

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

LUKÁŠ ZACHAR

BRNO 2008



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 HUMAN TISSUES 3D MODELLING BY CT/MR DATA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ ZACHAR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PŘEMYSL KRŠEK, Ph.D.

BRNO 2008

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Zachar Lukáš**

Obor: Informační technologie

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

Kategorie: Počítačová grafika

Pokyny:

1. Stručně prostudujte problematiku 3D modelování lidských tkání v medicíně
2. Analyzujte problematiku 3D interaktivního modelování
3. Navrhněte programový systém pro 3D interaktivní modelování lidských tkání na základě CT/MR dat
4. Implementujte a otestujte navržený systém
5. Zhodnoťte dosažené výsledky a stanovte další vývoj projektu

Literatura:

1. Žara J., Beneš B., Felkel P.: Moderní počítačová grafika. 1. vyd. Praha, Computer press 1998, 448 s., ISBN 80-7226-049-9
2. Drastich A.: Zobrazovací systémy v lékařství. 1. vyd. Brno, Rektorát VUT v Brně 1990, 512 s., ISBN 80-214-0220-2

Při obhajobě semestrální části projektu je požadováno:

- Splňte první tři body zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

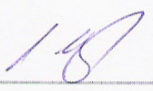
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kršek Přemysl, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2
L.S.


doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení:

Bytem:

Narozen/a (datum a místo):

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta

se sídlem

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP:

Vedoucí/ školitel VŠKP:

Ústav:

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v*:

tištěné formě – počet exemplářů

elektronické formě – počet exemplářů

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

Abstrakt

Cieľom tejto práce je vytvoriť systém, umožňujúci tvorbu 3D modelov tkanív z CT alebo MRI snímok. Vstupné dáta sú vo formáte DICOM. Výber tkanív je určený užívateľom systému podľa ním zadaného rozsahu hodnôt v snímkoch. Aktuálny výber tkanív je farebne odlíšený. Tvorba polygonálneho modelu je uskutočnená pomocou algoritmu Marching cubes. Výsledný model je možné uložiť do formátu VRML alebo STL.

Klíčová slova

pochodujúce kocky, modelovanie tkanív, segmentácia obrazu

Abstract

The aim of this thesis is to create a system, which would make a 3D model of tissues based on CT or MRI scans. Input data is in the DICOM format. Selection of tissues is done by the user by specifying range of values in scans. The current selection is highlighted. The creation of polygonal model is done by the Marching cubes algorithm. It is possible to save the model in format VRML or STL.

Keywords

marching cubes, tissues modelling, image segmentation

Citace

Lukáš Zachar: Interaktivní 3D modelování lidských tkání na základě CT/MR dat, bakalářská práce, Brno, FIT VUT v Brně, 2008

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

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Přemysla Krška, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Zachar
12. mája 2008

Poděkování

Ďakujem vedúcemu bakalárskej práce Ing. Přemyslovi Krškovi, Ph.D. za odbornú pomoc, cenné pripomienky a návrhy pri riešení tejto práce. Ďakujem Matejovi Grochalovi za kontrolu formy textu.

© Lukáš Zachar, 2008.

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
2	Rozbor problematiky	3
2.1	Požiadavky na systém	3
2.2	Zdroje dát	3
2.3	Objemové dáta	4
2.4	Segmentácia	4
2.5	Metóda Marching Cubes	6
3	Návrh	9
3.1	Analýza	9
3.2	Štruktúra	10
3.3	Použité nástroje	11
3.3.1	MDSTk	11
3.3.2	OpenSceneGraph	11
3.4	Zjednodušenie Marching cubes	12
4	Implementácia	13
4.1	Beh programu	13
4.2	Triedy	13
4.2.1	CameraManipulator	13
4.2.2	Cube	13
4.2.3	EventHandler	14
4.2.4	GUI	15
4.2.5	MC	15
4.2.6	OSGScene	16
4.2.7	Rez	16
4.2.8	Threshold	17
4.2.9	VolumeModel	17
5	Výsledky	18
5.1	Model panvy	18
5.2	Model hlavy	19
6	Zhodnotenie a záver	22
6.1	Funkčnosť systému	22
6.2	Možnosti rozšírenia	22
6.3	Zhrnutie	23

Kapitola 1

Úvod

V súčasnosti sa nachádzajú nové možnosti využitia spolupráce počítačovej grafiky a medicíny. Medzi hlavné oblasti patria nové výukové systémy (napríklad simulácie operácií), virtuálna endoskopia, príprava a plánovanie operácií a zjednodušenie diagnostiky ochorení. Pri terajšom postupe musí diagnostika prebiehať s pomocou vyškolených pracovníkov – rádiológov, ktorí sú schopní vytvoriť si mentálny obraz na základe väčšieho počtu snímok a nájsť v ňom prípadné patogénne znaky. Medzi nevýhody tohto postupu okrem pracovnej vyťaženia expertov patrí aj vytváranie bariér pri predvádzaní liečebných postupov laickému pacientovi. Tento postup tiež komplikuje prácu chirurgom pri plánovaní postupu operácií. Jedným z cieľov aplikácie technológií počítačovej grafiky v medicínskom využití je tvorba validných 3D modelov. Využitím týchto modelov sa výrazne zjednoduší príprava na medicínske zákroky a zároveň sú ľahšie predstaviteľné aj pre pacienta - laika.

Obsahom tejto bakalárskej práce je návrh a implementácia ľahko ovládateľného systému, ktorý umožňuje interaktívne vytvárať modely ľudských tkanív. Celá problematika je opísaná v šiestich kapitolách. Do prvej kapitoly patrí tento úvod. V druhej kapitole sa nachádza podrobnejší rozbor problematiky spracovania medicínskych obrazov, je tu tiež uvedený popis algoritmu Marching Cubes. Tretia kapitola je venovaná návrhu systému, vysvetľuje jeho časti a predstavuje použité nástroje. V štvrtej kapitole je opísaná implementácia tohto systému. Obsahuje bližší popis riešenia jednotlivých tried. Piata kapitola ilustruje výslednú prácu so systémom a predstavuje výsledky modelovania tkanív. Posledná, šiesta kapitola, je venovaná zhodnoteniu systému a návrh možných rozšírení do budúcnosti.

Kapitola 2

Rozbor problematiky

V tejto kapitole sa nachádza rozbor problémov spojených s medicínskymi systémami a modelovaním tkanív. Nachádza sa tu tiež popis kľúčového algoritmu Marching Cubes.

2.1 Požiadavky na systém

V medicínskom využití platia tri hlavné kritériá: validita, realističnosť a cenová dostupnosť. [11] Praktickým dôsledkom týchto kritérií je nutnosť opatrnej práce s medicínskymi obrazmi a ich optimalizácie. Nie je prípustné stratiť akýkoľvek podstatný detail ako dôsledok optimalizácie výsledného obrazu. Zároveň musí systém priniesť výrazné zníženie nákladov, či už finančných alebo časových. Dôraz je na efektívnosť výsledného použitia.

2.2 Zdroje dát

Pri veľkom počte diagnostických úkonov je veľmi dôležité poznať vnútro tela pacienta. Prvým prístrojom, ktorý to umožnil, bol röntgen. Jeho vlastnosťou je projekcia trojrozmerných dát na plochú dosku, produkuje sa tak tzv. sumačný obraz, pri ktorom sa priemety jednotlivých orgánov sčítajú. [1] Pre mnohé vyšetrenia je to postačujúce, ale napríklad pri plánovaní operácií je nutné zhotoviť väčší počet snímok. Inak nie je možné zistiť presnú polohu orgánov a ochorení. S rozvojom technológií sa postupne začali vyvíjať nové možnosti neinvazívneho pohľadu do ľudského tela. V súčasnosti sú k vytvoreniu komplexného obrazu vnútorného prostredia sú vhodné najmä:

- Počítačová tomografia (CT), vychádza z princípu röntgenu. Vytvára sa séria snímok z rôznych uhlov, ktoré sú počítačom vyhodnotené a je skonštruovaný obraz vnútorného prostredia pacienta. Oproti röntgenu je značne zvýšená škála identifikovateľných absorpčných rozdielov. Taktiež sú použité citlivejšie snímače, kvôli zníženiu množstva škodlivého žiarenia, ktoré musí pacient prijať pri vyšetrení.
- Magnetická rezonancia (MRI), spočíva vo využití charakteristického magnetického momentu jadier chemických prvkov. Vplyvom vonkajšieho magnetického poľa dochádza k vychýleniu jadier. Po odstránení tohto poľa je odmeraná veľkosť indukovaného prúdu pri vrátení sa jadra atómu do pôvodnej polohy. Namerané hodnoty sú spracované počítačom a je zhotovený obraz vnútra pacienta. Prednosťou tejto metódy v porovnaní s CT je presnejšie zachytenie mäkkých tkanív a absencia röntgenového

ziarenia. Nevýhodami sú väčšia finančná náročnosť a zvýšená tvorba artefaktov vo výslednom obraze (napríklad deformácie obrazu vplyvom prítomnosti piersingu).

Výsledky vyšetrení sú najčastejšie uložené vo formáte DICOM ako séria rezov. Približné veľkosti údajov zobrazuje tabuľka 2.1.

Typ vyšetrenia	Matica	Hustota byte/pixel	Mbytov/obraz	Obrazov/vyšetrenie	MBytov/vyšetrenie
CT	512 x 512	2	0.524	25	13.1
MRI	512 x 512	2	0.524	30	15.7

Tabuľka 2.1: Typická veľkosť vyšetrení (podľa [5])

DICOM

Digital Imaging and Communications in Medicine (DICOM) je štandard pre ukladanie, manipuláciu, tlač a príjem medicínskych obrazov. Je zložený z definície súborového formátu a sieťového protokolu. Umožňuje integráciu skenerov, serverov, pracovných staníc, tlačiarň a sieťových zariadení od viacerých výrobcov do jedného komunikačného a archivačného systému. [6] Je tým umožnená bezproblémová prenosnosť dát. Kvôli vylúčeniu možnosti chybného pridelenia snímok nesprávne pacientovi sú základné identifikačné údaje pevnou súčasťou súboru. Každý súbor môže obsahovať len jedny obrazové dáta. Ich reprezentácia je určená typom dát, sú podporované rôzne formáty, napr. JPEG, RLE. Štandard obsahuje aj možnosť kompresie, táto vlastnosť nebýva vždy implementovaná.

2.3 Objemové dáta

Dáta získané pomocou CT alebo MRI je možné zrekonštruovať do trojrozmernej pravouhlej mriežky diskretných hodnôt. Jednotlivé hodnoty sú reprezentované voxelmi¹. Hlavným problémom takýchto objemových dát je veľké množstvo údajov v nich obsiahnutých a z toho vyplývajúce zvýšené nároky na kapacitu pamäte a výkon procesora pri spracovaní. Problématická je rotácia objemu iným uhlom ako pravým. Pri prevzorkovaní dochádza k strate informácií. [12] Výhodou oproti spojitě reprezentovaným objektom je presné zachovanie vnútornej štruktúry objektu a jednoduchšia reprezentácia nepravidelných útvarov.

Pre umožnenie modelovania týchto dát je potrebné nájsť spôsob ako ich previesť do povrchovej reprezentácie. Častým postupom je aproximácia povrchu sieťou trojuholníkov. Tento povrch je určený miestami s konštantnou hodnotou voxelov. Najznámejším a najpoužívanejším algoritmom uskutočňujúcim tento prevod je *Marching Cubes* (*pochodujúce kocky*).

2.4 Segmentácia

Pod pojmom segmentácia v počítačovej grafike myslíme rozdelenie obrazu do viacerých regiónov podľa zadaného kritéria. Dôvodom je možnosť ich ďalšieho spracovania, napríklad modelovania. Existuje mnoho rôznych prístupov, krátky prehľad uvedený v tejto sekcii vychádza z [4].

¹voxel = volumetric pixel

Prahovanie (Thresholding)

Princíp prahovania spočíva v rozdelení obrazu do dvoch skupín: objekt a pozadie. Rozdelenie je vykonané na základe porovnania skalárnej hodnoty skúmaného pixelu so zadanou prahovou hodnotou. Túto hodnotu môže zadať užívateľ, prípadne môže byť založená na štatistickom spracovaní obrazu (napríklad medián obrazu). Prahovanie dosahuje najlepšie výsledky, pokiaľ je možné presne určiť hodnotu hranice. Nevýhodou tohto prístupu je neuvažovanie priestorovej charakteristiky obrazu. Prahovanie je náchylné k nezvládnutiu šumu alebo nehomogenity oblasti.

Rast regiónov (Region growing)

Vychádza z podobného princípu ako semienkové vyplňanie oblasti. Na začiatku je manuálne označený pixel, ktorý patrí do vybranej oblasti. Následne sa výber rozšíri do okolia, až kým nebude splnené kritérium pre ukončenie výberu. Jedným z kritérií môže byť detekcia hrany. Podobne ako prahovanie, aj rast regiónov je citlivý na nehomogenitu vybranej oblasti a šum. Pri splnení určitých podmienok môže dôjsť k spojeniu dvoch rozdielnych oblastí.

Klasifikátory (Classifiers)

Klasifikátory patria medzi techniky rozpoznávania vzorov. Pred použitím v praxi je potrebné tieto metódy naučiť segmentovať obraz pomocou trénovacej množiny dát. Existuje viacero prístupov, ako môžu byť trénovacie množiny aplikovaná pri segmentácii. Princíp jednoduchého klasifikátora typu najbližší sused: každý pixel je zaradený do tej istej triedy ako bod trénovacej množiny s najbližšou hodnotou. Rozšíreným parametrickým klasifikátorom je Bayesov klasifikátor, ktorý využíva predpoklad nezávislého náhodného rozdelenia vzoriek.

Zhlukovanie (Clustering)

Zhlukovanie využíva podobný princíp ako klasifikátory, ale nepoužíva trénovaciu množinu. Vychádza z princípu iteratívnej segmentácie obrazu a tvorby charakteristických znakov tried. Najpoužívanejšími algoritmi sú: K-Means, Fuzzy-C-Means a maximalizácia očakávania. K-Means spočíva v iteratívnom počítaní priemernej intenzity každej segmentačnej triedy, priraduje každý pixel do triedy s najbližšou priemernou hodnotou. Fuzzy-C-Means zovšeobecňuje K-Means pridaním fuzzy logiky. Maximalizácia očakávania kombinuje zhlukovanie s predpokladom Gausovho rozdelenia hodnôt. Zhlukovanie je závislé na počiatkovej segmentácii (parametroch).

Umelé neurónové siete (Artificial neural networks)

Sú paralelnou sieťou uzlov napodobňujúcou učenie živých tvorov. Učenie sa tejto siete je dosiahnuté pomocou zmien váh spojení medzi uzlami. Neurónové siete reprezentujú paradigma strojového učenia. Najrozšírenejšie je ich použitie ako klasifikátorov, váhy sú určené pomocou trénovacej množiny a neurónová sieť segmentuje nové dáta. Pretože obsahuje veľa spojení, je možné zahrnúť priestorové vzťahy do klasifikátorov.

Deformovateľné modely (Deformable models)

Sú to fyzicky motivované modelovo založené techniky určenia hraníc regiónov pomocou využitia uzavretých kriviek. Využíva sa simulovanie pôsobenia vonkajších a vnútorných síl

na krivku. Segmentácia začína umiestnením uzavrenej krivky do blízkosti hranice. Následne sa na krivky aplikuje pôsobenie síl. Vnútorne sily sa snažia zabezpečiť hladký povrch krivky a vonkajšie sily nútia krivku prilnúť k hranici objektu. Vďaka tomuto prístupu dosahuje táto metóda dobré výsledky pri obraze s obsahom šumu. Jej nevýhodou je nutnosť počiatočného umiestenia krivky a špecifikácie jej parametrov.

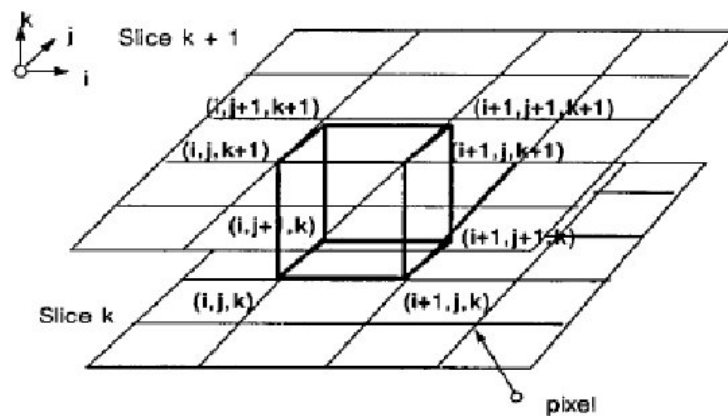
Atlasom riadené prístupy (Atlas-Guided Approaches)

Tento prístup segmentácie je veľmi efektívny pokiaľ existuje šablóna, podľa ktorej je možné rozdeliť obraz do tried. Pozostáva z atlasu, ktorý vznikne na základe segmentovanej anatómie. Postup segmentácie pozostáva z nájdenia transformácie atlasu do spracovávaného obrazu. Kvôli anatomickým rozdielom je nutné použiť kombináciu lineárnych a nelineárnych transformácií. Pretože atlas je už segmentovaný, všetky informácie o štruktúre sú prenesené do výsledného obrazu. Segmentácia s použitím atlasu dosahuje najlepšie výsledky pri segmentácii štruktúr, ktoré sú stabilné vo väčšine populácie.

2.5 Metóda Marching Cubes

William E. Lorensen a Harvey E. Cline publikovali v roku 1987 algoritmus, ktorý nazvali Marching Cubes. Vytvára trojuholníkový model povrchov s konštantnou hustotou z objemových dát. [12] Popis algoritmu a obrázok kocky sú prevzaté z [2].

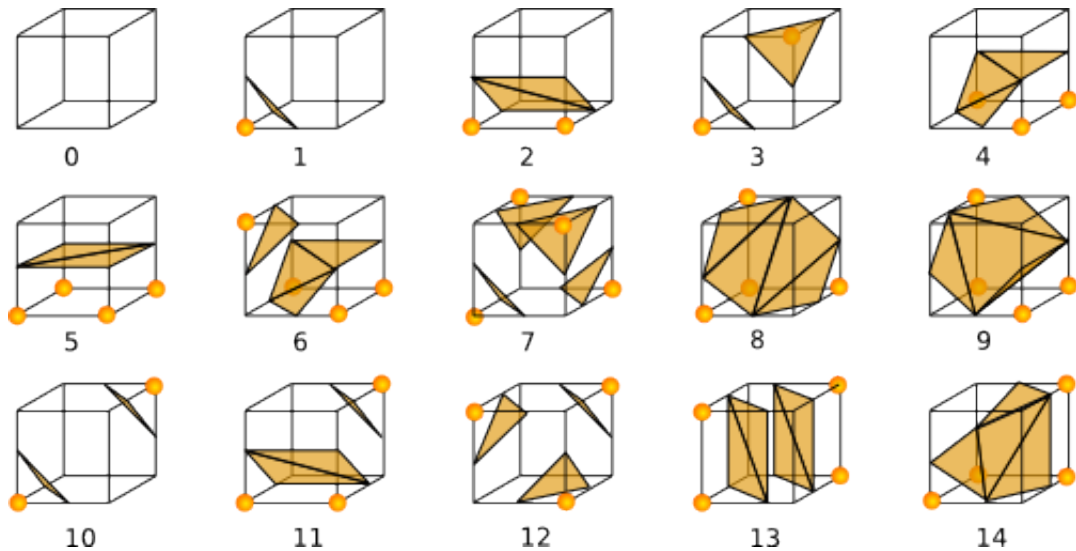
Existujú dva hlavné kroky pri riešení problémov s konštrukciou v povrchu. Prvým je lokalizácia povrchu na základe užívateľom zadaných hodnôt, druhým krokom je výpočet normál povrchu. Algoritmus využíva prístup rozdeľuj a panuj (divide-and-conquer) k lokalizácii povrchu v logickej kocke tvorenej ôsmimi pixelmi z dvoch príľahlých rezov.



Obrázok 2.1: Pochodujúca kocka (prevzaté z [2])

Algoritmus určí ako plocha pretne kocku, potom sa presunie na ďalšiu kocku (odtiaľ pochádza názov algoritmu). Aby bolo možné určiť rez kockou, je každému vrcholu priradená logická jednotka, pokiaľ leží na vybranom povrchu a logická nula pokiaľ leží mimo tohto povrchu. Povrch pretína kocku na tých hranách, kde jej jeden vrchol leží vnútri a druhý mimo tohto povrchu. Na základe toho predpokladu je možné určiť topológiu povrchu vo

vnútri kocky. Celkovo je možných 256 kombinácií, v ktorých môže vybraný povrch pretnúť kocku. Vďaka využitiu symetrií je možné najšť 15 unikátnych tried (na obrázku 2.2). Najprv sa aplikuje poznatok, že topológia povrchu sa nezmení, ak sú hodnoty vo všetkých vrcholoch vymenené. Zvyšné možnosti sú tvorené rotáciami základných tried. Pri triede 0 sú všetky vrcholy vnútri resp. mimo povrchu, preto sa pri nej nevytvárajú žiadne trojuholníky. Všetky ostatné triedy produkujú jeden až štyri trojuholníky.



Obrázok 2.2: Možné prípady pri Marching cubes [7]

Využitím indexu je získaná pretnutá hrana. Index vytvorený podľa binárneho ohodnotenia vrcholov. LSB² predstavuje hodnotu vrcholu 0. Konkrétny bod pretnutia na hrane sa vypočíta lineárnou interpoláciou. Interpolácie vyššieho stupňa prinášajú iba malé zlepšenie oproti lineárnej interpolácii, preto nie je výhodné ich použiť. Posledným krokom je výpočet normál pre jednotlivé trojuholníky. Využíva sa vlasnosť, že povrch sa nachádza medzi dvoma objemami, ktoré sa líšia svojou hodnotou a smer normály je totožný so smerom vektoru gradientu dát. Vektor gradientu je možné získať deriváciou funkcie hodnôt

$$\nabla f = (g_0, g_1, g_2)$$

Pre určenie vektoru gradientu na povrchu sa najprv určia vektory gradientu na vrcholoch kocky a potom sú lineárne interpolované do bodu prieniku. Gradient vo vrchole (i, j, k) je vypočítaný pomocou rozdielov troch koordinátov

$$G_x(i, j, k) = \frac{D(i+1, j, k) - D(i-1, j, k)}{\Delta x}$$

$$G_y(i, j, k) = \frac{D(i, j+1, k) - D(i, j-1, k)}{\Delta y}$$

$$G_z(i, j, k) = \frac{D(i, j, k+1) - D(i, j, k-1)}{\Delta z}$$

kde $D(i, j, k)$ je hodnota pixelu (i, j) na reze k a $\Delta x, \Delta y, \Delta z$ sú dĺžky hrán kocky. Vydelenie gradientu jeho dĺžkou zaručuje normovanú normály.

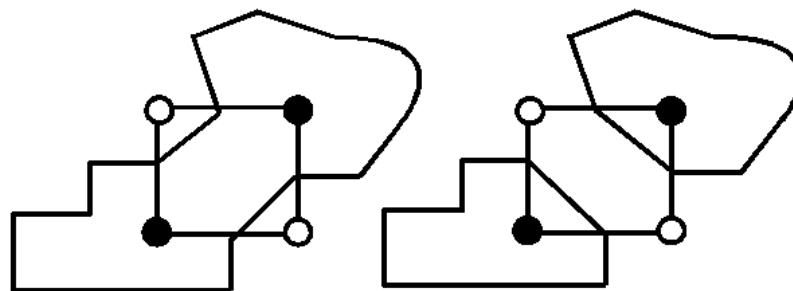
²najmenej významný bit

Celý algoritmus možno zhrnúť do 7 bodov:

1. Načítanie štyroch rezov do pamäte
2. Spracovanie dvoch rezov a vytvorenie kocky zo štyroch bodov z prvého a druhého rezu
3. Výpočet indexu kocky podľa hodnôt vo vrcholoch
4. Vyhľadanie pretnutých hrán na základe indexu
5. Na základe lineárnej interpolácie nájdenie bodu prieniku na hrane
6. Výpočet normály
7. Výstup trojuholníkov a normál

Jedným z negatív tohto algoritmu je produkcia veľkého množstva trojuholníkov. Výnimkou nie je vytvorenie viac ako 500 tisíc trojuholníkov.

Ďalšou nevýhodou je produkovanie nespojitého povrchu pri určitých kombináciách jednotlivých prípadov. Konkrétne sa jedná o prípady tried 3,4,6,7,10,12,13. Jeden spôsob riešenia problému a bližší prehľad je možné nájsť v [8]. Ilustrácia nejednoznačnosti v triedach 5 a 10 je na obrázku 2.3.



Obrázok 2.3: Nejednoznačnosť triedy 5, resp. 10

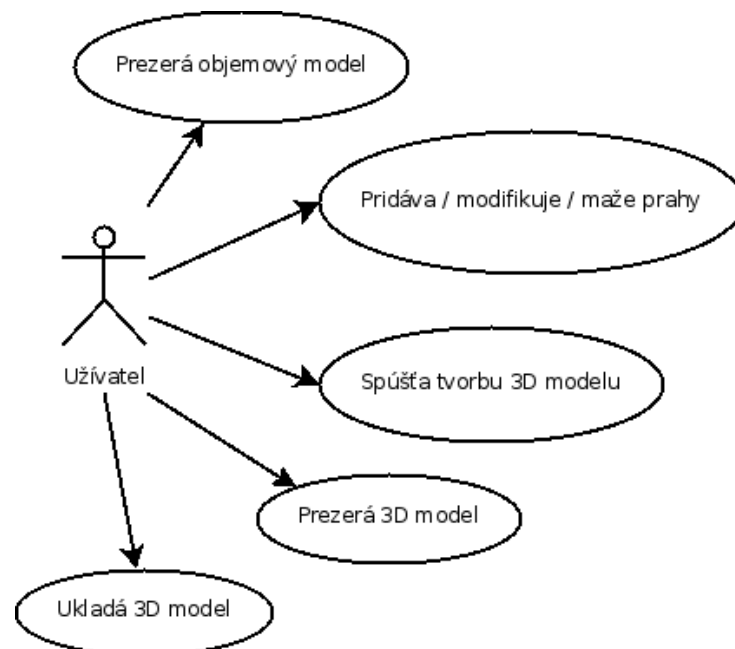
Kapitola 3

Návrh

Kapitola uvádza štruktúru systému a krátky prehľad použitých nástrojov.

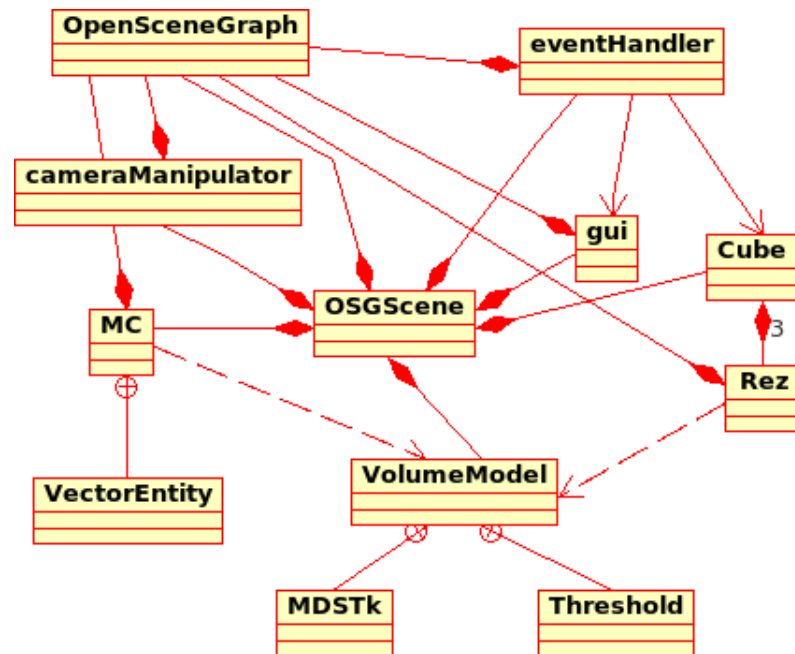
3.1 Analýza

Cieľom tejto práce je vytvoriť systém, ktorý umožní interaktívnou formou vybrať oblasti záujmu z medicínskeho objemového modelu a následne vytvoriť 3D polygonálny model vybranej oblasti. Princíp celého systému spočíva v interaktívnej segmentácii objemových dát užívateľom, ktorý určením dolného a horného prahu hodnôt zvolí tkanivá pre spracovanie. Následne sa tento interval použije v algoritme Marching Cubes pri zisťovaní príslušnosti vrcholu do vybraného povrchu. Pokiaľ hodnota v danom vrchole patrí do intervalu prahu, tak je vrchol označený ako vnútorný. V opačnom prípade je označený ako vonkajší. Po spracovaní algoritmom Marching Cubes je výsledný trojuholníkový model zobrazený. Diagram prípadov užitia je na obrázku 3.1.



Obrázok 3.1: Diagram prípadov užitia

3.2 Štruktúra



Obrázok 3.2: UML diagram tried

Systém je výhodné rozdeliť do dvoch logických častí podľa druhu zobrazených objektov. Popis tried je uvedený v sekcii 4.2, UML diagram je na obrázku 3.2.

Objemový model

Prvá časť umožňuje manipuláciu s objemovým modelom, ktorý je multiplanárne zobrazený. Jednotlivé textúry znázorňujúce rezy podľa rovín XY, XZ a YZ sú umiestnené v priestore s ohľadom na skutočnú pozíciu v objeme. Pokiaľ je k modelu priradený aspoň jeden prah tak hodnoty doň patriace sú zvýraznené jeho farbou. Prah je možné pridať priamym zadaním jeho farby a intervalu. Druhým spôsobom je možnosť definície jeho farby a okolia, s následným výberom konkrétneho voxelu. Na základe jeho hodnoty a zadaného okolia sa vytvorí interval. Priradené prahy je možné upravovať. Keď je užívateľ spokojný s určením prahu, môže spustiť tvorbu polygonálneho modelu.

Polygonálny model

Druhá časť zobrazuje výsledný polygonálny model, získaný pomocou algoritmu Marching Cubes. Model je možné uložiť do formátu STL alebo VRML. Pokiaľ by výsledný model nezodpovedal predstavám užívateľa, je možné sa vrátiť späť do prvej časti a upraviť hodnoty prahu.

3.3 Použité nástroje

Pri výbere nástrojov bol kladený dôraz na prenositeľnosť medzi platformami a legálnosť ich použitia v tejto práci. Obidva nástroje spĺňajú obe podmienky.

3.3.1 MDSTk

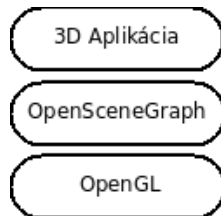
Medical Data Segmentation Toolkit je kolekciou nástrojov pre spracovanie 2D/3D obrazov pôvodne určený pre segmentáciu medicínskych obrazov. Medzi jeho základné vlastnosti patrí multiplatformovosť a jednoduchá modulárna štruktúra. [9]

Umožňuje jednoduchú manipuláciu s medicínskymi dátami. Obsahuje triedu *mdsDensityModel*, ktorá umožňuje pracovať s objemovým modelom. Umožňuje tiež získavať rezy objemového modelu a tiež jeho korektné načítanie z formátu DICOM.

Súčasťou balíku MDSTk je trieda *VectorEntity*, ktorá je určená pre prácu s polygonálnymi modelmi. Udržiava kontext hrán a vrcholov, s možnosťou vyhľadávania susediacich trojuholníkov, výpočtom normál trojuholníkov. Výsledný polygonálny model je možné ukladať celú sieť do formátov STL a VRML.

3.3.2 OpenSceneGraph

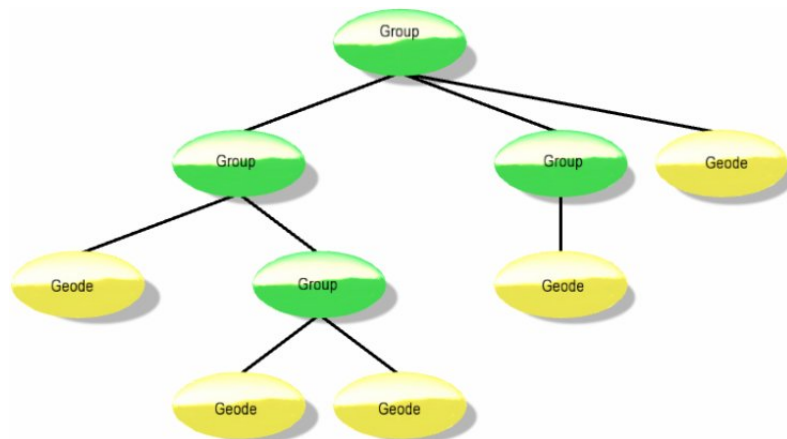
OpenSceneGraph je vysoko výkonný otvorený toolkit pre 3D grafiku. Je využívaný v oblastiach ako vizuálna simulácia, hry, virtuálna realita, vedecké vizualizácie a modelovanie. Je napísaný podľa štandardu C++ a OpenGL, beží na všetkých platformách MS Windows, OSX, GNU/Linux, IRIX, Solaris, HP-Ux, AIX and FreeBSD operačných systémoch. [10] Schéma aplikácie využívajúcej OpenSceneGraph je na obrázku (3.3). Slúži k riadeniu scény



Obrázok 3.3: Zaradenie OpenSceneGraphu

a optimalizovaniu jej vykreslovania. Charakteristikou scény je usporiadanie v tvare stromu, príklad je uvedený na obrázku 3.4. Obsahuje uzly pre transformácie, prepínanie zobrazenia. Umožňuje rôzne nastavenia scény, každý podstrom môže mať vlastné nastavenie osvetlenia, viditeľnosti a podobne. Je tak možné jednoducho vytvárať rôzne oznamy pre užívateľa, dynamicky prepínať zobrazenia podscén bez dlhého čakania na ich vytvorenie. Pre zjednodušenie správy pamäte slúži povinné používanie tzv. smart pointers¹. Jednotlivé objekty je možné použiť aj viackrát, môžu sa líšiť iba zaradením do iného podstromu rôznych transformácií.

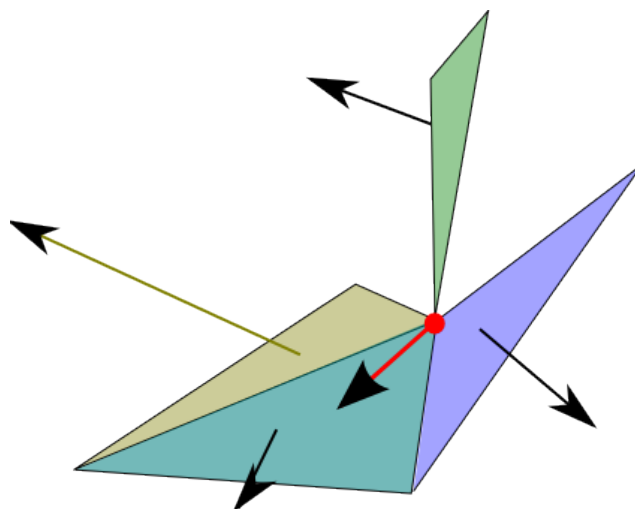
¹ http://en.wikipedia.org/wiki/Smart_pointer



Obrázok 3.4: Typická schéma scény, obrázok je z [3]

3.4 Zjednodušenie Marching cubes

Pôvodný algoritmus počíta normály interpoláciou gradientu. Tento spôsob je príliš časovo náročný, pretože sa musia počítať pre každý generovaný vrchol trojuholníka. Zjednodušenie použité v tomto systéme spočíva v použití váženého priemeru normál trojuholníkov v okolí vrcholu (obrázok 3.5 znázorňuje spôsob výpočtu červeného vrcholu). Pre každý vrchol tak prebehne tento výpočet iba raz.



Obrázok 3.5: Znázornenie výpočtu normály

Ďalším zjednodušením je nahradenie lineárnej interpolácie pretnutia hrany použitím stredu hrany.

Kapitola 4

Implementácia

Kapitola je venovaná popisu výsledku vývoja z pohľadu programátora. Jazykom implementácie je C++, ktorý umožňuje prenositeľnosť medzi platformami na úrovni zdrojových kódov. Hlavnou implementačnou platformou bol Ubuntu Linux(7.10 resp. 8.04). Je použitý OpenSceneGraph 2.2 a MDSTk 0.7.2beta.

4.1 Beh programu

Program vyžaduje parameter s menom a cestu k súboru obsahujúcemu objemové dáta. Po spustení tento súbor načíta, vytvorí jeho multiplanárny model a následne sa tento model zobrazí. Užívateľ ho môže prezerať a interaktívne zadáva hodnoty prahov. Vybrané oblasti sa označujú priradenou farbou k prahu. Až je užívateľ spokojný s výberom oblasti, spustí tvorbu polygonálneho modelu podľa jedného z vybraných prahov. Po skončení tvorby je výsledný model zobrazený a užívateľ ho môže uložiť. Ak by bol s modelom nespokojný, je možné sa vrátiť a opraviť hodnoty prahu.

4.2 Triedy

V tejto časti sú uvedené jednotlivé triedy tvoriace systém. V ich popise sú uvedené skutočnosti, ktoré by nemuseli byť zrejmé z programovej dokumentácie a pokladám ich za dôležité. Taktiež sú tu uvedené myšlienky, na ktorých je daná trieda postavená.

4.2.1 CameraManipulator

Cieľom tejto triedy je implementácia manipulácie s kamerou. Posun je možný len na základe pohybu myši. Trieda je zdedená z *osgGA::TerrainManipulator*, a pôvodný kód manipulátora bol upravený tak, aby pri presune kamery pomocou myši nebola zohľadnená zotrvačnosť pohybu. Všetky ostatné vlastnosti jeho implementácie boli vyhovujúce a preto zostali zachované.

4.2.2 Cube

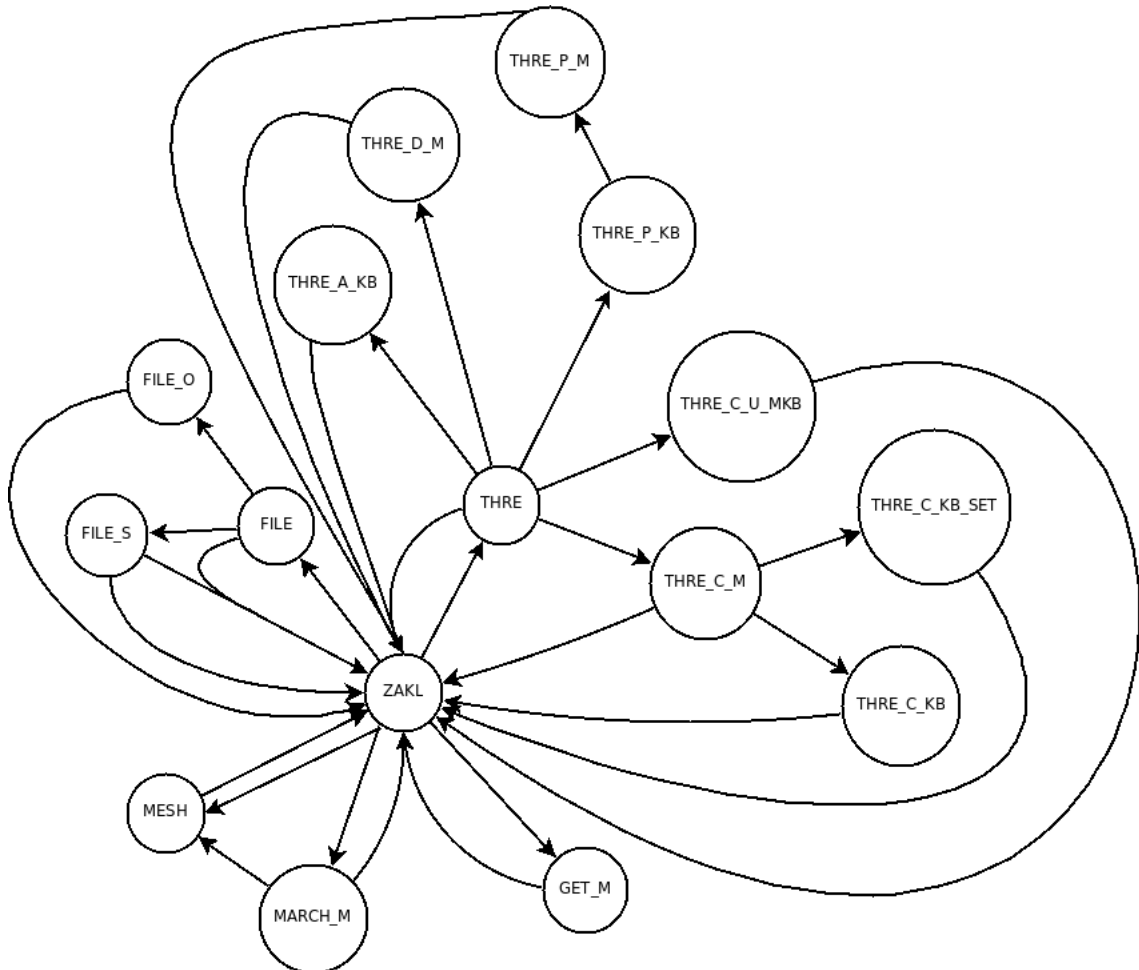
Obsahuje celú geometriu pre multiplanárne zobrazenie objemového modelu spolu s nastavením osvetlenia. Priradenie konkrétneho modelu, ktorý sa má zobraziť, je vyriešené pomocou **konštruktora**. Priradenie modelu je nemenné. Táto trieda je závislá na triede *Rez*, pretože ju používa na zobrazenie rezu danou rovinou. Objekty predstavujúce rez sú

umiestnené v polohe, v akej je rez vedený. Pozícia je určená pomocou umiestnenia rezu do *osg::PositionAttitudeTransform*. Tak je umožnený jednoduchý posun iba zmenou pozície a obnovením textúry rezu. Aktuálne pozície sú uložené v triede *Cube*. Je možné ich získať pomocou metódy **getPosition**. Posun rezov je relatívny k ich aktuálnej pozícii pomocou **movePosition**. Je možné tým zjednodušiť výpočet novej pozície pri požiadavkách na posun mimo tejto triedy (odstránením zisťovania aktuálnej pozície prípadne redundancie uloženia polohy). Obnovenie zobrazenia, teda znovu načítanie textúr, je vynútitelné pomocou **forceRedraw()**. Metódou pre obnovenie pozícií do počiatočného stavu, teda umiestnenie rezov do stredu osi po ktorej sú posúvané, je možné volaním metódy **resetPozicie**.

4.2.3 EventHandler

Predstavuje jadro interakcie s užívateľom. Jej účelom je spracúvať užívateľov vstup, zabezpečiť aktuálne zobrazenie doplňujúcich informácií a správne reagovať na požiadavky užívateľa.

Trieda je zdedená z *osgGA::GUIEventHandler*. Samotnú obsluhu udalostí predstavuje metóda **handle**. Obsluha udalostí sa riadi stavovým automatom podľa obrázku 4.1.



Obrázok 4.1: Schéma stavov obsluhy udalostí

Pri každej zmene stavu zavolá aktualizáciu zobrazenia informácií užívateľovi. Pre obsluhu udalostí, ktorých zdrojom je myš, sú využité metódy **mousePush** a **mouseMove**. Prvá menovaná slúži pre detekciu objektu, do ktorého užívateľ klikol. Rozpoznanie objektu pracuje na základe jeho pomenovania. Pokiaľ je to na základe stavu automatu žiadúce, je pomocou `osgViewer::View::computeIntersections` nájdený označený objekt a je vykonané spracovanie udalosti podľa druhu objektu. Môže sa jednať o výber hodnoty prahu, zistenie hodnoty voxelu z rezu, alebo označenie rezu, ktorý sa bude posúvať.

Metóda **mouseMove** slúži pre spracovanie pohybu myši. Jediným stavom, kedy je aktívna je prípad predchádzajúceho označenia niektorého z rezov. Smer a veľkosť pohybu vychádza z pohybu myši. Tento pohyb je dvojrozmerný a závisí na pozícii kamery. Preto je potrebné transformovať súradný systém kamery na súradný systém scény. Výsledný posun rezu je určený podľa pohybu na príslušnej ose (os X pre rez YZ, os Y pre rez XZ a os Z pre rez XY).

Vstup od užívateľa je realizovaný pomocou zásobníka, kam sa postupne ukladajú stlačené znaky. Manipuláciu s ním uskutočňujú metódy pre pridanie znaku na koniec (**buffAdd**), zmazanie posledného znaku (**buffDelLast**) a úplné zmazanie obsahu zásobníka (**buffFlush**). Aby bolo zaručené zobrazenie aktuálneho obsahu, tak po každej zmene zásobníka nasleduje aktualizácia zobrazenia volaním `GUI::show`.

Manipulácia s prahmi prebieha na základe uloženého vstupu zo zásobníka. Aby bolo urýchlené zadávanie prahov, tak pre farby je možné použiť jedno písmenové skratky (r – červená, g – zelená, b – modrá). Je nutné dodržať formát tvorby prahu (písmeno, resp. číselné hodnoty RGB, dolnú a hornú hranicu prahu, oddelené čiarkami). Rozpoznanie počtu skupín číslíc, resp. prítomnosti písmena prebieha v **parseBuff**. Na základe počtu týchto skupín realizujú metódy **threshAdd**, **updateThresh** a **threshPlane** odpovedajúce činnosti. Posledná menovaná neprijíma celý interval prahu, ale len hodnotu okolia od hodnoty voxelu získanej z rezu. Pri úprave už existujúcich prahov sa nové hodnoty aplikujú až po ich potvrdení.

4.2.4 GUI

Je triedou, ktorej jedinou úlohou je zobrazovanie doplňujúcich informácií užívateľovi. Textové informácie môžu byť zobrazené maximálne v troch riadkoch. Prvý z nich slúži k popisu stavu, v ktorom sa systém nachádza. Ďalšie slúžia k zobrazeniu textového vstupu (iba v niektorých stavoch) a inštrukcií potrebných pre ďalšiu činnosť.

Stav, ktorý má zobrazíť táto trieda je potrebné pri každej zmene nastaviť. Slúži k tomu metóda **show**. Zobrazované textové hlásenia sú nemennou súčasťou triedy. Aby bolo možné zobrazíť vstup užívateľa, je potrebné tento vstup predať tejto triede metódou **show**. Aktualizovanie tohto vstupu závisí na triede `eventHandler`.

K zobrazenému objemovému modelu je možné znázorniť farby a hodnoty priradených prahov. Tvorbu tohto náhľadu zabezpečuje metóda **createThresholdShow**, ktorá farbou prahu zobrazí text rozsahu intervalu hodnôt. Zmenu zobrazenia (zapnuté/vypnuté) tohto prehľadu zabezpečuje metóda **toggle**. Pre prípadné opätovné vytvorenie slúži metóda **update**.

4.2.5 MC

V tejto triede sa nachádza implementácia algoritmu Marching Cubes. Tvorba trojuholníkového modelu tkanív pozostáva z dvoch krokov:

1. Volaním metódy **set** je priradený objemový model a index prahu určený k spracovaniu.
2. Zavolaním metódy **go** dôjde k spracovaniu vstupu algoritmom.

Priebeh spracovania prebieha v dvoch fázach. Tým je zvýšená časová náročnosť celého spracovania, ale je čiastočne eliminovaná pamäťová náročnosť výsledného uloženia trojuholníkovej siete v scéne. Je to možné vďaka využitiu tvorby trojuholníkov pomocou indexov vrcholov, čím sú súradnice každého vrcholu uložené v pamäti iba raz. Ušetrenie miesta v pamäti vyplýva z rozdielnej veľkosti indexu (integer) a súradníc vrcholov (3 x double).

Preto je najprv polygonálny model vytváraný v triede *vctl::MCTris*. Po zapnutí detekcie existencie vrcholov je pri každom vložení trojuholníka uskutočnený test jedinečnosti vrcholu. Pokiaľ sa v modeli už vrchol nachádza tak je použitý pre tvorbu trojuholníka. Tým je zároveň vytvorená sieť hrán a je možné jednoducho zistiť priľahlé trojuholníky.

Po vytvorení tejto siete nasleduje spracovanie do reprezentácie vhodnej pre zobrazenie pomocou OpenSceneGraphu. V prvom rade je vytvorený zoznam vrcholov, každému je priradený index podľa poradia jeho vytvorenia. V tomto kroku sú zároveň vypočítané normály vrcholov. Nasledujúci postup spočíva vo vytvorení trojuholníkov použitím indexov ich vrcholov. Po vytvorení geometrie je nastavené osvetlenie tejto časti scény. Potom je možné pridať výsledný uzol do hlavnej scény a zobraziť ho.

K uloženiu výsledného modelu slúži **saveModel**. Formát výstupného súboru je detekovaný na základe prípony súboru. Samotné uloženie do súboru obsluhujú metódy *vctl::MCTris::LoadSTL* a *vctl::MCTris::SaveSTL*.

4.2.6 OSGScene

Táto trieda je jadrom riadenia činnosti programu. Jej hlavnou úlohou je vytvárať a prepájať ostatné triedy. O zobrazenie scény sa stará *osgViewer::Viewer*, ktorého správanie je možné upravovať nastavením obsluhy kamery a udalostí od užívateľa.

Postup zobrazenia scény pozostáva z volania metódy **setModel**. V jej obsluhu prebehne vytvorenie objemového modelu, konštrukcia jeho multiplanárneho zobrazenia a pridanie tejto konštrukcie do celej scény. V scéne môže byť vždy zobrazený len jeden z modelov (objemový alebo polygonálny), preto sú tieto modely umiestnené v scéne pomocou *osg::Switch*. Takto je umožnené jednoduché prepínanie zobrazenia modelov.

Samotné zobrazenie modelov a spustenie obsluhy udalostí užívateľa prebieha zavolaním **showModel**. Vytvorenie a následné pridanie polygonálneho modelu realizuje volanie metódy **doMarchCubes**. Pokiaľ boli pridané obidva modely je možné prepínať ich zobrazenie pomocou **switchVolumeMesh**.

4.2.7 Rez

Je triedou reprezentujúcou rez objemovým modelom jednou z rovín XY, XZ, alebo YZ. Vzhľadom k tomu, že je väčšina kódu pre obsluhu týchto rezov rovnaká, prípadne sa líši iba parametrom nimi volanej metódy, bolo možné ich zlúčiť do jednej triedy. Určenie typu rezu a priradenie konkrétneho objemového modelu pre zobrazovanie textúr je nutné vykonať už pri volaní **konštruktora**. Rez je zložený zo štvorca a textúry. Vytváraný štvorec má dĺžku hrany 1, následne je zväčšený podľa rozmerov daného rezu. Pamäťový priestor pre textúru je alokovaný podľa rozmerov rezu. Textúra zobrazená na štvorci je len jedna, využitie viacnásobného textúrovania by bolo pre tento prípad príliš zložité. Pre možnosť

rozlíšenia konkrétnych rezov v modeli, je každý rez pomenovaný podľa osi, v ktorej sa môže pohybovať. Obnovenie textúry je prevedené vždy po volaní metódy **update**.

4.2.8 Threshold

Trieda obsahujúca hodnoty dolného a horného prahu spoločne s farbou, ktorou sa odliši od ostatných. Prístup k údajom je priamy, bez použitia set/get metód.

4.2.9 VolumeModel

Základom tejto triedy je atribút typu *mds::img::CDensityVolume*, v ktorom je uložený objemový model. Načítanie a vytvorenie tohto modelu zabezpečuje metóda **load**. Formát súboru musí byť podporovaný toolkitom MDSTk.

Každý model by mal mať priradený aspoň jeden prah. Prah je vytvorený vždy v inej triede ako VolumeModel, ale jeho životný cyklus je ukončený v nej. Preto je potrebné prah po vytvorení priradiť k modelu zavolaním metódy **addThreshold**. Odobratie a následné zmazanie prahu zabezpečuje metóda **delThreshold**. Pokiaľ je potrebné pracovať s niektorým z prahov, napr. chceme upraviť jeho interval, tak na získanie referencie slúži metóda **getThreshold**. Odstránenie a zmazanie všetkých prahov modelu prevádza metóda **delAllThre**.

Hlavnou činnosťou tejto triedy je vytváranie textúr zobrazujúcich rez niektorou zo základných rovín (XY,XZ,YZ) v určenej polohe. Podporovaný je 8 bitový RGB formát textúr. Pretože hodnoty v objemovom modeli sú 16 bitové, tak je potrebné ich lineárne interpolovať do 256 stupňov odtieňov šede. Metóda **fillData** zabezpečuje generovanie textúry. Textúra je vytváraná v už existujúcom poli, aby sa zbytočne nespotreboval procesorový čas. Ten, kto túto metódu volá, musí zabezpečiť korektné alokovaný priestor. Povolenie resp. zakázanie použitia prahov pri tvorbe textúr umožňuje metóda **setShowThresholds**. Pokiaľ je povolené použitie prahu, tak sú voxely doň patriace vykreslené v príslušnej farbe. Pre prípady, v ktorých nie je dôležitá farba, ale len príslušnosť resp. nepríslušnosť do danej oblasti existuje metóda **ThreInOut**. Jedným z jej parametrov je pole, ktoré je naplnené odpovedajúcimi hodnotami. Voxel patriaci do daného prahu je reprezentovaný hodnotou 1. Alokácia naplneného pola je starosťou volajúceho.

K získaniu hodnôt jednotlivých rezov sú použité interné metódy MDSTk. Tieto metódy umožňujú získať celý rez jedným zavolaním. Keďže je predpoklad viacnásobného získania hodnôt rezov, tak by alokácia pamäte pri každom volaní metód bola zbytočne časovo náročná. Vzhľadom ku konštantným rozmerom objemového modelu je možné alokovať priestor pre získavané rezy raz (pri konštrukcii triedy) a využívať ho až do konca existencie triedy.

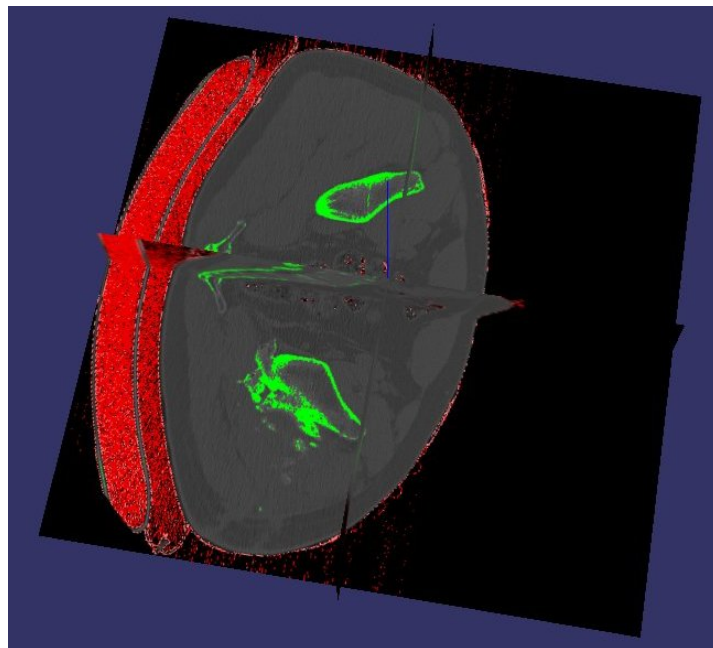
Kapitola 5

Výsledky

Táto kapitola obsahuje ukážky zobrazenia hodnôt prahov a výsledné 3D modely. Zdrojom dát boli CT snímky, ktoré sú voľne dostupné k stiahnutiu na stránkach toolkitu MDSTk. [9]

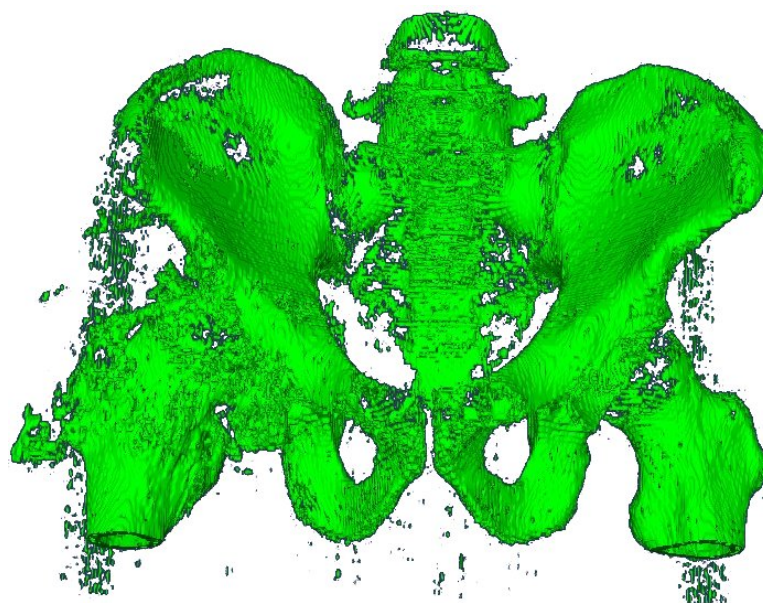
5.1 Model panvy

Prvým testovacím súborom bol model panvy. Snímka programu zobrazujúca segmentáciu obrazu je na obrázku 5.1. Hodnoty prahu označeného zelenou farbou boli zvolené tak, aby vybrali kosti. Zobrazený červený prah je len ilustráciou, že model nemusí mať priradený iba jeden prah.



Obrázok 5.1: Zobrazenie objemového modelu

Výstupom tohto nastavenia je model na obrázku 5.2. Farba prahu je priradená k hotovému modelu. Dĺžka spracovania a počet trojuholníkov modelu je uvedený v tabuľke 5.1.



Obrázok 5.2: Výsledný model kostí

5.2 Model hlavy

Druhým testovacím súborom bol model hlavy. Segmenácia povrchu lebky, teda koža a sliznica je na obrázku 5.3. Na obrázku 5.4 je zobrazený výstupný model. Dĺžka spracovania a počet trojuholníkov modelu je uvedený v tabuľke 5.1.

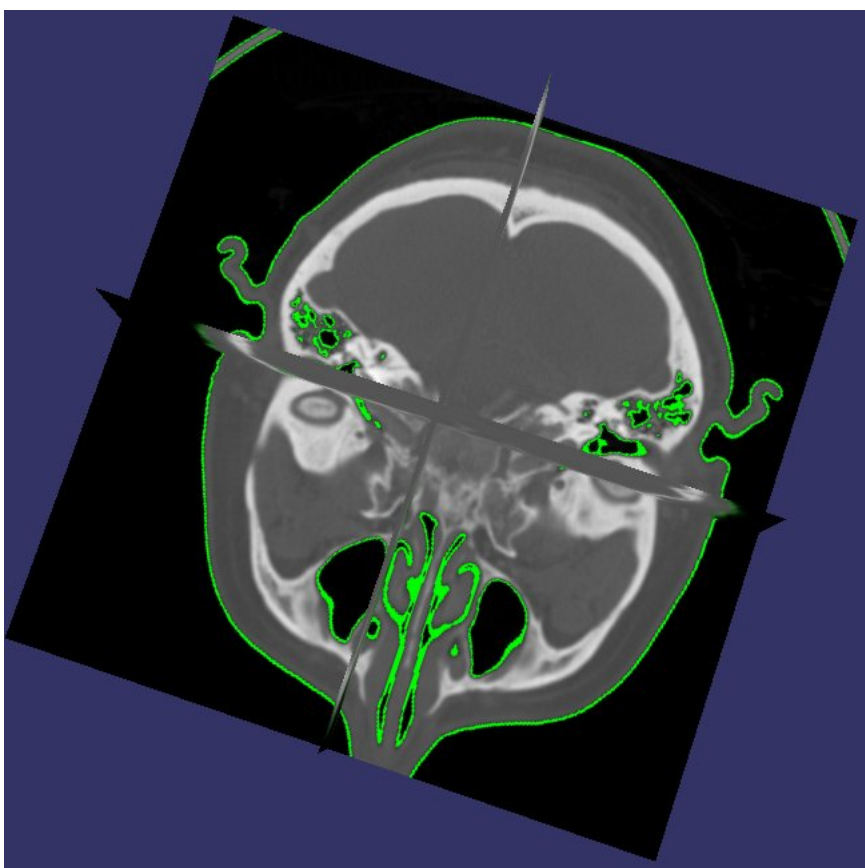
Obrázok 5.5 predstavuje ten istý model hlavy, ale hodnoty prahu sú nastavené aby bol vytvorený model lebky. Je zreteľne vidno spojitú oblasť kostí. Výsledný model je na obrázku 5.6. Dĺžka spracovania a počet trojuholníkov modelu je uvedený v tabuľke 5.1.

Prehľad výsledkov

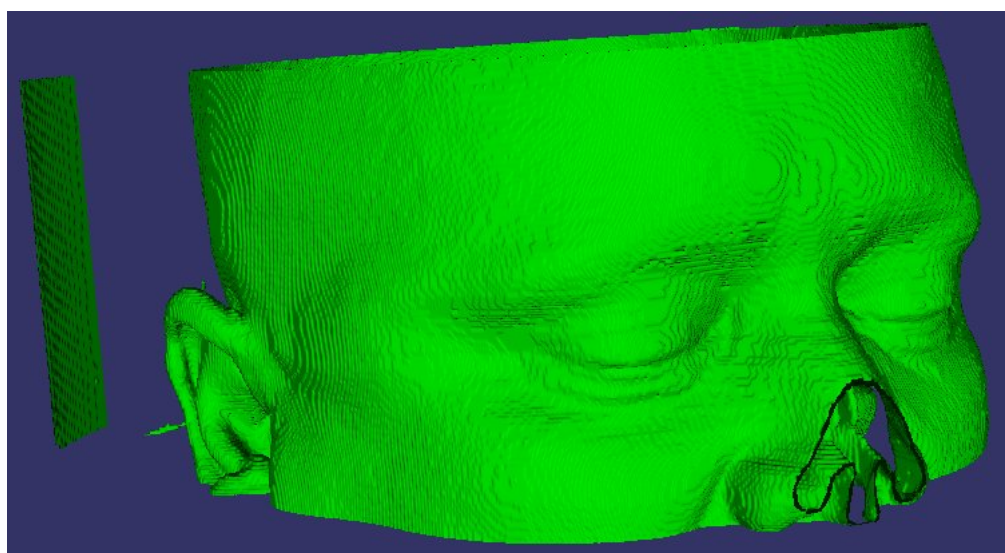
Nasledujúca tabuľka sumarizuje typ vybraného tkaniva, rozsah hodnôt voxelov tvoriacich dané tkanivo, počet trojuholníkov tvoriacich model, čas potrebný k spracovaniu modelu (doba vytvorenia modelu) a celkovú dobu od spustenia vektorizácie po zobrazenie modelu.

Tkanivo	Rozsah hodnôt	Počet trojuholníkov	Doba spracovania	Doba do zobrazenia
Kosť - panva	1252 - 2437	1 089 118	29 s	36 s
Koža, sliznica	210 - 674	2 554 016	98 s	103 s
Kosť - lebka	1113 - 3062	2 677 036	85 s	92 s

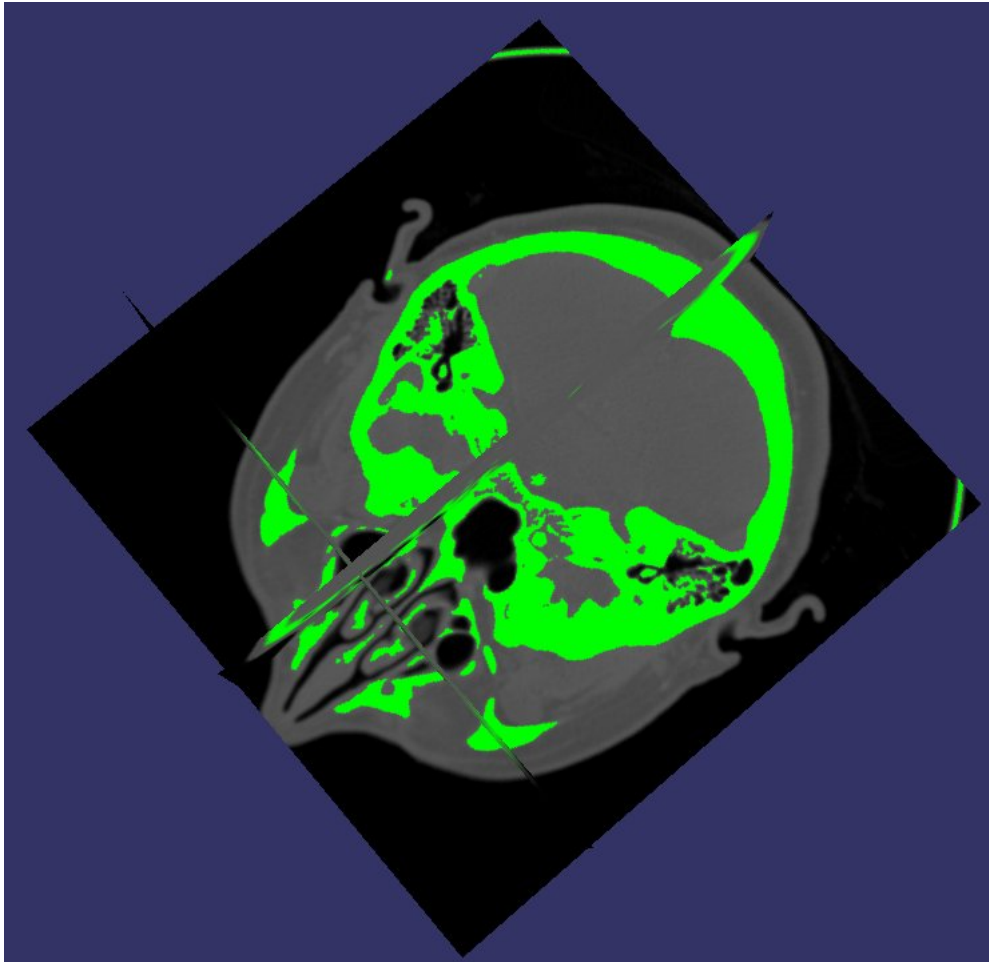
Tabuľka 5.1: Parametre a hodnoty modelov



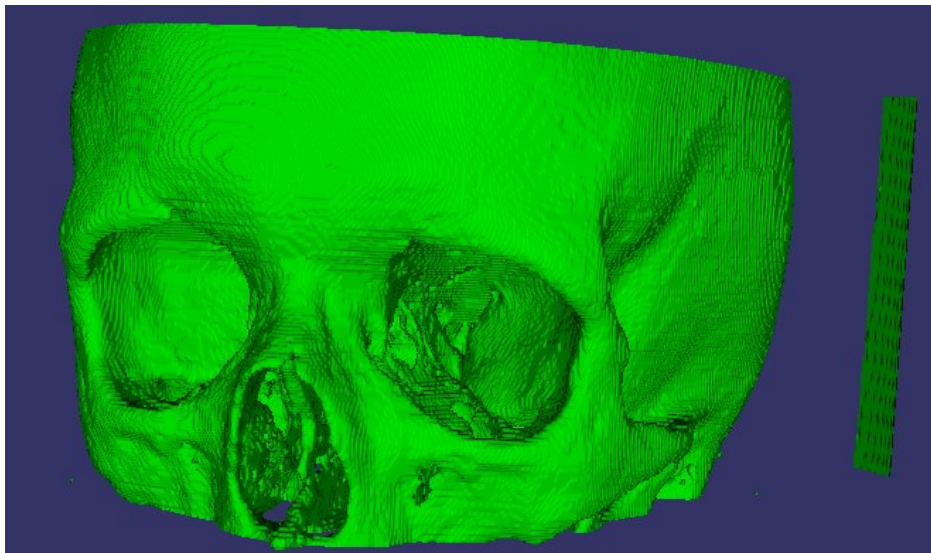
Obrázok 5.3: Segmentácia modelu povrchu hlavy



Obrázok 5.4: Výsledný model povrchu hlavy



Obrázok 5.5: Výber kostí lebky



Obrázok 5.6: Výsledný model lebky

Kapitola 6

Zhodnotenie a záver

6.1 Funkčnosť systému

Z hľadiska užívateľa je najnáročnejšou časťou výber prahov, ktorý väčšinou prebieha na základe opakovaných korekcií. Vďaka interaktívnosti tohto výberu je možné okamžite odhadnúť, či zvolené hodnoty sú správne a prípadne ich modifikovať. Správnosť určenia intervalu prahu má vplyv nielen na výsledný model, ale aj na dĺžku doby spracovania. Čas potrebný k vytvoreniu modelu je priamo úmerný počtu generovaných trojuholníkov. Pokiaľ je prah nastavený nevhodne, tak veľkosť spotrebovanej pamäte a čas behu spracovania sú príliš veľké. Veľké pamäťové nároky nastávajú v prípadoch, pokiaľ je vybraná oblasť rozptýlená po objeme, alebo existuje veľa osamotených voxelov, ktoré nepatria do oblasti a tým túto oblasť rozdeľujú.

Najlepšie výsledky boli dosiahnuté pri tvorbe modelov kostí. Výber hodnôt prahu bol pre ne dobre kontrolovateľný a preto bol zároveň najjednoduchší. Vybrané oblasti boli dostatočne súvislé a tvorba modelu bola pomerne rýchla.

6.2 Možnosti rozšírenia

Zmena spôsobu segmentácie

Spôsob segmentácie pomocou prahovania je síce najjednoduchší, ale často sa vo vybranej ploche vyskytujú voxely, ktoré síce nesplňajú kritérium výberu, ale pre úspešnú tvorbu modelu by mali byť označené. Z dôvodu tejto nehomogenity vznikne v modeli príliš veľa objektov. Riešením je použitie sofistikovanejších foriem segmentácie, napríklad Fuzzy-C-Means.

Zjednodušenie výsledného modelu

Pretože algoritmus Marching Cubes pracuje s každou kockou osobitne, sú rozľahlé plochy modelované veľkým počtom trojuholníkov. Odstránením nevýznamných vrcholov by bolo možné vytvoriť túto plochu menším počtom trojuholníkov bez deformácie modelu a následnej straty informácií. Je možné uvažovať aj o odstránení nepodstatných trojuholníkov, ale táto úprava by mohla spôsobiť neplatnosť modelu.

Grafické rozhranie

Vhodným rozšírením by bolo vytvoriť štandardné grafické užívateľské rozhranie, napríklad pomocou toolkitu wxWidgets. Systém by tým získal prirodzený vzhľad prostredia v ktorom je spustený. Užitočným rozšírením by bola možnosť ukladania často používaných hodnôt prahov a možnosť ich pridávania pomocou menu. Medzi ďalšie rozšírenia užívateľského rozhrania by mohla patriť aj možnosť definície klávesových skratiek užívateľom.

Zobrazenie viacerých 3D modelov

Prvý krok k tomuto rozšíreniu, spracovanie viacerých prahov pri zobrazovaní, je už implementovaný. K plnému využitiu tejto možnosti je potrebné vytvoriť prostredie, v ktorom by užívateľ mohol jednoduchým spôsobom ovládať zobrazenie a výber z týchto modelov.

Editoru 3D modelu

Po zobrazení výsledného modelu je možné spozorovať, že bol spracovaný aj šum. Ak je, až na tento jav, model kvalitný, tak výhodnejším postupom oproti novej segmentácii je odstrániť šum z modelu manuálne. Pridaním editora by celá činnosť prebiehala v jedinom programe.

6.3 Zhrnutie

Tento systém nevychádzal zo žiadneho existujúceho projektu a v súčasnosti nemá žiadnych nasledovníkov. Systém je schopný vytvoriť 3D model, pre svoju činnosť potrebuje pomerne veľké množstvo pamäte. Pred použitím v praxi je potrebné implementovať najmä efektívnejšiu formu segmentácie a zjednodušenie výsledného modelu.

Pri tvorbe tohto systému som si mohol vyskúšať celý cyklus tvorby softvéru, od návrhu až po testovanie. Naučil som sa tiež využívať prostriedky tretích strán a bližšie som spoznal výzvy spojené s modelovaním ľudských tkanív.

Literatúra

- [1] Drastich, A.: *Zobrazovací systémy v lékařství*, ročník 1. Brno: Rektorát VUT v Brně, 1990, ISBN 80-214-0220-2, 512 s.
- [2] Lorensen, W.; Cline, H.: Marching Cubes: A high resolution 3D surface construction algorithm. In *Computer Graphics*, ročník 21, July 1987, ISSN 0097-8930, s. 163–169.
- [3] Martz, P.: *OpenSceneGraph Quick Start Guide*. Computer Graphics Systems Development Corporation, 2007.
- [4] Pham, D.L.; Xu, Ch.; Prince, J.L.: Current methods in medical image segmentation. In *Annual Review of Biomedical Engineering*, ročník 2, August 2000, ISSN 1523-9829, s. 315–337.
- [5] Rhodes, M. L.: Computer Graphics and Medicine: A Complex Partnership. In *IEEE Computer Graphics and Applications*, ročník 17, January-February 1997, ISSN 0272-1716, s. 22–28.
- [6] www stránky: Digital Imaging and Communications in Medicine.
<<http://en.wikipedia.org/wiki/DICOM>>, (apríl 2008).
- [7] www stránky: Marching cube cases.
<<http://commons.wikimedia.org/wiki/Image:MarchingCubes.svg>>, (apríl 2008).
- [8] www stránky: The Marching Cubes.
<<http://www.polytech.unice.fr/~lingrand/MarchingCubes/algo.html>>, (apríl 2008).
- [9] www stránky: Medical Data Segmentation Toolkit.
<<http://www.fit.vutbr.cz/~spanel/mdstk/>>, (apríl 2008).
- [10] www stránky: OpenSceneGraph.
<<http://www.openscenegraph.org/projects/osg>>, (apríl 2008).
- [11] Vidal, F.; Bello, F.; Brodlie, K.; aj.: Principles and Applications of Computer Graphics in Medicine. In *computer graphics forum*, ročník 25, 2006, ISSN 0167-7055, s. 113–137.
- [12] Žára, J.; Beneš, B.; Felkel, P.: *Moderní počítačová grafika*, ročník 1. Praha: Computer press, 1998, ISBN 80-7226-049-9, 448 s.