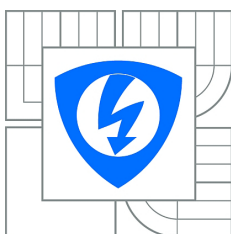


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

VIZUALIZACE 3D DAT V BIOMEDICÍNSKÝCH APLIKACÍCH VISUALIZATION OF 3D DATA IN BIOMEDICAL APPLICATIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL KARZEL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETRA PODLIPNÁ

Abstrakt

Tato práce popisuje základní princip optické koherentní tomografie a předzpracování naměřených surových dat. Předzpracování je zaměřeno zejména na filtraci šumu, odstranění artefaktů, normalizaci, konverzi a kompresi surových dat. Takto předzpracovaná data jsou uložena do souboru *.PFRG jako "preprocessed fringe data". Tato předzpracovaná data je možno vizualizovat pomocí jednoduchého programu, který umožňuje zobrazení dat třemi způsoby. Objemová data reprezentována voxely. Rekonstrukce objemu algoritmem pochodující kostky. Řezy objektem podél os X, Y, a Z.

Summary

This thesis describes the basic principle of optical coherence tomography and preprocessing of the measured raw data. Preprocessing is focused mainly on noise filtration, removing artifacts, normalization, conversion and compression of raw data. In this way preprocessed data is saved in a *. PFRG file as "preprocessed fringe data". Those preprocessed data will be visualised by simply software, which support three methods of visualisation. Volume data represented by voxels. Reconstruction of volume by marching cube algorithm. Cuts through volume along X, Y and Z axis.

Klíčová slova

Optická koherentní tomografie, 3D vizualizace, pseudo-zbarvení, pochodující kostky, shader, raycasting, raytracing, předzpracování, filtrace, šum, artefakty, normalizace, konverze, komprese, Michelsonův interferometr

Keywords

Optical coherence tomography, 3D visualization, pseudo-colors, marching cube, shaders, raytracing, raycasting, preprocessing, filtering, noise, artifact, normalization, conversion, compression, fringe, Michelson interferometer

KARZEL, M. *Vizualizace 3D dat v biomedicínských aplikacích*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 38 s. Vedoucí Ing. Petra Podlipná.

Prohlašuji, že svou bakalářskou práci na téma Vizualizace 3D dat v biomedicínských aplikacích jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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. 40/2009Sb.

V Brně dne

.....
(podpis autora)

Michal Karzel

Chtěl bych poděkovat vedoucí mé bakalářské práce Ing. Petře Podlipné za zájem a věcné připomínky k organizaci a tvorbě práce. Největší poděkování ale patří mé rodině a přátelům za podporu během studia.

Michal Karzel

Obsah

1 Úvod	3
2 Metody 3D vizualizace dat	4
2.1 Modelování a reprezentace těles	4
2.1.1 Objemová data	4
2.1.2 Trojúhelníkové sítě	5
2.1.3 Hraniční reprezentace	5
2.1.4 Modelování pomocí deformací	5
2.2 Detekce povrchu z objemových dat	5
2.2.1 Rovnoběžné řezy a jejich obrysy	5
2.2.2 Pochodující kostky	6
3 Vykreslení 3D objektů	8
3.1 Procesor	8
3.2 Grafický procesor	8
3.2.1 Pole vertexů	9
3.2.2 Shadery	10
3.2.3 Vertex shader	10
3.2.4 Fragment shader	10
4 Barevná reprezentace tělesa a objemových dat	11
4.1 Odstíny šedi	11
4.2 Pseudo-barevné zobrazení	12
4.3 Osvětlení	17
5 Princip optické koherentní tomografie	18
5.1 Schéma Michelsonova interferometru	18
5.2 Schéma optické koherentní tomografie	19
5.3 Fyzikální princip Optické koherentní tomografie	20
5.4 Rozlišení Optické koherentní tomografie	22
6 Zvolené metody zobrazení OCT dat	24
6.1 Trojúhelníkové sítě	24
6.2 Bodové zobrazení	25
7 Předzpracování OCT dat	26
7.1 Specifikace a načtení dat	26
7.2 Filtrace načtených snímků	28
7.3 Konverze, komprese a uložení dat	31
8 Popis software použitého pro vizualizaci	32
8.1 Specifikace	32
8.2 Návod k použití	32
9 Závěr	34

10 Seznam použitých zkratk a symbolů	36
11 Seznam obrázků a tabulek	37
12 Seznam příloh	38

1. Úvod

Optická koherentní tomografie je neinvazivní topografická zobrazovací a diagnostická technika. Ve vysoké kvalitě, jednotek mikrometru, dokáže nasnímat tenké vrstvy vzorků. Světlo blízké infračervenému spektru dokáže pronikat až 3 mm do hloubky. Tato metoda se široce využívá v oftalmologii. Má mnoho využití v biologických aplikacích. Fyzikální princip optické koherentní tomografie je analogický s ultrasonografi, rozdílem je, že optická koherentní tomografie nepoužívá ultrazvuk, ale světlo blízké infračervenému spektru, přibližně 1 mikron.

Cílem této práce je naměřená surová data, která jsou ve formátu FRG, předzpracovat pro další použití. Problematika předzpracování je rozdělena na tři hlavní části: Filtrace, Normalizace a Ukládání. Filtrací je zejména myšleno odstranění rušivých elementů, ale i odstranění obrazových artefaktů vzniklých nevhodnými odrazy a klasických čárových artefaktů. Čárové artefakty se vyskytují ve směru osy x a z . Normalizace zajišťuje využití celého rozsahu dostupných hodnot. Pro dosažení maximální bezztrátové komprimace souboru, ale také rychlosti a spolehlivosti načítání dat, bylo třeba vymyslet řádné algoritmy. Jednotlivé fáze předzpracování jsou diskutovány v kapitole 7.

Filtrace naměřených dat by měla zajistit hlavně kvalitu a přesnost zobrazení naměřených dat. Bez použití vhodných filtrů by výsledek ztrácel smysl. Často jsou data tak silně znehodnocena šumem a artefakty, že jsou nehodnotitelná. Základní zobrazení je realizováno v odstínech šedi, kterých je 256. Pro kvalitu zobrazení a dobrou rozeznatelnost odstínu bodu je nanejvýš vhodné data předem normalizovat. To znamená roztáhnout data přes celou škálu. Jelikož tyto operace zaberou značnou dobu, tak se data ukládají do souboru a není nutné je již nikdy předzpracovat znovu.

Takto předzpracovaná data lze následně vizualizovat pomocí vhodného programu. Pro účely této bakalářské práce byl napsán program, který dokáže data zobrazit třemi způsoby. První způsob je nejzákladnější zobrazení bodové kdy je každý naměřený bod reprezentován svým voxelem. Pro toto zobrazení je vyhotovena škála pseudo-barevných zobrazení. Druhá metoda, nejvýznamnější, sofistikovaným algoritmem rekonstruuje objemovou reprezentaci tělesa. Tento proces je značně náročný, proto je omezen pouze na řezy objektem. Třetí metoda realizuje zobrazení vždy pouze jednoho řezu objektem ve třech osách.

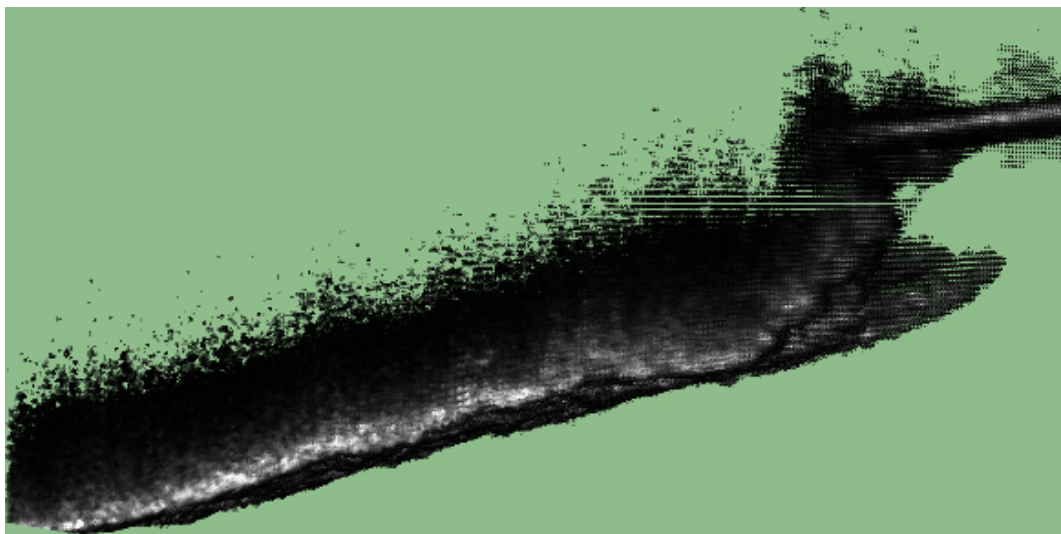
2. Metody 3D vizualizace dat

2.1. Modelování a reprezentace těles

V prostorovém zobrazování má mnoho objektů charakter tělesa. Tato tělesa jsou reprezentací skutečných objektů, které zaujímají určitý objem. Těleso je vlastně jen množinou bodů v trojrozměrném prostoru, splňujících určité podmínky. Pokud definujeme vztahy mezi sousedícími body, můžeme nahlížet na těleso jako na sjednocení dvou množin, které nemají žádné společné prvky. První je množina vnitřních bodů a druhá je množina hraničních bodů. Těleso je definováno striktními pravidly o vztazích sousedících bodů. Vnitřní body mohou sousedit s vnitřními i hraničními body, kdežto body hraniční musí sousedit alespoň s jedním hraničním bodem, vnitřním bodem a vnějším bodem. Tato definice vylučuje z množiny těles objekty jako například úsečky, křivky a plochy. Neobsahují žádné vnitřní body jako těleso, ale jen body hraniční neboli vnější. Tyto objekty jsou však používány k popisu těles [7].

2.1.1. Objemová data

Toto zobrazení zajišťuje nejvěrohodnější zobrazení, kdy každému bodu v definovaném prostoru je přiřazena naměřená hodnota. Zobrazení je sice nejpřesnější, ale je velice náročné na vykreslení a následnou manipulaci. V takovémto vykreslení při nasnímaných datech o velikosti $1024 \cdot 512 \cdot 512$ pixel, může být přibližně $268 \cdot 10^6$ hodnot. Reálně je to po filtraci a optimalizaci kolem $20 \cdot 10^6$ hodnot. Více v kapitole 6.2.



Obrázek 2.1: OCT sken modelovaný objemovými daty

2.1.2. Trojúhelníkové sítě

Jelikož je trojúhelník velice jednoduchý, je oblíbeným objektem pro modelování těles. Má mnoho dobrých vlastností. Hlavní výhodou je to, že oproti obecným mnohoúhelníkům je vždy konkávní a všechny jeho vrcholy leží v jedné rovině. Je podporován grafickým procesorem, což minimalizuje zátěž procesoru. Pro jeho vyplňování existují velice rychlé algoritmy. Více o této metodě viz kapitola 6.1 [7].

2.1.3. Hraniční reprezentace

Nejběžnější způsob reprezentace těles je hraniční popis tělesa, tedy popis množiny hraničních bodů. Informace o vnitřních bodech se neuchovávají. Většinou je lze odvodit z popisu hranice. Je to nejpřirozenější popis tělesa. Prakticky každý člověk kreslí tělesa právě pomocí jejich obrysů, což je podmnožina hranice [7]. Tato metoda však není vhodná pro reprezentaci medicínských dat.

2.1.4. Modelování pomocí deformací

Tato metoda rozšiřuje základní modelování pomocí sčítání či odečítání částí nebo celých základních tvarů jako jsou, válec koule, krychle a toroid o několik operací umožňujících tvarovat skoro libovolná tělesa. Základními operacemi jsou deformace měřítek, deformace zašpičatění, zkroucení a ohýbání. Tyto deformace mohou být uplatněny buď globálně, na celý objekt, nebo lokálně, pouze na část objektu. Výjimkou je deformace měřítek, tu je možno vždy uplatnit jen na celý objekt.

2.2. Detekce povrchu z objemových dat

Aby bylo možné reprezentovat naměřená data jako těleso, je potřeba objemová data převést do povrchové reprezentace. Jak je výše uvedeno, velmi často se pro popis tělesa povrchovou reprezentací využívají trojúhelníkové sítě. K výpočtu povrchových reprezentací bylo vytvořeno hned několik algoritmů. V základu rozlišujeme dva typy metod. První pracuje s obecně ekvidistantně nebo neekvidistantně rozloženými měřenými hodnotami, druhé pracují pouze s daty naměřenými pro ekvidistantní mřížku ve všech směrech. U medicínských měřících zařízení se setkáváme vždy s druhým typem dat. Jen na malé odchylky pracují zařízení s izotropními voxely.

2.2.1. Rovnoběžné řezy a jejich obrisy

Tato metoda se řadí do prvního typu rekonstrukčních algoritmů. Pracuje s daty procházením jednotlivých řezů v naměřených datech. Pro každý řez metoda stanoví hraniční body a jejich obrisy. Z takto stanovených kontur dále sestavujeme těleso pomocí opláštění.

Rekonstrukce objemu opláštěním

Procházením řezu se stanoví pro každé sousedící řezy nejefektivnější propojení sítě trojúhelníků. Takto rekonstruované objemové body pokládáme za objemovou reprezentaci tělesa.

Tato metoda je efektivní pro výpočet povrchového objemu tělesa, kdežto pro sestavení vnitřní struktury tělesa se tato metoda jen těžko realizuje. Stanovit trojúhelníkovou síť pro vnitřní dutiny je velice programově i procesorově náročné.

2.2.2. Pochodující kostky

Loresen a Cline publikovali roku 1987 algoritmus pochodující kostky. Tuto metodu pak aplikovali k nalezení izoplochy a jejímu pokrytí sítě trojúhelníků v tomograficky naměřených datech. Tento algoritmus se naopak řadí mezi algoritmy rekonstruuující objem tělesa z objemových dat pro ekvidistantní mřížku. Důležitým vstupním parametrem, kromě naměřených dat, je hodnota prahu rozhodující o platnosti bodu [7].

Sestavení krychle z dat

Prvním krokem výpočtu je sestavení krychle ze vstupních dat. Krychli tvoří vždy čtyři body z jedné plochy a čtyři body z plochy sousedící. Tyto body reprezentují vrcholy krychle. Krychle se pohybuje s jednou prostorovou souřadnicí konstantní, například z.

Prahování vrcholů

V tomto kroku se aplikuje vstupní prahová hodnota na vrcholy krychle, kde platí, že body menší než prahová hodnota jsou vnitřní, naopak body větší nebo rovny prahové hodnotě jsou vnější. Pokud jsou všechny body krychle ohodnoceny stejně, pokračujeme znovu sestavením krychle vedle. Tato krychle se neúčastní výpočtu.

Indexy pro tabulku

Index je sestaven z 8 bitů, kde každý vrchol reprezentuje jeden bit. Tím dostaneme index od 0 do 255. Tento index se používá pro výběr případu z případové tabulky. Tabulka obsahuje seznam hran krychle, která musí danou plochu promítat. Byla vytvořena na základě patnácti případů, které natočením nebo inverzí bodů reprezentují všech 256 případů. Viz obrázek 2.2.

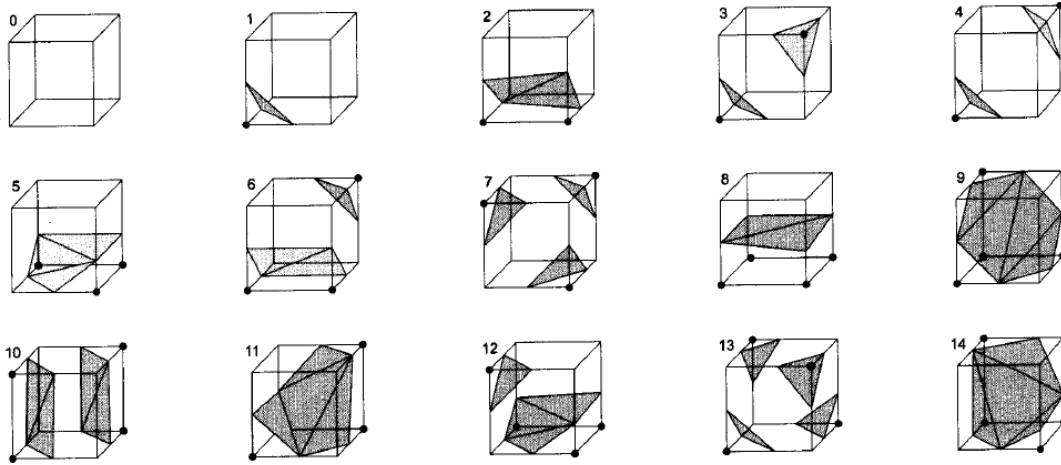
Interpolace bodu trojúhelníků

Přesnou polohu vrcholu trojúhelníku na hraně získáme lineární interpolací. Interpolaci jednoduše vypočítáme podle vzorce 2.1.

$$h(Q) = h(A) + (h(B) - h(A)) \cdot \frac{|Q - A|}{|B - A|} [7] \quad (2.1)$$

V nejhorším případě lze trojúhelník umístit do nejbližšího vrcholu. Pro méně přesné stanovení polohy trojúhelníku je možno trojúhelník umístit do poloviny hrany. Naopak kvadratická interpolace viz. vzorec 2.2 by určila polohu ještě přesněji.

2.2. DETEKCE POVRCHU Z OBJEMOVÝCH DAT



Obrázek 2.2: Případy krychle[7]

$$\begin{aligned}
 t &= \frac{(Q - B)}{(c - B)} \\
 C_0(t) &= (1 - t^3) \\
 C_1(t) &= 3 \cdot t^3 - 6 \cdot t^2 + 4 \\
 C_2(t) &= -3 \cdot t^3 + 3 \cdot t^2 + 3 \cdot t + 1 \\
 C_3(t) &= t^3 \\
 h(Q) &= C_0(t) \cdot h(A) + C_1(t) \cdot h(B) + C_2(t) \cdot h(C) + C_3(t) \cdot h(D) [7]
 \end{aligned}
 \tag{2.2}$$

3. Vykreslení 3D objektů

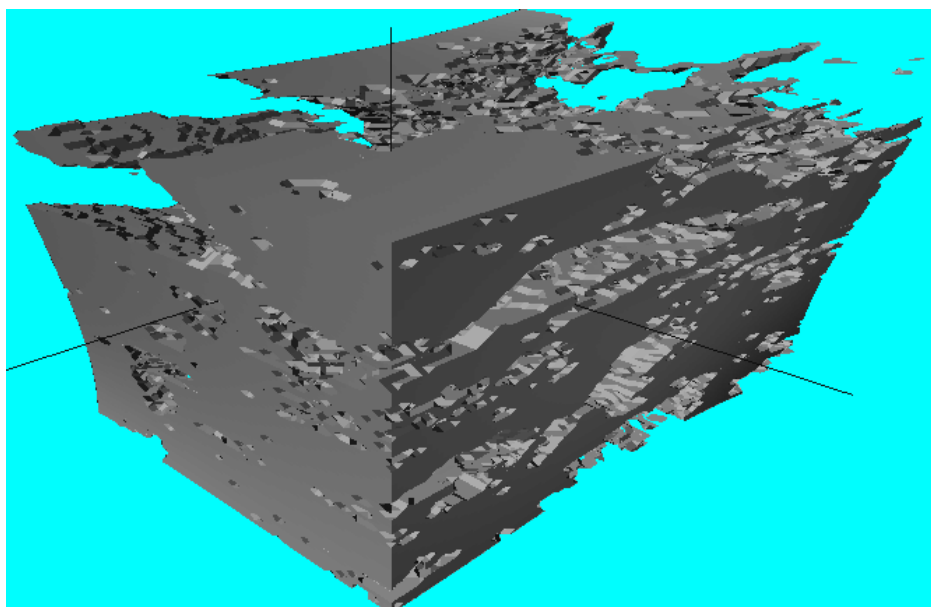
Obecně je třeba zvolené data vykreslit. To je možno provést opět dvěma způsoby. První je sestavení dat pomocí procesoru. Druhou možností je využít akcelerovaného prostředí grafické karty.

3.1. Procesor

CPU neboli také procesor je primárně sestaven ke zpracování matematických operací a programových příkazů. Proto většinou není dostatečně výkonný pro zpracování grafických dat. Jelikož ale často má přeci jen větší výkon než *GPU*, nebo *GPU* nepodporuje požadované grafické funkce, používá se *CPU*.

Zobrazovací seznam

Zobrazovací seznam řeší problém dlouhého sestavování velkého množství vertexů pomocí cyklů. Proces vykreslování pomocí cyklů se provede jen jednou při inicializaci programu a vertexy se uloží do zobrazovacího seznamu. Zavoláním zobrazovacího seznamu se opět vertexy vykreslí. Zavolání je pouze příkaz k opětovnému vyslání dat do grafického jádra.



Obrázek 3.1: Vykreslení rekonstruovaného objemu bez použití shaderů

3.2. Grafický procesor

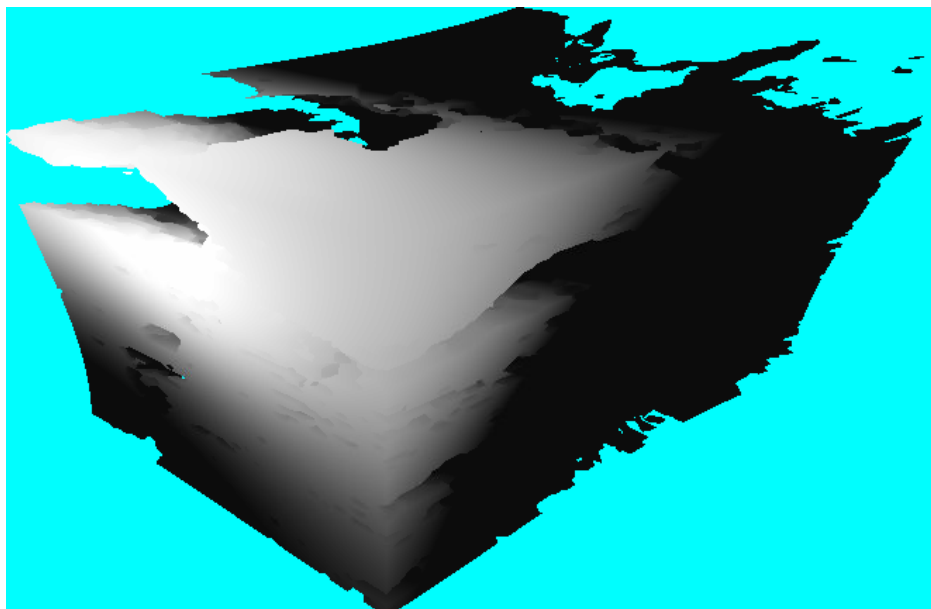
Grafický procesor od verze OpenGL vyšší jak 2.1 umožňuje vykreslení objektu pomocí grafického procesorového jádra. Tento způsob vykreslení je mnohem efektivnější ze strany vytížení procesoru, ale i rychlosti zpracování. Grafické procesorové jádro je optimalizováno pro grafické výpočty a tak efektivněji dokáže sestavovat například trojúhelníkové sítě.

3.2.1. Pole vertexů

Jednoduchým načtením dat do grafických bufferů a následným zavoláním vykreslení již grafický procesor sestaví a vykreslí objekt či data. To se provádí ve více krocích, aby bylo možno vykreslit data, je potřeba do bufferů nahrát všechna nezbytná data. V prvním kroku se nahrají souřadnice vrcholů, to je pro všechny módy vykreslení stejné. V kroku druhém se pro vykreslení *Trojúhelníků*, *Trojúhelníkových sítí*, *Trojúhelníkových vějířů* a dalších metod nahrají indexy vrcholů. Indexy určují pořadí vykreslení, ale i konečný počet vykreslených bodů, který právě odpovídá počtu indexů. Za třetí, znovu pro všechny metody stejně, se nahrají Barvy vertexů. Toto byly nezbytné informace pro vykreslení. Vykreslení je možno doplnit o mnoho dalších parametrů, například parametry světla, stínů a materiálu.

3.2.2. Shadery

Grafické procesorové jádro s verzí OpenGL 3.0+ umožňuje použití takzvaných shaderů. Tyto shadery jsou jednoduché programy. Určují jakým způsobem má grafické procesorové jádro zpracovat data z bufferů. Umožňují upravit chování jednotky pro úpravu vrcholů a zpracování fragmentů [9].



Obrázek 3.2: Vykreslení rekonstruovaného objemu s použitím shaderů

3.2.3. Vertex shader

Tento shader obstarává výpočet nad jednotlivými vertexy samostatně. Program shaderu může ovlivnit pouze některé vlastnosti vertexu jako pozici, normálu, texturovací souřadnice a barvu. Vertex shader nedokáže odebrat či přidat vertexy. Důležité je, že tento shader při zpracování vertexu zná pouze zpracovávaný vrchol. Vertex shader obvykle přepočítává souřadnice světové na obrazové souřadnice. Obstarává osvětlení objektu [9].

3.2.4. Fragment shader

Shader zpracovává každý fragment, může měnit barvu a hloubku fragmentu. Fragment shader naopak umožňuje odstraňovat fragmenty. Pokud je použita textura, je aplikována právě tady [9].

4. Barevná reprezentace tělesa a objemových dat

Aby se bylo možno lépe orientovat v prostoru a pro odlišení přechodu a denzit objektu, je nezbytně nutné objekt nasvětlovat. Pro vyjádření denzity pixelu se používá barva.

4.1. Odstíny šedi

V základním zobrazení se používá odstínů šedi. V takovémto zobrazení bílá barva vyjadřuje nejnižší denzitu a naopak černá reprezentuje denzitu největší. Pro tyto účely je potřeba hodnoty denzity normalizovat. Barvu se zadává buď jako *integer* v rozsahu 0 až 255 nebo jako *GLfloat* od 0.0f do 1.0f.

$$C_{norm} = \frac{C \cdot C_n}{C_{max}} \quad (4.1)$$

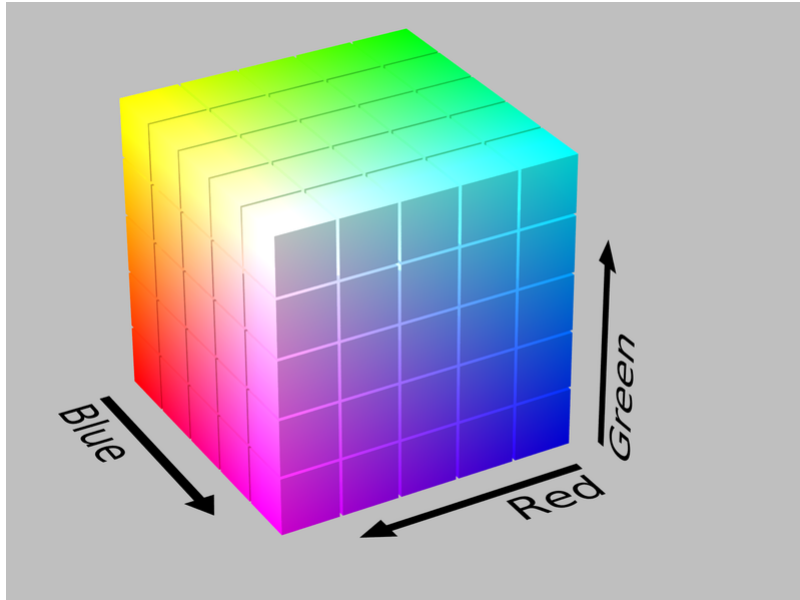
Vzorec 4.1 uvádí obecně normalizaci barvy, kde

C_{norm}	- Výsledná normalizovaná hodnota
C	- aktuální barva
C_n	- velikost maximální normalizované barvy, pro integer 255 a pro GLfloat 1.0f
C_{max}	- velikost maximální barvy

Každá z těchto proměnných je typu *Color3*, je to vektor, který obsahuje barvu vyjádřenou jako *RGB*.

4.2. Pseudo-barevné zobrazení

Toto zobrazení realizuje přepočítání jednočíslné vlastnosti voxelu (densita) na barvu. Barva může být sestavená z RGB, HSV nebo HSL složek. RGB reprezentuje barvu namíchanou ze tří základních zobrazovacích barev, to je červená, zelená a modrá. Tento model je reprezentován krychlí kde složky RGB leží na osách. Znázorněno na obrázku 4.1.



Obrázek 4.1: Rozložení barev v modelu RGB

4.2. PSEUDO-BAREVNÉ ZOBRAZENÍ

Vybarvení tělesa pomocí RGB se provádí přepočítáním denzity na jednotlivé barvy. Dosazením kladných, stejně velikých čísel do modelu (obrázek 4.1) jsou právě odstíny šedi.

Barevné mapy:

Šedotónová

$$R = G = B = D \quad (4.2)$$

Proud

$$\begin{aligned} VF &= 4 \cdot D \\ R &= \text{Min}(VF - 1.5, -VF + 4.5) \\ G &= \text{Min}(VF - 0.5, -VF + 3.5) \\ B &= \text{Min}(VF + 0.5, -VF + 2.5) \end{aligned} \quad (4.3)$$

Podzim

$$\begin{aligned} R &= 1 \\ G &= D/VP - 1 \\ B &= 0 \end{aligned} \quad (4.4)$$

Zima

$$\begin{aligned} R &= 0 \\ G &= D/VP - 1 \\ B &= 1 - 0.5 \cdot D/VP - 1 \end{aligned} \quad (4.5)$$

Jaro

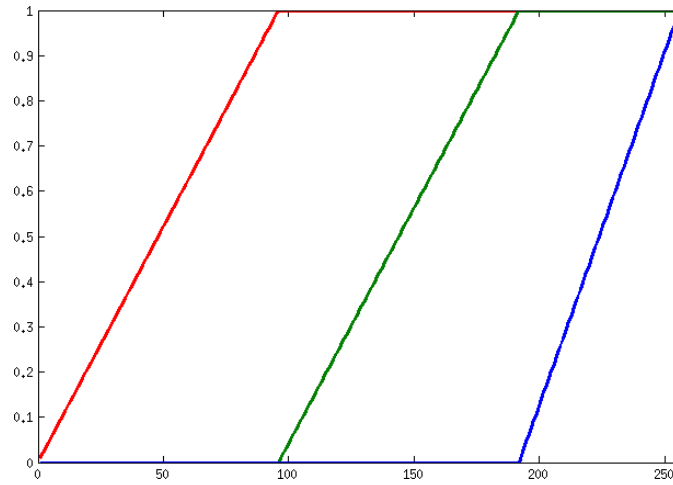
$$\begin{aligned} R &= 1 \\ G &= D/VP - 1 \\ B &= 1 - D/VP - 1 \end{aligned} \quad (4.6)$$

Léto

$$\begin{aligned} R &= D/VP - 1 \\ G &= 0.5 + R/2 \\ B &= 0.4 \end{aligned} \quad (4.7)$$

Chladná

$$\begin{aligned} R &= D/VP - 1 \\ G &= 1 - D/VP - 1 \\ B &= 1 \end{aligned} \quad (4.8)$$

Teplá

Obrázek 4.2: Rozložení složek RGB teplé barevné mapy

Kost

$$\begin{aligned}
 R &= 7 \cdot G_R(D) + H_B(D))/8 \\
 G &= 7 \cdot G_G(D) + H_G(D))/8 \\
 B &= 7 \cdot G_B(D) + H_R(D))/8
 \end{aligned}
 \tag{4.9}$$

Měď

$$\begin{aligned}
 R &= \text{Min}(D \cdot 1.25/VP, 1) \\
 G &= D \cdot 0.7812/VP \\
 B &= D \cdot 0.4975/VP
 \end{aligned}
 \tag{4.10}$$

Čáry

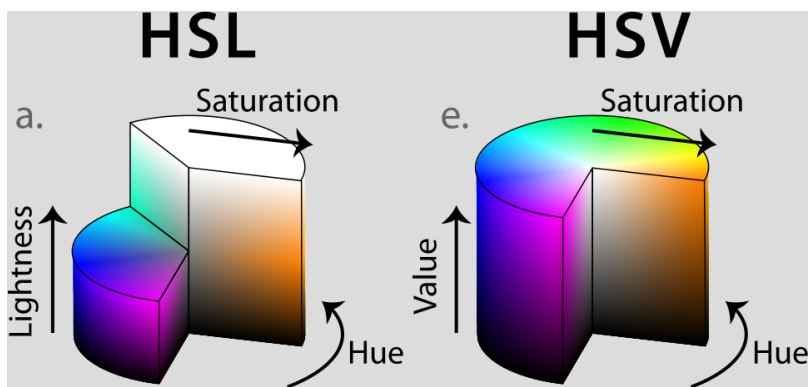
$$RGB(D) = \begin{bmatrix} 0.00 & 0.00 & 1.00 \\ 0.00 & 0.50 & 0.00 \\ 1.00 & 0.00 & 1.00 \\ 0.00 & 0.75 & 0.75 \\ 0.75 & 0.00 & 0.75 \\ 0.75 & 0.75 & 0.00 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}
 \tag{4.11}$$

Matice ve vzorci 4.11 se opakuje pro každých $7 \cdot D$.

Vzorce 4.2 až 4.11 uvádí výpočet RGB složek barvy z denzity, kde

R	- Červená složka barvy
G	- Zelená složka barvy
B	- Modrá složka barvy
D	- Denzita voxelu v rozsahu 0 - 1
VF	- Čtyřnásobná hodnota denzity
VP	- Délka barevné palety
$G_R(D)$	- Červená složka šedotonového modelu pro denzitu D
$G_G(D)$	- Zelená složka šedotonového modelu pro denzitu D
$G_B(D)$	- Modrá složka šedotonového modelu pro denzitu D
$H_R(D)$	- Červená složka teplého modelu pro denzitu D
$H_G(D)$	- Zelená složka teplého modelu pro denzitu D
$H_B(D)$	- Modrá složka teplého modelu pro denzitu D

HSV a HSL využívají válcového rozložení barev, kde osy reprezentují jednotlivé barevné složky. Pro HSV to je H (*Hue*) jako odstín, S (*Saturation*) jako nasycení a V (*Value*) jako hodnota. HSL se liší pouze poslední složkou L (*Lightness*) světlost. Pro lepší znázornění je přiložen obrázek 4.3.



Obrázek 4.3: Rozložení složek HSV a HSL

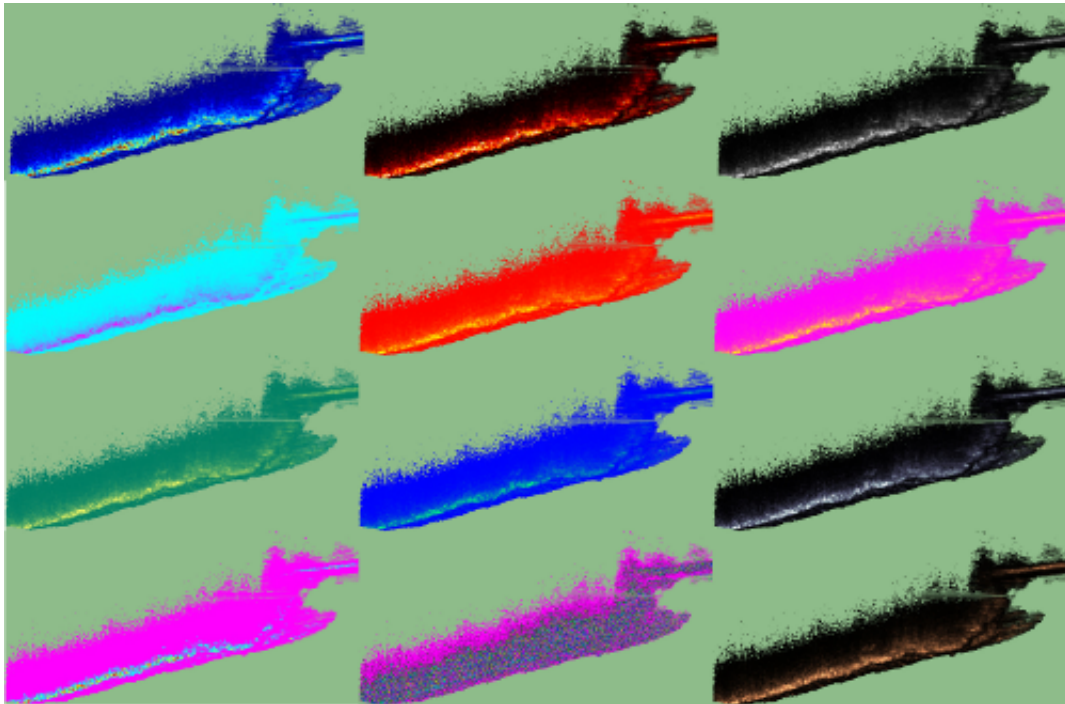
Barevná mapa HSV:

$$\begin{aligned} H &= D \\ S &= 1 \\ V &= 1 \end{aligned} \quad (4.12)$$

Vzorec 4.12 uvádí výpočet HSV složek barvy z denzity kde,

H	- Odstín barvy
S	- Nasycení barvy
V	- Hodnota barvy
D	- Denzita voxelu v rozsahu 0 - 1

Zobrazení denzity objektu pomocí převodu na pseudobarevné zobrazení pomůže lépe rozlišovat přechody denzit v objektu. Zobrazení využívá barevné mapy, takže existuje velká spousta možností zobrazení. V medicínských aplikacích jsou barevné palety předdefinované se specifickými parametry pro jednotlivé případy. Je dále možno měnit jednotlivé intezity barev a tak docílit požadovaného zobrazení.



Obrázek 4.4: Barevné mapy zleva nahoře doprava dolů: Proud, Léto, Šedotónová, Chladná, Teplá, Podzim, Jaro, Zima, Kost, HSV, Čáry, Měď

4.3. Osvětlení

Raycasting

Metoda raycasting je postavena na protínání světelných paprsků vyslaných od pozorovatele do scény, s jednotlivými objekty. Pokud nebudou tělesa průhledná, pro stanovení barvy se uvažuje pouze průsečík, který je k pozorovateli nejbližší. Další průsečíky nejsou pro pozorovatele viditelné, jsou zastíněny blíže postavenými objekty. Obecně platí, že v základu se vysílá tolik paprsků, kolik je pixelů na displeji. Tím je určen i směr paprsků.

Raytracing

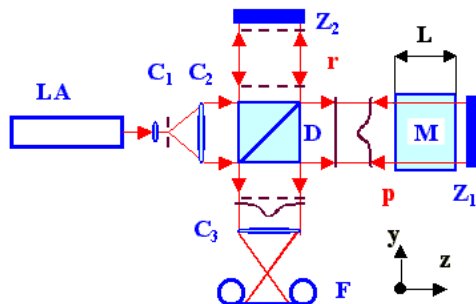
Tato metoda je velice významným rozšířením metody *raycasting*. Vyslání a stanovení barvy v prvním kroku probíhá stejně. V dalších krocích se však spočítá odraz a lom paprsku. Pro tyto *sekundární* paprsky se rekurzivně počítá průběh dále. Rekurze však musí být omezena pouze na několik, obvykle tři až čtyři, aby výpočet skončil v reálném čase. Tento výpočet však už dostačuje k dosažení fotorealistického zobrazení.

5. Princip optické koherentní tomografie

Optická koherentní tomografie (dále OCT) je neinvazivní a bezkontaktní zobrazovací metoda, která využívá zdroj záření v blízké infračervené oblasti. Toto záření je pak distribuováno do vzorku, odkud se odrazí zpět na detektor. OCT je optickou obdobou ultrasonografie a to B-scanu. OCT však díky interferenci dosahuje mnohem větší hloubkové rozlišovací schopnosti. U ultrasonografie se měří rozdíl času, OCT měří vzdálenost interferenčních proužků, které jsou shodné s fázovým posunem odražené a vstupní vlny. OCT je založená na principu Michelsonova interferometru.

5.1. Schéma Michelsonova interferometru

Světlo blízké infračervenému záření zprostředkováno laserem (LA) je přenášeno optickým vláknem do paprskového rozdělovače, na obr. 5.1 realizovaném čočkami C_1 , C_2 . Čočkami je toto světlo upraveno na svazek o větším průměru a děličem D , který rozdělí tento svazek na svazek předměťový p a referenční r . Dále je svazek p směřován na měřený vzorek umístěný v prostoru M o tloušťce L . Po průchodu se odrazí od zrcadla Z_1 a putuje zpět. Původní vlnoplocha se po průchodu vzorkem tam a zpět deformuje. Referenční svazek je směřován na zrcadlo Z_2 , kde se odrazí a putuje zpět do děliče D , kde se svazky znovu spojí a díky jiným drahám svazků p a r vzniká interference. Čočka C_3 slouží k upravení svazku a zobrazení roviny na CCD snímáči F . Vzdálenost mezi paprskovým rozdělovačem D a referenčním zrcadlem Z_2 se plynule mění. Když je vzdálenost mezi zdrojem světla a vzorkem v prostoru M stejná jako vzdálenost mezi zdrojem světla a referenčním zrcadlem Z_2 , odražený svazek p ze vzorku interaguje se svazkem z referenčního zrcadla Z_2 na interferenční obrazec v děliči D . [1][2]

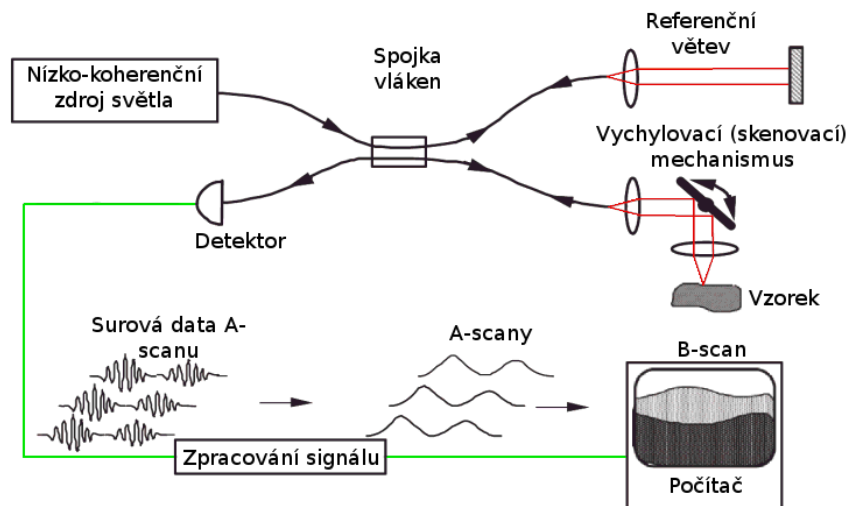


Obrázek 5.1: Michelsonův interferometr[2]

LA	- laser
$C_1 - C_3$	- čočky
D	- dělič
Z_1	- měřený zrcadlový povrch
Z_2	- referenční zrcadlový povrch
F	- CCD snímáč
p	- předměťový svazek
r	- referenční svazek

5.2. Schéma optické koherentní tomografie

Princip obecné optické koherentní tomografie je znázorněn na obr. 5.2. Světlo z nízko-koherentního zdroje, nejčastěji laseru, je vedeno skrze optická vlákna, která vedou skrze spojku optických vláken, která slouží jako jednoduchý Michelsonův interferometr. Spojka optických vláken rozdělí světlo stejným dílem do vlákna vzorku a referenčního vlákna. Paprsek vyzářený z referenčního vlákna se odrazí od referenčního zrcadla a zpožděný se vrací zpět do vlákna. Paprsek z vlákna vzorku projde skenovacím mechanismem, který vychyluje paprsek na vzorek ve směru osy x nebo z. Proces vychylování, nebo jinak řečeno skenování vzorku, je pro jeho přesnost a rychlost řízen mikroprocesorem nebo počítačem. Paprsek se odrazí zpět přes skenovací mechanismus do vlákna, kde se opět spojí s referenčním paprskem a výsledné světlo dopadá na detektor. Signál z detektoru je dále zpracován na A-scan, který reprezentuje tvar vzorku a hloubku jeho odrazivosti s přesnou polohou skenovacího paprsku. Jelikož skenovací mechanismus zaostřuje na různé roviny vzorku, dostáváme více A-scanů, které jsou pak pomocí algoritmů zpracovány, filtrovány a uloženy do počítače jako B-scan. Opakováním tohoto postupu s posuvem vždy o jeden krok ve směru osy z dosáhneme získání 3D dat. Což je jen spojení více B-scanů za sebou na ose z. [3]



Obrázek 5.2: Blokové schéma Optické koherentní tomografie [3]. Tučné čáry zobrazují optické dráhy. Červené zobrazují vystupující paprsek světla a zelené reprezentují elektrické dráhy signálu.

5.3. Fyzikální princip Optické koherentní tomografie

Na obrázku obr. 5.1 můžete vidět uložení vzorku M , u kterého předpokládáme více vnitřních prostředí, od kterých by se mohl paprsek částečně odrazit nebo rozptýlit. Pro zjednodušení popisu budeme předpokládat odraz paprsku pouze na jednom rozhraní prostředí. Popis pro vícevrstvý vzorek je obdobný[4][5].

Ve směru osy z dopadá na polopropustnou destičku interferometru rovinná vlna. [5]

$$\vec{E} = E_0(k) \cos(\omega t - kz) \quad (5.1)$$

$$\omega = |(\delta\varphi \div \delta t)_z| \quad (5.2)$$

$$k = |(\delta\varphi \div \delta z)_z| \quad (5.3)$$

\vec{E}	- vektor elektrické intenzity světelné vlny
E_0	- amplituda světelné vlny
ω	- rychlost změny fáze na čase
k	- rychlost změny fáze na vzdálenosti

Průchodem paprskového děliče D se rozdělí na dva rovnoměrné paprsky. Paprsek referenční r a předmětový p . Jelikož jsou tyto paprsky rovnoměrně rozděleny, musíme tento výraz vydělit $\sqrt{2}$, aby po umocnění amplitud vyšla právě polovina.

$$\vec{E}_r = \frac{E_0(k)}{\sqrt{2}} \cdot \cos(\omega t - kz) \quad (5.4)$$

$$\vec{E}_p = \frac{E_0(k)}{\sqrt{2}} \cdot \cos(\omega t - kz) \quad (5.5)$$

Referenční paprsek putuje k zrcadlu Z_2 , kde se zpozdí o fázi $k * l_r$, fázi přibere ještě jednou, a to při návratu do děliče paprsků D .

$$\vec{E}_r = \frac{E_0(k)}{\sqrt{2}} \cdot \cos(\omega t - k(z + 2l_r)) \quad (5.6)$$

Je-li odrazivost definována jako poměr odraženého výkonu ku dopadajícímu, dostaneme vztah $R_r = r_r^2$, pak je vlna po odrazu definována tímto vztahem.

$$\vec{E}_r = \frac{E_0(k)}{\sqrt{2}} \cdot r_r \cdot \cos(\omega t - k(z + 2l_r)) \quad (5.7)$$

Pro předmětový svazek můžeme rovnici brát za obdobnou, jelikož je vzorek jednovrstvý, rovnice bude vypadat takto.

$$\vec{E}_p = \frac{E_0(k)}{\sqrt{2}} \cdot r_p \cdot \cos(\omega t - k(z + 2l_p)) \quad (5.8)$$

5.3. FYZIKÁLNÍ PRINCIP OPTICKÉ KOHERENTNÍ TOMOGRAFIE

Oba svazky r a p mají stejnou vlnovou délku a neměnný fázový posun v čase, jsou tedy koherentní. Pro jejich sloučení tedy použijeme interferenci.[5]

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cdot \cos(\delta), \quad (5.9)$$

I_1, I_2 - světelné intenzity jednotlivých vln
 δ - fázový rozdíl vln

Fázový rozdíl vln se spočítá jako

$$\delta = 2k \cdot (l_p - l_r). \quad (5.10)$$

Dosadíme do vzorce (ref. .)

$$I(k) = \frac{I_0(k)}{4(R_r + R_p)} + 2I_0(k)\sqrt{R_r R_p} \cdot \cos(2k(l_p - l_r)) \quad (5.11)$$

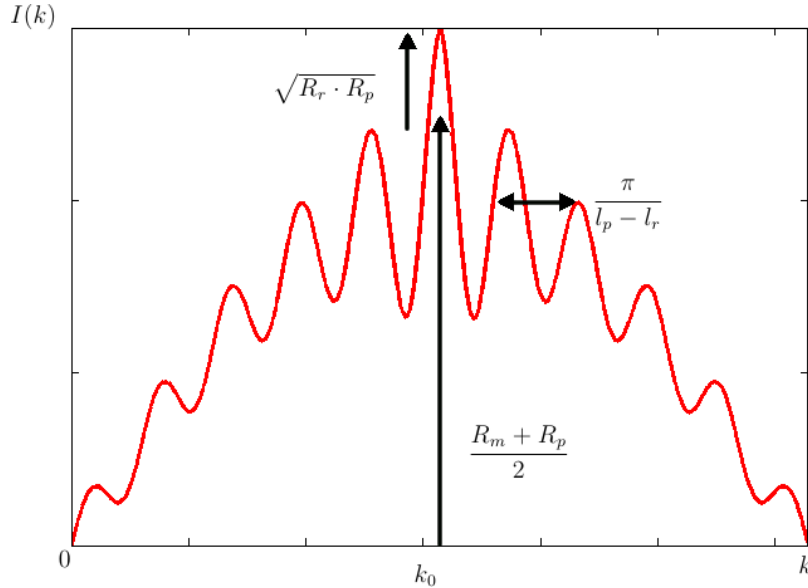
$$I_0(k) = E_0^2(k) \quad (5.12)$$

Pokud do Michelsonova interferometru nepustíme pouze jednu rovinnou vlnoplochu, ale spoustu rovinných vlnoploch o podobných frekvencích, vznikne tzv. rovinné klubko. Toto klubko je popsáno pološířkou klubka Δk a centrální frekvencí k_0 . Když si výraz (5.11) rozdělíme podle součtu dostaneme dvě části. První část $I_0(k) \div 4(R_r + R_p)$ popisuje rozložení vstupních intenzit v klubku, zmenšených odrazivostmi jednotlivých částí přístroje. Část druhá je zajímavější, rozdíl $l_p - l_r$ bude v klubku konstantní, ale jednotlivé vlny obsažené v klubku budou mít různé hodnoty vlnočtu k . Proto budou mít některé z vln obsažené v klubku kosinus maximální a jiné zase minimální. Například matematické odvození dvou po sobě jdoucích maxim [4][5][6],

$$k' - k = \frac{\pi}{l_p - l_r}. \quad (5.13)$$

Rozkmit interferenčních minim a maxim je začátek druhé poloviny vzorce (5.11) $2I_0(k)$.

Pokud použijeme difúzní zdroj světla, je na výstup přenesen zeslabený profil intenzity vstupního rovinného klubka, překrytý modulaci interferenčními minimy a maximy. Vzdálenost vrstvy, na které došlo k odrazu, lze určit z rozkmitu a rychlosti střídání maxim a minim.



Obrázek 5.3: Ukázka spektrálního interferogramu u OCT [3]

5.4. Rozlišení Optické koherentní tomografie

Axiální

Vzorec pro rozlišení v rovině axiální vychází z gaussovského amplitudového spektra použitého světla. Výpočet tohoto spektra můžeme vyjádřit jako součin jeho šířky K v polovině maximální amplitudy ($FWHM$ full-width and half-maximum) a celé amplitudy viz vzorec 5.14 [3][4].

$$\Delta K_{FWHM} \cdot \Delta z_{FWHM} = 8 \cdot \ln 2 \quad (5.14)$$

Jednoduchou úpravou vzorce 5.14 získáme

$$\Delta z_{FWHM} = \frac{8 \cdot \ln 2}{\Delta K_{FWHM}}. \quad (5.15)$$

Rozšířením vzorce 5.15 o vlnové délky v závislosti na amplitudě spektra získáme slibované axiální rozlišení OCT.

$$\Delta z_{FWHM} = \frac{2 \cdot \ln 2}{\pi} \cdot \frac{\bar{\lambda}^2}{\Delta \lambda_{FWHM}} \quad (5.16)$$

kde,

- $\bar{\lambda}$ - střední hodnota vlnové délky zdroje
- $\Delta \lambda$ - šířka spektra v polovině amplitudy

Transverzální rozlišení

Transverzální rozlišení závisí zejména na vzorkovací frekvenci a průměru sondy emitující paprsek. Toto rozlišení je dáno geometrickou velikostí zaostřeného bodu v určené tomografické rovině. Je vyjádřeno vztahem, [4]

$$\Delta d_{FWHM} = 2 \cdot \sqrt{\ln 2} \cdot \frac{\bar{\lambda}}{\pi \cdot \theta} \quad (5.17)$$

5.4. ROZLIŠENÍ OPTICKÉ KOHERENTNÍ TOMOGRAFIE

kde

$\bar{\lambda}$
 θ

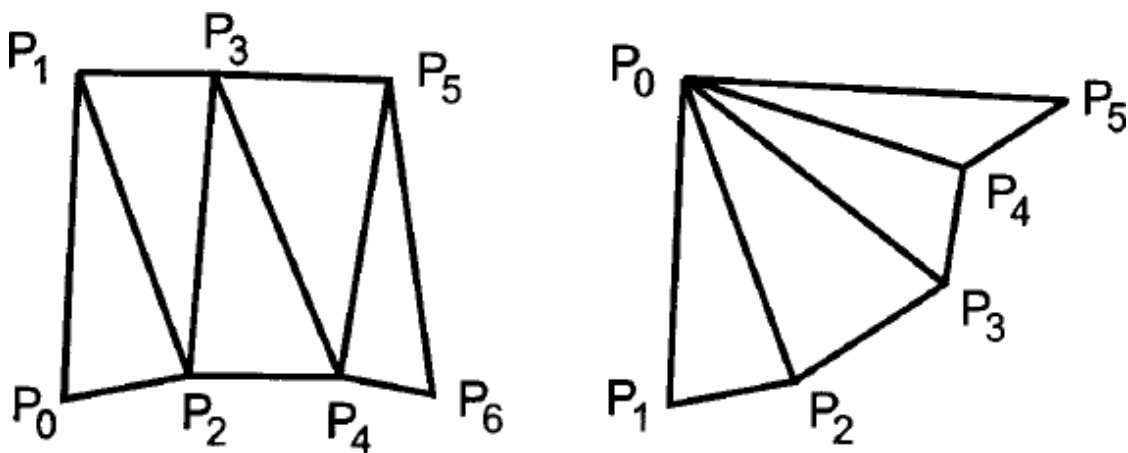
- střední hodnota vlnové délky zdroje
- úhel divergence paprsku s Gaussovským profilem intenzity

6. Zvolené metody zobrazení OCT dat

Tato kapitola se zaměří na vybrané metody 3D vizualizace, které budou použity k modelování těles nasnímaných optickou koherentní tomografií.

6.1. Trojúhelníkové sítě

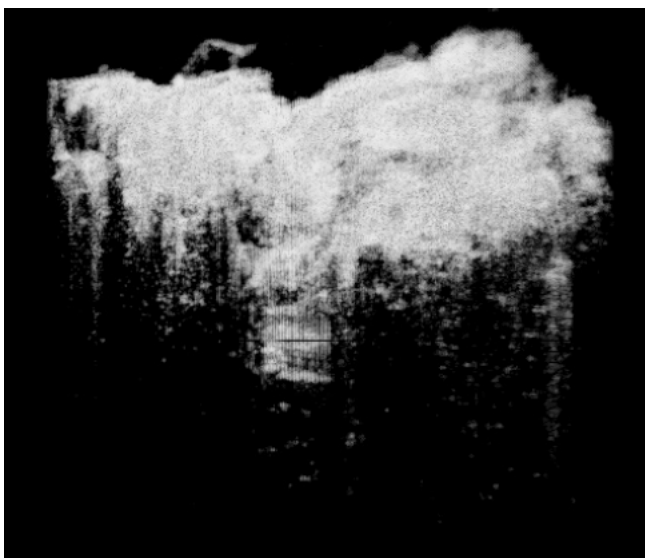
Trojúhelníkové sítě vycházejí ze spojení trojúhelníků, kdy každý z těchto trojúhelníků sdílí alespoň jednu hranu s jiným trojúhelníkem v této síti. Pro realizaci takovéto sítě bývají data rozdělena do dvou logických skupin: *geometrické* a *topologické*. V geometrické skupině jsou uložena data popisující vrcholy trojúhelníků, ve skupině topologické jsou zaznamenány údaje o tom, které body tvoří trojúhelník, popřípadě které trojúhelníky spolu sousedí. Modelování těles trojúhelníkovou sítí však není optimální. Potýkáme se zejména s problémem optimalizace trojúhelníkových sítí při převodu z jiné reprezentace. Účelem je při použití co nejméně trojúhelníků zobrazit těleso co nejpřesněji. Druhý problém se vyskytuje v topologickém rozložení dat. Při zasílání dat do grafického procesoru musíme zasílat pouze jedno pole dat a tak je předchozí rozdělení dat na dvě skupiny nevhodné, data je třeba zpracovat. Zpracování je třeba optimalizovat tak, aby bylo prováděno co nejméně operací s každým vrcholem sítě. Takového zpracování dosáhneme rozdělením dat do *pruhu trojúhelníků* nebo *vějíře trojúhelníků*, jak je znázorněno na obrázku 6.1 [7].



Obrázek 6.1: *pruh trojúhelníků* (vlevo) a *vějíř trojúhelníků* (vpravo)

6.2. Bodové zobrazení

Jelikož nemáme k dispozici geometrický popis tělesa, ale pouze matici vzorků v určitém objemu, nejjednodušší metodou je bodové zobrazení. Data obvykle obsahují rozptýlené vzorky, které nesou informaci o své poloze v prostoru a hodnotu bodu, která reprezentuje většinou nějakou fyzikální či chemickou vlastnost vzorku v daném bodu. Počet takto naměřených dat pak odpovídá citlivosti optické koherentní tomografie, viz kapitola 5.4. Tato metoda je sice velice přesná a zabraňuje chybám, které vznikají při převodu do jiné vizualizace, ale je velice náročná na procesor a paměť počítače. Pro náročnost této metody je prakticky nemožná animace tělesa. Jelikož je tato metoda nezaměnitelná při zobrazování objemů, bude zahrnuta s určitými omezeními.



Obrázek 6.2: Bodové zobrazení objemu

7. Předzpracování OCT dat

7.1. Specifikace a načtení dat

Optická koherentní tomografie, na které bylo provedeno měření, poskytovala data ve více formátech. Výstup ve formátu *.jpg není vhodný k dalšímu zpracování a to zejména charakterem formátu, který se při každém opětovném uložení komprimuje, tím se ztrácí kvalita naměřených dat. Formát zvolený pro další zpracování (*.FRG) obsahuje takzvaná "RAW data", neboli surová data. Tento formát je nekomprimovaný, tím pádem obsahuje nejkvalitnější data, ale samozřejmě je to na úkor velikosti souboru.

Soubor je uložen binárně, proto je nutné při načítání specifikovat délku jednotlivých načítaných úseků. Specifikace této délky je zejména závislá na datovém typu obsahu. Pro získání korektních dat je třeba znát počet znaků a datový typ. Násobkem těchto hodnot získáme přesnou délku úseku podle vzorce 7.1.

$$L = bits \cdot N \quad (7.1)$$

L	- délka úseku
$bits$	- počet bitů datového typu
N	- počet znaků v úseku

Soubor *.FRG je strukturovaný a obsahuje nejen data, ale i informace o pořízení. Obsahuje hlavičku, která je obsažena na začátku souboru. Hlavička obsahuje data, viz tabulka 7.1.

Datový typ "schar" je reprezentován 8 bity, to znamená, že může obsahovat pouze znaky ASCII a rozšířené ASCII tabulky. Tyto znaky jsou kódovány čísly od 0 do 255. "int32" má délku 32 bitů což jsou 4 bajty, takže by měl podle vzorce 7.2 nabývat hodnoty od 0 do 4 294 967 296. Jelikož je to takzvaný "signed int32" se znaménkem, nabývá hodnoty v rozmezí $\pm 2\,147\,483\,647$. Toto je popsáno vzorcem 7.4.

$$N = 2^{bits} \quad (7.2)$$

$$N = \pm 2^{bits-1} \quad (7.3)$$

N	- rozmezí nabývaných hodnot
$bits$	- délka datového typu v bitech

a $bits-1$, odečítáme 1 bit, který je rezervován pro znaménko.

Po přečtení hlavičky pokračujeme výpočtem pro načtení dat snímku, jelikož jsou data samotného snímku uložena jako data před Fourierovou transformací, délku dat spočítáme jako násobek FFT , šířky snímku a to vynásobíme 2, protože Fourierova transformace duplikuje data kolem osy x, je nutné přičíst hodnotu 40, která je konstantně daná a je to velikost hlavičky každého snímku popsané níže.

$$N_B = 40 + FFT * W \cdot 2 \quad (7.4)$$

7.1. SPECIFIKACE A NAČTENÍ DAT

- N_B - počet bajtů každého snímku
 FFT - délka rychlé Fournierovy transformace z hlavičky souboru
 W - šířka snímku

Když už je vypočítána velikost každého snímku, je možno nezávisle číst data jednotlivých snímků. Před načtením čistých dat snímku je třeba přečíst hlavičku souboru.

- Uplynutý čas - čas, který uplynul od předchozího snímku, "int32" o délce 1
 Systémový čas - čas počítače v době snímání snímku, "int32" o délce 1
 Rezervováno - rezervováno pro další položky hlavičky, "int8" o délce 32

Pokud se tedy sečtou délky úseků $2 \cdot int32 + 32 \cdot int8 = 2 \cdot 4 + 32 \cdot 1 = 40$, výsledná hodnota reprezentuje délku hlavičky každého snímku. Poté je možno přečíst surová data snímku jako matici o velikosti $FFT \cdot W$ a datového typu "int16", kterou je následně nutno podrobit zpětné Fourierově transformaci.

Tabulka 7.1: Údaje v hlavičce souboru [8]

Zápis	Popis
Identifikátor souboru	- určuje typ souboru a je datového typu "schar", o délce 16 znaků
Počet snímků	- obsahuje počet obrázků obsažených v tomto souboru, "int32" o délce 1
Šířka snímku	- šířka snímku v pixelech, "int32" o délce 1
Hloubka snímku	- hloubka snímku v pixelech, "int32" o délce 1
Počet snímků v oddílu	- počet snímků zařazených do jednoho oddílu, "int32" o délce 1
Počet oddílů	- celkový počet oddílů, "int32" o délce 1
Délka FFT	- délka rychlé Fourierovy transformace, "int32" o délce 1
Velikost snímku	- velikost snímku v bajtech, "int32" o délce 1
Délka záznamu	- nahraná délka, "int32" o délce 1
PSOCTfringeMode	- PSOCT mod souboru *.FRG (0 nebo 1), "int16" o délce 1
Průměrovací číslo	- počet záznamů použitých k průměrování, "int16" o délce 1
Zobrazovací mód	- zobrazovací mód (0, 1 nebo 2), "int16" o délce 1
Dopplerův jev	- Dopplerův posuv (0 nebo 1), "int16" o délce 1
Rudý posuv, Dopplerův jev	- Prahové hodnoty pro Dopplerovské zobrazení, rudý posuv, "int32" o délce 1
Modrý posuv, Dopplerův jev	- Prahové hodnoty pro Dopplerovské zobrazení, modrý posuv, "int32" o délce 1
Rezervováno	- sekce rezervována pro další hlavičkové položky, "int8" o délce 448
Data snímku	- surová data jednotlivých snímků, "int32" o délce 1

Tabulka 7.2: Doplnující údaje [8]

PSOCT mod souboru *.FRG	0 – Negativní (Nevybráno při načítání) 1 – Pozitivní (Vybráno při načítání)
Zobrazovací mód	0 – Standardní OCT mód 1 – Doppler OCT mód 2 – PSOCT mód
Dopplerův jev:	0 – Negativní (Nevybráno při načítání) 1 – Pozitivní (Vybráno při načítání)

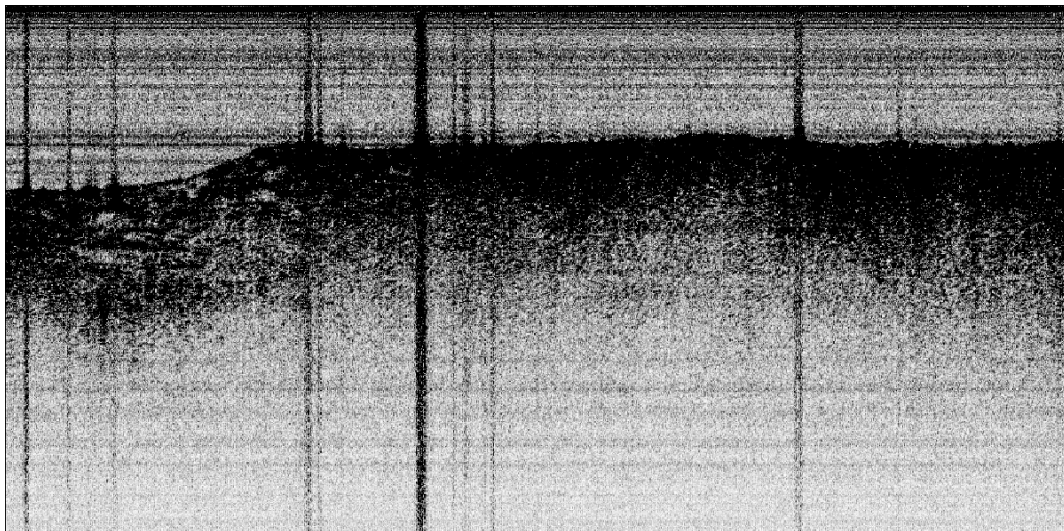
7.2. Filtrace načtených snímků

Pro korektní zobrazení je nutno načtené snímky filtrovat. OCT snímek na tomto zařízení je poškozen zejména šumem a takzvaným čárovým artefaktem. Pro odstranění čárového artefaktu byl použit algoritmus, který vychází z podstaty toho, že rušení ve snímku může být multiplikativní nebo aditivní. Jelikož se čáry vyskytují vždy v celé délce či šířce snímku a adekvátně se sčítají s užitečným signálem, je zřejmé, že se jedná o šum aditivní. Algoritmus je navržen tak, aby vypočítal průměrnou hodnotu sloupce či řádku a tuto průměrnou hodnotu pak od něj odečte, viz vzorec 7.5. Jelikož se artefakt vyskytuje podél obou os, X a Y, měl by se tento algoritmus provést postupně pro oba směry. U snímků, které jsou filtrovány zde, to není nutné. Pro dostatečné odstranění čárového artefaktu postačí snímek podrobit filtraci ve směru osy Y.

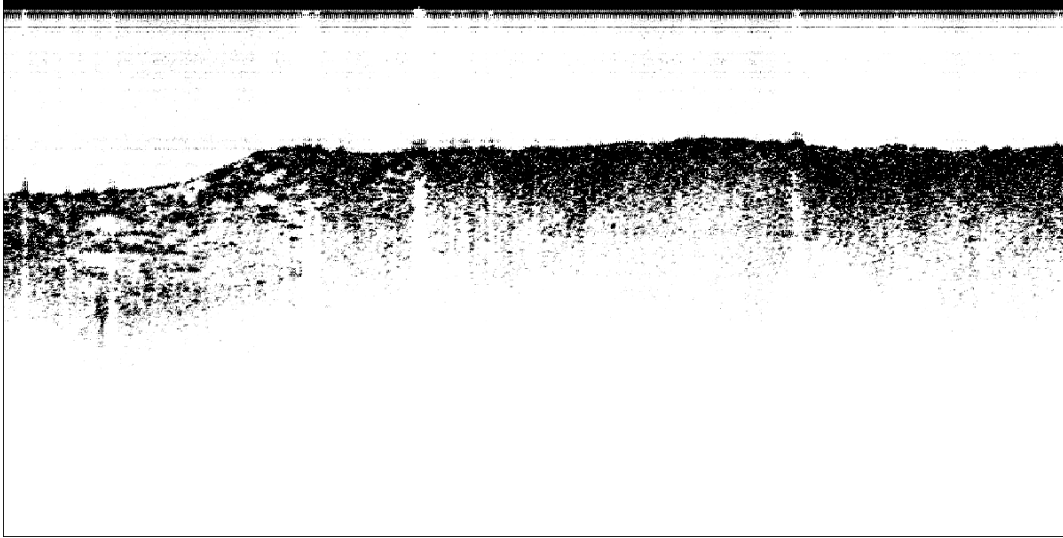
$$Sloupec_1 = Sloupec_1 - \sum \frac{Sloupec_1}{W} \quad (7.5)$$

kde

W - šířka snímku



Obrázek 7.1: Originální invertovaný obrázek

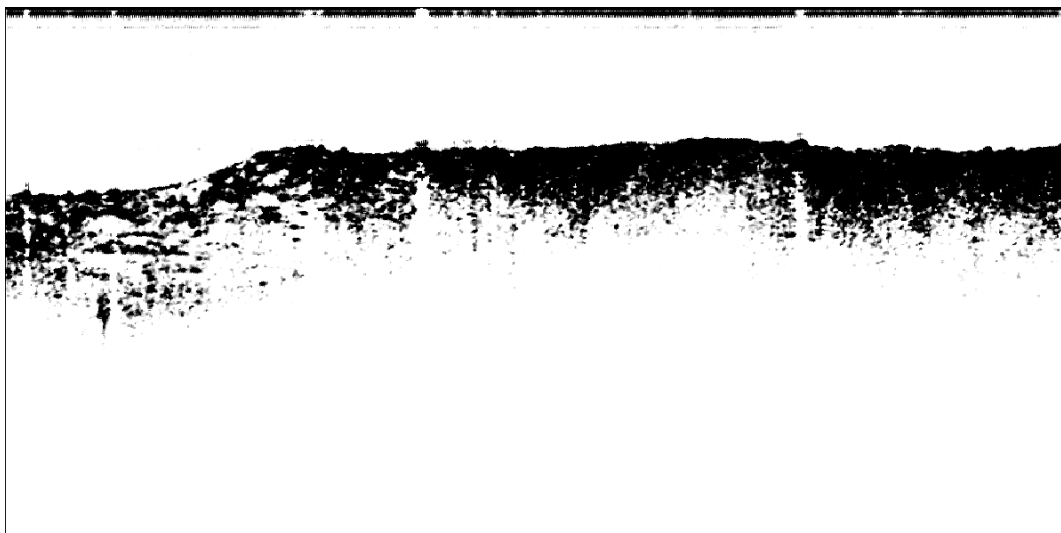


Obrázek 7.2: Odstranění čárového artefaktu podél osy Y, invertovaný obrázek

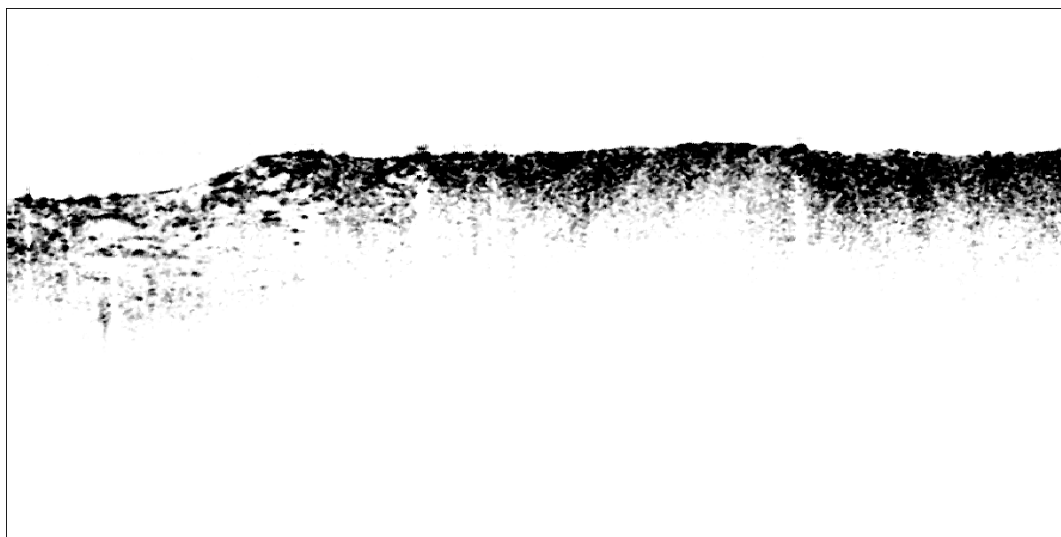
Odstranění šumu může být realizováno průměrováním nebo mediánovou filtrací. Filtrace průměrovacím filtrem poskytuje sice dobrou filtraci, ale rozmazává hrany, kdežto mediánový filtr odstraní šum a naopak zvýrazní hrany. Pro OCT snímky je vhodnější filtrace mediánová, která je realizována v prostředí MATLAB filtrem `medfilt2(A, [m n])`; Vstup A této funkce reprezentuje vstupní 2D matici dat určených k filtraci, jednotlivé snímky OCT. Matice vstupů $[m n]$ určují velikost okna pro výpočet mediánu. Výstupem je pak filtrovaný snímek.

U snímku se vyskytuje ještě jeden artefakt, tím je zarušení v horní části snímku. Je to pruh o šířce cca 15 pixelů, který se postupně zesvětluje. Použité filtry tento artefakt neodstraní, a proto je potřeba jej odstranit na konec. Jelikož se vyskytuje v horní části snímku, kde by se již neměl vyskytovat užitečný signál, je možno jej odstranit smazáním těchto hodnot, respektive nahrazením těchto hodnot nulou, která reprezentuje černou barvu. Po několika desítkách testů na různých vzorcích byla stanovena oblast pro smazání na šířku 30 pixelů.

Pro lepší zobrazení dat se provádí i normalizace, která zajistí zejména rovnoměrné rozložení hodnot v odstínech šedi. Vedlejší funkcí je odstranění nevyužívaných dat po filtraci. Algoritmus, kterým se odstraňuje čárový artefakt, jen posune hodnotu dat mimo zobrazované hodnoty. To znamená, že jsou záporné a je snadné je odstranit opětovným nahrazením těchto hodnot nulou pro všechny záporné hodnoty.



Obrázek 7.3: Mediánová filtrace, invertovaný obrázek



Obrázek 7.4: Normalizace, invertovaný obrázek

7.3. Konverze, komprese a uložení dat

Velikost takového souboru fluktuuje kolem 2 GB, což je pro zpracování veliké množství dat. Je zbytečné a velmi časově náročné načítat a filtrovat takto veliký soubor při každém spuštění, proto je cílem této práce vymyslet vhodnou konverzi a kompresi těchto dat. Jelikož je využívána pouze malá část ze surových dat, a to třídídimenzionální souřadnice a normalizovaná hodnota odstínu šedi, je komprese založená pouze na ukládání těchto dat v binárním módu. Pro ještě větší úsporu velikosti souboru je ignorována černá složka dat. Ignorována může být pouze za předpokladu, že se tuto složku nezapomene, ale bude předpokládáno, že všechny voxely (pixely), které nebudou definovány v souboru, jsou černé. Pro souřadnice je používán datový typ "integer*2", jelikož ve směru osy x je 1024 hodnot a pro hodnotu odstínu šedi "integer*1", který zajistí 256 hodnot.

Do souboru je okopírována celá hlavička, která zabírá pouze málo místa a obsahuje mnoho užitečných dat pro další zpracování. Poté následuje jeden veliký balíček dat obsahující výše zmíněná data uložená jako 4 hodnoty. Výsledkem takovéto komprese je úspora velikosti až na 6 až 10 procent původní velikosti. Samotný proces pro jeden snímek trvá přibližně 1,5 sekundy. Standardní počet snímků v oddílu je přibližně 480, což činí 12 minut. Načtení takto zpracovaného souboru pak trvá řádově v jednotkách sekund. Z původních asi 250 milionů hodnot se ukládá pouze okolo 7 milionů hodnot.

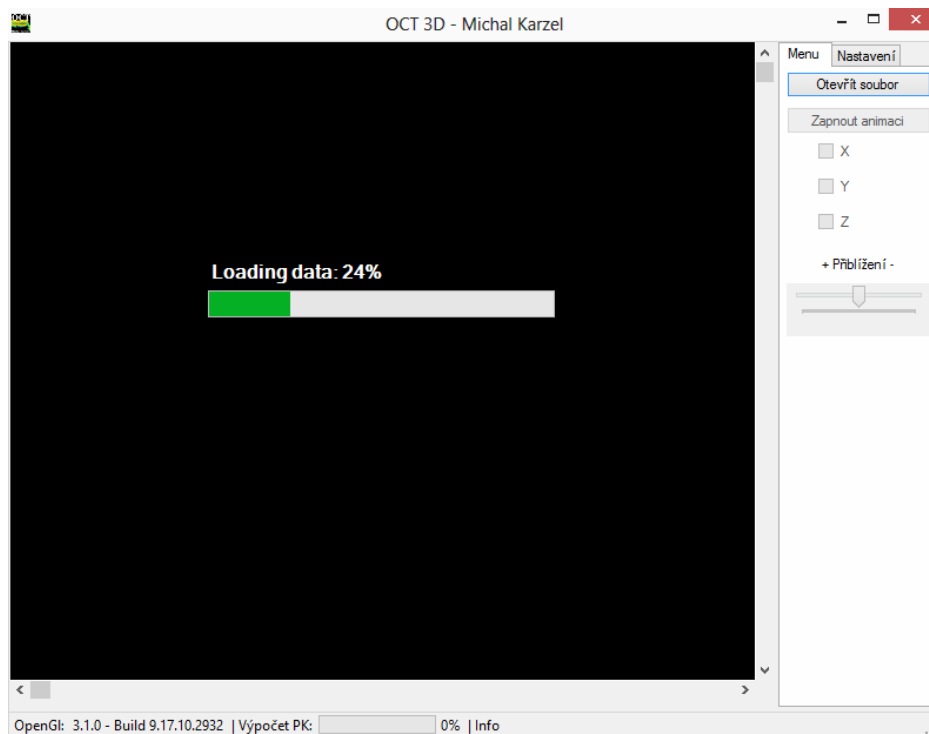
8. Popis software použitého pro vizualizaci

8.1. Specifikace

Tabulka 8.1: Minimální požadována hardwarová a softwarová specifikace

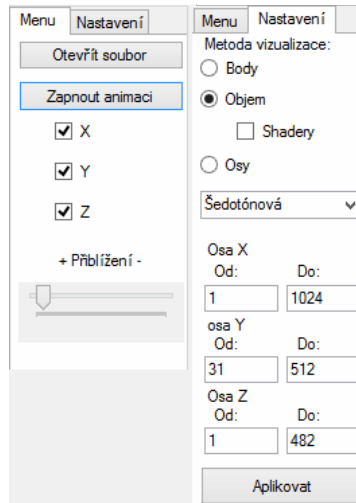
Parametr	Hodnota
Operační systém	Windows 64bit
verze OpenGL	3.0+ (2.1+ bez shaderů)
operační paměť	4GB+

8.2. Návod k použití



Obrázek 8.1: Aplikace pro 3D vizualizaci OCT dat.

Po spuštění aplikace nezle provádět žádné operace dokud nebude vybrán soubor s daty. Kliknutím na tlačítko se otevře průzkumník pro výběr PFRG souboru s přednastaveným filtrem. Výběrem souboru se spustí načítání dat. Po úspěšném načtení dat se povolí nastavení výběru dat ve všech třech osách a vykreslí se základní bodové zobrazení. V základu se nastaví mezní hodnoty podle velikosti načtených dat, takže je možno využít ještě zobrazené řezu v osách. Také se automaticky na pozadí spustí výpočet rekonstrukce pomocí algoritmu pochodující kostky (2.2). Pro informovanost uživatele je průběh vizualizován ukazatelem ve stavové liště programu. Po dokončení se automaticky povolí vizualizace rekonstruovaného objemu tělesa. Pokud jsou dostupné shadery, je možno vykreslení oběma metodami viz. obrázek 8.2.

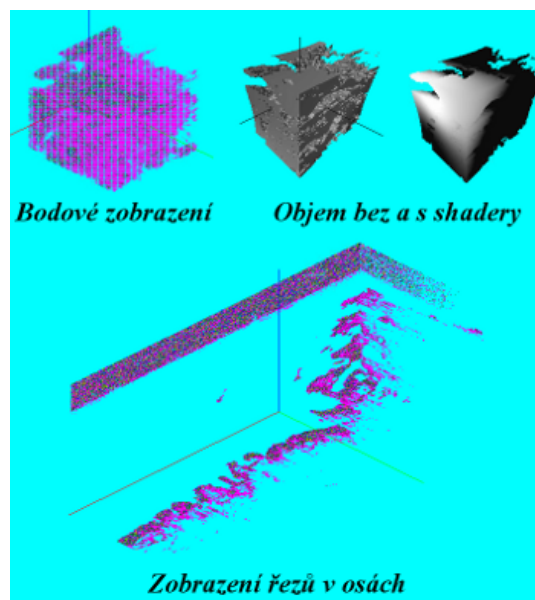


Obrázek 8.2: Nastavení aplikace.

Pro metodu bodového a osového zobrazení lze zvolit z dvanácti barevných map z rolovacího menu pod volbou metody vizualizace.

Manipulovat s vykreslenými daty lze pomocí dvou posuvníků kolem zobrazovacího okna. Vertikální posuvník rotuje tělesem kolem osy X, horizontální kolem osy Y. Rotace kolem osy Z je možná rolováním kolečka myši. Posuvník umístěný v *Menu* jako poslední položka umožňuje přiblížení či oddálení vizualizovaných dat.

Pole, která umožňují nastavení rozsahu dat, slouží ke třem účelům. U bodového zobrazení omezují rozmezí vykreslených dat. U osového vykreslení udává pole *Od* hodnotu na dané ose, pro kterou se data vykreslí. U poslední metody pole *Od* udává počáteční hodnotu pro kterou se rekonstruuje objem. Konečná hodnota je stanovena na *Od* plus 100 pixelů.



Obrázek 8.3: Shrnutí metod vizualizace.

9. Závěr

Surová data naměřená optickou koherentní tomografií jsou předzpracována ve více krocích. Prvním krokem je filtrace, která úspěšně odstranila všechny šum i artefakty. Odstranění artefaktů zajišťuje důmyslný algoritmus popsáný v kapitole 7.3. Zbytkový šum je odfiltrován mediánovou filtrací. Druhým krokem je normalizace, díky které bylo dosaženo kvalitní optické rozlišitelnosti bodů s různými optickými vlastnostmi. Normalizace zajišťuje využití celého spektra odstínů šedi přepočítáním vstupních hodnot. Krok třetí je konverze souboru, který pak obsahuje pouze data potřebná pro další využití. Dále jsou data komprimována. Kompresi je založená na uložení dat v binárním modu, kdy každá z hodnot zabírá čím jak nejméně bajtů. Předzpracování dokáže zmenšit velikost výsledného souboru na 6-10% původního obsahu. Toto je velice důležité pro další využití a rychlost zobrazovacího softwaru.

Takto předzpracovaná data se používají k zobrazení pomocí 3D programu. Tento program umožňuje zobrazení OCT snímků jako těleso. Naměřená data mohou být zobrazena třemi způsoby. První způsob je zobrazení bodové. V bodovém zobrazení je podporováno dvanáct barevných map. Díky této sadě je možno sledovat velmi zřetelně přechody mezi denzitami objektu. Jelikož je tato metoda velice náročná na hardware, je lepší zobrazovat pouze části objektu. Nelze určit, jaké množství bodů je hraniční, jelikož se to liší podle parametrů počítače, na kterém byla aplikace spuštěna. Bodové zobrazení je sestavováno zobrazovacím seznamem (viz. 3.1), je proto možno toto zobrazení využít i pro OpenGL 2.1+. Samotné sestavení není náročné, ale i přesto je pro uživatele vložen ukazatel průběhu.

Druhá metoda vizualizuje rekonstruovaný objem tělesa. Tento objem spolehlivě rekonstruuje algoritmus pochodující kostky (2.2.2). Tato metoda pracuje s velkou spolehlivostí až 99%. Pro účely této aplikace je to nadmíru uspokojující. Zlepšení by se dalo realizovat buď metodou pochodujících čtyřstěnů, která lépe rozdělí prostor, nebo eliminací potenciálně nejednoznačných pozic trojúhelníků, kdy nemusí přesně dosedat hrany sousedících trojúhelníků. Pro obě metody vylepšení lze ještě realizovat optimalizaci počtu trojúhelníků sloučením trojúhelníků ležících v jedné rovině. Pro tuto metodu je realizováno sestavení zobrazovacím seznamem, ale také s využitím shaderů, kdy sestavení i vykreslení obstarává GPU (3.2). Zobrazovací seznam je přibližně stejně náročný jako sestavení pomocí shaderů. Vykreslení shadery, jak je uvedeno v kapitolách 3.1 a 3.2, je výrazně kvalitnější. Velice jednoduché shadery obstarávají vykreslení, osvětlení, ale také i vyhlazení tělesa, které je velice zřetelné.

Třetí metoda vizualizuje tři k sobě kolmé řezy objektem. Takovéto zobrazení vizualizuje spojitosti mezi vrstvami objektu. Toto zobrazení je pouze doplňující a proto na něj nebyl kladen důraz.

Literatura

- [1] SANDEPER SAXENA and TRAVIS A. MEREDITH: *Optical Coherence Tomography in Retinal Disease*. New York: McGraw-Hill, 1988. p. 3-9 ISBN 0-07-160187-2.
- [2] Odbor termomechaniky a techniky prostředí, Fakulta strojního inženýrství, Vysoké učení technické v Brně. Dostupné z <http://ottp.fme.vutbr.cz/~pavelek/optika/1303.htm>
- [3] Drexler, W., Fujimoto, J.G.: *Optical Coherence tomography: Technology and applications*. New York: McGraw-Hill, 2008, p. 48 ISBN 978-3-540-77549-2.
- [4] Fecher, A., Drexler, W., Hitzenberger, C.K., Lasser, T.: *Optical coherence tomography – principles and applications* Published 2003, p. 65
- [5] Aplikovaná optika. Brno: *Kartotéka fyzikálních principů základních optických zařízení a metod užívaných v oftalmologii* Masarykova universita, Fakulta lékařská, 2010, p. 31
- [6] Izatt, J.A., Choma, M.A.: *Theory of optical coherence tomography. Optical Coherence Tomography Technology and application*. 2008, ISBN: 978-3-540-77549-2 p. 27
- [7] J. Žára, B. Beneš, J. Sochor, P. Felkel: *Moderní počítačová grafika* Computer Press, a.s. nám. 28. dubna 48 Brno, ISBN: 80-251-04254-0 p. 237-262
- [8] *THORLABS Swept Source OCT System Software User's Manual*. 18100-D03, Rev D, 5/26/10 p. 69-70
- [9] ZLÁMAL, J.: *Jazyky pro programování shaderů*. [Bakalářská práce.] Brno: MU, FI, 2008. 3 s.

10. Seznam použitých zkratek a symbolů

W	Šířka snímku v pixelech
FFT	Fourierova transformace
N_B	počet bajtů každého snímku
N	rozmezí nabývaných hodnot
L	délka úseku
$\bar{\lambda}$	střední hodnota vlnové délky zdroje
Θ	úhel divergence paprsku s Gaussovským profilem intenzity
$\Delta\lambda$	šířka spektra v polovině amplitudy
ΔK_{FWHM}	šířka spektra
Δd_{FWHM}	transverzální rozlišení
Δz_{FWHM}	axiální rozlišení
\vec{E}	vektor elektrické intenzity světelné vlny
E_0	amplituda světelné vlny
ω	rychlost změny fáze na čase
k	rychlost změny fáze na vzdálenosti
\vec{E}_r	vektor elektrické intenzity světelné vlny referenčního svazku
\vec{E}_p	vektor elektrické intenzity světelné vlny předmětového svazku
z	osa z
t	čas
R_r	odrazivost referenčního zrcadla
R_p	odrazivost předmětového zrcadla
I	světelná intenzita vlny
δ	fázový rozdíl vln

11. Seznam obrázků a tabulek

- Obrázek 2.1 OCT sken modelovaný objemovými daty
- Obrázek 2.2 Případy krychle[7]
- Obrázek 3.1 Vykreslení rekonstruovaného objemu bez použití shaderů
- Obrázek 3.2 Vykreslení rekonstruovaného objemu s použitím shaderů
- Obrázek 4.1 Rozložení barev v modelu RGB
- Obrázek 4.2 Rozložení složek RGB teplé barevné mapy
- Obrázek 4.3 Rozložení složek HSV a HSL
- Obrázek 4.4 Barevné mapy zleva nahoře doprava dolů: Proud, Léto, Šedotónová, Chladná, Teplá, Podzim, Jaro, Zima, Kost, HSV, Čáry, Měď
- Obrázek 5.1 Michelsonův interferometr[2]
- Obrázek 5.2 Blokové schéma Optické koherentní tomografie [3]. Tučné čáry zobrazují optické dráhy. Červené zobrazují vystupující paprsek světla a zelené reprezentují elektrické dráhy signálu.
- Obrázek 5.3 Ukázka spektrálního interferogramu u OCT [3]
- Obrázek 6.1 *pruh trojúhelníků* (vlevo) a *vějíř trojúhelníků* (vpravo)
- Obrázek 6.2 Bodové zobrazení objemu
- Obrázek 7.1 Originální invertovaný obrázek
- Obrázek 7.2 Odstranění čárového artefaktu podél osy Y, invertovaný obrázek
- Obrázek 7.3 Mediánová filtrace, invertovaný obrázek
- Obrázek 7.4 Normalizace, invertovaný obrázek
- Obrázek 8.1 Aplikace pro 3D vizualizaci OCT dat
- Obrázek 8.2 Nastavení aplikace
- Obrázek 8.3 Shrnutí metod vizualizace
- Tabulka 7.1 Údaje v hlavičce souboru [8]
- Tabulka 7.2 Doplnující údaje [8]
- Tabulka 8.1 Minimální požadována hardwarová a softwarová specifikace

12. Seznam příloh

FRG2PFRG.m Zpracování a konverze dat Matlab

OCT3D.exe Vizualizační software

OpenTK.dll Knihovna OpenGL pro C#

OpenTK.GLControl.dll
Knihovna OpenGL pro C#

OpenTK.xml

OpenTK.GLControl.xml

lumitekz3D.PFRG
Předzpracovaná data

snimek3D.PFRG
Předzpracovaná data

1.png Dodatečné obrázky

2.png Dodatečné obrázky

3.png Dodatečné obrázky

4.png Dodatečné obrázky