

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

**Image-driven simplifikace
trojúhelníkových sítí
s použitím percepční metriky**

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 9. května 2012

Zdeněk Malík

Poděkování

Tímto bych chtěl poděkovat především panu Ing. Liboru Vášovi, Ph. D. za cenné rady, které mi dal v souvislosti s touto prací i studiem. Také bych chtěl poděkovat celé grafické skupině na Katedře informatiky za jejich ochotu při řešení problémů, své rodině a přátelům za jejich podporu.

Abstract

Image-driven simplification of triangle meshes using perceptual metrics

This Thesis deals with the possibilities of image-driven simplification of triangle meshes using perceptual metrics. The Theoretical section describes the geometric and image-driven simplification methods that are used in the practical part. The goal of this work is to determine whether perceptual image metrics reduce the value of geometric perceptual metrics compared to non-perceptual metrics. In this Thesis there are implemented two geometric non-perceptual metrics, one non-perceptual image-driven metric and one perceptual image-driven metric. These metrics are examined in the final section.

Obsah

1	Úvod	1
2	Trojúhelníkové sítě	2
2.1	Struktura	2
2.2	Problémy	2
3	Simplifikace	4
3.1	Záměr simplifikace	4
3.2	Simplifikace geometrie	5
3.2.1	Odstranění vrcholu	5
3.2.2	Kontrakce hrany	5
3.2.3	Kontrakce páru vrcholů	7
3.2.4	Shlukování vrcholů	7
4	Simplifikační metody	9
4.1	Chybové metriky	10
4.1.1	Hausdorffova vzdálenost	10
4.1.2	Kvadrická chybová	10
4.1.3	MSDM	12
4.1.4	MSDM2	14
4.2	Cenové funkce	14
4.3	Obrazové metriky	15
4.3.1	Metody porovnání dvou obrazů	16
4.3.2	Jednoduchá obrazová metrika MSE	18
4.3.3	Metrika SSIM jako simplifikační metoda	19
4.3.4	Zrychlení výpočtů obrazových metrik	19
5	Implementace v prostředí MVE-2	21
5.1	Prostředí MVE-2	21
5.1.1	Moduly	21
5.1.2	Mapy	22

5.2	Implementace simplifikačního algoritmu	22
5.2.1	Třída Collapser	23
5.2.2	Prioritní fronta EdgeCollapsePQ	26
5.2.3	Podpůrné Třídy	26
5.3	Mechanismus pro implementaci metrik	26
5.3.1	Návrh obecného mechanismu	26
5.3.2	Bázová třída BaseSimplificator	27
5.4	Implementace rendereru	28
5.4.1	Třída Renderer	29
5.4.2	Třída Mesh	30
5.4.3	Třída Camera	30
5.4.4	Řešené problémy	31
5.5	Implementace simplifikačních metrik	31
5.5.1	Třída Simplificator	31
5.5.2	Implementace kvadratické metriky	33
5.6	Implementace obrazových metrik	34
5.6.1	Třída ImageMetrics	35
5.6.2	Třída Utils	35
5.6.3	Implementace SSIM metriky	35
5.6.4	Cenové funkce obrazových metrik	36
5.7	Modul SimplificatorModule	37
5.8	Modul MeshComparerModule	37
5.8.1	Třída MeshComparer	38
6	Simplifikace modelů	39
6.1	Příprava modelů	39
6.1.1	Tvorba modelů v Blenderu	40
6.2	Simplifikační mapa	41
6.3	Program mapy	42
7	Měření a dosažené výsledky	43
7.1	Porovnání modelů	44
7.1.1	Hausdorffova vzdálenost	44
7.1.2	Vzdálenost MSDM2	44
7.1.3	Měření indexu RMS a SSIM	45
7.1.4	Porovnání modelů uživateli	46
7.2	Výsledky měření	47
8	Závěr	53

1 Úvod

V dnešní době se vizualizace trojrozměrných trojúhelníkových sítí (modelů) stává běžnou v různých oblastech našeho života. Ať už se jedná o průmyslové vizualizace nových produktů nebo lékařské vizualizace CT snímků, aj. V mnoha případech však stále nedokážeme zobrazovat dostatečně rychle některé vysoce kvalitní modely. V takovém případě je vhodné použít metody, které trojúhelníkovou síť zjednoduší. Tyto metody se také nazývají simplifikační a tato práce se některými z nich zabývá.

Simplifikačních metod je celá řada. Od rychlých, které zjednodušují trojúhelníkové sítě jednoduchými geometrickými metodami, až po složité obrazové percepční metriky, které jsou časově náročné na výpočet. Percepčními metrikami se rozumí takové metody, které respektují lidské vnímání (velmi často vizuální).

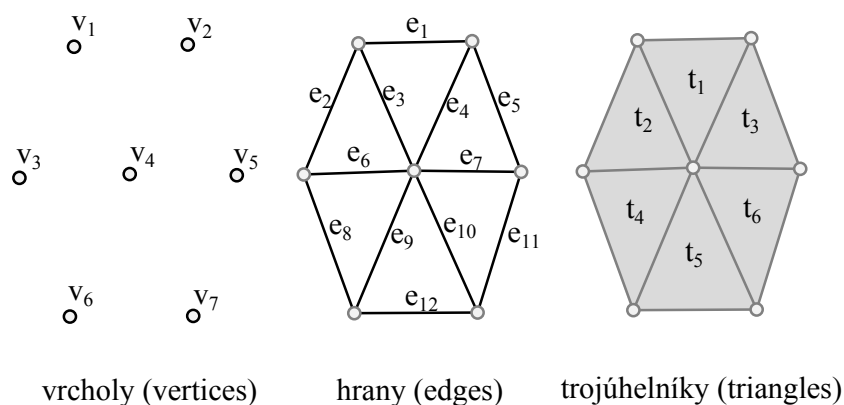
Práce porovnává percepční a nepercepční simplifikační obrazové metriky vzhledem k hodnotě percepční geometrické metriky. Toto porovnání i s dalšími metodami simplifikace bude provedeno v kapitole 7, kde jsou vyhodnoceny změřené výsledky.

Cílem práce je potvrzení či vyvrácení tvrzení, že pokud se pro simplifikaci obrazovými metrikami použije percepční obrazová metrika SSIM v roli cenové funkce, pak dojde k poklesu percepční geometrické metriky MSDM2 vyhodnocené na výsledku. Poklesem se rozumí pokles vzhledem k případu použití nepercepční obrazové metriky MSE za předpokladu stejného cílového počtu primitiv.

2 Trojúhelníkové sítě

2.1 Struktura

V počítačové grafice jsou trojúhelníkové sítě základním stavebním kamenem reprezentace trojrozměrných modelů. Tyto sítě jsou tvořeny body (vrcholy), hranami a trojúhelníky, viz obrázek 2.1. Využitím trojúhelníkové struktury lze docílit vysoce kvalitních trojrozměrných modelů, které jsou zapotřebí v různých odvětvích vizualizace, ať se jedná o průmyslové vizualizace, lékařské vizualizace nebo zábavní průmysl. Obzvláště v průmyslu a lékařství se využívají modely, které dosahují desítky milionů bodů a trojúhelníků. Takové modely jsou velmi často výsledkem různých skenovacích technik, například CT, magnetická rezonance a dalších. Naopak v zábavním průmyslu pro speciální efekty nebo pro počítačové hry se využívá modelů, které jsou modelovány ručně za použití speciálních modelovacích softwarů. Obvykle tyto modely nejsou tak složité jako ve výše uvedených případech.



Obrázek 2.1: Struktura trojúhelníkové sítě

2.2 Problémy

Při skenování či ručním modelování trojúhelníkové sítě může docházet k nechtěným defektům na trojúhelníkové síti. Tyto defekty mohou v některých případech vést k problematické vizualizaci nebo mohou ovlivnit samotnou strukturu a práci s ní. Časté problémy jsou:

- protínání jednotlivých trojúhelníků,
- otočená orientace trojúhelníků - v počítačové grafice je orientace trojúhelníku definována pořadím vrcholů proti směru hodinových ručiček. Jestliže dojde k záměně pořadí dvou vrcholů, dojde k otočení orientace daného trojúhelníku.
- hrana sdílená více než dvěma trojúhelníky,
- překrývající se trojúhelníky (dva trojúhelníky na stejných bodech).

V praxi se takovým případům snažíme co nejvíce vyhýbat, případně tyto problémy detekovat a opravit. Zvláště v oblasti simplifikace k takovým defektům dochází velmi často a je nutno je řešit, protože dokáží velmi ovlivnit výsledky simplifikačních metod.

3 Simplifikace

3.1 Záměr simplifikace

Reprezentace vysoce detailních modelů je velmi náročná na výpočetní zdroje a v takové míře komplexnosti obvykle nelze provádět vizualizaci v reálném čase. Zde přichází na řadu tzv. simplifikace modelu, která vytvoří z vysoce detailního modelu jeho aproximaci s mnohem menším počtem trojúhelníků. Zatímco aproximovaný model není přesný jako originál, ztráta určité míry detailnosti může být akceptovatelná. Díky menšímu počtu trojúhelníků je kladena menší zátěž na výpočetní zdroje a umožní tak rychlejší vizualizaci daného modelu.

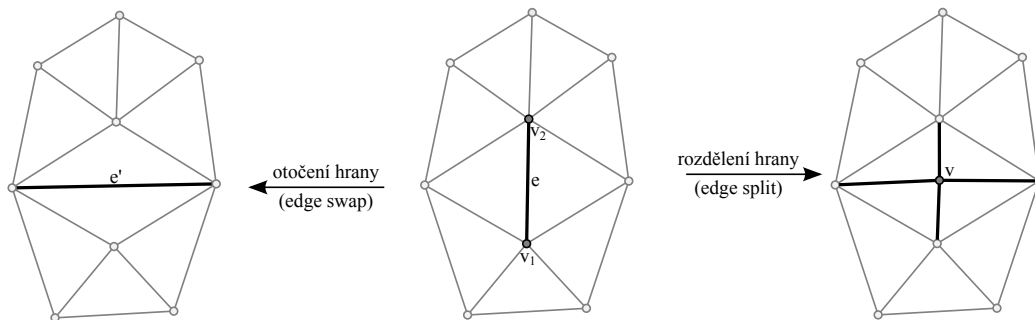
Představme si, že máme trojrozměrný model, který je tvořen převážně rovinnými plochami. Pak více trojúhelníků v jedné rovině lze často bez větších ztrát na kvalitě modelu sloučit a dosáhneme tak znatelného nárůstu rychlosti zpracování modelu a snížení velikosti dat potřebných k jeho uložení do paměti. Máme-li naopak model, kde je jen minimum sousedících trojúhelníků v jedné rovině, dochází při simplifikaci k viditelnému snížení kvality modelu.

Hlavním cílem simplifikace je najít takový algoritmus, který dokáže provést optimální aproximaci modelu. Všeobecné optimální řešení neexistuje, protože záleží na dané aplikaci. Někdy může být hlavní prioritou kvalita modelu, ale za cenu dlouhého výpočtu simplifikace, nebo naopak v určitých aplikacích může být důležitější čas, za který dosáhneme rozumného výsledku.

Simplifikaci lze rozdělit na tři spolu související problémy. Jedním je výběr simplifikačního kroku (kontrakce hrany, decimace vrcholu), druhým je ohodnocení jednotlivých primitiv, které jsou postupně odstraňovány (tzv. *cenové funkce*), a třetím problémem je měření chyby simplifikace (nebo-li *chybové metriky*). Cenové funkce a chybové metriky jsou spolu však tak úzce svázané, že budou dále popsány v kapitole 4 společně s metodami, které chybové metriky i cenové funkce využívají.

3.2 Simplifikace geometrie

Aby došlo ke zjednodušení trojúhelníkové sítě, je nutné provádět odstraňování vrcholů, hran a trojúhelníků. Na obrázku 3.1 jsou znázorněné základní operace, které lze provádět na trojúhelníkové síti. Pro simplifikaci geometrie se otočení hrany (edge swap) a rozdělení hrany (edge split) nepoužívá. Na obrázku 3.2 jsou zobrazeny lokální operace používané pro simplifikaci viz [7].



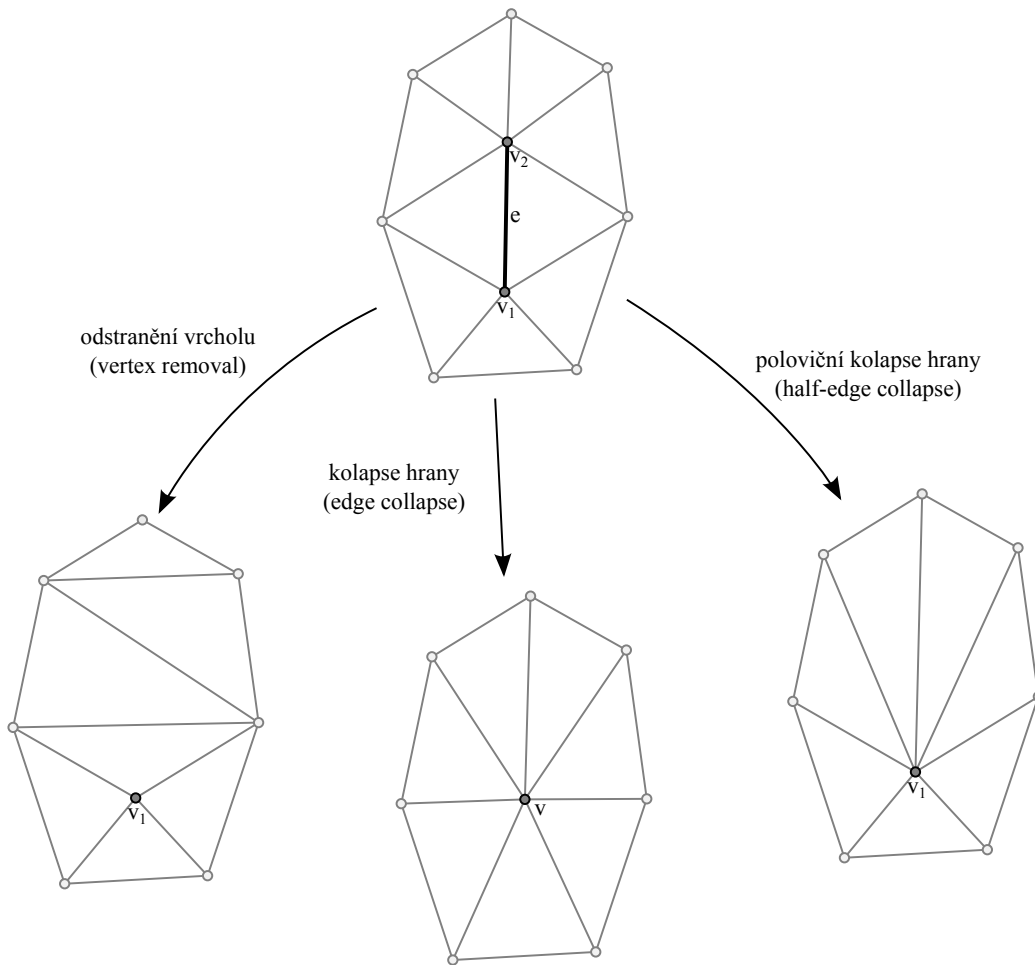
Obrázek 3.1: Operace konektivity trojúhelníkové sítě

3.2.1 Odstranění vrcholu

Jednou z metod je odstranění vybraného vrcholu sítě a okolních trojúhelníků. Tím vznikne díra, která musí být znovu rekonstruována (triangulace). Tento problém není úplně triviální a podrobně se jím zabývá například Schroeder a kol. v odborném článku [9]. Metoda odstranění vrcholu je velmi výhodná pro simplifikaci téměř rovinných modelů, protože lze odstranit více vrcholů v jeden okamžik a triangulovat až v následujícím kroku. Naopak pro křivé plochy je tímto způsobem dosahováno horšího výsledku než v následujících metodách.

3.2.2 Kontrakce hrany

Jedná se o velmi oblíbenou metodu, která se snadno implementuje a lze s ní docílit vynikajících výsledků. Metoda provede sloučení koncových vrcholů vybrané hrany do jednoho vrcholu, který zaujímá místo jednoho z původních vrcholů (*half-edge collapse*), nebo je nový vrchol umístěn do libovolného jiného místa. Sloučením vrcholů dojde k odstranění vybrané hrany a přilehlých

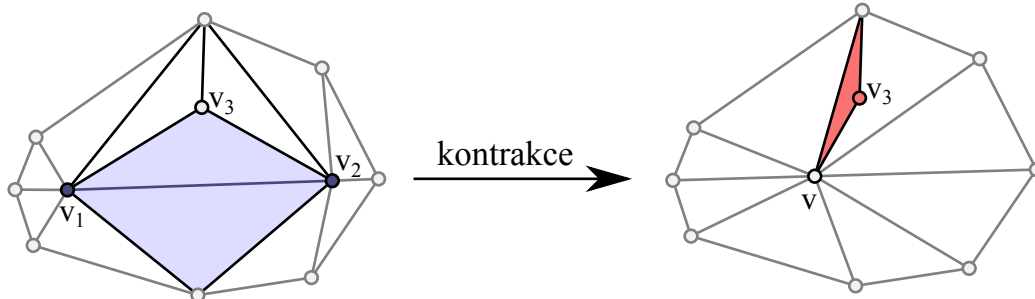


Obrázek 3.2: Lokální operace na trojúhelníkové síti

trojúhelníků. V případě ucelené trojúhelníkové sítě dojde k odstranění právě dvou trojúhelníků, v neucelené pak může na okrajích sítě dojít ke kontrakci hrany, která má pouze jeden přilehlý trojúhelník.

Při kontrakci hrany může vzniknout situace, kdy dojde k odtržení dvou trojúhelníků z trojúhelníkové sítě a vytvoří chybnou konektivitu, viz obrázek 3.3. Řešením je odstranění těchto přebytečných trojúhelníků a jejich vrcholu. Pokud tedy dojde ke kontrakci hrany trojúhelníku, jejíž protilehlý vrchol má právě tři přilehlé trojúhelníky, je tento vrchol i s jeho přilehlými trojúhelníky odstraněn rovněž. V takovém případě můžeme při kontrakci jedné hrany odstranit čtyři nebo šest trojúhelníků, je-li shodná situace i na druhém trojúhelníku dané hrany. Dále může nastat situace, kdy dojde ke kontrakci hrany v úzkém místě trojúhelníkové sítě. Pak zde vznikne hrana incidentní se čtyřmi

trojúhelníky. Řešením takové situace je obvykle zamítnutí této kontrakce.



Obrázek 3.3: Vznik chybné konektivity kontrakcí hrany ve speciálním případě

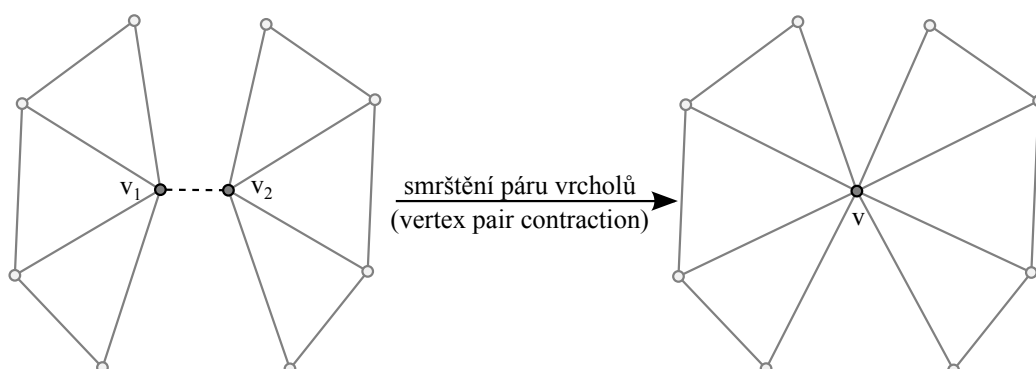
Síla této metody je v možnostech umístění nového vrcholu, vzniklého po kontrakci. Existuje široká škála metod, které řeší umístění vrcholu tak, aby výsledek vyhovoval dané aplikaci. Jednoduchou metodou může být umístění vrcholu v polovině kontrahované hrany, nebo v libovolném bodě na této hraně. Složitější metody dokáží vybrat například takovou pozici vrcholu, aby zůstal zachován objem modelu (viz práce Lindstroma a kol. [7]), tvar a některé zajišťují, že nedojde k překrytí dvou trojúhelníků. Umístění vrcholu je velmi často součástí simplifikačních metod, které jsou probrány v kapitole 4 na straně 9. Touto metodou se podrobně zabývá práce [8]. Metoda kontrakce hrany je také použita v praktické části této práce.

3.2.3 Kontrakce páru vrcholů

Metoda kontrakce páru vrcholů se od kontrakce hrany liší převážně tím, že spojí i dva body, které spolu nejsou spojeny hranou, viz obrázek 3.4. Tato metoda dokáže velmi dobře pracovat s neucelenými modely i s modely rozbitými na komponenty. U této metody může být problematický výběr správného páru vrcholů. I zde existuje nepřehledné množství metod, které opět závisí na aplikaci. Výběr páru může být například takový, že simplifikaci chceme pouze pospojovat jednotlivé komponenty modelu a neprovádět simplifikaci na ucelených částech, viz [4].

3.2.4 Shlukování vrcholů

V této metodě se vytvoří ohraničující kvádr (tzv. *bounding box*), který je rozdělen do mřížky. V každé buňce jsou pak vrcholy smršťovány do jednoho



Obrázek 3.4: Kontrakce páru vrcholů nespojených komponent

vrcholu a všechny trojúhelníky náležící těmto vrcholům jsou smazány nebo opraveny tak, aby vyhovovaly novému uspořádání. V závislosti na velikosti mřížky může být tato simplifikace velice rychlá a dokáže drasticky zjednodušit model. Slabým místem této metody je, že nedokáže simplifikovat velké ploché části rozprostírající se přes několik buněk mřížky. Metodu je možné vylepšit použitím datové struktury *octree*, která adaptivně rozdělí buňky na menší části, čímž lze docílit lepších výsledků.

4 Simplifikační metody

V procesu simplifikace trojúhelníkové sítě obvykle vyžadujeme, aby docházelo ke zjednodušování sítě nejprve v místech, které sít' poškodí co nejméně. Aby bylo možné simplifikaci řídit, je nutné nejprve ohodnotit místa trojúhelníkové sítě na základě dané aplikace. Tato operace se nazývá *cenová funkce* a jejím úkolem je přiřadit ohodnocovaným primitivům (nejčastěji hranám) reálné číslo tak, aby udávalo cenu simplifikace daného primitiva.

Při každém kroku simplifikace dochází ke zvětšování odchylky od originální sítě, proto je potřeba určitým způsobem měřit, jak moc se liší původní trojúhelníková sít' od její aproximace (simplifikované sítě). Měření odchylky (chyby) simplifikace se používají tzv. *chybové metriky*, jejichž výsledkem je opět reálné číslo, které udává danou odchylku.

Simplifikační metody se snaží tyto chybové metriky minimalizovat výběrem správné cenové funkce a často také opravou simplifikované části - například umístěním nově vzniklého vrcholu při použití metody kontrakce hrany. V některých případech je však obtížné přesně definovat, jakou chybovou funkci se snaží daná simplifikační metoda minimalizovat. Představme si případ, kdy je použita metoda kontrakce hrany a hrany jsou ohodnoceny podle jejich velikostí. Dochází ke kontrakci hran postupně od nejkratší po nejdelší a pozice nového vrcholu kontrahované hrany je volena v místě jednoho z původních vrcholů hrany. Takový způsob má určitý smysl, ale nelze jednoznačně určit, jakou chybovou metriku minimalizuje. Mnoho chybových metrik se však tímto způsobem inspirovalo např. *Hausdorffova vzdálenost*.

Ke každé chybové metrice lze zkonstruovat cenovou funkci, která je založena na změně hodnoty chybové metriky při uvažovaném kroku simplifikace. Ani taková cenová funkce nezaručuje globální minimum dané chybové metriky.

4.1 Chybové metriky

4.1.1 Hausdorffova vzdálenost

Hausdorffova vzdálenost měří podobnost dvou povrchů trojúhelníkové sítě, přičemž využívá *euklidovské vzdálenosti* pro zjištění minimální vzdálenosti bodů na povrchu \mathbf{X} a \mathbf{Y} . Nejprve definujeme minimální vzdálenost bodu x , který náleží povrchu \mathbf{X} , k bodům z povrchu \mathbf{Y} :

$$d(x, \mathbf{Y}) = \min_{y \in \mathbf{Y}} \|x - y\|$$

Pak hodnota *Hausdorffovy vzdálenosti* je rovna:

$$d(\mathbf{X}, \mathbf{Y}) = \max_{x \in \mathbf{X}} d(x, \mathbf{Y})$$

Je nutné podotknout, že tato vzdálenost není symetrická. Platí tedy, že $d(\mathbf{X}, \mathbf{Y}) \neq d(\mathbf{Y}, \mathbf{X})$. Symetrická *Hausdorffova vzdálenost* je tedy definována:

$$d_s(\mathbf{X}, \mathbf{Y}) = \max[d(\mathbf{X}, \mathbf{Y}), d(\mathbf{Y}, \mathbf{X})] \quad (4.1)$$

Jedná se o nepercepční metodu, která pouze porovnává vzdálenosti bodů a nijak nebere v úvahu lidské vnímání. Podrobně se touto metrikou zabývá práce [1] a práce [2].

4.1.2 Kvadrická chybová

Kvadrickou chybovou metrikou představili Garland a Heckbert v jejich práci [4]. Tato metrika byla navržena pro simplifikaci metodou kontrakce páru vrcholů, ale dá se bez problémů využít i pro metodu kontrakce hran. Jde o kompletní metrikou, která řeší umístění nového vrcholu, cenovou funkci i výpočet geometrické chyby.

Výpočet chyby

Každému vrcholu trojúhelníkové sítě je na počátku přiřazena symetrická matice \mathbf{Q} o rozměrech 4×4 , která určuje ve vrcholu $\mathbf{v} = [v_x \ v_y \ v_z \ 1]^T$ chybu v kvadrickém tvaru:

$$\Delta(\mathbf{v}) = \mathbf{v}^T \mathbf{Q} \mathbf{v} \quad (4.2)$$

Výpočet chyby pak spočívá v součtu čtverců vzdáleností daného vrcholu k rovinám, které jsou určeny sousedními trojúhelníky, jež se ve vrcholu protínají. Výpočet lze zapsat takto:

$$\Delta(\mathbf{v}) = \mathbf{v}^T \left(\sum_{\mathbf{p} \in \text{planes}(\mathbf{v})} \mathbf{K}_{\mathbf{p}} \right) \mathbf{v}$$

Matice $\mathbf{K}_{\mathbf{p}}$ se nazývá *fundamentální chybová kvadrika* a používá se pro nalezení čtverce vzdáleností z jakéhokoliv bodu v prostoru k rovině \mathbf{p} . Rovina \mathbf{p} je definována jako vektor $\mathbf{p} = [a \ b \ c \ d]^T$ a je definována rovnicí $ax + by + cz + d = 0$, kde musí platit $a^2 + b^2 + c^2 = 1$. Výpočet fundamentální matice $\mathbf{K}_{\mathbf{p}}$:

$$\mathbf{K}_{\mathbf{p}} = \mathbf{p}\mathbf{p}^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

Součet všech fundamentálních matic je pak výše zmíněná matice \mathbf{Q} , která s sebou nese informace o rovinách, jež byly v původních vrcholech.

Umístění nového vrcholu

Pro nový vrchol $\bar{\mathbf{v}}$, který vznikne z vrcholů \mathbf{v}_1 a \mathbf{v}_2 kontrakcí hrany $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \bar{\mathbf{v}}$ platí, že jeho matice $\bar{\mathbf{Q}} = \mathbf{Q}_1 + \mathbf{Q}_2$. Jeho pozice by měla být taková, aby minimalizovala chybu $\Delta(\bar{\mathbf{v}})$. Jde o lineární problém nalezení minima a vypočítá se z matice $\bar{\mathbf{Q}}$.

$$\bar{\mathbf{v}} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.3)$$

Může ovšem nastat situace, že daná matice je singulární a nemá matici inverzní. V takovém případě je nutné zvolit jiný způsob umístění vrcholu jinou metodou, výběrem nejlepší pozice na kontrahované hraně, průměrem vrcholů hrany, a nebo výběrem pozice jednoho z vrcholů hrany.

Cenová funkce

Jak již bylo zmíněno výše, metrika umožňuje výpočet ceny kontrakce hrany $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \bar{\mathbf{v}}$ a její hodnota je rovna chybě:

$$\bar{\mathbf{v}}^T (\mathbf{Q}_1 + \mathbf{Q}_2) \bar{\mathbf{v}} \quad (4.4)$$

Zde je vidět, jak jsou někdy cenové funkce svázány s chybou simplifikace.

4.1.3 MSDM

Metodu *Mesh Strctural Distortion Measure (MSDM)* představil Lavoué a kolektiv ve své práci [5]. Jedná se o percepční metodu pro porovnání dvou trojúhelníkových sítí. Metoda vychází z Wangovy práce [10] pojednávající o obrazové metodě SSIM (viz část 4.3.3 na straně 19), která porovnává 2D obrazy s ohledem na lidské vnímání člověka. Podobně jako Wang a kolektiv použili ve své práci porovnání jasu, Lavoué a kolektiv upravili metodu pro porovnání křivosti. Křivost ve vrcholech sítě je založena na metodě *Normal Cycle*, kterou představil Cohen-Steiner a Morvan [3]. Místo lokálního okénka každého pixelu na 2D obrazu (11×11 sousedních pixelů) použili lokální okénko pro každý vrchol trojúhelníkové sítě ve tvaru koule se středem ve vrcholu v a poloměrem r . Hodnoty křivosti jsou uloženy ve vrcholech. Body protínající kouli jsou rovněž přidány do výpočtu a jejich křivosti jsou interpolovány od svých sousedů podle vzorce:

$$C(v_e) = \frac{d_2}{d_1 + d_2} C(v_1) + \frac{d_1}{d_1 + d_2} C(v_2)$$

kde d_1 a d_2 jsou vzdálenosti průsečíku od prvního a druhého vrcholu hrany a $C(v_e)$ je křivost bodu v_e . Pro každé okénko x je pak definován průměr křivostí μ_x a směrodatná odchylka σ_x jako:

$$\mu_x = \frac{1}{n} \sum_{v_i \in x} C(v_i)$$

$$\sigma_x = \sqrt{\frac{1}{n} \sum_{v_i \in x} (C(v_i) - \mu_x)^2}$$

v_i je i -tý vrchol okénka x a n je počet vrcholů v tomto okénku. Pro dvě okénka x a y na dvou trojúhelníkových sítích je definována kovariance jako:

$$\sigma_{xy} = \frac{\sigma_{xy}^x + \sigma_{xy}^y}{2}$$

$$\sigma_{xy}^x = \frac{1}{n} \sum_{v_i \in x} (C(v_i) - \mu_x) (C(u_i) - \mu_y)$$

Obdobně je pak definována funkce σ_{xy}^y . u_i je nejbližší vrchol okénka y k vrcholu v_i z okénka x .

Podobně jako Wang a kolektiv definovali tři rovnice pro porovnání jasu, kontrastu a struktury, Lavoué a kolektiv upravili rovnice pro tuto metodu a definovali tak tři rovnice pro porovnání křivosti (L), kontrastu (C) a struktury (S) takto:

$$L(x, y) = \frac{\|\mu_x - \mu_y\|}{\max(\mu_x, \mu_y)}, \quad C(x, y) = \frac{\|\sigma_x - \sigma_y\|}{\max(\sigma_x, \sigma_y)}, \quad S(x, y) = \frac{\|\sigma_x \sigma_y - \sigma_{xy}\|}{\sigma_x \sigma_y}$$

Pro lokální okénko se vypočítá vzdálenost MSDM:

$$LMSDM(x, y) = (\alpha \times L(x, y)^a + \beta \times C(x, y)^a + \gamma \times S(x, y)^a)^{\frac{1}{a}}$$

Lavoué a kolektiv ve své práci uvádějí, že hodnota a by měla být volena mezi 2,5 a 4 (v práci použili hodnotu 3) a koeficienty α, β a γ jsou zvoleny 0,4, 0,4, a 0,2. Celková MSDM vzdálenost je rovna:

$$MSDM(X, Y) = \left[\frac{1}{M} \sum_{j=1}^M LMSDM(x_j, y_j)^a \right]^{\frac{1}{a}} \in [0, 1] \quad (4.5)$$

4.1.4 MSDM2

Metoda MSDM2 byla představena v Lavouéovo práci [6] a jedná se o vylepšení předchozí metody MSDM. Tato metoda lépe koreluje s lidským vnímáním. Hlavní zlepšení této metody oproti původní MSDM jsou:

- zakřivení je závislé na měřítku modelu - zlepšená robustnost algoritmu,
- MSDM2 není závislé na konektivitě trojúhelníkové sítě,
- statistické funkce nyní používají Gaussovu váhovou funkci,
- průchod vyhodnocování je symetrický, to znamená, že je měřena vzdálenost originální sítě od zkoumané a obráceně, a s použitím několika-násobného měřítka modelu,
- metoda více odpovídá lidskému vnímání a rozhodování.

Metoda zkoumá zakřivení pomocí několika různých poloměrů okénka (koule) - tím bylo docíleno mnohem lepšího výsledku. Do původních vzorců pro výpočet statistických údajů o křivosti, kontrastu a struktury byla přidána Gaussova váhová funkce a vzorce byly upraveny pro výpočet více měřítek modelu. Všechny vzorce včetně popisu celé metody jsou uvedeny v práci [6].

4.2 Cenové funkce

Hlavním úkolem cenových funkcí je ohodnotit zkoumanou část trojúhelníkové sítě. Při použití metody kontrakce hrany se ohodnocují jednotlivé hrany a jsou jim přiřazovány reálné hodnoty, které udávají cenu kontrakce. Hrany

jsou pak seřazeny v prioritní frontě a jsou postupně kontrahovány od nejnižší ceny. Na začátku simplifikace je potřeba provést ohodnocení hran pro celou trojúhelníkovou síť. Následuje samotná simplifikace a hrany jsou vybírány z prioritní fronty. Po kontrakci hrany dojde ke změně struktury v okolí kontrakce, a proto je nutné přepočítat cenu všem hranám, které jsou v tomto okolí.

Cenové funkce mohou mít různé podoby a mohou během výpočtu zkoumat různé aspekty kontrakce hrany. Cenové funkce jsou obvykle úzce svázané s chybovými metrikami. Často při svém výpočtu provedou dočasnou kontrakci zkoumané hrany a na základě této kontrakce ohodnotí cenu hrany podle dané chybové metriky. Cena je pak vypočítána podle toho, jak moc se po této operaci daná oblast poškodila. Není to však podmínka a existují i takové cenové funkce, které žádné chybové metriky neuvažují a ohodnocují pouze na základě konektivity sítě. Například již dříve zmíněné ohodnocení hran podle jejich délky.

Některé cenové funkce jsou uvedeny jako součást metrik v částech 4.1 a 4.3.

4.3 Obrazové metriky

Obrazové metriky nahlíží na problematiku simplifikace trojúhelníkových sítí z trochu jiného úhlu. Geometrické metriky se využívají pro simplifikaci modelů, které budou použity pro další výpočty. Výsledný model proto musí být model co nejpřesnější z pohledu geometrie a případně i objemu a tvaru. Obrazové metriky zkoumají model z vizuálního pohledu a řeší převážně to, jak bude model po simplifikaci vypadat. Jako názorný příklad lze uvést použití těchto simplifikačních metrik na modelu, který je dutý (dutá koule apod.). Pokud na tento model použijeme geometrickou metriku, dojde k simplifikaci rovnoměrně na celém modelu včetně vnitřních ploch. Při použití obrazové metriky dojde ke zjednodušení převážně vnitřku, neboť z pohledu člověka tato část není zvenku vidět.

Jak už název napovídá, obrazové metriky neporovnávají geometrii modelu, ale jejich snímky pořízené vizualizačním softwarem (*renderer*). Tyto metriky jsou založené na metodách, které měří podobnost dvou zadaných snímků. Na základě této podobnosti mohou ohodnocovat hrany a určit tak cenu jejich simplifikace v závislosti na změně v těchto snímcích. Obrazové

metriky často přebírají funkci umístění nového vrcholu z jiných, většinou geometrických, metrik. Měření chyby a ohodnocování hran je prováděno na základě indexu určujícího podobnost obrazů.

Při použití obrazových metrik nastává problém, jak zvolit umístění kamer, které snímají model tak, aby pokryly co největší počet úhlů pohledu na model. Tyto metriky mají tendenci zjednodušovat převážně části, které nejsou vidět (vnitřní prostory modelu, zakryté části, apod.). Lindstrom a Turk ve své práci [7] zvolili umístění kamer do vrcholů geometrického tvaru *dodecahedronu* pro cenovou funkci a *rhombicuboctahedronu* pro výsledné porovnání modelů. Takové rozdělení je voleno proto, že pro cenovou funkci stačí menší rozlišení snímků a menší počet vrcholů než pro výsledné porovnání a výpočet indexu podobnosti. Porovnání a výpočet indexu podobnosti vyžaduje vyšší rozlišení snímků a větší počet pozic kamer pro zvýšení přesnosti. *Dodecahedron* obsahuje 20 vrcholů a *rhombicuboctahedron* 24 vrcholů. Tyto tvary jsou vepsány do tvaru koule o poloměru, který je roven úhlopříčce ohraničujícího kvádrů daného modelu (tzv. *bounding box*). Pro rychlé výpočty cen hran postačí velikost obrazů 256×256 a pro porovnání výsledného modelu velikost 512×512 .

4.3.1 Metody porovnání dvou obrazů

Metoda RMSE

Metoda RMSE (*Root Mean Square Error*) porovnává obrazy pixel po pixelu a vrací nezápornou hodnotu určující podobnost těchto obrazů. Výpočet je prováděn podle vzorce:

$$d_{RMS}(Y^0, Y^1) = \sqrt{\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (y_{ij}^0 - y_{ij}^1)^2} \quad (4.6)$$

kde y^0 a y^1 jsou jasové složky prvního a druhého obrazu, které se vypočítají podle NTSC koeficientů (pro RGB = [0.299, 0.587, 0.114]). m a n jsou rozměry zadaných obrazů v pixelech.

Hodnota RMSE udává rozdíl jasových složek obrazů. Čím je tato hodnota nižší, tím je tento rozdíl menší. Může se tak stát, že pro různé obrazy můžeme dostat stejné hodnoty, i přestože se vizuálně velmi liší a naopak, viz práce

Wanga a kol. [10].

Percepční obrazová metrika SSIM

Wang a kol. ve své práci [10] představili percepční obrazovou metriku, která zkoumá podobnost obrazů podle lidského vnímání obrazu. Metoda SSIM (*Structural SIMilarity*) zkoumá strukturu obrazů a je založena na statistických výpočtech pro tři složky obrazu. Porovnává se jasová složka, kontrastní složka a strukturní složka obrazu. Pro výpočty je vhodné převést obraz na posloupnost jeho jasových hodnot nebo-li signál \mathbf{x} . Pro jasovou složku signálu \mathbf{x} se vypočítá její střední hodnota jako:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i$$

Od signálu \mathbf{x} se odečte střední hodnota, takže je signál rozprostřen okolo nuly. Pro výpočet kontrastu signálu se používá výpočtu směrodatné odchylky σ_x :

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)$$

Následuje vydělení signálu \mathbf{x} jeho směrodatnou odchylkou σ_x a tím dojde k jeho normalizování. Tím budou mít dva porovnávané signály stejnou směrodatnou odchylku. Struktura obrazů je prováděna na normalizovaných signálech $(\mathbf{x} - \mu_x)/\sigma_x$ a $(\mathbf{y} - \mu_y)/\sigma_y$.

Výsledný index podobnosti je počítán na základě porovnání jednotlivých složek. Porovnání jasové složky je provedeno podle vzorce:

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (4.7)$$

kde C_1 je konstanta, která zabraňuje nestabilitě pokud $\mu_x^2 + \mu_y^2$ je blízko nule. Tato konstanta se volí na základě rozsahu hodnot pixelu L (255 pro černobílé obrazy) a hodnoty $K_1 \ll 1$, což je volitelná konstanta. C_1 se pak vypočítá jako $C_1 = (K_1 L)^2$.

Porovnání kontrastní složky je velice podobné:

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (4.8)$$

kde $C_2 = (K_2L)^2$ a opět je $K_2 \ll 1$.

K porovnání struktury obrazu se využívá faktu, že oba obrazy byly normalizovány, a proto výpočtem korelace mezi nimi dostaneme hodnotu vyjadřující podobnost obrazových struktur.

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (4.9)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$$

Z rovnic 4.7, 4.8 a 4.9 lze nyní složit funkci, která vrátí podobnost signálů \mathbf{x} a \mathbf{y} :

$$SSIM(\mathbf{x}, \mathbf{y}) = l(\mathbf{x}, \mathbf{y})^\alpha \cdot c(\mathbf{x}, \mathbf{y})^\beta \cdot s(\mathbf{x}, \mathbf{y})^\gamma \quad (4.10)$$

kde konstanty $\alpha > 0$, $\beta > 0$ a $\gamma > 0$ a slouží k nastavení závislosti těchto složek. Vzorec pro výpočet podobnosti obrazů metodou SSIM ve speciálním tvaru je následující:

$$SSIM(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4.11)$$

Pro K_1 a K_2 jsou ve Wangově práci stejně jako v této práci použity hodnoty 0,01 a 0,03.

4.3.2 Jednoduchá obrazová metrika MSE

Lindstorm a Turk ve své práci [7] uvádějí jednoduchou obrazovou metriku, která je založena na metodě RMSE. Metoda RMSE je obecně vhodná pro porovnání původního obrazu modelu s obrazem výsledného simplifikovaného

modelu. Pro cenovou funkci se využívá upraveného vzorce, který neobsahuje odmocninu (zrychlení výpočtu) a porovnává původní obraz modelu, obraz v současném kroku simplifikace a obraz po odstranění testované hrany. Takto upravený vzorec se označuje jako průměr čtverců neboli MSE (Mean Square Error). Cena hrany se pak vypočítá podle vzorce:

$$f(e, k) = \sum_{h=1}^l \sum_{i=1}^m \sum_{j=1}^n [(y_{hij}^0 - y_{hij}^{k+1})^2 - (y_{hij}^0 - y_{hij}^k)^2] \quad (4.12)$$

kde l je počet zkoumaných snímků, m a n jsou rozměry snímku v kroku h a y_{hij}^0 , y_{hij}^k a y_{hij}^{k+1} jsou jasové složky pixelu na pozici i a j snímku originálu, snímku současné iterace a snímku po dodatečné kontrakci hrany.

4.3.3 Metrika SSIM jako simplifikační metoda

Použití metody SSIM jako simplifikační metody je velmi snadné a podobně jako v části 4.3.2 lze metodu SSIM použít pro porovnání snímku originálního modelu a snímku modelu po dodatečné kontrakci hrany a tím získáme hodnotu, jejíž rozdíl od jedničky můžeme použít jako cenu simplifikace zkoumané hrany. Jinak řečeno cena zkoumané hrany při použití metody SSIM je rovna průměru cen dané hrany vypočítané z N snímků:

$$f(e, \mathbf{X}, k) = \frac{1}{N} \sum_{i=1}^N (1 - SSIM(\mathbf{x}_{i0}, \mathbf{x}_{ik+1})) \quad (4.13)$$

kde \mathbf{X} je množina všech snímků z N pozic kamery a k je číslo snímku současné iterace.

4.3.4 Zrychlení výpočtů obrazových metrik

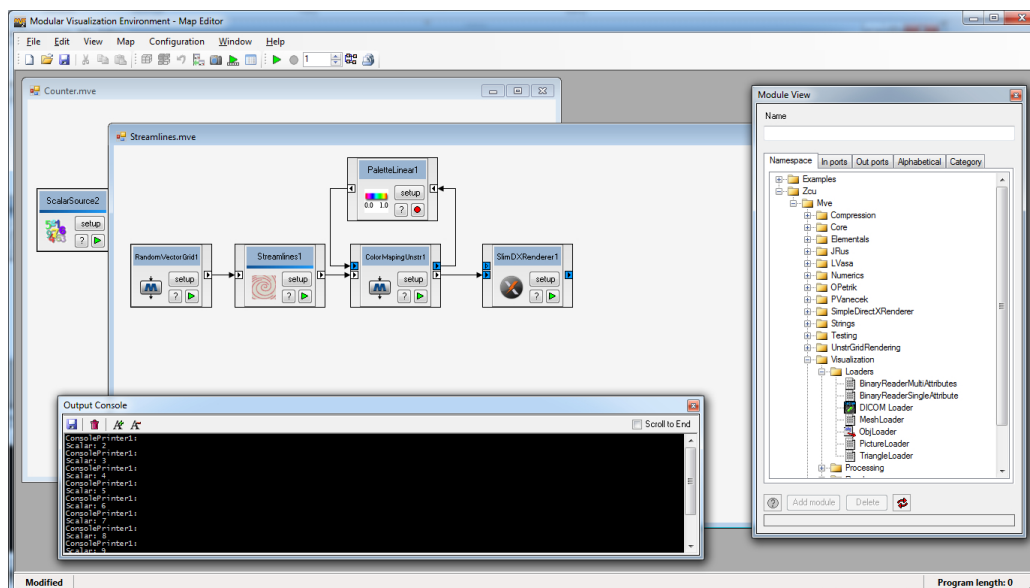
Počítání ceny na celém snímku je zbytečné, protože při kontrakci hrany dojde pouze ke změně v jejím okolí, tedy na trojúhelnících, které sousedí s danou hranou. Z tohoto důvodu lze vypočítat na snímku tzv. aktivní oblast simplifikace. Tato oblast je vlastně ohraničujícím obdélníkem všech trojúhelníků, které s hranou sousedí. Vzhledem k tomu, že nemusí být prováděn výpočet nad celým obrazem (například v rozlišení 256×256), ale postačí zkoumat pouze malou část snímku, dojde k velké úspoře času. Při použití metody

SSIM musí být zkoumána oblast větší než 11×11 pixelů, z důvodu použití okénka o stejné velikosti. Pro zlepšení výsledku metody SSIM se k tomuto okolí ještě přidává okraj, aby bylo zahrnuto i malé okolí ohraničujícího obdélníku.

5 Implementace v prostředí MVE-2

5.1 Prostředí MVE-2

Prostředí MVE-2 (*Modular Visualisation Environment 2*) je projekt, který vznikl na Západočeské univerzitě v Plzni. Je vyvíjen skupinou inženýrských a doktorandských studentů a vědeckých pracovníků. Projekt používá technologii .NET. Jeho hlavním cílem je vytvořit modulárně orientovanou platformu pro výzkum a vzdělávání.



Obrázek 5.1: Hlavní okno prostředí MVE-2

5.1.1 Moduly

MVE-2 je založeno na principu spojování modulů do funkčního celku. Modul je blok, který provádí definovanou činnost nad jeho vstupními porty a propaguje výsledky na jeho výstupní porty. Nemusí mít vždy vstupní i výstupní porty, ale měl by mít vždy alespoň jeden z nich, aby se dal dále propojovat s ostatními moduly. Moduly mají své nastavení parametrů, které lze v prostředí měnit.

Výhodou použití modulárního prostředí spočívá v odděleném řešení problémů nezávisle na ostatních programátorech, kteří pracují na svých projektech.

5.1.2 Mapy

Soustava propojených modulů je umístěna na tzv. mapě. Mapa umožňuje spustit proces definovaný moduly a jejich propojením. Těchto map může být spuštěno i více v jeden okamžik. Mapa také umožňuje nastavit program, podle kterého se provede spouštění mapy několikrát po sobě s různými parametry.

Tato práce se bude dále zabývat vytvářením dvou modulů a zapojením několika map, které tyto moduly využívají. Jak již bylo zmíněno výše, prostředí MVE-2 je založeno na technologii .NET a všechny moduly jsou psány v jazyce C#. Stejně tak v celé této práci bude využíváno právě jazyka C# a vývojového prostředí Microsoft Visual Studio 2010.

Více informací o prostředí MVE-2 a možnost jeho stažení je uvedeno na stránkách <http://herakles.zcu.cz/research/mve2>.

5.2 Implementace simplifikačního algoritmu

Jako metoda simplifikace geometrie byla zvolena kontrakce hrany (viz část 3.2.2 na straně 5). Tato metoda již byla v MVE-2 připravena v rámci modulu *Clusterizer*, a proto bylo jádro tohoto modulu převzato se souhlasem autora Ing. Oldřicha Petříka. Původní modul *Clusterizer* prováděl shlukování vrcholů do klusterů podobně jako metoda 3.2.4, ale s použitím kontrakce hran. Z tohoto modulu byly využity pouze třídy a metody, které tvořily metodu kontrakce hran a byly přeneseny do nového projektu. Modul *Clusterizer* počítal pouze s body dané trojúhelníkové sítě, proto bylo nutné převzaté třídy a metody upravit pro použití s trojúhelníky, aby mohla být provedena zpětná rekonstrukce modelu.

5.2.1 Třída *Collapser*

Třída *Collapser* je modifikovaná verze třídy *Clusterizer*. Metody byly upraveny tak, aby pracovaly pouze s kontrakcí hrany a neprováděly shlukování vrcholů. Třída také obsahuje definici delegátů pro cenovou funkci, umístění nového vrcholu, zastavovací podmínku a metodu, která se provede po kontrakci hrany. Delegát metody je vlastně reference na metodu, jež je definovaná podle daného předpisu.

Delegát cenové funkce obsahuje tři parametry - zkoumanou hranu, které bude v metodě přiřazena cena, odkaz na první vrchol a odkaz na druhý vrchol této hrany. Delegát metody umístění nového vrcholu má dva parametry - referenci na první vrchol hrany, který bude odstraněn, a druhý vrchol hrany, do kterého bude první vrchol sloučen. Tento vrchol zůstává zachován. Delegát zastavovací podmínky definuje opět tři parametry - počet zbývajících trojúhelníků, hran a cenu hrany čekající na kontrakci. Poslední delegát, který je zavolán po kontrakci hrany, neobsahuje parametry. Je zde z důvodu použití pro obrazové metriky, viz část 5.6 na straně 34.

Konstruktor

V konstruktoru dochází pouze k inicializaci privátních proměnných a k přiřazení proměnných zadaných v parametru konstruktoru.

Parametry konstruktoru jsou:

- pole trojúhelníků - pole všech trojúhelníků dané trojúhelníkové sítě pro sestavení soustavy hran a přiřazení trojúhelníků jednotlivým vrcholům,
- počet vrcholů - celkový počet vrcholů trojúhelníkové sítě,
- delegát cenové funkce - reference na cenovou funkci, která je definována podle výše zmíněného delegáta,
- delegát zastavovací podmínky - obdobně jako delegát cenové funkce,
- delegát metody umístění nového vrcholu,
- delegát metody po kontrakci hrany - tento parametr není povinný, ale v této práci se využívá.

Metoda BuildPQ

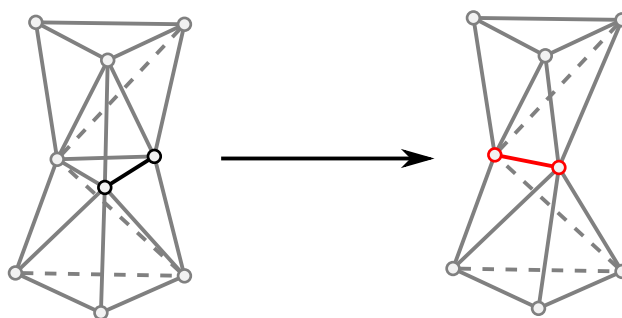
Metoda má za úkol vytvoření datových struktur potřebných pro simplifikaci. Tuto metodu je nutné zavolat před spuštěním simplifikace. Při vytvoření třídy *Collapser* je znám seznam trojúhelníků, které nesou informaci o indexech vrcholů, a pole pozic vrcholů. Simplifikační mechanismus však pracuje s hranami, a proto je nutné z těchto dvou datových struktur vytvořit seznam hran. V rámci této metody dochází k prvnímu ohodnocování vytvářených hran - prvotní fáze simplifikace. Zároveň se v metodě přiřazují odkazy na trojúhelníky vrcholům pro pozdější rekonstrukci sítě.

Metoda StartSimplification

Při zavolání metody *StartSimplification* se spustí simplifikační algoritmus. Vybere se první hrana s nejnižší cenou a zavolá se zastavovací podmínka, která určí, zda má algoritmus pokračovat, nebo dojde k zastavení simplifikace. Pokud algoritmus pokračuje je provedena metoda *CollapseStep*, která provede kontrakci hrany. Metoda *CollapseStep* je volána pouze v případě, že je daný krok validní - o tuto kontrolu se stará metoda *ValidCollapseStep*, viz níže. Jestliže krok není validní, je nastavena cena dané hrany na nejvyšší možnou hodnotu a zařadí se zpět do fronty (na poslední místo). Jakmile dojde k zastavení simplifikace, je nutné přepočítat indexy původních vrcholů, aby bylo možné použít menší datové struktury pro jejich uchování.

Metoda ValidCollapseStep

Metodě se předává parametr hrany, která je právě zkoumána. V této metodě se zjišťuje, zda po kontrakci hrany nedojde k chybné konektivě - vznik hrany incidentní s více než dvěma trojúhelníky. Takový problém může snadno nastat ve zúžených místech trojúhelníkové sítě, viz obrázek 5.2. Kontroluje se tedy případ, kdy daná hrana sousedí se dvěma hranami, které společně tvoří trojúhelník, ale tento trojúhelník není součástí trojúhelníkové sítě. V takovém případě je daná kontrakce hrany zamítnuta.



Obrázek 5.2: Problém kontrakce ve zúženém místě

Metoda CollapseStep

Jak již bylo výše zmíněno, metoda *CollapseStep* provádí samotnou kontrakci hrany. Vybere hranu s nejmenší cenou, její dva vrcholy a provede sloučení jednoho vrcholu do druhého. Zde je zapotřebí, aby se veškeré informace prvního vrcholu převedly do vrcholu druhého a současně došlo k odstranění trojúhelníků, které obsahují oba vrcholy. Zároveň se zde řeší problém uvedený v kapitole 3 v části 3.2.2 na straně 5. Pokud byla v konstruktoru předána funkce pro umístění nového vrcholu, je zavolána, aby umístila nově vzniklý vrchol. Následně je vybraná hrana odstraněna a pro všechny hrany, které jsou nyní spojeny s novým vrcholem musí být aktualizovány a přepočítány jejich ceny metodou *CreateEdge*.

Metoda CreateEdge

Této metodě se předávají dva parametry - index prvního a druhého vrcholu. Následuje vytvoření hrany typu *ConnectedEdge* mezi těmito vrcholy a zavolání cenové funkce, která tuto hranu ohodnotí. Tuto novou hranu pak metoda vrátí.

Metoda GetTriangles

Ve většině případů je od třídy požadována rekonstruovaná trojúhelníková síť simplifikovaného modelu. Tato metoda nejprve spustí samotnou simplifikaci. Po jejím dokončení projede všechny vrcholy sítě a poskládá nový seznam trojúhelníků ze záznamů v jednotlivých vrcholech. Jakmile je vytvořen seznam trojúhelníků, vytvoří se datová struktura, která obsahuje trojúhelníkovou síť.

5.2.2 Prioritní fronta EdgeCollapsePQ

Hrany jsou uchovávány v prioritní frontě, která je řazena podle jejich ceny. Tato fronta vhodně umísťuje nové hrany a drží tak vždy hranu s nejnižší cenou na začátku fronty. Celá tato třída byla převzata z původního modulu *Clusterizer*.

5.2.3 Podpůrné Třídy

Třída *ConnectedEdge*

Třída *ConnectedEdge* nese informace o dané hraně, její cenu, index a indexy vrcholů této hrany.

Třída *ConnectedPoint*

Podobně jako předchozí třída i třída *ConnectedPoint* obsahuje informace, ale tentokrát o vrcholu. Obsahuje informaci, zda byl tento vrchol odstraněn, index jeho potomka, jeho vlastní index, seznam hran, ke kterým je přiřazen, seznam sousedních trojúhelníků, seznam všech sousedních bodů a seznam vrcholů, které do něj byly sloučeny.

Obalující třída *TriangleEx*

Tato třída pouze obaluje strukturu *Triangle* do třídy, pro usnadnění použití referencování trojúhelníků.

5.3 Mechanismus pro implementaci metrik

5.3.1 Návrh obecného mechanismu

Jak již bylo v teoretické části této práce zmíněno, existuje mnoho simplifikačních metrik. Je tedy vhodné navrhnout takový mechanismus pro implemen-

taci nových metrik, aby bylo snadné implementovat novou metriku. Částečně tento problém řeší již výše popsaná třída *Collapser*, která s tímto přístupem počítá díky delegátním metodám pro cenovou funkci, umístění nového vrcholu a zastavovací podmínky. Proto bylo použito abstraktní třídy, která bude obsahovat instanci třídy *Collapser* a proměnné, které jsou společné pro všechny metriky. Při implementování nové simplifikační metriky bude stačit pouze implementovat abstraktní metody.

5.3.2 Bázová třída BaseSimplificator

Všechny nové simplifikační metriky jsou odvozeny od této bázové třídy. Třída obsahuje proměnné pro trojúhelníkovou síť, pozice vrcholů, maximální chybu, počet trojúhelníků a instanci třídy *Collapser*.

Konstruktor

Obsahuje konstruktor, kterému se předávají následující parametry:

- mesh - trojúhelníková síť zadaná MVE-2 třídou pro trojúhelníkové sítě,
- maximální chyba - maximální chyba simplifikace - při použití zastavovací podmínky založené na maximální chybě,
- maximální počet trojúhelníků - počet trojúhelníků, při kterém se zastaví simplifikační proces,
- pid - reference na ukazatel stavu modulu.

Konstruktor vyhledá a uloží informace o trojúhelníkové síti - uloží všechny trojúhelníky a body. Pak inicializuje instanci třídy *Collapser* a předá jí všechny potřebné parametry společně s referencemi na abstraktní metody uvedené níže.

Metoda Initialize

Třída *Collapser* potřebuje před spuštěním simplifikace provést přípravu trojúhelníkové sítě před samotnou simplifikací. Tato část je tzv. prvotní fáze simplifikace, kdy mimo jiné dojde k ohodnocení všech hran, ale ještě nedojde ke

kontrakcím. Proto je tato metoda oddělena od spuštění simplifikace, aby bylo možné například měřit čas této fáze. Tato metoda musí být zavolána, jinak nedojde ke spuštění simplifikace.

Metoda `GetConnectivity`

Tato metoda zajišťuje spuštění simplifikace a následné vrácení simplifikované trojúhelníkové sítě v MVE-2 třídě pro trojúhelníkové sítě.

Abstraktní metody

- Cenová funkce,
- zastavovací podmínka,
- metoda umístění nového vrcholu,
- metoda po kontrakci hrany,
- metoda uvolnění zdrojů - pokud by byly využity zdroje, které je nutné uvolnit, musí být tato metoda implementována.

Všechny výše uvedené abstraktní metody mají stejný tvar jako delegátní metody v části 5.2.1 na straně 23.

5.4 Implementace rendereru

Tato práce zkoumá mimo jiné možnosti a výhody obrazových metrik. K tomu, aby bylo možné porovnávat snímky zadané trojúhelníkové sítě, je nutné implementovat systém, který bude tyto snímky pořizovat co nejrychlejší možnou cestou, a bude možné jej použít přesně pro potřeby obrazových metrik. Jako nejlepší možný způsob byla zvolena implementace vlastního rendereru, který bude vytvářet snímky ze zadaných pozic kamery a tyto snímky předávat simplifikační třídě. Velká část rendereru byla převzata z vlastního modulu *SlimDXRenderer*, který slouží k vizualizaci trojúhelníkových sítí.

Vizualizace trojúhelníkové sítě je řešena pomocí knihovny *SlimDX*. Jedná se o volně šiřitelnou knihovnu pro .NET, která používá technologii *DirectX*

- v rendereru je využito pouze *Direct3D9*. Informace o knihovně *SlimDX* a její stažení lze nalézt na stránkách www.slimdx.org.

5.4.1 Třída **Renderer**

Hlavní částí rendereru je tato stejně pojmenovaná třída. Kompletní popis všech metod, jejich parametrů a použitých proměnných je nad rámec této práce, a proto zde budou uvedeny pouze důležité metody, které jsou potřebné pro obrazové metriky.

Důležité proměnné

Renderer obsahuje mimo jiné čtyři nejdůležitější proměnné, které souvisí s použitím pro obrazové metriky. Je to instance třídy *Mesh* (viz níže), offscreen textura pro rendering - jedná se o texturu, která je uložena v paměti grafické karty, a na kterou je prováděno veškeré kreslení (nevyužívá se plátna na obrazovce), dvojrozměrné pole textur v systémové paměti pro uložení jednotlivých snímků a pole kamer, ze kterých budou snímky pořizovány.

Metoda **ChangeResolution**

Jde o velmi důležitou metodu, která provádí nastavení rozlišení daných snímků a jejich počet na jednu kameru. Celkový počet snímků je určen podle počtu uložených kamer. Při simplifikaci je nutné porádít snímky originálního modelu, snímky po dočasné kontrakci zkoumané hrany a někdy také snímky v dané iteraci simplifikace. Tato metoda je většinou volána pouze jednou pro nastavení parametrů dané simplifikační metody. Změna rozlišení se hodí zejména v případě zkoumání vlivu rozlišení na výsledky obrazové metriky.

Metoda **CreateCameras**

Tato metoda zajišťuje pouze vytvoření pole kamer ze zadaných pozic. Kamery jsou nastaveny tak, že jejich zorný úhel je 60° a poměr stran je roven 1, protože se vytvářejí čtvercové snímky.

Metoda Render

Zavoláním této metody dojde k pořízení jednoho snímku z kamery, jejíž index je zadán jako parametr. Model je osvětlen jedním světlem, které je umístěno v místě kamery, takže jsou osvětleny všechny viditelné plochy. Tento snímek je uložen do pole textur v systémové paměti, do dimenze zadané opět parametrem. V této metodě dojde k vykreslení modelu na texturu, která je uložena v paměti grafické karty, a následně je přesunuta do systémové paměti. Proč byl zvolen takový postup je uvedeno v části 5.4.4 na straně 31.

Metoda SetupMesh

Před samotným vykreslením modelu, je samozřejmě nutné model připravit pro vykreslování a převést datovou strukturu z MVE-2 do podoby, kterou dokáže renderer vykreslit s využitím *Direct3D9*. Cílem této metody je vytvořit seznam vrcholů, normál a indexů vrcholů pro trojúhelníky a tyto parametry předat třídě *Mesh*. Rovnou se v této třídě provádí změna měřítko modelu tak, aby se celý vešel do kreslicí plochy a vypočítá se rádius ohraničující koule modelu.

5.4.2 Třída Mesh

Tato třída definuje datovou strukturu pro trojúhelníkovou síť, která využívá prostředky technologie *Direct3D9*, aby bylo možné tuto síť vykreslit. Třída obsahuje metodu *InitBuffers*, která provede převod pozic vrcholů, normálových vektorů a indexů pro trojúhelníky na datové struktury *Direct3D9*. Tato trojúhelníková síť je pak vykreslena metodou *Draw*.

5.4.3 Třída Camera

Třída *Camera* byla kompletně převzata z projektu *SlimDXRendererSystem*. Obsahuje velké množství metod pro manipulaci s kamerou a nastavení projekce. Jedinou důležitou informací o této třídě je, že počítá pohledovou a projekční matici používanou pro vykreslování.

5.4.4 Řešené problémy

V rámci implementace rendereru vzniklo několik problémů, které bylo nutné řešit. Prvním problémem byl čas výpočtu metody pro získání bitmapy z textury. Čas přesunu textury z grafické paměti do systémové paměti a následně převod do bitmapy zabral většinu času celého výpočtu obrazové metriky, a tak byla zvolena práce přímo nad texturou v systémové paměti. Aby byl tento čas ještě o něco kratší, bylo použito vykreslování na jednu velkou texturu o rozměrech několik megapixelů. Do určitého rozlišení tento přístup pracoval dobře, ale pro rozlišení snímku 512×512 tento přístup selhal. Proto došlo k rozdělení textur do dvojrozměrného pole. Na druhou stranu toto rozdělení nijak výrazně neovlivnilo čas převodu do systémové paměti.

5.5 Implementace simplifikačních metrik

V rámci této práce byly implementovány čtyři cenové funkce. Jako první byla zvolena metoda ohodnocení hrany podle její velikosti. Druhá cenová funkce byla implementována dle kvadratické chybové metriky (viz část 4.1.2 na straně 10) a jako poslední dvě cenové funkce byly implementovány obrazové metriky MSE a SSIM (viz část 4.3 na straně 15). Z kvadratické metriky byla použita rovněž metoda umístění nového vrcholu, která je společná pro všechny zkoumané metody. Z tohoto důvodu byly implementovány všechny cenové funkce v jedné třídě.

5.5.1 Třída Simplificator

Tato třída je odvozena od báze třídy *BaseSimplificator* a především implementuje všechny její abstraktní metody. Pro použití obrazových metrik obsahuje instanci rendereru, referenci na textury uložené v rendereru a pro výpočet nové pozice vrcholu po kontrakci hrany. Obecně i pro celou kvadratickou metriku, je implementováno pole matic \mathbf{Q} pro každý vrchol.

Konstruktor

Konstruktor obsahuje stejné parametry jako bazová třída *BaseSimplificator* a doplňují je parametry:

- rozlišení snímků - hodnota pro rozlišení snímku v jednom směru (používají se čtvercové snímky),
- použití aktivních oblastí snímku - aktivní oblasti snímku jsou popsány v části 4.3.4 na straně 19,
- SSIM okraj - toto číslo označuje, kolik pixelů bude přidáno ke každé aktivní oblasti jako okraj,
- metoda - určuje, jaká metoda (cenová funkce) bude použita pro simplifikaci.

V konstruktoru dojde k nastavení potřebných parametrů a inicializování rendereru v případě, že je použita obrazová metrika MSE nebo SSIM. V případě obrazových metrik dojde také k inicializování kamer a vytvoření snímků originálního modelu. Nakonec je rovnou spuštěna inicializace collapseru - prvotní fáze simplifikace.

Metoda CostFunction

Metoda cenové funkce implementuje abstraktní metodu bazové třídy. V cenové funkci se nejprve zjišťuje, zda kontrakce hrany je validní a neprovede otočení trojúhelníků, viz metoda *ValidContraction*. Dále jsou vypočítány aktivní body kontrakce - jedná se o všechny body, které sousedí s vrcholy zkoumané hrany. Vynásobením světové matice rendereru, pohledové a projekční matice kamery dostaneme zobrazení těchto bodů na daném snímku. Vybráním krajních bodů získáme ohraničující obdélník - tento výpočet zajišťuje metoda *ComputeActiveAreas*. Následuje výběr jedné z cenových funkcí podle zadané metody.

Metoda StopCondition

Podobně jako cenová funkce i tato metoda implementuje svojí abstraktní metodu z bazové třídy. Tato metoda zajišťuje pouze, aby byla simplifikace

ukončena, jakmile dojde ke snížení počtu trojúhelníků na zadaný maximální počet trojúhelníků.

Metoda ValidContraction

Tato metoda zabraňuje, aby došlo ke kontrakci hrany, po které dojde k otočení orientace některého z okolních trojúhelníků. Metoda vypočítá normály okolních trojúhelníků a uloží je. Následuje dočasná kontrakce hrany a vypočítání nových normál trojúhelníků. Tyto normály před a po kontrakci se porovnají a pokud jsou v opačném směru, pak je tato kontrakce hrany zamítnuta.

Metoda AfterCollapse a Dispose

Pomocí metody *AfterCollapse* se v případě použití obrazových metrik vytvoří snímky poslední iterace a metoda *Dispose* uvolní zdroje použité rendererem.

5.5.2 Implementace kvadrické metriky

Metoda umístění nového vrcholu po kontrakci hrany byla vybrána z kvadrické metriky z toho důvodu, že je jednoduchá na implementaci a obsahuje i ohodnocení hran. Existuje samozřejmě lepší řešení, například řešení uvedené v práci [7]. Toto řešení je ale velmi složité a na výsledcích zkoumaných metod by se zlepšení patrně příliš neprojeвило. Všechny metody jsou součástí výše uvedené třídy *Simplificator*.

Metoda ComputeQ

Jak již bylo uvedeno v kapitole 4 části 4.1.2 na straně 10, pro kvadrickou metriku se využívá k výpočtu chyby roviny trojúhelníků, které náležejí zkoumanému vrcholu. Tyto roviny jsou celou dobu simplifikace uchovávány v matici **Q**. Její výpočet provádí tato metoda.

Metoda `ComputeVertexPosition`

Metodě jsou předány dva vrcholy jako parametry. Pokud pro tyto vrcholy neexistuje matice \mathbf{Q} , je vypočítána výše zmíněnou metodou `ComputeQ`, jinak je matice vybrána z pole matic \mathbf{Q} . Prove se součet těchto matic a převod na homogenní tvar. Následuje test matice \mathbf{Q} , zda lze provést inverzi - tento test provádí metoda `TestMatrixQ`. Pokud matice není invertovatelná, je jako pozice vrcholu volena pozice prostředního bodu mezi prvním a druhým vrcholem. Pokud je invertovatelná, pak je pozice nového vrcholu uložena v posledním sloupci této matice.

Metoda `SetNewVertexPosition`

Tato metoda je opět implementací abstraktní metody umístění nového vrcholu z báze třídy. V první řadě je proveden výpočet pozice vrcholu metodou `ComputeVertexPosition`. Následně je uložen do pole matic \mathbf{Q} součet $\mathbf{Q}_1 + \mathbf{Q}_2$ na pozici druhého vrcholu (první je odstraněn). Pokud je použita obrazová metrika, pak je nutné tento bod aktualizovat i na trojúhelníkové síti používané v rendereru.

Cenová funkce

Tím, že byla použita kvadrická chybová metrika pro výpočet umístění vrcholu, byla implementace cenové funkce naprosto triviální. Nejprve se vypočítá umístění nového vrcholu a následně se provede vynásobení vrcholu a matice $\mathbf{Q} - \bar{\mathbf{v}}^T(\mathbf{Q}_1 + \mathbf{Q}_2)\bar{\mathbf{v}}$.

5.6 Implementace obrazových metrik

Součástí této práce je použití obrazových metrik pro simplifikaci a měření podobnosti dvou trojúhelníkových sítí. Pro komparaci trojúhelníkových sítí byla zvolena implementace obrazových metrik RMSE a SSIM (viz část 4.3.1 na straně 16) na celých snímcích a pro simplifikaci metodu MSE a SSIM na aktivních oblastech simplifikace (část 4.3.2 a 4.3.3).

5.6.1 Třída ImageMetrics

Jelikož je použití simplifikačních a komparačních obrazových metrik v této práci na sobě nezávislé a je použito odděleně, byla zvolena implementace metrik ve statické třídě, která obsahuje pouze statické metody pro výpočet indexu danou metodou.

Základ metod je stejný a jako vstupní parametry jsou použity dvě nebo tři textury, které obsahují snímky k porovnání. U všech metod je ještě možné přidat aktivní oblast, nad kterou má být metoda prováděna. Všechny metody pracují přímo nad danou texturou a nad jejími daty. Z každé textury jsou postupně vybírány pixely a přepočítávány jejich RGB hodnoty na jas dle NTSC koeficientů uvedených v části 4.3.1 na straně 16. Metody se liší pouze v rovnici výpočtu indexu. V případě metody MSE je použito tří textur.

Třída obsahuje tedy metody pro výpočet RMSE - *CompareRMS*, pro MSE - *CompareMSE* a pro SSIM - *CompareSSIM*.

5.6.2 Třída Utils

V části 4.3 na straně 15 bylo uvedeno, že k použití obrazových metrik se využívá soustavy kamer uspořádaných do tvaru *dodecahedronu*, případně do tvaru *rhombicuboctahedronu*. Vytvoření pole umístění vrcholů těchto tvarů je právě ve statické třídě *Utils*, která obsahuje pouze dvě statické metody. Metodu *CreateDodecahedron* a metodu *CreateRhombicuboctahedron*, které tyto pole vrcholů vytvoří.

5.6.3 Implementace SSIM metriky

Implementace SSIM metriky byla použita z webového zdroje <http://www.lomont.org/Software/Misc/SSIM/SSIM.html> a upravena pro použití v této práci. Původní implementace obsahovala navíc chybu ve výpočtu, kvůli které docházelo ke špatným výsledkům. Implementace této metody je rozdělena do dvou tříd.

Třída SlidingWindow

Třída reprezentuje matici, ve které jsou uložena obrazová data. Instance této třídy je zároveň použita jako klouzající okénko, které se pohybuje po druhé matici. Obsahuje definici základních operátorů pro sčítání, odečítání, násobení a dělení.

Třída SSIM

V této metodě je samotná implementace metody SSIM. Na počátku je vytvořeno klouzající okénko o rozměrech 11×11 a použitím Gaussova filtru o poloměru 1,5. Vytvoření tohoto okénka provede metoda *Gaussian*. Dále obsahuje metodu *Filter*, která provede filtrování prvního zadaného obrazu pomocí zadaného okénka. Následně metoda *ComputeSSIM* vypočítá index podle rovnic a parametrů, které jsou uvedeny v části 4.3.1 na straně 16. Hodnoty indexu SSIM se pohybují v intervalu $\langle 0, 1 \rangle$, kde číslo udává shodu obrazů. Toto číslo však není vhodné pro použití k ocenění hrany, a proto byla implementována funkce *ComputeSSIMCost*, která pouze provede odečtení $1 - \text{ComputeSSIM}$. Tato hodnota je pak použita jako cena simplifikace zkoumané hrany.

5.6.4 Cenové funkce obrazových metrik

Cenové funkce pro porovnání pomocí metriky MSE a SSIM jsou implementovány v jedné metodě - *CompareImageDriven*, která je součástí třídy *Simplificator*. V úvodu metody se vypočítají okrajové body pro aktivní oblast metodou *ComputeActiveArea*. Následuje výpočet umístění nového vrcholu po kontrakci hrany a dočasné kontrakci hrany na trojúhelníkové síti rendereru. Dále přicházejí na řadu samotné cenové funkce pro metodu MSE a SSIM, které jsou počítány nad jednotlivými snímky modelu a následně je použita jako cena kontrakce průměr těchto hodnot. Snímky modelu jsou pořízeny soustavou kamer ve tvaru *dodecahedronu*.

5.7 Modul *SimplificatorModule*

Aby bylo možné použít implementaci simplifikačních metrik v prostředí MVE-2, byl vytvořen modul s názvem *SimplificatorModule*. Modul obsahuje dva vstupní porty a jeden výstupní. Jako vstup je brána originální trojúhelníková síť a na výstup je posílána síť simplifikovaná podle zadaných parametrů. Druhý vstupní port je nepovinný a ovlivňuje maximální počet trojúhelníků, na které má být síť zdecimována.

Modul obsahuje nastavitelné parametry:

- simplifikační metoda - na výběr jsou metody *EdgeLength* - ohodnocení hran podle délky hrany, *Quadric* - použití kvadratické metriky, jednoduchá obrazová metrika *MSE* a *SSIM* metrika,
- rozlišení snímků pro obrazové metriky,
- okraj aktivních oblastí při použití *SSIM* metriky,
- použití aktivních oblastí - zda mají být použity, nebo bude použit k ohodnocování hran celý snímek,
- maximální počet trojúhelníků, který zůstane po simplifikaci.

5.8 Modul *MeshComparerModule*

Přestože je tato práce orientována převážně na problém simplifikace, pro změření výsledků jednotlivých metod bylo nutné implementovat i modul, který bude používat metodu *RMSE* a *SSIM* pro porovnání snímků originálního modelu a výsledného simplifikovaného modelu dané metody. Modul přijímá dvě trojúhelníkové sítě na jeho vstupních portech a výsledek metody vypisuje do konzole.

Modul obsahuje nastavitelné parametry:

- porovnávací metoda - na výběr jsou metody *RMS* a *SSIM*,
- rozlišení použitých snímků pro obrazové metriky,

- uložení snímků - tento parametr určuje, zda budou použité kontrolní snímky uloženy na disk. V případě použití metody SSIM budou uloženy také snímky SSIM mapy, ze které se metoda SSIM počítá.

5.8.1 Třída *MeshComparer*

Modul *MeshComparerModule* používá třídu *MeshComparer* k obsluze porovnávacího procesu. Obsahuje jedinou metodu *Compare*, které jsou předány zadané dvě trojúhelníkové sítě a parametry nastavené v modulu. Metoda vytvoří snímky prvního a druhého modelu a provede porovnání jednotlivých snímků zadanou metodou. Index je podobně jako cena simplifikace průměrem indexů všech snímků ze zadaných kamer, které jsou v tomto případě uspořádány do tvaru *rhombicuboctahedronu*.

6 Simplifikace modelů

Pro ověření simplifikačních metod bylo použito sedmi modelů. Tři modely byly vybrány takové, aby na nich byla otestována stabilita algoritmů a zbylé čtyři modely testují reálné použití simplifikačních metod. Přestože jsou první tři modely spíše zkušební, byly použity i pro hodnocení algoritmů v kapitole 7 na straně 43. Popis těchto modelů a jejich příprava je níže v části 6.1.

Některé modely byly upraveny nebo celé vymodelovány v programu Blender verze 2.62, který je volně k dispozici pod licencí GNU/GPL na stránkách <http://www.blender.org>. Modely *Lion*¹ a *Cow* jsou součástí distribuce prostředí MVE-2. Model *Horse* byl stažen z webové adresy <http://www.artist-3d.com>². Modely *Box*, *Surface* a *Vase* byly ručně vymodelovány a model *Monkey* byl použit z programu Blender, kde je jako standardní primitivum. Snímky originálních modelů jsou uvedeny v příloze A.

Model	Počet trojúhelníků	Po simplifikaci
Box	768	300
Surface	4352	2000
Vase	7808	4000
Lion	4422	2500
Cow	5798	3000
Horse	7160	3500
Monkey	4236	2000
Lion:	4422	2500

6.1 Příprava modelů

Celý simplifikační algoritmus dokáže fungovat i nad neucelenými sítěmi, ale program počítající metriku MSDM2 potřebuje k výpočtu ucelenou síť. Z tohoto důvodu bylo nutné modely *Cow* a *Horse* upravit v programu Blender. Model *Cow* byl rozdělen na dvě komponenty - tělo a ocas, takže bylo nutné tyto dvě komponenty spojit. Model *Horse* obsahoval přebytečné trojúhelníky,

¹Model *Lion* byl věnován vývojářům prostředí MVE-2 pro studijní účely panem Ing. Ivo Hanákem, Ph.D.

²K modelu *Horse* se váže licence pro nekomerční použití

které se překrývaly s jinými. Tyto vady byly odstraněny a poté byl model již ucelený.

6.1.1 Tvorba modelů v Blenderu

Model Box

Tento model byl vytvořen z krychle, na kterou byl použit modifikátor *subdivide*, který strany krychle rozdělil do mřížky. Následně byla mřížka převedena na trojúhelníky. Tento model byl určen ke zkoumání simplifikace rovných ploch a pravoúhlých hran. Na tomto modelu se objevily chyby, které souvisely s inverzí matice $\mathbf{Q}_1 + \mathbf{Q}_2$. Na rovné ploše totiž nelze matici $\mathbf{Q}_1 + \mathbf{Q}_2$ invertovat a vznikaly zde chybně umístěné vrcholy. Díky tomuto modelu byl problém odhalen.

Model Surface

Model byl postaven na základu čtvercové plochy, která byla opět rozdělena na mřížku a vrcholy mřížky byly posunuty tak, aby vytvořily zakřivenou plochu. Následně byl použit modifikátor *soldify*, který tuto plochu vytáhl a vytvořil z ní objemový model. Následně byl model opět převeden na trojúhelníkovou síť a uložen.

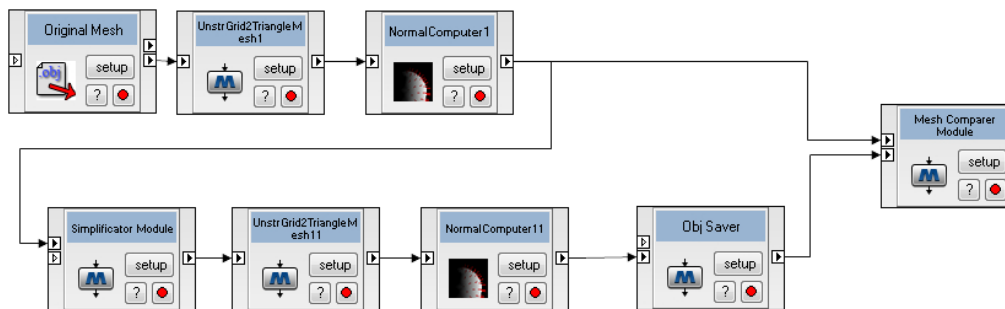
Model Vase

Jedná se o model, který zkoumá stabilitu použitých simplifikačních metrik na dutém modelu. Model byl tvořen z koule, která byla vytvarována do tvaru vázy a opět bylo použito modifikátoru *soldify*, který vytvořil vnitřní plochy vázy. U tohoto modelu byl předpoklad, že obrazové metriky selžou a jak je uvedeno v kapitole 7, není to chybou těchto metrik, ale použití umístění nového vrcholu. Přestože simplifikované verze tohoto modelu obsahují velké viditelné chyby, byl zahrnut tento model i do měření.

6.2 Simplifikační mapa

Po implementaci simplifikačního modulu a přípravě testovacích modelů přichází na řadu zpracování těchto modelů v prostředí MVE-2. Jak bylo uvedeno v popisu modulu *SimplificatorModule*, obsahuje vstupní port pro originální trojúhelníkovou síť a výstupní port, na který je přivedena simplifikovaná síť. Simplifikační modul nezachovává informace, které s sebou nese originální trojúhelníková síť, a proto zanikne například informace o normálách jednotlivých vrcholů. Je tedy nutné použít společně se simplifikačním modulem také modul pro výpočet normál. Dále je použit modul, který převede datovou strukturu *UnstrGrid* na strukturu *TriangleMesh*, kterou zpracovává modul pro výpočet normál. Simplifikační mapa tedy obsahuje:

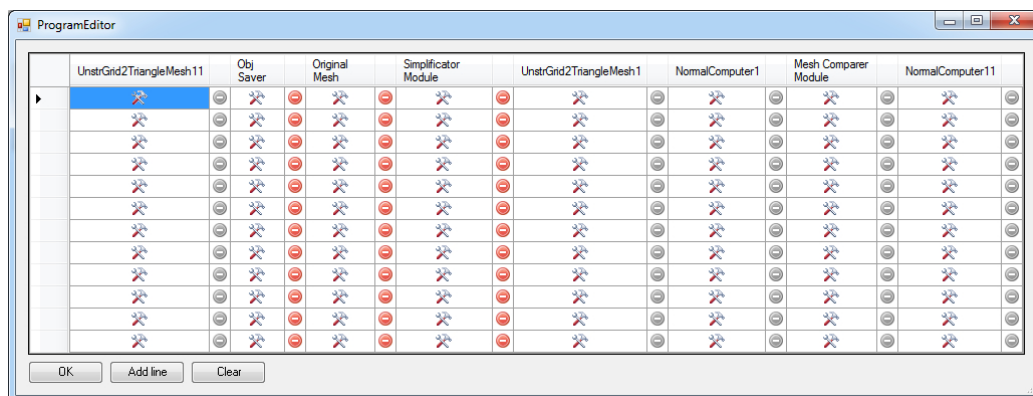
- Modul pro načtení modelu (*ObjLoader*),
- dva moduly pro převod datové struktury (*UnstrGrid2TriangleMesh*),
- dva moduly pro výpočet normál (*NormalComputer*) - jeden pro načtený model a druhý pro simplifikovanou síť,
- simplifikační modul (*SimplificatorModule*),
- modul pro uložení simplifikované sítě do formátu *.obj* (*ObjSaver*),
- modul pro komparaci originální trojúhelníkové sítě (*MeshComparerModule*) - místo tohoto modulu zde může být například vizualizační modul *SlimDXRenderer*, ale připojení tohoto modulu bylo užitečnější pro rychlou kontrolu výsledků.



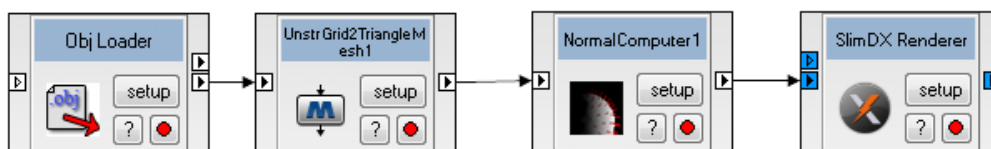
Obrázek 6.1: Simplifikační mapa

6.3 Program mapy

Mapy MVE-2 umožňují spuštění v režimu programu. Program lze editovat přímo v prostředí MVE-2, viz obrázek 6.2. Tento režim velice usnadnil hromadnou simplifikaci veškerých modelů všemi implementovanými metodami. Program umožňuje přidávat krok, ve kterém lze libovolně nastavit parametry jednotlivých modulů. Díky programu, který byl v tomto prostředí připraven, došlo k úspoře času, protože celý simplifikační proces nad všemi modely pracoval bez obsluhy. Po dokončení simplifikace byla provedena kontrola simplifikovaných modelů jednoduchou vizualizační mapou, viz obrázek 6.3.



Obrázek 6.2: Nastavení programu mapy



Obrázek 6.3: Vizualizační mapa

7 Měření a dosažené výsledky

Hlavním cílem této práce je zjistit, která z implementovaných metod provádí vizuálně lepší simplifikaci, a porovnat jednotlivé metody na základě změřených výsledků. U obrazových metrik se v této práci ještě ověřuje, zda má vliv na výsledek simplifikace použití vyššího rozlišení snímků použitých v cenové funkci. Časy simplifikace daných metod na použitých modelech lze nalézt v příloze B.

Pro měření bylo použito sedmi různých modelů, které byly simplifikovány následujícími metodami:

- EdgeLength - cenová funkce ohodnocující hrany podle jejich velikosti,
- Quadric - použití cenové funkce kvadrické chybové metriky,
- MSE-256 - použití obrazové metriky MSE při rozlišení snímků 256×256 ,
- MSE-512 - použití obrazové metriky MSE při rozlišení snímků 512×512 ,
- SSIM-256 - použití percepční obrazové metriky SSIM při rozlišení snímků 256×256 ,
- SSIM-512 - použití percepční obrazové metriky SSIM při rozlišení snímků 512×512 .

Porovnání originálních a simplifikovaných modelů bylo provedeno čtyřmi metodami:

- Hausdorffova vzdálenost - metoda výpočtu Hausdorffovy vzdálenosti,
- MSDM2 - percepční porovnání modelů metrikou MSDM2,
- RMS - porovnání modelů s využitím obrazové metriky RMS,
- SSIM - porovnání modelů s využitím percepční obrazové metriky SSIM.

Při tomto výzkumu by mělo být potvrzeno nebo vyvráceno tvrzení, že pokud se pro simplifikaci obrazovými metrikami použije percepční obrazová metrika SSIM v roli cenové funkce, pak dojde k poklesu percepční geometrické metriky MSDM2 vyhodnocené na výsledku. Poklesem se rozumí pokles

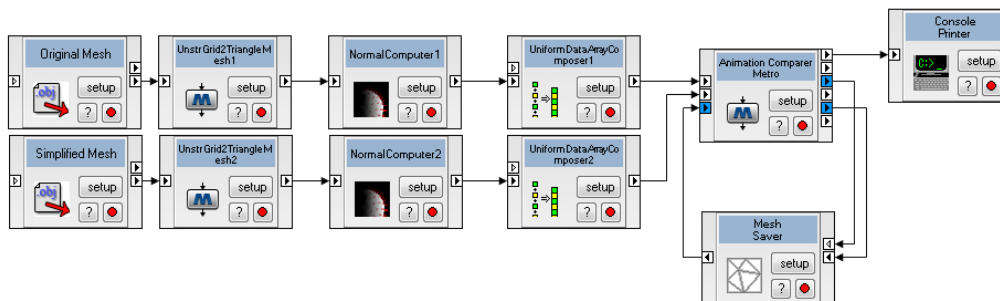
vzhledem k případu použití nepercepční obrazové metriky MSE za předpokladu stejného cílového počtu primitiv.

7.1 Porovnání modelů

7.1.1 Hausdorffova vzdálenost

Komparační mapa

K porovnání originálního modelu a zjednodušeného modelu pomocí Hausdorffovy vzdálenosti bylo potřeba v prostředí MVE-2 vytvořit mapu. Mapa využívá modul, který používá externí program Metro pro výpočet Hausdorffovy vzdálenosti. Program Metro je volně k dispozici na adrese <http://vcg.isti.cnr.it/activities/surfacegrevis/simplification/metro.html>. Protože modul porovnává dvě animace, bylo nutné použít ještě modul, který vytvoří ze zadaného modelu jednosnímkovou animaci. Na obrázku 7.1 je zapojení modulu v mapě.



Obrázek 7.1: Komparační mapa pro výpočet Hausdorffovy vzdálenosti

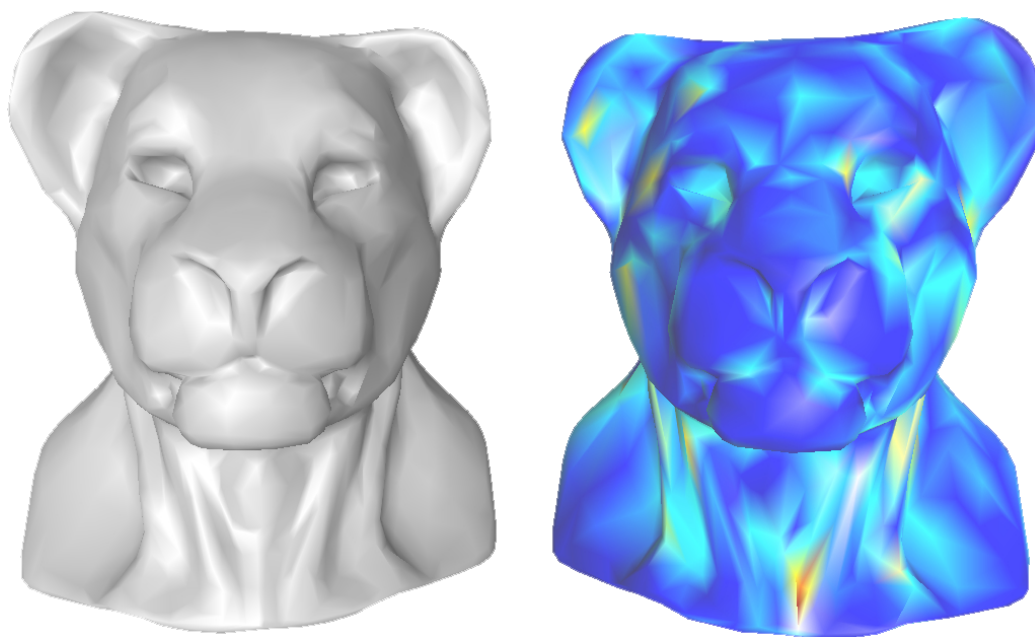
7.1.2 Vzdálenost MSDM2

Měření podobnosti dvou trojúhelníkových sítí bylo provedeno s využitím programu MEPP (*MESH Processing Platform*) verze 0.46.5, který je vyvíjen ve francouzském národním výzkumném centru. Je k dispozici na stránkách <http://liris.cnrs.fr/mepp/index.html> a je volně šiřitelný. Program umožňuje vizualizaci modelů a mnoho metod pro práci s nimi. Obsahuje

nejen metodu pro výpočet vzdálenosti MSDM2, ale i metodu pro výpočet první verze vzdálenosti MSDM. Stejně tak lze vypočítat a zobrazit křivosti trojúhelníkových sítí.

Změření hodnoty MSDM2

Nejprve je nutné načíst originální model a následně přidat do jeho prostoru (položka v menu *Open and Add (space)*) nebo jako snímek (*Open and Add (time)*) simplifikovaný model. Následně výběrem z možností *Components > Distance / Quality measure > MSDM2 > Compute MSDM2 distance* program vypočte vzdálenost MSDM2 mezi těmito dvěma modely. Poté je ještě možné zobrazit na simplifikovaném modelu barevnou mapu, která určuje, kde se simplifikovaný model nejvíce liší, viz obrázek 7.2.



Obrázek 7.2: Mapa vzdálenosti MSDM2 na modelu Lion metody SSIM-512

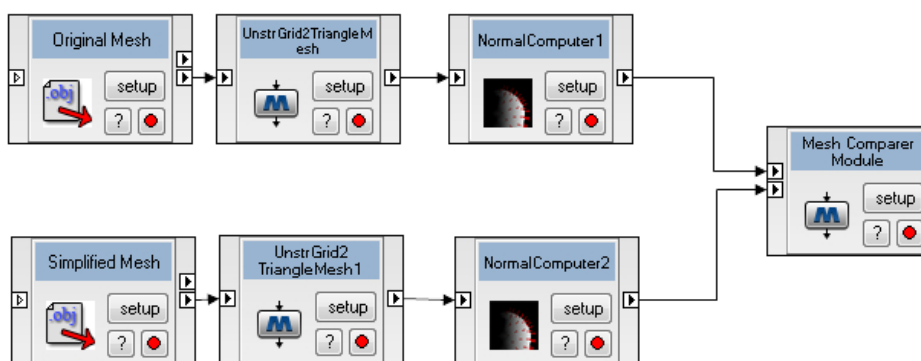
7.1.3 Měření indexu RMS a SSIM

Pro porovnání modelů obrazovými metrikami bylo využito modulu MeshComparerModule, který je uveden v části 5.8 na straně 37. Pro vyhodnocení indexu RMS i SSIM bylo použito rozlišení snímků 512×512 . Vyhodnocení

podobnosti modelů se provádí nad celými snímky a hodnoty nad snímky ze všech kamer jsou zprůměrovány.

Komparační mapa

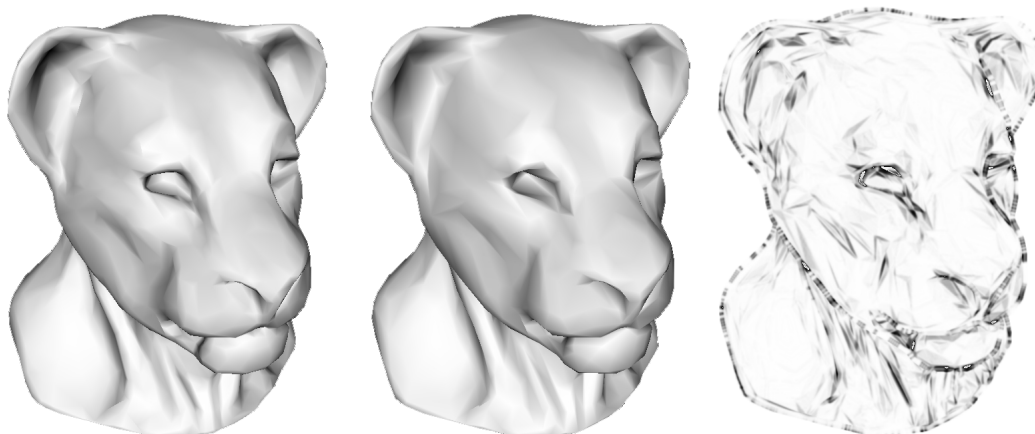
Zapojení modulu MeshComparerModule se skládá z připojení pouze dvou vstupních trojúhelníkových sítí. Modely jsou načteny ze souboru a poté dojde k přepočítání normál. Při použití komparace metodou SSIM modul umožňuje uložit snímky z výpočtu včetně SSIM mapy, která podobně jako v případě MSDM2 zobrazuje, kde dochází k rozdílu simplifikovaného modelu od originálu, viz obrázek 7.4. Mapa porovnávající dva modely obrazovými metrikami je znázorněna na obrázku 7.3.



Obrázek 7.3: Komparační mapa pro výpočet indexu obrazových metrik

7.1.4 Porovnání modelů uživateli

Součástí této práce bylo také zhodnocení simplifikovaných modelů i z pohledu uživatele. K tomuto průzkumu byl využit program MEPP, který umožnil otevření všech modelů v jednotlivých oknech a simplifikované verze v jednom okně seřadit za sebe jako snímky animace. Uživatel tak na jednom monitoru mohl prohlížet originální model a na druhém monitoru libovolně přepínal mezi simplifikovanými verzemi tohoto modelu. Celkem byl použit vzorek deseti respondentů, kteří byli řádně poučeni jak s programem zacházet a jak ohodnocovat jednotlivé modely. Bodování bylo určeno hodnotami 0 až 10,

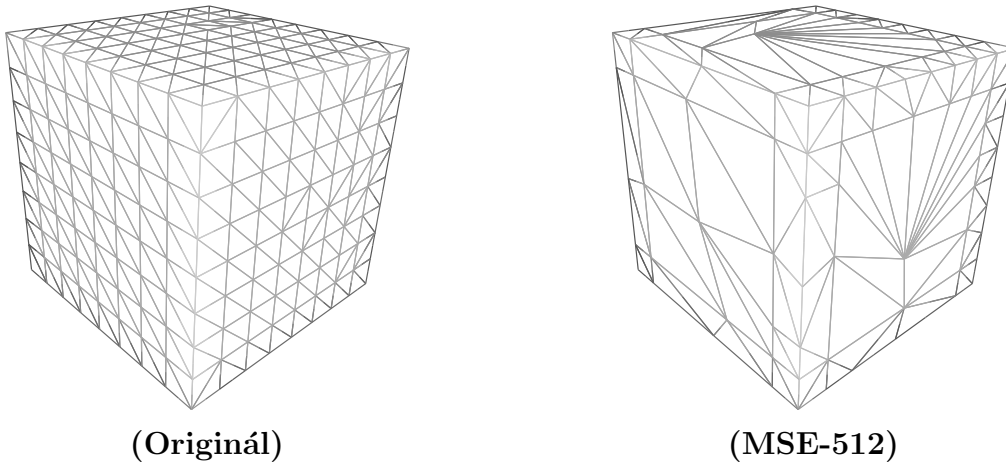


Obrázek 7.4: Snímky z jedné kamery při použití komparační metody SSIM včetně SSIM mapy

kteřé určovaly rozdíl zkoumaného modelu od jeho originálu. Hodnota 0 tedy představuje žádný rozdíl a hodnota 10 velký rozdíl. Uživatelé ale někdy nepoužili celý rozsah hodnot, a proto byly hodnoty normalizovány a namapovány na rozsah hodnot 0 až 1. Z těchto hodnot byl proveden průměr a výpočet konfidenčního intervalu pro ověření použitelnosti výsledku. Vzorek respondentů se sestával z technicky znalých i neznalých a z respondentů různé věkové kategorie. Přesto se jedná o velmi malý vzorek, který nelze považovat za statisticky dostatečný. Tato práce, ale nemá za úkol ověřit výsledky percepčních metod oproti lidskému vnímání, ale jejím úkolem je porovnat jednotlivé metody mezi sebou.

7.2 Výsledky měření

Jako první zkoumané modely byly zvoleny *Box*, *Surface* a *Vase*. Tyto modely jsou určeny převážně k prověření algoritmů použitých metod. Na modelu *Box* bylo zkoumáno především, jak se metody chovají na plochém modelu s ostrými hranami. Z tabulky 7.1 vychází nejlépe metoda MSE-512 následována metodou MSE-256. Dobré výsledky metody MSE jsou odůvodnitelné tím, že při kontrakci hran na okrajích krychle dochází ke znatelné změně jasu na obrázcích, a proto metody simplifikovaly nejvíce středy stran krychle. Metodu MSE vyhodnotila metrika MSDM2 jako nejlepší na tomto modelu, stejně tak uživatelé při výzkumu.



Obrázek 7.5: Porovnání metody MSE-512 oproti originálu

Pro porovnání hodnot vypočítaných komparačními metodami je využito Pearsonova korelačního koeficientu, který byl vyhodnocen podle uživatelského hodnocení. V případě metriky SSIM však vyšší hodnota značí lepší výsledek, proto bylo nutné korelační koeficient vynásobit hodnotou -1, protože by jinak vycházel záporný. V případě modelu Box vyhodnocení jednotlivých metod metrikou MSDM2 odpovídá z velké části rozhodování uživatelů a jejich hodnocení s korelačním koeficientem až 91%.

Výpočet Pearsonova korelačního koeficientu:

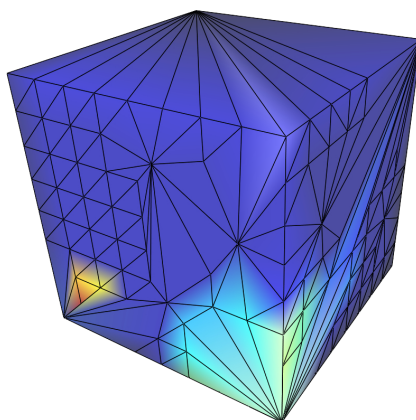
$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}} \quad (7.1)$$

kde \bar{x} a \bar{y} jsou střední hodnoty zadaných čísel.

Tabulka 7.1: Výsledky měření modelu Box

	Metoda	EdgeLength	Quadric	MSE-256	MSE-512	SSIM-256	SSIM-512	Korelační koeficient
Box	RMS	17,12260	35,52600	2,23962	1,23604	10,49556	9,56574	0,57546
	Hausdorff	0,03423	0,00000	0,00000	0,00000	0,00000	0,00000	0,78311
	SSIM	0,92354	0,92410	0,99833	0,99935	0,98151	0,98140	0,83331
	MSDM2	0,41466	0,33899	0,00002	0,00001	0,13598	0,14272	0,90828
	Uživatelé	0,96970	0,55649	0,25087	0,00000	0,36558	0,53160	

V tabulce 7.1 stojí za povšimnutí hodnoty, které vyhodnotila metoda výpočtu Hausdorffovy vzdálenosti. Metoda opravdu hodnotí pouze tvar modelu, zatímco metoda MSDM2 zohledňuje i normály trojúhelníků. Tento jev lze vi-



Obrázek 7.6: Vyhodnocení simplifikovaného modelu Box metodou MSDM2 (MSE-512)

dět na MSDM2 mapě na obrázku 7.6. Metoda MSE-512 skutečně provedla kontrakci hran převážně na středu stěn, takže tvar zůstal zcela zachován.

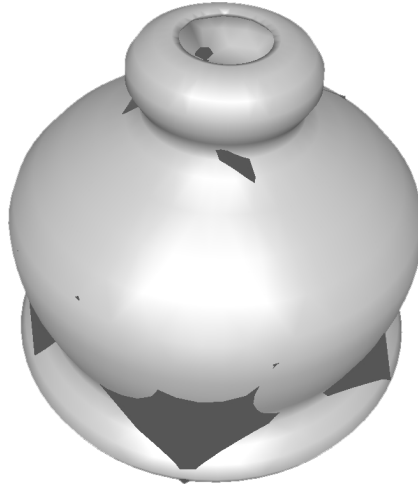
Model *Surface* je určen ke zkoumání postupu simplifikačních metod na křivé ploše, která má ostré okraje a je oboustranná. V případě metody EdgeLength došlo k poškození okrajů, a proto uživatelé i metrika MSDM2 udělili metodě nejhorší hodnocení. Zatímco metoda SSIM-512 obstála nejlépe. Rozhodování uživatelů se v tomto případě shodovalo s výpočtem metriky MSDM2 na 83%, viz tabulka 7.2.

Tabulka 7.2: Výsledky měření modelu Surface

	Metoda	EdgeLength	Quadric	MSE-256	MSE-512	SSIM-256	SSIM-512	Korelační koeficient
Surface	RMS	12,22225	6,93582	2,64534	2,96389	4,08028	3,89528	0,85218
	Hausdorff	0,00937	0,00108	0,00439	0,00318	0,00535	0,00535	0,39482
	SSIM	0,94814	0,98443	0,99370	0,99306	0,98761	0,98826	0,81496
	MSDM2	0,41575	0,27999	0,27419	0,28495	0,30067	0,26547	0,82739
	Uživatelé	1,00000	0,58333	0,42424	0,32576	0,23485	0,00000	

Zkoumaný model *Vase* byl určen ke zjištění, jak si obrazové metriky poradí s dutým modelem. Dle výsledků je na první pohled jasné, že tyto metody selhaly. Problém by nebyl ani tak v tom, že jako první zdecimovaly vnitřek vázy (což se očekávalo), ale tím, že použitá metoda umístění nového vrcholu nerespektuje zachování hranic vnějších ploch. Implementovaný renderer měl nastaven parametr ořezávání neviditelných ploch (trojúhelníků, jejichž orientace je proti směru hodinových ručiček), a proto obrazové metody ve většině případech nezaznamenaly umístění vrcholu mimo vnější plochy vázy. Vznikly tak chyby (viz obrázek 7.7), které ovlivnily veškeré hodnocení a nelze z nich

tedy vyvozovat závěr. I v tomto případě se uživatelské hodnocení velmi shodovalo s výsledky metricky MSDM2 až v 90%, viz tabulka 7.3.



Obrázek 7.7: Chyby při použití MSE-512 na modelu Vase

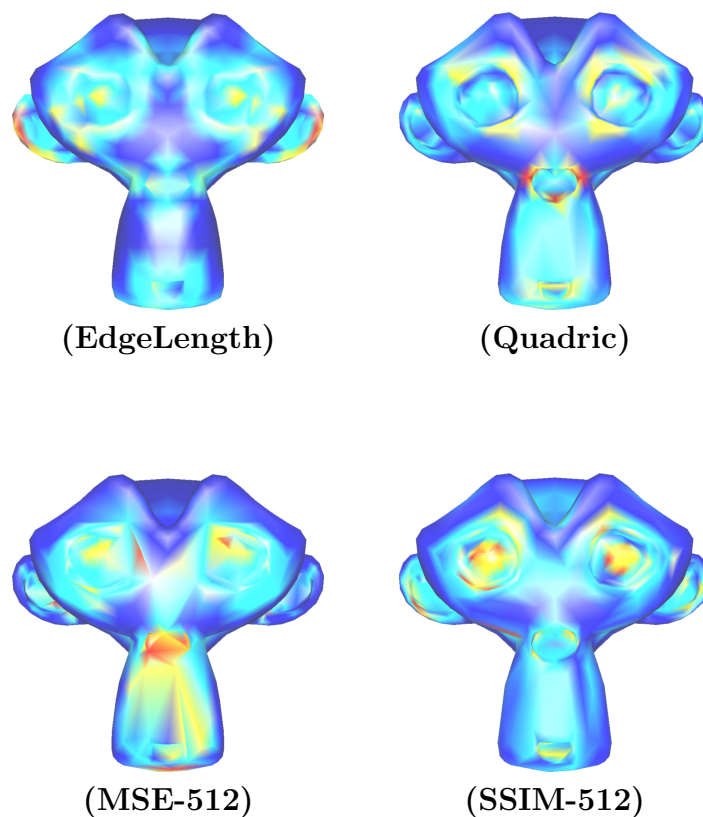
Tabulka 7.3: Výsledky měření modelu Vase

	Metoda	EdgeLength	Quadric	MSE-256	MSE-512	SSIM-256	SSIM-512	Korelační koeficient
Vase	RMS	6,14791	2,16102	4,64874	4,84736	5,34952	5,57709	0,60371
	Hausdorff	0,00578	0,00088	0,04960	0,07092	0,04868	0,04280	0,94635
	SSIM	0,97437	0,99503	0,99445	0,99373	0,99468	0,99430	-0,24306
	MSDM2	0,29916	0,17618	0,32033	0,31899	0,32707	0,32398	0,89702
	Uživatelé	0,40152	0,00000	0,79394	0,98485	0,72500	0,73712	

U hodnocení modelů *Lion*, *Monkey*, *Cow* a *Horse* se objevily velmi podobné výsledky metod. Dle hodnocení metricky MSDM2 vyšla jako nejlepší metoda Quadric následovaná metodou SSIM-512. Uživatelé však metodu Quadric jako nejlepší nevybrali, místo ní se na prvním místě umístila metoda SSIM-512. Zajímavé je, že u modelů *Lion* a *Horse* uživatelské hodnocení korelovalo spíše s hodnocením metody SSIM - 90% a 74%, kdežto s metrikou MSDM2 pouze na 67% a 36%. U modelů *Monkey* a *Cow* tomu bylo naopak, uživatelské hodnocení korelovalo s metodou SSIM na 18% a 26% a s metrikou MSDM2 na 66% a 78%.

Hodnocení metod metrikou MSDM2 lze vidět na příkladu modelu *Monkey* na obrázku 7.8. Metrika MSDM2 vytvoří barevnou mapu, kterou lze zobrazit v programu MEPP.

Na grafu 7.9 je porovnání metody MSE a SSIM vzhledem k výsledkům vzdálenosti MSDM2 na modelech *Lion*, *Monkey*, *Cow* a *Horse*. Je vidět, že



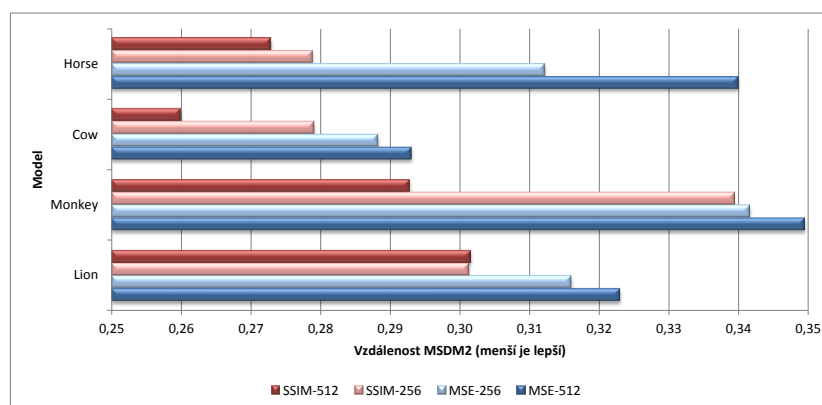
Obrázek 7.8: Porovnání modelu Monkey metodou MSDM2

při použití percepční metody SSIM opravdu dochází ke snížení hodnoty vzdálenosti MSDM2 vzhledem k použití metody MSE. Porovnání všech metod z hlediska uživatelského hodnocení naleznete v příloze C.

Všechny hodnoty, které byly použity včetně hodnot z uživatelského hodnocení jsou uloženy jako příloha na CD v souboru *results.xlsx*.

Tabulka 7.4: Výsledky měření modelů Lion, Monkey, Cow a Horse

	Metoda	EdgeLength	Quadric	MSE-256	MSE-512	SSIM-256	SSIM-512	Korelační koeficient
Lion	RMS	7,36659	4,84429	5,00837	5,44102	4,43640	4,89833	0,93045
	Hausdorff	0,00775	0,00244	0,00929	0,00859	0,00760	0,00640	0,08874
	SSIM	0,96841	0,97592	0,97707	0,97400	0,97975	0,97715	0,90208
	MSDM2	0,32811	0,29567	0,31601	0,32303	0,30128	0,30149	0,67159
	Uživatelé	0,90909	0,35606	0,30303	0,31818	0,21970	0,09091	
Monkey	RMS	5,83033	4,28803	4,10179	3,67797	5,19448	4,28280	0,32493
	Hausdorff	0,00768	0,00242	0,00899	0,00851	0,00807	0,00651	0,57170
	SSIM	0,97978	0,98633	0,98775	0,98966	0,98034	0,98603	0,17871
	MSDM2	0,32258	0,27194	0,34164	0,34957	0,33937	0,29274	0,66381
	Uživatelé	0,98182	0,18485	0,64545	0,74091	0,36970	0,00000	
Cow	RMS	7,61346	7,14232	4,77307	5,00594	7,36458	4,55628	0,47231
	Hausdorff	0,00707	0,00559	0,00913	0,01111	0,00697	0,00369	0,07175
	SSIM	0,97270	0,96977	0,98429	0,98248	0,96962	0,98521	0,25924
	MSDM2	0,31818	0,25358	0,28828	0,29312	0,27909	0,25997	0,77685
	Uživatelé	1,00000	0,28485	0,45152	0,21061	0,23333	0,17576	
Horse	RMS	7,67405	4,72746	4,70937	4,83796	5,26951	5,00340	0,74711
	Hausdorff	0,00754	0,00180	0,00636	0,00590	0,00480	0,00425	0,53231
	SSIM	0,97216	0,98463	0,98466	0,98421	0,98215	0,98326	0,73905
	MSDM2	0,31437	0,25803	0,31222	0,34001	0,27888	0,27289	0,35860
	Uživatelé	0,90909	0,43939	0,48485	0,40909	0,54545	0,03030	



Obrázek 7.9: Porovnání metody MSE a SSIM vzhledem k MSDM2

8 Závěr

Během této práce vznikl projekt, který může být dále rozšiřován a mohou být implementovány nové cenové funkce. Stejně tak lze snadno přidat nové obrazové metriky a využít rendereru pro jejich realizaci.

Podle výsledků se zdá, že použití percepční metody SSIM pro simplifikaci opravdu sníží percepční metriku MSDM2 oproti nepercepční metodě MSE, viz graf na obrázku 7.9. Kvadrická metoda sice dává velice podobné výsledky jako SSIM za mnohem menší čas, ale dle subjektivní studie tato metoda nedosahuje takové vizuální kvality, viz graf v příloze C. Toto tvrzení však nelze potvrdit, protože byl použit pouze malý vzorek respondentů. Také se ukázalo, že použití vyššího rozlišení pro snímky simplifikace pomocí metody SSIM produkuje kvalitnější modely, zatímco u nepercepční metody MSE to nelze obecně říci.

Pro další výzkum by mohlo být zajímavé implementovat metodu MSDM2 jako cenovou funkci a porovnat výsledky vzhledem k metodě SSIM. Stejně tak by se implementace v této práci dala vylepšit použitím komplexnější metody pro umístění nového vrcholu například podle práce [7]. Dalším možným vylepšením by mohlo být zlepšení rendereru tak, aby vytvářel více fotorealistické snímky. Dal by se také zkoumat vliv osvětlení modelu na simplifikaci metodou MSE nebo SSIM.

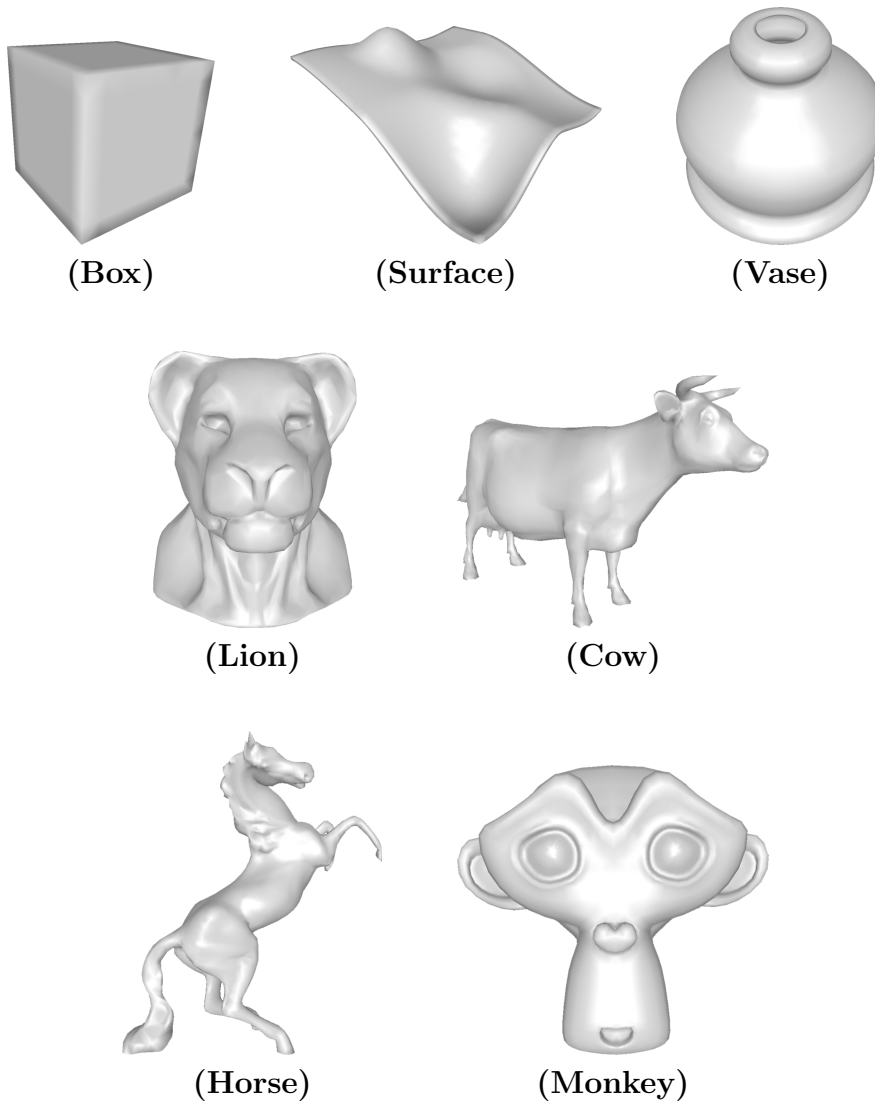
Literatura

- [1] Aspert, N.; Santa-Cruz, D.; Ebrahimi, T.: MESH: measuring errors between surfaces using the Hausdorff distance. In *IEEE International Conference*, ročník 1, Multimedia and Expo, Ieee, 2002, ISBN 0780373049, s. 705–708, doi:10.1109/ICME.2002.1035879.
- [2] Cignoni, P.; Rocchini, C.; Scopigno, R.: Metro: Measuring Error on Simplified Surfaces. *Computer Graphics Forum*, ročník 17, č. 2, Červen 1998: s. 167–174, doi:10.1111/1467-8659.00236.
- [3] Cohen-Steiner, D.; Morvan, J.-M.: Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry*, SCG '03, New York, NY, USA: ACM, 2003, ISBN 1-58113-663-3, s. 312–321, doi:10.1145/777792.777839.
URL <http://doi.acm.org/10.1145/777792.777839>
- [4] Garland, M.; Heckbert, P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, ISBN 0-89791-896-7, s. 209–216, doi:10.1145/258734.258849.
URL <http://dx.doi.org/10.1145/258734.258849>
- [5] Lavoué, G.; Gelasca, E. D.; Dupont, F.; aj.: Perceptually driven 3D distance metrics with application to watermarking. In *SPIE Applications of Digital Image Processing XXIX*, Srpen 2006, str. 63120L.
URL <http://liris.cnrs.fr/publis/?id=2437>
- [6] Lavoué, G.: A Multiscale Metric for 3D Mesh Visual Quality Assessment. *Computer Graphics Forum*, ročník 30, č. 5, Srpen 2011: s. 1427–1437, doi:10.1111/j.1467-8659.2011.02017.x.

-
- [7] Lindstrom, P.: *Model simplification using image and geometry-based metrics*. Dizertační práce, Atlanta, GA, USA, 2000, aAI9994428.
- [8] Ronfard, R.; Rossignac, J.: Full-range approximation of triangulated polyhedra. *Computer Graphics Forum*, ročník 3, č. 15, Srpen 1996: str. 67–76.
- [9] Schroeder, W. J.; Zarge, J. A.; Lorensen, W. E.: Decimation of triangle meshes. In *Computer Graphics*, ročník 26, editace E. E. Catmull, SIGGRAPH 92, Červenec 1992, ISBN 0-201-51585-7, str. 65–70.
- [10] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; aj.: Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, ročník 13, Duben 2004: s. 600–612, doi:10.1109/TIP.2003.819861.
URL <http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf>

Přílohy

A Použité modely



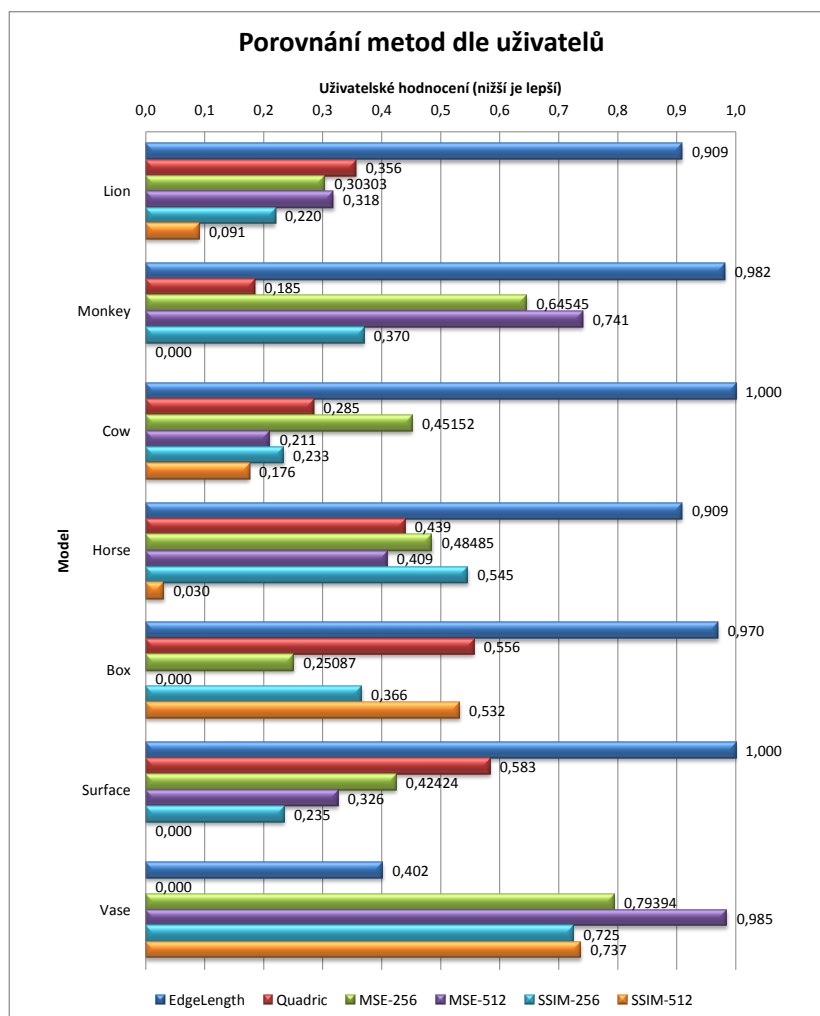
Obrázek A.1: Použité modely pro ověření metod simplifikace

B Čas simplifikace

	Metoda	EdgeLength	Quadric	MSE-256	MSE-512	SSIM-256
Lion	Čas prvotní fáze	00:00,056	00:00,058	00:33,006	01:45,756	01:38,383
	Čas simplifikační fáze	00:00,071	00:00,071	00:38,104	01:56,244	01:53,116
	Celkový čas	00:00,127	00:00,129	01:11,110	03:42,000	03:31,499
Monkey	Čas prvotní fáze	00:00,042	00:00,048	00:31,851	01:38,066	01:16,227
	Čas simplifikační fáze	00:00,080	00:00,085	00:41,710	02:10,843	01:24,688
	Celkový čas	00:00,122	00:00,133	01:13,561	03:48,909	02:40,915
Cow	Čas prvotní fáze	00:00,060	00:00,092	00:42,002	02:10,420	01:20,221
	Čas simplifikační fáze	00:00,099	00:00,118	00:53,176	02:48,931	01:35,511
	Celkový čas	00:00,159	00:00,210	01:35,178	04:59,351	02:55,732
Horse	Čas prvotní fáze	00:00,080	00:00,094	00:54,000	02:40,158	01:37,455
	Čas simplifikační fáze	00:00,150	00:00,132	01:11,524	03:36,195	02:04,108
	Celkový čas	00:00,230	00:00,226	02:05,524	06:16,353	03:41,563
Box	Čas prvotní fáze	00:00,043	00:00,009	00:08,122	00:26,052	01:14,214
	Čas simplifikační fáze	00:00,045	00:00,119	00:16,192	00:53,930	03:14,472
	Celkový čas	00:00,088	00:00,128	00:24,314	01:19,982	04:28,686
Surface	Čas prvotní fáze	00:00,072	00:00,048	00:33,014	01:49,922	01:17,476
	Čas simplifikační fáze	00:00,085	00:00,118	00:47,287	02:35,610	02:11,188
	Celkový čas	00:00,157	00:00,166	01:20,301	04:25,532	03:28,664
Vase	Čas prvotní fáze	00:00,089	00:00,094	00:55,018	03:08,745	02:46,897
	Čas simplifikační fáze	00:00,142	00:00,175	01:21,890	04:32,444	05:28,322
	Celkový čas	00:00,231	00:00,269	02:16,908	07:41,189	08:15,219

Obrázek B.1: Čas výpočtu metod [v minutách]

C Graf porovnání metod



Obrázek C.1: Graf porovnání metod z hlediska hodnocení uživatelů