- [7] Kršek, P.; Krupa, P.: Problematika 3D modelování tkání z medicínských obrazových dat. *Neurologie pro praxi*, ročník 6, č. 3, 2005: s. 149–153, ISSN 1335-9592.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=7907
- [8] Laboratory, D.: 3DimViewer. 2010, [Online].
URL <http://www.3dim-laboratory.cz/en/software/3dimviewer/>
- [9] Lakare, S.: *3D Segmentation Techniques for Medical Volumes*. Research Proficiency Exam, State University of New York, 2000.
- [10] Španěl, M.: *Delaunay-based Vector Segmentation of Volumetric Medical Images*. Dizertační práce, 2011.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=9452

- [11] University, L. K.; Kobbelt, L. P.: A Subdivision Scheme for Smooth Interpolation of Quad Mesh Data. 1998.
- [12] Wikipedie: Magnetická rezonance — Wikipedie: Otevřená encyklopedie. 2011, [Online].
URL http://cs.wikipedia.org/w/index.php?title=Magnetick%C3%A1_rezonance&oldid=6584782
- [13] Wikipedie: Výpočetní tomografie — Wikipedie: Otevřená encyklopedie. 2011, [Online].
URL http://cs.wikipedia.org/w/index.php?title=V%C3%BDpo%C4%8Detn%C3%AD_tomografie&oldid=6700913
- [14] Zorin, D.; Schröder, P.: Subdivision for Modeling and Animation. Technická zpráva, SIGGRAPH 2000, 2000, course Notes.

Příloha A

Obsah CD

- Zkompilovaný program s potřebnými knihovnami (**bin/**)
- Dokumentace programu (**doc/**)
- Zdrojové kódy (**src/**)
- Elektronická verze technické zprávy (**tech/**)
- Informace nutné pro spuštění a ovládání programu (**README.txt**)