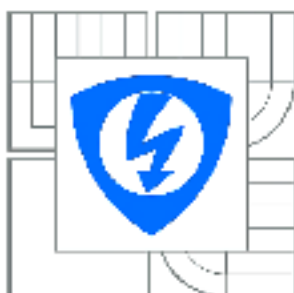




**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNologií**

**ÚSTAV TELEKOMUNIKACÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## **POROVNÁVÁNÍ 3D OBJEKTŮ V JAZYCE JAVA**

COMPARISON OF 3D OBJECTS IN JAVA LANGUAGE

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

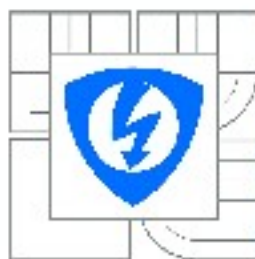
**Bc. TOMÁŠ ZAPLETAL**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ONDŘEJ ŠMIRG**

BRNO 2013



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Diplomová práce

magisterský navazující studijní obor  
Telekomunikační a informační technika

**Student:** Bc. Tomáš Zapletal

**ID:** 119682

**Ročník:** 2

**Akademický rok:** 2012/2013

**NÁZEV TÉMATU:**

**Porovnávání 3D objektů v jazyce JAVA**

**POKYNY PRO VYPRACOVÁNÍ:**

Úkolem diplomové práce je vytvořit program v jazyce JAVA, který bude schopen určit procentuální podobnost dvou 3D objektů a barevně odlišit ty části objektu, které jsou rozdílné. Student porovná několik algoritmů na vyhlazování 3D objektů a zhodnotí na kolik se objekt změnil.

**DOPORUČENÁ LITERATURA:**

[1] NIXON, Mark S a Alberto S AGUADO. Feature extraction and image processin. 2nd ed. Amsterdam: Academic Press, 2008, xv, 406 s. ISBN 978-0-12-372538-7.

[2] PARKER, J. Algorithms for image processing and computer vision. New York: Wiley Computer Publishing, 1997, xiii, 417 s. ISBN 04-711-4056-2.

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 29.5.2013

**Vedoucí práce:** Ing. Ondřej Smirg

**Konzultanti diplomové práce:**

**prof. Ing. Kamil Vrba, CSc.**  
*Předseda oborové rady*

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Hlavním úkolem této práce je načítat 3D modely v různých souborových formátech a zobrazovat je, vytvářet řezy z 3D modelů ale i zpětně z řezů vytvářet 3D modely, vyčíslit nepřesnosti po vyhlazení a dalších úpravách a zkoumat podobnost originálních a upravených modelů. To vše v jazyce Java a jeho nástavbách pro práci s 3D objekty.

Práce navazuje na předchozí projekt „3D model z MRI“.

## **KLÍČOVÁ SLOVA**

Java 3D, 3D grafika, podobnost, porovnávání

## **ABSTRACT**

The main aim of the thesis is loading of 3D models in different file formats and its displaying, creating of slices from 3D models and vice versa making 3D models from slices, quantify inaccuracy of anti-aliasing and other formatting and investigate similarity of original and edited models. Everything in Java programming language and its extensions for 3D object tasks.

Thesis builds on earlier project „3D shape from MRI“.

## **KEYWORDS**

Java 3D, 3D graphics, similarity, comparison

ZAPLETAL, T. Porovnávání 3D objektů v jazyce JAVA. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 51 s. Vedoucí diplomové práce Ing. Ondřej Šmirg.

## **PROHLÁŠENÍ**

Prohlašuji, že svou diplomovou práci na téma Porovnávání 3D objektů v jazyce JAVA jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

## **PODĚKOVÁNÍ**

Děkuji vedoucímu diplomové práce Ing. Ondřeji Šmirgovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne .....

.....

(podpis autora)

**Výzkum popsany v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.**

# OBSAH

<b>Seznam obrázků</b>	<b>x</b>
<b>Úvod</b>	<b>1</b>
<b>1 Teoretická část</b>	<b>2</b>
1.1 Nezávislý pohled na 3D svět lidskýma očima.....	2
1.2 Pohled na 3D svět v počítači.....	2
1.2.1 Pojmy ve 3D grafice.....	2
1.2.2 Geometrie polygonální.....	3
1.2.3 Geometrie volná.....	3
1.2.4 Omezení zobrazovacími zařízeními .....	4
1.2.5 Typy renderování.....	4
1.3 3D v medicíně.....	4
<b>2 Formáty vstupních a výstupních dat</b>	<b>6</b>
2.1 Wavefront (.obj).....	6
2.2 STL (.stl).....	6
2.3 Binární řezy (.png).....	6
<b>3 Řešený problém</b>	<b>7</b>
3.1 Vyhlazování.....	7
3.2 Typy vyhlazovacích technik.....	10
3.2.1 Proměnné mřížkování čtyřstěny.....	10
3.2.2 Delaunayova triangulace.....	11
3.2.3 „Úlomky“.....	11
<b>4 Požadavky na platformu</b>	<b>12</b>
4.1 Eclipse.....	12
4.2 Zprovoznění projektu.....	12
4.3 Výkon platformy.....	13
<b>5 Výpočty</b>	<b>14</b>
5.1 Výpočet objemu.....	14

5.2	Vyjádření podobnosti.....	14
<b>6</b>	<b>Aplikace</b>	<b>15</b>
6.1	Grafické rozhraní.....	15
6.2	Interaktivita s uživatelem.....	16
6.2.1	Kurzor myši.....	16
6.2.2	Lišta průběhu.....	16
6.2.3	Sledování průběhu procesu.....	17
6.3	Úpravy předchozího programu.....	17
6.4	Nové algoritmy.....	18
6.4.1	StartCompare.....	19
6.4.2	ImageHandler.....	19
6.4.3	ImgToMat.....	19
6.4.4	Compare.....	19
6.5	Další používané aplikace.....	19
<b>7</b>	<b>Měření vstupních dat</b>	<b>21</b>
7.1	Vlastnosti měření.....	21
7.1.1	Medicínské přístroje.....	21
7.1.2	Skener pro zpětné inženýrství.....	21
7.2	Měření skenerem.....	22
7.3	Výstup programu VXelements.....	24
7.3.1	Zpracování měření softwarem Creaform.....	24
7.3.2	Rozhraní programu VXelements a jeho použití.....	25
7.3.3	Extrakce modelu ze skeneru.....	25
7.4	Úprava v programu Blender.....	28
7.5	Načtení vstupních dat do aplikace.....	29
7.6	Zpracování ve Fiji.....	29
<b>8</b>	<b>Porovnávání</b>	<b>30</b>
8.1	Bitová úroveň.....	30
8.1.1	Objem.....	30
8.1.2	XOR.....	31
8.1.3	MAE.....	31
8.1.4	MSE.....	31
8.1.5	PSNR.....	32



8.2	Vektorová úroveň.....	32
8.3	Obrazové filtry.....	33
<b>9</b>	<b>Závěr</b>	<b>33</b>
	<b>Literatura</b>	<b>35</b>
	<b>Seznam zkratk</b>	<b>38</b>
	<b>Přílohy</b>	<b>39</b>

# SEZNAM OBRÁZKŮ

<b>obr. 1.1 – Příklad rekonstrukce z 2D do 3D</b>	<b>5</b>
<b>obr. 3.2 – vlevo originál, vpravo po vyhlazení</b>	<b>7</b>
<b>obr. 3.3 – porovnání velikosti vyhlazeného objektu s původním</b>	<b>8</b>
<b>obr. 3.4 – příklad nevhodného vyhlazování medicínských dat</b>	<b>9</b>
<b>obr. 3.5 – příklad vhodného vyhlazování medicínských dat</b>	<b>9</b>
<b>obr. 3.6 - mřížkování pomocí čtyřstěnů</b>	<b>10</b>
<b>obr. 3.7 – Delaunayova triangulace zobrazená v ploše</b>	<b>11</b>
<b>obr. 3.8 - úlomky, které nevhodně vyplní kouli</b>	<b>11</b>
<b>obr. 6.9 – Grafické rozhraní aplikace</b>	<b>15</b>
<b>obr. 6.10 – upravené načítání vstupů .obj pro porovnání</b>	<b>16</b>
<b>obr. 6.11 – neurčitý stav      obr. 6.12 – určitý stav</b>	<b>17</b>
<b>obr. 6.11 – neurčitý stav      obr. 6.12 – určitý stav</b>	<b>17</b>
<b>obr. 6.13 – okno monitorování průběhu</b>	<b>17</b>
<b>obr. 6.14 – UML diagram nových tříd</b>	<b>18</b>
<b>obr. 6.15 - uživatelské rozhraní aplikace Fiji</b>	<b>20</b>
<b>obr. 7.16– 3D skener VUT EXAscan</b>	<b>22</b>
<b>obr. 7.17 – měřený objekt – hruška</b>	<b>24</b>
<b>obr. 7.18 – horní část objektu v prostoru s markery</b>	<b>26</b>
<b>obr. 7.19 – nastavení vlastního souřadnicového systému</b>	<b>27</b>
<b>obr. 7.20 – sloučení obou projektů importem</b>	<b>27</b>
<b>obr. 7.21 – úpravy obou objektů, jejich spojení a odstranění duplicitních bodů</b>	<b>28</b>
<b>obr. 7.22 – finální model, vyrovnáno v osách</b>	<b>28</b>

<b>obr. 7.23 – originál, podvzorkovaný a vyhlazený model</b>	<b>29</b>
<b>obr. 8.24 – kroky XOR – originál, vyhlazený, rozdíl (XOR)</b>	<b>31</b>
<b>obr. 9.25 – výsledek porovnání dvou 3D modelů</b>	<b>34</b>

# ÚVOD

Diplomová práce je zaměřena na návrh aplikace, která načítá a zobrazuje 3D objekty z různých souborových formátů, ale také z řezů – dvourozměrných snímků získaných z magnetické rezonance. Snímky jsou pro aplikaci již předpřipraveny segmentací obrazu jako binární maska. Z načtených 3D modelů je aplikace schopna opět vytvořit nové řezy a uložit je jako sadu po sobě jdoucích snímků.

Snímky získané magnetickou rezonancí je před dalším zpracováním doktory potřeba upravit. Tyto úpravy nejsou bezetrátové a snižují přesnost dalších požadovaných zpracování. Programy upravují 3D modely pro lepší zobrazení, což způsobuje odchylky od skutečného tvaru objektu.

Aplikace porovnává objekty z hlediska podobnosti před a po úpravách, jako je například vyhlazování. Bylo zjištěno, že tyto úpravy vedou k deformaci původního modelu. Cílem práce je toto tvrzení potvrdit nebo vyvrátit a zjistit, jaké metody v jak velkém měřítku degradují originální objekt. Jinými slovy kvantifikovat odchylky upravených objektů proti původním. Jako základní kámen byla upravena aplikace z diplomové práce „3D model z MRI“ 9.

První část diplomové práce je koncipována jako více teoretická, protože je třeba objasnit vše okolo 3D světa a používaných technik v Javě. Terminologie v tomto oboru má mnoho zkratk, které jsou v textu použity. Jejich význam je na konci práce v seznamu zkratk.

V teoretické části je popsán svět ve 3D, Java rozhraní, objasněn základní problém, používané algoritmy a jejich klady a zápory. Jsou zde vysvětleny termíny potřebné k pochopení 3D světa nejen v Javě.

Praktická část se zabývá vytvořením dat pro aplikaci, úpravou 3D modelů a implementace aplikace načítající 3D modely. Dále popisuje algoritmy a principy vytvořené aplikace a vývojové diagramy. Stěžejní část je potom jejich vzájemné porovnávání a určení důležitosti jednotlivých parametrů vypočtených při porovnání.

# 1 TEORETICKÁ ČÁST

V následujícím textu je rozebrán svět ve třech dimenzích z pohledu programátora a aplikací – tedy člověka a stroje. Tato část práce popisuje svět tak, jak jej vidíme a rozdíl oproti formálnímu zápisu v programech.

## 1.1 Nezávislý pohled na 3D svět lidskýma očima

V této kapitole je přiblížena biologická podstata vnímání 3D reality člověkem. Také je zde lehká konfrontace se současnou technologií.

Obrazový vjem získáváme pomocí očí, které jsou poměrně primitivním zařízením. Díky tomu, že tyto fotoreceptory má člověk dva, dokáže mozek sestavit trojrozměrný obraz. Největší sílu v chápání trojrozměrného obrazu dodává právě mozek. Více informací na 9 9. Výpočetní výkon mozku a vůbec zpracování vstupních signálů z očí se stále zdá být poměrně vzdálené pro naše současná zařízení. Více informací na 9.

## 1.2 Pohled na 3D svět v počítači

Vývoj 3D grafiky na počítačích byl dalším krokem v grafice vůbec. V roce 1992 vznikla první specifikace OpenGL. Více informací na 9. Velký vliv na vývoj grafiky mají počítačové hry. Průkopníky ve vývoji byly specifikace OpenGL a Microsoft DirectX.

Jak se 3D grafika stávala žádanější, přibývalo firem zabývajících se vývojem her tvořících vlastní herní engine<sup>1</sup>. Více informací na 9. Z modernějších stojí za zmínku ty otevřené jako Unreal, Quake nebo CryEngine. SDK jsou pro nekomerční užití zdarma.

Vedle počítačových her nelze opomenout filmový průmysl, kde se nejen hojně využívá digitálních efektů, ale tvoří se celé virtuální světy. Software filmových studií je minimálně o třídu výše s kvalitou zobrazení, aby uchvátil diváka, ale již nelze vytvářet obraz v reálném čase, protože na to chybí potřebný výkon.

### 1.2.1 Pojmy ve 3D grafice

Pro vyjádření a orientaci v prostoru se používá kartézská soustava souřadnic, která popisuje každý bod v prostoru pomocí tří hodnot  $x$ ,  $y$ ,  $z$ . Ty udávají vzdálenosti bodu od počátku na jednotlivých vzájemně kolmých osách. Množina bodů v prostoru může tvořit celé objekty.

---

<sup>1</sup> Jádro aplikace, tvořící rozhraní mezi programátorem a systémovými záležitostmi. Obsahuje např. 2D/3D grafiku, fyziku, zvuk, skripty, umělou inteligenci, správu paměti atd.

Je vhodné zde uvést pojmy, užívané v oboru 3D grafiky:

- Face – polygon o třech vertexech. Jedná se o nejjednodušší plochu v prostoru.
- Izokontura ohraničuje segment, který obsahuje body s konstantní hodnotou.
- Izoplocha – plocha, kterou reprezentují body s konstantní hodnotou
- Kamera – jedná se o bod, ze kterého je viditelná scéna. Je zadána vertexem a třemi úhly.
- Objekt – základní stavební prvek v 3D grafice.
- Plane – plocha definovaná vzdáleností od počátku soustavy souřadnic a postavením v prostoru.
- Polygon – Též mnohoúhelník, je část roviny, ohraničená uzavřenou lomenou čarou. Má nejméně 3 vrcholy. Z polygonů se skládají objekty.
- Renderování – přepočítání 3D obrazu na 2D pro 2D zobrazovací zařízení
- Scéna – poskytuje kompletní informace o tom, co a jak bude zobrazováno. Obsahuje 3D objekty, definici světla, nastavení kamery a mnohé další informace.
- Vertex – vrchol definován třemi souřadnicemi  $x, y, z$ .
- Voxel – částice objemu představující hodnotu v pravidelné mřížce 3D prostoru počítačové grafiky

## 1.2.2 Geometrie polygonální

Dříve nejčastěji využívaná geometrie byla založena na principu polygonů. Dle teorie se dá libovolný předmět vyjádřit polygony. Ty vytvoří síť, která je povrchem daného objektu. Druhá věc je, kolik takových polygonů je třeba pro věrnou reprodukci daného objektu. Kvalita zobrazovaného tvaru objektu nejde nahoru přímo úměrně s rostoucím počtem polygonů. Pro dokonalé zobrazení křivky je zapotřebí množství polygonů blízké nekonečnu. Toho samozřejmě nelze dosáhnout na konvenčních strojích, a tak se nedokonalosti často dohánějí vyhlazováním. Více informací na 9.

## 1.2.3 Geometrie volná

Též free-form geometrie již používá křivky. Tvary jsou tvořeny hladkými čarami, které se vykreslují podle rovnic. Křivky jsou reprezentovány kontrolními a řídicími body, kterými křivka prochází. Dnes se křivky generují nejčastěji pomocí matematického modelu NURBS. Vzniklé plochy jsou oblé, pokud je to požadováno. Takto vytvořené objekty potřebují ke svému vzniku mnohem méně informací, než potřebuje polygonální modelování. Tím se ušetří i na velikosti souboru. Je to podobné, jako srovnávat vektorovou a rastrovou grafiku. Více informací na 9.

### 1.2.4 Omezení zobrazovacími zařízeními

Na běžném plochém monitoru se 3D zobrazování omezuje na jeden úhel kamery, nebo přesněji jednu pozici pozorovatele. Výpočetní jednotka obvykle počítá objekty jako celky a poté přizpůsobí zobrazení světelným podmínkám a spočítá pouze viditelné plochy k zobrazení. V úvahu se berou možné průhlednosti, lesklost povrchu, úhel dopadu světla a jeho intenzita. Tento převod se nazývá renderování.

Stereoskopické<sup>2</sup> vidění se dá napodobit např. softwarově, čehož se dá využívat opět ve videohrách. Na zobrazovací zařízení se umístí kamera, která snímá pozorovatele. Podle polohy jeho hlavy pak přizpůsobuje obraz tak, aby měl pocit, že skutečně vidí i za předměty na displeji. Více informací na 9.

Pro zobrazení třetího rozměru (hloubky) se také používají barevné filtry, např. anaglyfické<sup>3</sup> brýle. Dvoubarevné brýle filtrovaly obraz pro každé oko. Dnes existují vyspělejší techniky zobrazování, jako použití polarizačních brýlí a 100Hertzové HD televize, nebo promítačka v 3D kině. Jedno z nejkvalitnějších 3D kin v České republice vlastní renomovaná kanadská společnost IMAX.

### 1.2.5 Typy renderování

Pro renderování se používají různé techniky, jako např. pravouhlé, kosoúhlé či axiometrické. Více informací na 9. Některé z nich jsou zde pro přehled uvedeny:

- Multiplanární renderování (MPR)
- Objemové renderování (VR)
- Povrchové renderování (SR)

## 1.3 3D v medicíně

Aby mohli lékaři přesněji identifikovat procesy uvnitř lidského těla, používají různé techniky jako je počítačová tomografie (CT), magnetická rezonance (MRI) či ultrazvuk. Každá technika funguje na odlišném principu a i snímky získané touto metodou se liší.

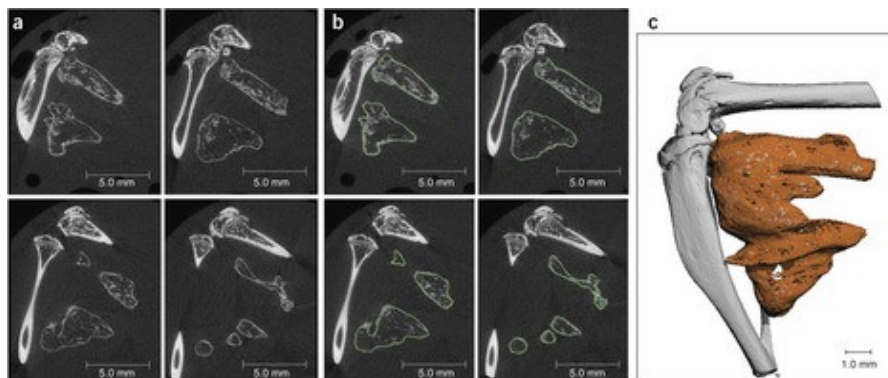
V odvětví medicíny také pokročily zobrazovací techniky. Pro lepší reprodukci obrazu je vhodné z většího množství 2D snímků rekonstruovat 3D obraz. Více informací na 9 9. Tyto snímky se získávají v řezech, které jsou poskládány za sebou.

V medicíně se často používá standard DICOM – Digital Imaging and Communications in Medicine. Tento standard obsahuje definice souborových formátů, komunikačních protokolů, ukládání, tisk a přenos informací v medicínském zpracování obrazu. Integruje tak skenovací zařízení, servery, stanice, tiskárny a další hardware od

<sup>2</sup> Technologie, umožňující prostorový zrakový vjem podle dvourozměrné předlohy

<sup>3</sup> Brýle v s jedním sklem červeným a druhým modrozeleným. Existují i jiné barevné kombinace.

různých výrobců, podporující tento standard. Více informací o standardu na 9.



obr. 1.1 – Příklad rekonstrukce z 2D do 3D

Jako ukázka je zde na obr. 1 .1 rekonstrukce kostní tkáně na základě 2D řezů.

Popis obrázku:

- a) 2D řezy vzorku, zobrazující uskupení přiléhajících kostí.
- b) Obrysy ektopické<sup>4</sup> kostní tkáně, zobrazeny zeleně.
- c) 3D obraz vzorku po 3D rekonstrukci.

Zdroj 9.

---

<sup>4</sup> Ektopický – uložení mimo obvyklé místo



## 2 FORMÁTY VSTUPNÍCH A VÝSTUPNÍCH DAT

Dnes si většina společností pro svůj software vytváří vlastní souborový formát. Aplikace, o které bude více popsáno v kapitole 6, dokáže načítat dva typy souborů, a to Wafefront (.obj) a STL (.stl). Formáty jsou rozebrány velice stručně, pouze pro přehled. Podrobnější popis lze nalézt v 9.

### 2.1 Wafefront (.obj)

Vyvinutý firmou Wavefront Technologies, pro jejich Advanced Visualizer. Existují varianty ASCII (.obj) a binární (.mod). Soubory obsahují několik návěstí, za nimi následuje seznam polygonů. K souboru .obj bývá přidružen další soubor s příponou .mtl, obsahující definice materiálů a barev pro daný objekt. Více informací v 999.

### 2.2 STL (.stl)

Vyvinut firmou 3D Systems, určen k návrhu průmyslových vzorů, prototypů a jejich realizaci na 3D tiskárnách. Díky své jednoduchosti je používán v mnoha CAD aplikacích. STL popisuje pouze povrchovou geometrii 3D objektů bez definice barev, textur či dalších atributů. Jedno ze základních pravidel tohoto formátu je, že každý trojúhelník musí sdílet dva své vrcholy s dvěma sousedními trojúhelníky. Opět existují varianty ASCII a binární. ASCII varianta je popisnější, takže zabírá více prostoru, a proto se tolik nepoužívá.

### 2.3 Binární řezy (.png)

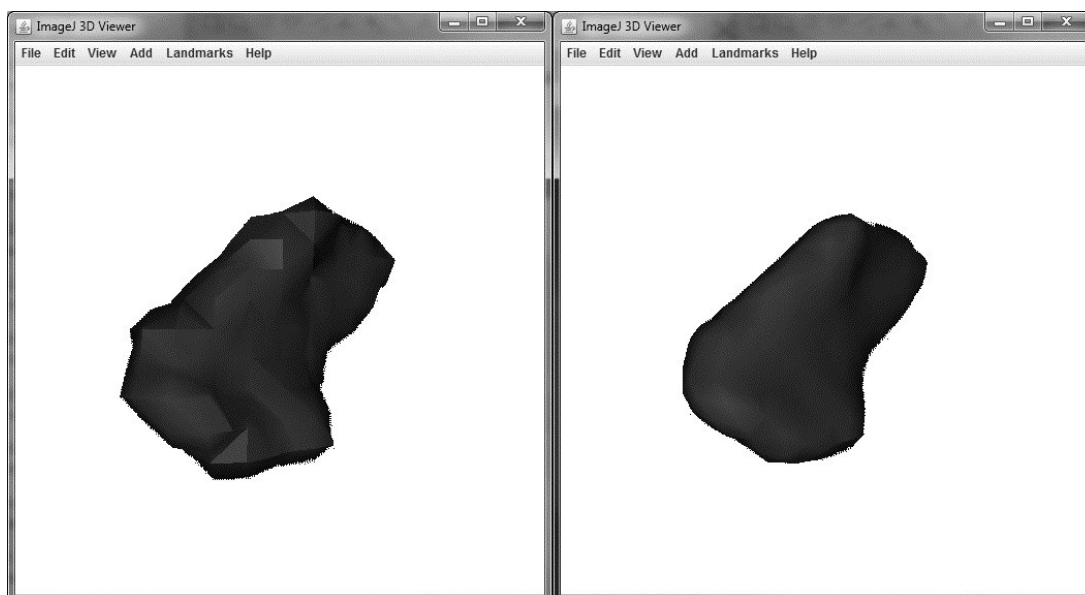
Formát .png je úsporným řešením pro rastrovou grafiku. Využívá bezeztrátovou kompresi, což je pro medicínské účely nezbytné. Je postaven na principu formátu .gif, s lepšími parametry, podporou průhledností, ale už se neujal pro animace.

## 3 ŘEŠENÝ PROBLÉM

Tato kapitola popisuje, co je nutné v dané problematice vyřešit. Používané metody mají své klady i zápory. Pro omezení negativ je nutné provést další kroky.

### 3.1 Vyhlazování

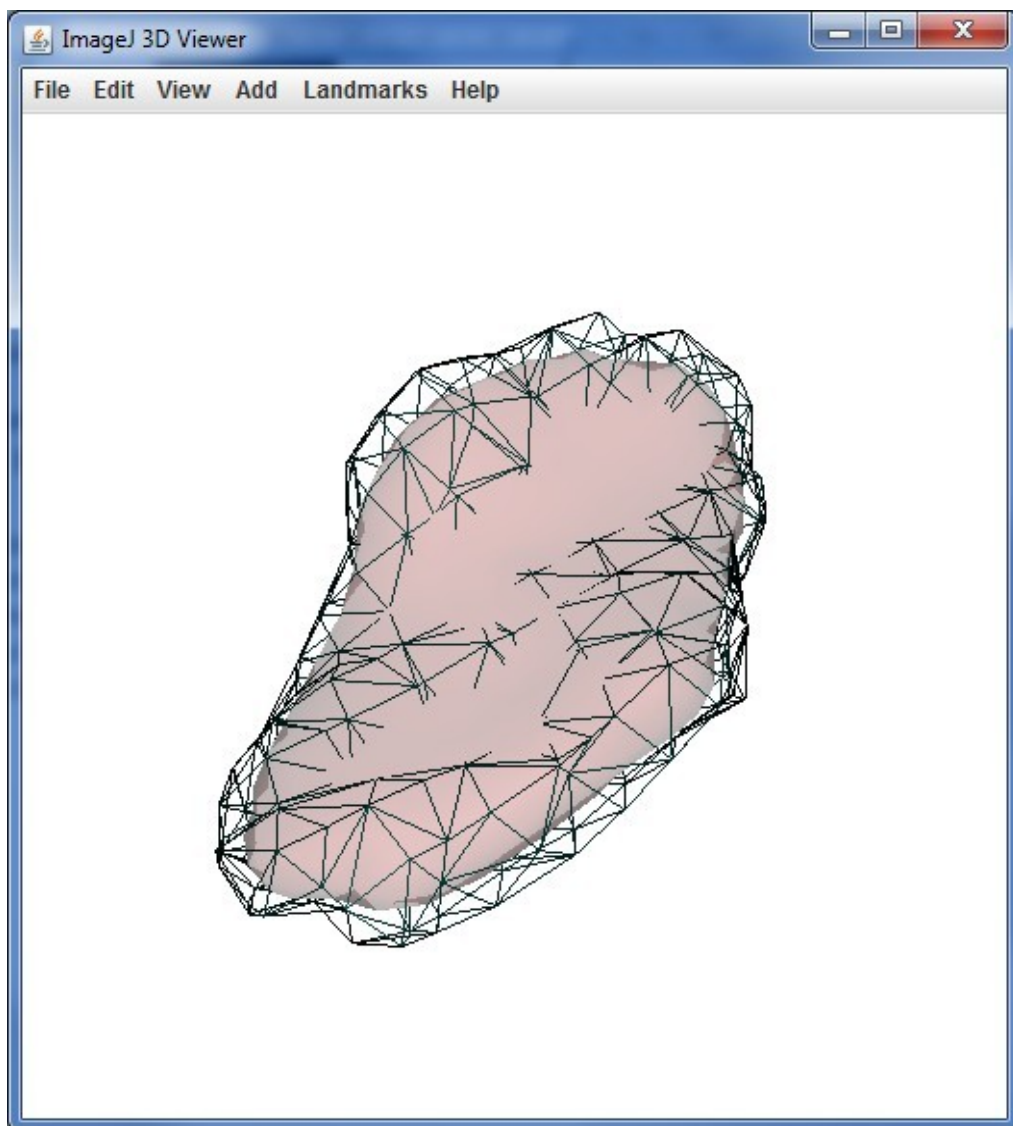
Pokud není k dispozici dostatečné množství přesných dat, nebo tato data nejsme schopni přesně zobrazit, objevují se nehezké hrany. Vyhlazování má za úkol přiblížit obraz realitě. Vyhlazování se dá rozdělit jako celoobrazovkové (FSAA) a vyhlazování modelů. Tato práce se zabývá vyhlazováním přímo objektů samotných.



obr. 3.2 – vlevo originál, vpravo po vyhlazení

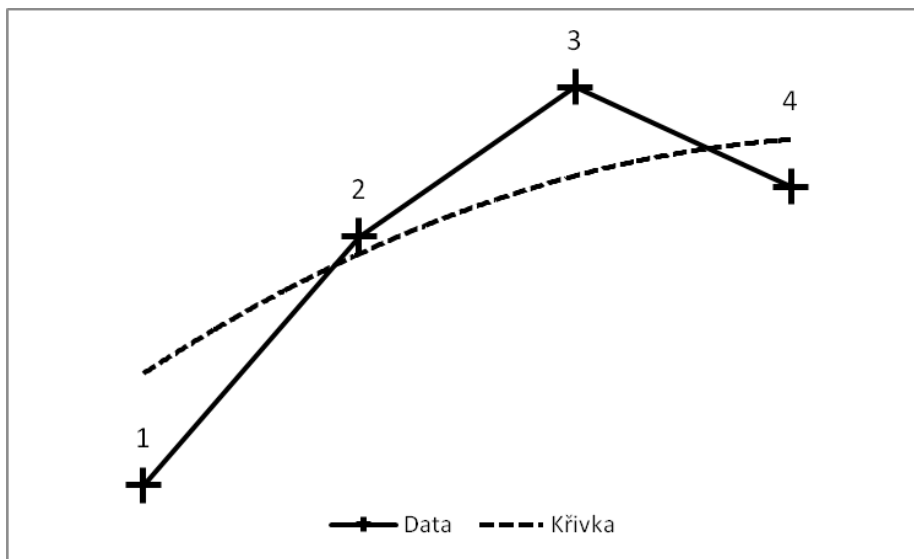
Na originálním snímku obr. 3.2 jsou patrné hrany polygonů. Vedle je tentýž objekt po úpravě vyhlazením. Ten vypadá na první pohled lépe, ale vytrácí se přesnost původního předmětu.

Algoritmus vyhlazování neposuzuje, které body jsou na dané pozici se stoprocentní jistotou a které jsou dopočítány. Bere objekt jako celek a vyhlazuje hrany. S přibývajícím počtem vyhlazovacích iterací se navíc redukuje objem objektu. Pro medicínské využití je toto nepřijatelné. V případě kloubních náhrad a jiných implantovaných částí je tolerance nežádoucí.

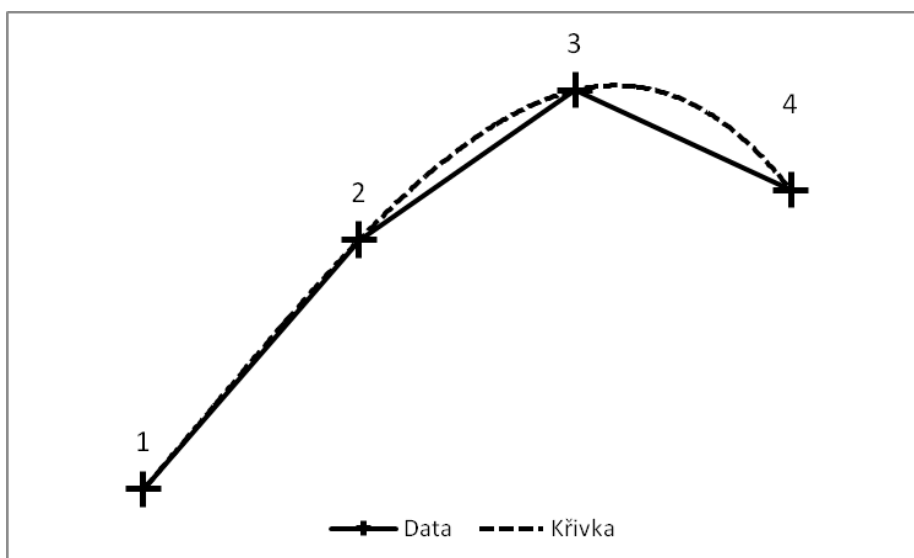


obr. 3.3 – porovnání velikosti vyhlazeného objektu s původním

Na obr. 3 .3 je porovnání, kde z původního objektu je zobrazen pouze obrys a uvnitř něj je objekt vyhlazený. Síť polygonů obklopuje téměř celý původní objekt. Pouze v místech, kde se vyhlazením objem zvětšil, je síť utopená.



obr. 3.4 – příklad nevhodného vyhlazování medicínských dat



obr. 3.5 – příklad vhodného vyhlazování medicínských dat

Na obr. 3.4 a obr. 3.5 je příklad fungování některých vyhlazovacích algoritmů. Body 1–4 představují naskenované pozice např. pro lidskou kost. Vyhlazovací algoritmus tyto body proloží křivkou.

V prvním případě prokládá body polynomem druhého řádu a neprochází body, o kterých se dá s největší jistotou říci, že se skutečně nacházejí na skenovaném předmětu. Jisté tedy je, že výsledné zobrazení nebude odpovídat předloze. Chybějící data jsou aproximována. Nová křivka neprochází změřenými body.

V druhém případě již křivka prochází všemi body. Nelze určit, kudy přesně má vést spojnice bodů, ale tyto body musí být podmnožinou této spojnice. Křivka je

interpolována. Chybějící data se hledají v intervalu známých hodnot tak, aby procházela vždy změřenými body. Více informací na 9

## 3.2 Typy vyhlazovacích technik

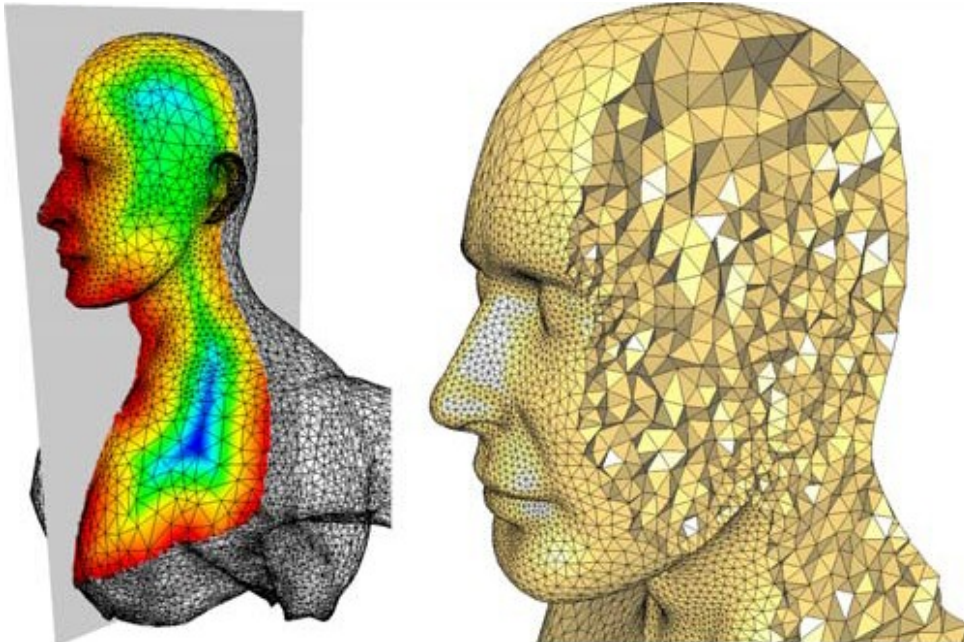
Techniky zobrazení trojrozměrných objektů pracují lépe, pokud je objekt přizpůsobený výpočtem již při jeho vzniku k co nejuvhodnější rekonstrukci jeho vzoru. Pokud tomu tak není, přijdou na řadu techniky vyhlazovací. Pracují na různých principech, ale obecně se snaží uspořádat data tak, aby výsledek co nejvíce odpovídal předloze. Následující techniky jsou úpravou modelu přepočítáním všech dat tak, aby byl výsledek lépe reprezentován a zároveň odpovídal předloze.

### 3.2.1 Proměnné mřížkování čtyřstěny

(Variational Tetrahedral Meshing)

3D generace mřížek používá čtyřstěny v takovém rozložení, že jsou buď disjunktní, nebo mají společný jeden face. Tato diskretizace prostoru je nutná pro většinu fyzikálně založených simulačních technik. Realistická simulace deformovatelných objektů v počítačové grafice potřebuje diskrétní doménu pro výpočty metod konečných prvků a konečného objemu (finite-element 9, finite-volume 9). Běžně jsou požadovány čtyřstěny s přibližně stejně velkými hranami (isotropic meshing 9).

Problém je, že z trojúhelníků se sice dá v 2D poskládat přesná plocha, ale z čtyřstěnu se nedá poskládat přesný prostor ve 3D. I přesto jsou hrany výsledného objektu oblejší.

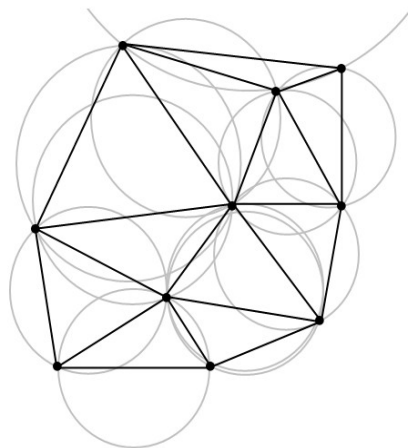


obr. 3.6 - mřížkování pomocí čtyřstěnu

Na obr. 3.6 je patrné, že zrna jsou hrubší tam, kde je malé zakřivení a jemnější v místech s většími detaily. Zdroj 9.

### 3.2.2 Delaunayova triangulace

Podle ruského matematika je pojmenována metoda, která tvoří kružnice procházející body tak, aby se žádný bod nenacházel uvnitř jiné kružnice.



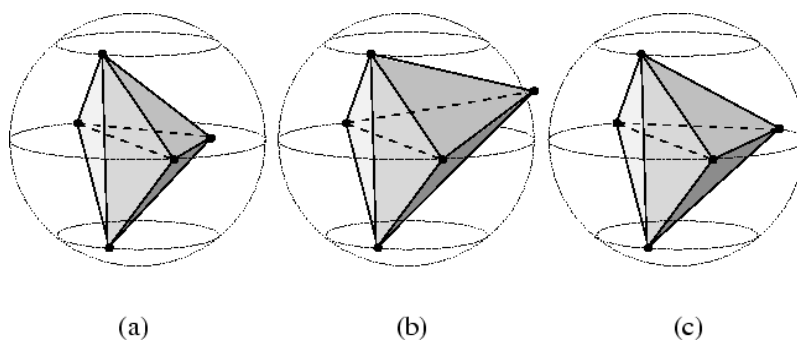
obr. 3.7 – Delaunayova triangulace zobrazená v ploše

Navíc se snaží minimalizovat maximální poloměr uzavírající koule. Co se stane, když tento cíl není splněn, popisuje další podkapitola. Metodu lze tedy použít i v prostoru, např. ke zjemnění výsledku mřížkování (Delaunay refinement 9).

### 3.2.3 „Úlomky“

(Slivers)

Nebo též třísky či šupiny, jsou čtyřstěny, které narušují pravidelnost struktury tím, že jejich hrany nejsou stejně velké – nejsou izotropické. Tento prvek je specifický svým nevhodným tvarem pro tvoření mřížek. Snahou je tyto úlomky eliminovat, kvůli garanci kvality modelu. 9



obr. 3.8 - úlomky, které nevhodně vyplní kouli

## 4 POŽADAVKY NA PLATFORMU

V této kapitole je rozebrána problematika vývojového prostředí a rozhraní pro tvorbu z hlediska vývoje. Je zde popsán průběh zprovoznění vývojového prostředí Eclipse na platformě Microsoft Windows 7 64bit a stručný popis prostředí samotného. Kapitola začíná stručným popisem vývojového prostředí

### 4.1 Eclipse

Vývojové prostředí Eclipse je software s otevřeným zdrojovým kódem, na kterém spolupracuje kolem tisícovky vývojářů, a užívají jej tisíce firem. Většina uživatelů jej používá na platformě Windows, což vede vývoj k maximalizování kompatibility a výkonu pro tento operační systém. Více informací na 9.

Eclipse je modulární vývojové prostředí, takže se dá rozšiřovat různými balíčky, knihovny a zásuvnými moduly. Doinstalovat se dá například podpora SVN, což umožňuje sdílení projektů a jejich zdrojových kódů, s označením data poslední změny, takže se dá spolupracovat s celým týmem programátorů. Více informací na 9 .

### 4.2 Zprovoznění projektu

Ke spuštění projektu v Eclipse je třeba mít nainstalován JVM, který je obsažen v JRE. JRE je součástí JDK, bez kterého se systém obejde, ale některé externí Javovské aplikace jej potřebují ke kompilování kódu. Rozdíly mezi JRE a JDK jsou popsány v literatuře 9. Dále se v 3D programování Eclipse neobejde bez Java 3D API.

Pro 64bitový systém jsou pro správnou funkčnost potřeba verze 64b, ale i 32b. Kroky potřebné ke správnému chování a možnosti pokračovat v projektu jsou tyto:

- IDE, např. Eclipse
- JRE 32
- JRE 64
- Java 3D API 32
- Java 3D API 64
- Soubor j3dcore-ogl.dll ve složce Windows
- Zkontrolovat Build path<sup>5</sup>, jestli jsou nalinkovány knihovny

---

<sup>5</sup> Java při kompilaci kontroluje, jestli jsou dostupné všechny náležitosti, jako např. soubory typu JAR, obsahující potřebné knihovny. Umístění těchto externích souborů je určeno právě v Build path.

### 4.3 Výkon platformy

Doktoři nevyžadují provoz v reálném čase v pravém slova smyslu. Nyní jsou zvyklí čekat na správnou úpravu obrazů a objektů i několik dní. Pokud bude výpočet trvat řádově několik minut, je to pro doktory dostatečné pro použití v reálném čase za provozu. Pacient po proměření odchází ke specialistovi do ordinace nebo počká v čekárně, zatímco se výsledky připravují. Již nemusí chodit znova jiný den pro výsledek.

Takový výkon může mít aplikace na platformě Java. V dobách kolem roku 2003 platilo, že běh aplikace napsané v Javě je výrazně pomalejší, než by to bylo u klasické nativní aplikace. To bylo způsobeno právě tím, že byte-code se musí interpretovat. Toto zpomalení bylo poměrně velmi znatelné – některé zdroje mluvily o 20 až 40 krát rychlejším běhu stejného programu, ale napsaném v C. Tohoto si ovšem firma Sun všimla a vyvinula tzv. *Just In Time Compiler* (JIT), který urychluje běh aplikace tak, že mezikód překládá do strojového kódu procesoru. Tato technologie sice opravdu razantně urychluje běh aplikace (kromě drobného zpoždění při startu, kdy se část překládá, aplikace spouštěná pod verzí 1.3 běží zhruba 6x rychleji, ve verzi 1.4 je to asi 12x rychleji), zato klade větší nároky na paměť. 9



## 5 VÝPOČTY

Kapitola 5 popisuje základní myšlenku možností porovnávání. Jeden z hlavních problémů je změna objemu upraveného 3D modelu. Podrobněji bude postup a metodika výpočtů popsána v dalších kapitolách.

### 5.1 Výpočet objemu

Existují externí programy, ale metoda nebo samotný kód pro Javu se v současné době nijak veřejně nešíří. Řešení této specifické potřeby si programátoři rádi nechají zaplatit. V této práci bude navržena vlastní metoda.

Vertexy každého objektu tvoří jeho povrch. Navržený postup zjistí minima a maxima na každé ze tří os. Tím se vytvoří kvádr, který uzavře celý objekt ve svém nitru. Poté se udělá v některé ose tolik řezů, kolik pixelů se v této ose zobrazuje do hloubky. Získané řezy jsou ukládány v binárních maskách a nemělo by být potřeba je ukládat do souboru. Stačí vždy udržet v paměti právě zpracovávaný řez. V každém řezu se poté sečte počet bodů uvnitř objektů. Součet všech vnitřních bodů ve všech řezech by měl být výsledný objem objektu, kde jednotkou bude pixel ve třetí mocnině. Pro porovnání upraveného objektu s originálem postačí hodnota v procentech původního objemu.

Podle náročnosti výpočtu se dá uvažovat o zjišťování, zda je výhodnější počítat osu, kde bude méně řezů s velkou plochou nebo hodně řezů o malé ploše. Rozdíl se může projevit jak v paměťové tak ve výpočetní náročnosti. Pro ještě lepší výkon by se každý řez mohl zpracovávat v novém vlákně, což by vedlo k nárůstu výkonu obzvláště na vícejádrových strojích nebo ve výpočetních gridech<sup>6</sup>.

Implementace tohoto algoritmu vyžaduje dobré porozumění souborových formátů pro 3D objekty. Prvními typy připadajícími v úvahu budou soubory s příponami .stl a .obj. Ty byly použity v diplomové práci, na kterou tato práce navazuje.

### 5.2 Vyjádření podobnosti

Jedná se o netriviální výpočet, který musí brát v potaz mnoho faktorů. Jednak hraje roli již zmiňovaný objem, dále jsou to tvary. Křivky mohou být konvexní či konkávní a to v různé míře. Každá odlišnost navíc musí být nějak ohodnocena. Čistý pixelový rozdíl bude také užitečná hodnota, ale nebude nejvhodnějším měřítkem pro porovnávání upravených objektů. Více o podobnosti v kapitole Měření vstupních dat.

---

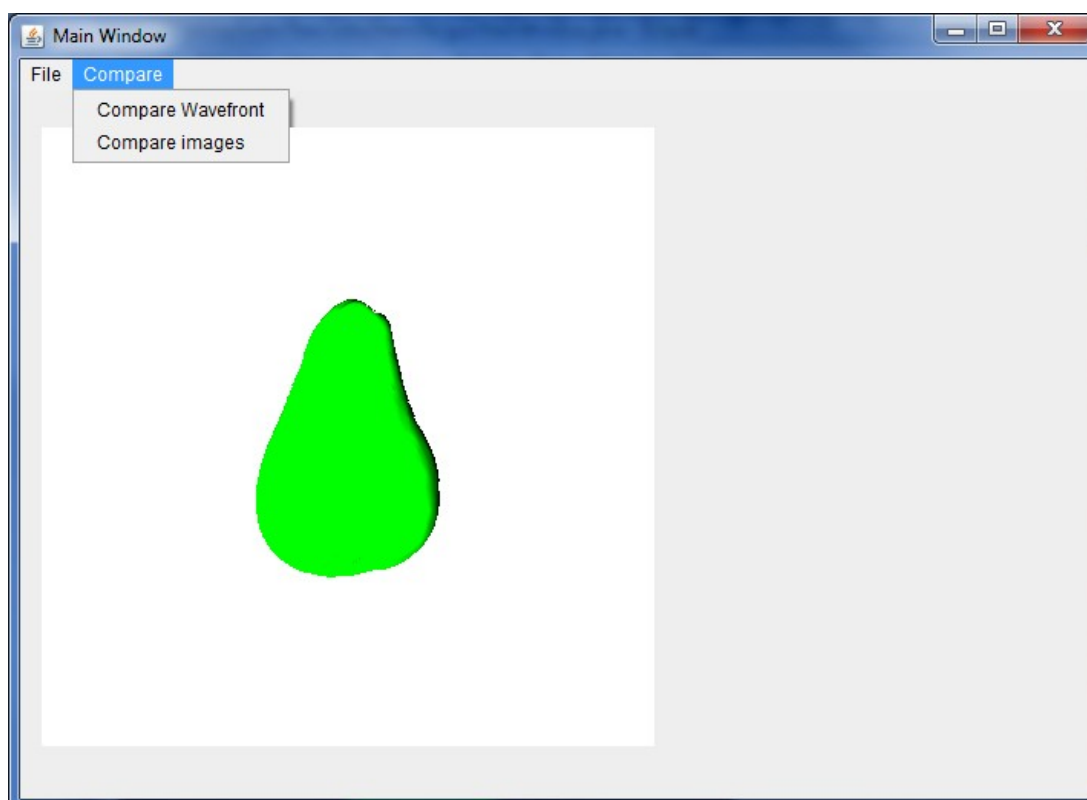
<sup>6</sup> Grid je větší počet počítačů, pracujících společně na nějakém cíli. Rozdíl mezi Clusterem a Gridem je dostupný z 9

## 6 APLIKACE

Popis aplikace samotné, její ovládání a přidaná rozšíření. Implementace nových metod je rozepsána tak, jak postupně vznikaly. Základem je aplikace vytvořená v diplomové práci 3D Model z MRI. 9

### 6.1 Grafické rozhraní

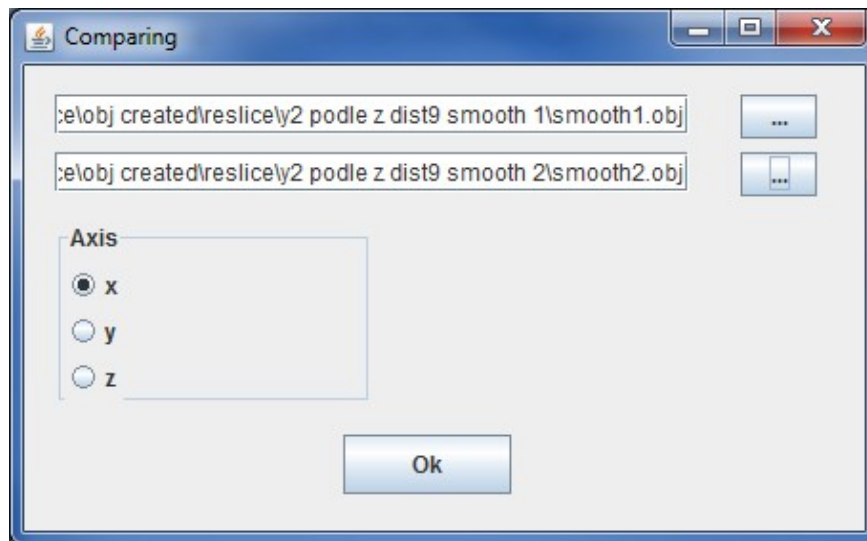
Aplikace v Javě jdou vstříc uživatelům a jsou uživatelsky přívětivé. Vzhled okna se podobá běžně používaným aplikacím. Obsahuje menu a plochu pro zobrazení otevíraných objektů.



obr. 6.9 – Grafické rozhraní aplikace

Podmenu *Import* je určeno k otevírání 3D modelů ve dvou podporovaných formátech, a to .stl a .obj (wavefront). Položka *Create 3D shape* vytvoří 3D model z 2D řezů. Položka *Reslice* vytvoří 2D řezy z načteného 3D modelu.

Oproti základnímu programu, z kterého práce vychází je přidáno menu „Compare“ obsahující položky porovnání 3D modelů typu Wavefront (.obj) nebo pro jednodušší kontrolu, pokud už jsou dostupné řezy samotné, porovnávání řezů. Dialog menu Compare wavefront je na obr. 6.10.



obr. 6.10 – upravené načítání vstupů .obj pro porovnání

## 6.2 Interaktivita s uživatelem

U aplikací s dlouhou dobou odezvy je nutné informovat uživatele, že se něco děje a že program stále pracuje. Navržená aplikace potřebuje pro výpočty řádově minuty a uživatel nemá možnost ověřit správnou funkčnost aplikace. Proto byly implementovány podpůrné prostředky, napomáhající orientaci o stavu programu.

### 6.2.1 Kurzor myši

Při importování 3D modelů a jejich otevíráním a následným zobrazením aplikací byla přidána vlastnost změny kurzoru myši do stavu čekání. Ten se podle použitého operačního systému prezentuje hodinami nebo animovaným kroužkem. Po vykonání dané aktivity se opět vrátí do normálního stavu.

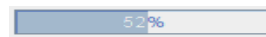
### 6.2.2 Lišta průběhu

Anglicky též progress bar zobrazuje stav a průběh činnosti při otevírání 3D modelu a porovnávání obrazů. U metod, které nejdou zcela atomizovat, je obtížné zobrazit aktuální stav. To je například načítání 3D modelu do plochy aplikace. Java si většinu procesů ohledně zobrazení scény a objektu řídí sama, není tedy transparentní průběh načítání dat. K tomu je vhodné použít progress bar s neurčitým stavem průběhu. Ten značí uživateli, že se stále něco děje.

Pokud lze zjistit stav průběhu, např. kolik dat již bylo porovnáno, je použit progress bar s odhadem aktuálního stavu zobrazující procenta.



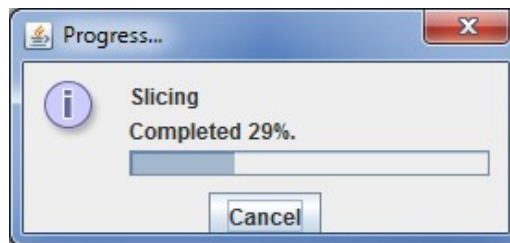
obr. 6.11 – neurčitý stav



obr. 6.12 – určitý stav

### 6.2.3 Sledování průběhu procesu

U časově velmi náročných procesů, jako je extrakce dvourozměrných řezů z 3D modelu, je vhodné použití okna monitorování průběhu, anglicky progress monitor. Ten uživatele informuje o stavu procesu a dává mu možnost zrušit neodpovídající úkol. Tento monitor je použit při extrakci dvourozměrných obrazů z 3D modelu a při porovnávání dvou modelů ve formátu wavefront (.obj).



obr. 6.13 – okno monitorování průběhu

## 6.3 Úpravy předchozího programu

Při implementaci nových metod bylo nutné modifikovat program z předchozí diplomové práce, na který tato práce navazuje. Nejvíce změn stávajícího kódu vyžadovala třída hlavního okna `MainWindow` a třída tvorby dvourozměrných snímků `Reslice` společně s `ResliceSetting`.

Do hlavního okna přibyla textová oblast pro zobrazování výsledků porovnávání a nové položky menu. Výsledky lze zobrazovat pomocí komponenty `Label`, ale pro praktičnost byla zvolena komponenta `TextArea`, z které lze výstup označit a kopírovat k dalšímu zpracování. Pro snadnější výpisy aplikace byla implementována metoda `redirectSystemStreams()`, která bohužel omezuje funkčnost zobrazení 3D modelu v `JPanelu`, a tak byla deaktivována. Dále byla přidána v kapitole 6.2.2 zmiňovaná lišta průběhu, která se zobrazuje při načítání a úspěšném otevření modelů.

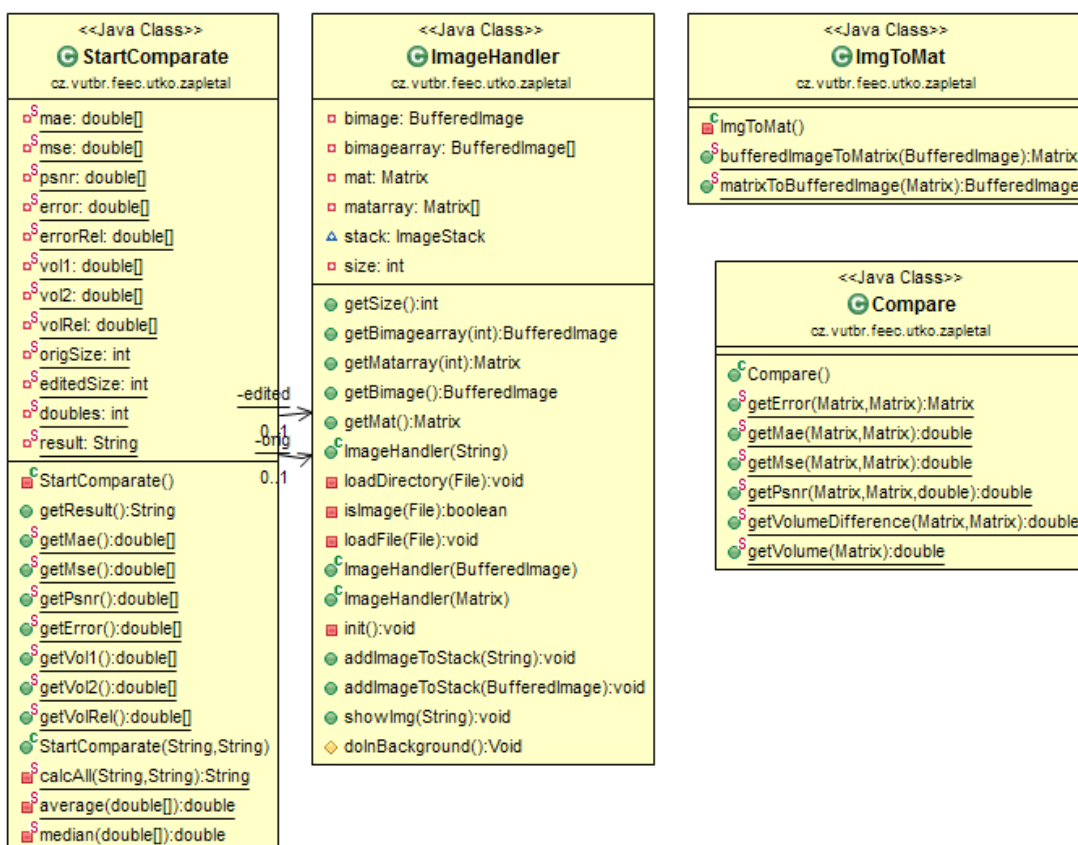
Pro vstupní dialog porovnávání modelů byla upravena třída `ResliceSetting`, do které bylo přidáno rozpoznávání volající funkce v režimu porovnávání nebo v klasickém vytváření řezů. Při volání třídy `ResliceSetting` je přidáván parametr „comparing“ typu `Boolean`, který označuje požadovaný režim. Sadou podmínek je poté upraveno dialogové okno, dotazující se na vstupní data. Přibyl další řádek pro vstupní soubor 3D modelu, načítající data zvlášť pro originální a pro upravený model.

Byla zrušena možnost uživatelem volený počet řezů v režimu porovnávání, místo toho program zjistí velikost modelů a nastaví si počet řezů podle ní. V režimu tvorby řezů bylo odstraněno omezení dělat maximálně 15 řezů. Výstupní řezy jsou nyní ukládány do složek, pojmenovaných dle načteného modelu. Tím se zabrání přepisování výstupů při porovnávání dvou modelů umístěných v jedné složce.

Po potvrzení vstupního nastavení se spustí monitor průběhu, který je popsán v kapitole 6.2.3. Pro jeho správnou inicializaci a chod bylo nutné přidat dědičnost z třídy `SwingWorker`, která se stará o zpracování vláken v pozadí a sledování jejich stavu. Volání takové třídy se spouští příkazem `execute`, který spustí metody obsažené v metodě `doInBackground()`. Aby se sledování průběhu procesu správně aktualizovalo, je nutné přidat aktualizaci stavu do metod s největší časovou náročností na místa, kde se kód pravidelně vrací v cyklech. Tato aktualizace byla přidána do metody nacházející trojúhelníky nazvané `getTriangle()`.

## 6.4 Nové algoritmy

Pro účely porovnávání byl pro přehlednost přidán další balík, který obsahuje několik tříd. Jejich vzájemné propojení je znázorněno na UML diagramu tříd.



obr. 6.14 – UML diagram nových tříd

### 6.4.1 StartCompare

Nadřazená třída `StartCompare` má na starosti režii ostatních metod v balíku, jejím vstupem jsou pouze cesty ke složkám, obsahující řezy k porovnání. Vhodným voláním statických metod udržuje v paměti pouze vstupní řezy a všechny výpočty si nechává zpracovat dalšími statickými třídami. Tyto statické třídy jsou takzvanými jedináčky<sup>7</sup>, tzn. nevytváří se jejich instance, ale jejich metody jsou sdílené a přístupné také všem ostatním třídám.

### 6.4.2 ImageHandler

Třída `ImageHandler` zpracovává všechny vstupní obrazy. Její konstruktor je přetížen několika různými vstupy.

Třída může být vytvořena cestou k souboru či složce, kterou rozliší a buď načte jeden soubor, nebo otevře všechny soubory typu obrázků a uloží je do vytvořeného pole obrázků. Dále může být vytvořena přímo vstupním obrazem typu `BufferedImage`, což bude výhodné, pokud se nebude porovnávat sada řezů z pevného disku ale přímo z paměti těsně po vytvoření daného řezu, což by pomohlo zvýšit výkon. Další možností je vytvoření třídy z matice typu `Matrix`, která se dále využívá k výpočtům, obsahuje tedy také data potřebná k vytvoření obrazu.

### 6.4.3 ImgToMat

Třída `ImageToMat` umí převádět obrázky na matice a naopak. Třída `ImageHandler` využívá převodu obrázků na matice, aby mohla efektivně porovnávat obrázky mezi sebou. Je podobně jako `StartCompare` taktéž statická, tedy není třeba vytvářet její instance, aby bylo možné použít její metody.

### 6.4.4 Compare

Stěžejní část kódu obsahuje další statická třída `Compare`. Obsahuje všechny metody, porovnávající obrazy. Pracuje s maticemi, které vůči sobě porovnává. Metody této třídy budou podrobněji vysvětleny v kapitole 8 Porovnávání.

## 6.5 Další používané aplikace

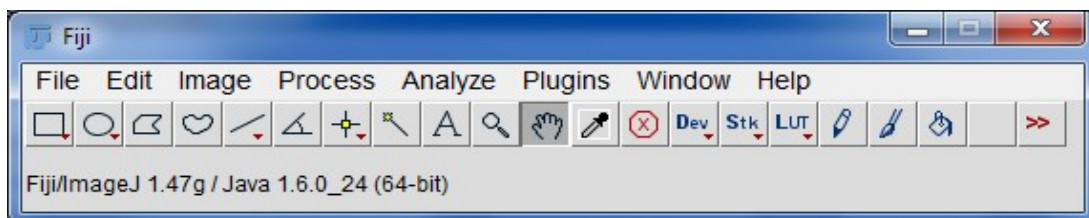
Pro získání všech potřebných dat byly použity další aplikace, umožňující editaci modelů, jako otáčení os či vyhlazování.

Pro editaci modelů byl použit program *Blender*, který je zdarma ke stažení ze stránek [blender.org](http://blender.org). *Blender* je velice sofistikovaný nástroj pro úpravu 3D modelů a scény. Jeho použití je podrobněji popsáno v kapitole 7.4.

---

<sup>7</sup> Jedináček – Singleton – v Javě často používaný vzor, pokud má v celé aplikaci existovat jediná instance třídy

Další použité programy dostupné zdarma jsou *ImageJ* nebo *Fiji*. Aplikace *ImageJ* vyžadovala přítomnost JDK. *Fiji* tento kit nevyžaduje a má v základu více knihoven a zásuvných modulů. Jedná se vlastně jen o jinou distribuci *ImageJ*. Ke stažení k dispozici z [www.fiji.sc](http://www.fiji.sc).



obr. 6.15 - uživatelské rozhraní aplikace Fiji

## 7 MĚŘENÍ VSTUPNÍCH DAT

Pro účely testování bylo využito laboratoře a 3D skeneru, což umožnilo změření vlastního předmětu a získání tak vlastních vstupních dat pro výsledné měření. Kapitola popisuje postup od změření až po extrakci výsledných 3D modelů.

### 7.1 Vlastnosti měření

Velikost rozlišení závisí na použité metodě při získávání dat. Přístroje na principu magnetické rezonance nebo počítačové tomografie získávají data do hloubky zkoumaného objektu v odstínech šedi. Použitý ruční skener ve skutečnosti zaznamenává pouze povrch. Verze přístroje EXAscan nepodporuje snímání barvy předmětu.

#### 7.1.1 Medicínské přístroje

Medicínské snímky typu DICOM, které jsou pořizovány převážně metodami počítačové tomografie nebo magnetickou rezonancí se pořizují s rozlišením okolo 0,4mm/pixel v horizontální rovině a zhruba 1 mm/pixel ve vertikální rovině.

Snímky jsou pořizeny v odstínech šedi. Je tedy nutné vypreparovat potřebné části obrazu pro další zpracování. K tomu slouží segmentace obrazu. Z obrazu v odstínech šedi se selektují data, která jsou pro doktora zajímavá. Do nedávné doby se tato segmentace prováděla pouze ručně, kdy radiolog překresloval naskenované snímky vlastní rukou. Nyní se používají výpočetní metody, které jsou někdy taktéž řešené v Javě. Podrobnější informace k segmentaci obrazu lze nalézt v diplomové práci, zabývající se segmentací pro 3D zobrazení 9 nebo bakalářské práci zabývající se segmentací medicínských dat 9.

#### 7.1.2 Skener pro zpětné inženýrství

Skener od firmy Creaform vlastněný VUT se v praxi nejčastěji používá pro zpětné inženýrství (reverse engineering), což znamená, že se z hotových výrobků snaží získat např. jejich předlohu ve formě CAD souboru. To nebrání použití pro účely této práce.

Při skenování zařízením EXAscan je rozlišení nastavitelné, až do rozlišení 0,05mm/pixel. Při takovéto citlivosti je již ale nutné velice pečlivě skenovat požadovaný objekt. Jinak v povrchu modelu zůstávají díry. Ty jdou post processingovými úpravami vyplnit, ale je to opět jen dopočítaná hodnota – tedy nemusí být vůbec přesná. Navíc při režimu HD skenování se velice rychle zaplní paměť připojeného počítače.

Zvýší-li se citlivost a přesnost, nebude potřeba tolik úprav. Při úpravě přesnějších dat se zvýší i přesnost upravených objektů, např. po použití vyhlazovacích technik. Přesnější přístroje ale navýší výslednou cenu měření, jeho rychlost a paměťové nároky, je tedy nutné najít kompromis mezi maximální možnou dosažitelnou přesností a dostupnými financemi. Ten je právě někde u současných metod, které stále potřebují



využívat úpravy obrazu. Specifikace výrobce skeneru v Tabulka 1.



obr. 7.16– 3D skener VUT EXAscan

## 7.2 Měření skenerem

Samotné měření probíhalo v kanceláři, kde se umístil notebook se softwarem, skener, zkoumaný objekt a rozmístili se markery<sup>8</sup>. Aby se skener zorientoval v prostoru, vyžaduje ve výhledu alespoň 5 markerů. Podle nich se určí poloha os a vzdálenost nebo

---

<sup>8</sup> Marker – černobílé kolečko ve formě nálepky s odrazivou vrstvou, sloužící k orientaci skeneru.

velikost předmětu. Pro omezení rušení z okolních odrazů je vhodné použít proti skeneru zástěnu. Byla použita kartonová krabice polepená markery. Markery byly nalepeny na stůl, měřený objekt a zástěnu. Tím vznikl trojrozměrný prostor, podle kterého se skener orientuje. S měřeným objektem se nesmí hýbat.

Skener dokáže zabrat povrch pouze do polohy, v které má stále přímou viditelnost alespoň na 5 markerů. Při úhlech okolo  $30^\circ$  a méně měl již skener potíže s rozpoznáním markerů a desynchronizoval se přenos dat do počítače. Zařízení pracuje v reálném čase a je možné pouhým pohybem zpět opět zasynchronizovat s viditelnými markery. V průběhu měření lze sledovat, jak se objekt objevuje na obrazovce. Skener také zobrazuje přesnost měření kontrolními diodami a také v okně skenovacího programu. Bylo nutné dodržovat vzdálenost skeneru od objektu tak, aby se laserový paprsek správně odrazil do detektorů odrazu. Rychlost pohybu skeneru má vliv na přesnost měření. Při dostatečné přesnosti a přiměřené rychlosti lze zvýšit citlivost výstupu z přednastavených 2mm/pixel na vyšší rozlišení.

Při použití zástěny nelze naskenovat celý objekt najednou. Teoreticky stačí změřit dvakrát, pokaždé z jedné strany, která byla předtím zakrytá zástěnou. Prakticky je ale potřeba změřit objekt ze tří stran, tvořících trojúhelník, protože při krajních úhlech, téměř rovnoběžných se zástěnou, přístroj nemá viditelnost na dostatek markerů, protože markery na stole a na zástěně jsou pod příliš ostrým úhlem. Ani na rovině zástěny ani na rovině stolu již nerozpoznává markery a klesá přesnost měření. Pokud v krajních úhlech rozpozná pouze několik markerů a software pozná, že se markery neshodují s již uloženými markery v měření, neumí si je přiřadit a po srovnání pohledu tyto markery zahodí.

Při přesunu zástěny se nesmí pohnout s objektem. Dále při spuštění další části měření je nutné zaměřit nejprve stabilní markery, které zůstaly na stejné pozici, jako v předchozím měření. Relativní poloha zástěny vůči objektu se změnila, ale při zasynchronizování na zástěnu, se její přesun projeví, jakoby se přesunul měřený objekt. V případě zaměření markerů na zástěně ještě před zaměřením markerů na stole, software nerozpozná přesun zástěny, naorientuje současný pohled na předchozí polohu zástěny a začne přepisovat předchozí data objektu, která jsou v kolizi se současným stavem. V takovém případě je nejrychlejší spustit nové měření. Při zaměření stabilních markerů se při detekci přesunuté zástěny začnou markery na ní zapisovat jako další, nové. V programu se objeví druhá stěna v pozici, kde předtím nic nebylo. Při třetím kole skenování je objekt načten v téměř uzavřeném prostoru, přestože v každém okamžiku byla zástěna jen z jedné strany objektu. Více o měření 3D skenerem v 9.



obr. 7.17 – měřený objekt – hruška

### 7.3 Výstup programu VXelements

VXelements je software pro 3D skenování vyvinutý firmou Creaform. Program slouží jako komunikátor při skenování mezi skenerem a počítačem. Uživatelské prostředí je velice přívětivé a začátečník se v něm dokáže ihned zorientovat a samostatně pracovat. Obsahuje 3 moduly:

- VXscan<sup>TM</sup> – celý věnován pořizování a optimalizaci 3D skenovaných dat.
- VXprobe<sup>TM</sup> – umožňuje uživateli pracovat s daty získanými pomocí HandyPROBE a sdílet je s ostatními komponenty VXelements nebo s dalším softwarem.
- VXtrack<sup>TM</sup> – přidává dynamické sledování schopností C-Track line pomocí dvou kamerových senzorů. Nyní lze snímat 30-3D obrazů za sekundu reflektorů nebo sad reflektorů v reálném čase pro studium různých potřeb, které zahrnují např. sledování pohybu.

#### 7.3.1 Zpracování měření softwarem Creaform

Pro pořízení 3D modelu byl použit modul VXscan<sup>TM</sup>, který po připojení skeneru dokáže zaznamenávat povrch a v reálném čase zobrazovat 3D povrch během skenování. Nabízí funkce související s optimalizací skenu, jako jsou záplaty děr, odstranění šumu, nebo oprava hranic. 9

Při větší rychlosti měření a obzvláště při větším nastaveném rozlišení se mohou v programu objevovat díry v povrchu objektu, díky nedostatku dat. Naměřená data jsou uložena v projektu měření. Změna citlivosti funguje jako post-processing, kdy

z naměřených dat program vypočítá jemnější povrch. Práce s projektem se poté výrazně zpomalí, protože se při každé změně přepočítává mnohem větší množství vektorů. Zvýšení citlivosti je tedy vhodné zapnout při měření, aby se dala kontrolovat kvalita vstupních dat, poté přepnout na nižší rozlišení, provést veškeré úpravy obrazu, a až jako poslední úpravu udělat opět zvýšení rozlišení. Poté lze vyexportovat kvalitní model v požadovaném formátu.

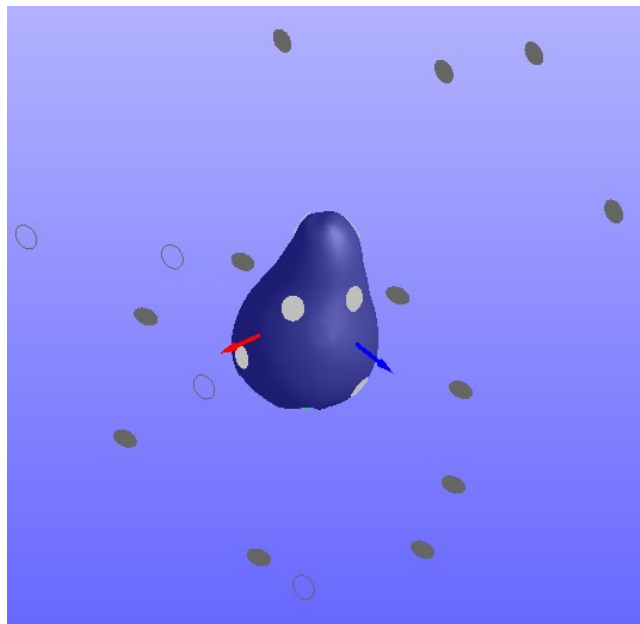
### 7.3.2 Rozhraní programu VXelements a jeho použití

Aplikace VXelements rozlišuje několik základních typů objektů měření. Jejich výčet je zde:

- Project – také označovaný jako Session, má uložená všechna data získaná skenerem a uživatelské úpravy
  - Positioning Targets – markery uložené v souřadnicovém prostoru, relativním vůči centrální ose
  - Entities – základ souřadnicového systému, lze ukládat i vlastní entity jako body, úsečky či rovinné útvary
  - Surface – povrch načtený skenerem bez úprav, má hrubší povrch, obsahuje trhliny
  - Facets – rekonstruovaný povrch, dopočítané chybějící body, mírně vyhlazený
  - Recycle Bin – při odstraňování nadbytečných dat se nabízí otázka, zda data odstranit nebo přesunout do koše (Delete vs. Remove). Remove je vratná akce, Delete trvalá.

### 7.3.3 Extrakce modelu ze skeneru

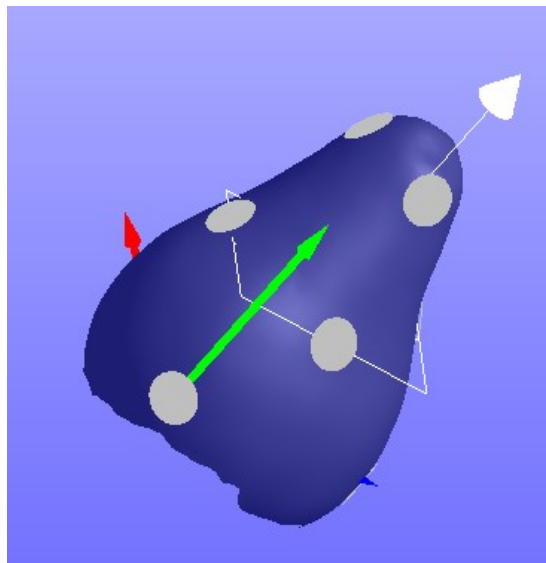
Při postupu uvedeném v kapitole 7.2 vzniká povrch i okolních předmětů, jako je podklad – stůl a zástěna. Ty lze vybrat a odstranit, zůstane tak pouze požadovaný objekt v prostoru orientovaném souřadnicovým systémem založeným podle počáteční polohy skeneru a markerů, viz obr. 7.18.



obr. 7.18 – horní část objektu v prostoru s markery

Protože souřadnicový systém se tvoří automaticky po prvním načtení markerů a záleží na poloze skeneru, nelze při dvou samostatných měřeních s rozdílnou polohou objektů docílit totožného souřadnicového systému. V takovém případě se nedá sloučit objekt naskenovaný z více stran do jednoho souboru, protože každý bude v prostoru jinak natočen.

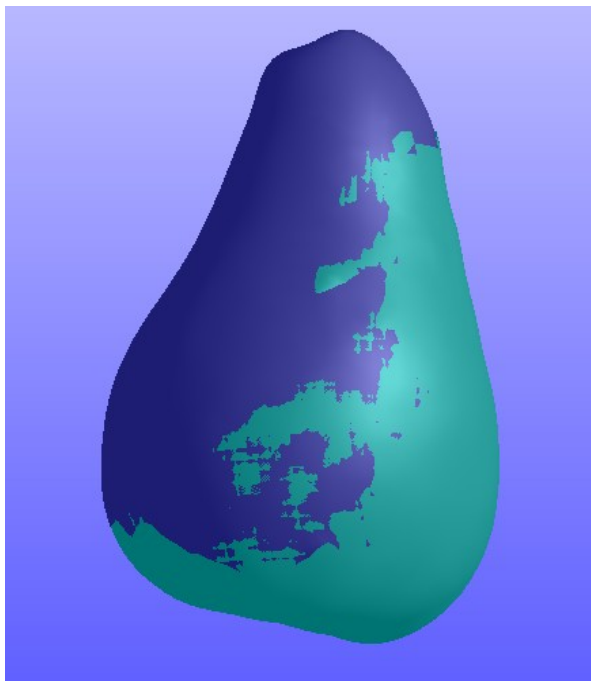
Jednou z možností synchronizace souřadnicového systému je polepení předmětu markery a vytvoření nového souřadnicového systému podle nich. K tomu je potřeba definovat bod – vybere se 1 marker, úsečku – vyberou se 2 markery a plochu – vyberou se 3 markery. Program hlídá, jestli jsou vůči sobě kolmé a vytvoří Entitu, která se dá nastavit jako výchozí souřadnicový systém.



obr. 7.19 – nastavení vlastního souřadnicového systému

Tento facet lze vyexportovat do formátu .obj, .stl nebo jiný a uložit pro další zpracování. Stále je to ale neuzavřený povrch.

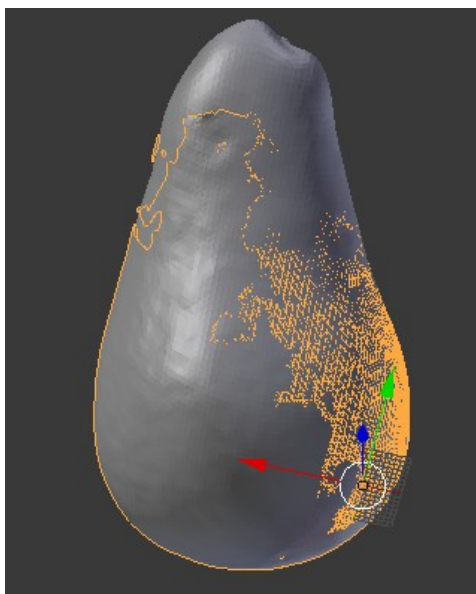
K vytvoření kompletního uzavřeného objektu slouží možnost importování povrchu z jiného projektu. Pokud souhlasí souřadnicový systém, povrchy se překryjí a vytvoří výsledný objekt. Tato sloučení zobrazuje obr. 7.20.



obr. 7.20 – sloučení obou projektů importem

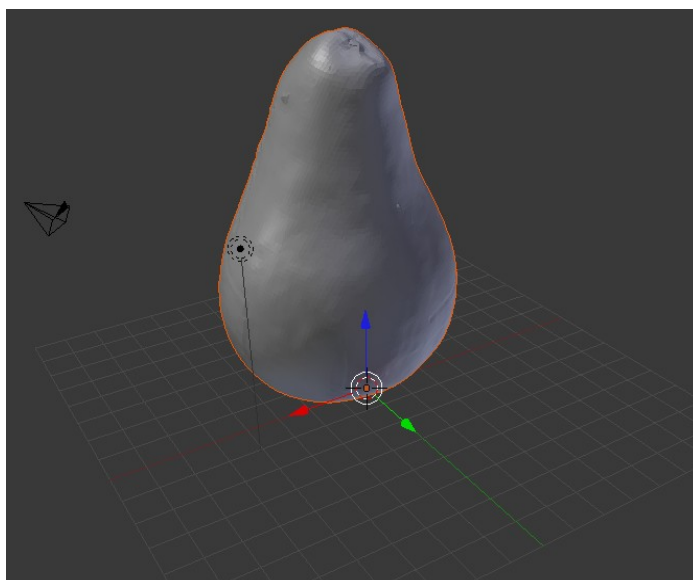
## 7.4 Úprava v programu Blender

Po exportu z programu VElements je vytvořen model, obsahující 2 překrývající se povrchy. K ucelení a vytvoření jednoho objektu slouží funkce odstranění duplicitních bodů. Ta má parametr toleranci vzdálenosti voxelů, který udává, jak blízké body se mohou odstranit.



obr. 7.21 – úpravy obou objektů, jejich spojení a odstranění duplicitních bodů

Když jsou odstraněny zdvojené body, vše je sloučeno do jednoho modelu, zbývá nastavit osy, které jsou z výstupu VElements orientovány náhodným směrem, podle toho, jaké body se vybraly při vytváření souřadnicového systému. Poslední krok je natočení objektu do požadovaných os. 9



obr. 7.22 – finální model, vyrovnáno v osách

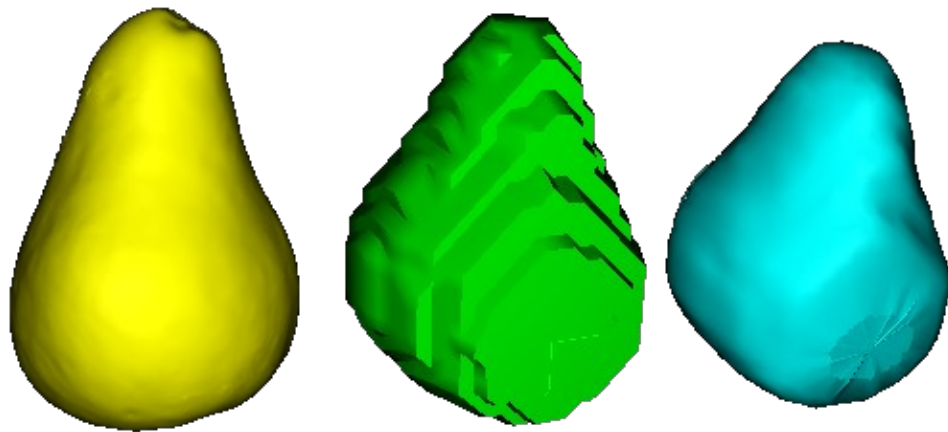
## 7.5 Načtení vstupních dat do aplikace

Aplikace 3D model z MRI má problém s načtením vlastních vstupních dat. Model pear.obj spolu se souborem pear.mtl jsou typu wavefront a aplikace nerozpozná návěští v hlavičce. Kód Javy vyžaduje pro správné otevření změnu návěští „o“ na návěští „g“. Tento problém způsobuje export souboru z programu *Blender*. Místo „Group name“ – g používá návěští „Object name“ – o.

Vyvinutá aplikace otevře vstupní data a funkcí Reslice z nich udělá řezy o zvoleném počtu. Následně z těchto řezů opět sestaví 3D model. Tím podvzorkuje data a připraví je tak k dalšímu zpracování.

## 7.6 Zpracování ve Fiji

Pro úpravu podvzorkovaných dat byl použit program *Fiji*, ve kterém lze otevírat soubory typu .obj a jiné. Na podvzorkovaný model bylo aplikováno vyhlazování. Na první pohled je zřejmé, jak se kvalita zhoršuje.



obr. 7.23 – originál, podvzorkovaný a vyhlazený model



## 8 POROVNÁVÁNÍ

Tato kapitola pojednává o metodách a způsobech porovnávání jak trojrozměrných tak dvourozměrných objektů. Díky vyvinuté aplikaci lze trojrozměrný obraz převést na sadu řezů – dvourozměrných snímků, které na sebe navazují.

Každý objekt má určitou velikost, po jeho úpravách se může změnit. Nelze zaručit, že se řezy provedou v obou objektech ve stejných úrovních. Proto je do metody porovnávání zavedeno automatické zjištění velikosti, které provede takový počet řezů, aby odpovídal velikosti jeden řez na jeden pixel výšky objektu. Tato výška nemusí být celé číslo, čímž vzniká nepřesnost.

Další zdroj chyb je v algoritmu tvořícím řezy, který byl převzat z diplomové práce *3D model z MRI*. Tato metoda implementuje algoritmus marching cubes, která zde není plně transparentní. V průběhu cyklického algoritmu jsou provedeny desetitisíce kroků a je tak nemožné najít zdroj chyb. Problém se nepodařilo napravit ani autorovi práce. Ve výsledných řezech se objevují chybějící úsečky, které negativně ovlivňují výsledky měření.

### 8.1 Bitová úroveň

Každý obraz lze chápat jako dvourozměrnou matici, obsahující hodnoty jasu pixelů. V barevném obraze jsou to tři hodnoty R,G,B respektive Y,Cb,Cr nebo jiné v závislosti na použitém kódování. 9

V jistém slova smyslu lze i dvourozměrný obraz chápat jako vícerozměrný, díky několika hodnotám pro každý jednotlivý bod. Tedy ne jen souřadnici  $x$  a  $y$ , ale hodnotu jasu jednotlivých barevných složek. V tomto případě pouze jedné jasové složky, protože výstupy segmentovaného obrazu jsou černobílé.

Základním porovnáním může být porovnávání na základě matice všech pixelů v obraze. U trojrozměrného obrazu jsou tyto matice samostatné pro každou barvu a každý řez objektem.

#### 8.1.1 Objem

Celkový objem je důležitá hodnota, protože bylo v průběhu této práce prokázáno, že úpravami jako je vyhlazování, se tato hodnota mění, respektive zmenšuje. To je v oblasti medicínských dat nepřipustné. Pokud by při rekonstrukci náhradní kostní tkáň neodpovídal tak zásadní rozměr, jako je objem, nemohl by pacient danou protézu používat nebo by vůbec neseděla do jeho pohybového aparátu.

Byly otestovány dvě metody zjištění objemu, jedna pracující s maticemi, kdy celou matici projde dvěma cykly for a hledá nenulové body. Druhá metoda využívala obrazového datového typu a metody získání histogramu implementované knihovnou II.

Ta ihned předávala parametr počtu bodů v dané jasové úrovni. Výkon metody s histogramem se dle změřených časů zpracování pohyboval zhruba na 30-ti násobku doby, kterou zabrala metoda dvou cyklů. Proto byla metoda s histogramem vyloučena.

Objem tělesa vypočtený z pixelů nemusí odpovídat skutečnému rozměru. Postačí relativní ztráta na objemu ve srovnání s původním objektem. Ta se u testovaných objektů pohybovala v jednotkách procent, což není zanedbatelná hodnota.

### 8.1.2 XOR

Bitový operátor XOR vyjadřuje čistou chybovost neboli rozdílnost dvou porovnávaných obrazů. Pokud je na dané souřadnici jednoho objektu jiná hodnota než na stejné souřadnici objektu druhého, narůstá měřená hodnota XOR. Ve výsledcích měření je uvedena opět jako relativní hodnota, závislá na objemu objektu v daném řezu.



obr. 8.24 – kroky XOR – originál, vyhlazený, rozdíl (XOR)

### 8.1.3 MAE

Mean absolute error neboli střední absolutní chyba se spočítá jako:

$$MAE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |x(m, n) - x'(m, n)| \quad [-]$$

Rovnice 1

- $x$             originální obraz
- $x'$             editovaný obraz
- $M, N$         šířka, výška obrazu

Číselně je to pak stejná hodnota jako výpočet chyby relativním XORem, který je principiálně totožný.

### 8.1.4 MSE

Mean square error neboli střední kvadratická chyba se spočítá podobně jako MAE:

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [x(m, n) - x'(m, n)]^2 \quad [-]$$

Rovnice 2

$x$             originální obraz  
 $x'$             editovaný obraz  
 $M, N$         šířka, výška obrazu

### 8.1.5 PSNR

Peak signal to noise ratio neboli špičkový poměr signálu k šumu se spočítá jako:

$$PSNR = 10 \log \frac{(2^n - 1)^2}{MSE} \quad [dB] \quad \text{Rovnice 3}$$

$n$         počet bitů na jeden vzorek; pro 8bit 255 hodnot

Hodnota PSNR by měla být co největší. Pro totožné vstupy vychází MSE=0 a tedy PSNR=  $\infty$ . Při měření obrazů, u kterých se ztráta objemu pohybovala okolo 7% se naměřené PSNR rovnalo 29dB. Hodnota  $n$  je pouze 2, protože se porovnává černobílý obraz. Protože průběh PSNR je logaritmický, pohybuje se se zvyšující se chybou logaritmickou závislostí. Spolu se ztrátou objemu je to nejvíce vypovídající hodnota, zohledňující více faktorů, jako je velikost, počet chyb a bitová hloubka.

## 8.2 Vektorová úroveň

Na vektorové úrovni se zkoumá podobnost a změna tvarů na základě křivek a ploch. Návaznost bodů lze popsat jako závislost nějaké funkce. Při prokládání bodů při vyhlazování to může být pravidelnost tvaru jako kružnice, přímka či polynomiální funkce. Záleží na použité metodě úpravy obrazu, jestli např. interpoluje nebo aproximuje hodnoty.

Další možnost je vyhledávání podobnosti

V jiném algoritmu, který by šel po obvodu objektu v daném řezu, by se daly vybrat skupiny po sobě jdoucích bodů, které na sebe navazují a udávají tak konvexnost nebo konkávnost křivky. Pokud by po obvodu objektu byl úsek, kde je mezi jednotlivými body narušena hranice objektu, dal by se označit za konkávní. Naopak úsek s navazujícími hodnotami by se označil za konvexní. Porovnávat by se daly tyto intervaly konvexnosti/konkávnosti.

Tyto algoritmy by byly časově extrémně náročné jak na porovnávání, tak na implementaci.

## 8.3 Obrazové filtry

Autokorelace a rychlá Fourierova transformace by mohla poskytnout další údaje o podobnosti objektů, ale jejich implementace se zastavila na nedostupnosti volných knihoven pro Javu k jejímu provedení. Naprogramovat vlastní algoritmus výpočtu dvourozměrné rychlé Fourierovy transformace

# 9 ZÁVĚR

Cílem této diplomové práce bylo v programovacím jazyku Java realizovat aplikaci, která načítá 3D objekty, umí z nich vytvořit řezy a z řezů opět objekty. Dále má tyto objekty umět porovnávat, což bylo hlavní náplní této práce. Tato diplomová práce navazuje na diplomovou práci Tomáše Menčíka: *3D model z MRI*. Některé metody této práce byly upraveny, další zcela nové přibyly. Závěr popisuje výsledný projekt a metodiku měření podobnosti 3D modelů.

Metody pro úpravu modelovaných dat ještě před porovnáním nejsou bohužel bezztrátové. Tyto neekvivalentní úpravy mají negativní dopad na výslednou kvalitu objektů. V medicínských podmínkách je nepřijatelné, aby např. kloubní aparát nahrazený umělou částí měl být odlišný od originálu. Protože vstupní data nelze nasnímat s nekonečnou přesností, je potřeba se vypořádat s tímto faktem a neprovádět úpravy, které vedou k degradaci modelu.

Součástí této práce bylo změření vlastních vstupních dat pro aplikaci. V laboratoři VUT proběhlo měření 3D skenerem vlastního objektu a jeho převedení na 3D model. S přesností změřených dat 1mm na 1 pixel trvá porovnávání velmi dlouho. V případě porovnávání dvou stejně přesných objektů by porovnání trvalo přibližně 16 hodin. Úpravy obrazu jsou tedy více než vhodné. Dokáží zkrátit dobu testování na zlomek času. Pro testovací účely se pracovalo s podvzorkovaným modelem, na kterém se prováděly úpravy jako je vyhlazování. S upraveným modelem trvalo porovnávání maximálně jednotky minut.

Vytvořením aplikace na porovnávání podobnosti je umožněno srovnávat kvalitu různých metod a umožnění výběru vhodných úprav s co nejmenším dopadem na kvalitu výsledného modelu. To může být hodnotným pomocníkem i při vývoji nových metod vyhlazování či dalších úprav modelů.

Vytvořená aplikace je uživatelsky přívětivá, informuje uživatele o stavu a průběhu práce. Podporuje porovnávání 3D modelů ve formátu wavefront .obj a sad řezů obrazu.

Porovnávání modelů změří velikost objektu v dané ose, vytvoří tolik řezů, kolik je velikost modelu v dané ose. Řezy 3D modelů pro každý objekt uloží do nové složky a ty následně porovnává dle indexů tak, aby do vstupu porovnávacího procesu šli řezy umístěné ve stejné rovině ve dvojicích. Tyto řezy jsou binárními obrázky, jsou převedeny do matic, s nimiž se dále počítá. Pro každou dvojici řezů se vypočtou dále uvedené chybovosti. Výsledek měření obsahuje tolik položek chybovostí, kolik je dvojic řezů. Do okna aplikace je vhodné uvádět průměr i medián těchto položek, pro svoje specifické vlastnosti. Čím více se průměr od mediánu liší, tím nepravidelnější je chybovost při průchodu tělesem po dané ose.

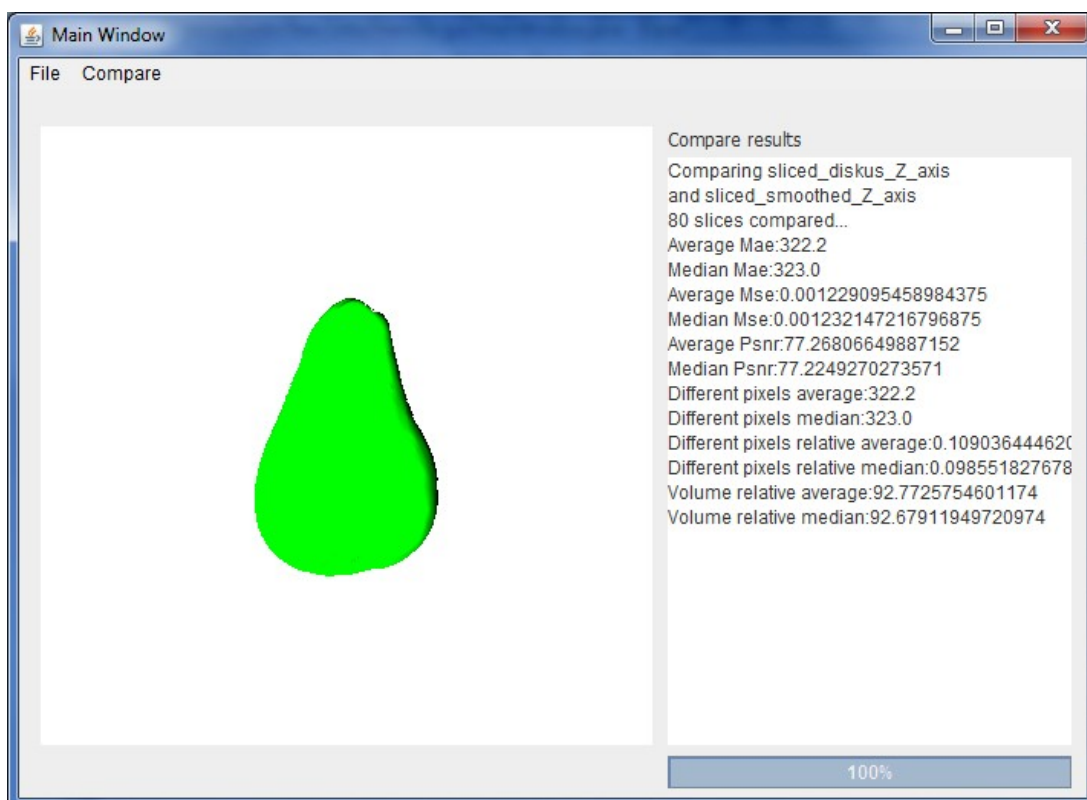
Srovnávány byly parametry objemu obou modelů, z čehož vyšla relativní hodnota změny objemu vztažená k prvnímu porovnávanému objektu. Upravený model zpravidla zmenšil svůj objem o několik procent.

Dále se měřila chybovost MAE, ve výsledku odpovídající metodě XOR vztažená k objemu. Její průměr a medián se příliš nelišil, takže se chybovost udržuje v určitém poměru, který by šel vyjádřit konstantou.

Další metoda vyjadřuje chybovost MSE, která je nezbytná pro výpočet dalších výsledků. Vychází v relativně malých číslech, ale její význam není zanedbatelný. Vychází z ní další metoda měření podobnosti, a to měření PSNR. Pro binární řezy vychází chybovost okolo 30-ti dB. Čím více, tím lépe. Pro totožné objekty vychází všechny chybovosti nulové, kromě PSNR, které bude nekonečně velké. Relativní objem pro totožné objekty vyjde 100%.

Výsledná aplikace po načtení modelu a porovnání originálu s jeho upravenou verzí k náhledu jako obr. 9.25.

V textu je uvedeno maximum zdrojů na literaturu, která je propojená s tématem. Všechny pojmy přeložené z cizího jazyka jsou uváděny se zdroji. V případě nesrovnalostí nebo nevhodného překladu je možno dohledat originální text a lépe tak porozumět problematice.



obr. 9.25 – výsledek porovnání dvou 3D modelů

# LITERATURA

- [1] ALLIEZ, P., COHEN-STEINER, D., YVINEC, M., et al.: *Variational tetrahedral meshing*. ACM Transactions on Graphics. 2005. 24, 3, s. 617-625.
- [2] BALETKA, T. *Pokročilá segmentace obrazu pro 3D zobrazení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 47 s. vedoucí diplomové práce Ing. Libor Boleček.
- [3] Blender. In: [online]. [cit. 2013-05-22]. Dostupné z: <http://www.blender.org/>
- [4] BOROUCHEKI, H., GEORGE, P., HECHT, F., et al.: *Delaunay mesh generation governed by metric specifications. Part 1 : Algorithms. Finite Elements in Analysis and Design* 1997 25, 61-83.
- [5] BOURKE, P. *STL format*. [online]. 1999, [cit. 2012-12-07]. Dostupné z: <http://paulbourke.net/dataformats/stl/>.
- [6] *Comparison of Grid Computing vs. Cluster Computing*. [online]. [cit. 2012-12-09]. Dostupné z: [http://www.jatit.org/research/introduction\\_grid\\_computing.htm](http://www.jatit.org/research/introduction_grid_computing.htm)
- [7] ČERNOCHOVÁ, P. *Diagnostika retinovaných zubů*. Praha: Grada Publishing, 2006, s. 73. ISBN 80-247-1269-5.
- [8] DICOM. In: [online]. [cit. 2013-05-22]. Dostupné z: <http://medical.nema.org/>
- [9] *Difference between JRE and JDK*. In: [online]. 2008 [cit. 2012-12-04]. Dostupné z: [http://it.toolbox.com/wiki/index.php/Difference\\_between\\_JRE\\_and\\_JDK](http://it.toolbox.com/wiki/index.php/Difference_between_JRE_and_JDK)
- [10] DOLEŽALOVÁ, Marie. *Fotorecepce u živočichů* [online]. 2007 [cit. 2012-12-04]. Diplomová práce. Masarykova univerzita, Přírodovědecká fakulta. Vedoucí práce Martin Vácha. Dostupné z: [http://is.muni.cz/th/77649/prif\\_m/diplomova\\_prace.txt](http://is.muni.cz/th/77649/prif_m/diplomova_prace.txt)
- [11] FENSTER, Aaron, D'onal B DOWNEY a H Neale CARDINAL. *Three-dimensional ultrasound imaging*. The John P Robarts Research Institute, London, Canada and the Department of Diagnostic Radiology and Nuclear Medicine, The University of Western Ontario, London, Canada. Dostupné z: <https://www.imt.liu.se/edu/courses/TBMT02/ultra/m105r1.pdf>
- [12] FLEISCHMANN, Peter. *Mesh Generation for Technology CAD in Three Dimensions: Delaunay Slivers*. Wien, 1999. Dostupné z: <http://www.iue.tuwien.ac.at/phd/fleischmann/node55.html>. Dissertation. Technischen Universität Wien Fakultät für Elektrotechnik.
- [13] FRÝZA, T.; HANUS, S. *Komprimace obrazových signálů pomocí transformace 3D-DCT*. Brno: VUT v Brně, 2002. s. 1-65.
- [14] HANÁK, P. *Segmentace obrazů magnetické rezonance*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 35 s. Vedoucí bakalářské práce Ing. Martin Havlíček
- [15] HRUBÝ, Martin. *Detekce změn v 3D MRI datech pomocí registrace*: diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra Kybernetiky, 2009. 68 s. Vedoucí práce byl Dr.Ing. Jan Kybic, Ph.D..
- [16] CHUNG, Johnny. *Head Tracking for Desktop VR Displays using the Wii Remote*. [online]. [cit. 2012-12-07]. Dostupné z: <http://johnnylee.net/projects/wii/>

- [17] Interpolation and Approximation. In: [online]. [cit. 2013-05-22]. Dostupné z: <http://www.math.odu.edu/~jhh/chs4.pdf>
- [18] Java Tutorials. [online]. [cit. 2013-05-29]. Dostupné z: <http://docs.oracle.com/javase/tutorial>
- [19] Johns Hopkins University (2006, January 26). *In The Mind's Eye: How The Brain Makes A Whole Out Of Parts*. ScienceDaily. [cit. 2012-12-04]. Dostupné z: <http://www.sciencedaily.com/releases/2006/01/060125082426.htm>.
- [20] KALLAI, Ilan, Olga MIZRAHI, Wafa TAWACKOLI et al.: *Microcomputed tomography-based structural analysis of various bone tissue regeneration models*. [online]. [cit. 2012-12-07]. Dostupné z: <http://www.nature.com/nprot/journal/v6/n1/full/nprot.2010.180.html>
- [21] KRIVÁ, Zuzana. *Adaptive finite volume methods in image processing*. Bratislava: Slovenská technická univerzita, 2004. 106 s. ISBN 80-227-2048-8.
- [22] LORENSEN, W.E.; CLINE, H.E. *Marching cubes: a high resolution 3D surface construction algorithm*. *Computer Graphics* 1987, 21(4):163–9.
- [23] MAZANEC, Jiří. *Program pro prohlížení a segmentaci 3D MRI dat*: diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra Kybernetiky, 2008. 58 s. Vedoucí práce byl Dr.Ing. Jan Kybic, Ph.D..
- [24] MENCLÍK, Tomáš. *3D model z MRI*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 54 s. Vedoucí práce byl Ing. Ondřej Šmirg.
- [25] MURRAY, J.D.; VAN RYPER, W. *Encyclopedia of Graphics File Formats, 2nd Edition*. 1996. 1116 s. ISBN 1-56592-161-5.
- [26] NIXON, Mark S a Alberto S AGUADO. *Feature extraction and image processin. 2nd ed*. Amsterdam: Academic Press, 2008, xv, 406 s. ISBN 978-0-12-372538-7.
- [27] O'CONNOR MEMORY JUDGEMENTS LAB. *The Physical Reality of fMRI Slice Acquisition Order*. [online]. [cit. 2012-12-07]. Dostupné z: <http://akiraconnor.org/2010/09/01/the-physical-reality-of-fmri-slice-acquisition-order/>
- [28] PARKER, J. *Algorithms for image processing and computer vision*. New York: Wiley Computer Publishing, 1997, xiii, 417 s. ISBN 04-711-4056-2.
- [29] PATZ, Pavel. *Časopis programátor* [online]. 22.07.2003 [cit. 2011-05-25]. Začínáme v Javě (1.): Začínáme. Dostupné z WWW: <<http://casopis.programator.cz/r-art.php?clanek=195>>. ISSN 1213-7359.
- [30] PIEQL, L.A.; TILLER, W. *The NURBS Book, 2nd edition*. 1996. 660 s. ISBN 978-3540615453.
- [31] PIHAN, Roman. *Oko (a mozek) versus fotoaparát*. [online]. 2005 [cit. 2012-12-04]. Dostupné z: [http://www.fotoroman.cz/techniques2/light\\_eye\\_camera.htm](http://www.fotoroman.cz/techniques2/light_eye_camera.htm)
- [32] PILATO, C, Brian W FITZPATRICK a C PILATO. *Version control with subversion*. 2nd ed. Beijing: O'Reilly, 2008, xxiv, 404 S. ISBN 978-0-596-51033-6
- [33] RAMEY, D.; ROSE, L.; TYERMAN, et al.: *MTL material format*. [online]. [cit. 2012-12-07]. 1995. Dostupné z: <http://paulbourke.net/dataformats/mtl/>.
- [34] REDDY, M. [Martinreddy.net](http://www.martinreddy.net) [online]. 1996, [cit. 2012-12-07]. *Wavefront Object Files*. Dostupné z: <http://www.martinreddy.net/gfx/3d/OBJ.spec>.
- [35] RYPL, D. STL File Format. [online]. 2005, [cit. 2012-12-07]. Dostupné z: <http://mech.fsv.cvut.cz/~dr/papers/Lisbon04/node2.html>

- [36] SHEWCHUK, J. R. *Delaunay Refinement Mesh Generation*. Pittsburgh, 1997. Dostupné z: <http://www.cs.cmu.edu/~quake-papers/delaunay-refinement.pdf>. School of Computer Science, Carnegie Mellon University.
- [37] SLÁMA, Lukáš. *Měření laserovým 3D skenerem* [online]. 2011 [cit. 2013-05-25]. Dostupné z: <http://stc.fs.cvut.cz/History/2011/Sbornik/papers/pdf/1100079-1.pdf>. Článek. ČVUT FS. Vedoucí práce Ing. BcA., Jan, Podaný Ph.D.
- [38] ŠPANĚL, M., KRŠEK, P., ŠVUB, M., et al.: *Delaunay-Based Vector Segmentation of Volumetric Medical Images*, In: Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns, CAIP 2007, Berlin Heidelberg, DE, Springer, 2007, s. 261-269, ISBN 3-540-74271-9, ISSN 0302-9743
- [39] ŠPANĚL, M., KRŠEK, P., ŠVUB, M., et al.: *Tetrahedral Meshing of Volumetric Medical Images Respecting Image Edges*, In: Proceedings of the 14th International Conference on Computer Analysis of Images and Patterns, CAIP 2011, Berlin Heidelberg, DE, Springer, 2011, s. 161-169, ISBN 978-3-642-23671-6, ISSN 0302-9743
- [40] *THE OPEN SOURCE DEVELOPER REPORT: 2011 ECLIPSE COMMUNITY SURVEY*. [online]. s. 30 [cit. 2012-12-04]. Dostupné z: [http://www.eclipse.org/org/community\\_survey/Eclipse\\_Survey\\_2011\\_Report.pdf](http://www.eclipse.org/org/community_survey/Eclipse_Survey_2011_Report.pdf)
- [41] VANĚČEK, Petr. *Počítačové hry: Herní engine* [online]. [cit. 2012-12-04]. Dostupné z: <http://herakles.zcu.cz/education/ph/slide/PH-02-tisk.pdf>. Západočeská univerzita v Plzni Fakulta aplikovaných věd.
- [42] VxElements. In: [online]. [cit. 2013-05-22]. Dostupné z: <http://www.3d-skenovani.cz/vxelements>
- [43] WRIGHT, R.S.; LIPCHAK, B. *OpenGL SuperBible, 3rd Edition*. 2004. 1200 s. ISBN 0-672-32601-9.
- [44] YAN, Dong-Ming, Bruno LÉVY a Yang LIU. *Isotropic Remeshing with Fast and Exact Computation of Restricted Voronoi Diagram*. Eurographics Symposium on Geometry Processing, 2009. Dostupné z: <http://research.microsoft.com/en-us/um/people/yangliu/publication/isotropic%20remeshing%20with%20fast%20and%20exact%20computation%20of%20restricted%20voronoi%20diagram.pdf>
- [45] YONGJIE Zhang , CHANDRAJIT Bajaj , BONG-SOO Sohn.: *Adaptive and Quality 3D Meshing from Imaging Data*: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications 2003, Seattle, Washington, USA, June 16 - 20, 2003. pages 286-291, ACM, 2003.
- [46] ZAPLETAL, T. Porovnávání 3D objektů v jazyce JAVA. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 28s. Vedoucí semestrální práce Ing. Ondřej Šmirg.
- [47] ZAPLETAL, T. *Nástroj pro vizualizaci průběhu simulace*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 42 s. Vedoucí bakalářské práce Ing. Radim Burget, Ph.D.
- [48] ZÍDEK, Karel. *Vývoj počítačové grafiky*. In: [online]. Masarykova univerzita Fakulta Informatiky. 2006 [cit. 2012-12-04]. Dostupné z: <http://www.fi.muni.cz/usr/jkucera/pv109/2006/xzidek2.htm>



## SEZNAM ZKRATEK

3D	Tři dimenze v prostoru – výška, šířka a hloubka
API	Application Programming Interface – rozhraní pro programování aplikací
ASCII	American Standard Code for Information Interchange – znaková sada, z které vychází většina současných standardů pro kódování textu
CAD	Computer aided design – počítačem podporované projektování
DICOM	Digital Imaging and Communications in Medicine
FSAA	Full-screen anti-aliasing – vyhlazování na úrovni pixelů, kde se obraz nadvzorkuje, provede se výpočet a opět se podvzorkuje. Výsledný efekt má přiblížit obraz realitě.
IDE	Integrated Development Environment – vývojové prostředí, obvykle zaměřené na jeden programovací jazyk
JDK	Java development kit – sada nástrojů pro Java vývojáře
JRE	Java runtime environment – část nástrojů nezbytných ke spuštění Java aplikací, navíc obsahuje některé knihovny
JVM	Java virtual machine – virtuální stroj od Javy, spouští Java aplikaci
NURBS	Non-uniform rational basis spline – model, definován skupinou kontrolních bodů s různou vahou a uzlovými vektory k reprezentaci křivky. Používán v CAD systémech.
SDK	Software development kit – sada nástrojů pro vývojáře aplikací obecně
SVN	Subversion – kontrola verzí pro více čtenářů/autorů
UML	Unified Modeling Language – grafický jazyk pro vizualizaci, specifikaci návrh a dokumentaci programových systémů.

# PŘÍLOHY

	REVscan	EXAscan	MAXscan	VIUscan
<b>Weight</b>	980 g (2.1 lbs.)	1.25 kg (2.75 lbs.)	1.27 kg (2.80 lbs.)	1.3 kg (2.85 lbs.)
<b>Dimensions</b>	160 x 260 x 210 mm (6.25 x 10.2 x 8.2 in.)	172 x 260 x 216 mm (6.75 x 10.2 x 8.5 in.)	172 x 260 x 216 mm (6.75 x 10.2 x 8.5 in.)	172 x 260 x 216 mm (6.75 x 10.2 x 8.5 in.)
<b>Measurement Rate</b>	18,000 measures/s	25,000 measures/s	18,000 measures/s	18,000 measures/s
<b>Laser Class</b>	II (eye-safe)			
<b>Resolution</b>	0.100 mm (0.004 in.)	0.050 mm (0.002 in.)	0.100 mm (0.004 in.)	0.100 mm (0.004 in.)
<b>Accuracy</b>	Up to 0.050 mm (0.002 in.)	Up to 0.040 mm (0.0016 in.)	Up to 0.050 mm (0.002 in.)	Up to 0.050 mm (0.002 in.)
<b>Volumetric Accuracy(1)</b>	0.020 mm + 0.200 mm/m (0.0008 in. + 0.0024 in./ft)	0.020 mm + 0.100 mm/m (0.0008 in. + 0.0012 in./ft)	0.020 mm + 0.025 mm/m (0.0008 in. + 0.0003 in./ft)	0.020 mm + 0.200 mm/m (0.0008 in. + 0.0024 in./ft)
<b>Volumetric Accuracy(1) (with MaxSHOT 3D)</b>	0.020 mm + 0.025 mm/m (0.0008 in. 0.0003 in./ft)	0.020 mm + 0.025 mm/m (0.0008 in. 0.0003 in./ft)	N/A	0.020 mm + 0.025 mm/m (0.0008 in. 0.0003 in./ft)
<b>Stand-Off Distance</b>	300 mm (12 in.)	300 mm (12 in.)	300 mm (12 in.) (Scan)	300 mm (12 in.)
<b>Depth of Field</b>	± 150 mm (±6 in.)	± 150 mm (±6 in.)	± 150 mm (±6 in.) (Scan)	± 150 mm (±6 in.)
<b>Laser Cross Area</b>	210 mm x 210 mm (8.2 x 8.2 in.)	210 mm x 210 mm (8.2 x 8.2 in.)  Hi-res: 60 mm x 60 mm (2.4 in. x 2.4 in.)	210 mm x 210 mm (8.2 x 8.2 in.)	210 mm x 210 mm (8.2 x 8.2 in.)
<b>Texture Resolution</b>	N/A	N/A	N/A	50 to 250 DPI
<b>Texture Colors</b>	N/A	N/A	N/A	24 bits, SRGB-calibrated
<b>Software</b>	VXelements			
<b>Output Formats</b>	.dae, .fbx, .ma, .obj, .ply, .stl, .txt, .wrl, .x3d, .x3dz, .zpr			

Tabulka 1 – technická specifikace 3D skenerů Creafom (použit byl EXAscan)