

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLAZOVÁNÍ POLYGONÁLNÍCH MODELŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADEK SVĚCHOVSKÝ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLAZOVÁNÍ POLYGONÁLNÍCH MODELŮ

POLYGONAL MODELS SMOOTHING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

RADEK SVĚCHOVSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. PŘEMYSL KRŠEK, Ph.D.

BRNO 2010

Abstrakt

Digitalizace objektů či převod 3D modelů do podoby povrchové reprezentace s sebou přináší nedokonalosti v podobě šumu. Tato práce rozebírá známé přístupy k eliminaci zašumění polygonálních modelů. Čtenář bude srozuměn se základními principy vyhlazování a především s výsledky srovnání jednotlivých metod mezi které patří Laplaceova metoda, algoritmus Laplace-HC, Taubinův filtr dolní propusti a bilaterální filtr.

Abstract

Object digitizing or 3D model transformation into surface representation brings defects in the form of noise. This thesis analyses the well-known approaches to the noise elimination from polygonal models. The reader will be concerned the fundamental principles of smoothing and foremost the results of the comparison of different methods including Laplace method, algorithm Laplace-HC, Taubin's low-pass filter and bilateral filter.

Klíčová slova

polygon, polygonální model, hraniční reprezentace, šum, odstranění šumu, vyhlazování, Laplace, Laplace-HC, Taubinův filtr, bilaterální filtr, BMDS

Keywords

polygon, polygonal model, boundary representation, noise, denoising, smoothing, Laplace, Laplace-HC, Taubin's filter, bilateral filter, BMDS

Citace

Radek Svěchovský: Vyhlazování polygonálních modelů, bakalářská práce, Brno, FIT VUT v Brně, 2010

Vyhlazování polygonálních modelů

Prohlášení

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

.....
Radek Svěchovský
17. května 2010

Poděkování

Je mou milou povinností zde, na tomto místě, poděkovat především vedoucímu mé bakalářské práce, panu docentu Přemyslu Krškovi, jenž byl ochoten věnovat mi svůj čas, rady a připomínky, které udělaly tuto práci lepší. Dále bych také velmi rád poděkoval všem autorům knihovny MDSTk, která mi byla poskytnuta a umožnila mi soustředit se pouze na jádro práce.

© Radek Svěchovský, 2010.

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

Obsah

1	Úvod	2
2	3D objekty a jejich reprezentace	3
2.1	Implicitní plochy	3
2.2	Constructive solid geometry	3
2.3	Objemová reprezentace těles	4
2.4	Hraniční reprezentace těles	4
3	Vyhlazovací metody	6
3.1	Simultánní a sekvenční uložení bodů	6
3.2	Laplaceova vyhlazovací metoda	6
3.3	Algoritmus Laplace-HC	8
3.4	Taubinův filtr dolní propusti	9
3.5	Bilaterální filtr	9
4	Návrh a implementace	10
4.1	Výběr programovacího jazyka a nástrojů	10
4.2	Modulární struktura	10
4.3	Návrh a realizace testovacích skriptů	12
4.4	Implementace vyhlazovacích metod	13
5	Výsledky testování	14
5.1	Laplaceova metoda	14
5.2	Algoritmus Laplace-HC	16
5.3	Taubinův filtr dolní propusti	17
5.4	Bilaterální filtr	18
5.5	Vzájemné srovnání vyhlazovacích metod	20
6	Závěr	22
A	Obrazové výsledky vyhlazování	24
B	Vybrané statistické výsledky	38
C	Obsah DVD	42

Kapitola 1

Úvod

S rozvojem digitalizace reálných objektů vyvstává taktéž potřeba automatické úpravy výsledných dat. Do skupiny automatizovaných modifikací hraničních modelů patří několik příbuzných technik, a sice zjednodušování sítě trojúhelníků známé pod názvem decimace, přesouvání vrcholů do lepších pozic, angl. remeshing, nebo právě vyhlazení modelu. Ať už je zdrojem modelu digitální 3D scanner (laserový nebo dotykový) nebo jej získáme převodem pomocí známých algoritmů z jiné 3D reprezentace, například z reprezentace objemové vzniklé transformací snímků počítačové tomografie nebo magnetické rezonance, disponuje takový model přidaným šumem, který je vhodné, vzhledem k velkému množství vrcholů, odstranit automatizovaně. Právě eliminací šumu s cílem co nejlépe zachovat původní rysy objektu se zabývá tato bakalářská práce, která si klade za cíl především porovnat některé z již existujících vyhlazovacích algoritmů z hlediska kvality vyhlazení, které je posuzováno hned podle několika kritérií, mezi které patří objem tělesa, rozsah zaoblení původně ostrých hran či celková míra vyhlazení.

V prvních dvou kapitolách dojde k teoretickému rozboru věci, především bude popsána polygonální reprezentace a dále bude čtenář seznámen se samotnými vyhlazovacími metodami a principy, na kterých vyhlazování pracuje. V další kapitole je pak nastíněn návrh a implementace aplikace, testovacího a vyhodnocovacího modulu. V následující části jsou prezentovány dosažené výsledky testování a porovnání jednotlivých vyhlazovacích metod. Závěrečná kapitola shrnuje dosažené výsledky a navrhuje případná rozšíření práce. V příloze **A** lze pak vidět obrazové výsledky některých vyhlazených testovacích modelů.

Kapitola 2

3D objekty a jejich reprezentace

Na reprezentaci 3D objektů v informačních technologiích jsou kladeny různorodé požadavky. Za jistých okolností vyžadujeme například pouhé zobrazení povrchu v co nejkratším čase, naopak metrické vlastnosti objektu, jeho vnitřní obsah nebo to, zda je vyrobitelný, jsou nepodstatné. Typické použití takových modelů je v zábavním průmyslu, případně se užívá pro rychlé zobrazení jinak definovaných modelů. V jiných případech je cílem zachování vnitřní struktury objektu, nebo jde o přesné geometrické vyjádření pomocí primitiv umožňující analytické výpočty.

Kromě těchto specifických požadavků jsou zde také požadavky obecné, jako je jednoznačnost interpretace, obecnost, co možná nejmenší paměťové nároky a další. Proto také existuje více skupin 3D reprezentací, které se zaměřují na splnění kýžených požadavků. Než bude věnován větší prostor pro popis *polygonálních modelů*, alespoň přehledově budou v této kapitole uvedeny i některé další typy modelů, které se mohou, po převodu na polygonální model, stát zdrojem dat pro vyhlazovací algoritmy. Některé informace byly čerpány z [7].

2.1 Implicitní plochy

Implicitní plocha¹ je souhrnný název, zavedený v roce 1988 J. Bloomenthalem, pro objekty, které jsou modelovány pomocí kostry a potenciálových funkcí. Základním elementem kostry je tzv. generátor, kterému je vždy přiřazena potenciálová funkce. Vyhodnocením potenciálových funkcí všech prvků kostry získáme pro libovolný bod v prostoru intenzitu potenciálového pole. Podle zadaného prahu T se určuje, zda je daný bod uvnitř, vně, nebo tvoří povrch objektu.

Mezi výhody takového modelování patří míra abstrakce (i složitější modely lze vyjádřit pomocí několika málo generátorů) a navíc je model úsporný z hlediska paměťových nároků. Nevýhodou pak může být zobrazení, které je nutno provádět raycastingem, případně řešit převodem na jiný, nejčastěji polygonální, model. Takový model poté vyžaduje úpravy jako je decimace, nebo vyhlazení.

2.2 Constructive solid geometry

CSG nebo také česky konstruktivní geometrie těles je dalším typem reprezentace 3D objektů. Je založena na stromové struktuře, kde listy stromu tvoří geometrická primitiva

¹Z angl. implicit surface, jinak známá též pod názvy Blobby objects, Soft objects nebo Meta balls.

(koule, krychle apod.), v uzlech vyššího řádu se pak uchovávají boolovské množinové operace – průnik, sjednocení a rozdíl, nebo transformace jakou je posunutí. I CSG je nutné zobrazovat raycastingem nebo převést na hraniční reprezentaci.

2.3 Objemová reprezentace těles

Objemová reprezentace se užívá nejčastěji, pokud neznáme přesný geometrický popis objektu. Známý mohou být pouze vzorky, které mají možnost uchovat i data neobrazového charakteru (např. hustotu). Základní jednotkou je voxel², který je uspořádán do pravidelné kartézské mřížky. Tato reprezentace se mimo jiné získává také ze snímků magnetické rezonance nebo počítačové tomografie. Převod na polygonální model se provádí algoritmem Marching cubes.

Samotný algoritmus není předmětem zkoumání této práce, ale jeho výsledkem je polygonální model, který je vhodné upravit decimací a následně vyhladit, protože disponuje nežádoucím charakteristickým šumem, který není vlastní zvláště organickým objektům.

2.4 Hraniční reprezentace těles

Pravděpodobně nejužívanější reprezentací 3D objektů v počítačové grafice je reprezentace hraniční³. Do této kategorie spadá několik typů popisu tělesa: drátový model, polygonální model a spline model. Jak název napovídá, jsou tyto objekty definovány pouze svým povrchem, tedy hranicí v prostoru, informace o vnitřním obsahu tělesa se neuchovává. Modely jsou pak popsány neprázdnou množinou vertexů (česky označované jako vrcholy), množinou hran a u polygonálních a spline modelů také případně pomocí množiny stěn.

2.4.1 Polygonální model

Polygonální model je nestrukturovaná, nejčastěji trojúhelníková (může být také složena ze čtyřúhelníků nebo mnohoúhelníků), síť definovaná vrcholy, hranami a stěnami. Výhodou trojúhelníkových sítí je, že oproti polygonům vyššího řádu není potřeba kontrolovat, zda vrcholy tvořící jednu stěnu polygonu opravdu geometricky náleží jedné rovině. Od ostatních druhů 3D reprezentace se polygonální model liší tím, že je po částech lineární a nespojitý a proto má smysl aplikovat vyhlazování. Výhodou je také, že jej lze zobrazovat v reálném čase, protože operace nezbytné pro vyobrazení jsou hardwarově akcelerovány v grafických čípech.

V paměti může být polygonální model reprezentován naivně pomocí seznamu vrcholů, seznamu hran a seznamů ploch, kde každý záznam hrany obsahuje právě dva ukazatele a záznam plochy právě tři ukazatele do seznamu vrcholů. Tato reprezentace je sice úsporná z hlediska paměťových nároků, nicméně získání některých topologických informací (např. vrchol sdílející stejnou hranu) je výpočetně náročnější. Existují však i složitější struktury, které umožňují rychlejší přístup k okolním plochám či hranám a mohou uchovávat i další informace mezi které patří například normály vrcholů nebo normály stěn. Tím ale klesá poměr geometrických a topologických informací. Mezi takové datové struktury patří okřídlená hrana, kterou však nelze použít pro nonmanifold⁴ modely, nebo tzv. půlhrana.

²Podobně jako je pro 2D rastrové obrazy pixel.

³Angl. boundary representation, zkráceně také B-rep.

⁴Nonmanifold je fyzicky nevyrobitelný objekt. Obsahuje nekonečně tenké hrany či bodové spoje.

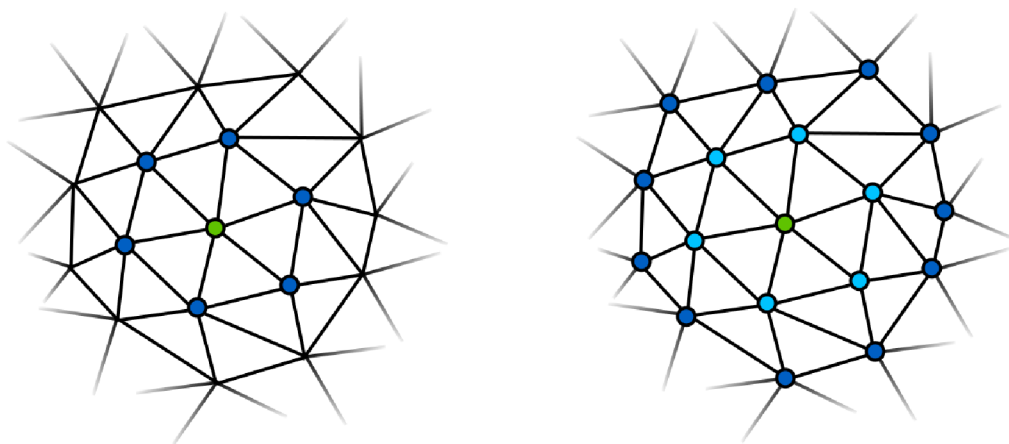
Podle článku J. Vollmera [5] lze obecný polygonální model přesně matematicky popsat jako n -tici (K, \mathbf{p}) , kde je $K \subseteq 2^V$ simplicialní komplex⁵, $\mathbf{p} : V \rightarrow \mathbb{R}^3$ je mapovací funkce vertexu v prostoru a V je neprázdná množina vrcholů. $E = \{e \in K \mid |e| = 2\}$ je pak množina hran a $F = \{f \in K \mid |f| = 3\}$ je množina stěn. Vzhledem k tomu, že vyhlazovací metody zkoumané v této práci nijak nemodifikují topologii polygonálního modelu a dochází pouze k úpravě geometrické informace vrcholů lze pro definici okolí vrcholu užít analogii s teorií grafů.

2.4.2 Okolí vrcholu

U vyhlazovacích algoritmů, jež pracují na principu modifikace polohy vrcholu na základě pozic okolních vrcholů, může vyvstat otázka, jak velké množství bodů bude do výpočtu nové pozice zahrnuto. V teorii grafů [2] je množina sousedních vrcholů $\text{adj}(x_i)$ ⁶ definována jako

$$\text{adj}(x_i) = \{x_j \in V \mid \{x_i, x_j\} \in E\} \quad (2.1)$$

Velikost této množiny je pak známa jako stupeň vrcholu $|\text{adj}(x_i)|$ a její převrácená hodnota se dále může použít jako vyvažovací koeficient pro určitý vrchol. Stejnou definici je možné aplikovat na polygonální modely, výsledná množina se nazývá *okolí vrcholu*, nebo také okolí prvního řádu [4]. Okolí vyššího řádu lze definovat rekurzivně a to tak, že výsledná množina sousedů je sjednocením okolí vrcholů tvořících okolí o řád nižší a samotným okolím o řád nižším. Nejlépe je to patrné na obrázku 2.1, kde je vlevo okolí prvního řádu a vpravo okolí řádu druhého, jako sjednocení světle a tmavě modrých vrcholů.



Obrázek 2.1: Okolí vrcholu

⁵Více o simplicialním komplexu a n -simplexu lze nalézt v [3].

⁶Také značena jako i^* .

Kapitola 3

Vyhlazovací metody

V této kapitole budou nejprve popsány obecné vlastnosti vyhlazovacích metod a posléze bude nastíněn princip porovnávaných a testovaných vyhlazovacích algoritmů, mezi které patří:

- Laplaceův algoritmus
- Algoritmus Laplace-HC
- Taubinův filtr dolní propusti
- Bilaterální filtr

3.1 Simultánní a sekvenční uložení bodů

Po aplikaci vyhlazovacího algoritmu na aktuálně zpracovávaný vrchol získáme novou pozici, kterou je následně potřeba jistým způsobem uchovat. Přístup k tomuto problému může být dvojitý. První možností je okamžité přepsání starých souřadnic vertexu souřadnicemi novými. Tím nenastává potřeba dodatečného paměťového prostoru, nicméně dochází k ovlivnění okolních vrcholů nově vypočítanými pozicemi. Navíc je zřejmé, že výsledek vyhlazení je ovlivněn pořadím vrcholů v datové struktuře. Tento přístup se nazývá *simultánní*¹.

Druhý, *sekvenční*², přístup ukládá nové souřadnice do pomocné kolekce a následně, až dojde v jedné iteraci k průchodu přes všechny vrcholy objektu, jsou staré pozice nahrazeny novými. Tento přístup je jak paměťově, tak i mírně výpočetně náročnější, z důvodu režie spojené s vkládáním a vyjímáním souřadnic z dočasného úložiště, ale nedisponuje žádným vedlejším efektem v podobě ovlivnění výsledku vyhlazení.

3.2 Laplaceova vyhlazovací metoda

Laplaceův algoritmus³ je nejjednodušší variantou vyhlazování a jak je z názvu patrné, vychází z diskrétní aproximace Laplaceova operátoru [6], která je vyjádřena vzorcem

$$\Delta x_i = \sum_{j \in \text{adj}(i)} w_{ij}(x_j - x_i) \quad (3.1)$$

¹Z angl. simultaneous version

²Z angl. sequential version

³V materiálech prof. Taubina je nazýván jako Gaussův filtr. [4]

kde x_i je aktuálně zpracovávaný vrchol, x_j je vrchol náležící okolí $\text{adj}(x_i)$ a w_{ij} je ohodnocení hrany mezi vrcholy x_i a x_j . Podle [5] bude i zde zavedena konvence, kde \mathbf{o}_i je originální bod, \mathbf{q}_i je aktuální bod před aplikací algoritmu a \mathbf{p}_i je bod po modifikaci pozice. Celý algoritmus je pak popsán následující rovnicí

$$\mathbf{p}_i = \mathbf{q}_i + \lambda \cdot \Delta \mathbf{q}_i \quad (3.2)$$

kde \mathbf{p}_i a \mathbf{q}_i dodržuje výše zmíněnou definici a λ je *vyvažovací koeficient*⁴ z intervalu $(0, 1)$. Tento činitel může být proměnný s ohledem na aktuální vrchol, nebo může být definován jako konstanta pro celý polygonální model.

Pokud je parametr w_{ij} v rovnici 3.1 konstantní z pohledu vyhlazovaného vertexu (tato situace nastává například při vážení hran převrácenou hodnotou stupně vrcholu – viz 2.4.2), lze jednoduchými úpravami

$$\begin{aligned} \Delta x_i &= w_{ij} \cdot \sum_{j \in \text{adj}(i)} (x_j - x_i) \\ \Delta x_i &= w_{ij} \cdot \left(\sum_{j \in \text{adj}(i)} x_j - |\text{adj}(x_i)| x_i \right) \end{aligned}$$

dojít ke tvaru operátoru

$$\Delta x_i = w_{ij} \cdot \sum_{j \in \text{adj}(i)} x_j - w_{ij} |\text{adj}(x_i)| x_i$$

V případě, kde je w_{ij} právě rovno převrácené hodnotě $|\text{adj}(x_i)|$ nabyde operátor tvar

$$\Delta x_i = \frac{1}{|\text{adj}(x_i)|} \cdot \sum_{j \in \text{adj}(i)} x_j - x_i$$

Po dosazení do vzorce 3.2 získáme tvar

$$\begin{aligned} \mathbf{p}_i &= \mathbf{q}_i + \lambda \cdot \left(\frac{1}{|\text{adj}(\mathbf{q}_i)|} \cdot \sum_{j \in \text{adj}(i)} \mathbf{q}_j - \mathbf{q}_i \right) \\ \mathbf{p}_i &= (1 - \lambda) \mathbf{q}_i + \frac{\lambda}{|\text{adj}(\mathbf{q}_i)|} \cdot \sum_{j \in \text{adj}(i)} \mathbf{q}_j \end{aligned} \quad (3.3)$$

který umožňuje rychlejší výpočet, jelikož nedochází k neustálému, v tomto případě zbytečnému, odčítání a násobení. Pokud je zároveň λ konstantní a rovna 1 je výsledný tvar algoritmu

$$\mathbf{p}_i = \frac{1}{|\text{adj}(\mathbf{q}_i)|} \cdot \sum_{j \in \text{adj}(i)} \mathbf{q}_j \quad (3.4)$$

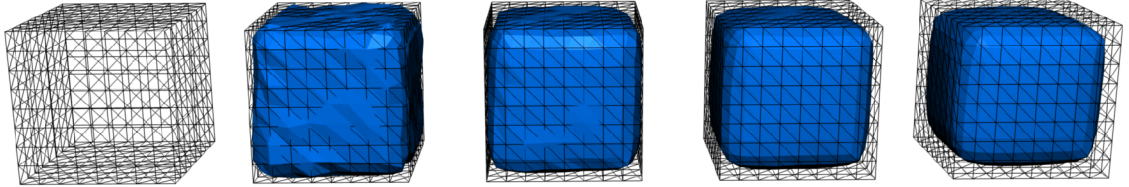
kde je nová pozice vrcholu vypočítána jako aritmetický průměr poloh vrcholů okolních. V [5] je právě tento tvar prezentován jako základní podoba Laplaceova algoritmu. V tomto článku je také kromě variant 3.3 a 3.4 uvedena ještě další modifikace ve tvaru

$$\mathbf{p}_i = (1 - \lambda) \mathbf{o}_i + \frac{\lambda}{|\text{adj}(\mathbf{q}_i)|} \cdot \sum_{j \in \text{adj}(i)} \mathbf{q}_j \quad (3.5)$$

⁴Z angl. scale factor

kde o_i je originální pozice vrcholů zašuměného modelu, neměnná v průběhu výpočtu a nezávislá na počtu iterací.

Laplaceova vyhlazovací metoda ve všech třech variantách však disponuje přinejmenším dvěma, dost podstatnými, nevýhodami. První z nich je smršťování vyhlazovaného modelu⁵, které v extrémním případě může vést ke smrštění do jednoho bodu [5]. Na obrázku 3.1 je vidět znatelné zmenšení objemu už po čtyřech iteracích.



Obrázek 3.1: Smrštění objektu po aplikaci Δ

Druhou nevýhodou pak je, že metoda žádným způsobem nechrání ostré hrany modelu, které jsou makroskopické v porovnání se šumem, který se snažíme odstranit.

3.3 Algoritmus Laplace-HC

První modifikací Laplaceova algoritmu je metoda Laplace-HC, která si klade za cíl zbavit Laplaceovu metodu smršťování. Základní myšlenka spočívá v tom, že se po jedné iteraci Laplaceova algoritmu 3.3 provede oprava, která posune body zpět směrem k jejich původním pozicím. Korekce je dána vzorcem

$$\mathbf{d}_i = \beta \mathbf{b}_i + \frac{1 - \beta}{|\text{adj}(i)|} \cdot \sum_{j \in \text{adj}(i)} \mathbf{b}_j$$

kde \mathbf{b}_i a \mathbf{b}_j jsou vektory dané vztahem

$$\mathbf{b}_i = \mathbf{p}_i - (\alpha \mathbf{o}_i + (1 - \alpha) \mathbf{q}_i)$$

Celý algoritmus je tedy následující

$$\begin{aligned} \mathbf{p}_i &= (1 - \lambda) \mathbf{q}_i + \frac{\lambda}{|\text{adj}(\mathbf{q}_i)|} \cdot \sum_{j \in \text{adj}(i)} \mathbf{q}_j \\ \mathbf{b}_i &= \mathbf{p}_i - (\alpha \mathbf{o}_i + (1 - \alpha) \mathbf{q}_i) \\ \mathbf{p}_i &= \mathbf{p}_i - \left(\beta \mathbf{b}_i + \frac{1 - \beta}{|\text{adj}(i)|} \cdot \sum_{j \in \text{adj}(i)} \mathbf{b}_j \right) \end{aligned} \quad (3.6)$$

Přibývají tak dva nové parametry α a β , které ovlivňují výsledek vyhlazení, naopak λ bývá nastavena na 1, čímž se dosahuje co možná nejrychlejšího vyhlazování. Důkaz konvergence této metody lze nalézt v materiálu autorů [5].

⁵Angl. shrinkage

3.4 Taubinův filtr dolní propusti

I vyhlazovací metoda prof. Taubina [4] se snaží odstranit problém se smršťováním objektu vyhlazovaného Laplaceovým algoritmem. Dosahuje se toho poněkud jednodušším způsobem než u algoritmu HC. Filtr vychází z Laplaceovy metody 3.3, ale je zde navíc zaveden záporný parametr μ . Vztah s parametrem μ vypadá následovně

$$\mathbf{p}_i = (1 - \mu)\mathbf{q}_i + \frac{\mu}{|\text{adj}(\mathbf{q}_i)|} \cdot \sum_{j \in \text{adj}(i)} \mathbf{q}_j \quad (3.7)$$

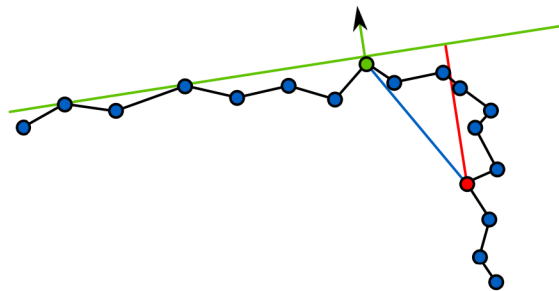
Vtaky 3.3 a 3.7 se ve výpočtu střídají po jedné iteraci a tak dochází k opětovnému smršťování a roztahování. Musí platit že $0 < \lambda < -\mu$, jinými slovy záporný koeficient μ musí být v absolutní hodnotě větší než lambda. Při správném nastavení parametrů se algoritmus chová jako filtr dolní propusti⁶. Více o výpočtu koeficientů se lze dozvědět v [4].

3.5 Bilaterální filtr

Poslední vyhlazovací metodou, která byla zahrnuta do testování, je bilaterální filtr⁷ [1]. Princip vyhlazení je inspirován bilaterálním filtrováním 2D rastrového obrazu. Od předchozích metod se mírně liší v tom, že se změna pozice vrcholu provádí pouze ve směru jeho normály a také dochází k dvojímu vážení vlivu okolních bodů na celkovou změnu. Jednak je Gaussovým filtrem

$$G(x) = e^{-\frac{x^2}{2\sigma^2}}$$

vyvažována absolutní vzdálenost vrcholu (na obrázku 3.2 vyznačena modrou úsečkou). Čím je bod v rámci okolí vzdálenější, tím menší měrou ovlivňuje změnu pozice. Dále je pak vážena vzdálenost od roviny definované vyhlazovaným bodem a jeho normálou (na řezu vyznačena červenou úsečkou). Toto vážení by mělo omezit vliv vrcholů, které pravděpodobně náležejí jiné rovině v prostoru, čímž se zabrání zaoblení hran modelu. Zachování ostrých hran je totiž cíl, kterého se snaží BMDS dosáhnout.



Obrázek 3.2: Vzdálenosti ovlivňující výpočet pomocí BMDS

⁶Angl. low-pass filter

⁷Známý pod zkratkou BMDS z angl. Bilateral Mesh Denoising

Kapitola 4

Návrh a implementace

V této části budou ve zkratce popsány využití nástroje, návrh výsledného programu a některé ne úplně obvyklé implementační detaily.

4.1 Výběr programovacího jazyka a nástrojů

Jako implementační jazyk bylo vybráno C++, které se stále těší relativně velké oblibě mezi programátory, jelikož umožňuje implementaci aplikací podle objektového návrhu a zároveň, vzhledem k tomu, že je to jazyk kompilovaný, jsou výsledné programy rychlé. Rychlost, ač byla také jedním z důvodů výběru tohoto jazyka, jelikož testovací modely mají relativně velké množství vrcholů, se ukázala jako nepodstatná. Limitujícím faktorem při testování bylo především množství generovaných dat. Druhým důvodem pro výběr C++ pak byla dostupnost knihoven pro práci s polygonálními modely.

V této práci je využita knihovna *Medical Data Segmentation Toolkit* (dále jen MDSTk, dostupná z URL <http://mdstk.sourceforge.net/>) ve verzi 0.8.1, která je vyvíjena už řadu let na Ústavu počítačové grafiky a multimédií FIT VUT v Brně. Jako vývojové prostředí byly zvoleny *NetBeans*, nově s podporou C++. Mezi další nástroje, hojně využívané při zpracování výsledků, patří *gnuplot*, *MeshLab* a *Blender*, programová dokumentace je generována pomocí programu *Doxygen*. Pro testování byla vytvořena sada BASH skriptů (viz 4.3), které je možno pomocí *MinGW* spouštět i pod operačními systémy MS Windows, ačkoli vývoj probíhal na Linuxu.

4.2 Modulární struktura

Od počátku byla aplikace navrhována jako skupina menších modulů, bez GUI, které by se daly jednoduše kombinovat s cílem dosažení vyšší funkcionality. Toto chování bylo realizováno snadno díky podpoře modulů v knihovně MDSTk, které jsou pak mezi sebou ve skriptech provázány pomocí anonymních rour. Kolekce se tedy skládá celkem z pěti modulů, jejichž význam bude letmo popsán.

4.2.1 Modul mdLoad a mdSave

Tyto dva moduly zajišťují pouze načítání a ukládání formátu STL, které je realizováno pomocí interních metod jedné z částí MDSTk, *VectorEntity*.

4.2.2 Modul mdNoise

V množině testovacích modelů této práce se bohužel nacházejí jen objekty vyhlazené, nebo uměle vytvořené a tudíž je bylo nutné umělým způsobem zašumět. Toto zajišťuje modul `mdNoise`, který na daný model aplikuje Gaussův šum, určený normálním rozdělením se střední hodnotou v nule. Směrodatná odchylka pro Gaussův šum je vypočítána jako $\sigma = e \cdot k$, kde e je střední hodnota délky hrany modelu a k je konstanta, kterou uživatel ovlivňuje míru šumu. Implicitně je k nastaveno na 0.2.

4.2.3 Modul mdFilter

Hlavním modulem je `mdFilter`, kterým jsou zastřešeny vyhlazovací algoritmy. Přepínači na vstupu uživatel ovlivňuje vyhlazovací metodu, počet iterací a jednotlivá nastavení daných vyhlazovacích algoritmů. Některé implementační zajímavosti jsou uvedeny v 4.4, detaily pak v programové dokumentaci v příloze C.

4.2.4 Modul mdStat

Poslední z pěti modulů je modul `mdStat`, který zajišťuje porovnání dvou 3D objektů. Referenční model je předáván parametrem přes příkazovou řádku ve formě názvu souboru. Vzhledem k tomu, že vyhlazovací metody nemění topologii objektů, nejsou povrchy nijak vzorkovány a porovnávají se sobě odpovídající plochy a vrcholy.

První monitorovanou veličinou je odchylka normál trojúhelníků, druhou pak absolutní vzdálenost mezi vrcholy promítnutá ve směru normály referenčního vrcholu, čímž se získá odchylka od roviny tvořené referenčním bodem a jeho normálou. Z těchto dvou veličin je následně počítána střední hodnota a směrodatná odchylka. Zároveň jsou uchovány jednotlivé odchylky pro možnost dalšího zpracování skripty např. do podoby histogramů. Poslední kontrolovanou vlastností je objem modelů. Ten je počítán intuitivním způsobem, lze jej však určit pouze u objektů tvořených povrchem bez děr a jiných artefaktů. V opačném případě by došlo k vyčíslení neodpovídajícího objemu. Algoritmus je dán následující posloupností operací:

1. Mimo objekt se vytvoří referenční plocha.
2. Pro všechny trojúhelníky se pomocí tří čtyřstěnů (tří 3-simplexů) určí objem mezi povrchem a referenční plochou.
3. Následně se tento dílčí objem k objemu celkovému přičte, nebo se od něj odečte, v závislosti na směru normály.

4.3 Návrh a realizace testovacích skriptů

Testování a srovnávání výsledků jednotlivých metod je zajištěno pomocí hierarchické kolekce skriptů. V prvotním návrhu byla zvažována možnost testování hrubou silou, to zn. že by parametry jednotlivých metod byly nastavovány automaticky pro všechny kombinace s určitým pevně daným krokem. Od tohoto řešení bylo upuštěno hned z několika důvodů. Jednak by bylo mírně složitější danou věc realizovat vzhledem k proměnnému množství parametrů a jejich rozsahů pro jednotlivé metody a jednak by byl zapotřebí mnohem větší diskový prostor, nebo by musela být implementována automatická analýza, která by už v průběhu testů odstraňovala nedostatečně vyhlazené výsledky. Tato analýza by také zvýšila celkový čas testování, který by už tak byl téměř neúnosný, zvláště pro algoritmus HC, který disponuje třemi parametry, což zapříčiňuje kubickou složitost při tomto typu testování. Jiným přístupem k této věci by mohlo být zjemňování kroku posunu parametrů se zlepšujícími se výsledky, například metodou půlení intervalu.

Řešení implementované v této práci je odlišné a není plně automatické. Spočívá v jednotlivých testech s jedním konkrétním nastavením pro určitou vyhlazovací metodu, kterým se předá jako parametr `model` k vyhlazení a porovnání. Tyto jednotky jsou plně funkční, autonomní a lze je vytvářet pomocí generátoru `scriptBuilder.sh`, což značně zkracuje čas nutný pro výrobu testů. Spouštět testy ručně jeden po druhém by však nebylo efektivní, proto byl vytvořen skript `trigger.sh`, který přebírá jako parametr jeden z testů a spustí jej nad všemi zašuměnými modely. Na nejvyšší úrovni v této hierarchii je pak skript `test.sh`, který, je-li to zapotřebí, přeloží moduly, aplikuje šum na testované modely, následně prochází adresář s jednotlivými testy a předává je jako argument skriptu `trigger.sh`. Tento přístup umožňuje v krátké době vygenerovat uživatelem požadované testy¹, spustit `test.sh` a po určitém čase bez zásahu kontrolovat výsledky.

Podobně jako v případě testů, byla i analýza výsledků vytvořena ve více vrstvách. První ze skriptů na nižší úrovni je `table.sh`, který prochází adresáře s výsledky a pro stejné testy, lišící se jen počtem iterací, vytváří přehledové tabulky statistických hodnot nad všemi modely ve formátu `csv` a \LaTeX . Druhým skriptem je `histogram.sh`, který pro každý model dělá histogram odchylek normál a histogram vzdáleností mezi vrcholy. Dále pak vytváří graf aproximací histogramů pro jeden model s různým počtem iterací, ve kterém je vidět jakým způsobem se vyhlazování vyvíjí s přibývajícím počtem opakování². Tyto dva skripty spouští `analyse.sh`, který po skončení odstraní všechny nepotřebné log soubory.

Mimo tyto dvě skupiny pak stojí `volume.sh`, který pouze provádí verifikaci výpočtu objemu modulem `mdStat` u objektů, pro které je možné jej vypočítat analyticky. Ukazuje se, že u těles, kde analytické řešení neobsahuje výpočty nad kružnicí (z testovaných to je krychle a jehlan), je odchylka v řádu 10^{-6} . Naopak u těles, které mají kruhovou podstavu (válec, kužel) nebo u koule je odchylka v řádu 10^{-3} , což je ale do značné míry způsobeno konečným počtem³ vrcholů, které tvoří mnohoúhelníkovou aproximaci kružnice.

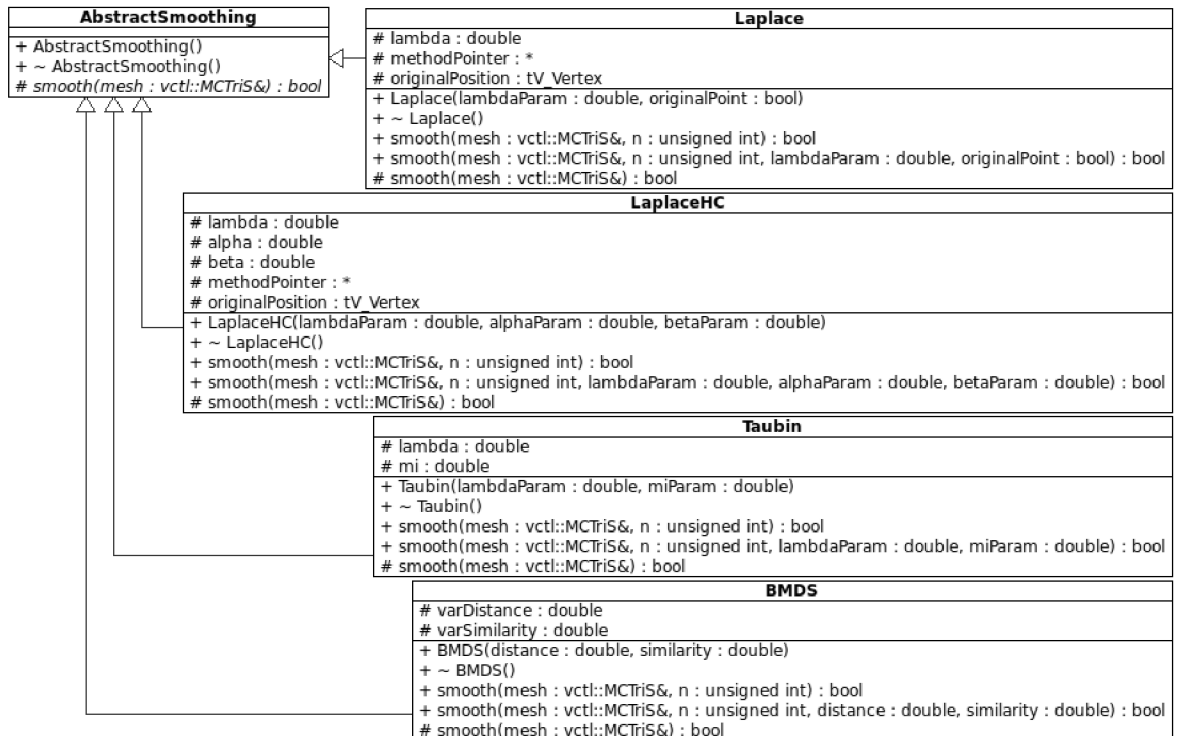
¹I test s nevhodným nastavením může být informačně zajímavý a přitom by byly jeho výsledky v případě heuristického řešení smazány

²Aproximace spojnicovým grafem byla nutná, na překrývajících se histogramech lze jen velmi těžce odlišit jednotlivé sloupce od sebe

³V případě testovacích modelů je to 96 vrcholů.

4.4 Implementace vyhlazovacích metod

Implementace filtrů probíhala podle zjednodušeného diagramu tříd na obrázku 4.1.



Obrázek 4.1: Zjednodušený diagram tříd

Důraz byl kladen především na výpočetní rychlost. Proto jsou všechny konstanty neměnicí se v průběhu výpočtu vyčísleny jen jednou a všude, kde je to možné, bylo využito inline metod. Také například u třídy Laplace byly zvlášť implementovány všechny tři varianty podle vztahů 3.3, 3.4 a 3.5, ačkoli implementace vzorce, který je platný pouze při parametru $\lambda = 1$ není vůbec nutná. Tato optimalizace minimalizuje počet operací, který je nezbytné provést během výpočtu. K výběru příslušné metody dochází pouze jednou a to na začátku vyhlazování. K tomuto chování je využita síla jazyka C++ v podobě ukazatele na metodu.

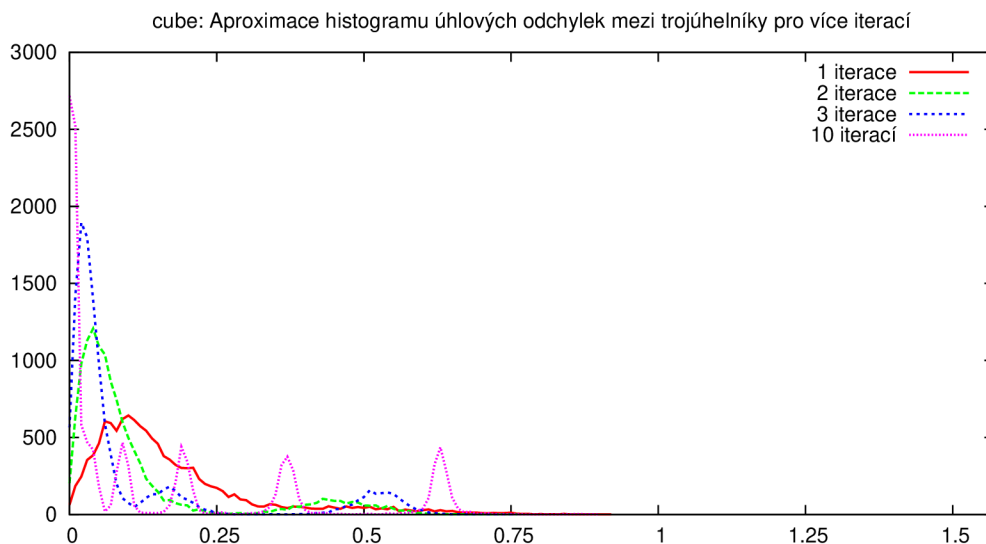
Kapitola 5

Výsledky testování

V následující kapitole budou rozebrány vybrané dílčí výsledky testů jednotlivých vyhlazovacích metod, posléze bude prezentováno srovnání metod v rámci nejspokojivějších dosažených výsledků. Všechny metody byly testovány pro více různých nastavení a iterací, celkově bylo provedeno okolo dvou set testů. Přehled testovacích modelů a tabulka s parametry jednotlivých metod se nachází v příloze A před obrazovými výsledky.

5.1 Laplaceova metoda

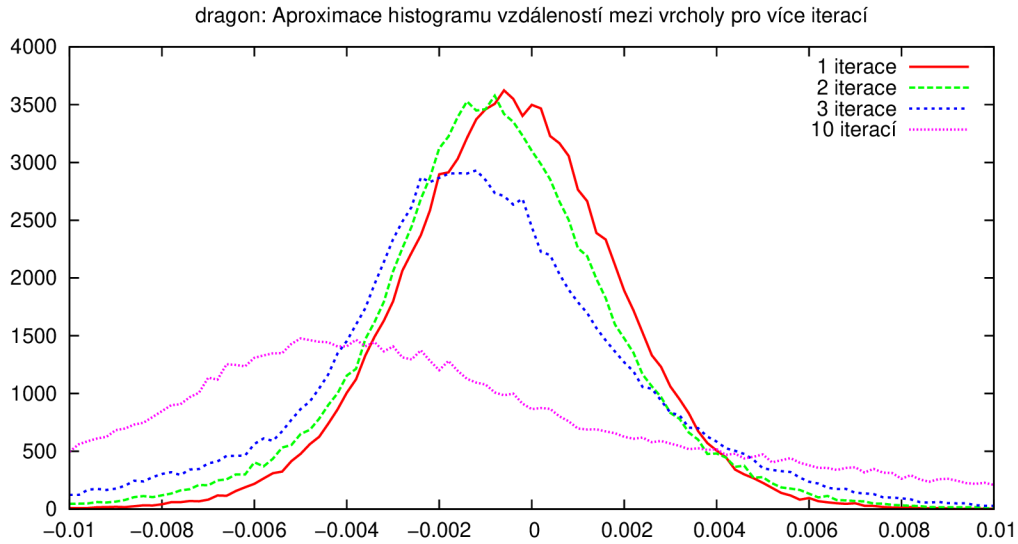
Prvním zajímavým, ale ne příliš vhodným nastavením Laplaceova algoritmu je $\lambda = 1$. Zde se potvrzují teoretické předpoklady a sice, že se model v průběhu vyhlazování smršťuje. Objekty jsou vyhlazené již po dvou či třech iteracích, což je vidět také na grafu 5.1, který vyjadřuje aproximaci histogramů úhlů sevřených normálami trojúhelníků na modelu *cube*.



Obrázek 5.1: Model cube

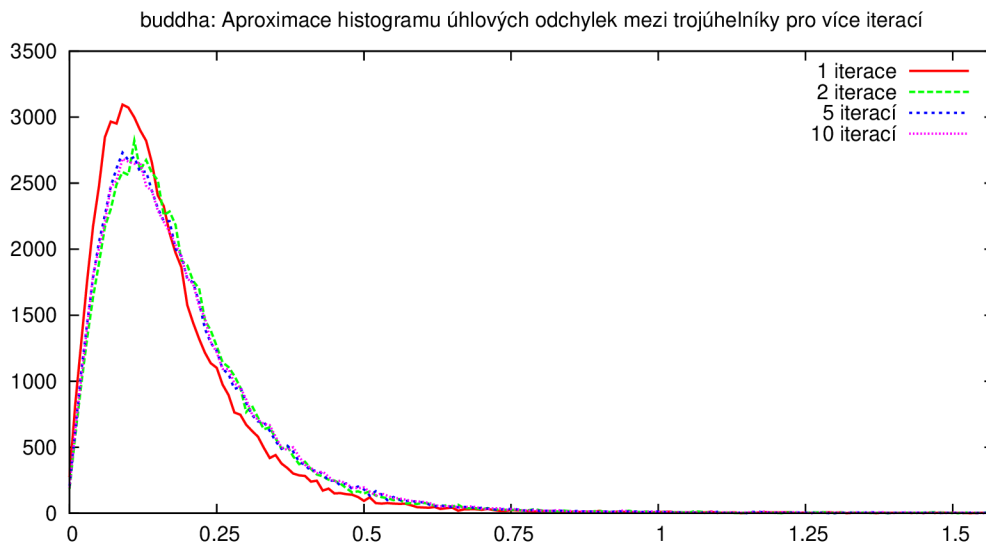
Již při třech iteracích má křivka relativně vysoké maximum a strmý průběh, nicméně si nelze nevšimnout dalších menších lokálních maxim, zejména pak pro 10 iterací, se středem podstatně posunutým k vyšším hodnotám. Tyto maxima způsobuje zaoblení ostrých hran, které se zvláště projevuje na geometrických obrazcích. Na grafu 5.2, na kterém je

vyobrazena vzdálenost mezi odpovídajícími si vrcholy pro model *dragon*, pak lze pozorovat smršťování objektu, kde, místo aby se rozptýl zmenšoval a docházelo ke zúžení, se křivka znatelně zplošťuje, její maximum klesá a střed se posunuje do záporných hodnot. Po třech opakováních algoritmu je objem průměrně 98,395 %. Další statistické údaje lze pro jiné metody nalézt v příloze B.



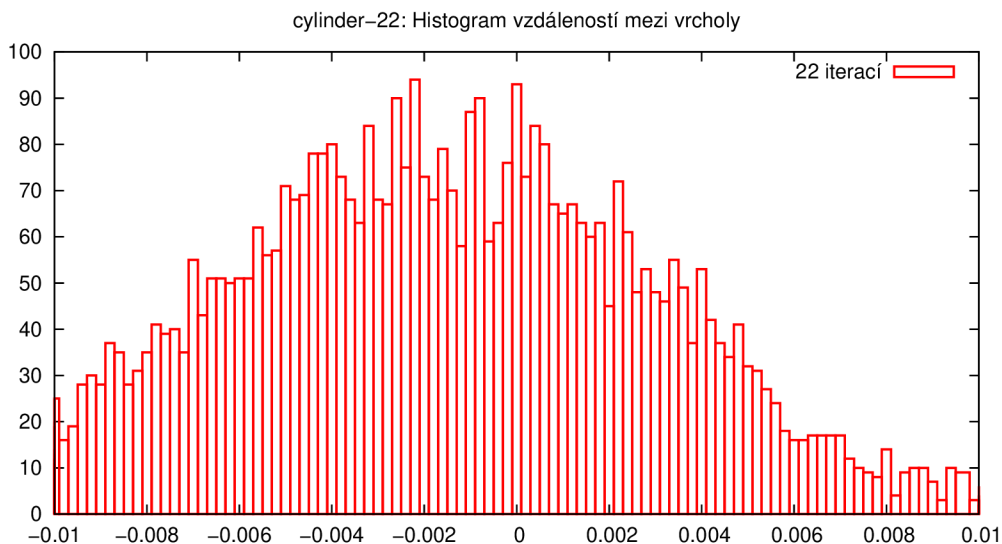
Obrázek 5.2: Model dragon

Dalšímu testování byla podrobena Laplaceova metoda s použitím originálních bodů podle vzorce 3.5 přejatého z [5]. Zde byl parametr λ roven 0,72. Tato metoda si, i přes relativně vysoký koeficient λ , vlivem originálních pozic vrcholů zahrnutých do výpočtu, udržuje původní objem o něco lépe. Po pěti iteracích je střední hodnota objemu 98,86 %. Podle grafu 5.3 na modelu *buddha* je patrné, že se při tomto nastavení po několika málo iteracích téměř úplně zastaví vyhlazování a dochází jen k pomalému smršťování, na kterém je však stále patrný šum.



Obrázek 5.3: Model buddha

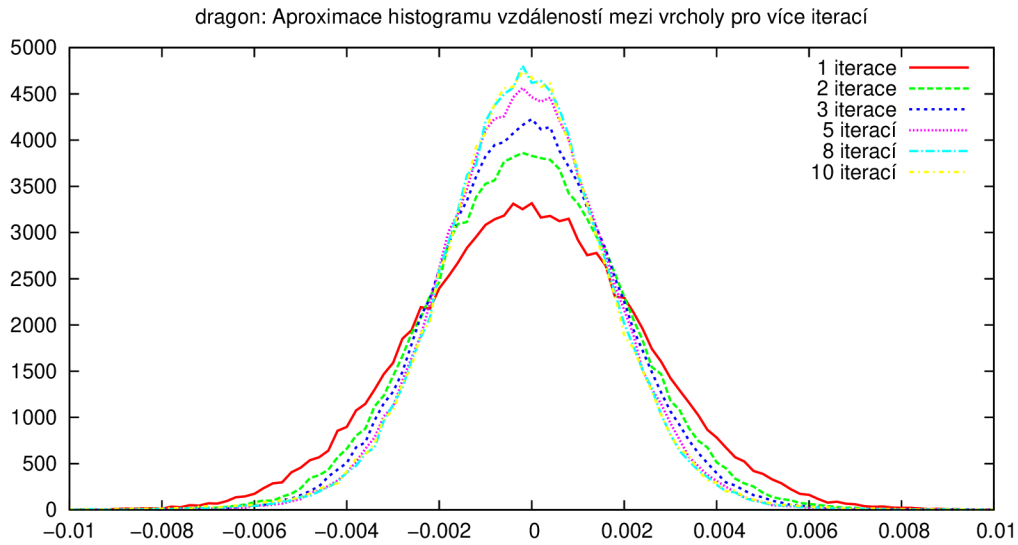
Nastavení se kterým bylo dosaženo uspokojivých výsledků se od předchozích značně liší. Koefficient λ byl nastaven na 0,08, což zapříčiní pomalejší postup vyhlazování. Výsledky v příloze A jsou vyobrazeny po 22 iteracích, i zde je však patrné nechtěné smršťování, které je vidět i v histogramu 5.4 v podobě posunutí středu křivky normálního rozložení k záporným hodnotám. Objem těles byl v průměru 99,020 %.



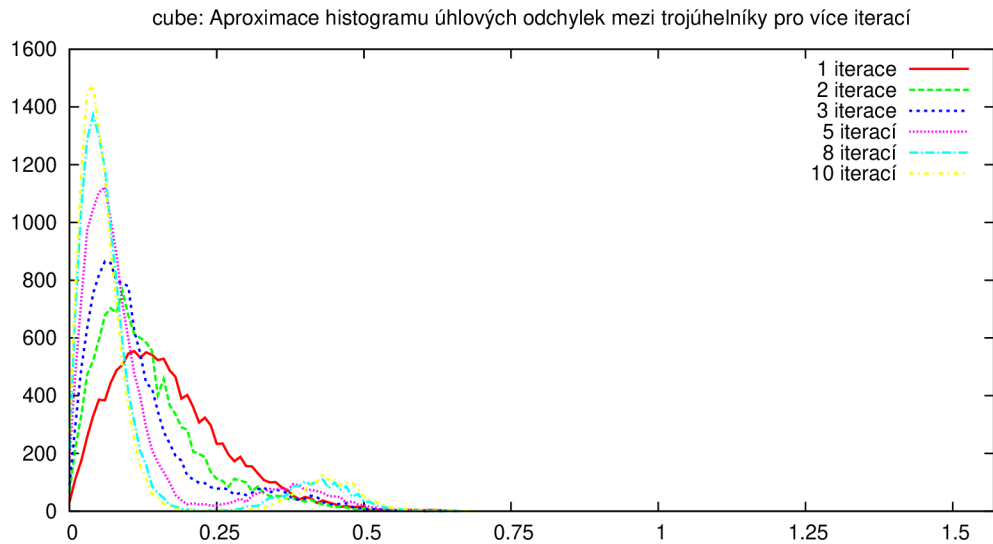
Obrázek 5.4: Model cylinder

5.2 Algoritmus Laplace-HC

U algoritmu HC bylo velmi dobrých výsledků dosaženo s argumenty $\lambda = 1,0$, který způsobuje rychlý postup vyhlazování, $\alpha = 0,0$ a $\beta = 0,5$ pro korekturní část algoritmu. Prezentované obrazové výsledky jsou získány po 8 iteracích pro všechny testované objekty. V grafu 5.5 nad modelem *dragon* je vidět, že si objekty vyhlazované touto metodou udržují objem, protože nedochází k posouvání středu Gaussovy křivky do záporných hodnot. Zároveň se zužuje oblast 3 sigma a maximum křivky roste, což znamená, že se vrcholy přibližují očekávaným pozicím. Průměrný objem je v tomto případě 99,898 %, při některých testech dokonce přesáhl 100 % objemu originálního modelu. Celkové vyhlazení lze pozorovat v grafu 5.6 nad modelem *cube*, kde je stále patrné mírné zaoblení v podobě lokálních maxim okolo 0,42 radiánu.



Obrázek 5.5: Model dragon



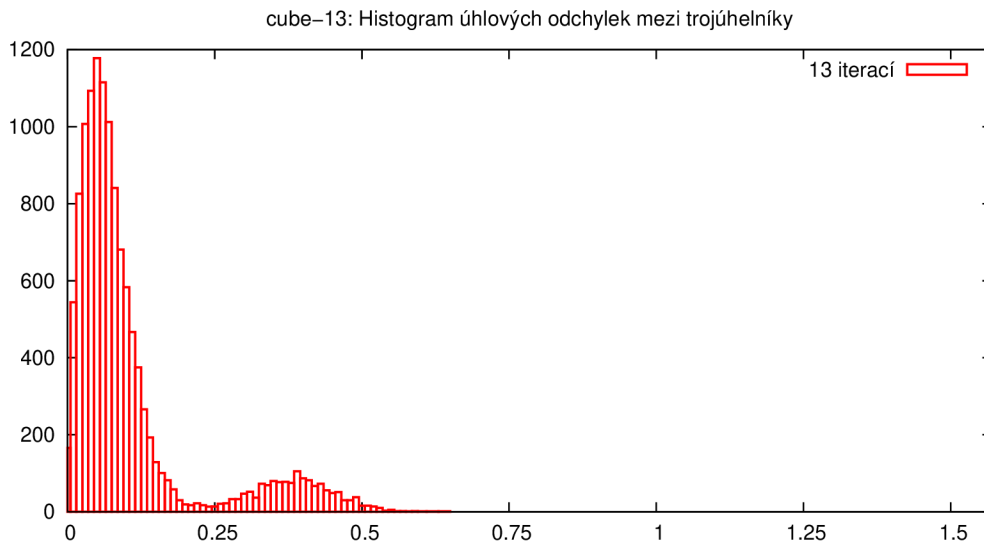
Obrázek 5.6: Model cube

5.3 Taubinův filtr dolní propusti

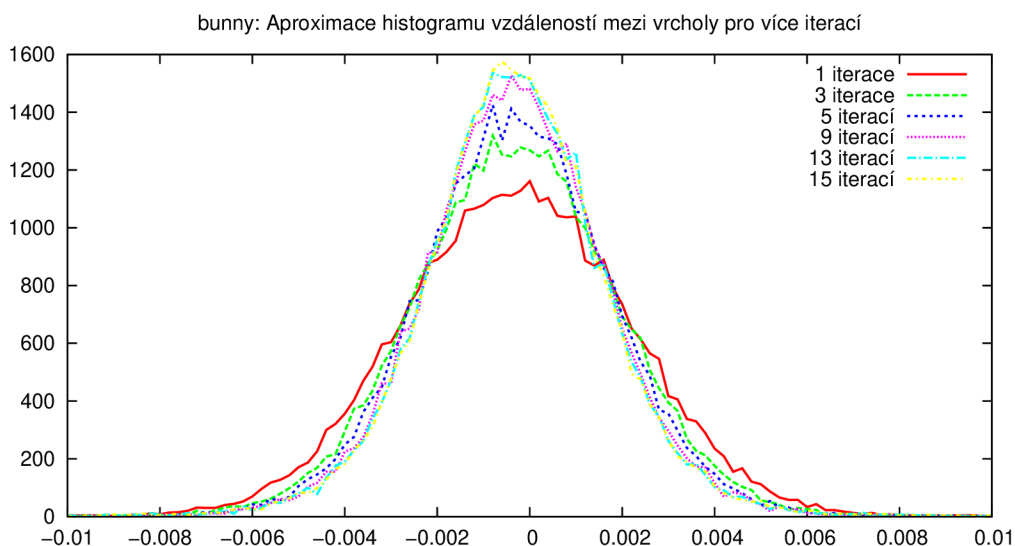
Příslušné obrazové výsledky v příloze A byly získány aplikací Taubinova filtru s nastavením $\lambda = 0,53$ a $\mu = -0,54$. Na vyhlazení bylo potřeba 13 iterací¹ a objem těles se pohyboval v průměru na hodnotě 99,670 %². Výsledky vyhlazení jsou podobné HC algoritmu, i zde je možno pozorovat malé odchylky na ostrých hranách v 5.7 pro 13 iterací nad modelem krychle. Jednoduchý princip tohoto algoritmu zároveň podle předpokladů zabraňuje smrštění, což je patrné na grafu vzdáleností vrcholů 5.8, kde se střed Gaussovy křivky téměř nepohybuje.

¹Jedna iterace je počítána zvlášť buď s použitím parametru λ nebo μ , nikoli jako dvojice průchodů.

²Měření bylo zaznamenáno vždy po liché iteraci, tj. po iteraci s kladným koeficientem λ , kdy dochází ke smrštění.



Obrázek 5.7: Model cube

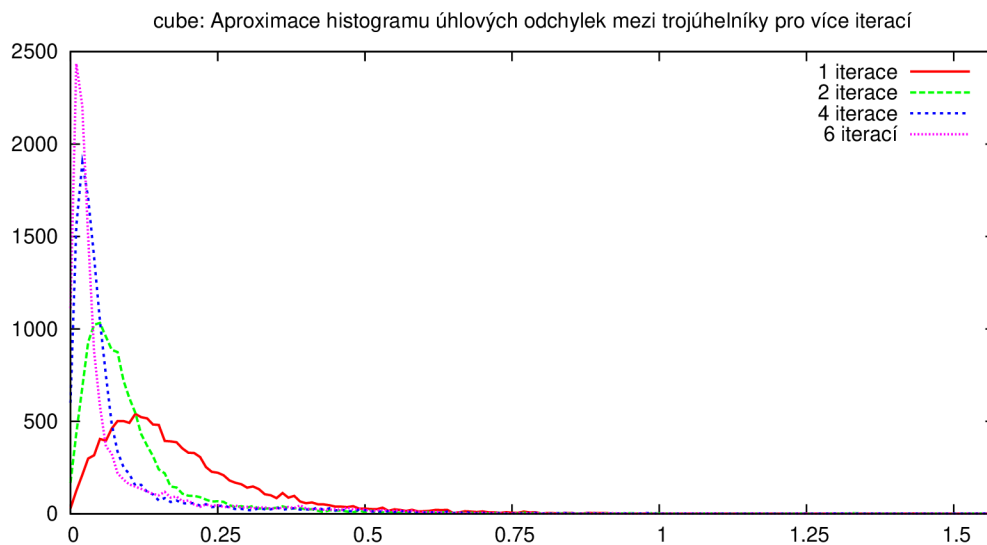


Obrázek 5.8: Model bunny

5.4 Bilaterální filtr

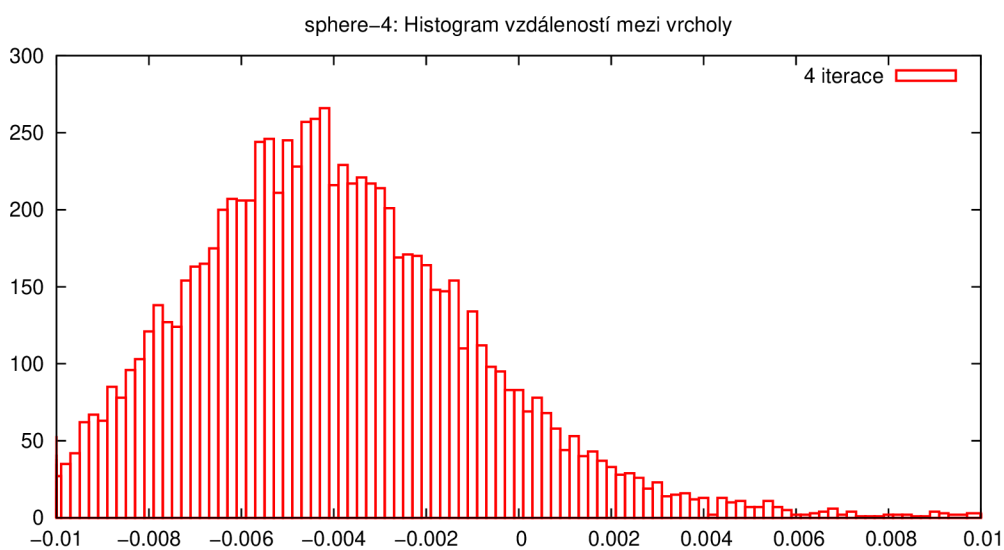
Pro bilaterální filtr bylo uděláno nejvíce testů, nicméně je třeba zmínit, že se zřejmě nepodařilo najít ideální nastavení pro komplexní objekty ze stanfordského repozitáře 3D modelů, jelikož vyhlazení nedosahovalo předpokládaných výsledků. Geometrické obrazce byly vyhlazovány s nastavením $\sigma_c = 0,2$ a $\sigma_s = 0,01$. Na grafu 5.9, pro srovnání opět nad modelem krychle, lze pozorovat vyhlazení bez menších lokálních maxim, což potvrzuje, že tento přístup při správném nastavení chrání ostré hrany na modelu.

U geometrických těles s kruhovou podstavou nebo u koule však došlo k zdatelnému smrštění, které dokazuje jak histogram 5.10 tak vizualizace na straně 36 v příloze A. U složitějších modelů *bunny*, *buddha* a *dragon* bylo aplikováno nastavení $\sigma_c = 0,02$, $\sigma_s = 0,01$, respektive $\sigma_c = 0,01$ a $\sigma_s = 0,005$, nicméně se nepodařilo dosáhnout nijak oslnivých výsledků,



Obrázek 5.9: Model cube

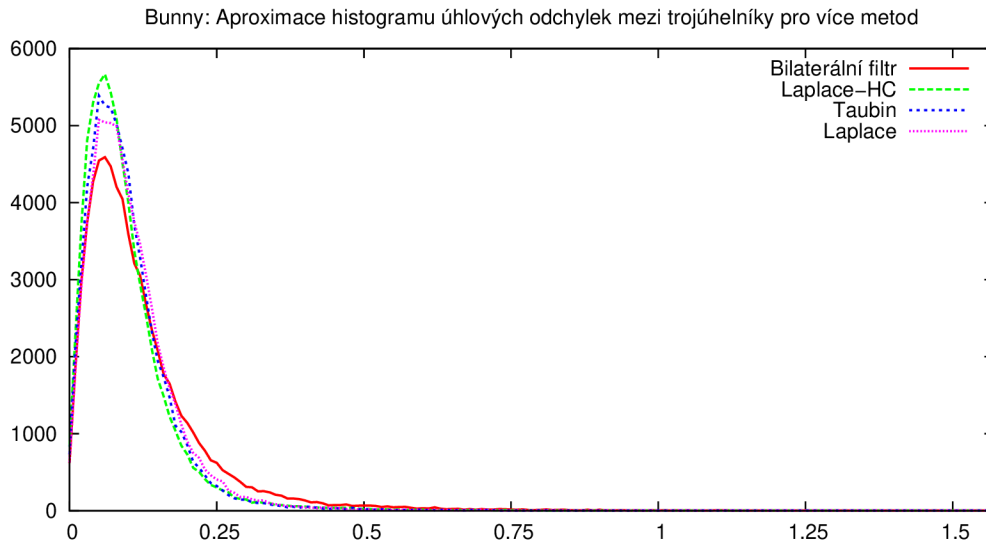
což bude patrné především v následující kapitole 5.5, kde jsou metody srovnány mezi sebou. Průměrný objem vyhlazených těles pomocí BMDS je 99,128 % a bylo potřeba maximálně čtyř iterací.



Obrázek 5.10: Model sphere

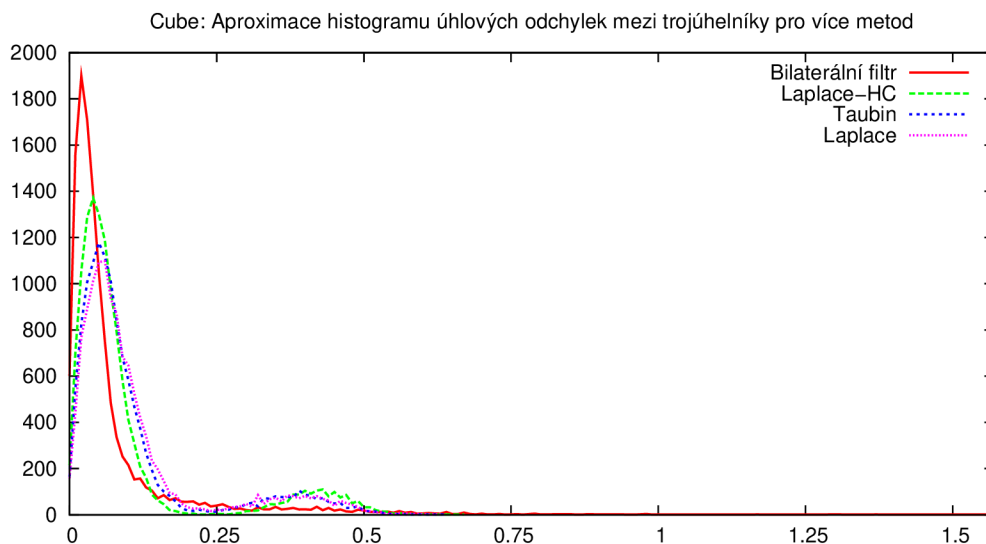
5.5 Vzájemné srovnání vyhlazovacích metod

Srovnání probíhalo mezi nejspokojivějšími výsledky pro jednotlivé algoritmy. Míru vyhlazení lze pro model *bunny* pozorovat na grafu 5.11, kde je zřetelně nejhorší bilaterální filtr, což je pravděpodobně způsobeno špatným nastavením metody.

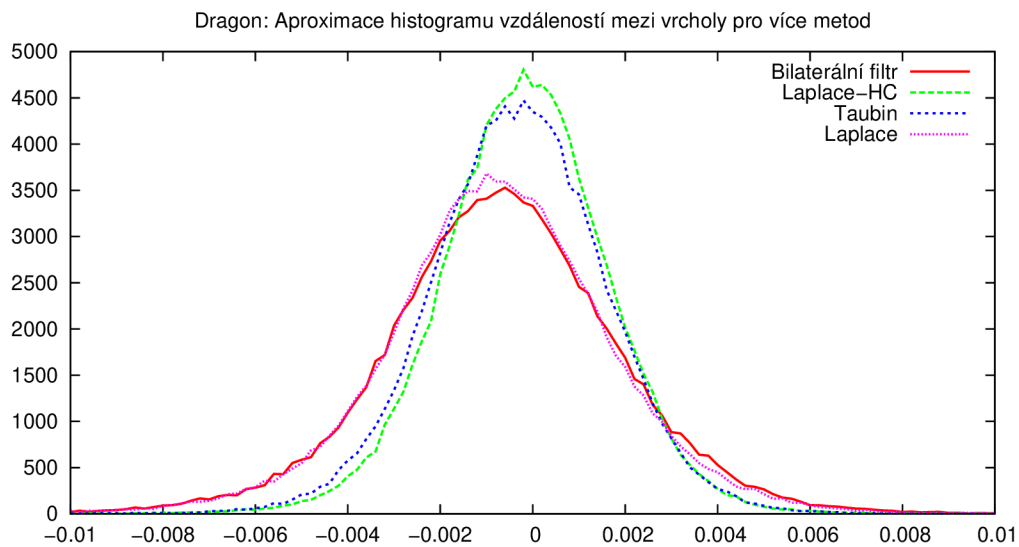


Obrázek 5.11: Model bunny

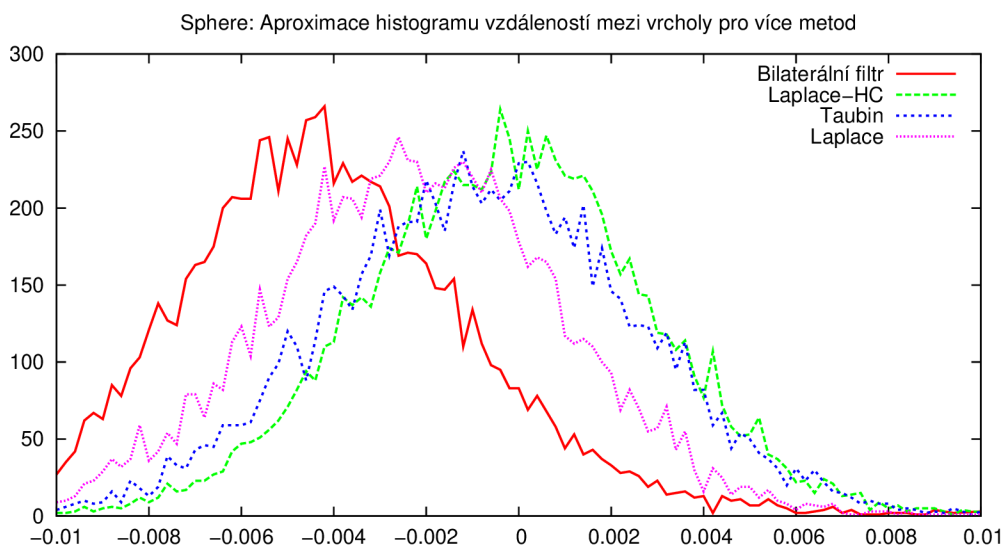
Pro další modely ze stanfordského repozitáře je průběh podobný. Naopak u geometrických obrazců je BMDS nejlepším algoritmem, což dokazuje graf 5.12 vytvořený nad modelem krychle. V grafech porovnávajících absolutní pozici vrcholů je bilaterální filtr pro stanfordské modely opět na úrovni Laplaceova algoritmu (viz graf 5.13). U geometrických těles pak má BMDS nejvyšší maximum, střed křivky bývá ale posunut do záporných hodnot. Nejmarkantnější je to na grafu modelu koule 5.14.



Obrázek 5.12: Model cube



Obrázek 5.13: Model dragon



Obrázek 5.14: Model sphere

Další grafy spolu s úplnými tabulkami statistických hodnot lze nalézt v příloze C, vybrané obrazové výsledky pak v příloze A.

Kapitola 6

Závěr

Na základě výsledků porovnávajících vyhlazovací metody mezi sebou lze konstatovat, že je situace odlišná pro geometrické obrazce s ostrými hranami a pro složitější modely ze stanfordského repositáře, které jsou spíše zaoblené. Na geometrických obrazcích je jasným vítězem BMDS. Hrany zůstávají ostré, rovné plochy jsou hladké, nicméně u některých obrazců došlo ke smrštění, takže v absolutních pozicích vrcholů mírně pokulhává. Nejhorší je pak Laplaceovo vyhlazování, kde je výsledek kompromisem mezi vyhlazením a mírou smrštění objektu. Na modelech získaných ze The Stanford 3D Scanning Repository je pořadí poněkud jiné. Nejlépe vyhlazoval algoritmus Laplace-HC, následován s minimálními rozdíly Taubinovým filtrem. Nejhorší výsledky naopak poskytoval bilaterální filtr, ve kterém je však v tomto směru pravděpodobně skrytý potenciál, jelikož se dá na základě [1] předpokládat, že se bohužel nepodařilo dosáhnout nejlepších výsledků z důvodu nesprávné kombinace parametrů.

Stanovených cílů, a sice implementovat některé ze známých vyhlazovacích metod a vytvořit testovací systém, který umožní porovnat výsledky vyhlazení mezi sebou, se podařilo dosáhnout. Implementovány byly čtyři metody, konkrétně Laplaceův algoritmus, modifikace Laplace-HC, Taubinův filtr a bilaterální filtr. Sada testovacích skriptů produkuje dostatečné množství užitečných grafů, histogramů a tabulek, na jejichž základě lze metody efektivně porovnávat.

Rozšíření této práce spočívá v implementaci dalších vyhlazovacích metod, případně ve vlastní invenci vyhlazovacích algoritmů. Také by mohlo být zajímavé znovu zvážit automatické nastavování parametrů a analýzu výsledků, což by vedlo k co možná nejlepšímu individuálnímu nastavení a tím i ke kvalitnějším výsledkům. V neposlední řadě by bylo užitečné rozšířit sadu testovacích objektů o zástupce obsahující přirozený šum (například modely po převodu algoritmem Marching Cubes nebo výstupy z 3D scanneru), protože ten má jiné vlastnosti než uměle vytvořený Gaussův šum.

Literatura

- [1] FLEISHMAN, S., DRORI, I. a COHEN OR, D. Bilateral Mesh Denoising. In *Proceedings of ACM SIGGRAPH 2003* [online]. New York: ACM, 2003 [cit. 24. dubna 2010]. S. 950–953. Dostupné z URL: www.cs.tau.ac.il/~dcor/online_papers/papers/shachar03.pdf. ISSN 0730-0301.
- [2] HLINĚNÝ, P. *Teorie Grafů* [online]. Brno: Fakulta informatiky Masarykovy univerzity, 2009, 25. listopad 2009 [cit. 22. dubna 2010]. 122 s. Dostupné z URL: <http://www.fi.muni.cz/~hlineny/Vyuka/GT/Grafy-text09.pdf>.
- [3] KOLEKTIV AUTORŮ. *Aplikovaná matematika II: M až Ž*. 1. vyd. Praha: SNTL - Nakladatelství technické literatury, 1978. 1139-2386 s.
- [4] TAUBIN, G. Curve and Surface Smoothing Without Shrinkage. In *Proceedings of the 5th International Conference on Computer Vision (ICCV '95)* [online]. Washington: IEEE Computer Society, 1995, Last updated April 07, 2008 [cit. 20. dubna 2010]. S. 852–857. Dostupné z URL: <http://mesh.brown.edu/taubin/pdfs/taubin-iccv95a.pdf>. ISBN 0-8186-7042-8.
- [5] VOLLMER, J., MENCL, R. a MÜLLER, H. Improved Laplacian Smoothing of Noisy Surface Meshes. In *EuroGraphics. Proceedings of the 20th Conference European Association for Computer Graphics (EuroGraphics '99)* [online]. Dortmund: Blackwell Publishers, 1999, Modifikace am 05.01.2009 [cit. 20. dubna 2010]. Dostupné z URL: <http://www.joerg-vollmer.de/downloads/article.pdf>.
- [6] WARDETZKY, M., MATHUR, S., KÄLBERER, F. et al. Discrete Laplace operators: No free lunch. In *Proceedings of the 5th Eurographics symposium on Geometry processing* [online]. Barcelona: Eurographics Association, 2007 [cit. 20. dubna 2010]. S. 33–37. Dostupné z URL: www.cs.columbia.edu/cg/pdfs/1180993110-laplacian.pdf. ISSN 978-3-905673-46-3.
- [7] ŽÁRA, J., BENEŠ, B., SOCHOR, J. et al. *Moderní počítačová grafika*. 2. vyd. Brno: Computer Press, 2004. 609 s. ISBN 80-251-0454-0.

Dodatek A

Obrazové výsledky vyhlazování

A.1 Modely a jejich vlastnosti

V následující tabulce je seznam testovaných modelů a jejich vlastností.

Název	Počet vrcholů	Počet hran	Počet trojúhelníků
buddha ¹	32328	97945	67240
bunny ¹	35949	104502	69668
cone	4226	12672	8448
cube	6146	18432	12288
cylinder	5154	15456	10304
dragon ¹	100250	301207	202520
pyramid	3074	9216	6144
sphere	9122	27360	18240

Tabulka A.1: Přehled modelů

Tabulka A.2 obsahuje jednotlivá nastavení vyhlazovacích metod, kterými bylo dosaženo zobrazovaných výsledků. Znak # označuje počet iterací.

Název	Laplace		Laplace-HC				Taubin			BMDS		
	#	λ	#	λ	α	β	#	λ	μ	#	σ_c	σ_s
buddha	22	0,08	8	1,0	0,001	0,5	13	0,53	-0,54	2	0,01	0,005
bunny	22	0,08	8	1,0	0,001	0,5	13	0,53	-0,54	3	0,02	0,01
cone ²	22	0,08	8	1,0	0,001	0,5	13	0,53	-0,54	4	0,2	0,01
cube	22	0,08	8	1,0	0,001	0,5	13	0,53	-0,54	4	0,2	0,01
cylinder ²	22	0,08	8	1,0	0,001	0,5	13	0,53	-0,54	4	0,2	0,01
dragon	22	0,08	8	1,0	0,001	0,5	13	0,53	-0,54	3	0,01	0,005
pyramid	22	0,08	8	1,0	0,001	0,5	13	0,53	-0,54	4	0,2	0,01
sphere	22	0,08	8	1,0	0,001	0,5	13	0,53	-0,54	4	0,2	0,01

Tabulka A.2: Přehled nastavení vyhlazování

¹Tento model byl získán ze stanfordského repozitáře. <http://graphics.stanford.edu/data/3Dscanrep/>

²Grafické výsledky tohoto modelu jsou pouze v příloze C.

A.2 Metro

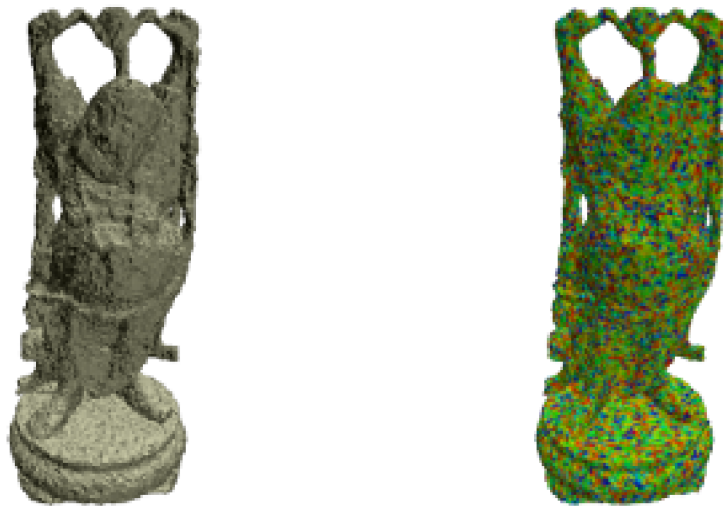
Pro vizualizaci posunutí vrcholů byl využit program *metro* (dostupné z URL <http://vcg.sourceforge.net/index.php/Metro>). Nastavení barev se řídí podle přechodu na obrázku A.1, kde červená značí nulovou odchylku a modrá je odchylka maximální. Pro modely přejaté z The Stanford 3D Scanning Repository je maximum nastaveno na odchylku 0,008. U geometrických těles je maximum 0,02, aby bylo co nejlépe využito celé spektrum barev.



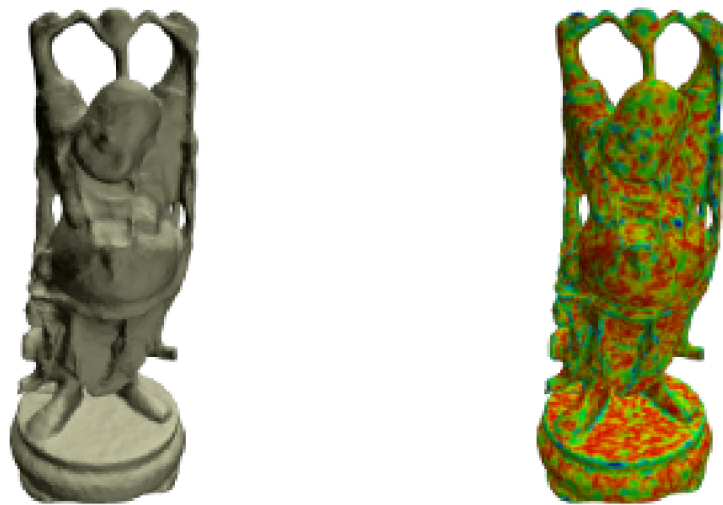
Obrázek A.1: Barevný přechod pro per-vertex painting



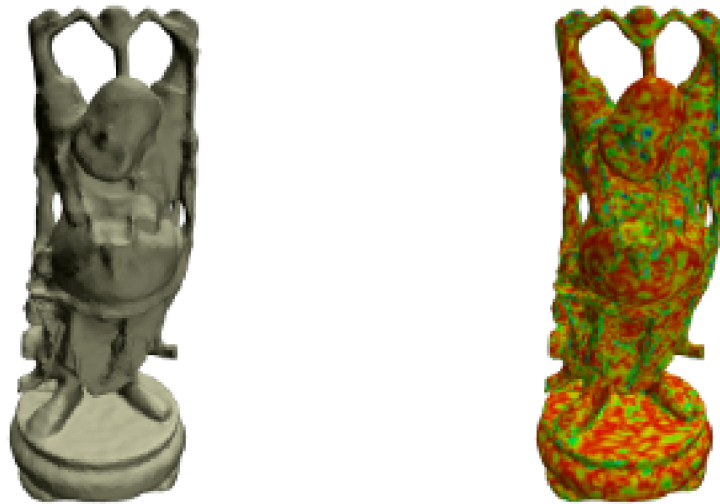
Obrázek A.2: Originální model



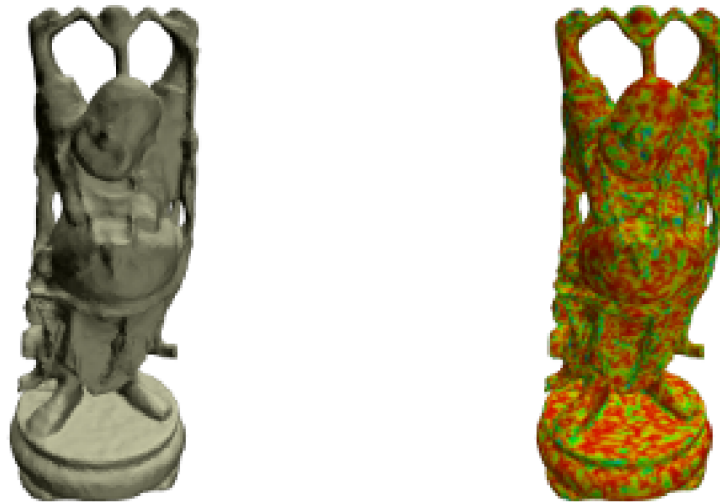
Obrázek A.3: Model po aplikaci šumu



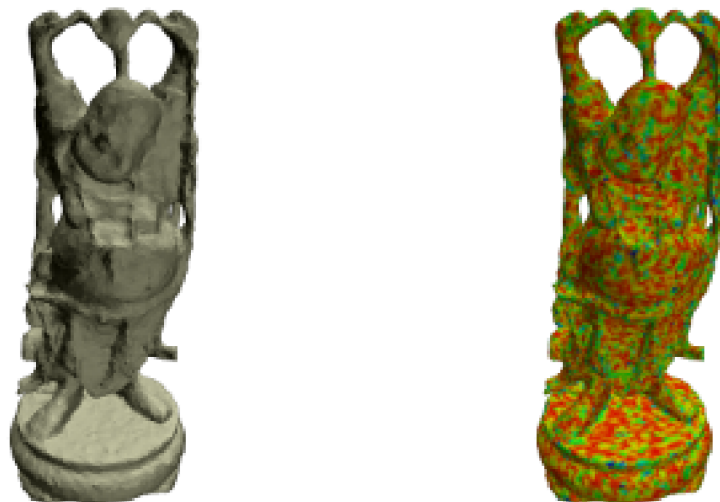
Obrázek A.4: Laplaceův algoritmus



Obrázek A.5: Algoritmus Laplace-HC



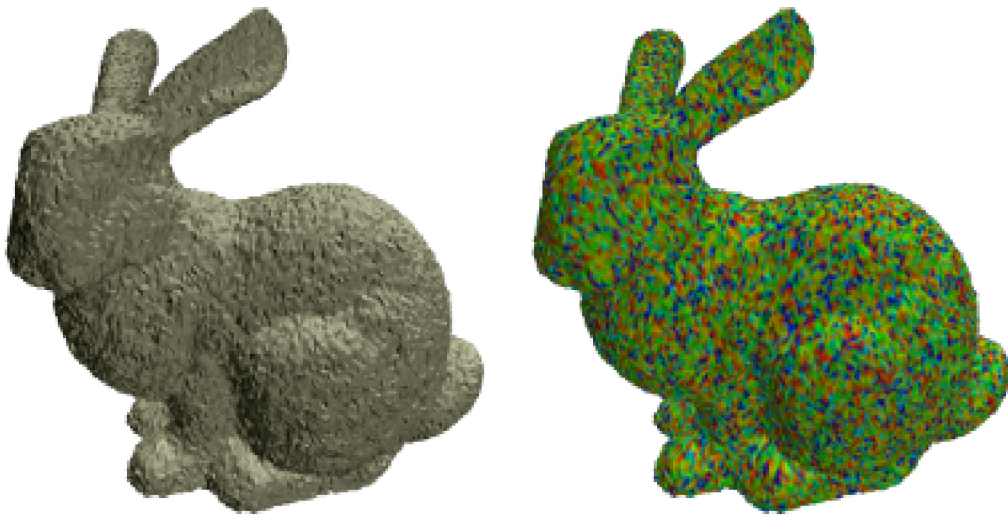
Obrázek A.6: Taubinův filtr dolní propusti



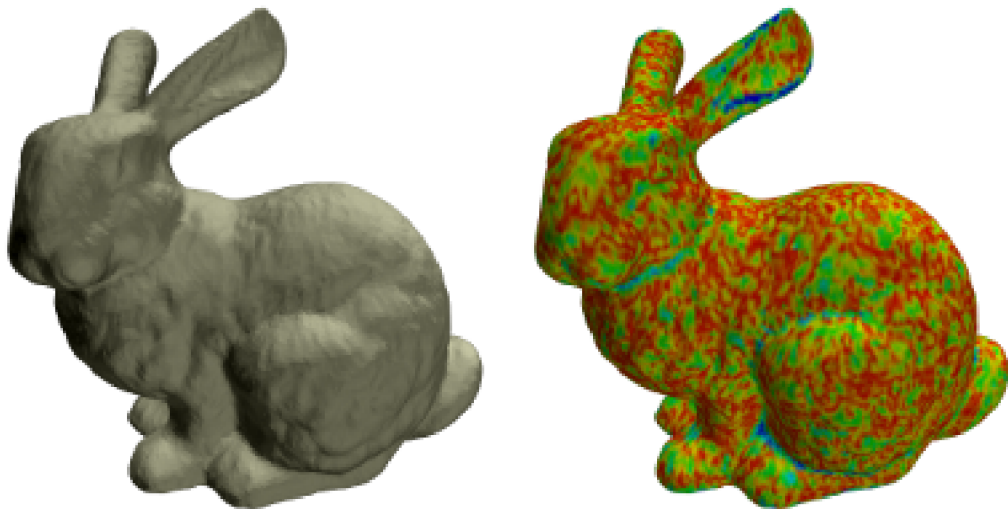
Obrázek A.7: Bilaterální filtr



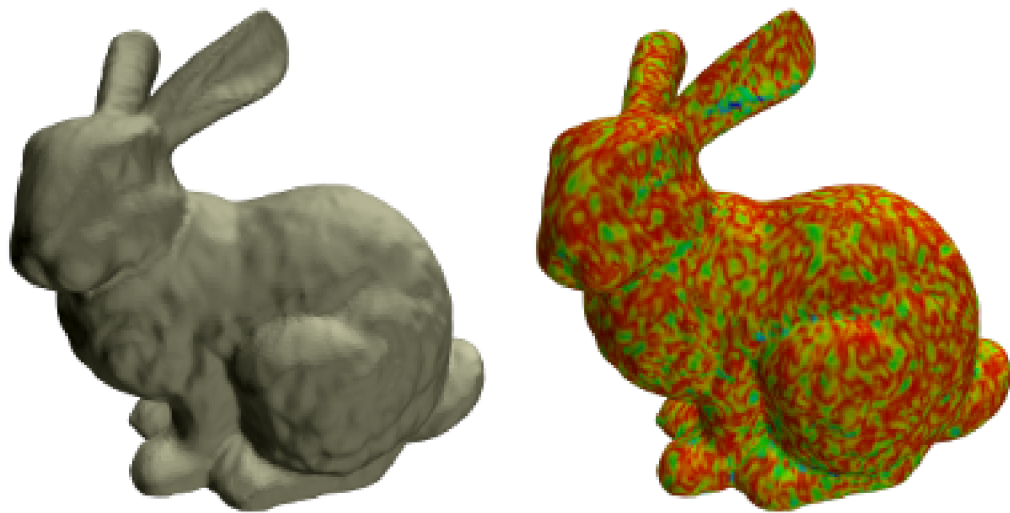
Obrázek A.8: Originální model



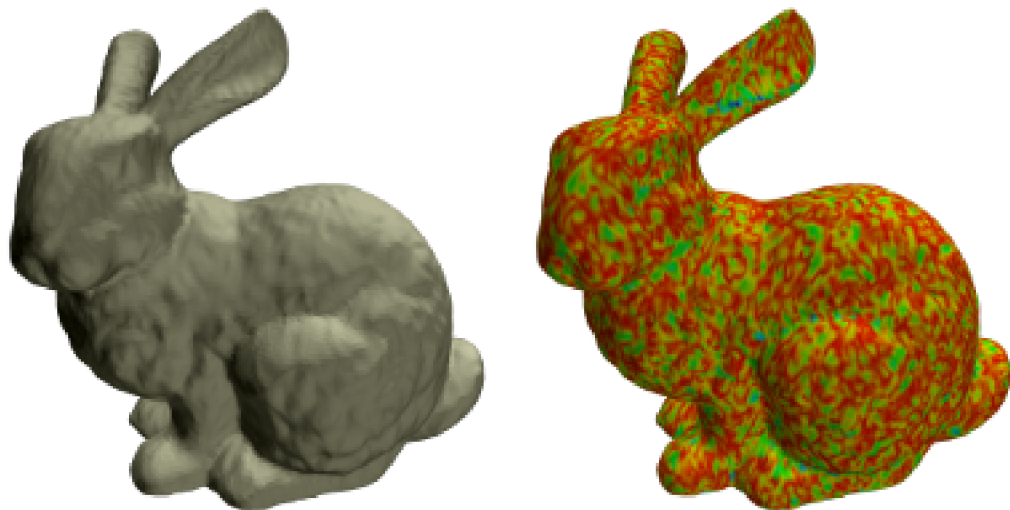
Obrázek A.9: Model po aplikaci šumu



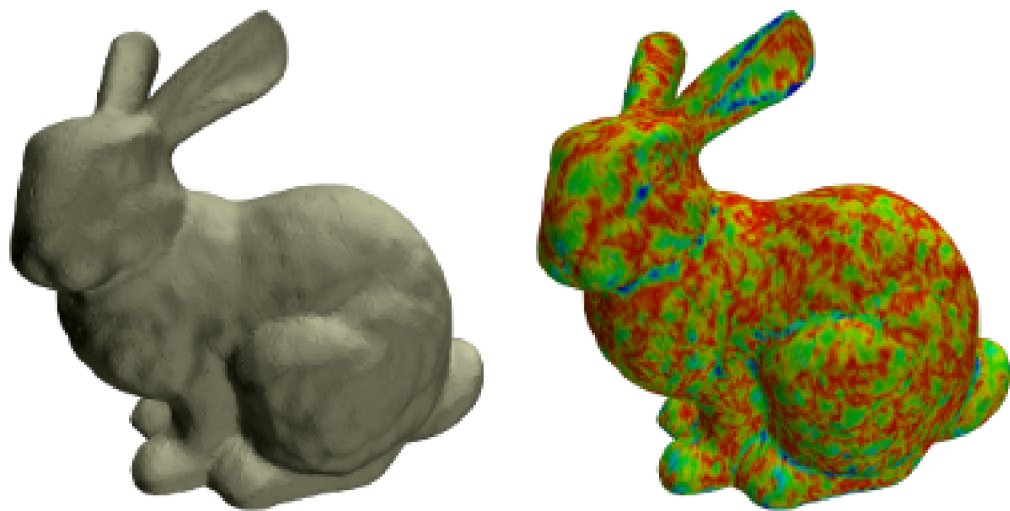
Obrázek A.10: Laplaceův algoritmus



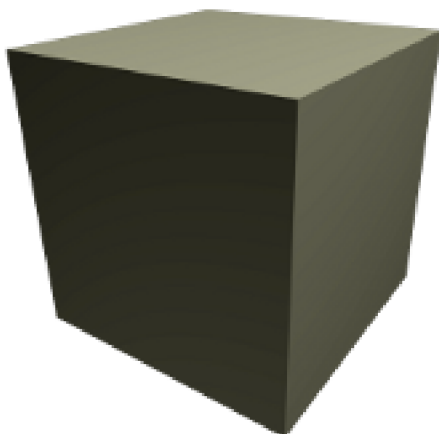
Obrázek A.11: Algoritmus Laplace-HC



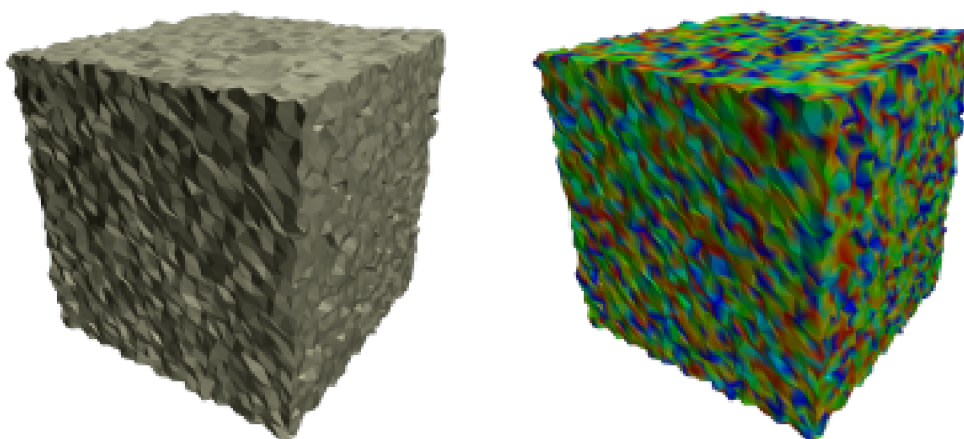
Obrázek A.12: Taubinův filtr dolní propusti



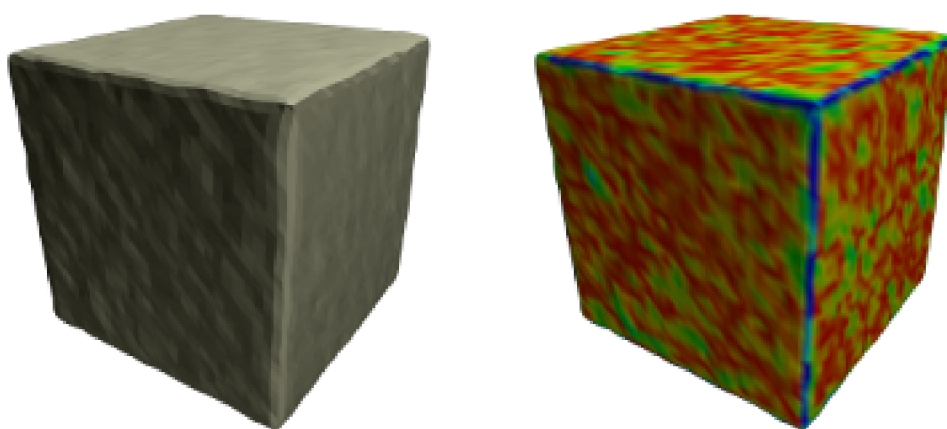
Obrázek A.13: Bilaterální filtr



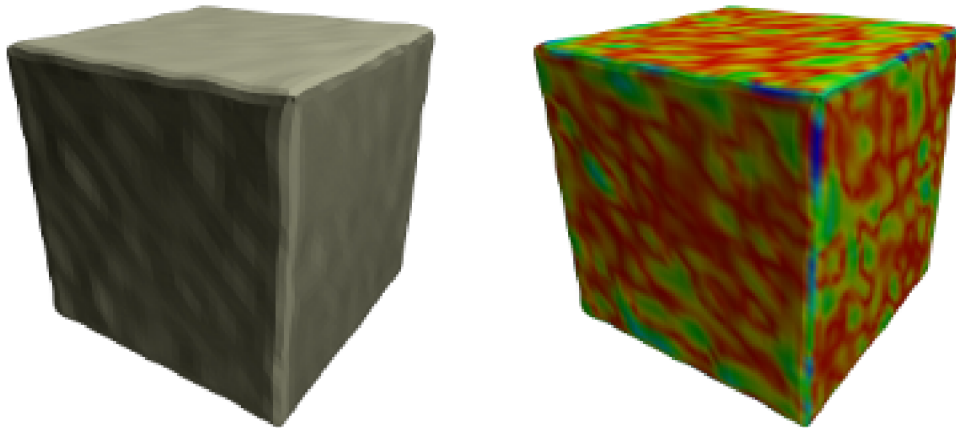
Obrázek A.14: Originální model



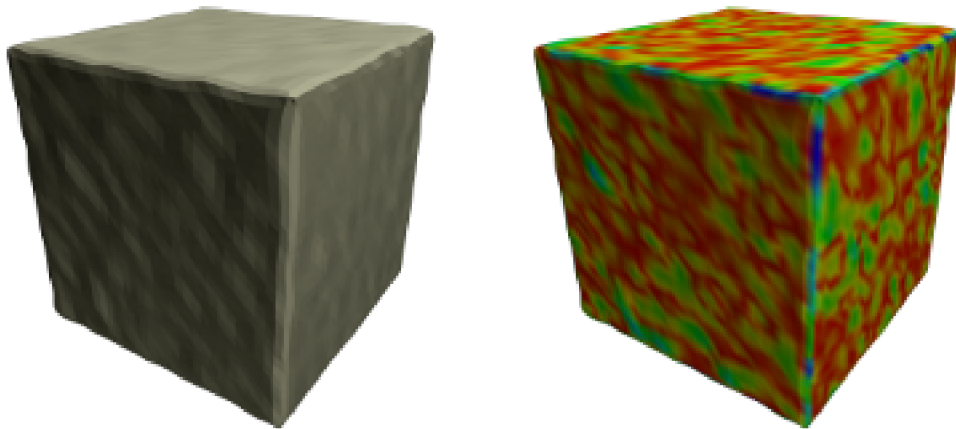
Obrázek A.15: Model po aplikaci šumu



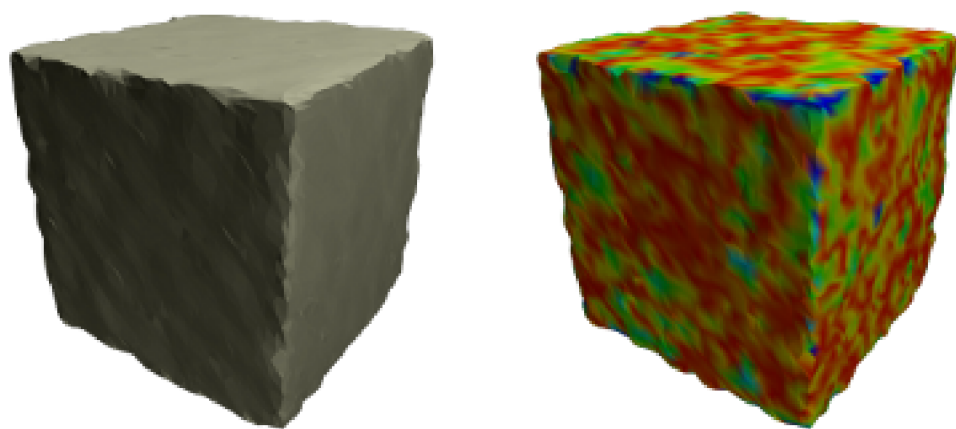
Obrázek A.16: Laplaceův algoritmus



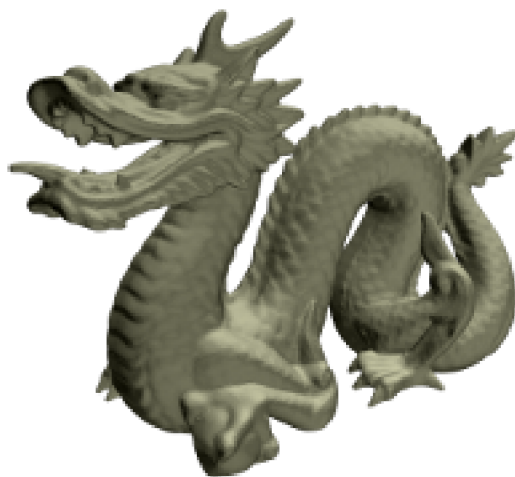
Obrázek A.17: Algoritmus Laplace-HC



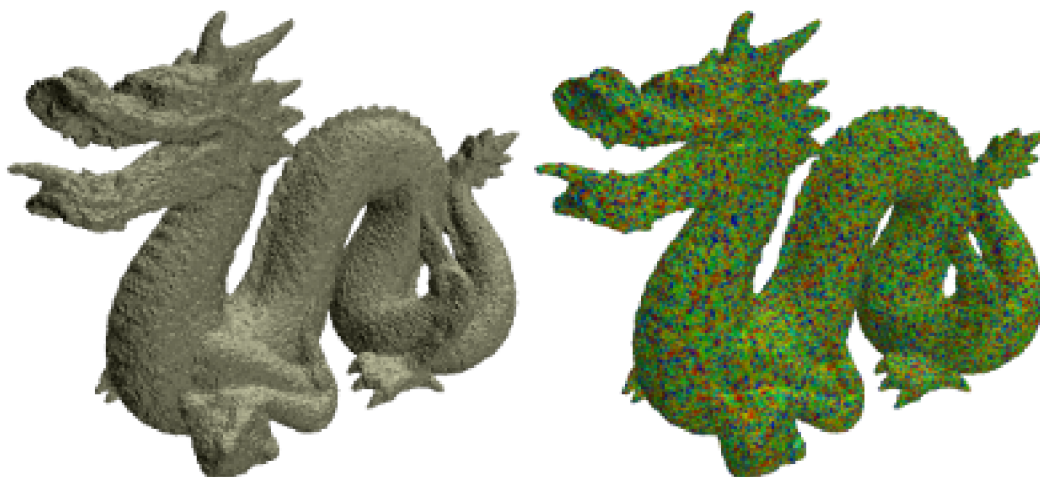
Obrázek A.18: Taubinův filtr dolní propusti



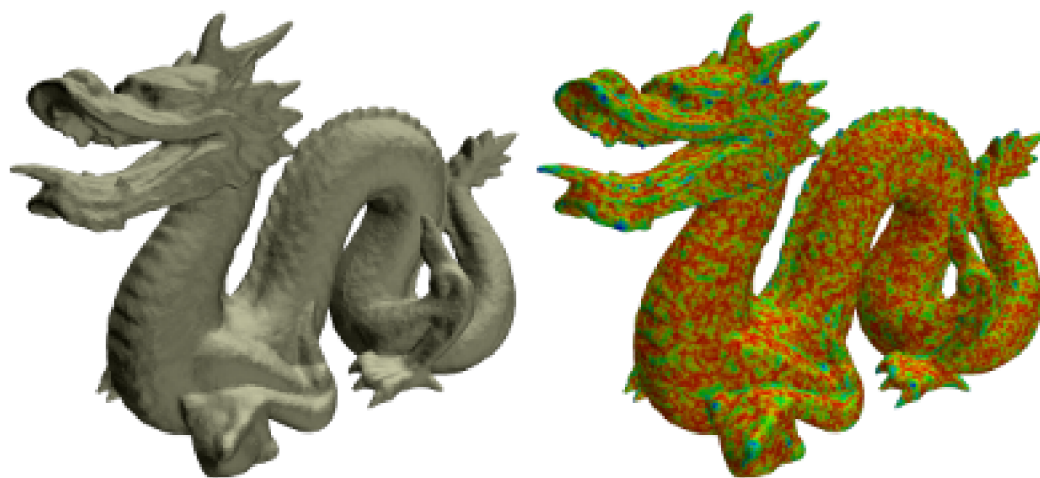
Obrázek A.19: Bilaterální filtr



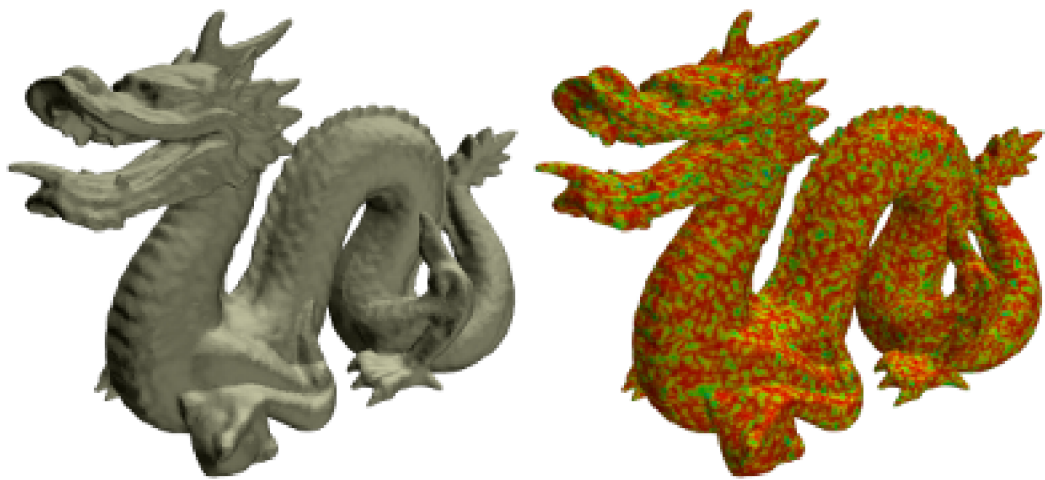
Obrázek A.20: Originální model



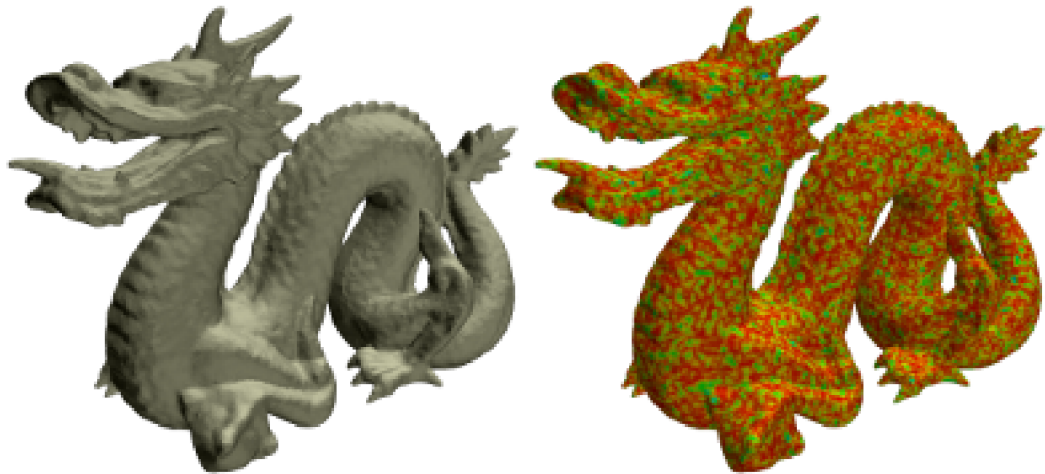
Obrázek A.21: Model po aplikaci šumu



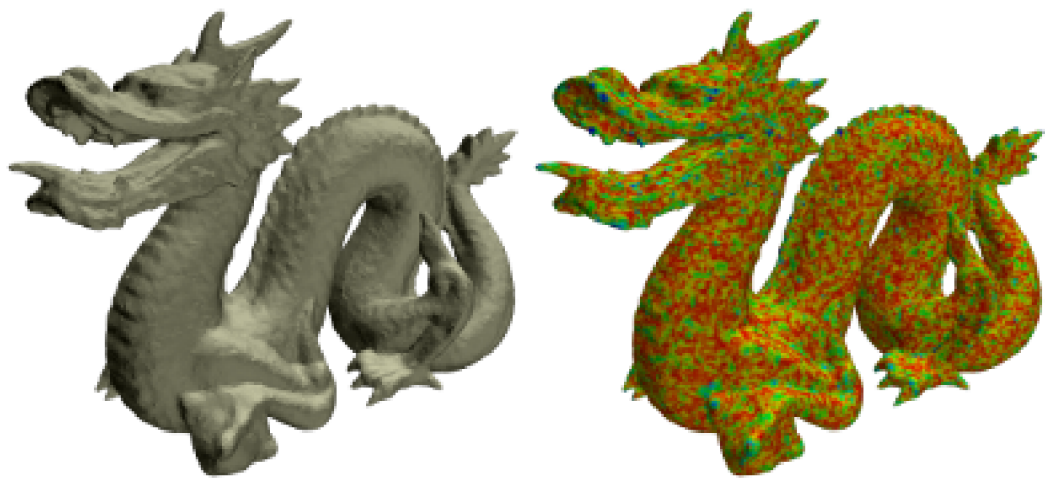
Obrázek A.22: Laplaceův algoritmus



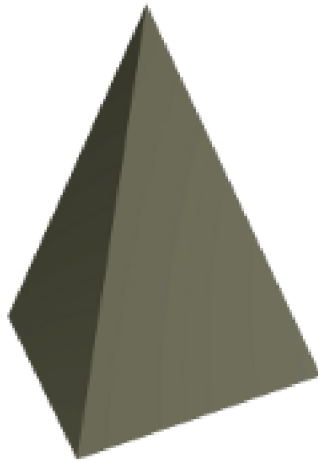
Obrázek A.23: Algoritmus Laplace-HC



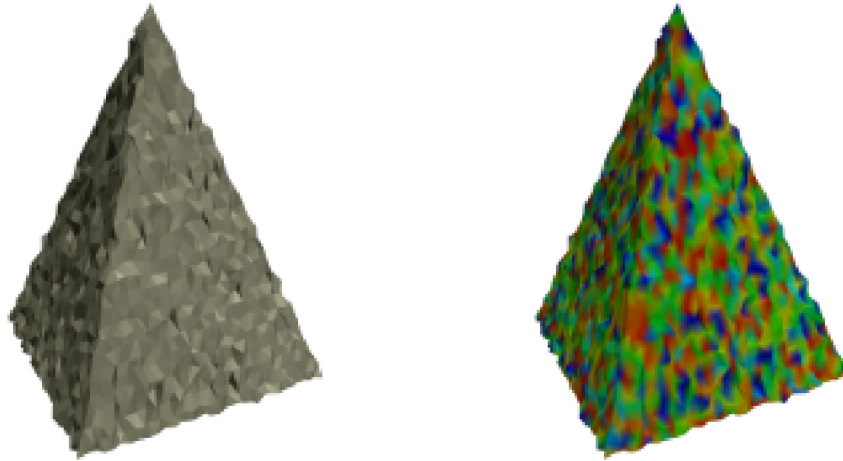
Obrázek A.24: Taubinův filtr dolní propusti



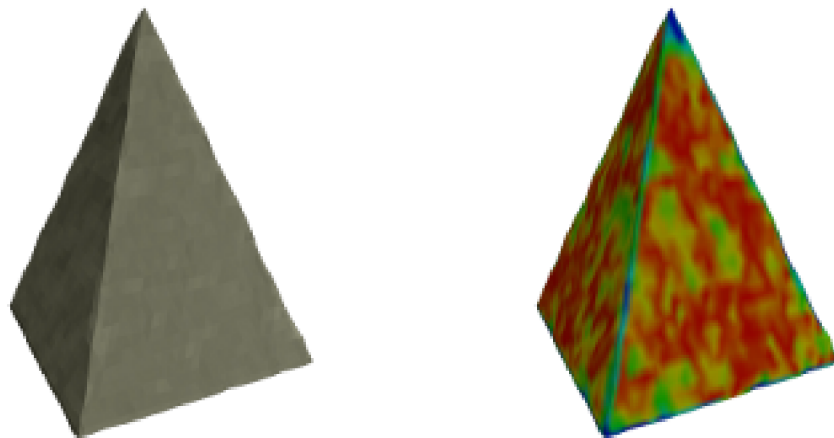
Obrázek A.25: Bilaterální filtr



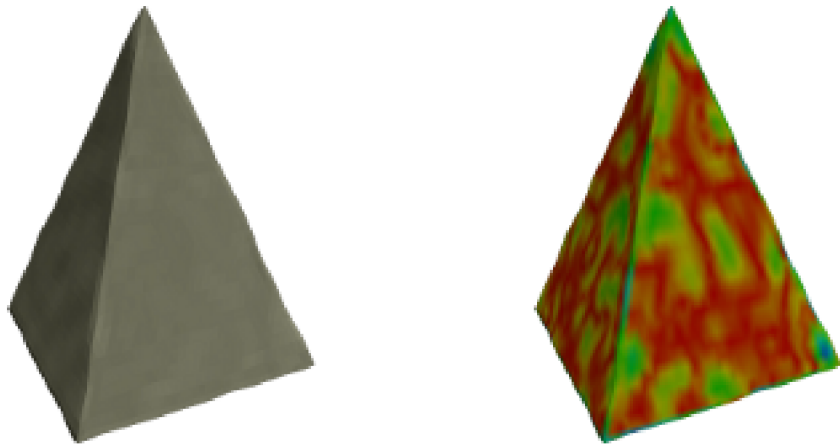
Obrázek A.26: Originální model



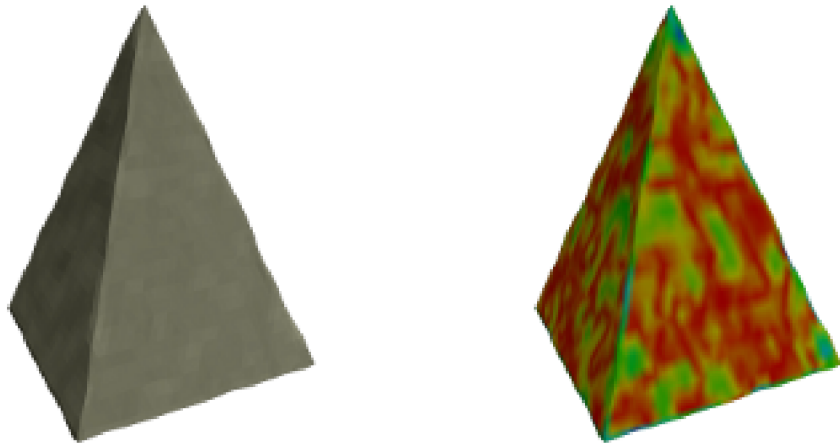
Obrázek A.27: Model po aplikaci šumu



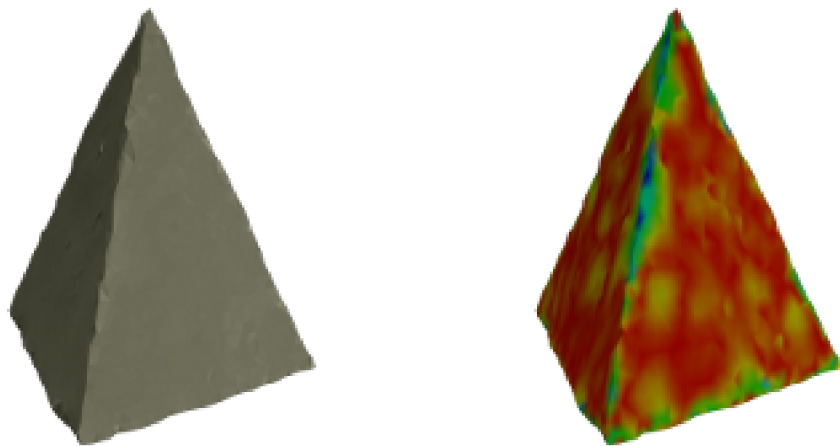
Obrázek A.28: Laplaceův algoritmus



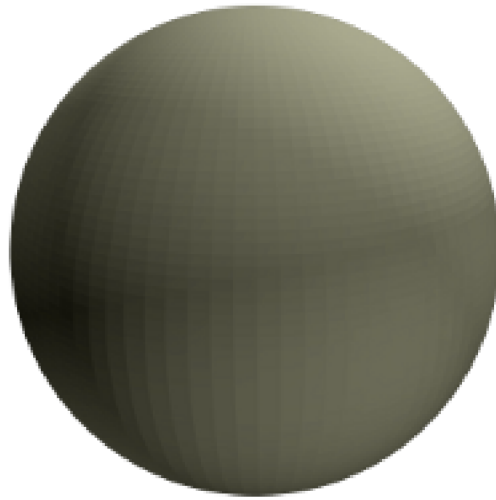
Obrázek A.29: Algoritmus Laplace-HC



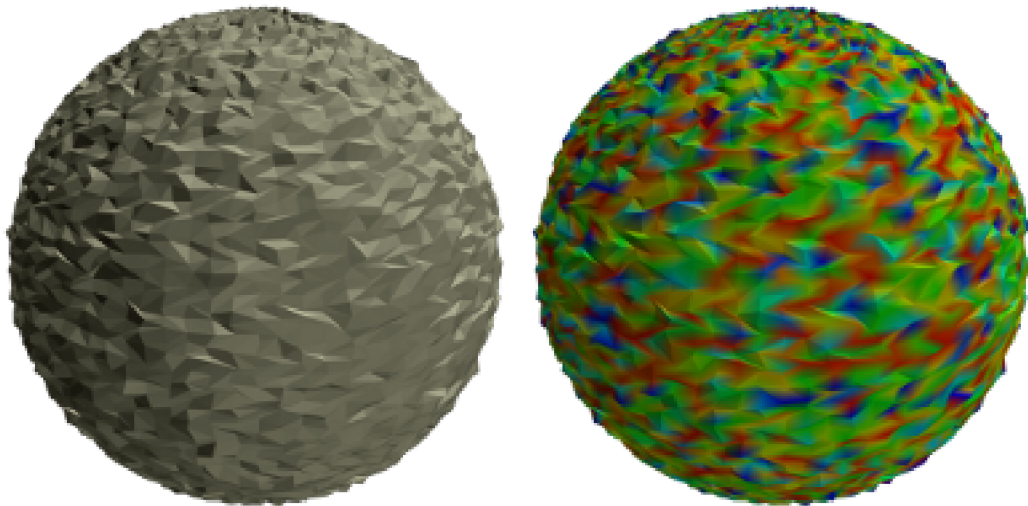
Obrázek A.30: Taubinův filtr dolní propusti



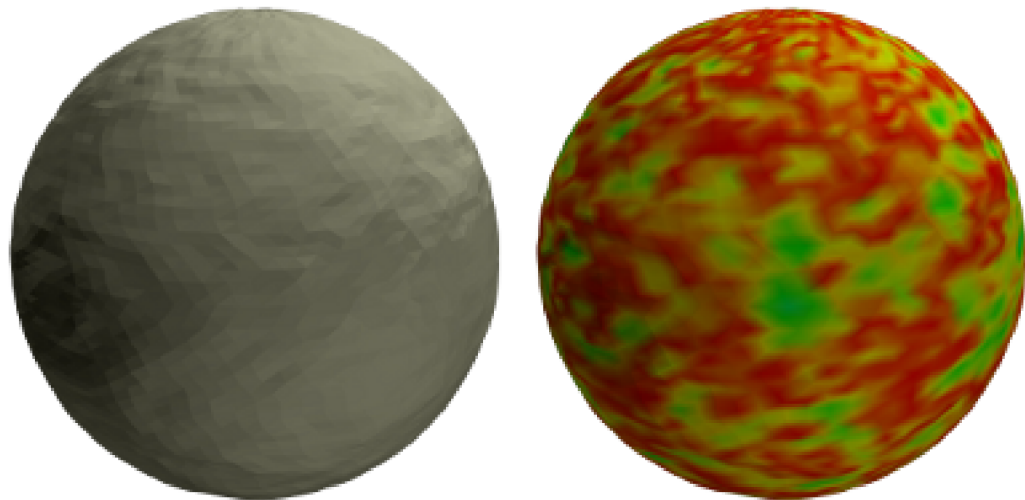
Obrázek A.31: Bilaterální filtr



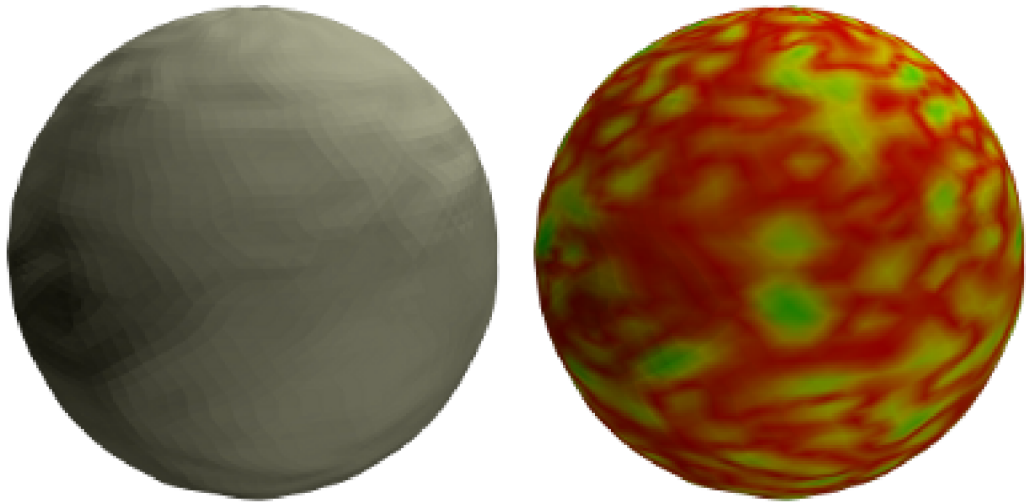
Obrázek A.32: Originální model



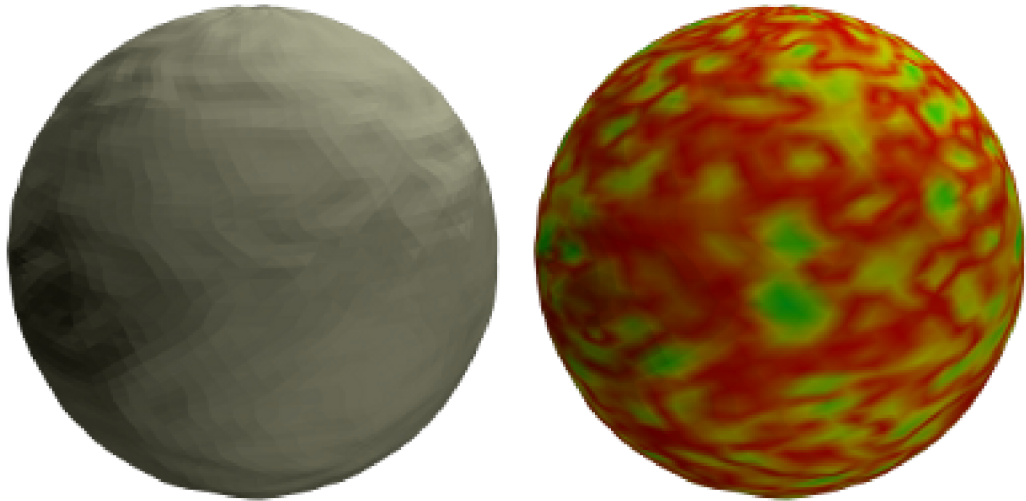
Obrázek A.33: Model po aplikaci šumu



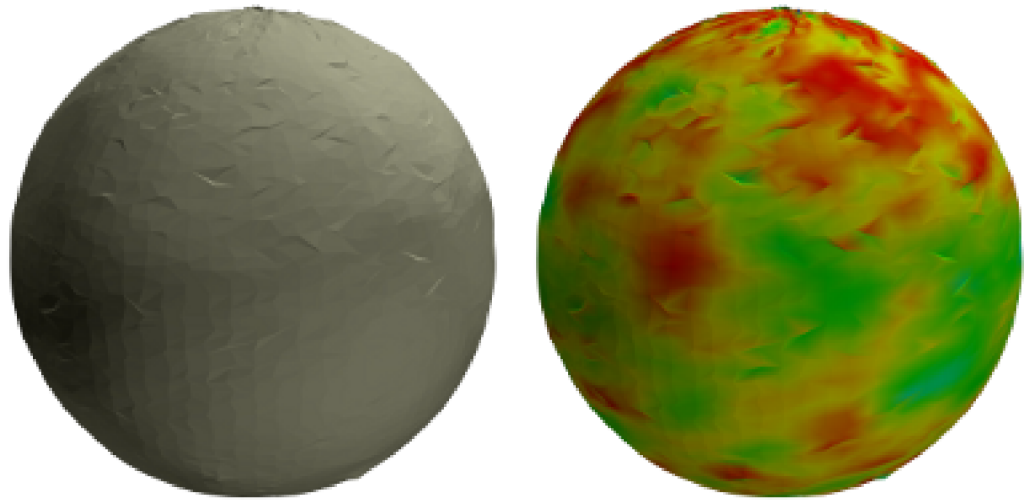
Obrázek A.34: Laplaceův algoritmus



Obrázek A.35: Algoritmus Laplace-HC



Obrázek A.36: Taubinův filtr dolní propusti



Obrázek A.37: Bilaterální filtr

Dodatek B

Vybrané statistické výsledky

B.1 Laplaceův algoritmus

Název	Iter.	Objem		Úhlová odchylka		Odchylka vzdáleností	
		Absolutní	[%] ²	Střední hodnota	σ ¹	Střední hodnota	σ ¹
buddha	20	–	–	0,187010	0,219191	0,002530	0,002061
buddha	22	–	–	0,187918	0,223240	0,002663	0,002178
buddha	25	–	–	0,191139	0,228943	0,002870	0,002352
bunny	20	2,540766	99,71 %	0,112483	0,077139	0,001922	0,001788
bunny	22	2,540039	99,68 %	0,108296	0,077133	0,001979	0,001871
bunny	25	2,538956	99,64 %	0,104573	0,078233	0,002084	0,002000
cone	20	4,151316	98,39 %	0,190068	0,284189	0,006038	0,009682
cone	22	4,146002	98,26 %	0,175093	0,267168	0,006032	0,010251
cone	25	4,138100	98,08 %	0,158119	0,249598	0,006088	0,011081
cube	20	5,795303	99,37 %	0,118641	0,107218	0,005120	0,006610
cube	22	5,791883	99,31 %	0,114350	0,112709	0,005132	0,007019
cube	25	5,786762	99,22 %	0,110204	0,120810	0,005205	0,007623
cylinder	20	6,731890	98,93 %	0,154482	0,183486	0,005930	0,008546
cylinder	22	6,724721	98,83 %	0,142431	0,164584	0,005956	0,009085
cylinder	25	6,714036	98,67 %	0,128861	0,147482	0,006071	0,009873
dragon	20	–	–	0,126482	0,169416	0,001899	0,001560
dragon	22	–	–	0,121662	0,169826	0,001962	0,001633
dragon	25	–	–	0,117425	0,170883	0,002077	0,001753
pyramid	20	1,226382	98,80 %	0,130398	0,132440	0,004979	0,006974
pyramid	22	1,224978	98,68 %	0,127663	0,139836	0,005051	0,007418
pyramid	25	1,222880	98,51 %	0,125723	0,150255	0,005204	0,008062
sphere	20	6,163341	99,41 %	0,122660	0,185913	0,003099	0,002327
sphere	22	6,159825	99,36 %	0,109475	0,166056	0,003081	0,002284
sphere	25	6,154567	99,27 %	0,093574	0,138268	0,003119	0,002250

Tabulka B.1: Laplaceův algoritmus

B.2 Algoritmus Laplace-HC

Název	Iter.	Objem		Úhlová odchylka		Odchylka vzdáleností	
		Absolutní	[%] ²	Střední hodnota	σ^1	Střední hodnota	σ^1
buddha	3	–	–	0,180448	0,221120	0,001741	0,001390
buddha	5	–	–	0,180860	0,228974	0,001812	0,001508
buddha	8	–	–	0,188317	0,236961	0,001982	0,001715
buddha	10	–	–	0,193556	0,240820	0,002099	0,001843
bunny	3	2,548051	99,99 %	0,116321	0,078643	0,001616	0,001379
bunny	5	2,547947	99,99 %	0,104315	0,076994	0,001539	0,001416
bunny	8	2,547828	99,98 %	0,099738	0,079186	0,001533	0,001520
bunny	10	2,547758	99,98 %	0,099305	0,080996	0,001554	0,001591
cone	3	4,204767	99,66 %	0,215521	0,326531	0,005931	0,006131
cone	5	4,203764	99,63 %	0,180904	0,290434	0,005485	0,006800
cone	8	4,202594	99,60 %	0,153463	0,252883	0,005182	0,007650
cone	10	4,201920	99,59 %	0,141954	0,234683	0,005083	0,008122
cube	3	5,829735	99,96 %	0,123811	0,097306	0,004948	0,004209
cube	5	5,829713	99,96 %	0,111366	0,110792	0,004705	0,004536
cube	8	5,829672	99,96 %	0,104926	0,124893	0,004589	0,005072
cube	10	5,829638	99,96 %	0,103719	0,131440	0,004572	0,005396
cylinder	3	6,804389	100,00 %	0,174377	0,222101	0,005849	0,005436
cylinder	5	6,803366	99,98 %	0,143184	0,178021	0,005501	0,005995
cylinder	8	6,801997	99,96 %	0,122387	0,147703	0,005305	0,006766
cylinder	10	6,801154	99,95 %	0,115544	0,140711	0,005261	0,007208
dragon	3	–	–	0,131676	0,177291	0,001530	0,001176
dragon	5	–	–	0,116083	0,174778	0,001425	0,001117
dragon	8	–	–	0,109733	0,174427	0,001401	0,001139
dragon	10	–	–	0,109040	0,175059	0,001419	0,001181
pyramid	3	1,240546	99,94 %	0,132246	0,120467	0,004437	0,004176
pyramid	5	1,240464	99,93 %	0,123426	0,139216	0,004279	0,004683
pyramid	8	1,240395	99,92 %	0,120090	0,157024	0,004230	0,005349
pyramid	10	1,240360	99,92 %	0,120630	0,164863	0,004253	0,005720
sphere	3	6,198450	99,98 %	0,140463	0,204644	0,003213	0,002461
sphere	5	6,198242	99,97 %	0,108479	0,161842	0,002801	0,002147
sphere	8	6,198031	99,97 %	0,084519	0,118774	0,002472	0,001898
sphere	10	6,197918	99,97 %	0,075369	0,104000	0,002328	0,001793

Tabulka B.2: Algoritmus Laplace-HC

B.3 Taubinův filtr dolní propusti

Název	Iter.	Objem		Úhlová odchylka		Odchylka vzdáleností	
		Absolutní	[%] ²	Střední hodnota	σ^1	Střední hodnota	σ^1
buddha	5	–	–	0,182051	0,212946	0,001861	0,001485
buddha	9	–	–	0,178227	0,220913	0,001891	0,001554
buddha	13	–	–	0,180207	0,227204	0,001950	0,001636
buddha	15	–	–	0,181753	0,230088	0,001983	0,001678
bunny	5	2,545740	99,90 %	0,123224	0,079471	0,001672	0,001423
bunny	9	2,545763	99,90 %	0,108662	0,076192	0,001593	0,001444
bunny	13	2,545801	99,90 %	0,103060	0,076337	0,001567	0,001484
bunny	15	2,545823	99,91 %	0,101546	0,076765	0,001563	0,001505
cone	5	4,187791	99,25 %	0,228044	0,331982	0,006017	0,006780
cone	9	4,187875	99,26 %	0,193013	0,300683	0,005612	0,007270
cone	13	4,188054	99,26 %	0,175195	0,283524	0,005391	0,007673
cone	15	4,188165	99,26 %	0,168327	0,273592	0,005314	0,007847
cube	5	5,818706	99,77 %	0,128781	0,093251	0,005008	0,004623
cube	9	5,819129	99,78 %	0,115453	0,103522	0,004743	0,004887
cube	13	5,819548	99,79 %	0,109492	0,112361	0,004618	0,005136
cube	15	5,819756	99,79 %	0,107648	0,115959	0,004579	0,005249
cylinder	5	6,781206	99,66 %	0,187602	0,237401	0,005874	0,005885
cylinder	9	6,781599	99,66 %	0,155090	0,193536	0,005515	0,006333
cylinder	13	6,782013	99,67 %	0,138792	0,170631	0,005335	0,006713
cylinder	15	6,782228	99,67 %	0,133245	0,163047	0,005277	0,006879
dragon	5	–	–	0,139518	0,173882	0,001609	0,001238
dragon	9	–	–	0,122162	0,172974	0,001517	0,001188
dragon	13	–	–	0,115045	0,173387	0,001480	0,001181
dragon	15	–	–	0,113030	0,173596	0,001472	0,001184
pyramid	5	1,236024	99,57 %	0,135132	0,113481	0,004548	0,004700
pyramid	9	1,236150	99,58 %	0,125220	0,129258	0,004351	0,005089
pyramid	13	1,236289	99,59 %	0,121500	0,141127	0,004268	0,005408
pyramid	15	1,236363	99,60 %	0,120537	0,145769	0,004247	0,005545
sphere	5	6,187302	99,80 %	0,156783	0,230778	0,003276	0,002512
sphere	9	6,187605	99,80 %	0,122764	0,182631	0,002937	0,002250
sphere	13	6,187942	99,81 %	0,105215	0,156391	0,002737	0,002098
sphere	15	6,188118	99,81 %	0,099078	0,147175	0,002662	0,002041

Tabulka B.3: Taubinův filtr dolní propusti

B.4 Bilaterální filtr

Název	Iter.	Objem		Úhlová odchylka		Odchylka vzdáleností	
		Absolutní	[%] ²	Střední hodnota	σ ¹	Střední hodnota	σ ¹
buddha	1	–	–	0,302703	0,375318	0,002201	0,001745
buddha	2	–	–	0,257284	0,361469	0,002288	0,001883
buddha	3	–	–	0,256339	0,362587	0,002691	0,002232
buddha	4	–	–	0,263901	0,366657	0,003163	0,002588
bunny	1	2,544293	99,85 %	0,198902	0,208556	0,001853	0,001454
bunny	2	2,540390	99,69 %	0,143315	0,162649	0,001881	0,001581
bunny	3	2,536520	99,54 %	0,132086	0,145697	0,002286	0,001962
bunny	4	2,532783	99,39 %	0,127483	0,134876	0,002763	0,002357
cone	1	4,201294	99,57 %	0,481726	0,621193	0,008105	0,006428
cone	2	4,191151	99,33 %	0,355730	0,591466	0,006059	0,005105
cone	4	4,167556	98,77 %	0,258499	0,519848	0,004924	0,004354
cone	6	4,144249	98,22 %	0,212749	0,458915	0,005289	0,004574
cube	1	5,824787	99,88 %	0,201318	0,193482	0,005901	0,004806
cube	2	5,819626	99,79 %	0,116981	0,159035	0,004211	0,004072
cube	4	5,806781	99,57 %	0,084573	0,142418	0,003659	0,005067
cube	6	5,789717	99,27 %	0,084163	0,150767	0,004150	0,006884
cylinder	1	6,795167	99,86 %	0,392080	0,525130	0,007708	0,006219
cylinder	2	6,780708	99,65 %	0,276626	0,491882	0,005610	0,004957
cylinder	4	6,742665	99,09 %	0,192654	0,431946	0,004614	0,004400
cylinder	6	6,702073	98,49 %	0,158242	0,373325	0,005436	0,004633
dragon	1	–	–	0,287694	0,388668	0,002134	0,001672
dragon	2	–	–	0,210329	0,362140	0,001891	0,001518
dragon	3	–	–	0,185602	0,345719	0,002022	0,001675
dragon	4	–	–	0,175999	0,334479	0,002279	0,001926
pyramid	1	1,238687	99,79 %	0,193505	0,196218	0,004953	0,004087
pyramid	2	1,236440	99,61 %	0,120734	0,174886	0,003637	0,003778
pyramid	4	1,230221	99,10 %	0,098502	0,165995	0,003509	0,005101
pyramid	6	1,222111	98,45 %	0,105939	0,183221	0,004340	0,006930
sphere	1	6,183967	99,74 %	0,303315	0,471568	0,004348	0,003613
sphere	2	6,163150	99,41 %	0,214274	0,449158	0,003427	0,002843
sphere	4	6,118920	98,70 %	0,153780	0,382039	0,004569	0,002774
sphere	6	6,074629	97,98 %	0,120031	0,323849	0,006551	0,003419

Tabulka B.4: Bilaterální filtr

¹Směrodatná odchylka (s_x)

²Objem vyhlazeného modelu vůči originálnímu modelu v procentech

Dodatek C

Obsah DVD

1. Programová dokumentace
2. Program metro
3. Knihovna MDSTk
4. Sada testovacích modelů
5. Sada zašuměných modelů
6. Sada vyhlazených modelů, grafů, histogramů a tabulek
7. Zdrojové kódy a BASH skripty