

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE

2004

Filip Vůjtěch

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE

Filip Vůjtěch
Vizualizace tomografických dat z vyšetření mozku
Katedra softwarového inženýrství
Vedoucí diplomové práce: Dr. Ing. Daniel Janeba
Studijní program: Informatika, Softwarové systémy, Počítačová grafika

Tímto bych rád poděkoval vedoucímu mé diplomové práce Dr. Ing. Danielu Janebovi za jeho trpělivost a podporu v celém průběhu tvorby této práce.

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

V Praze dne

Filip Vůjtěch

.....

Obsah

| | | |
|-------|---|----|
| 1 | Úvod | 9 |
| 1.1 | Vyšetření v nukleární medicíně | 9 |
| 1.2 | Zobrazování v nukleární medicíně | 9 |
| 1.2.1 | 2D data | 9 |
| 1.2.2 | 3D data | 9 |
| 1.3 | Cíle práce | 11 |
| 2 | Implementace | 12 |
| 2.1 | Cíle implementace | 12 |
| 2.2 | Volba prostředí a vývojových nástrojů | 12 |
| 2.3 | Vstupní data | 12 |
| 2.4 | Zobrazovací metody | 13 |
| 2.4.1 | Použité souřadné soustavy | 14 |
| 2.5 | Metody výpočtu normál povrchu | 14 |
| 2.5.1 | Metody odhadu normál založené na odhadu gradientu | 14 |
| 2.5.2 | Adaptivní metody odhadu normál založené na odhadu gradientu | 15 |
| 2.5.3 | Geometrické metody odhadu normál | 16 |
| 2.6 | Řezy | 16 |
| 2.6.1 | Jednoduchý řez trojúhelníkovým modelem | 17 |
| 2.6.2 | Uzavřenost povrchu modelu bez řezů | 17 |
| 2.6.3 | Dopočítání trojúhelníků povrchu v místě řezu | 18 |
| 2.6.4 | Více řezů modelem | 20 |
| 2.6.5 | Odfiltrování trojúhelníků řezovým výrazem | 21 |
| 2.6.6 | Textura řezu | 23 |
| 2.6.7 | Vylepšený algoritmus pro výpočet textury | 25 |
| 2.6.8 | Namapování textury na trojúhelníky řezu | 27 |
| 2.6.9 | Řezy bodovým modelem | 28 |
| 2.7 | Export | 29 |
| 2.7.1 | Export statických snímků | 29 |
| 2.7.2 | Export animace ze série statických snímků | 29 |
| 2.7.3 | Export statických snímků do NOP | 29 |
| 2.7.4 | Export 3D modelu do VRML | 31 |
| 2.8 | Klíčové body | 32 |
| 2.8.1 | Speciální interpolace řezových výrazů | 32 |
| 2.9 | Přepočítání modelu při změně parametrů | 33 |
| 2.9.1 | Změny metody zobrazení a výpočtu normál povrchu | 33 |
| 2.9.2 | Změny parametrů bez vlivu na 3D model | 33 |
| 2.9.3 | Změny parametrů s vlivem na 3D model | 34 |
| 2.10 | Závěr implementační části | 34 |
| 3 | Porovnání zobrazovacích metod | 36 |
| 3.1 | Cíle porovnání | 36 |
| 3.2 | Jak porovnávat | 36 |
| 3.3 | S čím porovnávat | 36 |
| 3.4 | NOP formát | 37 |
| 3.5 | Nástroj pro generování testovacích dat (TDC) | 37 |
| 3.5.1 | TDD soubor | 37 |
| 3.5.2 | Použití TDC | 40 |
| 3.5.3 | Interní postup generování testovací studie Interfile | 40 |
| 3.5.4 | Interní postup generování testovací projekce NOP | 43 |
| 3.6 | Nástroj pro porovnání NOP projekcí (NOPView) | 43 |
| 3.6.1 | Funkce NOPView | 44 |

| | | |
|-------|---|----|
| 3.6.2 | Statistiky | 46 |
| 3.7 | Srovnání implementovaných zobrazovacích metod | 47 |
| 3.7.1 | Použitá testovací data | 47 |
| 3.7.2 | Počet vrcholů povrchu použitých jednotlivými metodami | 48 |
| 3.7.3 | Geometrie povrchu | 49 |
| 3.7.4 | Normály povrchu | 50 |
| 3.8 | Shrnutí výsledků | 57 |
| 3.9 | Závěr srovnávací části | 57 |
| 4 | Ovládání aplikace | 58 |
| 4.1 | Souřadná soustava | 59 |
| 4.2 | Panel hlavních ovládacích prvků | 60 |
| 4.2.1 | Pacient | 60 |
| 4.2.2 | Renderovací metoda | 61 |
| 4.2.3 | Povrch | 62 |
| 4.2.4 | Řezy | 62 |
| 4.2.5 | Stínování a barvy | 64 |
| 4.2.6 | Exportuj | 65 |
| 4.2.7 | Posun a rotace modelu | 68 |
| 4.2.8 | Barevné tabulky / Prahování | 68 |
| 4.2.9 | Orientace pacienta / modelu | 70 |
| 4.3 | Panel ovládacích prvků pro tvorbu animace | 72 |
| 4.3.1 | Tlačítko Akce | 73 |
| 4.4 | Nastavení barev a světla | 75 |
| 4.5 | Řezy modelem | 76 |
| 4.6 | Export do AVI | 78 |
| 4.7 | Export do VRML | 79 |
| 4.7.1 | Nastavení a průběh exportu | 79 |
| 4.7.2 | Prohlížení výsledku | 80 |
| 4.8 | Renderovací metody | 82 |
| 4.8.1 | Marching cubes | 83 |
| 4.8.2 | Marching tetrahedrons | 84 |
| 4.8.3 | Dividing cubes | 86 |
| 4.8.4 | Metody výpočtu normál povrchu | 89 |
| 4.9 | Formát dat ve schránce | 92 |
| 4.10 | Klávesové zkratky použitelné v modulu TomoAVI | 95 |
| 4.11 | Užitečné typy v aplikačním modulu TomoAVI | 97 |
| 5 | Závěr | 99 |

Seznam obrázků

| | |
|--|----|
| Obr. 1: TCS řezy..... | 10 |
| Obr. 2: MIP projekce a použitá paleta..... | 10 |
| Obr. 3: Povrchový rendering..... | 10 |
| Obr. 4: Možnosti aplikace..... | 11 |
| Obr. 5: Dělení trojúhelníku..... | 19 |
| Obr. 6: Cykly..... | 20 |
| Obr. 7: Jednoduchý řez modelem s vygenerovaným povrchem řezu..... | 20 |
| Obr. 8: Filtrování trojúhelníků řezu..... | 22 |
| Obr. 9: Odfiltrovaný víceřezový model..... | 22 |
| Obr. 10: Část textury řezu zasahující do matice intenzit..... | 23 |
| Obr. 11: Mapování souřadnic textury na souřadnice matice intenzit..... | 24 |
| Obr. 12: Vliv rozlišení textury na výsledek..... | 25 |
| Obr. 13: Bilineární interpolace..... | 26 |
| Obr. 14: Vrstvy v rámci urychlené oblasti..... | 27 |
| Obr. 15: OpenGL souřadnice textury..... | 28 |
| Obr. 16: Oblast renderingu trojúhelníku do NOPu..... | 31 |
| Obr. 17: Interpolace rovin v řezovém výrazu..... | 33 |
| Obr. 18: Řez studií zobrazující plnou kouli..... | 41 |
| Obr. 19: Řez studií zobrazující plný a dutý kvádr..... | 42 |
| Obr. 20: Řez studií zobrazující součet a rozdíl těles..... | 42 |
| Obr. 21: Inverzní intenzita tělesa..... | 42 |
| Obr. 22: NOPView..... | 43 |
| Obr. 23: Menu Soubor..... | 44 |
| Obr. 24: Menu Úpravy..... | 44 |
| Obr. 25: Inverze zobrazení NOPu..... | 44 |
| Obr. 26: Dialog pro zvýraznění odchylek..... | 45 |
| Obr. 27: Zvýraznění odchylek..... | 45 |
| Obr. 28: Menu Zobrazení..... | 45 |
| Obr. 29: Statistika..... | 46 |
| Obr. 30: Ukázky rozložení intenzit testovacích studií $R=12$ | 47 |
| Obr. 31: Ukázka vizualizace testovací studie Rovné povrchy s hranami..... | 48 |
| Obr. 32: MC povrch intenzity 1500, 3500, 6000..... | 48 |
| Obr. 33: DC povrch intenzity 3500 s dělením/bodem $1/10, 5/2, 10/1$ | 48 |
| Obr. 34: Odchytky normál povrchu duté koule $R=12 T=1$ | 53 |
| Obr. 35: Aditivní graf histogramů odchylek normál na duté kouli $R=12 T=1$ | 53 |
| Obr. 36: Aditivní graf histogramů odchylek normál na duté kouli $R=12 T=0,5$ | 54 |
| Obr. 37: MC - Odchytky normál rovných povrchů a hran..... | 55 |
| Obr. 38: MC - Aditivní graf histogramů odchylek normál rovných povrchů a hran..... | 55 |
| Obr. 39: DC - Odchytky normál rovných povrchů a hran..... | 56 |
| Obr. 40: DC - Aditivní graf histogramů odchylek normál rovných povrchů a hran..... | 56 |
| Obr. 41: Aplikační modul TomoAVI..... | 58 |
| Obr. 42: Souřadný systém 3D modelu..... | 59 |
| Obr. 43: Pole intenzit..... | 59 |
| Obr. 44: Panel hlavních ovládacích prvků..... | 60 |
| Obr. 45: Oblast pacient..... | 60 |
| Obr. 46: Oblast renderovací metoda..... | 61 |
| Obr. 47: Okno s informacemi o 3D modelu..... | 62 |
| Obr. 48: Oblast povrch..... | 62 |
| Obr. 49: Jednoduché řezy..... | 63 |
| Obr. 50: Pokročilé řezy..... | 63 |

| | |
|--|----|
| Obr. 51: Oblast stínování a barvy | 64 |
| Obr. 52: Ukázka Gouraud stínování | 64 |
| Obr. 53: Ukázka Flat stínování | 65 |
| Obr. 54: Menu pro výběr typu exportu | 65 |
| Obr. 55: Parametry exportu do JPEG | 65 |
| Obr. 56: Parametry exportu do BMP | 66 |
| Obr. 57: Parametry exportu do schránky | 66 |
| Obr. 58: Parametry exportu screenshotu do DB | 67 |
| Obr. 59: Parametry exportu do NOP | 67 |
| Obr. 60: Oblast Posun a rotace modelu | 68 |
| Obr. 61: Oblast barevné tabulky / prahování | 69 |
| Obr. 62: Prahování | 69 |
| Obr. 63: Oblast Orientace pacienta / modelu | 70 |
| Obr. 64: Ukázka změny pořadí snímků | 70 |
| Obr. 65: Ukázka změny orientace modelu | 71 |
| Obr. 66: Ukázka změny orientace modelu | 71 |
| Obr. 67: Ukázka změny orientace modelu | 71 |
| Obr. 68: Panel ovládacích prvků pro tvorbu animace | 72 |
| Obr. 69: Menu s operacemi pro vybrané klíčové body | 73 |
| Obr. 70: Okno pro nastavení barev a světla | 75 |
| Obr. 71: Jednoduché řezy | 76 |
| Obr. 72: Pokročilé řezy | 76 |
| Obr. 73: Okno pro export do AVI | 78 |
| Obr. 74: Okno pro nastavení parametrů exportu do VRML | 79 |
| Obr. 75: Okno zobrazující průběh exportu do VRML | 80 |
| Obr. 76: Srovnání VRML modelu s originálem | 81 |
| Obr. 77: Okno s nastavením Renderovací metody | 82 |
| Obr. 78: Metody detekce a zobrazení povrchu | 82 |
| Obr. 79: Metody výpočtu normál povrchu | 83 |
| Obr. 80: Marching cube | 84 |
| Obr. 81: Marching cube isoplocha | 84 |
| Obr. 82: Ukázka povrchu koule rekonstruované metodou Marching cubes | 84 |
| Obr. 83: Rozdělení oblasti mezi osmi voxely na čtyřstěny | 85 |
| Obr. 84: Marching tetrahedron | 85 |
| Obr. 85: Marching tetrahedron isoplocha | 86 |
| Obr. 86: Ukázka povrchu koule rekonstruované metodou Marching tetrahedrons | 86 |
| Obr. 87: Dividing cube (dělení 1,1,1) | 87 |
| Obr. 88: Dividing cube (dělení 1,1,1) | 87 |
| Obr. 89: Ukázka modelu koule (dělení 1,1,1) - velikost bodu 1 pixel | 87 |
| Obr. 90: Ukázka modelu koule (dělení 1,1,1) - velikost bodu 10 pixelů | 88 |
| Obr. 91: Dividing cube (dělení 2,2,2) | 88 |
| Obr. 92: Ukázka modelu koule (dělení 2,2,2) - velikost bodu 1 a 5 pixelů | 89 |
| Obr. 93: Upřesnění nastavení metody Dividing cubes | 89 |
| Obr. 94: Konvoluční matice GG3 | 90 |
| Obr. 95: Konvoluční matice GG6 | 90 |
| Obr. 96: Konvoluční matice GG26 | 91 |
| Obr. 97: Konvoluční matice SOB | 91 |

Seznam tabulek

| | |
|--|----|
| Tabulka 1 | 18 |
| Tabulka 2: Měření výsledků vylepšeného algoritmu (Duron 750MHz, matice 128x128xhl).. | 27 |
| Tabulka 3: Počty elementů povrchu | 49 |
| Tabulka 4: Geometrie povrchu koule | 50 |
| Tabulka 5: Geometrie vnějšího povrchu duté koule..... | 50 |
| Tabulka 6: Odchytky normál povrchu koule..... | 51 |
| Tabulka 7: Odchytky normál povrchu duté koule..... | 52 |
| Tabulka 8: Odchytky normál rovných povrchů a hran..... | 54 |

Název práce: Vizualizace tomografických dat z vyšetření mozku

Autor: Filip Vůjtěch

Katedra: Katedra softwarového inženýrství

Vedoucí diplomové práce: Dr. Ing. Daniel Janeba

e-mail vedoucího: daniel.janeba@email.cz

Abstrakt: Implementace 3D povrchových renderovacích algoritmů pro zobrazování tomografických dat mozku do uživatelsky přívětivého prostředí. Práce s daty ve formátu Interfile 3.3, ECAT 7.x. Zaměření na algoritmy umožňující plynulou manipulaci se zobrazovaným modelem jako jsou Marching cubes, Marching tetrahedrons a Dividing cubes. Tvorba animací ze sekvencí snímků se zvolenými parametry. Konverze výsledného modelu do VRML pro umístění na Webu. Implementace a porovnání metod pro výpočet normál povrchu modelu. Srovnání náročnosti a vhodnosti použití jednotlivých algoritmů. Porovnání kvality výsledku na různých typech dat.

Klíčová slova: počítačová tomografie, povrch, vizualizace, animace, VRML

Title: Visualization of tomographic data from brain investigation

Author: Filip Vůjtěch

Department: Department of Software Engineering

Supervisor: Dr. Ing. Daniel Janeba

Supervisor's e-mail address: daniel.janeba@email.cz

Abstract: User-friendly implementation of 3D surface rendering algorithms for visualization of a tomographic brain data. Working with data in Interfile 3.3, ECAT 7.x format. Intent on algorithms efficient for fast and fluent manipulation with model as are the Marching cubes, the Marching tetrahedrons and the Dividing cubes algorithm. Animation generation from sequence of parameter snapshots. Conversion of the result to VRML for Web presentation. Implementation and comparison of surface normal detection methods. Comparison of costiness and propriety of mentioned algorithms. Comparison of result quality on several types of data.

Keywords: computer tomography, surface, visualization, animation, VRML

1 Úvod

1.1 Vyšetření v nukleární medicíně

Pro zjištění stavu a funkce orgánů v těle pacienta se v nukleární medicíně využívají radiofarmaka. Radiofarmaka jsou látky, které jsou chemicky označeny radionuklidem. Radionuklid se v těle pacienta rozpadá a produkt rozpadové reakce je detekován vnějším zařízením. V místech, kde se nachází nejvíce radiofarmaka, dochází k nejsilnější emisi rozpadových produktů. Detekované záření tedy vypovídá o distribuci radiofarmaka v těle pacienta a tím i o funkci nebo metabolismu zkoumaného orgánu. Naměřené hodnoty jsou zpracovány softwarem dodávaným většinou přímo výrobcem snímacího zařízení do podoby dvojrozměrné nebo trojrozměrné pravoúhlé matice intenzit záření v prostoru. Výsledky takového měření se nazývají studií. Studie dělíme dále na statické a dynamické. Statické studie zobrazují naměřené intenzity záření v jednom okamžiku, zatímco dynamické studie obsahují hodnoty záření určité oblasti naměřené několikrát po sobě.

Není cílem této práce popisovat přesně postupy používané při vyšetřování pacientů v nukleární medicíně, omezíme se tedy na představu matice $I(i,j,k)$ hodnot intenzit záření v prostoru (či matice $I'(i,j)$ v případě dvojrozměrné studie).

1.2 Zobrazování v nukleární medicíně

Dynamickou studii lze ve zjednodušení pojmout jako několik statických studií. Ve skutečnosti se pro zkoumání dynamických studií využívají zcela odlišné postupy. Tato práce se omezuje na zpracování statických dat.

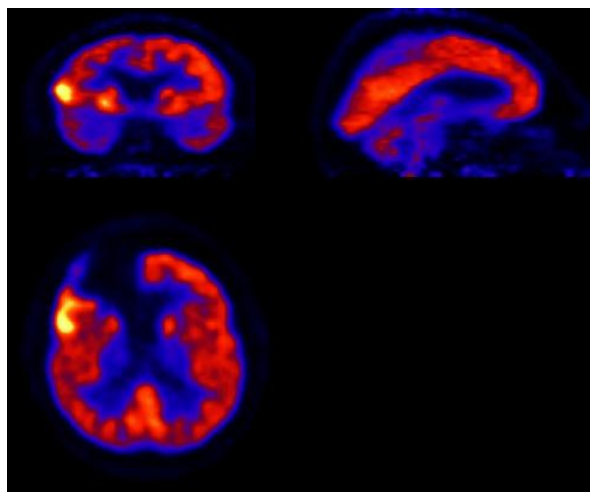
1.2.1 2D data

Zobrazení dvojrozměrné matice intenzit záření není problémem. K zobrazení těchto dat stačí namapovat hodnoty intenzit na jednotlivé barvy. K tomu používá většina medicínských zobrazovacích systémů takzvané palety. Výsledek lze pak přímo vykreslovat na výstup.

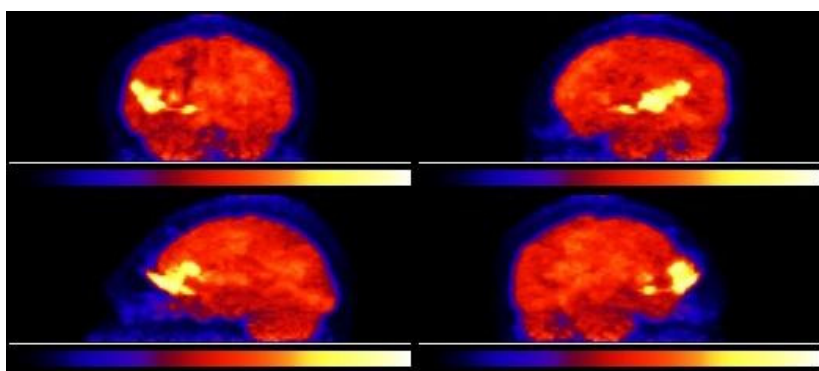
1.2.2 3D data

Trojrozměrná data nelze zobrazovat přímo jako data dvojrozměrná. Pro zobrazení těchto dat se používá několik metod. Je to zobrazování TCS řezů (Obr. 1). kdy se z trojrozměrné matice dat vybere v každém směru v určitém místě jeden plátek hodnot a ten se jako v případě 2D dat po aplikování palety barev přímo zobrazí. Zkratka TCS je složením počátečních písmen anglického překladu slov transverzální, koronální a sagitální popisujících v medicíně směr pohledu na pacienta. Tato metoda je klíčová pro popis nálezu u pacienta.

Dalším přístupem k zobrazení 3D dat je jejich přímá projekce na průmětnu (Obr. 2), kdy v daném směru pro každý pixel výsledného obrázku procházíme mřížku matice intenzit po pomyslném paprsku. Nejčastější metodou je zde metoda MIP (maximum intensity projection). Tato metoda si při průchodu 3D daty zapamatuje nejvyšší intenzitu záření, na kterou narazila, a tu použije po opaletování pro výslednou barvu daného pixelu výstupu.

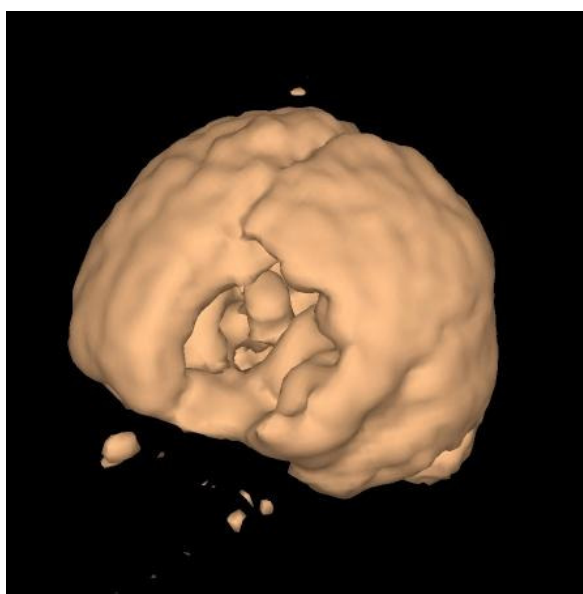


Obr. 1: TCS řezy



Obr. 2: MIP projekce a použitá paleta

Další používanou metodou zobrazení prostorových dat je takzvaný rendering (Obr. 3). Sem spadají prakticky všechny pokročilé metody zobrazování prostorových dat. Právě na rendering je zaměřena tato práce. Metody projekce a renderingu se používají pro získání lepší představy o vzájemné poloze orgánů a pro prezentování výsledků.

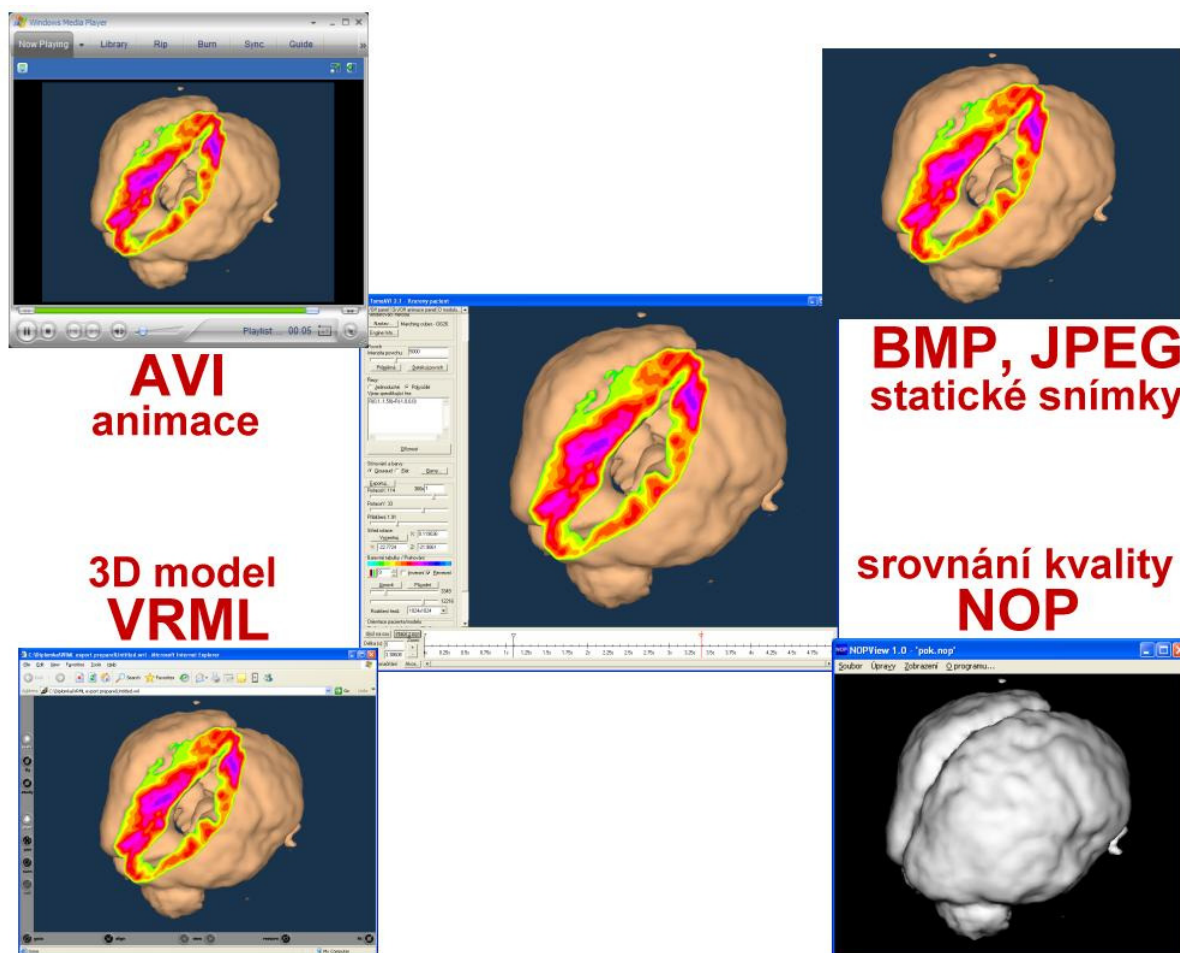


Obr. 3: Povrchový rendering

1.3 Cíle práce

Tato práce si kladla za cíl dvě věci. Jedním cílem bylo vytvoření uživatelsky přívětivé aplikace, která by umožňovala především jednoduchou a rychlou práci s tomografickými studiemi. Měla by nabízet několik povrchových zobrazovacích metod z oblasti renderingu a široké možnosti nastavení parametrů, stejně jako podporu exportu výsledků do různých formátů, tak aby byla tato aplikace co nejlépe využitelná pro tvorbu prezentací. Vedle možnosti exportovat statické obrázky by tu měla být možnost vytvářet i složité animace ze sekvencí snímků. Postupem času se dostavil nápad umožnit export výsledku přímo v 3D podobě. Jako vhodný formát se ukázal VRML.

Druhým cílem je pak porovnání kvality a náročnosti jednotlivých implementovaných zobrazovacích metod. Výsledky testů by měly být použitelné i mimo oblast zobrazování dat nukleární medicíny. Toto porovnání by tak mělo brát v úvahu charakter vstupních dat a tím dát odpovědi na otázku, jaké zobrazovací metody a jejich nastavení dávají na konkrétních typech dat nejlepší výsledky.



Obr. 4: Možnosti aplikace

2 Implementace

2.1 Cíle implementace

Jak již bylo řečeno, tato aplikace je určena (kromě možnosti porovnávat kvalitu jednotlivých zobrazovacích metod) především pro vytváření tomografických prezentací. Cílem bylo vytvořit kompaktní, snadno použitelnou aplikaci, která pro své použití nepotřebuje hlubší znalosti problematiky 3D zobrazování ani nukleární medicíny. Nutností v takovém případě je propracovaná interaktivní nápověda provázaná přímo s ovládním aplikace. Nápověda musí být dostatečně stručná, ale zároveň vysvětlující. Uživatel by se neměl dostat do stavu, kdy něčemu nerozumí a nemá k dispozici možnost odkázat se na podrobnější část nápovědy rozebírající danou problematiku. Aby mohly být výsledky aplikace co nejlépe využity, bylo potřeba implementovat množství výstupních formátů (Obr. 4). Vedle vytváření podkladů pro tomografické prezentace jako primární funkce aplikace bylo nutné implementovat podporu pro následné porovnávání kvality a vhodnosti použití jednotlivých implementovaných renderovacích metod.

2.2 Volba prostředí a vývojových nástrojů

Jako cílová platforma byl zvolen operační systém Microsoft Windows, kvůli jeho nejsilnějšímu zastoupení. Snahou bylo zachovat kompatibilitu s co největším množstvím verzí tohoto systému. Podporovány jsou, až na drobná omezení (Kap. 4), prakticky všechny 32bitové verze. Pro zobrazování 3D grafiky byla zvolena grafická knihovna OpenGL (později se ukázalo, že implementace OpenGL v jistých verzích systému Microsoft Windows není bezchybná, což trochu zkomplikovalo vývoj aplikace)¹. Vývojovým nástrojem byl zvolen Borland C++Builder verze 5.

Jako základ aplikace slouží systém ZobNuk původně vyvinutý pro Klinikou nukleární medicíny a endokrinologie Fakultní nemocnice Motol skupinou studentů MFF-UK pro výukový předmět softwarový projekt vedený Dr. Ing. Danielem Janebou. Aplikací, které se týká tato práce, je aplikační modul TomoAVI verze 2.1. Rozhodnutí nestavět samostatnou aplikaci bylo učiněno na základě skutečnosti, že systém ZobNuk již v praxi funguje a poskytuje stabilní základ pro vývoj medicínských aplikací. Základem aplikace TomoAVI 2.1 byl modul TomoAVI 1.0, který byl zastoupen v původní verzi systému ZobNuk.

Nápověda je založena na jazyku HTML a zkompileovaná do CHM souboru².

2.3 Vstupní data

Vstupem aplikace je soubor hodnot intenzit záření naměřených ve tkáni pacienta. V nukleární medicíně se používá řada formátů pro uložení těchto dat. K aplikaci TomoAVI se data dostanou již ve společném interním formátu systému ZobNuk. Vstupními daty je tedy trojrozměrné pole hodnot $I(x,y,z)$ udávající rozložení zkoumaného záření v prostoru v homogenní pravoúhlé mřížce. Jednotlivé vrstvy podél osy z se nazývají snímky a jednotlivé položky matice dat se nazývají voxely (Kap. 4.1). Voxel (oblast tvaru kvádru) nemusí mít stejnou velikost ve všech směrech. Spolu se vstupními daty jsou zadány i velikosti voxelu. Velikost voxelu závisí hlavně na možnostech a nastavení snímací aparatury. Jednotlivé hodnoty v poli jsou u medicínských dat nejčastěji typu 16bit unsigned integer. Pro vyšší přesnost a hlavně pro účely srovnávání použitých zobrazovacích algoritmů byla

¹ Clipping Problems with Generic Implementation of OpenGL for Windows 2000, XP Home and Professional

² HTML Help Compiled Help File

implementována podpora pro vstupní hodnoty s plovoucí desetinnou čárkou. Formát dat s plovoucí desetinnou čárkou se používá při zpracování testovacích dat (Kap. 3).

2.4 Zobrazovací metody

Jak již bylo zmíněno dříve, prostorová (nebo též objemová) data nukleární medicíny lze zobrazovat několika způsoby. Jsou to TCS řezy, přímá projekce a nakonec tzv. rendering. Renderingem rozumíme veškeré pokročilé zobrazovací metody využívající netriviální algoritmy. Takových metod je pochopitelně celá řada. Není smyslem práce implementovat veškeré známé metody. Byla vybrána sada metod, které vyhovují výchozím požadavkům aplikace. Tato aplikace byla navržena především pro interaktivní tvorbu materiálů k tomografickým prezentacím. Požadavkem na zobrazovací techniky je jejich použitelnost ve vyšším rozlišení výstupu a možnost real-time manipulace s výsledkem zobrazení (vyhnout se složitým výpočtům při změně úhlu pohledu).

Vybrané metody spadají do kategorie povrchových zobrazovacích algoritmů. Tyto algoritmy mají za cíl zobrazit povrch pomyslného tělesa určeného tou oblastí prostoru, kde naměřená intenzita záření je větší či rovna zadané hodnotě. Tuto hodnotu zadává uživatel. Takový povrch se nazývá isoplocha, neboť prochází prostorem v místech, kde hodnota záření je rovna zadané hodnotě a tudíž konstantní. Implementovány jsou metody Marching cubes, Marching tetrahedrons a Dividing cubes. Detailní rozbor těchto algoritmů není smyslem této práce, výklad je v tomto směru omezen na stručné seznámení s jejich principy. Je doporučeno seznámit se s nimi v Kap. 4.8.

Existuje mnoho jiných metod zobrazování, než jsou právě povrchové zobrazovací metody. Je to například použití poloprůhlednosti isoploch, či objemové zpracování, kdy se do výsledného obrazu započítává celý objem zkoumaného prostoru. Používají se například různé druhy sledování paprsku. Každá z těchto metod má většinou zcela odlišný výstup a nelze je nijak kvalitativně porovnávat. Spíše než o přesnost zobrazení jde u nich o názornost výsledku.

Povrchové metody bez použití poloprůhlednosti mají jednu velkou nevýhodu a tou je, že zobrazují pouze povrch a není zřejmé co se nachází pod ním. Obejít toto omezení lze například použitím řezů. Řezy mohou z výsledného tělesa, z něhož vidíme jen povrch, část oddělit a zobrazit, co je uvnitř. V aplikaci jsou implementovány řezy libovolnou kombinací rovin (Kap. 2.6). Model generovaný algoritmy Marching cubes a Marching tetrahedrons se skládá z trojúhelníků, zatímco model generovaný algoritmem Dividing cubes je složen z jednotlivých bodů. Přístup k výpočtu a zobrazování řezům je v těchto případech odlišný.

Aby mohl být takový povrch smysluplně zobrazen s pomocí osvětlení, je třeba znát příslušné normály povrchu v každém jeho bodě. Implementováno bylo několik metod výpočtu normál tohoto povrchu. Následně je z geometrie povrchu a jeho normál vygenerován za pomoci OpenGL výsledný obraz. OpenGL používá Gouraudovo stínování. Při Gouraudově stínování se osvětlení a výsledná barva povrchu vypočítá pouze ve vrcholech trojúhelníků a uvnitř trojúhelníků se barva povrchu počítá lineární interpolací barev mezi vrcholy. U bodového modelu je osvětlení a barva vypočítána pro každý bod povrchu a ten je následně zobrazen s danou velikostí na výstup.

Speciální variantou výsledného zobrazení takového povrchu je výstup do tzv. NOP formátu³. Tato funkce byla implementována kvůli možnosti porovnat kvalitu jednotlivých zobrazovacích algoritmů včetně vlivu jejich možných nastavení (Kap. 3). Pro generování NOP výstupu je implementován zvláštní renderovací algoritmus, který bere ohled na odlišný přístup k zobrazení (místo barvy je do souboru ukládána normála povrchu viditelného zobrazujícího se v této části průmětny) a větší přesnost. Větší přesnost je dosažena použitím Phongova stínování a zpětnou zobrazovací metodou (Kap. 2.7.3).

³ NOP – NOrmal Picture, formát navržený speciálně pro možnost porovnávání kvality zobrazovacích metod aplikace TomoAVI

2.4.1 Použité souřadné soustavy

Aplikace používá při zpracování dat dvě souřadné soustavy. Jednou souřadnou soustavou je přímo souřadná soustava prvků (voxelů) trojrozměrné matice intenzit I . Matice I má velikost $SizeX \times SizeY \times SizeZ$ prvků. Prvky jsou číslovány od nuly, tedy v ose x matice jsou voxely číslovány $0 \dots SizeX-1$ apod. Druhou souřadnou soustavou je soustava souřadnic výsledného 3D modelu. Tato soustava je volena tak, že střed matice I je zobrazen do počátku. Necht' m_x, m_y, m_z jsou rozměry voxelu ve směru osy x, y, z v milimetrech. Tyto hodnoty jsou součástí vstupních dat. Voxelu se souřadnicemi x, y, z v matici intenzit odpovídají souřadnice x_{mo}, y_{mo}, z_{mo} v rámci 3D modelu (trojúhelníkového či bodového), kde

$$x_{mo} = \left(x - \frac{SizeX - 1}{2} \right) m_x, \quad y_{mo} = \left(y - \frac{SizeY - 1}{2} \right) m_y, \quad a \quad z_{mo} = \left(z - \frac{SizeZ - 1}{2} \right) m_z.$$

2.5 Metody výpočtu normál povrchu

Aby mohl být povrch modelu zobrazen s pomocí osvětlení, je potřeba znát normálové vektory povrchu modelu. V případě trojúhelníkového modelu (Marching cubes, Marching tetrahedrons) je nutné znát normálové vektory ve vrcholech trojúhelníků. V případě bodového modelu (Dividing cubes) je potřeba znát normálové vektory v každém bodě reprezentujícím povrch. Model je zobrazován pomocí grafické knihovny OpenGL s použitím Gouraudova stínování. Pokud uživatel přepne do režimu Flat (Kap. 4.2.5), vykresluje se trojúhelník celý pouze jednou barvou, do osvětlení se započítává normála v těžišti trojúhelníku. Jiný Phongův model osvětlení je kvůli přesnosti použitý při exportu do formátu NOP (Kap. 3). Existuje více metod, jak odhadnout normálu v daném místě povrchu.

2.5.1 Metody odhadu normál založené na odhadu gradientu

Cílem je najít normálu povrchu modelu v daném místě. Povrch modelu byl generován tak, aby odděloval tu část prostoru, kde hodnota intenzity záření je vyšší než zadaná mez, od prostoru, kde hodnota intenzity je menší než zadaná mez. Geometrický průběh povrchu modelu je tedy přímo ovlivněn průběhem funkce intenzity. Z toho také vycházejí všechny metody založené na odhadu gradientu funkce velikosti intenzity v prostoru. Pokud by byla známa funkce gradientu $\Delta I(x, y, z)$ průběhu intenzity v celém prostoru, normála povrchu v bodě B by byla rovna $-\frac{\Delta I(B)}{|\Delta I(B)|}$. Záporné znaménko je zde z toho důvodu, normála povrchu

směřuje principiálně do oblasti prostoru s nižší intenzitou. Implementované metody výpočtu normál povrchu založené na odhadu gradientu pracují tak, že různými způsoby odhadnou gradient funkce intenzity ve středech voxelů. Z odhadu gradientů získají normálu povrchu a tu si ukládají do pomocné matice nazývané matice normál a označené N . N má stejnou velikost jako I . Normála povrchu v obecném bodě B se vypočítá jako normalizovaný vektor získaný trilineární interpolací nejbližších hodnot matice N .

Na odhad gradientu se používají různé metody. Všechny implementované metody na výpočet gradientů jsou založeny na konvoluci matice I s maticí $3 \times 3 \times 3$. Všechny použité matice jsou vlastně upravené hranové detektory založené na první derivaci. V následujícím textu je u každé metody uvedena příslušná konvoluční matice $3 \times 3 \times 3$. Je podél osy z rozdělena do tří matic 3×3 , kde u každé matice 3×3 směřuje osa x doprava a osa y nahoru. Uvedené matice jsou pro odhad x -ové složky gradientu. Konvoluční matice pro odhad y -ové a z -ové složky gradientu jsou obdobné.

Implementovány jsou čtyři metody tohoto druhu. Jsou to:

1. Gray-level gradient (3 susedi). Dále GG3.

$$\text{Konvoluční matice GG3: } \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$z-1 \qquad \qquad z \qquad \qquad z+1$

2. Gray-level gradient (6 susedů). Dále GG6.

$$\text{Konvoluční matice GG6: } \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$z-1 \qquad \qquad z \qquad \qquad z+1$

3. Gray-level gradient (26 susedů, Zucker-Hummel). Dále GG26.

$$\text{Konvoluční matice GG26: } \begin{pmatrix} -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \end{pmatrix}, \begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ -1 & 0 & 1 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix}, \begin{pmatrix} -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \end{pmatrix}$$

$z-1 \qquad \qquad z \qquad \qquad z+1$

4. Sobel 3D operátor. Dále SOB.

$$\text{Konvoluční matice SOB: } \begin{pmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} -3 & 0 & 3 \\ -6 & 0 & 6 \\ -3 & 0 & 3 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{pmatrix}$$

$z-1 \qquad \qquad z \qquad \qquad z+1$

Po vypočítání i y-ové a z-ové složky se získá vektor určující směr odhadovaného gradientu. Skutečnost, že konvolucí nezískáme korektní velikost gradientu, není na závadu, protože výsledek se bude normalizovat na jednotkovou velikost. Gradienty byly počítány pro souřadnou soustavu matice I , zatímco normály budou použity v souřadné soustavě 3D modelu. Je proto nutné jednotlivé x-ové, y-ové a z-ové složky gradientu vydělit velikostí voxelu v daném směru, tedy hodnotami m_x, m_y a m_z . Výsledná normála je pak normalizovanou inverzí upraveného gradientu v daném místě.

2.5.2 Adaptivní metody odhadu normál založené na odhadu gradientu

Kromě metod uvedených v Kap. 2.5.1 jsou implementovány ještě adaptivní verze těchto metod. Jsou to:

1. Adaptivní Gray-level gradient (6 susedů). Dále AGG6.
2. Adaptivní Gray-level gradient (26 susedů). Dále AGG26.
3. Adaptivní Sobel 3D operátor. Dále ASOB.

Pro tenké objekty, které se svou tloušťkou blíží velikosti voxelu, je 6 či 26 okolních hodnot voxelů pro odhad gradientu příliš velká oblast. Adaptivní metody se snaží tuto nevýhodu eliminovat. AGG6 vychází z GG6, AGG26 vychází z GG26 a ASOB vychází ze SOB. V případě GG6 se odhad x-ové složky gradientu počítal v bodě x,y,z matice I jako $I(x+1,y,z) - I(x-1,y,z)$. V případě AGG6 se postupuje následujícím způsobem: Jestliže $I(x,y,z)$ je větší (menší) než obě sousední hodnoty $I(x-1,y,z)$ a $I(x+1,y,z)$, pak se x-ová složka gradientu odhaduje z hodnoty $I(x,y,z)$ a z menší (větší) z obou sousedních hodnot

$I(x-1, y, z)$ a $I(x+1, y, z)$. V opačném případě se hodnota odhadne klasicky jako $I(x+1, y, z) - I(x-1, y, z)$. Tuto ideu lze rozšířit i na AGG26 a ASOB. Rozdíl je v tom, že zde se uvažují i úhlopříčné přechody přes středový voxel.

2.5.3 Geometrické metody odhadu normál

Zcela jinou kategorií metod jsou odhady normál přímo z geometrie detekovaného povrchu. Tyto metody jsou implementovány pouze pro trojúhelníkový model. Zde se nepoužívá žádná pomocná matice normál získaných z odhadu gradientu funkce intenzity. Místo toho se normály vrcholů trojúhelníků počítají přímo z polohy trojúhelníků. Necht' P je množina všech trojúhelníků povrchu modelu. Necht' P^v je množina trojúhelníků povrchu modelu, kde pro každý trojúhelník $t \in P^v$ platí $v \in t$. Normálou trojúhelníku (značí se $n(t)$) se myslí jednotkový vektor kolmý k ploše trojúhelníku a směřující ve směru jeho přední strany (Kap. 2.6). Pak pro každý trojúhelník $t \in P$ povrchu modelu a každý jeho vrchol $v \in t$ se

vypočítá normála povrchu modelu ve vrcholu v jako
$$\frac{\sum_{t \in P^v} n(t)}{\left| \sum_{t \in P^v} n(t) \right|}.$$

Algoritmus počítající normály povrchu tímto způsobem pracuje tak, že pole vrcholů trojúhelníků povrchu modelu nejprve seřídí podle souřadnic. Práci má usnadněnou díky předtřídění vrcholů již samotným průchodem Marching cubes či Marching tetrahedrons algoritmem maticí I . Pak stačí procházet seříděné pole a z normál trojúhelníků sousedících shodných vrcholů vypočítat normály.

Právě popsaná metoda je implementována v aplikaci pod názvem Shape Normal. Druhou implementovanou metodou výpočtu normál z geometrie povrchu je metoda velice podobná předešlé a nese název Adaptive Shape Normal. Nevýhodou metody Shape Normal bylo to, že průměrovala všechny normály okolních trojúhelníků se stejnou váhou. Úvaha je následující. Každý trojúhelník povrchu má svou vlastní normálu kolmou k sobě samému. Při detekci povrchu použitými algoritmy vznikají v podstatě náhodně různě velké trojúhelníky. Jak bude která část povrchu rozdělena na trojúhelníky závisí na zvolené intenzitě a průchodu této intenzity mřížkou voxelů. Velký trojúhelník představuje zjednodušenou velkou část skutečné isoplochy. Malý trojúhelník představuje zjednodušenou malou část skutečné isoplochy. Normála velkého i malého trojúhelníku představuje průměrnou normálu povrchu na odpovídající skutečné isoploše. Pokud předpokládáme, že isoplocha nebyla příliš zakřivená, normála trojúhelníku odpovídá zhruba normále skutečné isoplochy nad těžištěm trojúhelníku. Když se pak počítá normála vrcholu, dá se očekávat, že normály trojúhelníků, které mají těžiště blízko daného vrcholu, spíše odpovídají skutečné normále isoplochy v místě vrcholu. Adaptivní metoda tedy pracuje tak, že počítá vážený průměr normál okolních trojúhelníků, kde váhou jsou převrácené hodnoty vzdáleností vrcholu od těžiště daného trojúhelníka. Výsledná normála vrcholu v je
$$\sum_{t \in P^v} \left(\frac{n(t)}{|v - \text{těžiště}(t)|} \right).$$
 Vektor je ještě třeba normalizovat na jednotkovou velikost.

2.6 Řezy

Základní informace o funkci řezů a jejich zadávání jsou v Kap. 4.5. Zde se výklad omezí na použitou implementaci. Bylo řečeno, že zde existují dva přístupy k reprezentaci povrchu modelu podle použitého algoritmu. U algoritmů Marching cubes a Marching tetrahedrons je povrch reprezentován pomocí trojúhelníků. V případě algoritmu Dividing

cubes je reprezentován pomocí bodů v prostoru. Výpočet i zobrazení řezů je v každém z těchto případů odlišné. Je třeba zmínit, že trojúhelníčky jsou narozdíl od bodů orientované v prostoru. Mají přední a zadní stranu. OpenGL vykreslí trojúhelník pouze v případě, že je na průmětně k pozorovateli otočen přední stranou. Poznává se to tak, že vrcholy v pořadí, v jakém jsou zadány, se zobrazují kolem středu trojúhelníku proti směru hodinových ručiček. Zachování této orientace je důležité pro správné zobrazení.

2.6.1 Jednoduchý řez trojúhelníkovým modelem

Nechť $t_i : i = 1, \dots, n$ jsou všechny trojúhelníky původního povrchu modelu bez řezu (trojúhelníky vygenerované algoritmem Marching cubes či Marching tetrahedrons). Trojúhelníkem se myslí trojice vrcholů v prostoru, označme je t_i^1, t_i^2, t_i^3 . Nechť r je rovina řezu zadaná rovnicí $ax + by + cz + d = 0$. Část prostoru, kde $ax + by + cz + d < 0$ se nazývá záporný poloprostor roviny r , značí se r^- . Část prostoru, kde $ax + by + cz + d > 0$ se nazývá kladný poloprostor roviny r , značí se r^+ . Cílem je odstranit ty z trojúhelníků povrchu, které leží v r^- . Problém nastává u trojúhelníků, kde jeden vrchol leží v r^- a jiný v r^+ neleží. V tomto případě je potřeba takový trojúhelník rozdělit na více trojúhelníků, které buď celé spadají do $r^- \cup r$, nebo celé do $r^+ \cup r$. Při rozdělení je nutné zachovat orientaci původního trojúhelníku a také uzavřenost povrchu (Kap. 2.6.2). Pouhým rozdělením trojúhelníků a odstraněním těch, které leží v r^- se získá sice část povrchu modelu ležícího v $r^+ \cup r$, ale pro zobrazení to nestačí neboť schází povrch v místě řezu modelem. Tento povrch je potřeba dopočítat.

2.6.2 Uzavřenost povrchu modelu bez řezů

Trojúhelníkový model $t_i : i = 1, \dots, n$ vygenerovaný kteroukoliv metodou musí splňovat jeden zásadní požadavek. Jeho povrch musí být uzavřený. Povrch je uzavřený, když neexistuje i, j takové, že existuje cesta od přední strany t_i k zadní straně t_j , která by neprocházela vnitřkem nebo hranou některého t_k ($1 \leq k \leq n$). Jinými slovy trojúhelníky povrchu modelu $t_i : i = 1, \dots, n$ rozdělují prostor na podprostory, kde každý takový podprostor sousedí s každým trojúhelníkem nejvýše z jedné strany trojúhelníku a to vždy ze stejné (buď se všemi z přední, nebo se všemi ze zadní strany). Podprostor sousedící s trojúhelníky z přední strany se nazývá vnější podprostor a podprostor sousedící s trojúhelníky ze zadní strany se nazývá vnitřní podprostor. Uzavřenost je důležitá pro korektní zobrazování povrchu modelu. Mnohem důležitější je ale pro dopočítávání trojúhelníků v místě řezu modelem.

Algoritmy Marching cubes a Marching tetrahedrons takové povrchy generují, ale pouze za předpokladu, že hodnoty na okrajích matice intenzit I jsou menší než hodnota intenzity pro detekci povrchu. Jestliže má matice I rozměry o, p, q , pak musí platit $\forall (i = 1 \dots o, j = 1 \dots p, k = 1 \dots q) ((i = 1 \vee i = o \vee j = 1 \vee j = p \vee k = 1 \vee k = q) \Rightarrow I(i, j, k) < I_{\text{povrch}})$. Z toho důvodu se k původní matici dat přidávají takové okraje uměle. Dále se předpokládá, že matice I takové okraje má.

Také musíme brát v úvahu skutečnost, že operace s čísly s plovoucí desetinnou čárkou mají ve výpočetní technice svá specifika. Zejména nepřesnost a tudíž i jiné výsledky v případě odlišně provedených výpočtů, které by sice měly dát stejný výsledek, ale vinou chyb při zaokrouhlování dají výsledek jiný. Například ne vždy platí $a + (b + c) = (a + b) + c$. Program musí být napsán tak, aby při generování povrchu s tímto počítal. Vrcholy, které mají mít z principu algoritmu stejné souřadnice, je musejí po výpočtu opravdu stejné mít. V opačném případě by následující algoritmus nemohl správně fungovat.

2.6.3 Dopočítání trojúhelníků povrchu v místě řezu

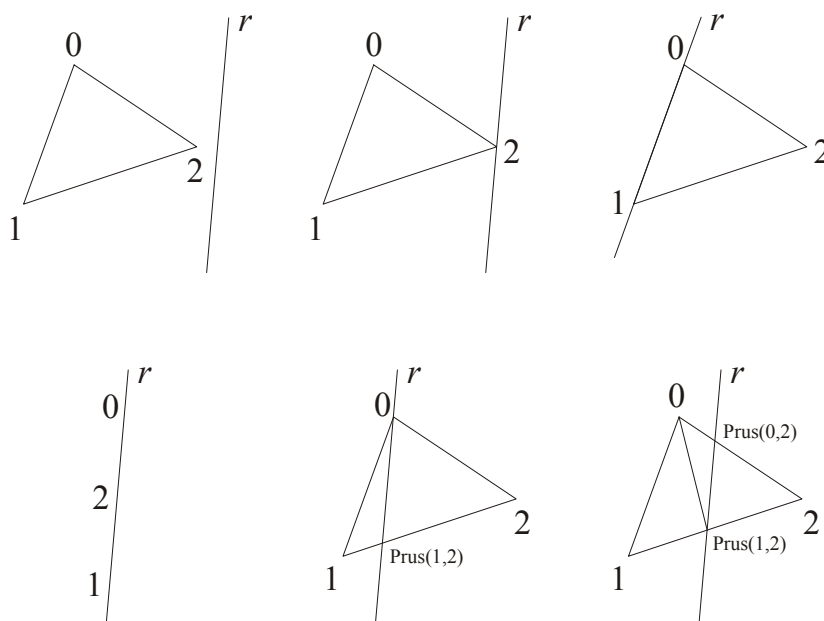
V místě, kde modelem prochází rovina řezu r , je třeba vygenerovat trojúhelníky reprezentující samotný řez. Jako vstup algoritmu máme pole trojúhelníků povrchu modelu $t_i : i = 1, \dots, n$, které ještě nejsou rozděleny rovinou řezu r . Algoritmus funguje pro každý t_i tak, že klasifikuje každý jeho vrchol buď jako záporný, rovinný, nebo kladný. Záporný je ten vrchol, který leží v r^- , kladný je ten, který leží v r^+ a rovinný je ten, který leží v r . Existuje tak 27 možností polohy trojúhelníku vůči rovině řezu. V závislosti na této poloze se provede určitý výpočet. To popisuje následující tabulka. V tabulce je první, druhý a třetí vrchol trojúhelníku označen 0, 1 a 2. Ve sloupečku 'v2v1v0' je kód klasifikace jednotlivých vrcholů v pořadí 2,1,0. Kladný vrchol je označen 10, záporný 01 a rovinný 00.

| | v2v1v0 | akce |
|----|--------|----------------------------|
| 1 | 000000 | nic |
| 2 | 000001 | Hrana(1,2) |
| 3 | 000010 | nic |
| 4 | 000100 | Hrana(2,0) |
| 5 | 000101 | nic |
| 6 | 000110 | Hrana(2,Prus(0,1)) |
| 7 | 001000 | nic |
| 8 | 001001 | Hrana(Prus(1,0),2) |
| 9 | 001010 | nic |
| 10 | 010000 | Hrana(0,1) |
| 11 | 010001 | nic |
| 12 | 010010 | Hrana(Prus(0,2),1) |
| 13 | 010100 | nic |
| 14 | 010101 | nic |
| 15 | 010110 | Hrana(Prus(0,2),Prus(0,1)) |
| 16 | 011000 | Hrana(0,Prus(1,2)) |
| 17 | 011001 | Hrana(Prus(1,0),Prus(1,2)) |
| 18 | 011010 | Hrana(Prus(0,2),Prus(1,2)) |
| 19 | 100000 | nic |
| 20 | 100001 | Hrana(1,Prus(2,0)) |
| 21 | 100010 | nic |
| 22 | 100100 | Hrana(Prus(2,1),0) |
| 23 | 100101 | Hrana(Prus(2,1),Prus(2,0)) |
| 24 | 100110 | Hrana(Prus(2,1),Prus(0,1)) |
| 25 | 101000 | nic |
| 26 | 101001 | Hrana(Prus(1,0),Prus(2,0)) |
| 27 | 101010 | nic |

Tabulka 1

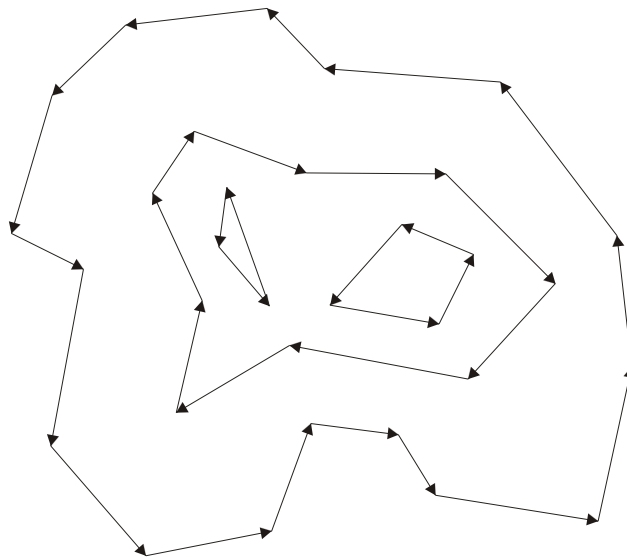
Algoritmus tedy vezme každý trojúhelník ze vstupu, spočítá jeho konfiguraci a podle tabulky provádí následující akce. Je třeba nejprve říci, co je to hrana. Hrana h je orientovaná úsečka v prostoru daná dvěma koncovými body. Funkce $Hrana(A,B)$ přidá do pomocného pole hran hranu vedoucí z bodu A do bodu B. $Hrana(0,1)$ například znamená hranu vedoucí z vrcholu 0 daného trojúhelníku do vrcholu 1 daného trojúhelníku. Je to tedy přímo hrana

tohoto trojúhelníku. Funkce $\text{Prus}(A,B)$ vypočítá souřadnice průsečíku úsečky $|AB|$ s rovinou řezu r . Akce jsou navrženy tak, že do pomocného pole vyplňují úsečky, ve kterých se trojúhelníky zasahující do r^- protínají s rovinou r . Při implementaci se opět musí dbát na to, aby výpočet příslušných průsečíků funkcí Prus byl komutativní, tedy $\text{Prus}(A,B)=\text{Prus}(B,A)$. Nebo dávat na vstup funkce Prus vrcholy tak, že vždy první argument je z r^- a druhý z r^+ . Orientace přidávané hrany je volena v závislosti na orientaci daného trojúhelníku a to tak, že při pohledu z r^- na rovinu r směřuje hrana vždy proti směru hodinových ručiček kolem oblasti pod zadní stranou trojúhelníku. Zpracovávané trojúhelníky, které protínají rovinu r , se nakonec ještě rozdělí rovinou r na 2 až 3 menší trojúhelníky (Obr. 5). Trojúhelníky z r^- lze pak snadno odfiltrout (Kap 2.6.5). Pro zjištění, zda je trojúhelník v r^- , se dosadí těžiště trojúhelníku do rovnice roviny r .

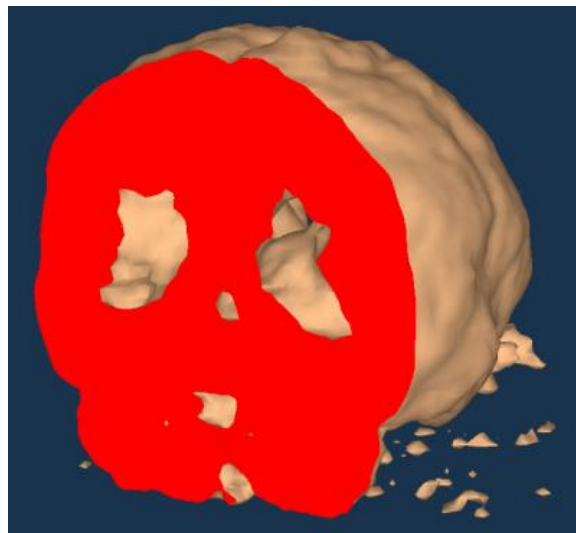


Obr. 5: Dělení trojúhelníku

Protože byl původní trojúhelníkový model uzavřený a orientace hran byla volena vhodně podle orientací trojúhelníků, lze výsledné vygenerované hrany pospojovat do uzavřených cyklů (Obr. 6). Tyto cykly se neprotínají a platí, že orientace cyklů při pohledu z r^- se při průchodu cykly od vnějších k vnitřním pravidelně střídá v pořadí proti směru hodinových ručiček, ve směru hodinových ručiček. Vnější cykly jsou liché, nejbližší vnitřní jsou cykly sudé atd. Triangulací oblastí mezi lichým a následujícím sudým cyklem je hledaný trojúhelníkový povrch řezu. Normálovým vektorem vrcholů trojúhelníků řezu je normála roviny r směřující do r^- . Ukázka povrchu řezu je na Obr. 7 v levé části vyznačena červeně.



Obr. 6: Cykly



Obr. 7: Jednoduchý řez modelem s vygenerovaným povrchem řezu

2.6.4 Více řezů modelem

Uživatel má možnost zadat více řezů modelem zároveň. Základem zadání řezových rovin a vztahů mezi nimi je řezový výraz (Kap 4.5). $R(a,b,c,d)$ tedy označuje rovinu $ax + by + cz + d = 0$. Jedna rovina se může v řezovém výrazu (dále RV) vyskytovat vícekrát a to v různých podobách zadání, případně i v opačné orientaci. Například rovina $R(1,1,1,0)$ je ekvivalentní s rovinou $R(2,2,2,0)$ a opačná (ale ne ekvivalentní) k rovině $R(-3,-3,-3,0)$. Zpracování modelu s více řezy probíhá tak, že roviny v RV rozdělíme do takzvaných tříd rovin. Každá třída rovin má jednu rovinu jako svého reprezentanta (nemusí to být nutně rovina z RV). Třída rovin s reprezentantem r se značí T^r . Třída T^r obsahuje všechny roviny z RV, které jsou ekvivalentní nebo opačné k r . Tříd rovin RV je nejvýše tolik jako rovin v RV.

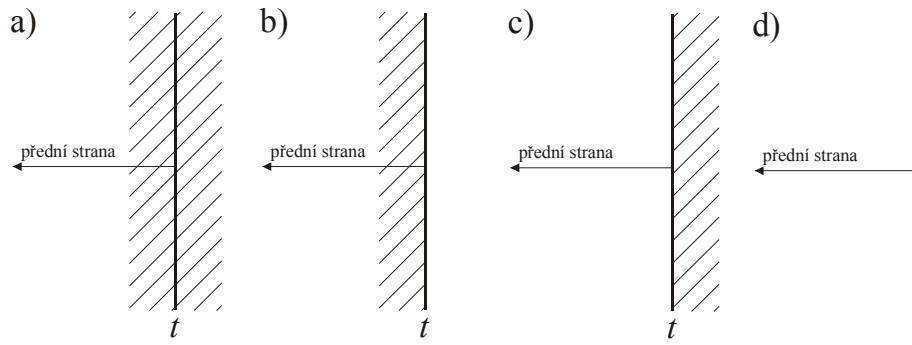
Pro každou třídu T^r z RV se provede jednoduchý řez povrchem modelu rovinou r zatím bez odfiltrování povrchu ležícího v r^- . Jednak se trojúhelníky protínající roviny z RV tímto rozdělí a zadruhé se tím získají trojúhelníky povrchu řezu pro každou třídu RV. Trojúhelníky povrchu řezu třídy T^r dále rozdělíme postupně všemi rovinami q , že T^q je třída

z RV a $T^q \neq T^r$. Nyní jsou všechny trojúhelníky povrchu modelu i jeho řezů rozřezány. Stačí podle řezového výrazu odfiltrvat nepotřebné z nich.

2.6.5 Odfiltrování trojúhelníků řezovým výrazem

Odfiltrováním se rozumí vyřazení určitých trojúhelníků z výsledného zobrazení. U trojúhelníků samotného povrchu modelu (tedy ne povrchu řezů) je situace snadná. Necht' r je rovina $R(a,b,c,d)$, tedy rovina určená rovnicí $ax + by + cz + d = 0$. Pak funkce H je definována jako $H(r, x, y, z) = ax + by + cz + d$. Vyhodnocení RV pro konkrétní bod $B = (x, y, z)$ prostoru probíhá tak, že výskyt zadání každé roviny v RV (označme ji r) je nahrazen hodnotou *false* v případě, že $H(r, B) < 0$, nebo je nahrazen hodnotou *true* v případě, že $H(r, B) \geq 0$. Hodnota *true* respektive *false* říká, že bod B leží respektive neleží v $r^+ \cup r^-$. RV je pak vyhodnocen jako logický výraz, kde $+$ je ekvivalentní s \vee a $*$ s \wedge . Hodnotu takového výrazu označíme $RV(B)$. Trojúhelník t povrchu se odfiltruje řezovým výrazem RV, když $RV(\text{těžiště}(t)) = \textit{false}$.

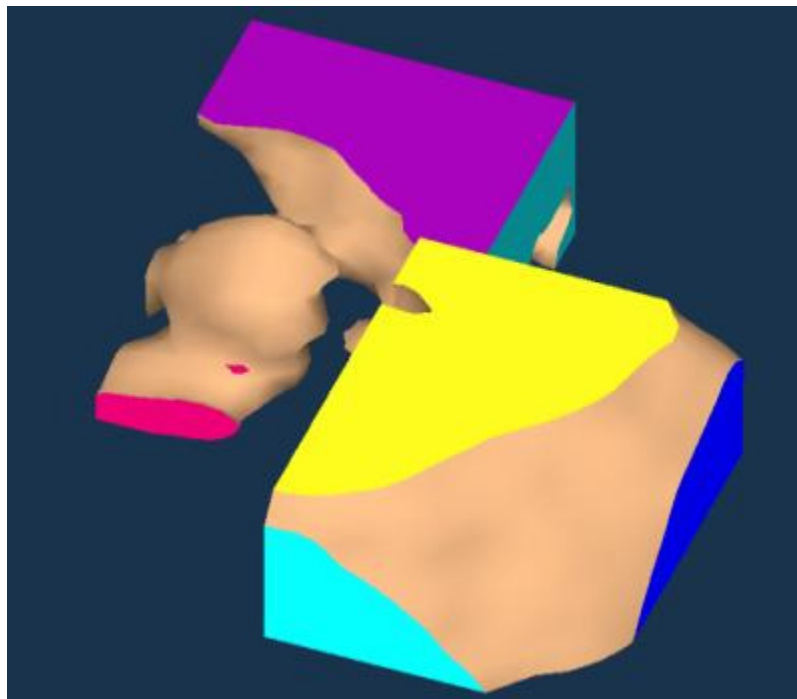
V případě filtrování trojúhelníků povrchu řezu je situace složitější. T^r je třída rovin z RV. Necht' $t_i^r : i = 1, \dots, n_r$ jsou neodfiltrované ale rozřezané trojúhelníky řezu této třídy. Zde nestačí vypočítat pro každý trojúhelník t_i^r hodnotu $RV(\text{těžiště}(t_i^r))$ a podle toho rozhodnout. U původního povrchu modelu narozdíl od povrchu řezů se vyhodnocením RV zjišťovalo, zda daný trojúhelník leží v oblasti prostoru, kterou je třeba odfiltrvat. Pokud ano, je takový trojúhelník vyřazen ze zobrazení. U trojúhelníků řezů je situace jiná. U trojúhelníků $t_i^r : i = 1, \dots, n_r$ je potřeba rozhodnout o tom, zda daný trojúhelník odděluje oblast, která se odfiltruje, od oblasti, která se neodfiltruje. Nebude odfiltrován pouze ten trojúhelník, který svou přední viditelnou stranou směřuje do odfiltrované oblasti a zadní stranou do oblasti neodfiltrované (Obr. 8 případ c). Každá třída T^r má podtřídy T^{r+} a T^{r-} . Třída T^{r+} obsahuje třídy z T^r , které jsou ekvivalentní s r . Třída T^{r-} obsahuje třídy z T^r , které jsou opačné k r . Necht' $t_i^{r+} : i = 1, \dots, n_r$ jsou neodfiltrované ale rozřezané trojúhelníky řezu podtřídy T^{r+} a $t_i^{r-} : i = 1, \dots, n_r$ jsou neodfiltrované ale rozřezané trojúhelníky řezu podtřídy T^{r-} . Rozdělení na podtřídy a dvě sady trojúhelníků u každé třídy je provedeno kvůli tomu, že na každý řez se lze dívat ze dvou stran. Jednak ze strany r^- , nebo ze strany r^+ . Povrch řezu by mohl být vidět i ze strany r^+ , pokud v třídě T^r je i rovina opačná k r . Trojúhelníky $t_i^{r+} : i = 1, \dots, n_r$ jsou shodné s $t_i^r : i = 1, \dots, n_r$, protože reprezentují povrch řezu při pohledu ze strany r^- a pro tento pohled byly $t_i^r : i = 1, \dots, n_r$ původně vygenerovány. Trojúhelníky $t_i^{r-} : i = 1, \dots, n_r$ jsou až na opačnou orientaci shodné také s $t_i^r : i = 1, \dots, n_r$, protože reprezentují povrch řezu při pohledu ze strany r^+ . V případě, že třída T^{r-} neobsahuje žádné roviny, není nutné $t_i^{r-} : i = 1, \dots, n_r$ uvažovat.



Obr. 8: Filtrování trojúhelníků řezu

Odfiltrování neviditelných trojúhelníků z $t_i^{r^+} : i = 1, \dots, n_r$ řezovým výrazem RV se provede následujícím způsobem. Funkce $RV^{r^+}(B)$ vychází z funkce $RV(B)$, která byla definována na začátku této kapitoly. Rozdíl je v tom, že funkce H se nahradí funkcí H^{r^+} . Nechť q je libovolná rovina a B libovolný bod prostoru. Funkce H^{r^+} je definována následovně. Když $q \notin T^r$, pak $H^{r^+}(q, B) = H(q, B)$. Když $q \in T^{r^+}$, pak $H^{r^+}(q, B) = 1$. Když $q \in T^{r^-}$, pak $H^{r^+}(q, B) = -1$. Funkce $RV^{r^-}(B)$ je definována podobně, také vychází z funkce $RV(B)$. Funkce H se nahradí funkcí H^{r^-} . Nechť q je libovolná rovina a B libovolný bod prostoru. Funkce H^{r^-} je definována následovně. Když $q \notin T^r$, pak $H^{r^-}(q, B) = H(q, B)$. Když $q \in T^{r^+}$, pak $H^{r^-}(q, B) = -1$. Když $q \in T^{r^-}$, pak $H^{r^-}(q, B) = 1$. Tímto postupem se vlastně zjistí průběh funkce RV pro okolí bodu B ve směru r^+ a r^- .

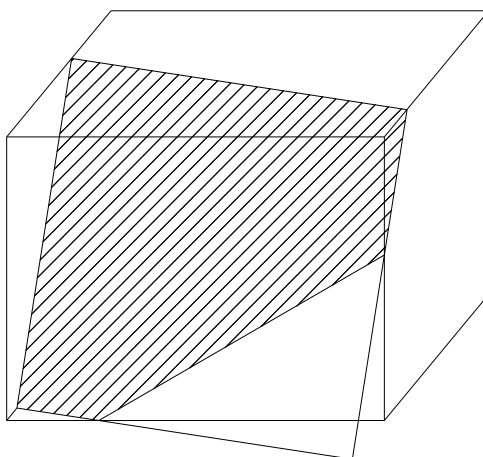
Nechť t je libovolný trojúhelník z $t_i^{r^+} : i = 1, \dots, n_r$ a RV je řezový výraz. Trojúhelník t nebude odfiltrován, pokud $RV^{r^+}(\text{těžiště}(t)) = \text{true}$ a zároveň $RV^{r^-}(\text{těžiště}(t)) = \text{false}$. Tyto dvě podmínky říkají, že ve směru přední strany trojúhelníka t je oblast, která se odfiltruje a ve směru zadní strany je oblast, která se neodfiltruje. Je to tedy případ c) z Obr. 8. Pro úplnost nechť t je libovolný trojúhelník z $t_i^{r^-} : i = 1, \dots, n_r$ a RV je řezový výraz. Trojúhelník t nebude odfiltrován, pokud $RV^{r^+}(\text{těžiště}(t)) = \text{false}$ a zároveň $RV^{r^-}(\text{těžiště}(t)) = \text{true}$. Ukázka výsledku je na Obr. 9.



Obr. 9: Odfiltrovaný víceřezový model

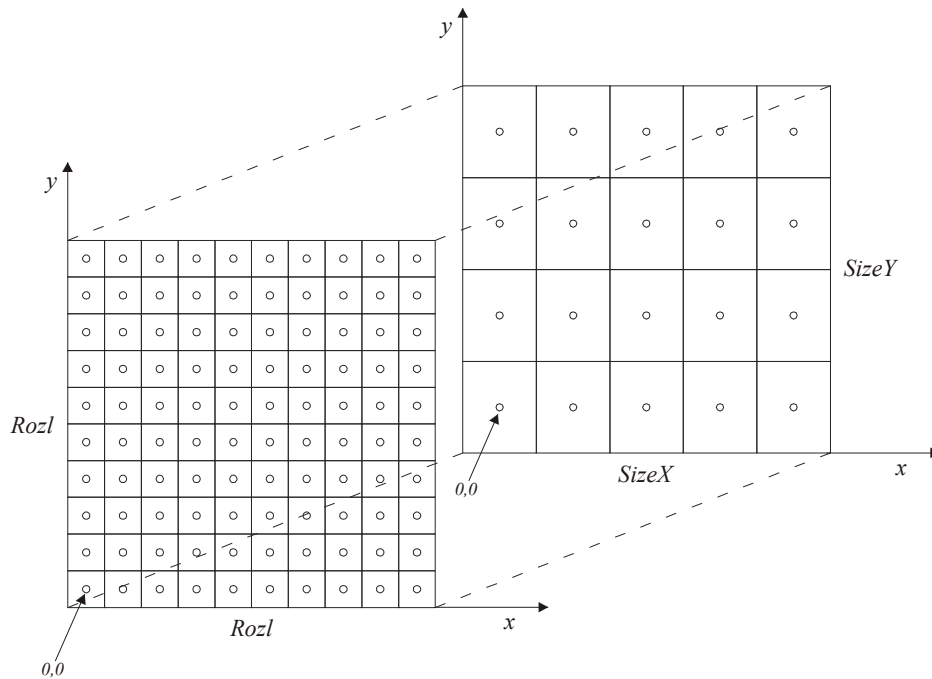
2.6.6 Textura řezu

K tomu, aby měl uživatel představu, co se nachází pod povrchem, pouhé řezy modelem nestačí. Bylo třeba znázornit rozložení intenzit záření v místě řezu. Vhodným prostředkem je použití textur. Pro každý řez určitou třídou se vypočítá jedna textura znázorňující opaletované intenzity v místě, kudy tento řez prochází maticí intenzit I . Protože zadání rovin může být zcela obecné, rovina řezu nemusí být kolmá k žádné ose soustavy souřadnic. Algoritmus pracuje tak, že vybere jednu osu soustavy souřadnic a počítá texturu řezu z pohledu podél této osy. Necht' m_x, m_y, m_z jsou rozměry voxelu ve směru osy x, y, z v milimetrech a r se zadáním $R(a, b, c, d)$ je rovina řezu. Algoritmus vybere osu x v případě, že $a.m_x \geq b.m_y$ a zároveň $a.m_x \geq c.m_z$. Algoritmus vybere osu y v případě, že $b.m_y \geq a.m_x$ a zároveň $b.m_y \geq c.m_z$. Jinak vybere osu z . Tím zajistí, že ve směru osy pohledu pro výpočet textury je změna souřadnic v rámci voxelu při pohybu podél roviny řezu co nejmenší. Tím je dosaženo nejvyšší přesnosti zobrazení. Tato vlastnost má také podstatný vliv na rychlost později popsaného vylepšeného algoritmu pro zrychlený výpočet textury řezu. Na Obr. 10 je vidět ukázka polohy textury řezu v matici dat a její část uvnitř matice. Směr pohledu pro výpočet textury je zepředu.



Obr. 10: Část textury řezu zasahující do matice intenzit

Předpokládejme, že směr pohledu pro výpočet textury řezu je ve směru osy z . Velikost počítané textury je $Rozl * Rozl$ pixelů a rozměry matice intenzit (dále jen I) v ose x, y a z jsou $SizeX, SizeY$ a $SizeZ$. Souřadnice textury se mapují na souřadnice os x, y matice I (Obr. 11). V případě, že osa pohledu pro výpočet textury není z (jako na obrázku), ale například x , pak se souřadnice textury mapují na osy y, z matice I . Podobně v případě pohledu v ose y . Princip algoritmu bude popsán pro případ pohledu ve směru osy z .



Obr. 11: Mapování souřadnic textury na souřadnice matice intenzit

Označme souřadnice v textuře jako x_T a y_T a souřadnice v matici I jako x_M , y_M a z_M . Souřadnice x_T se mapují lineární funkcí s posunem na x_M a souřadnice y_T se mapují lineární funkcí s posunem na y_M . Souřadnice z_M se dopočítávají z rovnice řezu. Jednoduchá verze algoritmu namapuje bod 0,0 textury do souřadnic $\frac{SizeX}{2 \cdot Rozl} - \frac{1}{2}$, $\frac{SizeY}{2 \cdot Rozl} - \frac{1}{2}$.

Další souřadnice bodů textury v matici se získají přičítáním kroku $\frac{dx_M}{dx_T} = \frac{SizeX}{Rozl}$ ve směru x a

kroku $\frac{dy_M}{dy_T} = \frac{SizeY}{Rozl}$ ve směru y. Nyní je potřeba podle rovnice roviny r řezu dopočítat také z-

ovou souřadnici. Je pravidlem, že roviny jsou zadávány v posunuté souřadné soustavě mající střed ve středu matice intenzit. Tento posun je vyjádřen hodnotami $ShiftX$, $ShiftY$ a $ShiftZ$.

$ShiftX = \frac{SizeX}{2} - \frac{1}{2}$, $ShiftY = \frac{SizeY}{2} - \frac{1}{2}$ a $ShiftZ = \frac{SizeZ}{2} - \frac{1}{2}$. Z-ová souřadnice z_M v I při zadání roviny $R(a,b,c,d)$ je pro x-ovou a y-ovou souřadnici x_M a y_M v I :

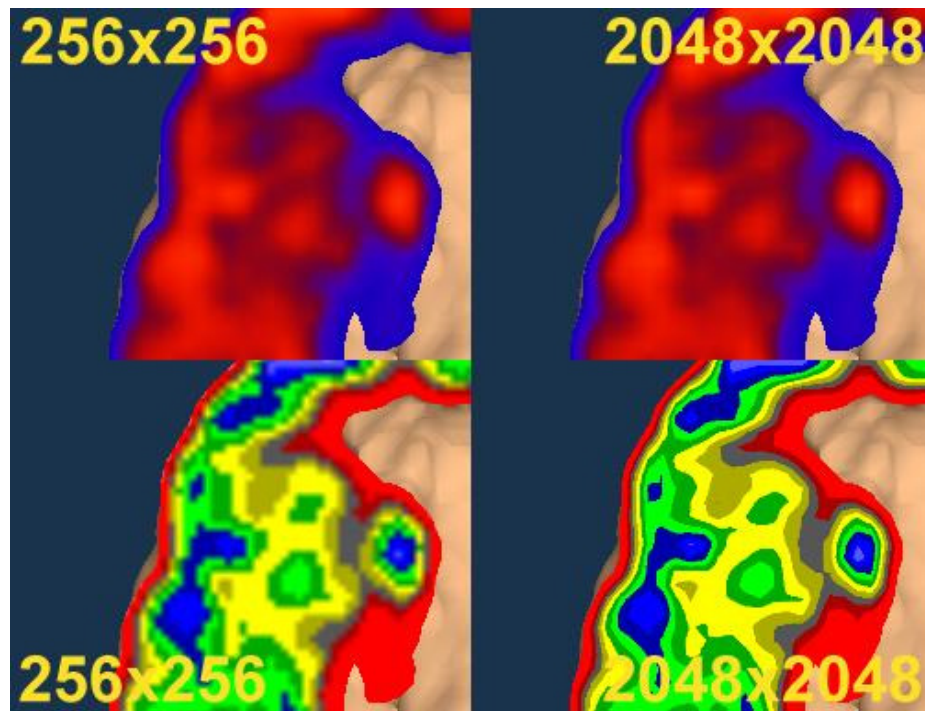
$$z_M = -\frac{m_x \cdot a \cdot (x_M - ShiftX) + m_y \cdot b \cdot (y_M - ShiftY) + d}{m_z \cdot c} + ShiftZ. \text{ Tímto způsobem se vypočítá z-}$$

ová souřadnice pro bod 0,0 textury. Další z-ové souřadnice bodů textury se vypočítají

přičítáním kroku $\frac{dz_M}{dx_T} = \frac{m_x \cdot a}{m_z \cdot c} \cdot \frac{dx_M}{dx_T}$ ve směru x a kroku $\frac{dz_M}{dy_T} = \frac{m_y \cdot b}{m_z \cdot c} \cdot \frac{dy_M}{dy_T}$ ve směru y.

Hodnota intenzity v pixelu (x_T, y_T) textury se získá trilineární interpolací nejbližších hodnot intenzit z matice I v místě (x_M, y_M, z_M) . Nevýhoda tohoto algoritmu je, že pro každý pixel textury se provádí přístup k osmi hodnotám matice I .

Pokud má použitá paleta ostré přechody, je pro dosažení dobrého výsledku potřeba vysoké rozlišení textury. Jednoduchý algoritmus pro výpočet textury je zde zbytečně pomalý, protože kvůli vypočítání intenzity jednoho pixelu textury algoritmus přistupuje vždy k osmi hodnotám z matice I . Rozlišení textury je většinou mnohem větší než rozlišení (velikost) matice I . Dochází tak mnohokrát za sebou k přístupu ke stejným prvkům matice I .



Obr. 12: Vliv rozlišení textury na výsledek

2.6.7 Vylepšený algoritmus pro výpočet textury

Pokud má textura vyšší rozlišení než matice I , dochází k tomu, že více sousedních pixelů textury se zobrazuje mezi stejné prvky matice I . Toho lze využít k urychlení výpočtu textury. Ideální by bylo, kdyby se hodnoty z matice I nečetli pro takové sousední pixely vícekrát a kdyby se trilineární interpolace nepočítala pro každý pixel. Nejprve je potřeba vyřešit interpolaci mezi čtyřmi hodnotami. $I^{bili}(x, y)$ je funkce bilineární interpolace intenzity v rovině mezi čtyřmi hodnotami k, l, m, n v pravoúhlé mřížce, kde x, y udávají polohu hledané hodnoty (Obr. 13). $I^{bili}(x, y) = (x.n + (1-x).m).y + (x.l + (1-x).k).(1-y)$ Funkce $I^{bili}(x, y)$ je pro pevné x lineární v proměnné y a naopak. Intenzita lze pro pixely textury, které se zobrazí v rozsahu od fx, fy do tx, ty (Obr. 13) počítat diferenčním algoritmem. Necht' $i_0 = I^{bili}(fx, fy)$, $i_1 = I^{bili}(tx, fy)$, $i_2 = I^{bili}(fx, ty)$ a $i_3 = I^{bili}(tx, ty)$, jsou pomocné hodnoty, ze kterých vychází diferenční algoritmus pro vyplnění oblasti $(fx...tx) \times (fy...ty)$. Hodnoty nx a ny udávají rozměr mřížky vyhodnocovaných intenzit uvnitř čtveřice hodnot k, l, m, n . Na Obr. 13 to je například 6 a 4. Necht' $StartI$, dI , $dStartI$ a ddI jsou pomocné proměnné. $StartI$ udává počáteční intenzitu pro výpočet jednoho řádku hodnot, dI udává změnu intenzity mezi dvěma sousedními body při průchodu řádku, $dStartI$ udává změnu počáteční intenzity při přechodu na další řádek a ddI udává změnu dI při přechodu na další řádek. $StartI = i_0$, $dI = \frac{i_1 - i_0}{nx - 1}$,

$$dStartI = \frac{i_2 - i_0}{ny - 1} \text{ a } ddI = \frac{\frac{i_3 - i_2}{nx - 1} - dI}{ny - 1}.$$

Diferenční algoritmus pak vyplňuje pole hodnot $nx * ny$

a pracuje takto:

```

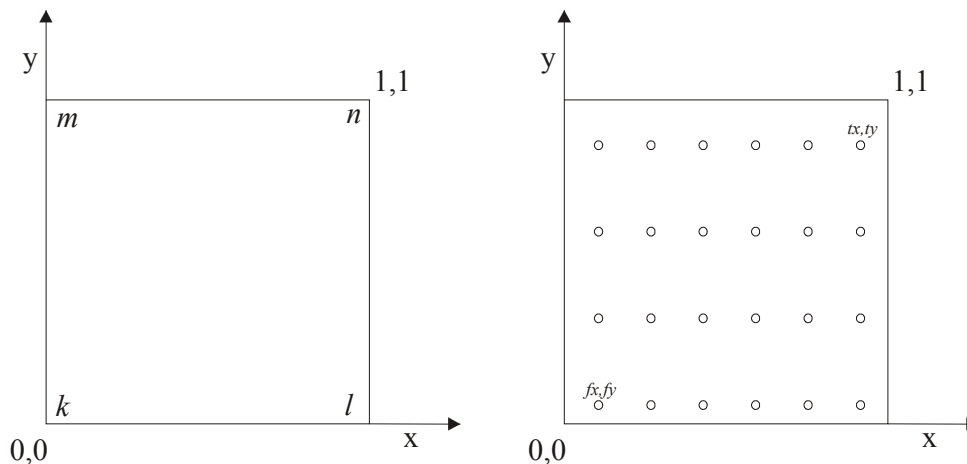
for y:=1 to ny
  begin
    intenzita:=StartI
    for x:=1 to nx

```

```

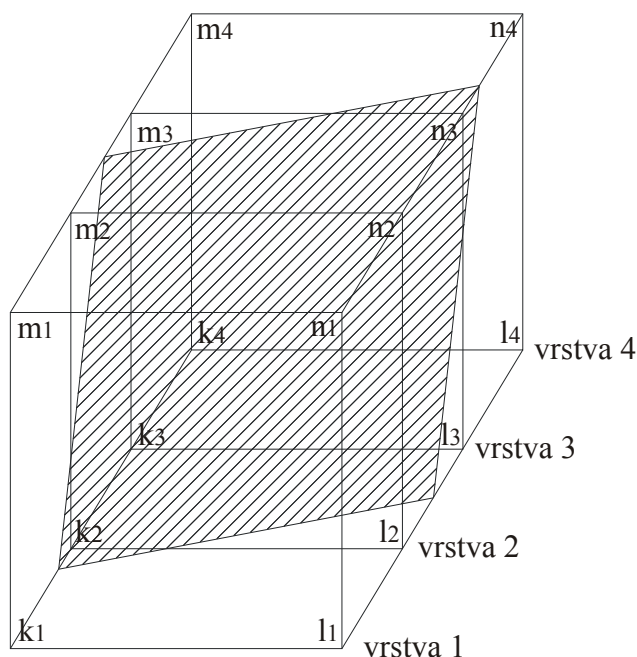
begin
  Intenzita(x,y) := intenzita
  intenzita := intenzita + dI
end
StartI := StartI + dStartI
dI := dI + ddI
end

```



Obr. 13: Bilineární interpolace

Algoritmus v předešlém odstavci pracuje pouze dvojrozměrně a lze tedy použít jen ve speciálním případě, že rovina řezu je kolmá k ose z a zároveň prochází přímo středy voxelů některé z vrstev matice I . V obecné případě může rovina řezu zasáhnout až do tří sousedních dvojic vrstev voxelů. Nemůže však zasáhnout do více než tří vrstev, protože směr pohledu pro výpočet textury byl vhodně zvolen podle roviny řezu (Kap. 2.6.6). Jestliže tedy rovina zasahuje do více vrstev počítají se diferenčním algoritmem popsáním výše hodnoty mezi více čtveřicemi voxelů a výsledná hodnota se následně lineárně interpoluje podél z-ové souřadnice (Obr. 14). Výrazné urychlení výpočtu je znát i v případě, že rovina zasahuje do více vrstev voxelů viz. měření zrychlení (Tabulka 2). V nejlepším případě se urychlení výpočtu blíží zhruba hodnotě 6,3x, což je výborná hodnota a značně se tím zvýší komfort při používání aplikace.



Obr. 14: Vrstvy v rámci urychlené oblasti

| Rozlišení textury | Čas výpočtu jednoduchým algoritmem (v ms) volba roviny bez vlivu | Čas výpočtu vylepšeným algoritmem (v ms) vhodně zvolená rovina | Poměrné zrychlení | Poměrné zpomalení | Čas výpočtu vylepšeným algoritmem (v ms) nehodně zvolená rovina | Poměrné zrychlení | Poměrné zpomalení |
|-------------------|---|---|-------------------|-------------------|--|-------------------|-------------------|
| 4096x4096 | 15032 | 2394 | 6,3 | | 2764 | 5,4 | |
| 2048x2048 | 3765 | 600 | 6,3 | | 721 | 5,2 | |
| 1024x1024 | 942 | 180 | 5,2 | | 230 | 4,1 | |
| 512x512 | 233 | 70 | 3,3 | | 90 | 2,6 | |
| 256x256 | 60 | 40 | 1,5 | | 57 | 1,1 | |
| 128x128 | 15 | 32 | | 2,1 | 40 | | 2,7 |
| 64x64 | 4 | 29 | | 7,3 | 50 | | 12,5 |
| 32x32 | 1 | 28 | | 28,0 | 57 | | 57,0 |

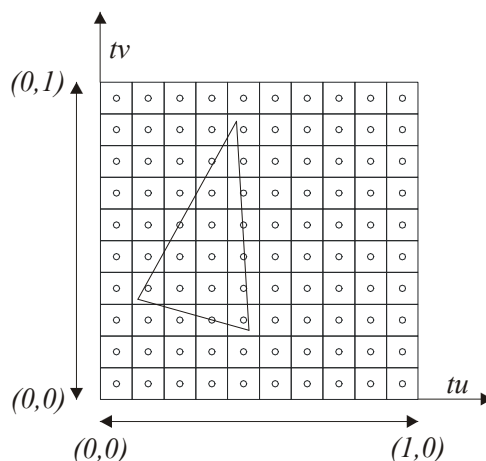
Tabulka 2: Měření výsledků vylepšeného algoritmu (Duron 750MHz, matice 128x128xhl)

2.6.8 Namapování textury na trojúhelníky řezu

Nechť $t_i^{r^+} : i = 1, \dots, n_r^+$ jsou trojúhelníky řezu třídou rovin T^r viditelné z poloprostoru r^- , které zbyly po odfiltrování pomocí RV (Kap. 2.6.5). Nechť $t_i^{r^-} : i = 1, \dots, n_r^-$ jsou trojúhelníky řezu třídou rovin T^r viditelné z poloprostoru r^+ , které zbyly po odfiltrování pomocí RV. Je potřeba namapovat souřadnice příslušné textury řezu na trojúhelníky $t_i^{r^+} : i = 1, \dots, n_r^+$ a $t_i^{r^-} : i = 1, \dots, n_r^-$. V OpenGL (DirectX apod.) mají textury pro mapování na povrch grafických primitiv souřadnice nezávislé na rozlišení textury a to v rozsahu $(0,1) \times (0,1)$ (Obr. 15). Každý trojúhelník řezu si nese informaci o poloze jeho jednotlivých vrcholů v textuře. Při renderování výsledného obrazu je na povrchu trojúhelníku zobrazována barva textury v příslušném místě. Pro obě sady trojúhelníků je mapování shodné. Nechť t je zpracováváný trojúhelník, v je jeho libovolný vrchol a x, y, z jsou souřadnice vrcholu v v prostoru a tu, tv jsou hledané souřadnice vrcholu v v textuře. Mapování závisí na směru, pro který byla textura vypočítána (Kap. 2.6.6). Nechť to je osa z (pro ostatní případy je výpočet

obdobný). Pak souřadnice vrcholu v v textuře jsou $tu = \frac{x}{m_x \cdot \text{SizeX}} + \frac{1}{2}$ a $tv = \frac{y}{m_y \cdot \text{SizeY}} + \frac{1}{2}$.

Takto se vypočítají souřadnice pro všechny vrcholy trojúhelníků $t_i^{r+} : i = 1, \dots, n_r^+$ a $t_i^{r-} : i = 1, \dots, n_r^-$. Přičtení jedné poloviny ve vzorci je z důvodu toho, že 3D model má střed souřadné soustavy ve středu matice intenzit I . Pro výsledné zobrazování se použijí textury, kde jednotlivé hodnoty intenzit byly namapovány na RGB barvy pomocí dané palety.



Obr. 15: OpenGL souřadnice textury

2.6.9 Řezy bodovým modelem

Zatím se práce zabývala trojúhelníkovým modelem generovaným algoritmy Marching cubes a Marching tetrahedrons. Algoritmus Dividing cubes ale generuje bodový model (Kap. 4.8.3). Algoritmy v předchozím textu se dají ve zjednodušené verzi použít i na tento model. Při filtrování samotného povrchu modelu (ne povrchu řezů) pomocí RV není třeba body narozdíl od trojúhelníků rozdělovat ani počítat těžiště. Stačí pouze pro každý bod B povrchu vypočítat $RV(B)$. Pokud $RV(B)=false$, bod B se vyřadí ze zobrazování.

U trojúhelníkového modelu se povrch řezů rekonstruoval pomocí trojúhelníků. U bodového modelu to jsou opět body. U trojúhelníkového modelu se řezy nepočítaly algoritmy Marching cubes nebo Marching tetrahedrons, ale byl vyvinut speciální algoritmus pro jejich generování z nerozřezaného modelu. Stejně je to i v případě řezů bodovým modelem. Každý řez je tvořen množinou bodů vygenerovaných speciálním algoritmem a následně odfiltrovaných RV. Je potřeba generovat body řezu dostatečně hustě v prostoru, aby nevznikaly mezery při jejich zobrazování. Musí být alespoň tak hustě, jako je generuje algoritmus Dividing cubes. Toho lze docílit následovně. Zvolí se směr pohledu pro výpočet řezu stejně jako v Kap. 2.6.6. Pro další výklad je to osa z . Ostatní případy jsou obdobné. Uvnitř matice intenzit I generujeme body se souřadnicemi x,y,z tak, že x a y jsou generovány stejně hustě jako body povrchu metodou Dividing cubes a zbylá souřadnice z se dopočítává z rovnice roviny řezu. Z matice I se získá z nejbližších voxelů pomocí trilineární interpolace hodnota intenzity v bodě x,y,z . Pokud je tato hodnota větší nebo rovna intenzitě zvolené pro detekci povrchu modelu, je tento bod zařazen mezi neodfiltrované body povrchu řezu daného řezu.

Ve výsledku ovšem nemusí být zobrazeny všechny takto vygenerované body řezů. Je opět potřeba body řezů odfiltrovat pomocí RV. Postup je obdobný jako u trojúhelníkového modelu (Kap. 2.6.5). Nechť r je rovina řezu modelem a T^r je její třída. Protože bod nemá narozdíl od trojúhelníku orientaci v prostoru a je viditelný ze všech stran, nemá řez v bodovém modelu dvě sady bodů, ale pouze jednu. Nechť B je libovolný bod povrchu řezu rovinou r , pak B není odfiltrován a je tedy zobrazen ve výsledku, pokud platí $RV^{r+}(B)=true$ a

zároveň $RV^{r-}(B)=false$ nebo platí $RV^{r+}(B)=false$ a zároveň $RV^{r-}(B)=true$. Tato podmínka říká, že v okolí bodu B je rovinou r určeno rozhraní mezi prostorem, který má být odfiltrován výrazem RV , a mezi prostorem, který nemá být odfiltrován výrazem RV . V bodě B je tedy povrch řezu rovinou r .

Rozložení hodnot intenzit na řezu bylo zobrazováno v trojúhelníkovém modelu pomocí textur. U bodového modelu se toto řeší přímým obarvením jednotlivých bodů řezu. Pro každý bod B řezu se zjistí příslušná hodnota intenzity trilineární interpolací hodnot z matice I . Pomocí dané palety se převede hodnota intenzity na RGB barvu a ta je uložena k bodu B pro následné zobrazování.

2.7 Export

Aplikace TomoAVI je především určená pro tvorbu tomografických prezentací. Z toho důvodu je vyžadována možnost exportu výsledků do co nejširšího spektra formátů (Kap 4.2.6).

2.7.1 Export statických snímků

V aplikaci je implementována podpora exportu statického 2D snímku, jak jej vidí v dané chvíli uživatel. Je možné snímek uložit do grafického formátu BMP, JPEG nebo je možné snímek zkopírovat do schránky systému Windows, odkud lze vložit do jiné aplikace. U exportu statického snímku je potřeba nastavit rozlišení výsledného obrázku. Maximální velikost je omezena možnostmi dané implementace OpenGL. Pro vytvoření výsledného snímku v zadaném rozlišení je nutné vytvořit nový renderovací kontext⁴ OpenGL v příslušném rozlišení v paměti mimo obrazovku a poté provést samotné zobrazení existujícího modelu v tomto kontextu.

2.7.2 Export animace ze série statických snímků

Aplikace umožňuje ukládat definované parametry modelu a jeho zobrazení na časovou osu do takzvaných klíčových bodů (Kap. 2.7). Následně je možné vygenerovat posloupnost statických snímků podle parametrů v klíčových bodech na časové ose a tyto snímky uložit jako animaci do AVI souboru (Kap. 4.3).

2.7.3 Export statických snímků do NOP

NOP (Normal Picture) je speciální formát vyvinutý pro uložení (Kap. 3.4) a následné porovnání výsledků jednotlivých algoritmů pro detekci a zobrazení povrchu v druhé části této práce (Kap. 3). Aplikace podporuje export povrchu modelu bez řezů. Export do NOP funguje podobně jako obyčejný export statických snímků. Transformuje 3D model do dvojrozměrné podoby vyrenderováním do 2D obrázku. Rozdíl je v tom, co se ukládá do pixelů. Standardní export statického snímku ukládá do každého pixelu RGB barvu osvětleného povrchu modelu v daném místě, případně barvu pozadí. Místo RGB barvy ukládá NOP export v každém pixelu normálu povrchu daného místa modelu.

Zatímco normální statické snímky jsou renderovány s pomocí OpenGL, pro generování NOP snímků je použita zvláštní renderovací metoda. Necht' matice T je maticí transformace souřadnic modelu do souřadnic průmětny. Pixely jsou v obrázku uloženy po řádcích zleva doprava (osa x obrázku) a řádky postupně zdola nahoru (osa y obrázku). V souřadné soustavě průmětny osa x odpovídá ose x výsledného obrázku, osa y odpovídá ose

⁴ Není cílem této práce popisovat používání OpenGL

y výsledného obrázku. Pixel (x,y) má souřadnice v soustavě průmětny také (x,y) . Osa z souřadné soustavy průmětny směřuje k pozorovateli. Z-ové hodnoty se používají k práci se Z-bufferem. Výsledný NOP se vytvoří vyrenderováním všech trojúhelníků povrchu modelu. Přestože dochází při výstupu do obrázku k zaokrouhlování přesné polohy v souřadnicích průmětny na jednotlivé pixely obrázku, je nutné kvůli použití NOP formátu pro porovnávání výsledků zachovat co nejvyšší přesnost zobrazení. Pro export trojúhelníkového modelu do NOP byla použita speciální přesná metoda zobrazení.

Nechť t je renderovaný trojúhelník a v_1, v_2 a v_3 jsou jeho vrcholy transformované maticí T do souřadnic průmětny. $v_1 = (x_1, y_1, z_1)$, $v_2 = (x_2, y_2, z_2)$ a $v_3 = (x_3, y_3, z_3)$. Trojúhelník v průmětně reprezentují všechny body (x,y) , kde platí:

$$x = a.x_1 + b.x_2 + c.x_3$$

$$y = a.y_1 + b.y_2 + c.y_3$$

$$1 = a + b + c$$

$$0 \leq a \leq 1$$

$$0 \leq b \leq 1$$

$$0 \leq c \leq 1$$

Nechť matice M je matice soustavy rovnic prvních tří řádků a tedy platí:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = M \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Algoritmus funguje následovně. Nechť $fromx, fromy, tox, toy$ jsou celá čísla:

$$fromx = \lfloor \min(x_1, x_2, x_3) \rfloor$$

$$fromy = \lfloor \min(y_1, y_2, y_3) \rfloor$$

$$tox = \lceil \max(x_1, x_2, x_3) \rceil$$

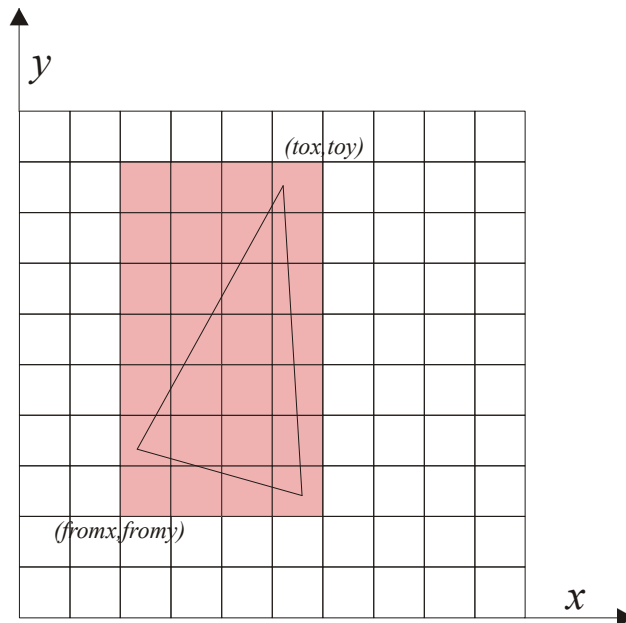
$$toy = \lceil \max(y_1, y_2, y_3) \rceil$$

Projdou se všechny pixely v oblasti určené $fromx...tox, fromy...toy$ (Obr. 16) a zjistí se jejich poloha v rámci trojúhelníku takto:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}$$

Pixel x_p, y_p je uvnitř trojúhelníku t , kdy platí $0 \leq a \leq 1$, $0 \leq b \leq 1$ a $0 \leq c \leq 1$. Z-ová souřadnice pixelu p je $z_p = a.z_1 + b.z_2 + c.z_3$. Hodnota z_p se porovná se Z-bufferem. Nechť normála povrchu ve vrcholech v_1, v_2 a v_3 je n_1, n_2 a n_3 . Pak normála povrchu zobrazeného v místě pixelu p je $n_p = a.n_1 + b.n_2 + c.n_3$.

V případě bodového modelu je postup jednoduchý a není zde podrobně uveden. Oblast, do které bod v obrázku zasahuje, se vyplní s pomocí Z-bufferu příslušnou hodnotou normály povrchu v tomto bodě.



Obr. 16: Oblast renderingu trojúhelníku do NOPu

2.7.4 Export 3D modelu do VRML

Zajímavou možností je export modelu, přesně tak jak jej vidí v dané chvíli uživatel včetně řezů, do jazyka VRML (Kap. 4.7). Výsledek je možné umístit na Web a zobrazit pomocí webového prohlížeče s integrovaným VRML klientem. Tato možnost je dostupná pouze při zobrazení trojúhelníkového modelu. Požitá verze VRML neumožňuje rozumné uložení bodového modelu a především neumožňuje nastavení velikosti bodu nutné pro zobrazení bodového modelu, tato možnost tedy není k dispozici.

Výsledek je exportován a uložen do struktury souborů, která vypadá následovně:

```
filename.wrl
.\filename_files
  shape.wrl
  tex0.jpg
  .
  .
  .
  texN.jpg
```

V souboru `filename.wrl` je uložen kód VRML, který popisuje scénu, ve které se model zobrazí, a veškeré geometrické transformace modelu. Nastavuje se zde barva pozadí, poloha pozorovatele a úhel pohledu, osvětlení scény a také natočení modelu k pozorovateli, změny orientace modelu apod. Je zde také definován uzel `SphereSensor`, který je propojen s uzlem `Transform` transformace polohy tělesa modelu na scéně. `SphereSensor` převádí pohyby myši při stisknutí tlačítka na rotační matici popisující otáčení modelu umístěného ve VRML kódu pod tímto uzlem. To umožňuje uživateli pomocí myši otáčet 3D modelem na scéně.

Samotný 3D model, který se zobrazuje na obrazovce, je pak uložen v souboru `shape.wrl`. Hlavní povrch je uložen pomocí VRML uzlu `IndexedFaceSet`⁵. `IndexedFaceSet` umožňuje definovat 3D povrch složený z mnohoúhelníků a zároveň definovat normály povrchu a mapování textur. Povrchy řezů jsou uloženy také pomocí `IndexedFaceSet` a navíc

⁵ Popis VRML je nad rámec této práce

mají definováno mapování textury příslušné k danému řezu. U jednotlivých řezů jsou specifikovány odkazy k těmto texturám, které jsou fyzicky uloženy v souborech `tex0.jpg` až `texN.jpg`, kde $N+1$ je počet tříd řezů v modelu. Pokud se textury na řezech nezobrazují, může to být způsobeno jejich příliš velkým rozlišením, které neumí daný prohlížeč zpracovat.

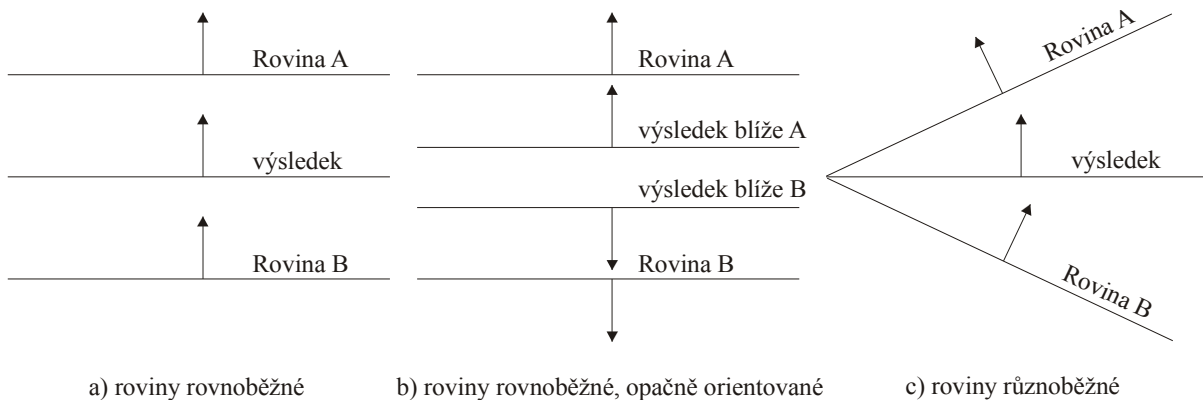
2.8 Klíčové body

Pro účely vytváření animací byla zavedena tzv. časová osa (Kap. 4.3). Na tuto časovou osu může uživatel ukládat parametry modelu a parametry jeho zobrazení do klíčových bodů. Parametry, které může klíčový bod obsahovat, najdete v Kap. 4.9. Klíčový bod obsahuje tedy toto nastavení a navíc čas, na kterém se nachází. Při vytváření animace se pak generují statické snímky, kde parametry modelu a zobrazení se načítají z časové osy v příslušném čase pro daný snímek animace. Model se musí vždy před vyrenderováním daného snímku přepočítat podle nových parametrů (Kap 2.7.2). Pokud probíhá načtení parametrů z osy v čase, kde se přímo nachází nějaký klíčový bod, pak stačí hodnoty rovnou načíst z klíčového bodu. V případě, že zmíněný čas leží mezi dvěma klíčovými body, je potřeba získat mezilehlé nastavení interpolací hodnot z těchto dvou bodů. Pokud se jedná o parametry udávané v jednoduchých číselných hodnotách, jako jsou hodnota intenzity pro detekci povrchu, barvy, otočení, zoom, poloha modelu, prahování palety, souřadnice jednoduchých⁶ řezů, a parametry osvětlení, tak se tyto hodnoty interpolují lineárně mezi přilehlými klíčovými body. U parametrů, které nabývají pouze hodnot ano a ne nebo je nelze interpolovat, jako jsou různé změny orientace modelu, číslo palety, orientace jednoduchých řezů a nastavení metody stínování (flat/gouraud) a podobně, se vezme hodnota z nejbližšího předešlého klíčového bodu. Speciální situace nastává u interpolace řezů.

2.8.1 Speciální interpolace řezových výrazů

Aby bylo možné použít při animacích dynamického pohybu obecných řezů, bylo nutné vyřešit problém, jak interpolovat dva řezové výrazy ze dvou sousedních klíčových bodů. Nakonec se ukázalo jako velmi vhodné následující řešení. Podmínkou pro interpolaci je shodná struktura řezových výrazů. To znamená, že oba řezové výrazy jsou shodné až na parametry jednotlivých zadání rovin uvnitř řezových výrazů. Tedy například řezový výraz $(R(0,0,1,0)+R(1,0,0,0))*R(1,1,0,0)$ je interpolovatelný s $(R(-1,2,1,0)+R(1,0,0,3))*R(2,1,0,-5)$, ale není interpolovatelný s $(R(0,0,1,0)+R(1,0,0,0))+R(1,1,0,0)$. V případě, že řezové výrazy z daných klíčových bodů nejsou interpolovatelné, vezme se přímo řezový výraz z prvního klíčového bodu. Tím se tedy problém interpolace obecných řezových výrazů zúžil na interpolaci dvou obecných rovin. Interpolace rovin je řešena odlišně pro případ, že dané roviny jsou rovnoběžné (Obr. 17 a),b)) a pro případ, že dané roviny jsou různoběžné (Obr. 17 c)). V případě rovnoběžných rovin se lineárně interpoluje poloha výsledné roviny mezi danými dvěma rovinami. U různoběžných rovin se pro získání výsledku lineárně interpoluje úhel sevřený těmito rovinami, přičemž výsledná rovina prochází průsečnicí daných rovin.

⁶ Ve smyslu zadávání uživatelem (Kap. 4.2.4)



Obr. 17: Interpolace rovin v řezovém výrazu

2.9 Přepočítání modelu při změně parametrů

Je možné nastavovat různé parametry modelu a jeho zobrazení (Kap. 4.9). Pro tyto parametry je pak vygenerován a zobrazen model na obrazovce, případně vyexportován do některého z dostupných formátů. Pokud uživatel změní některý z parametrů, je podle těchto parametrů nutné provést změny zobrazení případně i přepočítat samotný 3D model (trojúhelníkový či bodový) nebo jeho části. Většina parametrů lze ukládat do klíčových bodů a tedy měnit v rámci animace. Generování snímků animace je případ, kdy se nejvíce projeví optimalizace přepočítání modelu při změnách parametrů, neboť pro každou sekundu animace je třeba model mnohokrát zobrazit s různými parametry.

2.9.1 Změny metody zobrazení a výpočtu normál povrchu

Aplikace podporuje několik metod detekce a zobrazení povrchu modelu a několik metod pro výpočet normál povrchu. Tyto metody není možné ukládat do klíčových bodů a není tedy možné je použít dynamicky v animacích. Každá animace je vygenerována celá za použití jedné z těchto metod. Jsou implementovány tři metody pro detekci povrchu a to Marching cubes, Marching tetrahedrons a Dividing cubes. Jestliže uživatel přepíná mezi těmito třemi metodami, je celý dosavadní model uvolněn z paměti a následně se provede vygenerování nového modelu dle zadané metody včetně řezů. Při přepnutí metody výpočtu normál povrchu (Kap. 2.5) není přepočítáván celý model, ale pouze normály povrchu dle nové metody.

2.9.2 Změny parametrů bez vlivu na 3D model

Další skupinou parametrů jsou ty, jejichž změna nemá vliv na geometrický model. Lze je ukládat do klíčových bodů a snadno animovat, protože nevyžadují přepočítání 3D modelu. Jsou to většinou parametry zobrazení modelu, u kterých stačí pouze vygenerovat nový snímek pomocí OpenGL. Jsou to například úhly pohledu, barvy povrchu, pozadí, zoom, osvětlení, orientace modelu, posun a podobně. Mezi tyto parametry patří i informace o použité paletě, která má vliv především na výsledné barvy povrchu řezů. Tento parametr se trochu liší od naposledy jmenovaných parametrů, které nevyžadovaly žádné změny do modelu. Změna palety neovlivní 3D model, ale ovlivňuje textury na řezech (trojúhelníkový model), případně barevnost bodů řezu (bodový model). Pro urychlení přepočítávání textur si aplikace pamatuje pro každou výslednou texturu i její nepaletovanou verzi, podle které vygeneruje novou barevnou texturu. Nepaletovaná verze textury obsahuje místo barev původní vypočítané hodnoty intenzit v daném místě prostoru (Kap. 2.6.6). U bodového modelu se pro každý bod řezu zjistí intenzita v daném místě prostoru a následně podle palety i RGB barva bodu.

2.9.3 Změny parametrů s vlivem na 3D model

Poslední skupinou parametrů jsou ty, jejichž změna má přímý vliv na geometrický model. Tyto parametry lze ukládat do klíčových bodů a tedy i animovat. Jsou to intenzita pro detekci povrchu a parametry řezů modelem. Při změně intenzity povrchu je nutné znovu vygenerovat povrch modelu (jednou z metod MC, MT, DC) a případně i trojúhelníky nebo body reprezentující povrch řezů. Je to časově nejnáročnější změna modelu. Pokud nedošlo zároveň ke změně parametrů řezů, není potřeba přepočítávat textury na řezech.

Při změně parametrů řezů modelem (Kap. 2.6) se automaticky nepřepočítávají všechny řezy, ale je zde použito chytrější řešení. Necht' RV a RV' jsou původní a nový řezový výraz, podle kterého chceme model přepočítat. Necht' $T_i : i = 1 \dots n$ jsou třídy rovin řezů použitých v původním RV a $T'_j : j = 1 \dots n'$ jsou třídy rovin řezů použitých v novém RV' . Necht' $M_i : i = 1 \dots n$ jsou struktury v paměti popisující řezy modelem jednotlivými třídami řezů z původního RV a $M'_j : j = 1 \dots n'$ jsou struktury v paměti popisující řezy modelem jednotlivými třídami řezů z nového RV' . Obsahem těchto struktur jsou neodfiltrováné trojúhelníky (nebo body) daného řezu, dvě sady odfiltrovaných trojúhelníků připravených k zobrazení jedné a druhé strany řezu (nebo jedna sada odfiltrovaných bodů) a nepaletovaná a paletovaná textura daného řezu. Necht' R je množina rovin obsahující roviny, které se vyskytují v novém zadání řezů. Ekvivalence dvou tříd rovin T_i a T'_j se značí $T_i \cong T'_j$ a znamená, že rovina r , která je reprezentantem třídy T_i , je ekvivalentní nebo opačná k rovině q , která je reprezentantem třídy T'_j . Při přepočítání modelu postupuje takto. Uvolní se z paměti ty M_i , kde neexistuje j , že $T_i \cong T'_j$. Zcela nově se vygenerují M'_j , kde neexistuje i , že $T_i \cong T'_j$. V případě, že existuje i a j , že $T_i \cong T'_j$, pak se M_i přiřadí do M'_j . U těch $M'_j : j = 1 \dots n'$, které byly převzaty z $M_i : i = 1 \dots n$ se provede (pokud zároveň nebyla změněna intenzita pro detekci povrchu) pouze nové odfiltrování trojúhelníků (bodů) řezu podle RV' . V případě, že byla zároveň s řezy změněna i intenzita pro detekci povrchu modelu, provede se u převzatých M'_j i přepočítání neodfiltrovaných trojúhelníků (bodů) řezů. I v tomto případě se ale nemusejí u převzatých M'_j přepočítávat textury, což výrazně přispívá k rychlosti provedení změn modelu při změnách parametrů řezů.

2.10 Závěr implementační části

První z cílů této práce, což bylo vytvoření uživatelsky přívětivé aplikace, která by umožňovala jednoduchou a rychlou práci s tomografickými studiemi, byl v základních aspektech splněn. Několik implementovaných povrchových zobrazovacích metod z oblasti renderingu zdaleka nereprezentuje všechny možnosti zobrazení objemových tomografických dat. Povrchové zobrazování dat není jediné a snad ani nejvhodnější pro získání představy o rozložení záření v prostoru. Existuje mnoho jiných a názornějších způsobů zobrazení, jako je použití více povrchů s poloprůhledností, procházení dat podél paprsku (například MIP), kombinace více metod zobrazení a podobně. Tato aplikace je určena především pro tvorbu tomografických prezentací. Díky širokým možnostem nastavení zobrazení a mnoha exportním formátům, včetně vytváření animací a exportu 3D modelu do VRML, tento požadavek splňuje. Náповěda k aplikaci (Kap. 4) byla pečlivě zpracována a měla by umožnit použití aplikace i méně zkušeným uživatelům. Naopak zkušenějším by měla pomoci pochopit všechny nabízené možnosti aplikace.

Speciálně pro tuto aplikaci zde byla navržena metoda obecných rovinných řezů zadaných řezovým výrazem a implementovaných s pomocí tříd rovin. Dále zde byla navržena

metoda částečného přepočítávání modelu při změně zadání řezů, která podstatným způsobem ovlivňuje rychlost aplikace zejména při generování animací. V neposlední řadě byl pro rychlost a použitelnost aplikace klíčový návrh a implementace vylepšeného diferenčního algoritmu pro výpočet textur na řezech.

V budoucnu by bylo možné rozšířit aplikaci o jiné než povrchové zobrazovací metody, které by byly pro zobrazení intenzity v prostoru názornější. Zlepšením v oblasti exportu do VRML by mohlo být optimalizované použití indexů v uzlech IndexedFaceSet⁷, tím by byla zmenšena velikost výsledného souboru. Stejně tak by bylo možné implementovat kompresi výsledných VRML souborů, kterou VRML prohlížeče podporují. Další funkcí by v budoucnu mohla být možnost nastavit použití například kubické interpolace dvou klíčových bodů při tvorbě animací (Kap 2.8) místo lineární. Tím by měly přechody přes klíčové body hladší průběh.

Aplikace byla testována na operačních systémech Windows 98, 2000 a XP. Použitý hardware AMD Duron 750MHz, 128MB až 384MB RAM, grafický akcelerátor OpenGL GeForce2 MX, rozlišení monitoru 1024x768 až 1280x1024. Pro běžná medicínská data byla rychlost zpracování naprosto dostatečná pro interaktivní práci. Tedy aplikace je schopna kvalitního provozu na dnes běžně dostupném osobním počítači.

Druhým cílem bylo porovnání kvality a náročnosti jednotlivých implementovaných zobrazovacích metod. Touto problematikou se zabývá následující kapitola.

⁷ Popis VRML je nad rámec této práce

3 Porovnání zobrazovacích metod

3.1 Cíle porovnání

Tato aplikace podporuje více zobrazovacích metod (Kap. 2.4). Implementované metody spadají do kategorie povrchových zobrazovacích algoritmů. Tyto algoritmy zobrazují povrch procházející oblastí prostoru, kde naměřená intenzita záření je rovna zadané hodnotě. Tuto hodnotu zadává uživatel. Implementovány jsou metody Marching cubes, Marching tetrahedrons a Dividing cubes. Tyto metody jsou dobře použitelné pro následné libovolné změny úhlu pohledu na scénu, zvětšování, posun a podobně, neboť jednou vygenerovaný 3D model nevyžaduje v těchto případech žádné úpravy. Aby mohl být geometrický povrch modelu zobrazen s pomocí osvětlení, je potřeba znát normálové vektory povrchu modelu (Kap. 2.5). Existuje implementováno více metod, které se snaží odhadnout normálu v daném místě povrchu. Na volbě této metody značně závisí výsledek zobrazení povrchu modelu.

Cílem je porovnat jednotlivé metody detekce povrchu i metody pro výpočet normál povrchu především z kvalitativního hlediska.

3.2 Jak porovnávat

Klíčovým krokem každého porovnávání je zvolit pro vhodná srovnávací kritéria. V případě 3D modelů vygenerovaných různými povrchovými algoritmy, kde některé generují trojúhelníkový model a jiné nespojitý bodový model a kde geometrie výsledných povrchů se liší, není snadné zvolit vhodné kritérium. Společná věc u všech metod je ta, že výsledek je finálně vyrenderován na 2D průmětnu. Kvalitu metod by bylo možné porovnat srovnáním výsledných 2D obrázků zobrazujících stejná data při pohledu ze stejného úhlu. Úhly je potřeba zvolit vhodně, aby výsledky byly reprezentativní. Velmi důležitá je i volba dat, na které bude probíhat srovnávání.

Výsledek, který vidí uživatel při nastavení libovolné metody zobrazení, je obrázek složený z pixelů nesoucích barvy RGB. Tyto barvy byly vypočítány pomocí definovaného osvětlení scény a použitého světelného modelu (v případě OpenGL bylo použito Gouraudovo stínování). Bylo by jistě možné srovnávat výsledky algoritmů přímo srovnáním barev jednotlivých odpovídajících pixelů na vyrenderovaných obrázcích. Problém je, že výsledky by závisely na nastaveném osvětlení scény. Osvětlení scény se ale netýká použitých algoritmů pro výpočet modelu, z toho důvodu by nemělo mít vliv na jejich porovnávání. Řešením, jak porovnat výstup jednotlivých algoritmů bez vlivu osvětlení, je neporovnávat obrázky obsahující konečné barvy po aplikaci osvětlení, ale ukládat do pixelů místo barev normály povrch viditelného v daném místě průmětny. Následně by bylo možné porovnávat vzájemné odchylky normál v odpovídajících si místech povrchu.

Další metodou je porovnání geometrie detekovaného povrchu. Lze porovnávat například jak moc se povrch modelu blíží povrchu předlohy, podle které byla testovací studie vygenerována, pomocí průměrné vzdálenosti od povrchu nebo směrodatné odchylky.

3.3 S čím porovnávat

Jak bylo řečeno, velmi důležitá volba dat, na které bude probíhat srovnávání. Srovnání kvality algoritmů je možné jen v případě, že je k dispozici vzorek určující, jak by měl vypadat ideální výsledek. Teprve porovnáním výsledků jednotlivých algoritmů s tímto ideálním výsledkem lze najít algoritmus dosahující nejlepších výsledků. Těžko lze použít k takovému účelu medicínská data zobrazující část těla pacienta, protože u nich není možné získat zároveň ideální výsledek zobrazení, který od algoritmů očekáváme. Pro účely porovnávání implementovaných algoritmů je potřeba znát přesný průběh hodnot intenzity v prostoru, ze

kterého vychází zpracovávaná studie. Skutečné studie jsou však jen nepřesným obrazem tohoto průběhu. Jedinou schůdnou možností je vytvořit umělé studie představující průběh intenzity v prostoru, který je přesně znám. Pak je možné vypočítat přesný průmět povrchu procházejícího prostorem v místech s konkrétní hodnotou záření a normály povrchu v daných místech průmětny. Takový průmět lze považovat za ideální výsledek zobrazení.

3.4 NOP formát

NOP (Normal Picture) je speciální formát vyvinutý pro srovnání výsledků jednotlivých algoritmů pro detekci a zobrazení povrchu. NOP soubor obsahuje data zobrazující 2D průmět povrchu modelu. Od klasického statického snímku vyrenderovaného pomocí OpenGL se liší v tom, že neukládá do každého pixelu RGB barvu osvětleného povrchu modelu v daném místě, případně barvu pozadí scény, ale normálu povrchu daného místa modelu (Kap. 2.7.3).

NOP je binární soubor. Uvnitř se skládá z hlavičky a samotných dat. Hlavička je na začátku souboru a je dlouhá 18 bytů. Obsahuje text identifikující NOP soubor ve tvaru `NOPfile1.0` následovaný počtem pixelů uloženého obrázku ve směru vodorovné osy x a počtem pixelů uloženého obrázku ve směru svislé osy y . Rozměry obrázku jsou typu `LITTLEENDIAN DWORD` (4 byte). Bezprostředně po hlavičce následují data obrázku.

Hodnota každého pixelu je reprezentována vektorem (n_x, n_y, n_z) . Složky vektoru jsou v souboru uloženy jako trojice 32bit floating point čísel ve formátu IEEE-754 (tak jak jsou v paměti PC). Pixely jsou uloženy po řádcích zleva doprava a řádky postupně zdola nahoru.

Je pravidlem, že v místech, kde se nezobrazil povrch, je místo barvy pozadí hodnota vektoru $(0, 0, -2)$ a ostatní normály povrchu jsou normalizovány na jednotkovou velikost. Tím lze odlišit bezpečně pozadí od promítnutého tělesa. Uložené normály povrchu musí být transformovány do souřadné soustavy průmětny a to tak, že osa x této souřadné soustavy odpovídá ose x obrázku, osa y odpovídá ose y obrázku. Osa z souřadné soustavy průmětny směřuje k pozorovateli.

3.5 Nástroj pro generování testovacích dat (TDC)

Za účelem získání testovací studie a ideálního průmětu hledaného povrchu (Kap. 3.3) byl navrhnut a implementován nástroj TDC neboli Test Data Creator. TDC je malý prográmk spouštěný z příkazové řádky. Jeho vstupem je textový soubor s příponou `'td'` neboli Test Data Description. Výstupem je pak studie typu Interfile obsahující 3D matici voxelů odpovídajících průběhu intenzity popsané ve vstupním souboru nebo je výstupem NOP soubor zobrazující normály daného povrchu v daném průmětu.

3.5.1 TDD soubor

Soubor s příponou `'td'` (Test Data Description) je textový soubor a slouží k popisu testovací scény a k zadání vlastností výstupu. Musí obsahovat záznam `output` a může obsahovat jeden či více záznamů typu `teleso`. Způsob, kterým se generuje výstup z TDD souboru je specifikován v Kap. 3.5.3. Zde je popsána syntaxe záznamů v TDD a stručný popis jejich parametrů. Záznam `output` popisuje vše co se týká vlastností výstupu. Ukázka je níže.

```
output
{
    SizeX=50;
    SizeY=50;
    SizeZ=50;
    SubD=3;
```

```

MaxI=20000;
I=10000;
RotX=30;
RotY=30;
Zoom=1;
ResX=1280;
ResY=960;
};

```

Záznam musí obsahovat všechny zmíněné položky. Jednotlivé položky mají následující (typ) a význam.

- SizeX ... (int) rozměr 3D matice v ose x
- SizeY ... (int) rozměr 3D matice v ose y
- SizeZ ... (int) rozměr 3D matice v ose z
- SubD ... (int) SubD^3 je počet hodnot intenzit uvnitř jednoho voxelu zprůměrovaných do výsledné hodnoty intenzity voxelu. Zjišťování intenzit probíhá uvnitř voxelu v pravouhlé mřížce mající v každém směru rozměr SubD prvků. Čím větší hodnota, tím větší přesnost a menší rychlost výpočtu intenzit výsledné testovací studie.
- MaxI ... (float) globální maximální intenzita voxelu
- I ... (float) globální intenzita povrchu
- RotX ... (float) úhel pohledu při generování NOP projekce, odpovídá hodnotě Rotace X z aplikace TomoAVI
- RotY ... (float) úhel pohledu při generování NOP projekce, odpovídá hodnotě Rotace Y z aplikace TomoAVI
- Zoom ... (float) přiblížení pohledu při generování NOP projekce, odpovídá hodnotě Zoom z aplikace TomoAVI
- ResX ... (int) rozlišení výsledného NOP souboru v ose x
- ResY ... (int) rozlišení výsledného NOP souboru v ose y

TDD soubor může obsahovat jeden či více záznamů typu `teleso`. Záznam `teleso` popisuje objekt ovlivňující průběh intenzity v prostoru. Jsou implementovány dva druhy těles, koule a kvádr. Scéna bez těles má všude nulovou intenzitu. Přidáním tělesa typu koule do scény, získá rozložení hodnot intenzity scény kulový tvar. Přidáním tělesa typu kvádr do scény, získá rozložení hodnot intenzity scény tvar kvádru. Tělesa lze spojovat i odečítat. Více v Kap. 3.5.3. Syntaxe pro zadávání těles typu kvádr s příkladem:

```

teleso Kvadr
{
    SX=0;
    SY=0;
    SZ=0;
    I=10000;
    T=1;
    MaxI=20000;
    Dute=0;
    Minus=0;
    RotX=0;
    RotY=0;
    RotZ=0;
    Lx=10;
    Ly=10;
    Lz=20;
};

```

Jednotlivé položky tělesa typu kvádr mají následující (typ) a význam.

- SX ... (float) x-ová souřadnice středu kvádru
- SY ... (float) y-ová souřadnice středu kvádru
- SZ ... (float) z-ová souřadnice středu kvádru
- I ... (float) intenzita na povrchu kvádru
- T ... (float) tloušťka. Intenzita se vzdáleností nad povrch kvádru lineárně klesá od hodnoty I až k nule, které dosáhne ve vzdálenosti $T/2$ nad povrchem. Zároveň intenzita se vzdáleností pod povrch kvádru lineárně stoupá od hodnoty I až k hodnotě $2*I$, které dosáhne ve vzdálenosti $T/2$ pod povrchem.
- MaxI ... (float) maximální intenzita generovaná tímto tělesem. Pokud je vypočítaná intenzita pro toto těleso v určitém bodě prostoru větší než MaxI, nahradí se hodnotou MaxI.
- Dute ... (int) nulová hodnota označuje plné těleso viz. význam hodnoty T. Nenulová hodnota označuje duté těleso, což znamená, že intenzita pod povrchem tělesa klesá k nule stejně jako nad povrchem viz. význam T.
- Minus ... (int) nulová hodnota označuje kladné těleso, nenulová záporné těleso (Kap. 3.5.3)
- RotX ... (float) úhel rotace tělesa ve stupních okolo osy x
- RotY ... (float) úhel rotace tělesa ve stupních okolo osy y
- RotZ ... (float) úhel rotace tělesa ve stupních okolo osy z
- Lx ... (float) polovina velikosti neotočeného kvádru ve směru osy x
- Ly ... (float) polovina velikosti neotočeného kvádru ve směru osy y
- Lz ... (float) polovina velikosti neotočeného kvádru ve směru osy z

Syntaxe pro zadávání těles typu koule s příkladem:

```
teleso Koule
{
    SX=0;
    SY=0;
    SZ=0;
    R=20;
    I=10000;
    T=2;
    MaxI=20000;
    Dute=0;
    Minus=0;
    RotX=0;
    RotY=0;
    RotZ=0;
};
```

Jednotlivé položky tělesa typu koule mají následující (typ) a význam.

- SX ... (float) x-ová souřadnice středu koule
- SY ... (float) y-ová souřadnice středu koule
- SZ ... (float) z-ová souřadnice středu koule
- R ... (float) poloměr koule
- I ... (float) intenzita na povrchu koule
- T ... (float) tloušťka. Intenzita se vzdáleností nad povrch koule lineárně klesá od hodnoty I až k nule, které dosáhne ve vzdálenosti $T/2$ nad povrchem. Zároveň

intenzita se vzdáleností pod povrch koule lineárně stoupá od hodnoty I až k hodnotě $2 \cdot I$, které dosáhne ve vzdálenosti $T/2$ pod povrchem.

- MaxI ... (float) maximální intenzita generovaná tímto tělesem. Pokud je vypočítaná intenzita pro toto těleso v určitém bodě prostoru větší než MaxI, nahradí se hodnotou MaxI.
- Dute ... (int) nulová hodnota označuje plné těleso viz. význam hodnoty T. Nenulová hodnota označuje duté těleso, což znamená, že intenzita pod povrchem tělesa klesá k nule stejně jako nad povrchem viz. význam T.
- Minus ... (int) nulová hodnota označuje kladné těleso, nenulová záporné těleso (Kap. 3.5.3)
- RotX ... (float) úhel rotace tělesa ve stupních okolo osy x (u koule nemá význam)
- RotY ... (float) úhel rotace tělesa ve stupních okolo osy y (u koule nemá význam)
- RotZ ... (float) úhel rotace tělesa ve stupních okolo osy z (u koule nemá význam)

3.5.2 Použití TDC

Program TDC generuje dva druhy výstupu. Jednak vytváří studie typu Interfile obsahující 3D matici hodnot intenzity v prostoru. Rozložení intenzity je popsáno ve vstupním TDD souboru pomocí těles (Kap. 3.5.3). TDC také vytváří NOP projekce těles popsaných v TDD. Program se volá z příkazové řádky takto:

```
tdc.exe input output [/f] [/n]
```

kde jednotlivé parametry znamenají:

- input ... název vstupního souboru s příponou *.tdd (Kap. 3.5.1)
- output ... název výstupního souboru bez přípony (doplň se podle typu výstupu)

Z parametrů /f a /n lze použít žádný nebo jeden z nich:

- žádný parametr ... bude vygenerována studie typu Interfile, soubory s příponami *.hdr a *.img. Data studie budou typu 16bit integer.
- /f ... bude vygenerována studie typu Interfile, soubory s příponami *.hdr a *.img. Data studie budou typu 32bit float. Doporučený formát pro TomoAVI.
- /n ... bude vygenerována NOP projekce, soubor s příponou *.nop.

Při generování Interfile studie musí být v aktuálním adresáři aplikace TDC soubor template.hdr obsahující šablonu hlavičky studie.

3.5.3 Interní postup generování testovací studie Interfile

Ze vstupního souboru se načtou údaje o velikosti výsledné studie a počtu testů na jeden voxel. Dále se načtou údaje o tělesech, podle kterých se mají hodnoty intenzit ve studii vypočítat. Pro každé těleso lze vypočítat intenzitu záření, kterou těleso představuje, v libovolném bodě prostoru. Zjišťování průměrné intenzity voxelu probíhá zprůměrováním SubD^3 intenzit uvnitř voxelu v bodech rozložených v pravouhlé mřížce mající v každém směru rozměr SubD prvků. Tato metoda byla zvolena pro její jednoduchost. Přesný výpočet intenzity voxelu z tvarů těles byl značně složitý. Na obrázku Obr. 18 je vidět rozložení intenzity v prostoru, ve kterém je definováno jedno těleso typu koule. Bílá barva označuje nulovou intenzitu a černá barva maximální intenzitu. Na obrázku Obr. 19 je vidět rozložení

intenzity v prostoru, ve kterém je definováno jedno těleso typu kvádr. Vlevo je plný kvádr a vpravo je dutý kvádr.

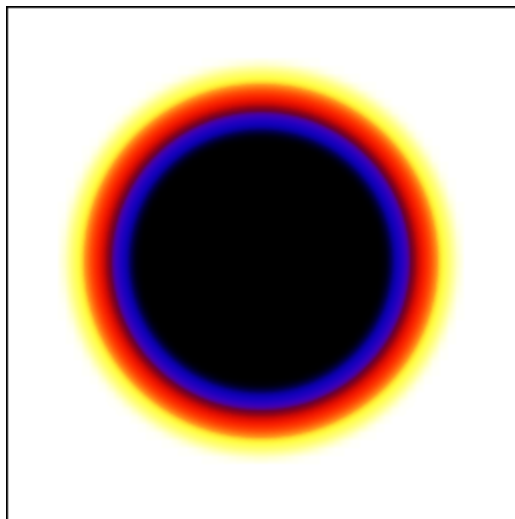
Pro účely testování bylo potřeba vytvářet testovací studie z kombinace více těles. Každé těleso ve vstupním souboru je buď kladné nebo záporné (parametr Minus). Zpracování více těles ze vstupního souboru probíhá tak, že nejprve se jednotlivá všechna kladná tělesa do jednoho kladného tělesa a všechna záporná tělesa do jednoho záporného tělesa. Pak se od sjednocení kladných těles odečte sjednocení záporných těles.

Je nutné definovat, co je to sjednocení a odčítání těles ve smyslu intenzit generovaných těmito tělesy. Má-li těleso A v bodě b intenzitu $A(b)$ a těleso B intenzitu $B(b)$, pak těleso C vzniklé sjednocením těchto dvou těles má v bodě b intenzitu $C(b)=\max(A(b), B(b))$. Výsledek sjednocení koule a kvádra je vidět na obrázku Obr. 20 vlevo.

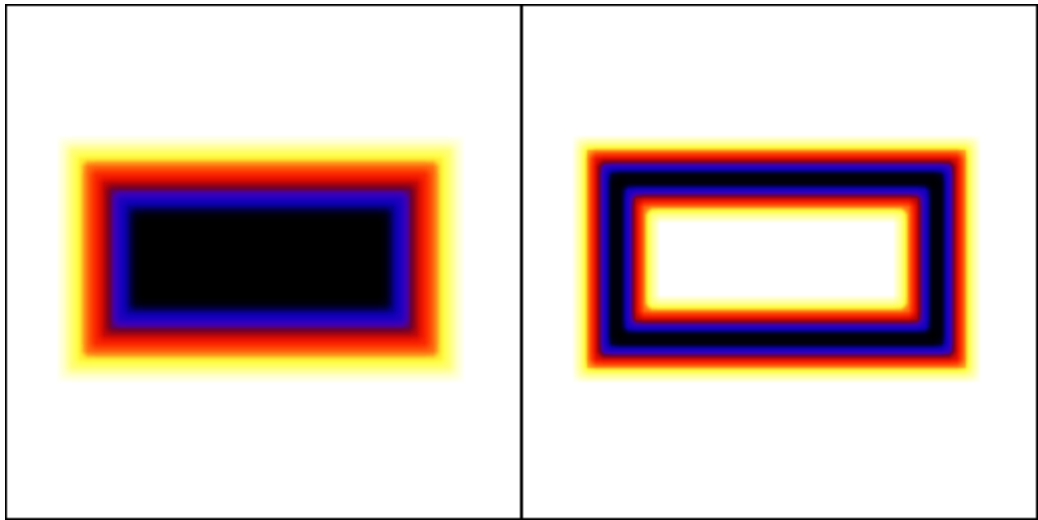
U odečítání těles je to složitější. Nejprve je potřeba definovat inverzní intenzitu tělesa. Má-li těleso A v bodě b intenzitu $A(b)$, pak má v bodě b inverzní intenzitu $A^-(b)$. Hodnota $A^-(b)$ se vypočítá podle Obr. 21. Výpočet inverzní intenzity je ovlivněn globálními hodnotami I a $\text{Max}I$ ze sekce output vstupního souboru. Globální $\text{Max}I$ představuje maximální intenzitu voxelu ve výsledných testovacích datech a globální I představuje předpokládanou intenzitu pro detekci povrchu testovaných algoritmů na těchto datech. Globální I by měla odpovídat I u jednotlivých těles. U konkrétního tělesa udává I hodnotu intenzity na geometrickém povrchu daného tělesa, povrch s intenzitou I detekovaný povrchovým algoritmem aplikace TomoAVI z dat studie by měl představovat geometrický povrch těles ze vstupu TDC.

Odečítání těles ve smyslu intenzit generovaných těmito tělesy je tedy definováno následovně. Má-li těleso A v bodě b intenzitu $A(b)$ a těleso B intenzitu $B(b)$, pak těleso C vzniklé odečtením tělesa A od tělesa B má v bodě b intenzitu $C(b)=\min(A^-(b), B(b))$. Výsledek odečtení kvádra od koule je vidět na obrázku Obr. 20 vpravo.

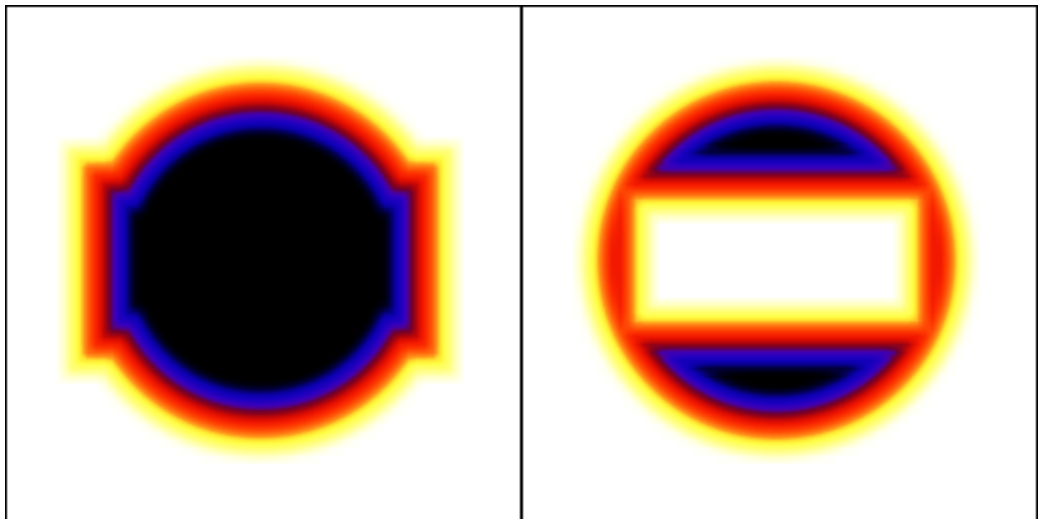
Tento postup výpočtu intenzity voxelu byl zvolen proto, že výsledná isoplocha vedená daty studie i gradienty intenzity v okolí isoplochy odpovídají geometrii těles použitých na vstupu algoritmu. Takové studie lze použít pro testování povrchových algoritmů.



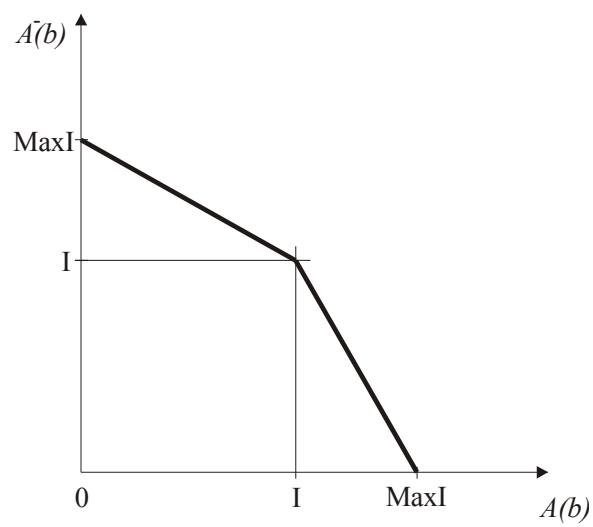
Obr. 18: Řez studií zobrazující plnou kouli



Obr. 19: Řez studií zobrazující plný a dutý kvádr



Obr. 20: Řez studií zobrazující součet a rozdíl těles



Obr. 21: Inverzní intenzita tělesa

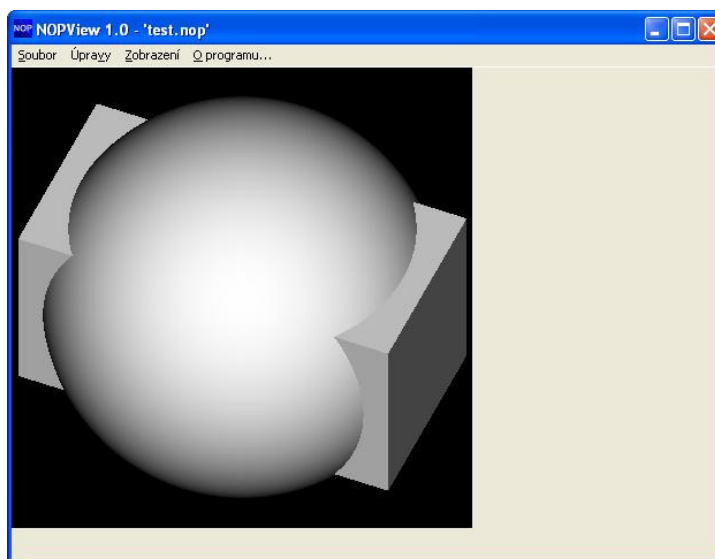
3.5.4 Interní postup generování testovací projekce NOP

Kromě vytváření testovacích studií typu Interfile obsahující 3D matici hodnot intenzity v prostoru umožňuje TDC generovat NOP projekce zadaných těles (Kap. 3.5.2). Parametry NOP projekce jsou zadány jako položky záznamu output ve vstupním souboru. Pro účely NOP projekce to jsou především položky RotX, RotY a Zoom udávající úhel a přiblížení pohledu podobně jako parametry Rotace X, Rotace Y a Zoom v aplikaci TomoAVI a pak to jsou položky ResX, ResY udávající rozměr výstupního souboru v pixelech.

NOP projekce je generována algoritmem, který počítá průsečíky paprsku průmětu s tělesy ze vstupního souboru a podle vzdálenosti průsečíků od průmětny a parametrů těles určí, který bod jakého tělesa je viditelný v daném pixelu. Pak vypočítá normálu povrchu v daném bodě a uloží do výstupního souboru. Pokud se z téže vstupních dat vygeneruje NOP projekce a zároveň testovací Interfile studie, nazývá se NOP projekce referenčním průmětem dané testovací studie.

3.6 Nástroj pro porovnání NOP projekcí (NOPView)

Aby bylo možné porovnávat dvě NOP projekce, byl navrhnout a implementován pomocný nástroj NOPView (Obr. 22). NOPView umožňuje načíst a zobrazovat dvě NOP projekce. Hlavní funkcí je porovnání těchto projekcí. Lze zobrazit buď každou projekci zvlášť nebo obrázek reprezentující odchylky odpovídajících si pixelů obou projekcí. U zobrazení jedné projekce jsou světle vyznačeny oblasti s malou odchylkou normály povrchu od pozorovatele a naopak tmavě vyznačeny oblasti s velkou odchylkou normály povrchu od pozorovatele. Při zobrazení rozdílu projekcí jsou světle vyznačeny oblasti s velkou vzájemnou odchylkou příslušných dvou normál povrchu z jedné a druhé projekce, naopak tmavě oblasti s malou odchylkou. Implementováno je několik metod, které vyjádří míru rozdílnosti projekcí číselně. Detaily funkcí aplikace NOPView jsou zřejmé z následujícího stručného ovládání programu.



Obr. 22: NOPView

3.6.1 Funkce NOPView

NOPView umožňuje pracovat zároveň se dvěma NOP soubory a navzájem je porovnávat. Jedna NOP projekce se nazývá hlavní a druhá referenční. Menu **Soubor** obsahuje tři položky (Obr. 23). **Otevřít hlavní...** načte hlavní NOP projekci ze souboru a **Otevřít referenční...** načte referenční NOP projekci ze souboru. **Konec** ukončí NOPView.

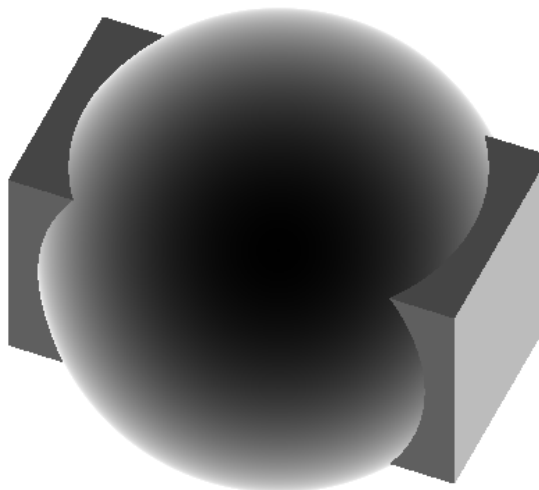
| | |
|-----------------------|--------|
| Otevřít hlavní... | Ctrl+O |
| Otevřít referenční... | Ctrl+T |
| <hr/> | |
| Konec | |

Obr. 23: Menu Soubor

Menu **Úpravy** obsahuje tři položky (Obr. 24). **Kopírovat** zkopíruje právě viditelné zobrazení projekcí do schránky. **Invertovat** invertuje barvy zobrazení NOP projekcí (Obr. 25). **Zvýraznit...** zobrazí dialogové okno (Obr. 26), kde je možné nastavit rozsah odchylek normálových vektorů, které se následně zvýrazní červenou barvou (Obr. 27). V případě jedné projekce se odchylkou myslí odchylka od směru k pozorovateli. V případě porovnání dvou projekcí se myslí vzájemná odchylka příslušných normál.

| | |
|--------------|--------|
| Kopírovat | Ctrl+C |
| Invertovat | Ctrl+I |
| Zvýraznit... | Ctrl+E |

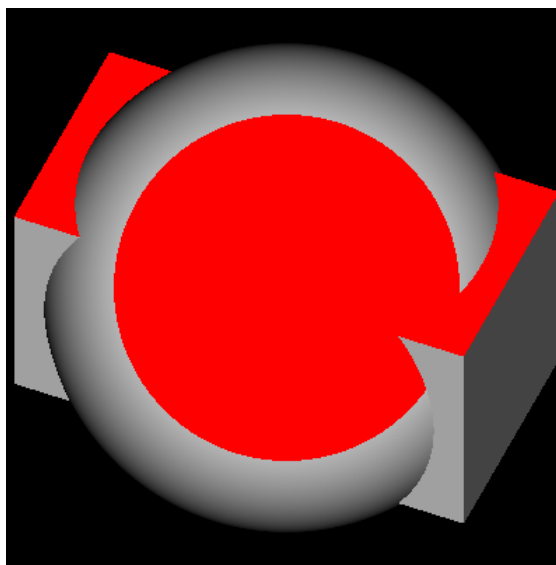
Obr. 24: Menu Úpravy



Obr. 25: Inverze zobrazení NOPu



Obr. 26: Dialog pro zvýraznění odchylek

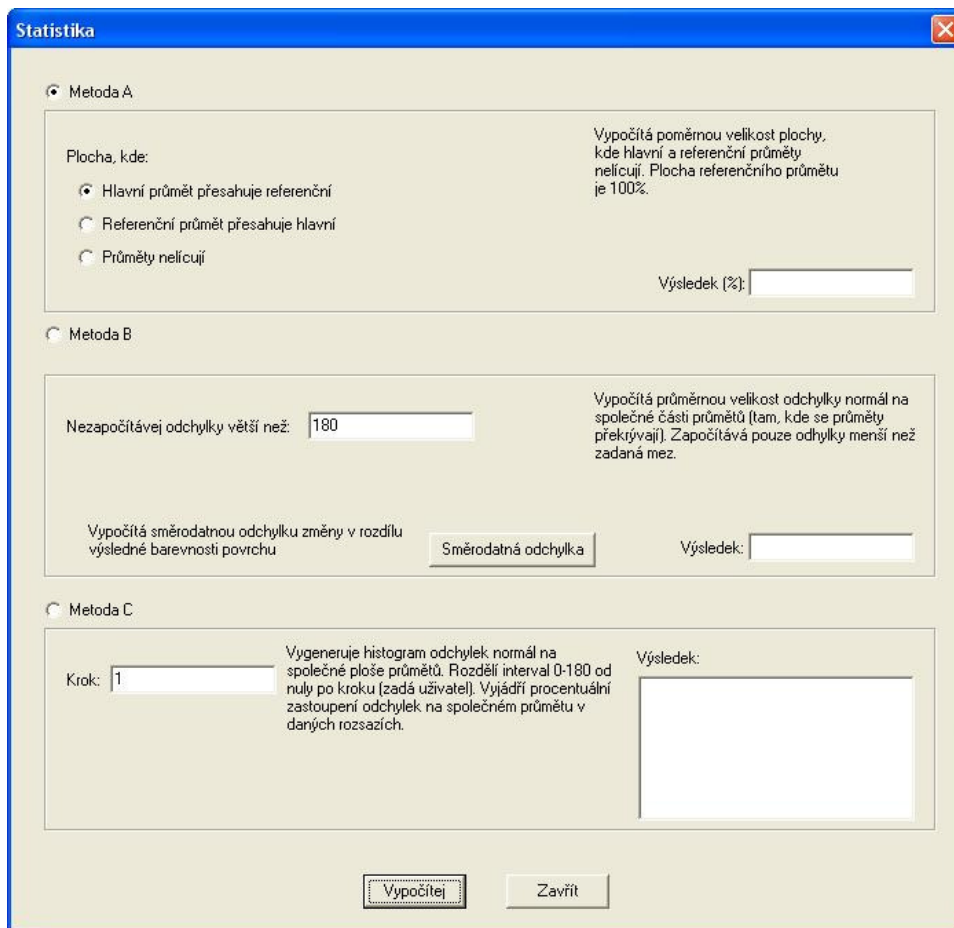


Obr. 27: Zvýraznění odchylek

Menu **Zobrazení** obsahuje čtyři položky (Obr. 28). **Hlavní** přepne na zobrazení odchylek hlavní projekce. **Referenční** přepne na zobrazení odchylek referenční projekce. **Rozdíl** přepne na zobrazení vzájemných odchylek hlavní a referenční projekce. **Statistika...** zobrazí dialogové okno (Obr. 29) určené ke statistickému zpracování vzájemných rozdílů obou projekcí.

| | |
|---------------|--------|
| Hlavní | Ctrl+H |
| Referenční | Ctrl+R |
| ● Rozdíl | Ctrl+Z |
| Statistika... | Ctrl+S |

Obr. 28: Menu Zobrazení



Obr. 29: Statistika

3.6.2 Statistika

NOPView umožňuje statisticky zpracovat rozdíly mezi hlavním a referenčním průmětem pomocí tří metod. Slouží k tomu dialogové okno **Statistika** (Obr. 29).

- Metoda A – Vypočítá poměrnou velikost plochy, kde oba průměty nelicují, vůči velikosti plochy referenčního průmětu. Počítá buď plochu, kde hlavní průmět přesahuje přes referenční, nebo plochu, kde referenční průmět přesahuje hlavní, nebo sjednocení těchto dvou ploch. Průmětem se rozumějí ty pixely NOP projekce, ve kterých se zobrazil povrch nějakého tělesa (vše kromě pozadí).
- Metoda B – Vypočítá průměrnou velikost vzájemné odchylky příslušných normál povrchu na společné části průmětu. Umožňuje nastavit maximální velikost odchylky, která se do průměru započítává. Tato hodnota sice říká, jak se v průměru blíží normály ideálnímu stavu, ale neříká nic o vizuální stránce výsledné projekce. Pro tento účel byla navržena speciální statistická charakteristika vypočítaná po stlačení tlačítka Směrodatná odchylka (viz. $Hlad(HP,RP)$, níže).
- Metoda C – Vygeneruje histogram odchylek normál na společné části průmětu. Rozdělí interval 0-180 od nuly po zadaném kroku na pravidelné intervaly. Vyjádří procentuální zastoupení odchylek normál v daných intervalech.

Aby bylo možné numericky zjistit, jak hladce se jeví povrch nějaké projekce, byl navržen speciální statistický ukazatel hladkosti $Hlad(HP,RP)$ nazvaný **Směrodatná odchylka změny rozdílu barevnosti**. Použitý byl při testování hladkosti povrchu koule zobrazované různými metodami. Necht' $HPz(i,j)$ a $RPz(i,j)$ jsou z-ové souřadnice příslušných normálových vektorů hlavní a referenční NOP projekce, i a j jsou souřadnice daného pixelu. Z-ová souřadnice míří

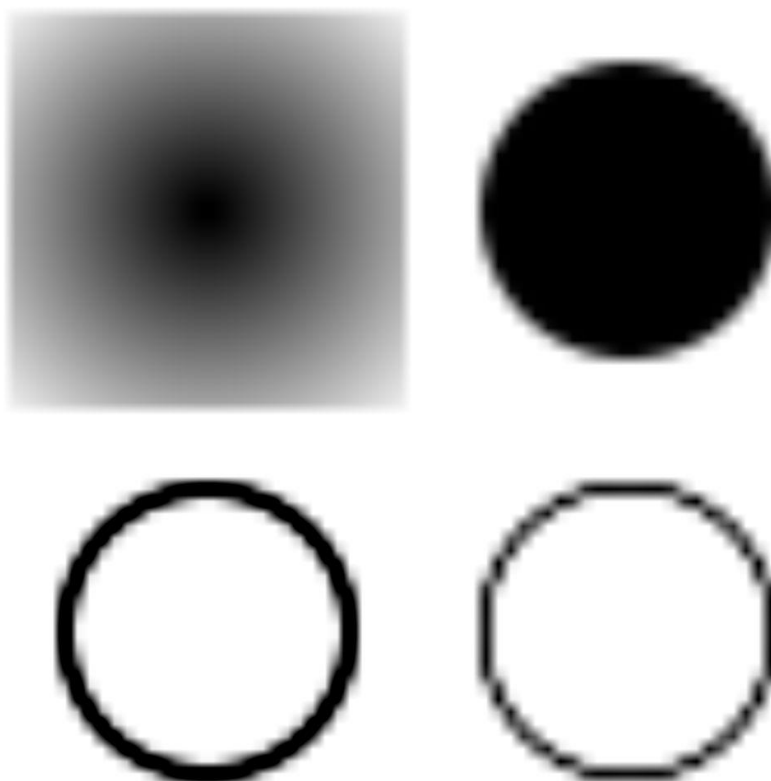
k pozorovateli, proto přímo ovlivňuje výsledně zobrazenou barvu, při použití čelního osvětlení. Necht' $dc(i,j)=HPz(i,j).255-RPz(i,j).255$. $dc(i,j)$ udává rozdíl v barevnosti při plném čelním osvětlení hlavním a referenční projekce. Cílem je vyjádřit stejnoměrnost barvy hlavní projekce vůči referenční. Na společném průmětu HP a RP se vypočítá statistický ukazatel $Hlad(HP,RP)$ jako společná směrodatná odchylka souboru hodnot $dc(i+1,j)-dc(i,j)$ a hodnot $dc(i,j+1)-dc(i,j)$. $Hlad(HP,RP)$ tedy říká, jak moc se mění rozdíl barevnosti hlavního a referenčního průmětu.

3.7 Srovnání implementovaných zobrazovacích metod

3.7.1 Použitá testovací data

Pro měření chování jednotlivých metod pro detekci povrchu a výpočet normál v různých situacích byla vygenerována pomocí nástroje TDC (Kap. 3.5) sada testovacích studií.

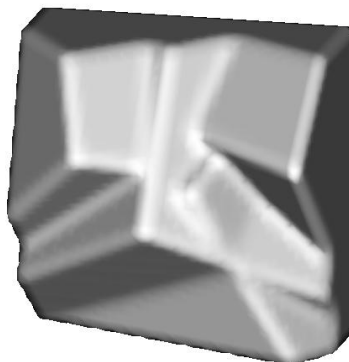
- Koule R=3 (plynulý přechod) matice intenzit 10x10x10
- Koule R=3 (ostrý přechod) matice intenzit 10x10x10
- Koule R=12 (plynulý přechod) matice intenzit 32x32x32
- Koule R=12 (ostrý přechod) matice intenzit 32x32x32
- Dutá koule R=3 T=1 matice intenzit 10x10x10
- Dutá koule R=3 T=0,5 matice intenzit 10x10x10
- Dutá koule R=12 T=1 matice intenzit 32x32x32
- Dutá koule R=12 T=0,5 matice intenzit 32x32x32
- Rovné povrchy s hranami (ostrý přechod) matice intenzit 32x32x32



Obr. 30: Ukázky rozložení intenzit testovacích studií R=12

Na obrázku Obr. 30 jsou ukázky řezů intenzitami kulových studií R=12. Duté koule jsou pomocí nástroje TDC vygenerovány z rozdílu dvou koulí o různých poloměrech s ostrým přechodem intenzity. Na obrázku Obr. 31 je pro ukádku zobrazena algoritmem MC isoplocha

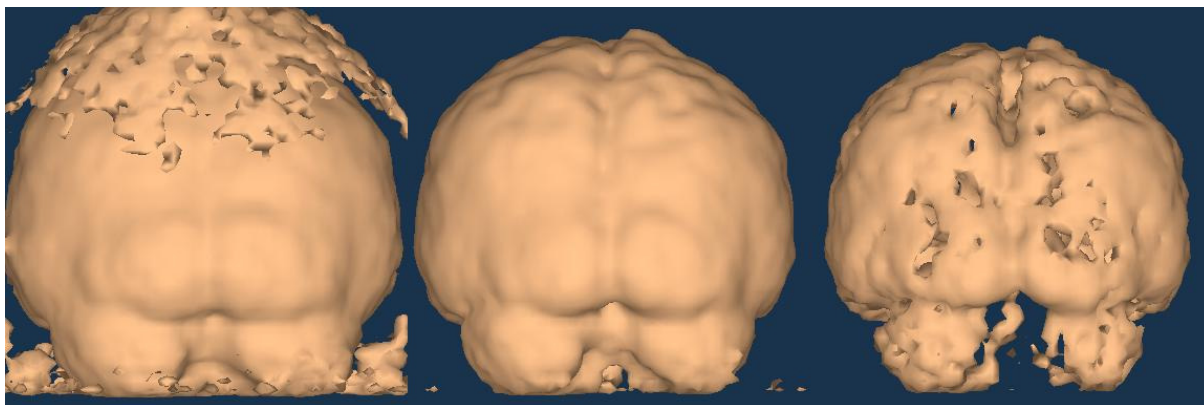
studie, kde intenzita vychází ze sloučení několika kvádrů s ostrým přechodem intenzity opět pomocí nástroje TDC. Z těchto testovacích studií byly vygenerovány NOP projekce o rozlišení 512x512 pixelů, na kterých proběhlo měření.



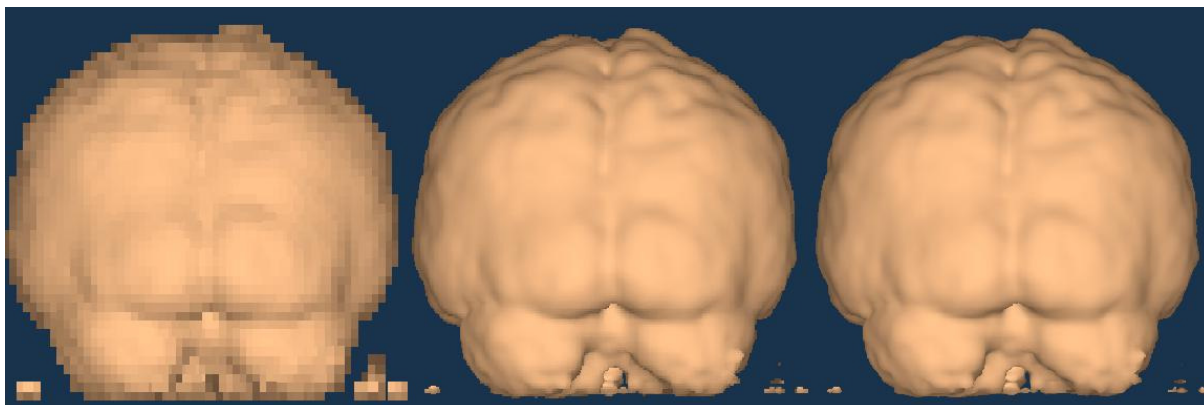
Obr. 31: Ukázka vizualizace testovací studie Rovné povrchy s hranami

3.7.2 Počet vrcholů povrchu použitých jednotlivými metodami

Jsou implementovány tři metody detekce povrchu, Marching cubes (MC), Marching tetrahedrons (MT) a Dividing cubes (DC). MC a TH používají pro rekonstrukci povrchu trojúhelníky a DC body povrchu. Počtem trojúhelníků (vrcholů) nebo počtem bodů povrchu je ovlivněna rychlost následného dalšího zpracování povrchu a práce s tímto modelem. Na ukázkovém modelu mozku byl měřen počet vrcholů nebo bodů povrchu generovaného jednotlivými metodami při různém nastavení intenzity povrchu. Na Obr. 32 a Obr. 33 jsou ukázky povrchu testovací studie určené pro měření.



Obr. 32: MC povrch intenzity 1500, 3500, 6000



Obr. 33: DC povrch intenzity 3500 s dělením/bodem 1/10, 5/2, 10/1

Výsledky měření pro různé metody a jejich nastavení a různé intenzity povrchu jsou v tabulce Tabulka 3. Počet trojúhelníků generovaných u stejného povrchu metodou TH je v průměru třikrát větší než u metody MC. Zatímco algoritmus MC generuje na jednu povrchovou „cube“ v průměru dva trojúhelníky, u metody TH to je 6 trojúhelníků. Metoda DC při dělení 1 generuje z principu algoritmu na každou „cube“ povrchu jeden bod. Počet bodů prostoru testovaných algoritmem DC při výpočtu povrchu roste s dělením kubicky, ale počet bodů povrchu roste pouze kvadraticky (téměř dokonale). Je to tím, že body reprezentují povrch a obsah plochy povrchu roste kvadraticky se zvětšováním rozlišení detekce. DC již při dělení 3 předstihne v počtu vrcholů MC. Při dělení 1 je DC vhodné pro interaktivní změny intenzity pro výpočet povrchu a okamžité zobrazování výsledku. Přesnost zobrazení je však nízká.

| Metoda | Dělení | Vel. bodu | Intenzita | Trojúh. | Vrcholů | Nárůst počtu | Očekávaný nárůst |
|--------|--------|-----------|-----------|---------|---------|--------------|------------------|
| MC | | | 1500 | 75004 | 225012 | | |
| | | | 3500 | 42900 | 128700 | | |
| | | | 6000 | 53384 | 160152 | | |
| TH | | | 1500 | 219992 | 659976 | | |
| | | | 3500 | 129432 | 388296 | | |
| | | | 6000 | 155268 | 465804 | | |
| DC | 1 | 10 | 1500 | | 36689 | | |
| | 1 | 10 | 3500 | | 21533 | | |
| | 1 | 10 | 6000 | | 26083 | | |
| | 2 | 5 | 1500 | | 142059 | 3,87 | 4,00 |
| | 2 | 5 | 3500 | | 84021 | 3,90 | 4,00 |
| | 2 | 5 | 6000 | | 102285 | 3,92 | 4,00 |
| | 3 | 3 | 1500 | | 317756 | 2,24 | 2,25 |
| | 3 | 3 | 3500 | | 188235 | 2,24 | 2,25 |
| | 3 | 3 | 6000 | | 229060 | 2,24 | 2,25 |
| | 4 | 3 | 1500 | | 562703 | 1,77 | 1,77 |
| | 4 | 3 | 3500 | | 333767 | 1,77 | 1,77 |
| | 4 | 3 | 6000 | | 406519 | 1,77 | 1,77 |
| | 5 | 2 | 1500 | | 878570 | 1,56 | 1,56 |
| | 5 | 2 | 3500 | | 521258 | 1,56 | 1,56 |
| | 5 | 2 | 6000 | | 634865 | 1,56 | 1,56 |
| | 10 | 1 | 1500 | | 3506126 | 3,99 | 4,00 |
| | 10 | 1 | 3500 | | 2082096 | 3,99 | 4,00 |
| | 10 | 1 | 6000 | | 2536860 | 4,00 | 4,00 |

Tabulka 3: Počty elementů povrchu

3.7.3 Geometrie povrchu

Jednou z metod porovnání vlastností povrchových zobrazovacích algoritmů je měření⁸ geometrie detekovaného povrchu. Geometrie byla měřena u zrekonstruovaného povrchu koule, kde se měřily tři parametry. Průměrná hodnota vzdáleností vrcholů detekovaného povrchu od středu koule, průměrná absolutní odchylka vrcholů od ideálního poloměru a nakonec směrodatná odchylka vzdáleností vrcholů od středu (Tabulka 4). Naměřené průměrné vzdálenosti od středu říkají, že všechny metody (až na DC 1) odhadly poloměr koule v průměru správně. Jediné DC s dělením 1 na kouli $R=3$ s ostrým i plynulým povrchem vykazuje větší chybu. Je to dáno velkým zakřivením koule a tím, že DC z principu nekoriguje polohy bodů povrchu podle intenzity jako ostatní metody. DC rozhoduje pro pevně dané body zda daný testovaný bod prostoru je či není bodem povrchu. To se projevilo i u průměrné odchylky od povrchu a hlavně u směrodatné odchylky povrchu. Nic na tom nezměnil ani poloměr testované koule ani plynulost intenzity na povrchu. S nárůstem dělení se hladkost a přesnost detekce zvyšuje ve všech ukazatelích a u ostrých přechodů předstihuje i TH. S hladkostí a přesností detekce na plynulých přechodech jsou ale MC a TH výrazně lepší než DC. TH u ostrého přechodu povrchu na MC zřetelně ztrácí.

⁸ Zkratky Ctrl+Shift+F12 nebo Ctrl+Shift+F11 v TomoAVI

| Metoda | Dělení | Průměrná vzdálenost od středu | | | | Průměrná odchylka od povrchu | | | | Směrodatná odchylka povrchu | | | |
|--------|--------|-------------------------------|-------|---------|-------|------------------------------|-------|---------|-------|-----------------------------|-------|---------|-------|
| | | | | | | | | | | | | | |
| MC | | 2,97 | 2,97 | 11,99 | 11,99 | 0,03 | 0,03 | 0,01 | 0,03 | 0,01 | 0,02 | 0,00 | 0,04 |
| TH | | 2,96 | 2,97 | 11,99 | 11,99 | 0,04 | 0,06 | 0,01 | 0,07 | 0,02 | 0,08 | 0,01 | 0,10 |
| DC | 1 | 3,26 | 3,26 | 12,01 | 12,01 | 0,42 | 0,42 | 0,38 | 0,38 | 0,42 | 0,42 | 0,44 | 0,44 |
| DC | 4 | 2,94 | 2,95 | 11,98 | 11,98 | 0,10 | 0,11 | 0,10 | 0,10 | 0,11 | 0,12 | 0,11 | 0,12 |
| DC | 8 | 2,92 | 2,92 | 11,98 | 11,98 | 0,08 | 0,08 | 0,05 | 0,06 | 0,06 | 0,07 | 0,05 | 0,07 |
| | | plynulý | ostrý | plynulý | ostrý | plynulý | ostrý | plynulý | ostrý | plynulý | ostrý | plynulý | ostrý |
| | | R=3 | R=3 | R=12 | R=12 | R=3 | R=3 | R=12 | R=12 | R=3 | R=3 | R=12 | R=12 |

Tabulka 4: Geometrie povrchu koule

Další měření geometrie detekovaného povrchu bylo provedeno na dutých koulích. Zde bylo provedeno měření směrodatné odchylky vzdáleností vrcholů vnějšího povrchu od středu koule (Tabulka 5). U duté koule detekují algoritmy kromě povrchu viditelného z vnější strany koule i vnitřní povrch oddělující volný prostor koule zevnitř. Pro změření směrodatné odchylky vrcholů vnějšího povrchu bylo potřeba rozhodnout, která část povrchu je vnitřní a která vnější. Za vnější povrch byl považován ten, který je od středu koule vzdálen více než je průměrná vzdálenost celého povrchu od středu koule. U DC s dělením 1 jsou opět viditelné značné odchylky. Při DC dělení 8 jsou naměřené odchylky srovnatelné s MC a opět předstihují TH. Otázkou je, proč dosahuje metoda TH na ostrých přechodech špatných výsledků. Je to způsobeno nevhodným rozložením tetraedrů uvnitř „cube“ (Kap. 4.8.2), které často způsobuje vznik ostrých nerovností na detekovaném povrchu (stačí na obrázku znázorňujícím rozložením tetraedrů přiřadit vrcholu nakreslenému nejbližší intenzitu 1 a ostatním intenzitu 0, pak detekovat povrch s intenzitou 0,5). Řešením by bylo jiné uspořádání tetraedrů, ale s tím by se zvýšil i tak dost vysoký počet detekovaných trojúhelníků. Marching tetrahedrons v této podobě mají oproti Marching cubes pouze nevýhody, ať už v počtech vrcholů, tak i v přesnosti.

| Metoda | Dělení | Směrodatná odchylka povrchu | | | |
|--------|--------|-----------------------------|-------|------|-------|
| | | | | | |
| MC | | 0,02 | 0,09 | 0,07 | 0,11 |
| TH | | 0,07 | 0,15 | 0,12 | 0,15 |
| DC | 1 | 0,29 | 0,23 | 0,31 | 0,32 |
| DC | 4 | 0,14 | 0,14 | 0,14 | 0,15 |
| DC | 8 | 0,10 | 0,11 | 0,10 | 0,12 |
| | | T=1 | T=0,5 | T=1 | T=0,5 |
| | | R=3 | R=3 | R=12 | R=12 |

Tabulka 5: Geometrie vnějšího povrchu duté koule

3.7.4 Normály povrchu

Je implementováno devět metod výpočtu normál povrchu modelu (Kap. 2.5). Metody SN a ASN lze použít pouze pro trojúhelníkové povrchy (výstup metod MC a TH). Ostatní lze zkombinovat s libovolnou metodou pro detekci povrchu. Z testovacích studií () Jednou použitou statistickou charakteristikou je průměrná hodnota odchylky normály modelu od ideální normály. Porovnávají se příslušné normály povrchu vygenerovaného kombinací implementovaných algoritmů a normály povrchu ideálního tělesa, podle kterého byla studie vygenerována. Průměrné odchylky neříkají nic o vizuální stránce výsledku, bylo potřeba definovat charakteristiku více lokální, popisující skutečný průběh normál povrchu modelu. Jednou takovou charakteristikou je Směrodatná odchylka změny rozdílu barevnosti dále jen SZR (Kap. 3.6.2) a klade si za cíl popsat jednou číselnou hodnotou výsledný vizuální dojem. Největší vypovídací hodnotu má v případě hladkých povrchů, kde se neočekávají přílišné odchylky normál od ideálu. Čím prudší změny na povrchu, tím je tato hodnota vyšší. Není příliš ovlivněna průměrnou chybou odchylky normál.

Naměřené srovnávací hodnoty pro koule R=3 a R=12 s ostrým a plynulým přechodem jsou v tabulce Tabulka 6. Měření bylo provedeno pro MC, TH, DC s dělením 5 a velikostí bodu 22 (případně 6 u R=12) a DC s dělením 40 a velikostí bodu 3.

Koule R=3 (plynulý přechod)

| Metody | Průměrná odchylka normály (stupně) | | | | Směrodatná odchylka změny rozdílu barevnosti | | | |
|--------|------------------------------------|--------|---------|---------|--|-------|---------|---------|
| | MC | TH | DC 5/22 | DC 40/3 | MC | TH | DC 5/22 | DC 40/3 |
| SN | 4,606 | 4,290 | | | 2,544 | 3,013 | | |
| ASN | 4,312 | 4,444 | | | 1,415 | 3,043 | | |
| GG3 | 11,837 | 11,923 | 11,658 | 11,658 | 1,421 | 1,466 | 2,980 | 1,638 |
| GG6 | 2,622 | 2,391 | 3,014 | 1,761 | 1,335 | 1,393 | 3,304 | 1,631 |
| AGG6 | 2,622 | 2,391 | 3,014 | 1,761 | 1,335 | 1,393 | 3,304 | 1,631 |
| GG26 | 2,626 | 2,357 | 3,021 | 1,706 | 1,350 | 1,407 | 3,297 | 1,635 |
| AGG26 | 3,454 | 3,077 | 3,396 | 2,449 | 1,408 | 1,435 | 3,413 | 1,658 |
| SOB | 2,594 | 2,332 | 3,003 | 1,678 | 1,347 | 1,404 | 3,298 | 1,634 |
| ASOB | 2,858 | 2,574 | 3,101 | 1,943 | 1,363 | 1,410 | 3,342 | 1,635 |

Koule R=3 (ostrý přechod)

| Metody | Průměrná odchylka normály (stupně) | | | | Směrodatná odchylka změny rozdílu barevnosti | | | |
|--------|------------------------------------|--------|---------|---------|--|-------|---------|---------|
| | MC | TH | DC 5/22 | DC 40/3 | MC | TH | DC 5/22 | DC 40/3 |
| SN | 5,823 | 12,024 | | | 2,697 | 5,648 | | |
| ASN | 9,860 | 17,145 | | | 1,953 | 8,141 | | |
| GG3 | 15,480 | 14,806 | 13,872 | 14,294 | 1,959 | 2,003 | 3,381 | 2,003 |
| GG6 | 6,171 | 5,234 | 4,723 | 4,292 | 1,619 | 2,113 | 3,493 | 1,991 |
| AGG6 | 6,171 | 5,234 | 4,723 | 4,292 | 1,619 | 2,113 | 3,493 | 1,991 |
| GG26 | 2,446 | 2,521 | 3,422 | 1,766 | 1,373 | 2,077 | 3,288 | 2,198 |
| AGG26 | 5,238 | 4,477 | 4,477 | 3,746 | 1,475 | 2,276 | 3,517 | 2,493 |
| SOB | 2,601 | 2,699 | 3,511 | 1,898 | 1,393 | 2,052 | 3,281 | 2,137 |
| ASOB | 2,740 | 2,646 | 3,472 | 1,972 | 1,384 | 2,108 | 3,335 | 2,240 |

Koule R=12 (plynulý přechod)

| Metody | Průměrná odchylka normály (stupně) | | | Směrodatná odchylka změny rozdílu barevnosti | | |
|--------|------------------------------------|--|--------|--|--|-----------|
| | MC | | DC 5/6 | MC | | DC 5/6 |
| SN | 0,650 | | | 0,806 | | |
| ASN | 0,451 | | | 0,513 | | |
| GG3 | 2,846 | | 3,172 | 0,407 | | 1,7398841 |
| GG6 | 0,171 | | 1,305 | 0,401 | | 1,7705379 |
| AGG6 | 0,171 | | 1,305 | 0,401 | | 1,7705379 |
| GG26 | 0,176 | | 1,307 | 0,401 | | 1,7704608 |
| AGG26 | 0,274 | | 1,323 | 0,431 | | 1,7784205 |
| SOB | 0,171 | | 1,306 | 0,401 | | 1,7704749 |
| ASOB | 0,207 | | 1,310 | 0,408 | | 1,7724994 |

Koule R=12 (ostrý přechod)

| Metody | Průměrná odchylka normály (stupně) | | | Směrodatná odchylka změny rozdílu barevnosti | | |
|--------|------------------------------------|--|--------|--|--|-----------|
| | MC | | DC 5/6 | MC | | DC 5/6 |
| SN | 2,516 | | | 1,592 | | |
| ASN | 3,165 | | | 1,654 | | |
| GG3 | 11,644 | | 12,359 | 1,619 | | 4,4201496 |
| GG6 | 5,404 | | 3,466 | 1,661 | | 3,136437 |
| AGG6 | 5,404 | | 3,466 | 1,661 | | 3,136437 |
| GG26 | 2,567 | | 2,476 | 0,838 | | 1,8524267 |
| AGG26 | 2,619 | | 2,536 | 0,887 | | 1,8815397 |
| SOB | 0,919 | | 1,553 | 0,740 | | 1,8215137 |
| ASOB | 0,939 | | 1,574 | 0,753 | | 1,8267933 |

Tabulka 6: Odchyly normál povrchu koule

Zde je stručné shrnutí výsledků měření z tabulky Tabulka 6. U modelu koule R=3 s plynulým přechodem dávají v průměrné odchylce i v SZR metody MC i TH srovnatelné výsledky. U modelu koule R=3 a ostrého přechodu dává TH horší výsledky u SZR. Je to dáno špatnou aproximací povrchu pomocí TH na ostrých přechodech. To dokládá i velmi výrazný nárůst SZR u TH pro normály povrchu vypočítané metodami založenými na geometrii povrchu. Metoda TH byla zde z dalšího testování vyloučena. DC 40 dává při tak velkém přiblížení (512x512 průmětna a studie 10x10x10) v průměrné odchylce dokonce lepší výsledky než MC, v praxi není ale možné tak vysoké dělení používat. DC 5/22 dává díky velkému bodu konstantně špatný výsledek SZR u ostrého i plynulého přechodu, průměrem je srovnatelná s MC. Metody MC a DC 5 s dělením 6 byly použity pro koule R=12. U plynulého přechodu dává MC bezkonkurenčně lepší výsledky díky velikosti bodu u DC. U ostrého přechodu jsou v průměrné odchylce MC a DC 5/6 téměř srovnatelné, ale v SZR je DC 5/6

horší. Výrazně špatných výsledků dosahuje DC 5/6 na ostrých přechodech s použitím normálové metody GG3, GG6 a AGG6.

Při porovnání výsledků metod pro výpočet normál je zřejmé, že metoda GG3 vykazuje značnou nepřesnost v odhadu normál. Je to dáno její nesymetričností z definice. Nedosahuje ani dobrých hodnot u SZR, dokonce s použitím DC vykazuje na ostrém přechodu s malým zakřivením velmi špatné výsledky. Metody SN a ASN dávají vždy špatné výsledky, pouze u ostrého přechodu na malém zakřivení předstihují v průměrné odchylce GG3, GG6, AGG6. ASN má oproti SN výrazně lepší výsledky v SZR téměř všude, zhoršuje však průměrnou odchylku. Konstantně dobrých výsledků dosahuje SOB a částečně i ASOB.

Naměřené srovnávací hodnoty pro duté koule R=3 a R=12 s ostrým a plynulým přechodem jsou v tabulce Tabulka 7. Měření bylo provedeno pro MC a TH. Zde je stručné shrnutí výsledků. Výsledky TH u tenkých objektů jsou výrazně špatné. Další komentář výsledků patří k metodě MC.

Metody SN a ASN mají na tenkém objektu výrazně dobré výsledky a to při R=3 i R=12. Metody GG3, GG6 a AGG6 mají výsledky špatné ve všech ukazatelích.

Dutá koule R=3

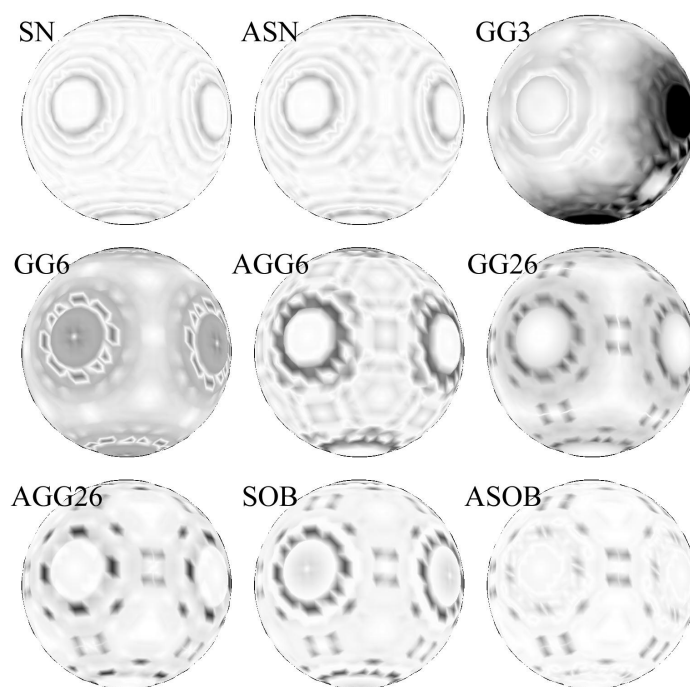
| Tloušťka | Průměrná odchylka normály (stupně) | | | | Směrodatná odchylka změny rozdílu barevnosti | | | |
|----------|------------------------------------|--------|--------|--------|--|--------|-------|--------|
| | T=1 | | T=0,5 | | T=1 | | T=0,5 | |
| Metody | MC | TH | MC | TH | MC | TH | MC | TH |
| SN | 5,611 | 19,102 | 5,793 | 26,890 | 2,722 | 11,635 | 1,234 | 10,990 |
| ASN | 9,760 | 22,663 | 5,625 | 26,207 | 1,913 | 13,092 | 1,221 | 11,415 |
| GG3 | 27,778 | 30,984 | 23,492 | 27,888 | 3,170 | 13,470 | 2,550 | 19,899 |
| GG6 | 14,175 | 18,174 | 15,576 | 22,116 | 2,101 | 13,749 | 2,040 | 18,253 |
| AGG6 | 8,900 | 15,598 | 7,387 | 16,910 | 2,066 | 15,637 | 2,019 | 19,312 |
| GG26 | 4,106 | 13,531 | 5,971 | 16,420 | 1,655 | 14,477 | 1,356 | 18,661 |
| AGG26 | 4,616 | 11,796 | 10,763 | 19,094 | 1,731 | 13,789 | 1,653 | 16,710 |
| SOB | 2,174 | 11,251 | 2,830 | 13,618 | 1,564 | 14,755 | 1,165 | 19,598 |
| ASOB | 2,501 | 12,196 | 7,958 | 18,311 | 1,660 | 15,121 | 1,433 | 18,613 |

Dutá koule R=12

| Tloušťka | Průměrná odchylka normály (stupně) | | | | Směrodatná odchylka změny rozdílu barevnosti | | | |
|----------|------------------------------------|--|--------|--|--|--|-------|--|
| | T=1 | | T=0,5 | | T=1 | | T=0,5 | |
| Metody | MC | | MC | | MC | | MC | |
| SN | 4,243 | | 5,647 | | 3,067 | | 3,735 | |
| ASN | 5,191 | | 6,738 | | 3,286 | | 3,761 | |
| GG3 | 26,097 | | 21,625 | | 5,150 | | 7,772 | |
| GG6 | 16,111 | | 17,348 | | 6,261 | | 9,402 | |
| AGG6 | 12,209 | | 13,799 | | 6,068 | | 8,176 | |
| GG26 | 13,434 | | 15,083 | | 4,152 | | 5,374 | |
| AGG26 | 8,694 | | 11,231 | | 5,066 | | 4,337 | |
| SOB | 8,556 | | 10,200 | | 5,428 | | 6,726 | |
| ASOB | 4,181 | | 8,027 | | 3,157 | | 4,710 | |

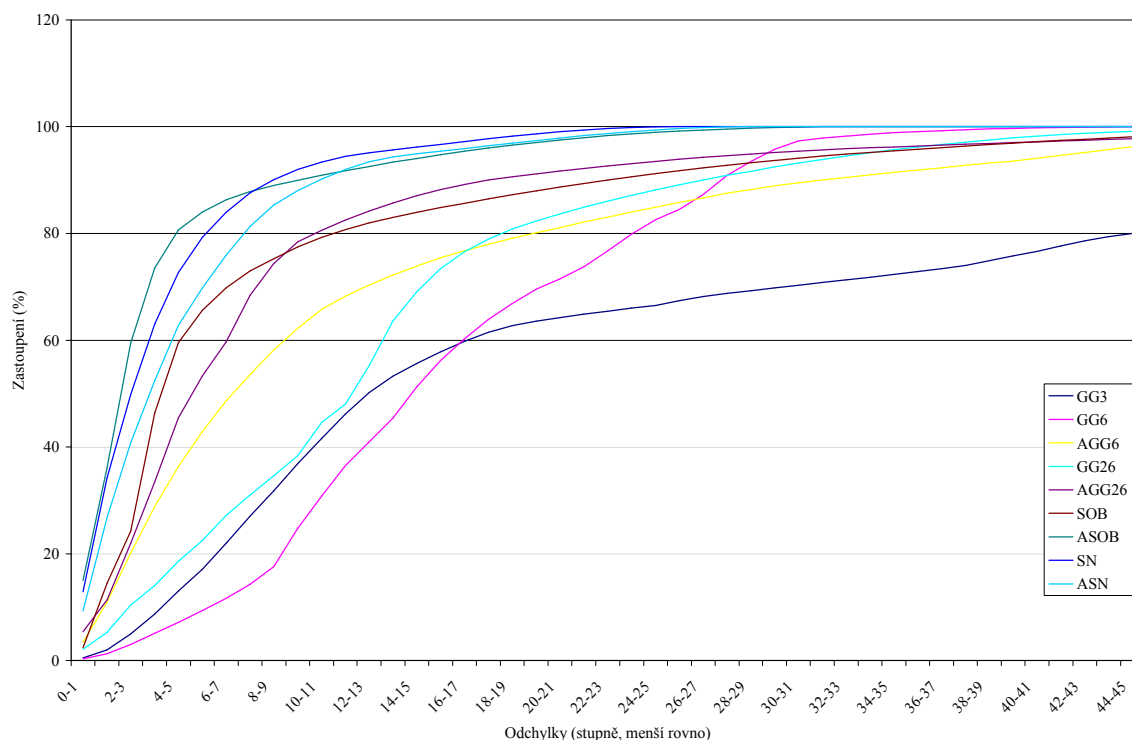
Tabulka 7: Odchylky normál povrchu duté koule

Odchylky normál povrchu duté koule R=12 T=1 jsou pro ukázkou znázorněny na obrázku Obr. 34 (použita metoda MC).



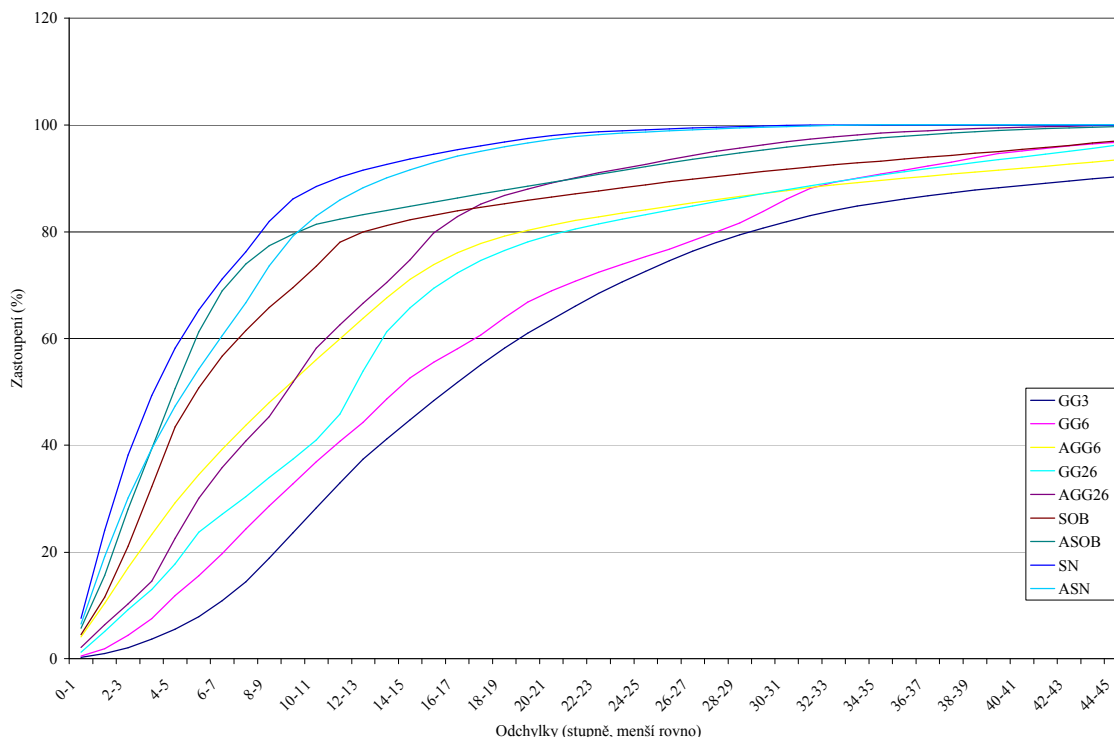
Obr. 34: Odchylky normál povrchu duté koule R=12 T=1

Graf (Obr. 35) zobrazuje upravený histogram odchylek normál na duté kouli R=12 tloušťky 1 při použití metody MC. Na vodorovné ose jsou vyznačeny odchylky ve stupních a na svislé ose je vyznačený procentuální počet pixelů v projekci, které mají odchylku menší rovnou hodnotě na vodorovné ose. Čím prudší vzestup a dosažení hodnoty 100, tím lépe. Výrazně špatné výsledky mají GG3, GG6, špatné výsledky mají AGG6 a GG26. Velmi dobré výsledky naopak mají ASOB, SN, ASN.



Obr. 35: Aditivní graf histogramů odchylek normál na duté kouli R=12 T=1

Graf (Obr. 36) zobrazuje upravený histogram odchylek normál na duté kouli R=12 tloušťky 0,5 při použití metody MC. Výrazně špatné výsledky mají GG3, GG6, GG26, špatné výsledky mají AGG6. Velmi dobré výsledky mají opět ASOB, SN, ASN.



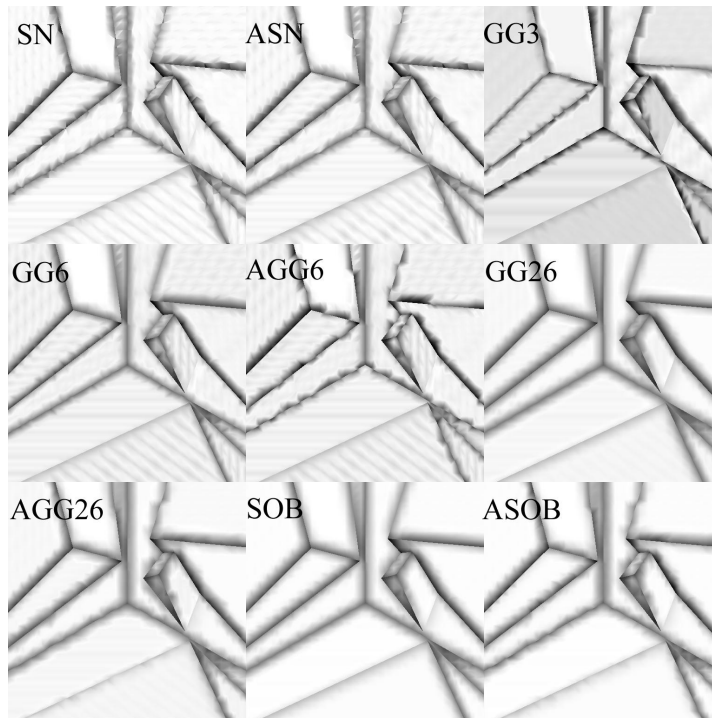
Obr. 36: Aditivní graf histogramů odchylek normál na duté kouli R=12 T=0,5

Další data, na kterých bylo odzkoušeno chování jednotlivých zobrazovacích metod, vznikly sloučením několika kvádrů s ostrým přechodem intenzity. Kvádry byly natočeny pod různými úhly. Na těchto datech se testovalo chování metod na rovných plochách procházejících prostorem pod různým úhlem. V tabulce (Tabulka 8) jsou naměřené hodnoty pro metody MC a DC s dělením 8 a velikostí bodu 4. Zvolené statistické charakteristiky se zde ukazují jako nedostatečné, nedokáží totiž dostatečně odlišit kvality jednotlivých normálových metod (MC Obr. 37 a DC Obr. 39). Je to způsobeno vlivem odchylek na hranách, kde dochází ke značným chybám. Ani omezení maximální velikosti odchylky pro započítávání do statistik nedává srozumitelné výsledky. Řešení je v porovnání aditivních verzí histogramů jednotlivých metod (MC Obr. 38 a DC Obr. 40).

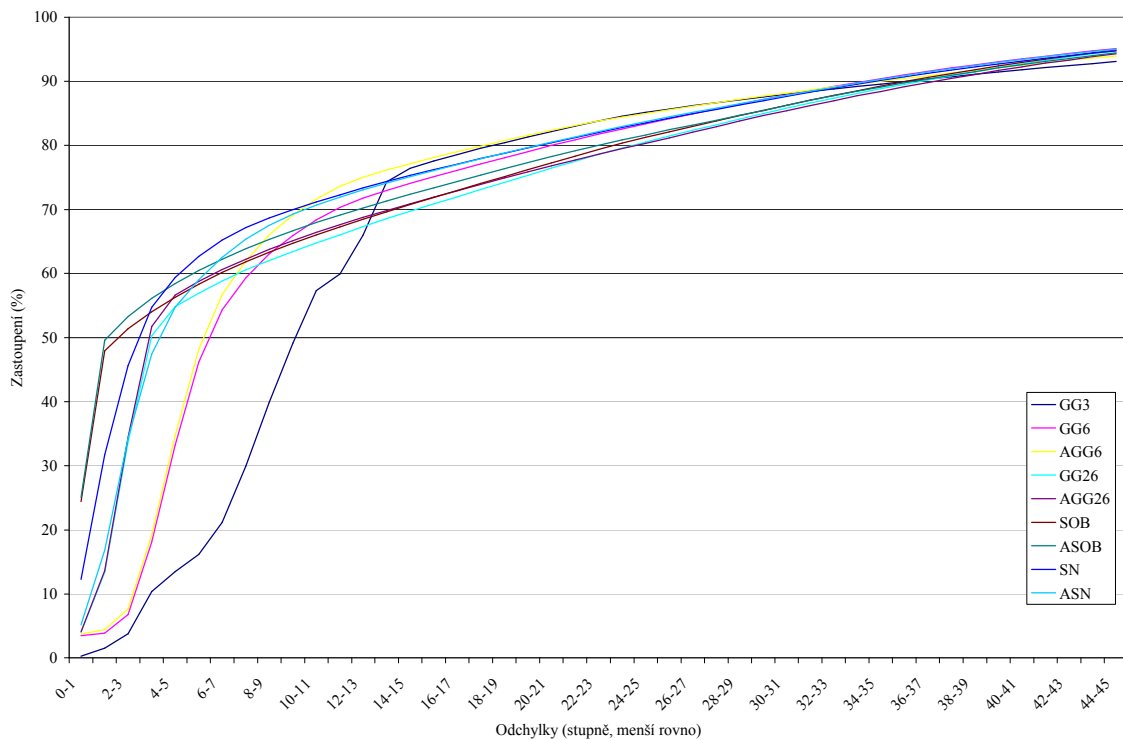
Rovné povrchy s hranami (ostrý přechod)

| Odchylky | Průměrná odchylka normály (stupně) | | | | Směrodatná odchylka změny rozdílu barevnosti | | | | | |
|----------|------------------------------------|--------|----------------|--------|--|--------|----------------|--------|----------------|--------|
| | všechny odchylky | | max. 15 stupňů | | všechny odchylky | | max. 22 stupňů | | max. 10 stupňů | |
| Metody | MC | DC 8/4 | MC | DC 8/4 | MC | DC 8/4 | MC | DC 8/4 | MC | DC 8/4 |
| SN | 11,005 | | 3,478 | | 4,536 | | 2,392 | | 1,528 | |
| ASN | 11,511 | | 4,203 | | 4,410 | | 2,229 | | 1,415 | |
| GG3 | 15,184 | 15,317 | 8,712 | 7,224 | 4,349 | 5,951 | 1,667 | 4,045 | 1,119 | 3,484 |
| GG6 | 12,873 | 11,480 | 5,878 | 3,746 | 4,237 | 4,996 | 2,229 | 2,880 | 1,210 | 2,228 |
| AGG6 | 12,667 | 11,184 | 5,825 | 3,666 | 4,325 | 5,062 | 2,265 | 2,892 | 1,285 | 2,280 |
| GG26 | 12,635 | 12,582 | 4,053 | 3,697 | 4,109 | 4,559 | 1,603 | 2,357 | 1,131 | 1,626 |
| AGG26 | 12,495 | 12,421 | 3,957 | 3,612 | 4,238 | 4,806 | 2,666 | 3,073 | 1,362 | 1,856 |
| SOB | 11,578 | 11,642 | 2,964 | 2,788 | 4,084 | 4,509 | 1,490 | 2,187 | 0,977 | 1,367 |
| ASOB | 11,314 | 11,342 | 2,862 | 2,670 | 4,143 | 4,633 | 2,235 | 2,449 | 1,077 | 1,450 |

Tabulka 8: Odchylky normál rovných povrchů a hran

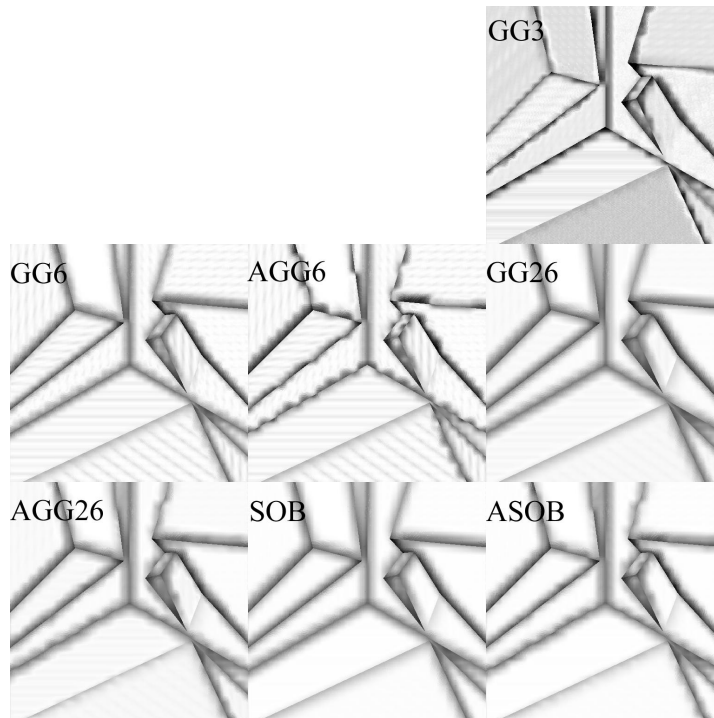


Obr. 37: MC - Odchylky normál rovných povrchů a hran

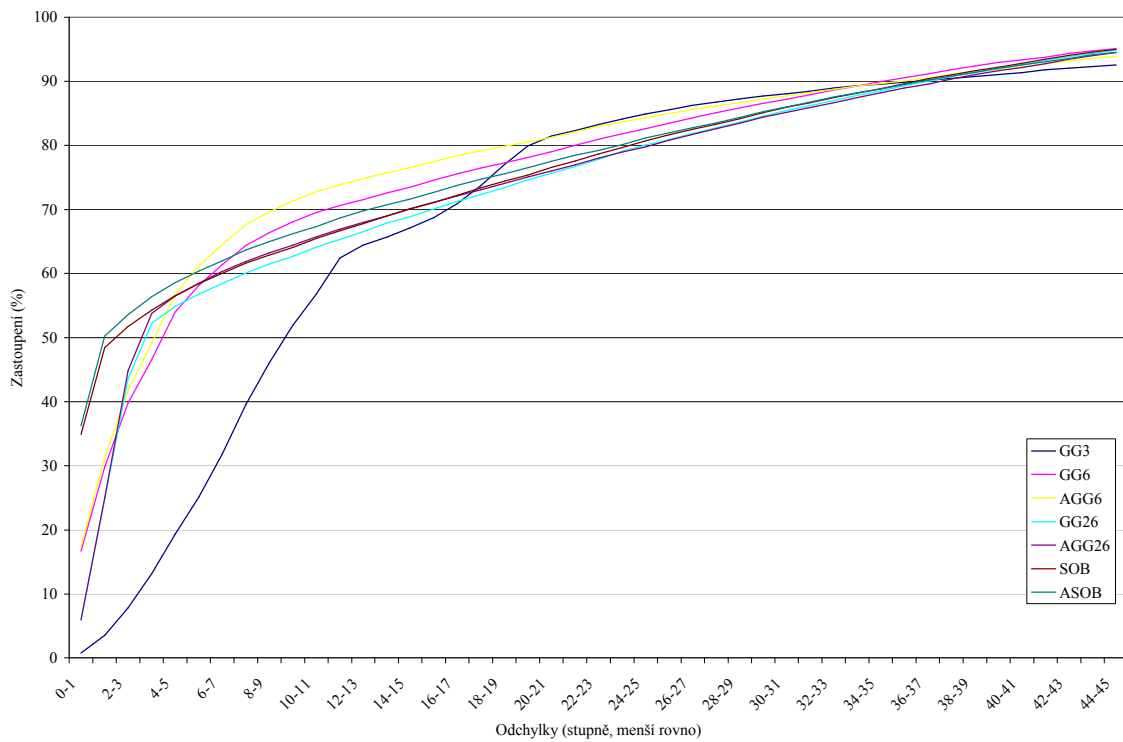


Obr. 38: MC - Aditivní graf histogramů odchylek normál rovných povrchů a hran

Špatné výsledky při použití metody MC opět vykazují GG3, GG6 a AGG6. Dobré výsledky naopak vykazují ASOB a SOB. Histogram neříká nic o hladkosti zobrazení a tak třetí nejlepší metodou je SN. GG26 a AGG26 produkují hladší výsledek v rovině, ale mají velké průměrné odchylky.



Obr. 39: DC - Odchyly normál rovných povrchů a hran



Obr. 40: DC – Aditivní graf histogramů odchylek normál rovných povrchů a hran

Špatné výsledky na těchto datech při použití metody DC opět vykazuje GG3. Dobré výsledky opět vykazují ASOB a SOB.

3.8 Shrnutí výsledků

Porovnávaly se tři metody detekce povrchu implementované v aplikaci TomoAVI, jsou to Marching cubes (MC), Marching tetrahedrons (MT) a Dividing cubes (DC). Zároveň se porovnávalo devět metod pro výpočet normál povrchu (Kap. 2.5).

Metoda TH detekuje povrch s velkým počtem trojúhelníků, ale nedosahuje ani kvalit metody MC, která generuje trojúhelníků v průměru třikrát méně. Metoda TH dává na ostrých přechodech špatné výsledky. Je to způsobeno nevhodným rozložením tetraedrů, které často způsobuje vznik ostrých nerovností na detekovaném povrchu. Metoda DC je vhodná pro rychlý výpočet a zobrazení modelu v případě, že dělení je maximálně 2. U většího dělení předstihne metoda DC metodu MC v počtu použitých vrcholů. Výsledky zobrazení metody DC nejsou pochopitelně příliš dobré pokud je větší nárůst velikosti bodu.

Porovnáním výsledků metod pro výpočet normál je zřejmé, že nejhorší metodou je GG3. Ve srovnání s ostatními metodami má mnohonásobně větší průměrnou odchylku. To vyplývá již z nesymetričnosti odhadu. Většinou dosahují velmi špatných výsledků metody GG6 a AGG6. Překvapivě narozdíl od GG26 stabilně dobré výsledky vykazuje metoda SOB. Spolu s metodou ASOB mají dobré výsledky na ostrých přechodech. Metoda ASOB pak dosáhla lepších na tenkém objektu. Také metody SN a ASN jsou dobře použitelné pro tenké objekty, jinde dosahují průměrných výsledků.

3.9 Závěr srovnávací části

Provedeným srovnáním byla odhalena slabá místa některých implementovaných metod zobrazení povrchu. Některé metody se ukázaly přímo jako nevhodné. Například metoda Marching tetrahedrons se ukázala, v podobě v jaké byla implementována, jako nevyhovující. Některé metody vyšly ze srovnávacích testů velice dobře, pro směřodatnější ověření chování by bylo potřeba provést mnohem více testů na mnohem větším souboru dat, to by vyžadovalo automatizované generování testovacích dat a automatické vyhodnocování výsledků. V tom případě by bylo potřeba definovat jiná srovnávací kritéria fungující na obecnějších datech. Ideální výsledek zobrazení je těžko definovatelný, vstupní data prostorových studií nesou málo informací na dokonalou rekonstrukci povrchu. Algoritmus nemá tušení, co data reprezentují a rozlišovací schopnost končí na úrovni voxelu. Bylo by jistě možné navrhnout algoritmus, který by v datech hledal geometrické tvary a podle nich generoval povrch i normály. V zobrazování medicínských dat však toto řešení nemá opodstatnění.

4 Ovládání aplikace

Aplikační modul TomoAVI pro tomografické prezentace. Verze 2.1

Požadavky modulu TomoAVI na hardware a software

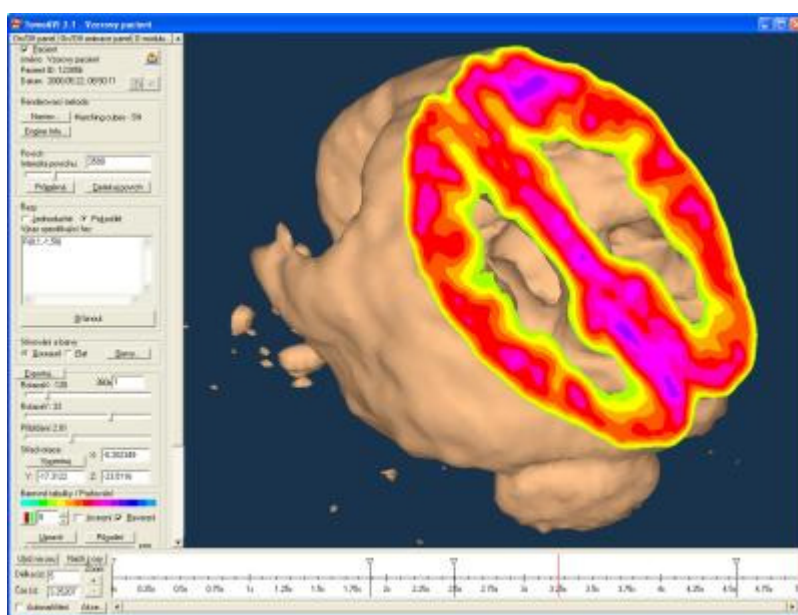
- **Software** - Operační systém **Windows 98, Me, 2000, XP nebo novější**. Fungující grafické rozhraní **OpenGL**. Modul TomoAVI využívá především knihovny OpenGL Client a OpenGL Utility Library. Doporučeno rozlišení minimálně **800x600**. Barevná hloubka musí být alespoň **16bitů (64k barev)**. Pro funkci exportování animací do formátu AVI je potřeba knihovna **toAvi.dll**, která se instaluje spolu s tímto aplikačním modulem. K zobrazení vyexportovaných VRML souborů je nutné mít nainstalován **VRML klient** (například volně šiřitelný VRML klient Cortona společnosti ParallelGraphics). Modul lze spustit i na Windows 95 a NT 4.0, ale je nutné doinstalovat podporu pro help (Internet Explorer 4.0 a hhupd_old.exe, nebo pouze Internet Explorer 5.0 a vyšší) a do Windows 95 se musí nainstalovat OpenGL rozhraní (opengl95.exe). Příslušné exe soubory a instalace Internet Exploreru naleznete na stránkách společnosti Microsoft.
- **Hardware** - Platforma **x86**. Výkon procesoru a velikost operační paměti podle objemu zpracovávaných tomografických dat a nastavení vlastností zobrazení modelu. Doporučeno alespoň **128MB RAM** a hardwarově **akcelerované OpenGL**.

Funkce modulu

Aplikační modul TomoAVI slouží k vizualizaci proměnlivé intenzity záření v prostoru. Nejprve je detekován podprostor určený prahovou intenzitou (Kap. 4.2.3). Ten je vymezen oblastmi s intenzitou vyšší než prahová. Tato metoda se používá především pro vizualizaci tomografických dat. Abychom měli přehled i o oblastech s vyšší intenzitou, než je prahová, je možné specifikovat řezy, jimiž je model rozřezán (Kap. 4.5). Na řezech se zobrazí intenzity v daném místě obarvené danou paletou (Kap. 4.2.8).

K dispozici je několik metod detekce povrchu modelu a jeho zobrazení (Kap. 4.8). Pro účely diplomové práce je implementován export do **NOP** formátu (Kap. 4.2.6).

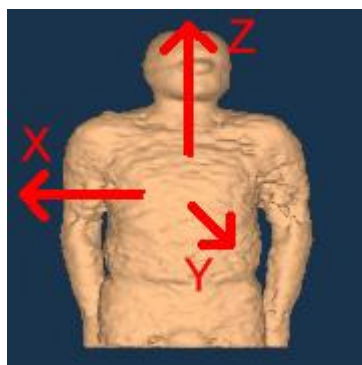
Výsledný obrázek lze exportovat (Kap. 4.2.6) mnoha způsoby a následně zpracovávat v jiných programech. Většinu parametrů lze měnit v čase pomocí časové osy, tím lze dosáhnout animování modelu (Kap. 4.3). Takový model můžete exportovat do animace AVI (Kap. 4.6). Nejčastěji používané uživatelské postupy jsou stručně popsány v užitečných tipech (Kap. 4.11).



Obr. 41: Aplikační modul TomoAVI

4.1 Souřadná soustava

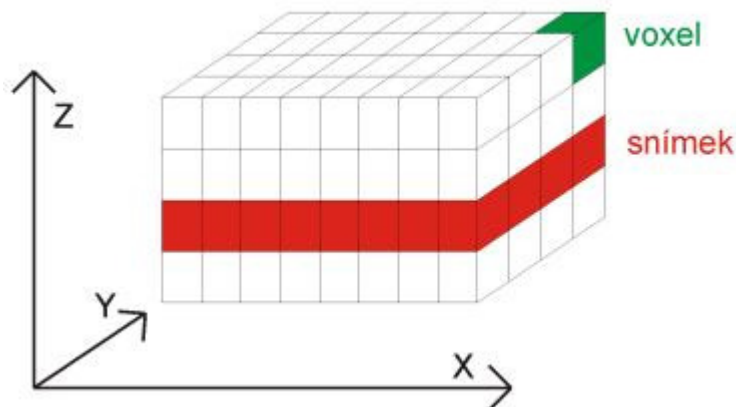
Pro manipulaci s modelem a především pro řezy modelem je potřeba orientovat se v souřadném systému 3D modelu. Je zde použita klasická **kartézská souřadná soustava**. Počátek soustavy souřadnic **(0,0,0)** je umístěn do středu 3D matice dat studie. Jednotka na každé ose odpovídá délce 1mm ve skutečnosti. Vstupní data většinou obsahují informaci o měřítku modelu, ale může se stát, že neodpovídají realitě, je proto důležité, dávat si na toto pozor. Při spuštění modulu TomoAVI je pohled nastaven tak, že **osa x míří doleva, osa z nahoru** a **osa y proti pozorovateli**. Tento směr os odpovídá zaběhlému standardu v medicíně. Na pacienta-model se díváme zepředu, jako by ležel na vyšetřovacím lůžku a my se dívali shora.



Obr. 42: Souřadný systém 3D modelu

3D Matice hodnot intenzity záření v prostoru

Vstupními daty modulu TomoAVI je trojrozměrné pole hodnot udávající rozložení zkoumaného záření v prostoru. Jednotlivé vrstvy podél **osy z** se nazývají **snímky** a jednotlivé položky matice dat se nazývají **voxely**. Hodnota voxelu vyjadřuje průměrnou hodnotu záření uvnitř voxelu. Voxel nemusí mít stejnou velikost ve všech směrech.



Obr. 43: Pole intenzit

4.2 Panel hlavních ovládacích prvků

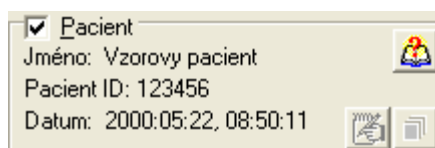
Nachází se v levé části okna modulu TomoAVI. Lze schovat nebo zobrazit pomocí tlačítka **On/Off panel**. Zkratka F11. Slouží k nastavování parametrů modelu. Pokud chcete pracovat s modelem v průběhu času (vytváření animací), použijte panel ovládacích prvků pro tvorbu animací (Kap. 4.3).



Obr. 44: Panel hlavních ovládacích prvků

4.2.1 Pacient

Oblast **Pacient** je určena pro zobrazení údajů o pacientovi.



Obr. 45: Oblast pacient

Pacient

Pomocí tohoto zaškrťovacího políčka lze zobrazit/skrýt údaje o pacientovi. Implicitně jsou údaje skryty. Zkratka Alt+P. Ukázka je na obrázku 45.

Jméno, Pacient ID a Datum

Příslušné údaje o pacientovi. Ukázka je na obrázku 45.

Tlačítko

Zobrazí okno nápovědy s užitečnými tipy (Kap. 4.11). Zkratka Ctrl+U. Ukázka je na obrázku 45.

Tlačítko

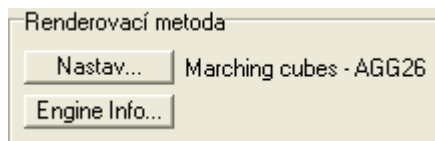
Zobrazí okno určené k editaci textových poznámek k dané studii (Viz. CD). Tato funkce není dostupná v off-line módu (Viz. CD). Zkratka Ctrl+T. Ukázka je na obrázku 45.

Tlačítko

Zobrazí okno se screenshoty (Viz. CD). Tato funkce není dostupná v off-line módu (Viz. CD). Zkratka Ctrl+O. Ukázka je na obrázku 45.

4.2.2 Renderovací metoda

Oblast **Renderovací metoda** je určena k nastavení metody výpočtu 3D modelu a jeho zobrazení a též k získání bližších informací o zobrazeném 3D modelu.



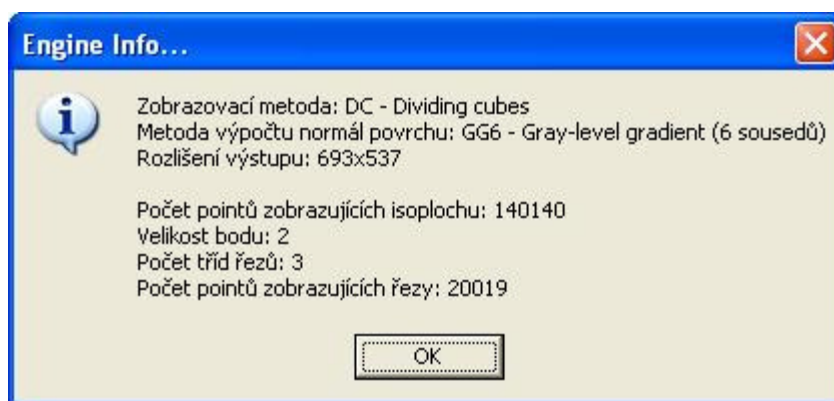
Obr. 46: Oblast renderovací metoda

Nastav...

Zobrazí okno určené k nastavení metody generování a zobrazování povrchu 3D modelu a k nastavení metody pro výpočet normál povrchu (Kap. 4.8). Normály povrchu slouží pro výpočet osvětlení modelu. Vedle tlačítka je informace o aktuální metodě zobrazení a výpočtu normál (zde například Marching cubes - AAG26). Zkratka Ctrl+Z. Ukázka je na obrázku 46.

Engine Info...

Zobrazí okno se základními informacemi o 3D modelu. Zkratka Ctrl+I. Ukázka je na obrázku 46.



Obr. 47: Okno s informacemi o 3D modelu

4.2.3 Povrch

Tato oblast je určena pro nastavování parametrů pro detekci povrchu modelu. Více se dozvíte v kapitole 4.



Obr. 48: Oblast povrch

Intenzita povrchu

Hodnota intenzity podle které se detekuje povrch modelu. Pro novou detekci povrchu zadejte hodnotu a stiskněte Enter, nebo tlačítko Detekuj povrch. Zkratka F5. Ukázka je na obrázku 48.

Posuvník nacházející se pod editačním polem intenzity povrchu slouží k plynulé a okamžité změně intenzity detekovaného povrchu. V závislosti na nastavené renderovací metodě (Kap. 4.8) a aktuální intenzitě může posuvník reagovat s velkým zpožděním. Zkratka Ctrl+F5. Ukázka je na obrázku 48.

Průměrná

Do políčka Intenzita povrchu vyplní průměrnou hodnotu intenzity nacházející se v datech studie. Zkratka Alt+M. Ukázka je na obrázku 48.

Detekuj povrch

Přepočítá a zobrazí model s povrchem podle nové intenzity (viz. Intenzita povrchu). Zkratka Alt+D. Ukázka je na obrázku 48.

4.2.4 Řezy

Tato oblast je určena pro specifikování řezů modelem. Více se o řezech dozvíte v kapitole 4.5.

Jednoduché a Pokročilé

Řezy modelem lze zadávat dvěma způsoby.

- Jednoduchý způsob je určen pro rychlé zadávání maximálně tří na sebe kolmých řezů.



Obr. 49: Jednoduché řezy

- Pokročilé zadávání umožňuje zkušenému uživateli nadefinovat prakticky libovolné řezy pomocí řezového výrazu (Kap. 4.5).



Obr. 50: Pokročilé řezy

Pomocí políček **Jednoduché** a **Pokročilé** lze přepínat mezi způsoby zadávání. Zkratky Alt+J a Alt+K.

Řez X, Y a Z

Udává polohu řezu kolmého na danou osu (viz. kap. 4.1). Hodnota je udávána v milimetrech (pokud je měřítko udané v datovém souboru správné!). Pokud je políčko prázdné, řez se nepočítá. Zkratka F6. Ukázka zadávání jednoduchých řezů je na obrázku 49.

Plus nebo Minus

Udává orientaci daného řezu. Pokud je na tlačítku napsáno **Plus**, neořezává se část modelu v kladném poloprostoru (Kap. 4.1). Pokud je na tlačítku napsáno **Minus**, neořezává se část modelu v záporném poloprostoru (Kap. 4.1). Ukázka zadávání jednoduchých řezů je na obrázku 49.

Vyseknout klín nebo Zobrazit klín

Při zadání více než jednoho řezu se musí specifikovat způsob odfiltrování části modelu. V případě Jednoduchého zadávání jsou možnosti dvě.

- Vyseknout klín znamená, že se ve výsledku odfiltruje pouze ta část, která je odfiltrována všemi řezy zároveň.
- Zobrazit klín znamená, že se ve výsledku odfiltruje to, co je odfiltrováno alespoň jedním řezem.

Pomocí políček **Vyseknou klín** a **Zobrazit klín** lze přepínat mezi způsoby filtrování. Ukázka zadávání jednoduchých řezů je na obrázku 49.

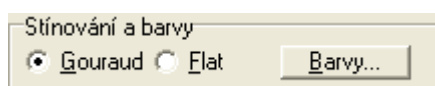
Oříznout

Toto tlačítko slouží k přepočítání modelu s novými parametry řezů. Stejného efektu docílíte stiskem klávesy Enter při zadávání polohy řezů. Zkratka Alt+O. Ukázka zadávání řezů je na obrázku 49.

Výraz specifikující řez

Slouží k zadávání řezového výrazu v případě Pokročilého zadávání. Více se o pokročilých řezech dozvíte v kapitole 4.5. Zkratka F6. Ukázka zadávání pokročilých řezů je na obrázku 50.

4.2.5 Stínování a barvy



Obr. 51: Oblast stínování a barvy

Gouraud a Flat

Tento ovládací prvek není dostupný při nastavené renderovací metodě Dividing cubes (Kap. 4.8.3). Další informace se metody Dividing cubes netýkají.

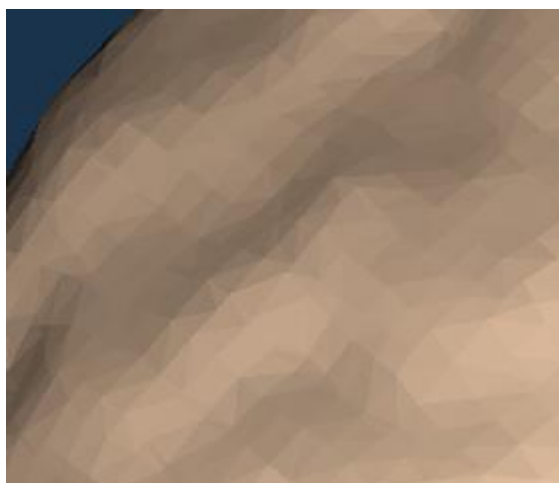
Povrch 3D modelu sestává z velkého množství trojúhelníků. Tyto trojúhelníčky lze (kromě metody Dividing cubes) zobrazovat dvěma způsoby. Osvětlení a tudíž i výsledná barva se počítá buď pro celý trojúhelníček nebo pro každý vrchol zvlášť.

- **Gouraud** je implicitní způsob zobrazení povrchu. Povrch zobrazený touto metodou se jeví hladší. Osvětlení a tedy i barva se počítá pro každý vrchol zvlášť a uvnitř trojúhelníčku se barva vypočítá lineární interpolací.



Obr. 52: Ukázka Gouraud stínování

- **Flat** metoda počítá osvětlení a barvu stejnou pro celý trojúhelníček. Vyniknou tak hrany na povrchu mezi jednotlivými trojúhelníčky způsobené digitálním zpracováním tomografických dat.



Obr. 53: Ukázka Flat stínování

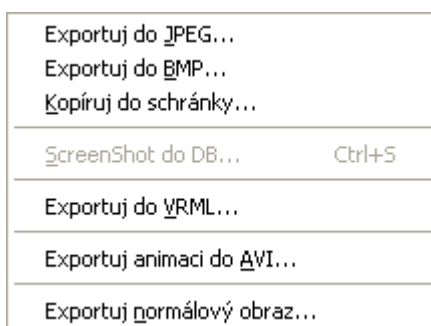
Pomocí políček **Gouraud** a **Flat** lze přepínat mezi způsoby zobrazování. Zkratky Alt+G a Alt+F. Ukázka je na obrázku 51.

Barvy...

Zobrazí okno určené k nastavení barev povrchu modelu, jeho osvětlení a podkladu renderovací plochy (Kap. 4.4). Zkratka Alt+B. Ukázka je na obrázku 51.

4.2.6 Exportuj...

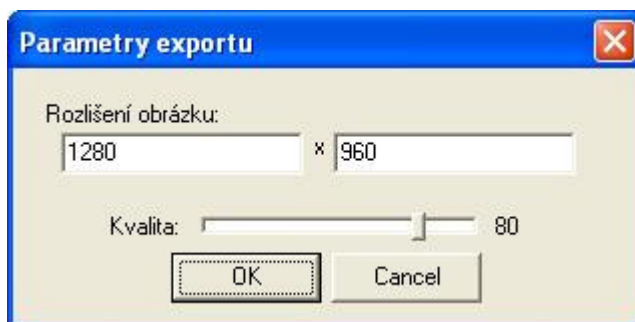
Zobrazí menu pro výběr typu exportu. Zkratka Alt+E. Toto tlačítko naleznete na obrázku 60.



Obr. 54: Menu pro výběr typu exportu

Exportuj do JPEG

Umožňuje exportovat aktuálně zobrazený model jako obrázek do souboru typu JPEG. Zobrazí nejprve okno pro zadání parametrů exportu. Ukázka menu je na obrázku 54.

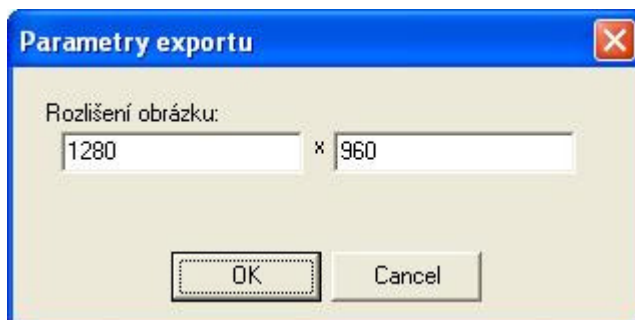


Obr. 55: Parametry exportu do JPEG

- **Rozlišení obrázku.** Umožňuje zadat rozměr obrázku v obrazových bodech (pixelech). Ukázka je na obrázku 55.
- **Kvalita.** Slouží pro zadání míry komprese souboru JPEG. Čím menší kvalita, tím menší soubor. Ukázka je na obrázku 55.

Exportuj do BMP

Umožňuje exportovat aktuálně zobrazený model jako obrázek do souboru typu BMP. Zobrazí nejprve okno pro zadání parametrů exportu. Ukázka menu je na obrázku 54.

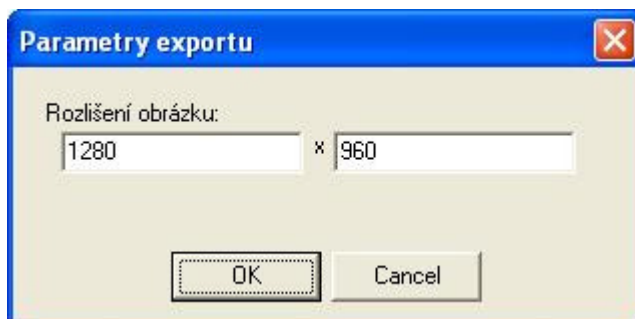


Obr. 56: Parametry exportu do BMP

- **Rozlišení obrázku.** Umožňuje zadat rozměr obrázku v obrazových bodech (pixelech). Ukázka je na obrázku 56.

Kopíruj do schránky

Umožňuje exportovat aktuálně zobrazený model jako obrázek do schránky, odkud ho můžete vložit do jiného programu pro další zpracování. Zobrazí nejprve okno pro zadání parametrů exportu. Ukázka menu je na obrázku 54.

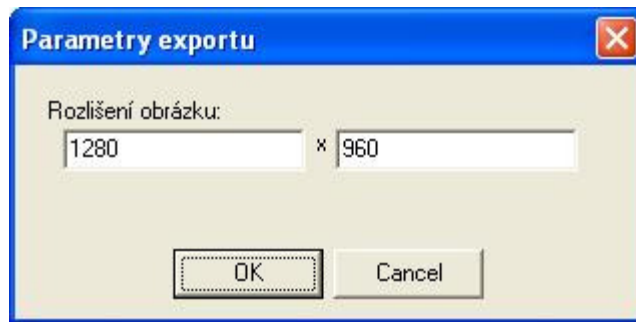


Obr. 57: Parametry exportu do schránky

- **Rozlišení obrázku.** Umožňuje zadat rozměr obrázku v obrazových bodech (pixelech). Ukázka je na obrázku 57.

ScreenShot do DB

Umožňuje exportovat aktuálně zobrazený model jako ScreenShot do databáze. Zobrazí nejprve okno pro zadání parametrů exportu. Tato funkce není dostupná v off-line módu (Viz. CD). Více se o ScreenShotech dozvíte v kapitole 7.1. Zkratka Ctrl+S. Ukázka menu je na obrázku 54.



Obr. 58: Parametry exportu screenshotu do DB

- **Rozlišení obrázku.** Umožňuje zadat rozměr ScreenShotu v obrazových bodech (pixelech). Ukázka je na obrázku 58.

Exportuj do VRML

Zobrazí okno pro export 3D modelu do VRML (Kap. 4.7). Tato funkce není dostupná pro renderovací metodu Dividing cubes (Kap. 4.8.3). Ukázka menu je na obrázku 54.

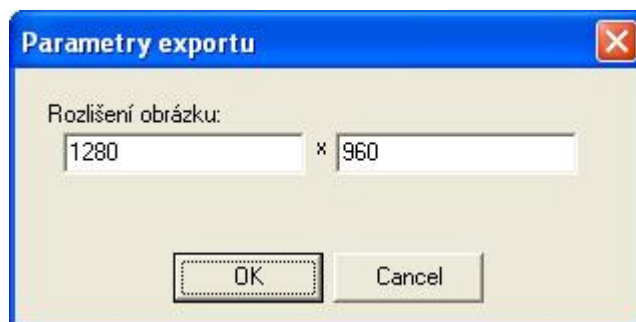
Exportuj animaci do AVI

Zobrazí okno pro vytváření animací z časové osy (Kap. 4.6). Ukázka menu je na obrázku 54.

Exportuj normálový obraz

Umožňuje exportovat aktuálně zobrazený model jako normálový obraz do souboru typu NOP (Normal picture). Zobrazí nejprve okno pro zadání parametrů exportu. Ukázka menu je na obrázku 54.

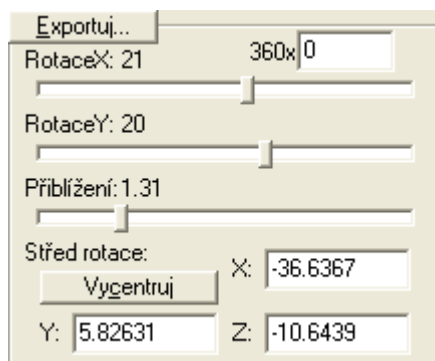
Export do NOP je v modulu TomoAVI implementován pro účely diplomové práce. Je to speciální formát ukládající přímo normály povrchu a používá se pro porovnávání kvalitativních vlastností jednotlivých renderovacích metod (Kap. 4.8).



Obr. 59: Parametry exportu do NOP

- **Rozlišení obrázku.** Umožňuje zadat rozměr obrázku v obrazových bodech (pixelech). Ukázka je na obrázku 59.

4.2.7 Posun a rotace modelu



Obr. 60: Oblast Posun a rotace modelu

RotaceX

Posuvník sloužící k otáčení modelu podle svislé osy. Zkratka F7. Ukázka je na obrázku 60.

RotaceY

Posuvník sloužící k otáčení modelu podle vodorovné osy. Zkratka F8. Ukázka je na obrázku 60.

Přiblížení

Posuvník sloužící k zvětšování a zmenšování modelu. Zkratka F9. Ukázka je na obrázku 60.

360x

Políčko udávající počet celých otočení o 360 stupňů podle svislé osy. Zkratka Ctrl+F7. Ukázka je na obrázku 60.

Vycentruj

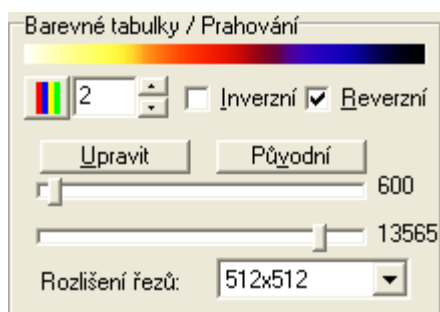
Nastaví střed otáčení podle aktuálně zobrazované části modelu. Zkratka Alt+C. Ukázka je na obrázku 60.

X, Y a Z

Souřadnice středu otáčení modelu. Zkratka F10. Ukázka je na obrázku 60.

4.2.8 Barevné tabulky / Prahování

Tento panel je určený pro nastavování aktuální palety (barevné tabulky) a Rozlišení řezů modelu (rozlišení řezů nelze nastavovat v případě renderovací metody Dividing cubes (Kap. 4.8.3)). Palety jsou společné pro všechny prohlížeče (moduly). Bližší informace o paletách naleznete u editoru palet (Viz. CD). Dále zde uživatel může nastavit Prahování. Ukázka panelu je na obrázku 61.



Obr. 61: Oblast barevné tabulky / prahování

Tlačítko

Zobrazí okno určené k práci se společnými paletami (Viz. CD). Zkratka Ctrl+P. Ukázka panelu je na obrázku 61.

Aktuální číslo palety

Číslo aktuální vybrané barevné tabulky. Zkratka F2. Ukázka panelu je na obrázku 61.

Inverzní a Reverzní

Pomocí těchto checkboxů můžete invertovat jednotlivé barvy aktuální palety tj. provést operaci XOR na všechny RGB barvy aktuální palety. Reverze palety znamená překlopení (přehození pořadí) jejich barev. Zkratka Alt+I a Alt+R. Ukázka panelu je na obrázku 61.

Prahování

Při zobrazování hodnot intenzit dat studie se používají dolní a horní práh intenzit. Všechny intenzity menší nebo rovny dolnímu prahu se zobrazí první barvou v aktuální barevné tabulce, podobně všechny intenzity větší nebo rovné hornímu prahu se zobrazí poslední barvou aktuální barevné tabulky. Intenzity mezi těmito 2 prahy se lineárně nainterpolují na barvy aktuální barevné tabulky. Ukázka panelu je na obrázku 62.



Obr. 62: Prahování

Posuvníky prahování

Posuvníky umožňují měnit dolní a horní práh intenzit, které se mapují na jednotlivé barvy aktuální barevné tabulky. Zkratky F3 a F4. Ukázka panelu je na obrázku 62.

Upravit

Stiskem tlačítka **Upravit** se přizpůsobí rozsah obou posuvníků podle jejich aktuálních pozic. Tím umožní jemnější barevné rozlišení při posunu posuvníků. Zkratka Alt+U. Ukázka panelu je na obrázku 62.

Původní

Stiskem tlačítka **Původní** se nastaví rozsah obou posuvníků podle skutečných hodnot intenzit ve zdrojových datech tj. nastaví minimální hodnotu posuvníku pro dolní práh podle minimální intenzity v datech studie, maximální hodnotu posuvníku pro horní práh podle maximální intenzity. Zkratka Alt+V. Ukázka panelu je na obrázku 62.

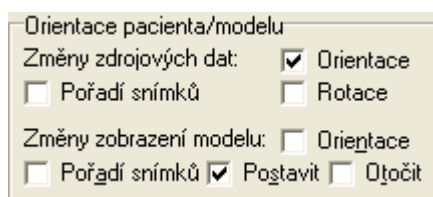
Rozlišení řezů

Výběr rozlišení pomocných textur použitých pro zobrazování řezů modelu. V případě použití renderovací metody Dividing cubes (Kap. 4.8.3) není výběr rozlišení možný, protože řezy se zobrazují jinou metodou (pomocí bodů). Na různých počítačích se může nabídka rozlišení lišit podle možností grafického systému. Čím vyšší rozlišení, tím lepší výsledek, ale také vyšší nároky na paměť a rychlost grafického systému počítače. Zkratka Ctrl+F2. Ukázka panelu je na obrázku 61.

Implicitně se rozlišení nastaví automaticky vhodně podle rozlišení 3D dat studie. Pro palety bez ostrých přechodů je toto rozlišení většinou dostačující.

4.2.9 Orientace pacienta / modelu

Tato oblast umožňuje měnit polohu pacienta zadávanou spolu se studií před a po vypočítání 3D modelu.



Obr. 63: Oblast Orientace pacienta / modelu

Změny zdrojových dat

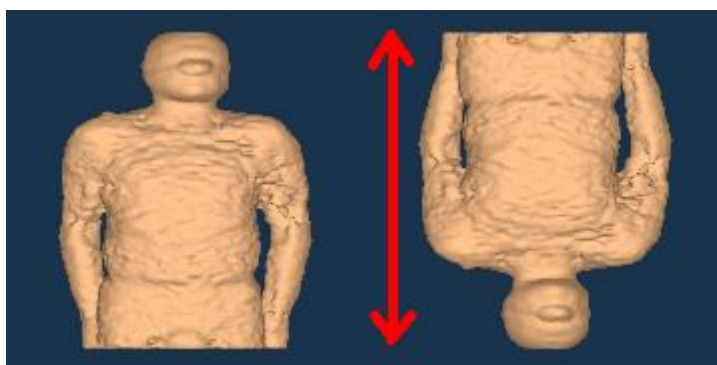
Následující políčka mění zdrojová data. Po jejich změně dochází tedy ke kompletnímu přepočítání dosud vytvořeného modelu, což může trvat poměrně dlouhou dobu. Tyto změny také nelze ukládat na časovou osu, nelze je tedy animovat. Ukázka panelu je na obrázku 63.

- **Pořadí snímků.** Umožňuje použít opačné řazení snímků, než které je použité ve zdrojových datech. Zkratka Ctrl+A. Ukázka panelu je na obrázku 63.
- **Orientace.** Umožňuje použít opačnou hodnotu klíče **patient orientation**, než která je ve zdrojových datech. Zkratka Ctrl+N. Ukázka panelu je na obrázku 63.
- **Rotace.** Umožňuje použít opačnou hodnotu klíče **patient rotation**, než která je ve zdrojových datech. Zkratka Ctrl+R. Ukázka panelu je na obrázku 63.

Změny zobrazení modelu

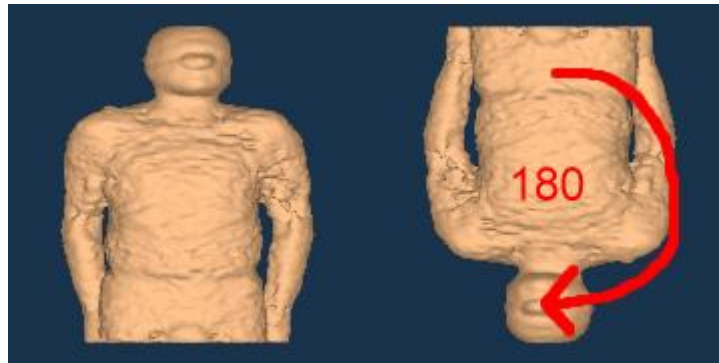
Následující políčka nemění zdrojová data, pouze mají vliv na výsledné zobrazení modelu. Po jejich změně nedochází tedy k přepočítání dosud vytvořeného modelu. Tyto změny lze ukládat na časovou osu, lze je tedy animovat. Ukázka panelu je na obrázku 63.

- **Pořadí snímků.** Umožňuje použít opačné řazení snímků, než které je použité ve zdrojových datech. Zkratka Alt+A. Ukázka panelu je na obrázku 63.



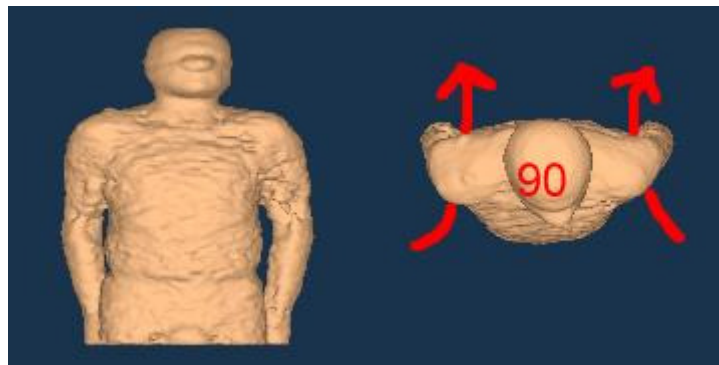
Obr. 64: Ukázka změny pořadí snímků

- **Orientace.** Umožňuje použít opačnou hodnotu klíče **patient orientation**, než která je ve zdrojových datech. Zkratka Alt+N. Ukázka panelu je na obrázku 63.



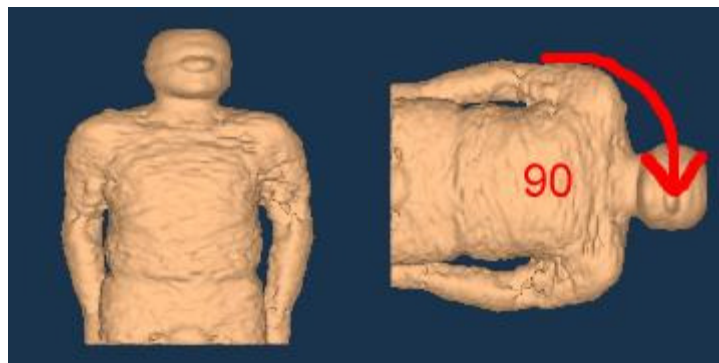
Obr. 65: Ukázka změny orientace modelu

- **Postavit.** Změní orientaci pacienta o 90 stupňů tak, jako kdyby si pacient z lehu stoupl. Zkratka Alt+S. Ukázka panelu je na obrázku 63.



Obr. 66: Ukázka změny orientace modelu

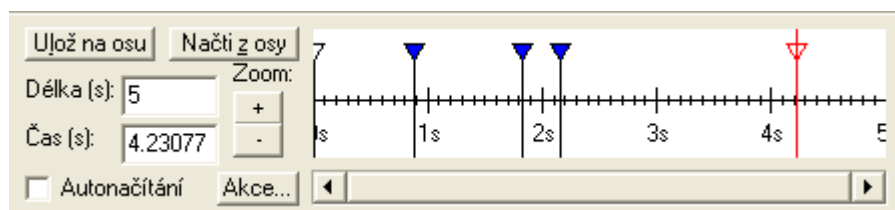
- **Otočit.** Změní orientaci pacienta o 90 stupňů tak, jako kdyby se pacient otočil napříč k lehátku. Zkratka Alt+T. Ukázka panelu je na obrázku 63.



Obr. 67: Ukázka změny orientace modelu

4.3 Panel ovládacích prvků pro tvorbu animace

Nachází se v dolní části okna modulu TomoAVI. Lze schovat nebo zobrazit pomocí tlačítka **On/Off animace panel**. Zkratka F12. Slouží k tvorbě animace a nastavování parametrů modelu v čase.



Obr. 68: Panel ovládacích prvků pro tvorbu animace

Ulož na osu

Vloží aktuální stav modelu do časové osy na aktuální čas. Tomuto bodu v čase se pak říká **klíčový bod**. Pokud již v daném čase nějaký klíčový bod existoval, je přepsán. Klíčové body jsou zobrazovány na časové ose. Zkratka Alt+L. Ukázka panelu je na obrázku 68.

Načti z osy

Načte z časové osy z aktuálního času stav modelu. Pokud byl v aktuálním čase nějaký **klíčový bod**, načtou se údaje přímo z něj. V případě, že tam nebyl, **interpolují** se hodnoty lineární interpolací z přilehlých klíčových bodů. Zkratka Alt+Z. Ukázka panelu je na obrázku 68.

Délka

Udává celkovou délku animace v sekundách. Pokud délku animace zmenšujete, mohou se některé klíčové body, které přesahují její délku, vymazat. Zkratka Ctrl+F3. Ukázka panelu je na obrázku 68.

Čas

Udává aktuální čas na časové ose v sekundách. Zkratka Ctrl+F8. Aktuální čas je zobrazován na časové ose červeně. V případě, že po zadání nového času stisknete **Shift+Enter**, dojde k označení klíčových bodů ležících mezi původním a novým aktuálním časem. Ukázka panelu je na obrázku 68.

Zoom

Pomocí tlačítek + a - umožňuje zvětšovat nebo zmenšovat výřez aktuálně viditelný na časové ose. Zkratky Ctrl+F12 a Ctrl+F11. Ukázka panelu je na obrázku 68.

Autonačítání

Pokud je políčko zaškrtnuté, dochází k automatickému načítání modelu z časové osy, jakmile je změněn aktuální čas. Pokud je měněn aktuální čas pomocí myši na časové ose, jsou z osy načítány pouze parametry nevyžadující složité výpočty. Komplettní model je z osy načten až po uvolnění tlačítka myši. Zkratka Alt+Q. Ukázka panelu je na obrázku 68.

Časová osa

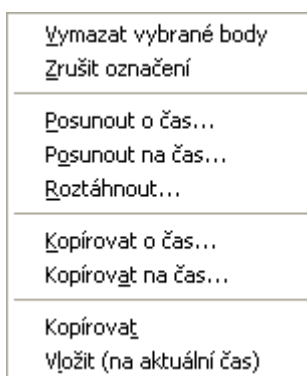
Výřez s časovou osou zabírá většinu plochy panelu pro tvorbu animace. Zobrazuje časovou osu, klíčové body a aktuální čas. **Posuvníkem** lze pohybovat s výřezem. Zkratka Ctrl+F9. Pomocí myši lze měnit klíčové body na ose. Ukázka panelu je na obrázku 68.

- **Pravé tlačítko myši** nastaví aktuální čas. Pokud se v bezprostřední blízkosti nacházel klíčový bod, je aktuální čas nastaven přesně na čas klíčového bodu.

- **Levé tlačítko myši** označí klíčové body pro další editaci. Pokud kliknete mimo klíčové body, výběr se zruší, pokud kliknete na klíčový bod, vybere se. Pokud se stisknutým levým tlačítkem táhnete myší, vybere se více klíčových bodů. Se stisknutou klávesou **Shift** lze rozšířit označení o další klíčové body.
- **Levé tlačítko myši** posunuje nebo kopíruje vybrané klíčové body. V případě, že stisknete levé tlačítko myši na jednom z vybraných klíčových bodů, tažením ho můžete přemístit. Pokud u toho držíte stisknutou klávesu **Shift**, tak se po uvolnění myši vytvoří kopie klíčových bodů.

4.3.1 Tlačítko Akce...

Zobrazí menu s výběrem operací pro vybrané klíčové body. Zkratka Alt+X. Toto tlačítko naleznete na obrázku 68.



Obr. 69: Menu s operacemi pro vybrané klíčové body

Vymazat vybrané body

Odstraní vybrané klíčové body z časové osy. Ukázka menu je na obrázku 69.

Zrušit označení

Zruší označení klíčových bodů. Ukázka menu je na obrázku 69.

Posunout o čas

Umožňuje posunout vybrané klíčové body na časové ose o přesně stanovený čas. Ukázka menu je na obrázku 69.

Posunout na čas

Umožňuje posunout vybrané klíčové body na časové ose na přesně stanovený čas. Ukázka menu je na obrázku 69.

Roztáhnout

Změní rozestupy (měřítko) klíčových bodů podle stanoveného zvětšení/zmenšení. Za střed zvětšování se považuje klíčový bod s nejmenším časem. Ukázka menu je na obrázku 69.

Kopírovat o čas

Umožňuje zkopírovat vybrané klíčové body na časové ose o přesně stanovený čas. Ukázka menu je na obrázku 69.

Kopírovat na čas

Umožňuje zkopírovat vybrané klíčové body na časové ose na přesně stanovený čas. Ukázka menu je na obrázku 69.

Kopírovat

Zkopíruje parametry modelu ve vybraných časových bodech do schránky. Následně je můžete vložit na časovou osu na jiný čas, nebo do jiné studie. Data jsou ve schránce uložena v textovém formátu (Kap. 4.9), lze je tedy zpracovávat v jiném programu a následně opět použít v modulu TomoAVI. Ukázka menu je na obrázku 69.

Vložit (na aktuální čas)

Vloží klíčové body ze schránky na aktuální čas na časové ose. Klíčové body lze ručně upravovat v libovolném textovém editoru, protože mají textový tvar (Kap. 4.9). Ukázka menu je na obrázku 69.

4.4 Nastavení barev a světla

Podle zadaných parametrů je z tomografických dat vypočítán matematický 3D model. Aby bylo možné tento model zobrazit, je nutné specifikovat vlastnosti osvětlení scény, podle kterých se vypočítá výsledný obrázek.



Obr. 70: Okno pro nastavení barev a světla

Podklad

Slouží k nastavení barvy podkladu renderovací plochy. V políčku napravo vidíte aktuální barvu. Pomocí tlačítka **Nastav** ji lze změnit. Ukázka okna je na obrázku 70.

Povrch

Slouží k nastavení barvy povrchu modelu. V políčku napravo vidíte aktuální barvu. Pomocí tlačítka **Nastav** ji lze změnit. Ukázka okna je na obrázku 70.

Automaticky podle palety

Při zaškrtnutí tohoto políčka se ignoruje uživatelem zadaná barva povrchu modelu a místo toho se použije automaticky barva podle intenzity povrchu a aktuální palety (Kap. 4.2.8). Ukázka okna je na obrázku 70.

Okolní světlo

Udává míru okolního osvětlení od 0 (0%) do 255 (100%). Ukázka okna je na obrázku 70.

Odražené světlo

Udává míru osvětlení modelu směrem od pozorovatele od 0 (0%) do 255 (100%). Ukázka okna je na obrázku 70.

4.5 Rezy modelem

Co jsou to řezy

Model bez řezů ohraničuje podprostor, kde je intenzita vyšší nebo rovna intenzitě zadané v Intenzita povrchu (Kap. 4.2.3). Barva povrchu je všude stejná, protože i intenzita je na celém povrchu stejná. Abychom měli přehled i o oblastech s vyšší intenzitou, než je prahová, je možné specifikovat řezy. Na řezech se zobrazí intenzity v daném místě obarvené danou paletou (Kap. 4.2.8).

Jednu rovinu řezu bez ohledu na její vztah k jiným rovinám řezu nazýváme **řezovou rovinou**.

Řezy modelem lze zadávat dvěma způsoby.

- Jednoduchý způsob je určen pro rychlé zadávání maximálně tří na sebe kolmých řezů. Bližší informace naleznete v kapitole 4.2.4.



Obr. 71: Jednoduché řezy

- Pokročilé zadávání umožňuje zkušenému uživateli nadefinovat prakticky libovolné řezy pomocí Řezového výrazu.



Obr. 72: Pokročilé řezy

Pomocí políček **Jednoduché** a **Pokročilé** lze přepínat mezi způsoby zadávání. Zkratky Alt+J a Alt+K.

Ořezání modelu probíhá tak, že je rozdělen rovinami v prostoru (Kap. 4.1) na několik částí. Některé části modelu jsou odfiltrovány a některé zůstanou. Na rozhraní řezů se vykreslí intenzity v daném místě modelu. Před vykreslením je na intenzity namapována paleta (Kap. 4.2.8).

Jak probíhá filtrování

O odfiltrování dané části prostoru vymezeného řezovými rovinami se rozhoduje různě podle způsobu zadávání řezů.

- U Jednoduchého způsobu naleznete informace v kapitole 4.2.4.
- U Pokročilého zadávání se rozhoduje podle Řezového výrazu.

Řezový výraz

Řezový výraz se používá u pokročilého zadávání řezů. Základem jsou následující operátory.

- **R(a,b,c,d)** - udává rovinu v prostoru (Kap. 4.1) pomocí rovnice $ax+by+cz+d=0$. Poloprostor, kde je $ax+by+cz+d \geq 0$, se neodfiltruje. Část modelu neodfiltrovaná rovinou $ax+by+cz+d=0$ se nazývá **polomodel definovaný rovinou $ax+by+cz+d=0$** . Podobně se definuje část modelu neodfiltrovaná výrazem alfa jako **polomodel definovaný výrazem alfa**.
- + - výsledkem je spojení polomodelů z levé a pravé strany. Příklad: $R(1,0,0,0)+R(0,1,0,0)$.
- * - výsledkem je průnik polomodelů z levé a pravé strany. Příklad: $R(1,0,0,0)*R(0,1,0,0)$.
- (a) - pomocné symboly (klasická funkce závorek)

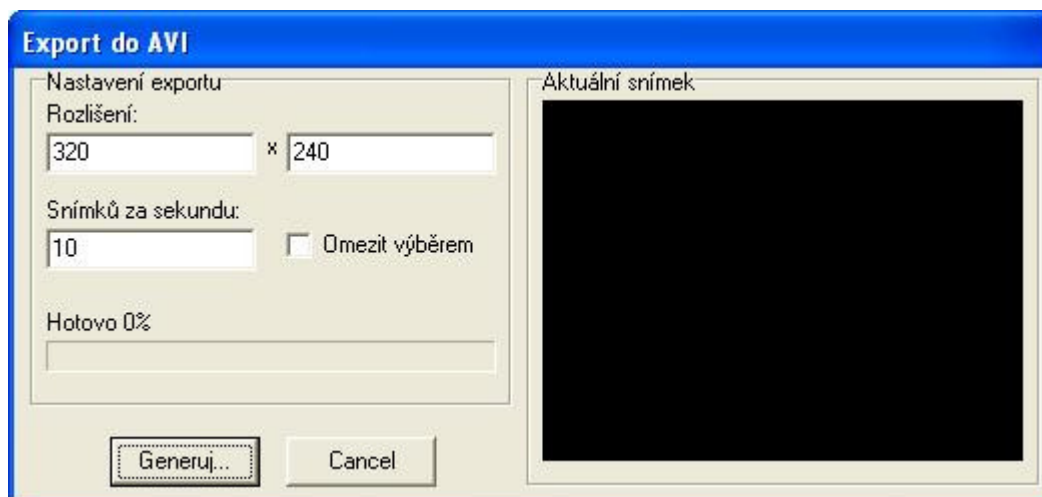
Příklady řezových výrazů

$R(1,0,0,-x)+R(0,1,0,-y)+R(0,0,1,-z)$ - je ekvivalentní jednoduchému zadávání řezů s volbou **vyseknout klín**, kde x, y a z jsou hodnoty z příslušných políček x, y a z.

$R(1,0,0,-x)*R(0,1,0,-y)*R(0,0,1,-z)$ - je ekvivalentní jednoduchému zadávání řezů s volbou **zobrazit klín**, kde x, y a z jsou hodnoty z příslušných políček x, y a z.

4.6 Export do AVI

Aby mohl proběhnout export animace (Kap. 4.3) do AVI souboru, musí být na časové ose (Kap. 4.3) alespoň jeden klíčový bod. V okně **Export do AVI** nastavíte základní parametry animace. Potom stisknete tlačítko **Generuj...** a vyberete umístění výsledné animace. Následně jsou vám nabídnuty dostupné **kodeky** pro komprimaci animace. Pokud zvolíte **bez komprese**, může být výsledná animace dosti velká (až stovky MB). V jakémkoli okamžiku můžete generování animace přerušit.



Obr. 73: Okno pro export do AVI

Rozlišení

Umožňuje zadat rozměr snímku animace v obrazových bodech (pixelech). Ukázka okna je na obrázku 73.

Snímků za sekundu

Umožňuje zadat počet snímků, které budou v animaci zobrazeny za každou jednu sekundu. Ukázka okna je na obrázku 73.

Omezit výběrem

Omezí rozsah generování animace na oblast časové osy určenou vybranými klíčovými body. Ukázka okna je na obrázku 73.

Hotovo

Zobrazuje průběh generování animace. Ukázka okna je na obrázku 73.

Generuj...

Vyberete umístění výsledné animace. Následně jsou vám nabídnuty dostupné **kodeky** pro komprimaci animace. Pokud zvolíte **bez komprese**, může být výsledná animace dosti velká (až stovky MB). V jakémkoli okamžiku můžete generování animace přerušit tlačítkem **Cancel**. Ukázka okna je na obrázku 73.

Cancel

Přeruší generování animace nebo zavře okno **Export do AVI**. Dosud vygenerovaná část animace zůstane uložena v cílovém souboru. Ukázka okna je na obrázku 73.

4.7 Export do VRML

Export do VRML se od jiných dostupných exportů (Kap. 4.2.6) z modulu TomoAVI liší v tom, že není vyexportován pouze obraz 3D modelu, ale je vyexportován celý aktuálně zobrazený 3D model v trojrozměrné podobě. VRML (Virtual Reality Modeling Language) je jazyk navržený pro popis jednotlivých 3D objektů i celých 3D scén a používá se často na webových stránkách. K prohlížení vyexportovaných VRML souborů je nutné mít nainstalován **VRML klient** (například volně šiřitelný VRML klient Cortona společnosti ParallelGraphics). Export do VRML není dostupný při nastavené renderovací metodě Dividing cubes (Kap. 4.8.3) z důvodu vykreslování modelu pomocí bodů.

4.7.1 Nastavení a průběh exportu

Po zvolení exportu do VRML (Kap. 4.2.6) nastavíte v dialogovém okně parametry exportu, poté zadáte cílový adresář a název souboru pro uložení výsledku. Export může trvat delší dobu, záleží na složitosti modelu.



Obr. 74: Okno pro nastavení parametrů exportu do VRML

Snížit rozlišení textur

Umožňuje specifikovat nižší rozlišení textur, na které se aktuální textury řezů přepočítají, než budou vyexportovány. Některé VRML prohlížeče mohou mít s příliš velkými texturami problémy. Ukázka okna je na obrázku 74.

JPEG kvalita textur

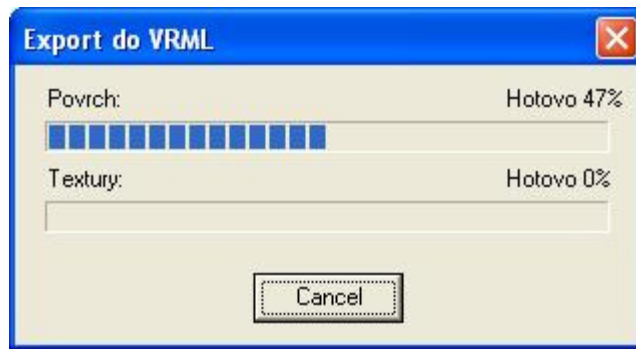
Slouží pro zadání míry komprese souborů JPEG, do kterých jsou při exportu ukládány textury řezů modelu. Čím menší kvalita, tím menší soubor. Ukázka okna je na obrázku 74.

Exportuj údaje o pacientovi

Při zaškrtnutí tohoto políčka budou osobní údaje pacienta zapsány do VRML souboru a kdokoli si je bude moci přečíst! Zvláště pokud hodláte soubory zpřístupnit na internetu, nedoporučuje se tuto volbu zaškrtnout. Ukázka okna je na obrázku 74.

Průběh exportu

Průběh exportu do VRML je zobrazován v samostatném okně.



Obr. 75: Okno zobrazující průběh exportu do VRML

Povrch

Zobrazuje průběh ukládání 3D povrchu modelu i povrchu jeho řezů do výsledného souboru. Ukázka okna je na obrázku 75.

Textury

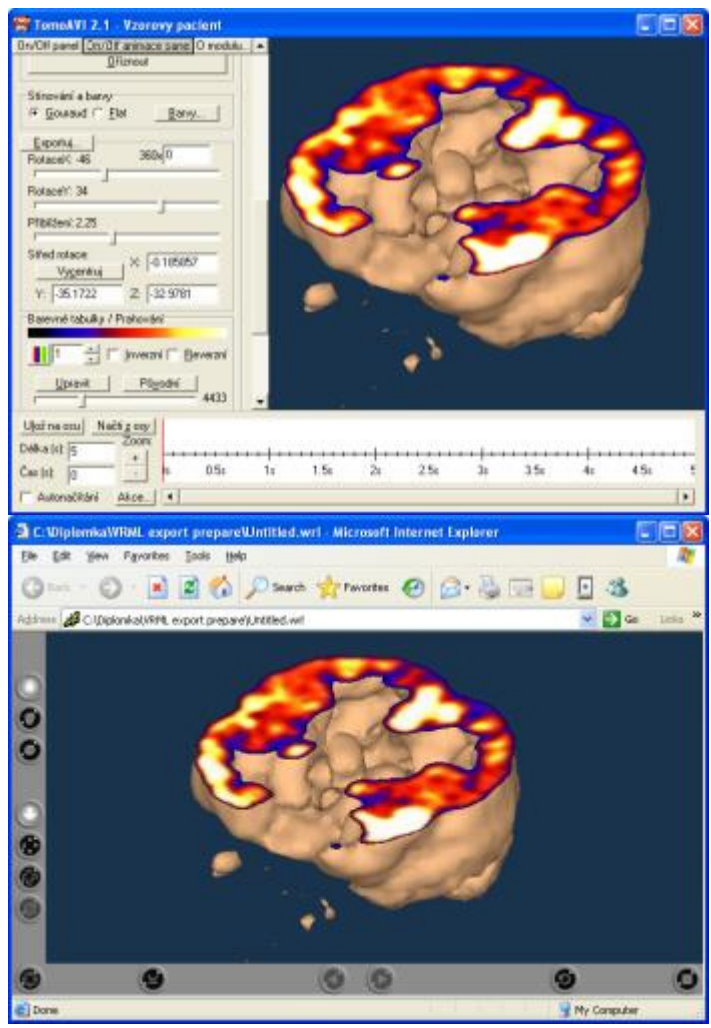
Zobrazuje průběh ukládání textur řezů modelu do JPEG souborů. Ukázka okna je na obrázku 75.

Cancel

Přeruší generování VRML modelu. Dosud vygenerovaná část modelu zůstane uložena v cílovém souboru. Výsledek zřejmě nepůjde korektně zobrazit. Ukázka okna je na obrázku 75.

4.7.2 Prohlížení výsledku

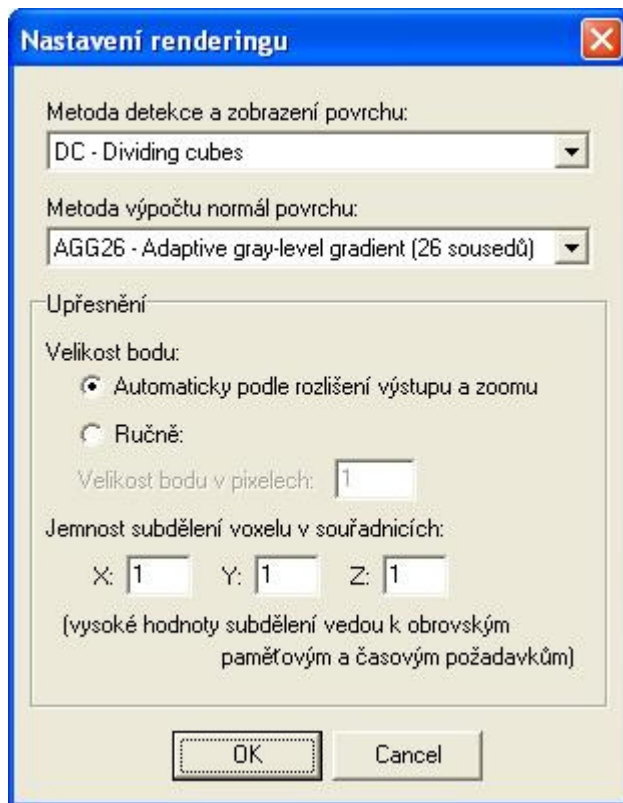
Na následujícím obrázku vidíte původní 3D model zobrazený přímo v modulu TomoAVI a pro srovnání tentýž 3D model po exportu do VRML a zobrazený uvnitř webového prohlížeče. S VRML modelem je možné uvnitř webového prohlížeče otáčet pomocí myši a prohlížet si ho ze všech stran. Veškeré parametry modelu včetně barvy pozadí odpovídají originálu.



Obr. 76: Srovnání VRML modelu s originálem

4.8 Renderovací metody

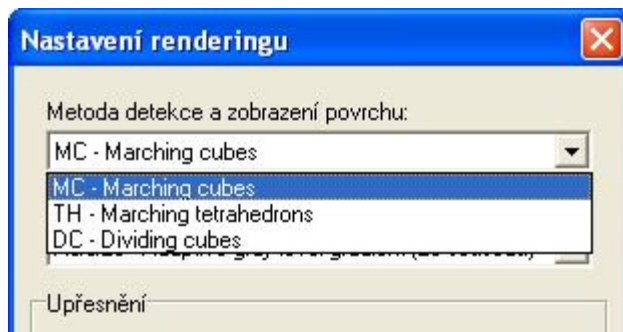
3D Model (bez řezů) ohraničuje podprostor, kde je intenzita vyšší nebo rovna intenzitě zadané v Intenzita povrchu (Kap. 4.2.3). Aplikační modul TomoAVI podporuje několik metod pro detekci a zobrazení povrchu 3D modelu. Barva povrchu modelu je všude stejná, protože i intenzita je na celém povrchu stejná. K zobrazení modelu je třeba znát normály povrchu. Existuje celá řada metod, jak tyto normály odhadnout. Aplikační modul TomoAVI podporuje více metod pro výpočet normál povrchu 3D modelu. Metodu pro výpočet a zobrazování modelu a pro výpočet normál povrchu je možné zvolit v okně viz. obr. dole po stisku tlačítka **Nastav...** (Kap. 4.2.2).



Obr. 77: Okno s nastavením Renderovací metody

Metody detekce a zobrazení povrchu

Aplikační modul TomoAVI podporuje několik metod pro detekci a zobrazení povrchu 3D modelu.

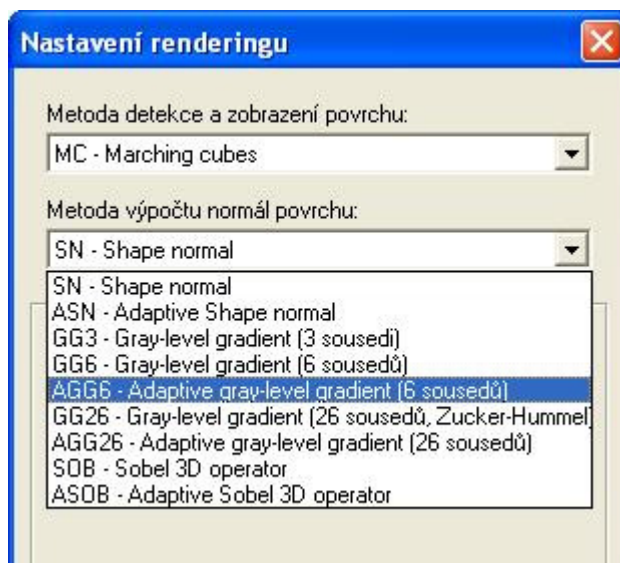


Obr. 78: Metody detekce a zobrazení povrchu

- Marching cubes (Kap. 4.8.1)
- Marching tetrahedrons (Kap. 4.8.2)
- Dividing cubes (Kap. 4.8.3)

Metody výpočtu normál povrchu

Aplikační modul TomoAVI podporuje více metod pro výpočet normál povrchu 3D modelu.

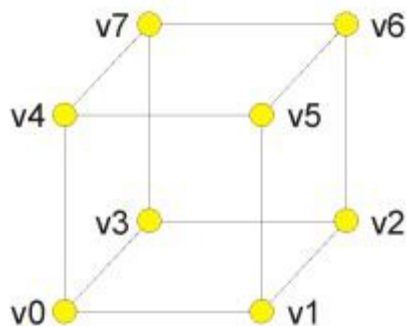


Obr. 79: Metody výpočtu normál povrchu

- Shape normal (Kap. 4.8.4)
- Adaptive Shape normal (Kap. 4.8.4)
- Gray-level gradient (3 sousedí) (Kap. 4.8.4)
- Gray-level gradient (6 sousedů) (Kap. 4.8.4)
- Adaptive gray-level gradient (6 sousedů) (Kap. 4.8.4)
- Gray-level gradient (26 sousedů, Zucker-Hummel) (Kap. 4.8.4)
- Adaptive gray-level gradient (26 sousedů) (Kap. 4.8.4)
- Sobel 3D operator (Kap. 4.8.4)
- Adaptive Sobel 3D operator (Kap. 4.8.4)

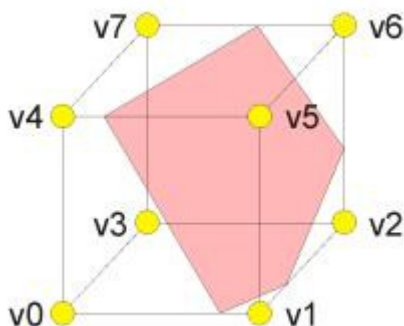
4.8.1 Marching cubes

Metoda Marching cubes rekonstruuje povrch 3D modelu pomocí polygonů. Je to známá a používaná metoda a není zde cílem ji detailně popisovat. Povrch prochází místy, kde je intenzita záření rovna nastavené prahové intenzitě. Protože je na této ploše intenzita záření všude stejná, říká se tomuto povrchu isoplocha. Algoritmus Marching cubes rekonstruuje povrch vždy v oblasti mezi osmi sousedními voxely a to nezávisle na sousedních osmicích. Na následujícím obrázku vidíte středy osmi sousedních voxelů z matice intenzit.



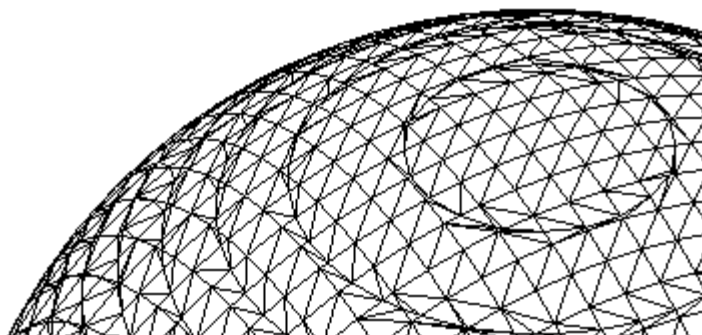
Obr. 80: Marching cube

Voxely se rozdělí na ty, co mají hodnotu intenzity menší než prahovou hodnotu, a na ty, co mají hodnotu intenzity větší nebo rovno prahové hodnotě. Isoplocha pak prochází těmi hranami, kde každý z koncových vrcholů leží v opačné množině. Na následujícím obrázku je vidět isoplocha proložená prostorem mezi voxely.



Obr. 81: Marching cube isoplocha

Na předchozím příkladě mají voxely v_0, v_2, v_3, v_4, v_7 hodnotu intenzity větší než prahová hodnota a leží pod isoplochou, zatímco voxely v_1, v_5, v_6 mají hodnotu intenzity menší než prahovou a leží tudíž nad isoplochou. Celkem existuje 256 možných kombinací zařazení voxelů pod a nad isoplochu. Nejčastější případ je ten, kdy všech osm voxelů leží pod isoplochou, nebo naopak nad isoplochou. Tyto případy jsou nejjednodušší, protože pak není potřeba v této oblasti žádný povrch generovat.

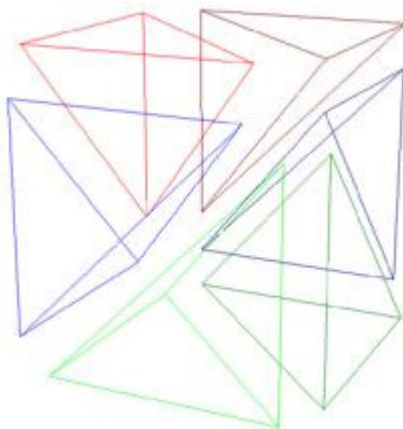


Obr. 82: Ukázka povrchu koule rekonstruované metodou Marching cubes

4.8.2 Marching tetrahedrons

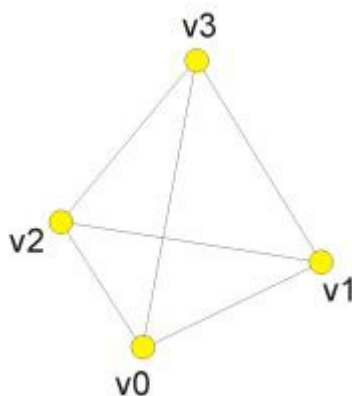
Metoda Marching tetrahedrons rekonstruuje povrch 3D modelu pomocí polygonů podobně jako metoda Marching cubes (Kap. 4.8.1). Je to méně známá metoda vhodná především pro vstupní data s nepravidelnou topologií, kde není možné použít například Marching cubes.

Není zde cílem tuto metodu detailně popisovat. Narozdíl od algoritmus Marching cubes, který rekonstruuje povrch vždy v oblasti mezi osmi sousedními voxely, metoda Marching tetrahedrons rekonstruuje povrch vždy mezi čtveřicí bodů v prostoru (uvnitř tetraedru - čtyřstěnu). V případě naší pravoúhlé matice vstupních intenzit je potřeba vhodně zvolit rozdělení na čtyřstěny. Na následujícím obrázku je vidět použité rozdělení.



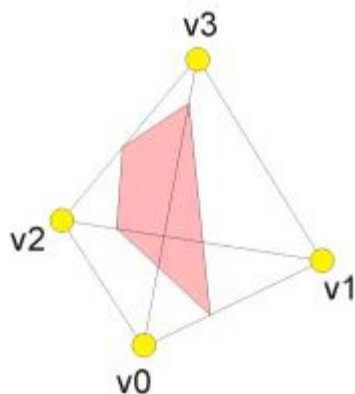
Obr. 83: Rozdělení oblasti mezi osmi voxely na čtyřstěny

Každý čtyřstěn je určen čtyřmi voxely z původní osmice. Isoplocha se řeší v každém takovém čtyřstěnu zvlášť.



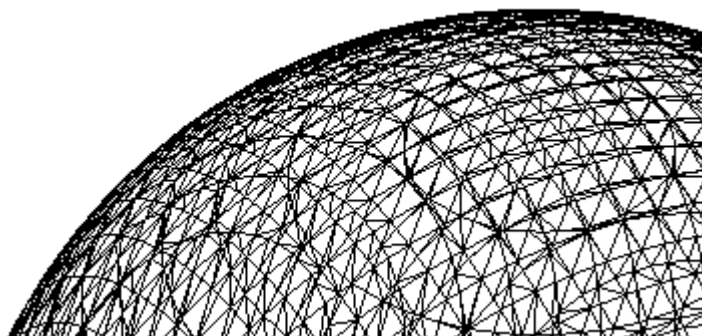
Obr. 84: Marching tetrahedron

Stejně jako u metody Marching cubes se voxely rozdělí na ty, co mají hodnotu intenzity menší než prahovou hodnotu, a na ty, co mají hodnotu intenzity větší nebo rovno prahové hodnotě. Isoplocha pak prochází těmi hranami, kde každý z koncových vrcholů leží v opačné množině. Na následujícím obrázku je vidět isoplocha proložená prostorem mezi voxely.



Obr. 85: Marching tetrahedron isoplocha

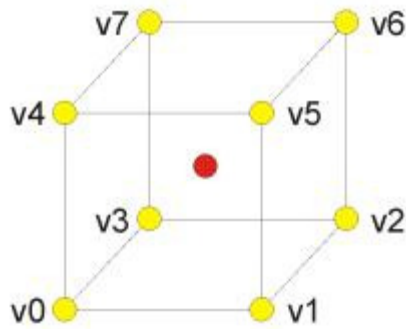
Na předchozím příkladě mají voxely v1 a v3 hodnotu intenzity větší než prahová hodnota a leží pod isoplochou, zatímco voxely v0 a v2 mají hodnotu intenzity menší než prahovou a leží tudíž nad isoplochou. Oproti metodě Marching cubes je zde celkově pouze 16 možných kombinací zařazení voxelů pod a nad isoplochou. Nejčastější případ je ten, kdy všechny čtyři voxely leží pod isoplochou, nebo naopak nad isoplochou. Tyto případy jsou nejjednodušší, protože pak není potřeba v této oblasti žádný povrch generovat.



Obr. 86: Ukázka povrchu koule rekonstruované metodou Marching tetrahedrons

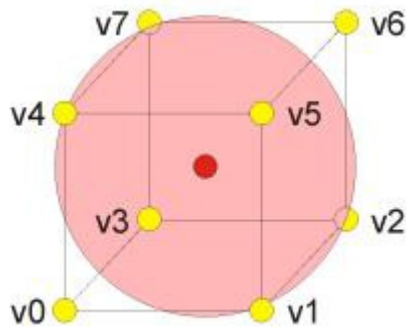
4.8.3 Dividing cubes

Metoda Dividing cubes se od předchozích dvou metod značně liší, neboť nerekonstruuje povrch modelu pomocí polygonů, ale skládá povrch z jednotlivých bodů. Tato metoda podobně jako metoda Marching cubes (Kap. 4.8.1) pracuje vždy s osmicí sousedních voxelů. Voxely se také rozdělí na ty, co mají hodnotu intenzity menší než prahovou hodnotu, a na ty, co mají hodnotu intenzity větší nebo rovno prahové hodnotě. V případě, že všech osm voxelů patří do první skupiny nebo naopak všech osm patří do druhé skupiny, žádný bod povrchu se negeneruje. V ostatních případech je jisté, že mezi touto osmicí prochází hledaná isoplocha. Vygeneruje se tedy jeden bod povrchu viz. následující obrázek.



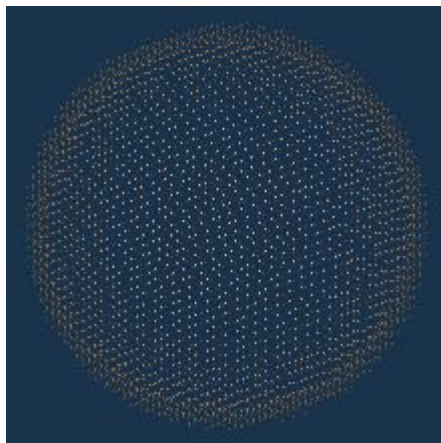
Obr. 87: Dividing cube (dělení 1,1,1)

Abychom mohli povrchové body vyrenderovat, musíme jim přidělit vhodnou velikost.



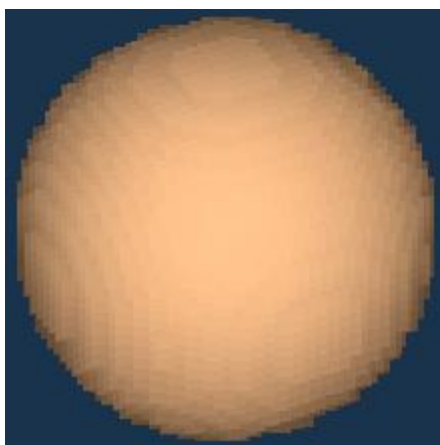
Obr. 88: Dividing cube (dělení 1,1,1)

Pokud zvolíme velikost bodů jeden pixel a pokud jsou při renderování výsledného obrázku sousední povrchové body od sebe vzdáleny více jak jeden pixel, povrch se nejeví celistvě viz. následující obrázek.



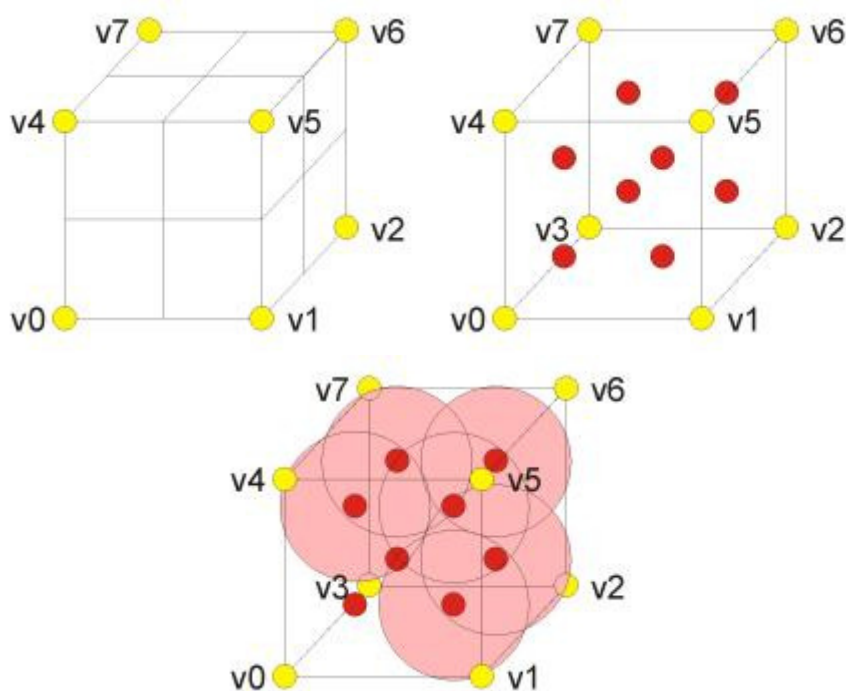
Obr. 89: Ukázka modelu koule (dělení 1,1,1) - velikost bodu 1 pixel

Abychom mohli povrch zobrazit souvisle, musíme bodům přidělit vhodný poloměr. V dalším příkladě je velikost bodu 10 pixelů.



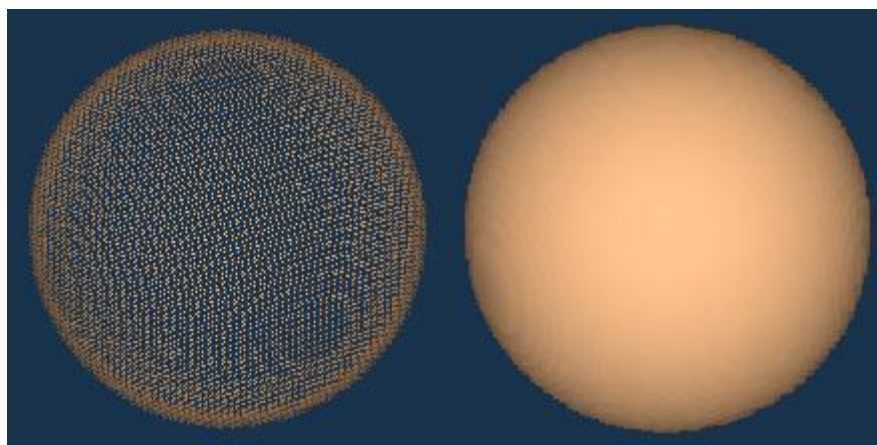
Obr. 90: Ukázka modelu koule (dělení 1,1,1) - velikost bodu 10 pixelů

Na předchozím příkladě je vidět, že výsledek je značně hrubý. Abychom zjemnili zobrazení, můžeme místo jednoho povrchového bodu zobrazovat bodů více. Kostku určenou osmicí vertexů můžeme rozdělit na více menších. Hodnotu intenzit nově vzniklých vrcholů menších krychlí vypočítáme trilineární interpolací. V každé z menších krychlíček řešíme situaci stejně jako v předchozím případě. Příklad dělení dvěma v každém směru vidíme v následující ukázce.



Obr. 91: Dividing cube (dělení 2,2,2)

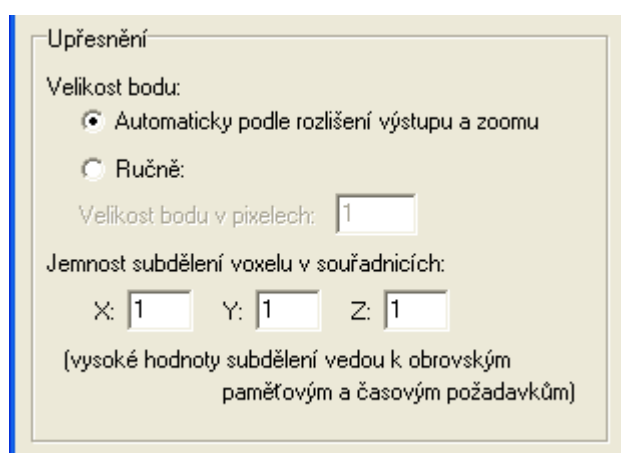
A výsledek při dělení dvěma v každém směru vidíte na modelu koule dole. První obrázek je s velikostí bodu 1 pixel a druhý s velikostí bodu 5 pixelů.



Obr. 92: Ukázka modelu koule (dělení 2,2,2) - velikost bodu 1 a 5 pixelů

Upřesnění (ovládací panel)

V aplikačním modulu TomoAVI lze u metody Dividing cubes nastavit několik parametrů.



Obr. 93: Upřesnění nastavení metody Dividing cubes

Velikost bodu - Lze volit buď automatickou velikost, kdy aplikace sama nastaví velikost bodu tak, aby byl povrch zobrazen souvisle, nebo zadat vlastní velikost bodu v pixelech. (Upozornění: některé implementace grafického rozhraní OpenGL mají maximální velikost bodu omezenou, může to způsobit nesouvislé zobrazení povrchu při velké velikosti bodu. Navíc může být bod zobrazován jako čtvereček místo kruhu)

Jemnost subdělení voxelu v souřadnicích - X,Y,Z. Uživatel může zadat jemnější dělení v různých směrech. I poměrně malé hodnoty mohou vést k výrazným paměťovým a časovým nárokům algoritmu, hodnoty je třeba volit s rozmyslem.

4.8.4 Metody výpočtu normál povrchu

Pro zobrazení povrchu modelu s osvětlením je třeba znát normály povrchu v klíčových bodech. Pro metody Marching cubes (Kap. 4.8.1) a Marching tetrahedrons (Kap. 4.8.2) jsou klíčovými body vrcholy vygenerovaných povrchových trojúhelníků. U metody Dividing cubes (Kap. 4.8.3) je zase potřeba zjistit normálu povrchu u každého povrchového bodu. Aplikační modul TomoAVI podporuje více metod pro výpočet normál povrchu 3D modelu. Kromě metody Shape normal pracují všechny metody pro zjišťování normál povrchu následovně:

Máme trojrozměrnou matici (Kap. 4.1) hodnot intenzity záření zkoumaného prostoru. Uděláme si matici stejné velikosti s tím rozdílem, místo hodnot intenzit voxelů bude obsahovat odhad normály případné isoplochy procházející tímto bodem. Pokud pak budeme

chtít odhadnout normálu povrchu v jiném místě uvnitř matice, tak hodnotu získáme trilineární interpolací z nejbližších hodnot. Každou jednotlivou složku normálových vektorů v matici normál vypočítáme konvolucí matice intenzit se speciální maticí 3x3x3. Cílem je vypočítat odhad gradientu funkce intenzity $I(x,y,z)$ v daném bodě. Jedná se v podstatě o jednoduché hranové detektory založené na první derivaci. U každé metody kromě Shape normal je použita konvoluční matice pro odhad x-ové složky gradientu uvedena (v případě, že měřítko 3D matice dat není ve všech směrech stejné, tedy že voxel není krychle, je nutné jednotlivé složky gradientu následně vydělit měřítkem v daném směru). Y-ové a z-ové složky gradientu záření vypočítáme analogicky. Hledaný normálový vektor je normalizovaný opačný gradientní vektor. Opačný proto, že hledáme vektor směřující k nižší intenzitě, tedy z tělesa ven.

U každé metody je uvedena konvoluční matice 3x3x3. Je podél osy z rozdělena do tří matic 3x3, kde u každé matice 3x3 směřuje osa x doprava a osa y nahoru. Uvedená matice je pro odhad x-ové složky gradientu.

Gray-level gradient (3 sousedi)

Bližší informace výše.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$z-1 \qquad \qquad z \qquad \qquad z+1$

Obr. 94: Konvoluční matice GG3

Gray-level gradient (6 sousedů)

Bližší informace výše.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$z-1 \qquad \qquad z \qquad \qquad z+1$

Obr. 95: Konvoluční matice GG6

Adaptive gray-level gradient (6 sousedů)

Tato metoda je podobná předchozí metodě, pouze dává lepší výsledky na tenkých objektech. Podrobnější popis je nad rámec tohoto helpu.

Gray-level gradient (26 sousedů, Zucker-Hummel)

Bližší informace výše.

$$\begin{pmatrix} -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ \frac{2}{\sqrt{3}} & 0 & \frac{2}{\sqrt{3}} \\ -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \end{pmatrix}_{z-1}, \begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ -1 & 0 & 1 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix}_z, \begin{pmatrix} -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ \frac{2}{\sqrt{3}} & 0 & \frac{2}{\sqrt{3}} \\ -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \end{pmatrix}_{z+1}$$

Obr. 96: Konvoluční matice GG26

Adaptive gray-level gradient (26 sousedů)

Tato metoda je podobná předchozí metodě, pouze dává lepší výsledky na tenkých objektech. Podrobnější popis je nad rámec tohoto helpu.

Sobel 3D operator

Bližší informace výše.

$$\begin{pmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{pmatrix}_{z-1}, \begin{pmatrix} -3 & 0 & 3 \\ -6 & 0 & 6 \\ -3 & 0 & 3 \end{pmatrix}_z, \begin{pmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{pmatrix}_{z+1}$$

Obr. 97: Konvoluční matice SOB

Adaptive Sobel 3D operator

Tato metoda je podobná předchozí metodě, pouze dává lepší výsledky na tenkých objektech. Podrobnější popis je nad rámec tohoto helpu.

Shape normal

Metoda Shape normal pro výpočet normál povrchu lze použít pouze u metody Marching cubes (Kap. 4.8.1) nebo Marching tetrahedrons (Kap. 4.8.2). Tato metoda počítá normálové vektory přímo z geometrie trojúhelníků povrchu. Normály vrcholů trojúhelníků povrchu modelu vypočítá tato metoda jako průměr geometrických normál trojúhelníků sousedících s daným vrcholem.

Adaptive Shape normal

Metoda Adaptive Shape normal pro výpočet normál povrchu lze použít pouze u metody Marching cubes (Kap. 4.8.1) nebo Marching tetrahedrons (Kap. 4.8.2). Tato metoda počítá normálové vektory přímo z geometrie trojúhelníků povrchu. Normály vrcholů trojúhelníků povrchu modelu vypočítá tato metoda jako vážený průměr geometrických normál trojúhelníků sousedících s daným vrcholem.

4.9 Formát dat ve schránce

Klíčové body v sobě nesou nastavení všech parametrů modelu. Lze je kopírovat do schránky nebo vkládat ze schránky na časovou osu. Klíčové body nacházející se ve schránce mají textový tvar a lze je tedy upravovat v libovolném textovém editoru. Aby modul TomoAVI porozuměl upraveným klíčovým bodům, musíte dodržet striktně následující formát.

Příklad jednoho takového klíčového bodu vidíte zde:

```
KlicBegin:
Time:0
IsoLevel:5000
Flat:0
PodkladRed:0.1
PodkladGreen:0.2
PodkladBlue:0.3
PovrchRed:0.73
PovrchGreen:0.62
PovrchBlue:0.51
OkolniSvetlo:0.3
OdrazeneSvetlo:0.7
UhelX:-85
UhelY:0.5
Zoom:1
StredOtaceniX:0
StredOtaceniY:0
StredOtaceniZ:0
Paleta:0
SubPaleta:0
AutoBarva:0
ModelOrient:0
ModelSlice:0
ModelPostavit:0
ModelOtocit:0
ThresholdMin:0
ThresholdMax:65535
RozliseniTextury:-1
JednoducheRezy:0
SouradniceRezuX:0
SouradniceRezuY:
SouradniceRezuZ:
RezyOR:1
OrientX:0
OrientY:1
OrientZ:0
RezyZadani:(R(0,0,1,0)+R(1,0,0,0))*R(1,1,0,0)
KlicEnd:
```

Musí být dodrženo pořadí a žádný řádek nesmí být vynechán. Takto zadaných klíčových bodů může být ve schránce libovolný počet. Význam jednotlivých řádků se dozvíte zde:

- **KlicBegin:** - informuje o začátku klíčového bodu
- **Time:** - čas ve kterém se klíčový bod nachází
- **IsoLevel:** - hodnota intenzity pro detekci povrchu
- **Flat:** - 0/1 zda je stínování Flat či Gouraud
- **PodkladRed:** - červená složka barvy podkladu renderovací plochy v rozsahu 0-1
- **PodkladGreen:** - zelená složka barvy podkladu renderovací plochy v rozsahu 0-1
- **PodkladBlue:** - modrá složka barvy podkladu renderovací plochy v rozsahu 0-1

- **PovrchRed:** - červená složka barvy povrchu modelu v rozsahu 0-1
- **PovrchGreen:** - zelená složka barvy povrchu modelu v rozsahu 0-1
- **PovrchBlue:** - modrá složka barvy povrchu modelu v rozsahu 0-1
- **OkolniSvetlo:** - množství okolního světla pro stínování povrchu modelu v rozsahu 0-1
- **OdrazeneSvetlo:** - množství odraženého světla ze směru od pozorovatele pro stínování povrchu modelu v rozsahu 0-1
- **UhelX:** - úhel otočení okolo svislé osy ve stupních v rozsahu -360000 až 360000
- **UhelY:** - úhel otočení okolo vodorovné osy ve stupních v rozsahu -90 až 90
- **Zoom:** - zvětšení zobrazení modelu v rozsahu 0.3 až 5.0
- **StredOtaceniX:** - x-ová souřadnice středu otáčení modelu v milimetrech v rozsahu -10000 až 10000
- **StredOtaceniY:** - y-ová souřadnice středu otáčení modelu v milimetrech v rozsahu -10000 až 10000
- **StredOtaceniZ:** - z-ová souřadnice středu otáčení modelu v milimetrech v rozsahu -10000 až 10000
- **Paleta:** - číslo palety použité pro obarvení intenzit na řezech v rozsahu 0 až neomezeně. Pokud paleta s takovým číslem neexistuje, je při importu nastavena na 0
- **SubPaleta:** - upřesňuje použití dané palety zadané řádkem Paleta a může mít následující hodnoty:
 - **0** - normální paleta
 - **1** - reverzní paleta
 - **2** - inverzní paleta
 - **3** - reverzní inverzní paleta
- **AutoBarva:** - 0/1 zda se použije barva povrchu modelu automaticky podle aktuální palety a prahování
- **ModelOrient:** - 0/1 zaškrtačací políčko viz. Změny zobrazení modelu - Orientace
- **ModelSlice:** - 0/1 zaškrtačací políčko viz. Změny zobrazení modelu - Pořadí snímků
- **ModelPostavit:** - 0/1 zaškrtačací políčko viz. Změny zobrazení modelu - Postavit
- **ModelOtocit:** - 0/1 zaškrtačací políčko viz. Změny zobrazení modelu - Otočit
- **ThresholdMin:** - spodní hodnota pro prahování palety. Hodnota mimo rozsah se upraví podle modelu
- **ThresholdMax:** - horní hodnota pro prahování palety. Hodnota mimo rozsah se upraví podle modelu
- **RozliseniTextury:** - udává rozlišení textur na řezech. Může nabývat hodnot:
 - **-1** - použije se automatické rozlišení textur
 - **n** - kde n je v rozsahu 32 až neomezeně. n by mělo být mocnina dvojky. Hodnota se upraví podle možností grafického systému
- **JednoducheRezy:** - 1/0 podle použití jednoduchých nebo pokročilých řezů
- **SouradniceRezuX:** - poloha řezu kolmého k ose x. Prázdná hodnota znamená žádný řez
- **SouradniceRezuY:** - poloha řezu kolmého k ose y. Prázdná hodnota znamená žádný řez

- **SouradniceRezuZ:** - poloha řezu kolmého k ose z. Prázdná hodnota znamená žádný řez
- **RezyOR:** - 1/0 podle vyseknutí nebo zobrazení klínu při jednoduchém zadávání řezů
- **OrientX:** - 1/0 podle orientace řezu kolmého k ose x (při jednoduchém zadávání). 1...Plus a 0...Minus
- **OrientY:** - 1/0 podle orientace řezu kolmého k ose y (při jednoduchém zadávání). 1...Plus a 0...Minus
- **OrientZ:** - 1/0 podle orientace řezu kolmého k ose z (při jednoduchém zadávání). 1...Plus a 0...Minus
- **RezyZadani:** - řezový výraz. Použije se při pokročilém zadávání řezů a neměl by být delší než 4000 znaků
- **KlicEnd:** - informuje o konci klíčového bodu

4.10 Klávesové zkratky použitelné v modulu TomoAVI

Některé klávesové zkratky jsou dostupné pouze pokud je otevřen příslušný ovládací panel.

Zkratky společné pro všechny prohlížeče

| Akce | Klávesy |
|---|----------------|
| Zobrazit editor palet | Ctrl+P |
| Zobrazit editor textových popisků studie | Ctrl+T |
| Uložit aktuální pohled jako screenshot do databáze | Ctrl+S |
| Zobrazit okno se screenshoty ke studii | Ctrl+O |
| Zobrazit okno nápovědy s užitečnými tipy | Ctrl+U |
| Kontextová nápověda | F1 |
| Nápověda | Alt+F1 |
| Skok na zadávání čísla palety | F2 |
| Skok na zadávání dolního prahu | F3 |
| Skok na zadávání horního prahu | F4 |
| Zobrazit / skrýt údaje o pacientovi | Alt+P |
| Inverze palety | Alt+I |
| Reverze palety | Alt+R |
| Upravit rozsah prahování podle aktuálního posuvníku | Alt+U |
| Vrátit původní rozsah prahování | Alt+V |

Zkratky pro prohlížeč TomoAVI

| Akce | Klávesy |
|--|----------------|
| Skok na rozlišení řezů | Ctrl+F2 |
| Skok na délku animace | Ctrl+F3 |
| Skok na posuvník intenzity povrchu | Ctrl+F5 |
| Skok na rotaci modelu o násobek 360ti stupňů | Ctrl+F7 |
| Skok na aktuální čas | Ctrl+F8 |
| Skok na scrollbar časové osy | Ctrl+F9 |
| Zoom '-' časové osy | Ctrl+F11 |
| Zoom '+' časové osy | Ctrl+F12 |
| Zdrojová data - orientace | Ctrl+N |
| Zdrojová data - pořadí snímků | Ctrl+A |
| Zdrojová data - rotace | Ctrl+R |
| Zobrazit okno s nastavením renderovací metody | Ctrl+Z |
| Zobrazit okno s informacemi o zobrazeném 3D modelu | Ctrl+I |
| Skok na intenzitu povrchu | F5 |
| Skok na zadávání řezů | F6 |
| Skok na posuvník rotace X | F7 |
| Skok na posuvník rotace Y | F8 |
| Skok na posuvník přiblížení | F9 |
| Skok na x-ovou souřadnici středu rotace | F10 |
| Zobrazit / skrýt panel hlavních ovládacích prvků | F11 |

| | |
|---|-------|
| Zobrazit / skrýt panel ovládacích prvků pro tvorbu animace | F12 |
| Zvolit jednoduché řezy | Alt+J |
| Zvolit pokročilé řezy | Alt+K |
| Nastavit intenzitu pro detekci povrchu na průměrnou hodnotu | Alt+M |
| Detekovat povrch | Alt+D |
| Oříznout model zadanými řezy | Alt+O |
| Gouraudovo stínování | Alt+G |
| Flat stínování | Alt+F |
| Zobrazit okno pro nastavení barev a světla | Alt+B |
| Zobrazit menu s výběrem exportu | Alt+E |
| Vycentrovat střed otáčení modelu | Alt+C |
| Model - orientace | Alt+N |
| Model - otočit | Alt+T |
| Model - pořadí snímků | Alt+A |
| Model - postavit | Alt+S |
| Uložit aktuální nastavení modelu na časovou osu | Alt+L |
| Načíst nastavení modelu z časové osy | Alt+Z |
| Zapnout / vypnout automatické načítání z časové osy | Alt+Q |
| Zobrazit menu s výběrem operací pro klíčové body | Alt+X |

4.11 Užitečné tipy v aplikačním modulu TomoAVI

Zde naleznete několik tipů, jak zacházet s modulem TomoAVI. Nejsou zde popsány zdaleka všechny možnosti tohoto modulu.

Jednoduchý výpočet povrchu modelu

1. Stiskem tlačítka "Průměrná" v sekci "Povrch" nastavte výchozí hodnotu intenzity pro detekci povrchu nebo zadejte hodnotu ručně dle vlastního uvážení do políčka "Intenzita povrchu".
2. Tlačítkem "Detekuj povrch" nebo stiskem klávesy Enter v políčku "Intenzita povrchu" se model aktualizuje dle nové povrchové intenzity.


Jednoduchá manipulace s modelem

1. Stiskněte a držte levé tlačítko myši na renderovací ploše.
2. Pohybujte myší. Model se otáčí podle pohybu myši.
3. Všimněte si pohybu posuvníků "RotaceX" a "RotaceY". Ty slouží k preciznímu nastavování otočení modelu.
4. Uvolněte levé tlačítko.
5. Stiskněte a držte pravé tlačítko myši na renderovací ploše.
6. Pohybujte myší. Model se přibližuje a vzdaluje podle pohybu myši.
7. Všimněte si pohybu posuvníků "Přiblížení". Ten slouží k preciznímu nastavování přiblížení modelu.
8. Uvolněte pravé tlačítko.

Jednoduchý řez modelem

1. Do políčka "Řez X" v sekci "Řezy" na hlavním panelu zadejte x-ovou souřadnici řezu 0 (nula). Pokud políčko "Řez X" není v sekci "Řezy" zobrazeno, přepněte pomocí přepínače "Jednoduché" zadávání řezů na jednoduché.
2. Stiskem tlačítka "Oříznout" nebo stiskem klávesy Enter v políčku "Řez X" se vypočítá a zobrazí řez modelu rovinou kolmou na osu x posunutou na zadanou souřadnici.
3. Do políčka "Řez Y" v sekci "Řezy" na hlavním panelu zadejte y-ovou souřadnici řezu 0 (nula).
4. Stiskem tlačítka "Oříznout" nebo stiskem klávesy Enter v políčku "Řez Y" se vypočítá a zobrazí model s dvěma řezy.
5. Zkuste libovolně změnit volbu "Vyseknout klín" / "Zobrazit klín", nebo "Plus" / "Minus".
6. Stiskem tlačítka "Oříznout" se model přepočítá a opět zobrazí.

Nastavení barev na řezech

1. Pomocí tlačítka  na hlavním panelu v sekci "Barevné tabulky / Prahování" se zobrazí editor palet, ve kterém můžete editovat modifikovat barevné palety.
2. Pomocí posuvníku v sekci "Barevné tabulky / Prahování" si nastavte dolní a horní práh aktuálního prahování. Hodnotám v tomto zvoleném intervalu se přiřadí barvy z aktuálně vybrané palety.
3. Do editačního políčka zadejte číslo palety, kterou chcete použít, a stiskněte klávesu Enter. Číslo palety můžete v editačním políčku též měnit pomocí šipek nahoru/dolů. Pořadí barev resp. invertování barev změňte zaškrtnutím volby "Reverzní" resp.



"Inverzní".

4. Tlačítkem "Upravit" zjemníte rozlišovací schopnost posuvníků podle jejich aktuální pozice, tlačítkem "Původní" se nastavení tohoto rozlišení vrátí zpět na původní hodnoty.
5. V nabídce "Rozlišení řezů" zvolte vlastní rozlišení řezů podle uvážení. Všimněte si vlivu na kvalitu zobrazení a rychlost výpočtu.

Jednoduchá animace

1. Stiskem tlačítka "On/Off animace panel" v horní části okna modulu zobrazte panel pro tvorbu animace v dolní části okna.
2. Pravým tlačítkem myši nastavte aktuální čas na časové ose na hodnotu 0 (nula)
3. Stiskem tlačítka "Ulož na osu" se uloží aktuální stav modelu do klíčového bodu na časovou osu.
4. Pravým tlačítkem myši nastavte aktuální čas na časové ose na hodnotu přibližně 2.5 s.
5. Změňte některé parametry modelu, například barvy na řezu, otočení. Můžete takto animovat libovolné parametry modelu včetně řezů a intenzity povrchu (kromě položek v Změny zdrojových dat v sekci Orientace pacienta/modelu).
6. Opět stiskněte tlačítko "Ulož na osu".
7. Pravým tlačítkem myši nastavte aktuální čas na časové ose na hodnotu přibližně 5 s.
8. Opět libovolně změňte některé parametry modelu.
9. Opět stiskněte tlačítko "Ulož na osu".
10. Nastavením aktuálního času a stiskem tlačítka "Načti z osy" si můžete prohlédnout model v libovolném časovém okamžiku. Zaškrtnutím políčka "Autonačítání" se model aktualizuje z časové osy po každé změně aktuálního času.
11. Po stisku tlačítka "Exportuj..." na hlavním panelu se zobrazí menu s možnými druhy exportu. Zvolte položku "Exportuj animaci do AVI...".
12. Stiskněte tlačítko "Generuj...".
13. Zvolte cílový soubor, do kterého se animace uloží.
14. Vyberte typ komprese animace.
15. Nyní se vygeneruje animace a uloží do cílového souboru. Tato akce může trvat dlouho a je jí možné přerušit tlačítkem "Cancel".
16. Výsledný soubor je možné přehrávat v počítači standardním způsobem.

Vytvoření screenshotů a textových popisků

1. Pomocí tlačítka  v sekci "Pacient" se zobrazí editor screenshotů.
2. Po stisku tlačítka "Exportuj..." se zobrazí menu s možnými druhy exportu. Zvolte položku "ScreenShot do DB...", potvrďte následující okno. Tím vložíte do editoru screenshotů aktuální obrázek modelu.
3. Pomocí tlačítka  v sekci "Pacient" se zobrazí editor textových popisků studie, který umožňuje vytvořit nový popis ke studii.

5 Závěr

Prvním z cílů této práce bylo vytvoření uživatelsky přívětivé aplikace, která by umožňovala jednoduchou a rychlou práci s tomografickými studiemi. Několik implementovaných povrchových zobrazovacích metod z oblasti renderingu zdaleka nereprezentuje všechny možnosti zobrazení objemových tomografických dat. Povrchové zobrazování dat není jediné a snad ani nejvhodnější pro získání představy o rozložení záření v prostoru. Existuje mnoho jiných a názornějších způsobů zobrazení, jako je použití více povrchů s poloprůhledností, procházení dat podél paprsku (například MIP), kombinace více metod zobrazení a podobně. Tato aplikace je určena především pro tvorbu tomografických prezentací. Díky širokým možnostem nastavení zobrazení a mnoha exportním formátům, včetně vytváření animací a exportu 3D modelu do VRML, tento požadavek splňuje. Speciálně pro tuto aplikaci zde byla navržena metoda obecných rovinných řezů zadaných řezovým výrazem a implementovaných s pomocí tříd rovin. Dále zde byla navržena metoda částečného přepočítávání modelu při změně zadání řezů, která podstatným způsobem ovlivňuje rychlost aplikace zejména při generování animací. V neposlední řadě byl pro rychlost a použitelnost aplikace klíčový návrh a implementace vylepšeného diferenčního algoritmu pro výpočet textur na řezech. V budoucnu by bylo možné rozšířit aplikaci o jiné než povrchové zobrazovací metody, které by byly pro zobrazení intenzity v prostoru názornější.

Druhým cílem této práce bylo provést srovnání implementovaných zobrazovacích metod. Byla odhalena slabá místa některých metod a některé metody se ukázaly přímo jako nevhodné. Některé metody vyšly ze srovnávacích testů naopak velice dobře, pro důkladnější ověření chování metod za všech okolností by bylo potřeba provést mnohem více testů na mnohem větším souboru dat, to by vyžadovalo automatizované generování testovacích dat a automatické vyhodnocování výsledků. Ideální výsledek zobrazení je těžko definovatelný, data prostorových studií nesou málo informací na přesnější rekonstrukci původního povrchu pod úrovní voxelu. Bylo by možné navrhnout algoritmus, který by v datech hledal složitější souvislosti a podle nich se řídil při rekonstrukci povrchu i normál povrchu. Je otázkou, zda v zobrazování medicínských dat má takové řešení opodstatnění.

Použitá literatura:

- [1] Aravind Krishnaswamy: *Semi-automatic Generation of Transfer Functions for Direct Volume Rendering. Gradient Estimators.*
http://www.aravind.ca/cs788h_Final_Project/cs_788h_s01_project.htm
- [2] Jiří Žára (2002): Přednáška *Virtual Reality. Introduction to VRML*
<http://www.cgg.cvut.cz/~zara/PGR012/>
- [3] Jiří Žára a kol. (1992): *Počítačová grafika: Principy a algoritmy.*
Grada, Praha
- [4] Josef Pelikán: Přednáška *Počítačová grafika I, II a III*
<http://cgg.ms.mff.cuni.cz/lectures/>
- [5] Kris Thielemans (2001): *A preliminary proposal for Interfile keys for PET (version 0.3)*
<http://www.cu.mrc.ac.uk/~kris/petinterfile.html>
- [6] Mark Segal, Kurt Akeley (1997): *The OpenGL Graphics System: A Specification.*
Silicon Graphics, Inc.
- [7] Norman Chin, Chris Frazier, Paul Ho, Zicheng Liu, Kevin P. Smith (1998): *The OpenGL Graphics System Utility Library.*
Silicon Graphics, Inc.
- [8] Paul Bourke (1997): *Polygonising a scalar field.*
<http://astronomy.swin.edu.au/pbourke/modelling/polygonise/>
- [9] Paul Bourke (1997): *Polygonising a Scalar Field Using Tetrahedrons.*
<http://astronomy.swin.edu.au/pbourke/modelling/polytetra/>
- [10] Rakesh Mullick, PhD, R. Nick Bryan, MD, PhD, and John Butman, MD, PhD:
Confocal Volume Rendering: Fast Segmentation-Free Visualization of Internal Structures.
Diagnostic Radiology Department, Clinical Center, National Institutes of Health (NIH), Bethesda and Department of Radiology, University of Pennsylvania Health System 3400 Spruce St., Philadelphia: APPENDIX A
- [11] Tang Ming-Jye Hu (1998): *Implementation of Data Viewer and Dividing Cubes Algorithm. Part II: Dividing Cubes Algorithm.*
<http://www.cs.umbc.edu/~ebert/693/THu/>
- [12] Ulf Tiede, Karl Heinz Hoehne, Michael Bomans, Andreas Pommert, Martin Riemer, and Gunnar Wiebecke (1990): *Investigation of Medical 3D-Rendering Algorithms.*
IEEE Computer Graphics & Applications, 41-53