

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

3D MESH A KVALITA TROJUHOĽNÍKOV
BAKALÁRSKA PRÁCA

2024

MARTIN KOŠOVSKÝ

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

3D MESH A KVALITA TROJUHOĽNÍKOV
BAKALÁRSKA PRÁCA

Študijný program: Informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra algebry a geometrie
Školiteľ: doc. RNDr. Andrej Ferko, PhD.
Konzultant: Mgr. Marcel Makovník, PhD.

Bratislava, 2024
Martin Košovský



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Martin Košovský
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: 3D mesh a kvalita trojuholníkov
3D Mesh and Triangle Quality

Anotácia: Po prehľade problematiky plánujeme vytvoriť statistický súbor dát o trojuholníkoch a navrhnuť vhodnú vizualizáciu vlastností digitálnych modelov vo vybraných databázach. Nakoniec vytvoríme reprezentatívny súbor údajov a prispějeme pluginom do projektu MeshLab.

Cieľ: Prehľad problematiky mier kvality a výber vhodných súborov podmieni špecifikáciu projektu. Návrh vhodnej vizualizácie alebo jej viacerých alternatív otestujeme na vybraných objektoch s cieľom rozšíriť funkčnosť metadát pre aplikácie najmä v digitálnom kultúrnom dedičstve. Dvojice objektov porovnáme pomocou referenčného nástroja Metro a našim prístupom.

Literatúra: BOTCH, M. et al. Polygonal Mesh Processing. Routledge 2010.
SKIENA, S. Data Science Design Manual. Springer 2017.

Kľúčové slová: 3D mesh, kvalita trojuholníkov, vizualizácia, MeshLab

Vedúci: doc. RNDr. Andrej Ferko, PhD.
Konzultant: Mgr. Marcel Makovník, PhD.
Katedra: FMFI.KAG - Katedra algebry a geometrie
Vedúci katedry: doc. RNDr. Pavel Chalmovianský, PhD.

Dátum zadania: 10.10.2023

Dátum schválenia: 16.10.2023
doc. RNDr. Dana Pardubská, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Čestné vyhlásenie:

Čestne vyhlasujem, že celú bakalársku prácu na tému „3D mesh a kvalita trojuholníkov“, vrátane všetkých jej príloh a obrázkov, som vypracoval samostatne, a to s použitím literatúry uvedenej v priloženom zozname a nástrojov umelej inteligencie. Vyhlasujem, že nástroje umelej inteligencie som použil v súlade s príslušnými právnymi predpismi, akademickými právami a slobodami, etickými a morálnymi zásadami za súčasného dodržania akademickej integrity a že ich použitie je v práci vhodným spôsobom označené.

Pod'akovanie:

Na tomto mieste by som chcel rád poďakovať všetkým, ktorí mi pomohli počas prípravy a písania tejto bakalárskej práce. Moje osobitné poďakovanie patrí môjmu vedúcemu práce doc. RNDr. Andrejovi Ferkovi, PhD., za jeho odborné vedenie. Ďalej by som sa rád poďakoval mojím rodičom, ktorý ma podporovali počas celej doby štúdia.

Abstrakt

Táto bakalárska práca sa zaoberá analýzou a vizualizáciou kvality trojuholníkových 3D sietí (meshov). Výsledkom našej práce je plugin pre open-source nástroj MeshLab, ktorý umožňuje efektívnu vizualizáciu rôznych kvalitatívnych metrík trojuholníkov. V práci sú podrobne preskúmané a analyzované existujúce metódy hodnotenia kvality trojuholníkov ako sú uhlová metrika, pomer strán, plošná metrika, metrika podielu polomerov vpísanej a opísanej kružnice a min-max výšková metrika. Okrem toho sme preskúmali rôzne typy vizualizácií, vrátane farebného mapovania a histogramov a obľúbené softvérové riešenia ako MeshLab, Blender a ParaView, ktoré sa využívajú na analýzu a vizualizáciu kvality 3D modelov. Na základe týchto poznatkov sme navrhli a následne implementovali nové vizualizačné techniky, ktoré zlepšujú možnosti analýzy a hodnotenia 3D modelov. Práca tiež zahŕňa konkrétne detaily implementácie a experimenty, ktoré demonštrujú funkčnosť a efektivitu navrhnutých riešení na vybraných 3D modeloch.

Kľúčové slová: 3D mesh, kvalita trojuholníkov, vizualizácia, MeshLab

Abstract

This bachelor thesis deals with the analysis and visualization of the quality of triangular 3D meshes. The result of our work is a plugin for the open-source tool MeshLab, which allows for efficient visualization of various qualitative metrics of triangles. The thesis thoroughly examines and analyzes existing methods of evaluating triangle quality such as the angle metric, aspect ratio, area metric, the radius ratio of the inscribed and circumscribed circles and the min-max height metric. In addition, we explored various types of visualizations, including color mapping and histograms and popular software solutions such as MeshLab, Blender and ParaView, which are used for the analysis and visualization of 3D model quality. Based on these insights, we designed and subsequently implemented new visualization techniques that enhance the capabilities of analysis and evaluation of 3D models. The thesis also includes specific implementation details and experiments that demonstrate the functionality and efficiency of the proposed solutions on selected 3D models.

Keywords: 3D mesh, triangle quality, visualization, MeshLab

Obsah

Úvod	1
1 Prehľad problematiky	3
1.1 Základné pojmy, kvalita meshu	3
1.2 Metriky kvality trojuholníkov	4
1.2.1 Uhlová metrika	4
1.2.2 Pomer strán	4
1.2.3 Plošná metrika	5
1.2.4 Metrika podiel polomerov	5
1.2.5 Min-Max výšková metrika	6
1.2.6 Zhrnutie metrik	6
1.3 Vizualizácie	7
1.3.1 Farebné mapovanie	7
1.3.2 Histogramy	7
1.3.3 Vizualizácia P. Pébay a T. Baker	8
1.3.4 Vizualizácia od André Guézieca	9
1.3.5 Kontúrové čiary	9
1.4 Existujúce softvérové riešenia	10
1.4.1 MeshLab – Quality Mapper	10
1.4.2 MerQuit	10
1.4.3 Blender	11
1.4.4 ParaView	12
1.5 Zhrnutie a sumarizácia	13
2 Špecifikácia projektu	15
2.1 Požiadavky	15
2.2 Opis navrhnutých vizualizácií	16
2.3 Používateľské rozhranie	18
2.4 Formáty vstupu a výstupu	19
2.4.1 Vstupné dáta	19
2.4.2 Výstupné dáta	20

2.5	Výber programovacieho jazyka	21
2.6	Formát pluginu a vloženie do Meshlabu	21
2.6.1	DLL formát	21
2.6.2	Predkompilovaný Plugin	22
2.6.3	Kompilácia pluginu zo zdrojového kódu	23
3	Implementácia a experimenty	25
3.1	Implementácia	25
3.1.1	Štruktúra pluginu	25
3.1.2	Funkcie potrebné pre vytvorenie pluginu	26
3.1.3	Farbenie hrán a vrcholov	27
3.1.4	Výpočet 1. metriky - Obsah trojuholníkov	28
3.1.5	Výpočet 2. metriky - Podiel polomerov	29
3.1.6	Výpočet 3. metriky - Uhlová metrika	29
3.1.7	Výpočet 4. metriky - Pomer strán	30
3.1.8	Výpočet 5. metriky - Min-Max výšková metrika	30
3.1.9	Kombinácia metrík	31
3.1.10	Výpočet uhlov pre farebné mapovanie vrcholov	32
3.2	Experimenty	33
3.2.1	Testovacie objekty	33
3.2.2	Časová efektívnosť	35
3.2.3	Príklady použitia rozšírenia	36
3.2.4	Nástroj Metro	39
	Záver	43
	Literatúra	46
	Príloha A	47

Zoznam obrázkov

1.1	Extrémne uhly	5
1.2	Metrika pomeru strán [4]	5
1.3	Metrika podiel polomerov [12],[4]	6
1.4	Min-Max výšková metrika	6
1.5	Vizualizácia podľa P. Pébay a T. Baker	8
1.6	A. Guézic, Vizualizácia č. 1	8
2.1	Kontextový diagram projektu	16
3.1	Pravidelný štvorsten	34
3.2	Stanford Bunny	34
3.3	Happy Buddha	34
3.4	Thai Statue	34
3.5	Vizualizácia kombinácie metrík na štvorstene	37
3.6	Vizualizácia kombinácie metrík na Stanford Bunny	37
3.7	Farebné mapovanie vrcholov na Stanford Bunny	38
3.8	Detailný pohľad na farebné mapovanie vrcholov	38
3.9	Výstup z filtra Hausdorff Distance	39
3.10	Upravený Stanford Bunny	39
3.11	Vizualizácia Happy Buddha	40
3.12	Výstup z Hausdorff Distance	40
3.13	Upravený Happy Buddha	40

Zoznam tabuliek

3.1	Čas vizualizácie v milisekundách pre model Stanford Bunny	36
3.2	Čas vizualizácie v milisekundách pre model Happy Buddha	36
3.3	Čas vizualizácie v sekundách pre model Thai Statue	36

Úvod

Vytvorenie presného a efektívneho 3D modelu je základným kameňom mnohých moderných technologických aplikácií, od počítačových hier, filmového priemyslu, virtuálnej reality, technického modelovania až po fyzikálne simulácie. V centre týchto pokročilých výtvorov ležia 3D siete (meshe), ktoré slúžia ako digitálna kostra pre reprezentáciu objektov v trojrozmernom priestore.

Kvalita jednotlivých trojuholníkov, z ktorých sú tieto 3D siete zložené, má priamy vplyv na vizuálnu pôsobivosť, výkonnosť a presnosť modelov. Preto je presná a intuitívna vizualizácia kvality týchto trojuholníkov dôležitá pre ich optimalizáciu a zlepšovanie vlastností celého modelu.

Cieľom tejto práce je preto navrhnúť a implementovať efektívne riešenie pre vizualizáciu vlastností digitálnych modelov, pričom prispejeme pluginom do projektu MeshLab.

V prvej kapitole poskytneme komplexný prehľad o kvalitatívnych metrikách a existujúcich softvérových riešeniach, ktoré sa zameriavajú na analýzu a hodnotenie 3D sietí. Táto kapitola nám umožní porozumieť rôznym prístupom a technológiám, ktoré sa v súčasnosti používajú na posudzovanie kvality meshov.

Druhá kapitola sa venuje špecifikácii nášho projektu a poskytuje pohľad na technické aspekty, ktoré sú s projektom spojené. To zahŕňa opis vstupných a výstupných dát, definovania požiadaviek na softvér a charakteristiky používateľského prostredia.

Posledná kapitola sa zameria na opis implementácie pluginu. Vysvetlíme, z akých komponentov je zostavený, aké funkcie obsahuje a aké prístupy boli použité na manipuláciu a analýzu meshu. Na záver tejto kapitoly budú predstavené experimenty, ktoré demonštrujú funkčnosť nášho riešenia a overujú jeho efektívnosť v praxi na vybraných 3D modeloch. Tieto experimenty nám poskytnú cenné informácie o vplyve našej práce na zlepšenie kvality a vizualizácie digitálnych modelov.

Kapitola 1

Prehľad problematiky

V tejto časti opíšeme problematiku 3D sietí (meshov), kvalít trojuholníkov a vizualizácií, ktoré sú základnými stavebnými kameňmi v digitálnom modelovaní a simuláciách. Rozoberieme základné pojmy nevyhnutné pre pochopenie širšieho kontextu problematiky, ako aj rôzne miery kvality, ktoré umožňujú efektívne hodnotenie a optimalizáciu 3D modelov. Táto kapitola nám poskytne pevný základ, na ktorom budeme stavať naše ďalšie poznatky.

1.1 Základné pojmy, kvalita meshu

Na začiatok by bolo vhodné, aby sme definovali a upresnili pojmy k názvu práce.

- 3D mesh alebo inak povedané 3D sieť alebo pletivo alebo nepravidelná sieť trojuholníkov (TIN) je štruktúrna stavba 3D modelu, ktorá sa skladá z polygónov. V 3D počítačovej grafike a modelovaní pevných telies je polygónová sieť súhrnom vrcholov, hrán a stien, ktoré definujú tvar mnohostenného objektu. Tieto steny sa zvyčajne skladajú z trojuholníkov (trojuholníková sieť), štvoruholníkov alebo iných jednoduchých konvexných polygónov [3], [9].
- Miera kvality trojuholníka je jednoznačné číselné ohodnotenie jeho tvaru podľa vybranej vlastnosti. Pod týmto pojmom si môžeme predstaviť funkciu, ktorá má na vstupe trojuholník a na výstupe je to väčšinou jednorozmerné reálne číslo z intervalu $[0, 1]$ [26].
- Vizualizácia kvality trojuholníkov v 3D meshi je proces zobrazovania meshu takým spôsobom, ktorý umožňuje ľahko identifikovať a hodnotiť geometrickú kvalitu jednotlivých trojuholníkov. Vizualizácia by mala byť navrhnutá tak, aby poskytovala rýchly a intuitívny prehľad o kvalite meshu.

1.2 Metriky kvality trojuholníkov

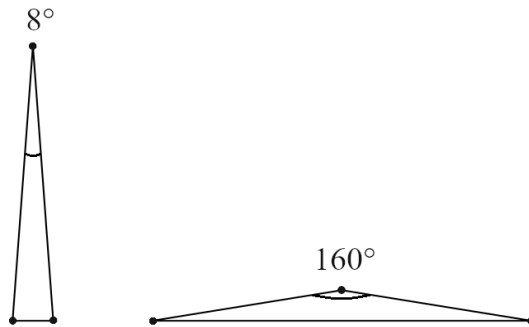
V nasledujúcej časti sa budeme venovať podrobnejšiemu preskúmaniu rôznych metód a prístupov, používaných pri hodnotení kvality trojuholníkových sietí, čiže meshov. Táto sekcia bude slúžiť ako teoretický základ, ktorý nám pomôže pochopiť, akými spôsobmi sa analyzuje a hodnotí kvalita trojuholníkov z hľadiska ich geometrických charakteristík. Miery kvality sú nevyhnutné pre pochopenie, ako variácie v geometrii trojuholníkov môžu mať významný dopad na presnosť, efektívnosť a vizuálnu pôsobivosť 3D modelov.

1.2.1 Uhlová metrika

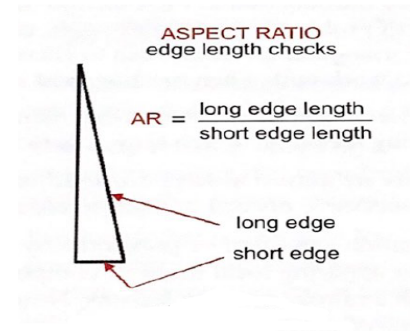
Uhlová metrika predstavuje základný nástroj v arzenáli techník posudzovania kvality trojuholníkov v rámci 3D modelovania. Zameriava sa na analýzu uhlov vnútri trojuholníka, je nevyhnutná pre identifikáciu a korekciu extrémnych uhlov – buď príliš ostrých alebo príliš tupých (obr. 1.1). Tieto extrémny môžu negatívne ovplyvniť numerickú stabilitu, efektívnosť a presnosť výpočtov, ako aj celkovú kvalitu vizualizácie. Ideálny trojuholník [21] by mal mať všetky tri uhly čo najbližšie k 60 stupňom rovnostranného trojuholníka. Takáto konfigurácia je považovaná za optimálnu, pretože umožňuje napríklad rovnomernú distribúciu napätia po celom meshi a minimalizuje riziko deformácií. Vďaka tomu sú trojuholníky s uhlami blízkymi 60° ideálne pre aplikácie, kde je dôležitá vysoká úroveň geometrickej presnosti a numerickej spoľahlivosti, vrátane pokročilých mechanických simulácií a presných fyzikálnych analýz [15].

1.2.2 Pomer strán

Metrika pomeru strán hodnotí relatívnu dĺžku strán trojuholníka, čím poskytuje informácie o jeho proporčnosti. Pomer strán je definovaný ako podiel dĺžky najdlhšej strany trojuholníka a dĺžky jeho najkratšej strany (obr. 1.2). Perfektný pomer, 1:1, naznačuje, že všetky strany trojuholníka sú identické, čo korešponduje s rovnostranným trojuholníkom. Trojuholníky s pomerom strán blízkym k 1 majú tendenciu byť geometricky stabilnejšie a poskytujú lepšiu kvalitu meshu pre vizualizácie a simulácie. Extrémne hodnoty pomeru strán, či už vysoké alebo nízke, môžu signalizovať, že trojuholník je neprimerane štíhly a úzky, čo by mohlo viesť k numerickým nestabilitám a zníženej presnosti simulácií. Trojuholníky s vysokým pomerom strán môžu viesť k vizuálnym artefaktom, keďže extrémne dlhé alebo krátke strany môžu spôsobiť nežiadúce efekty pri mapovaní textúr. Pomer strán je kritický pre zachovanie detailov a minimalizáciu artefaktov, najmä v aplikáciách ako 3D tlač, kde presnosť geometrie priamo ovplyvňuje kvalitu výsledného produktu [4], [22].



Obr. 1.1: Extrémne uhly



Obr. 1.2: Metrika pomeru strán [4]

1.2.3 Plošná metrika

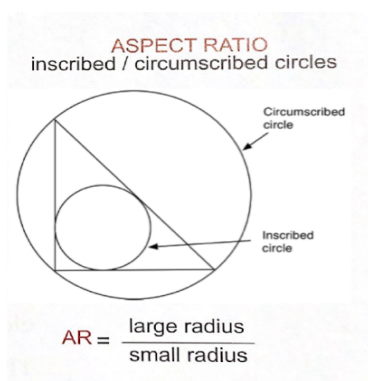
Plošná metrika predstavuje dôležitý nástroj pri hodnotení a optimalizácii 3D meshov, pretože sa zameriava na rozdiely v plochách jednotlivých trojuholníkov. Plošná metrika hodnotí rozdiel veľkosti plôch medzi trojuholníkmi v mriežke. Táto metrika je zásadná pre zabezpečenie, že distribúcia plochy medzi trojuholníkmi bola čo najrovnomernejšia, čo prispieva k uniformnej vizualizácii a presnosti modelov. Ideálny mesh by mal obsahovať trojuholníky s podobnými plochami, čím sa minimalizujú lokálne variácie hustoty meshu a zabezpečuje sa lepšia aproximácia povrchu modelovaného objektu. Rovnomerné rozloženie plochy má viacero výhod, vrátane zlepšenej estetickej kvality vizualizácií a zvýšenej efektivity pri simuláciách. V prípade, že sú plochy trojuholníkov v meshi príliš rozdielne, môže to viesť k neuniformnej detailnosti, kde niektoré oblasti môžu byť nadmerne detailné v porovnaní s inými, menej detailnými oblasťami. Táto metrika je obzvlášť relevantná v aplikáciách s vysokou úrovňou detailov, ako sú digitálne prototypy a komplexné vizuálne efekty [1].

1.2.4 Metrika podiel polomerov

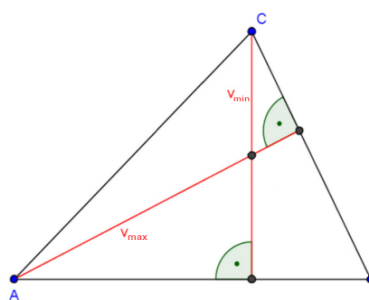
Metrika vpísanej a opísanej kružnice poskytuje dôležitý uhol pohľadu na geometrickú rovnomernosť a proporčnosť. Táto metrika porovnáva pomer medzi polermi opísanej kružnice (kružnicou obklopujúcou trojuholník) a vpísanej kružnice (kružnica dotýkajúca sa všetkých strán trojuholníka zvnútra), čím sa zisťuje, do akej miery sa trojuholníky blížia k ideálnemu, rovnostrannému tvaru (obr. 1.3). Malý pomer medzi týmito kružnicami signalizuje vyššiu kvalitu trojuholníka, keďže indikuje lepšiu rovnomernosť a minimalizuje riziko extrémnych uhlov alebo deformácií, ktoré by mohli ovplyvniť vizuálnu kvalitu. Táto metrika je nevyhnutná pre zabezpečenie, že meshe sú optimálne pre vizuálne prezentácie a numerické simulácie [4], [19].

1.2.5 Min-Max výšková metrika

Min-Max Výšková Metrika je metrika kvality trojuholníkov, ktorá sa zameriava na meranie extrémov výšok. Táto metrika porovnáva najmenšiu a najväčšiu výšku v rámci jednotlivých trojuholníkov v 3D sieti, čím poskytuje dôležité informácie o ich proporcionálnosti. Min-Max výšková metrika vypočítava pomer medzi najkratšou a najdlhšou výškou trojuholníka (obr. 1.4). Ideálny trojuholník podľa tejto metriky by mal mať malý rozdiel medzi týmito výškami, čo naznačuje, že je dobre proporcionálny a nemá tendenciu k deformácii. Táto metrika je zásadná pre identifikáciu trojuholníkov, ktoré majú potenciál byť príliš ploché [11], [4].



Obr. 1.3: Metrika podiel polomerov [12],[4]



Obr. 1.4: Min-Max výšková metrika

1.2.6 Zhrnutie metrick

V tejto časti sme sa venovali rozličným metrikám kvality trojuholníkových sietí. Analyzovali sme uhlovú metriku, pomer strán, plošnú metriku, metriku vpísanej a opísanej kružnice, a min-max výškovú metriku. Každá z týchto metrick má svoje špecifické výhody a použitie v závislosti od potrieb a požiadaviek projektov. Uhlová metrika nám umožňuje identifikovať a upraviť extrémne uhly, čím zvyšujeme numerickú stabilitu a vizuálnu pôsobivosť. Pomer strán je zase kritický pre zabezpečenie geometrickej stability a kvality meshu. Plošná metrika zabezpečuje rovnomerné rozloženie plochy medzi trojuholníkmi, čo má priamy vplyv na kvalitu vizualizácie a efektivitu simulácií. Metrika podiel polomerov nám poskytuje informácie o geometrickom usporiadaní trojuholníkov a ich priblížení k ideálnemu tvaru. Min-Max výšková metrika sa zameriava na proporčnosť trojuholníkov a pomáha identifikovať tie, ktoré by mohli byť príliš ploché, čo by mohlo viesť k deformáciám. Vďaka porozumeniu a správne mu využívaniu týchto metrick môžu vývojári a dizajnéri dosiahnuť lepšie výsledky pri tvorbe a analýze 3D modelov.

1.3 Vizualizácie

Vizualizácia je proces, ktorý umožňuje premeniť abstraktné dáta a informácie na grafické alebo vizuálne reprezentácie, čím výrazne zjednodušuje ich pochopenie a umožňuje hlbšiu analýzu. Tento proces je neoddeliteľnou súčasťou mnohých odborov, keďže umožňuje vizuálne predstaviť komplexné koncepty a modely v intuitívnej forme. Vizualizácia nachádza uplatnenie v rozličných oblastiach, vrátane vedeckého výskumu, kde pomáha objasňovať komplikované vedecké fenomény, v inžinierstve pre návrh a simuláciu nových technológií, v medicíne pri vizualizácii anatomických štruktúr alebo v architektúre a urbanizme, kde uľahčuje plánovanie a dizajn budov. Vizualizácia využíva pokročilé grafické a vizualizačné techniky na zobrazenie dátových súborov alebo matematických modelov v intuitívnej, vizuálne prístupnej forme, čím otvára dvere k novým objavom a pochopeniam v rámci daných oblastí. V neposlednom rade hrá kľúčovú rolu aj v digitálnom umení a zábavnom priemysle, kde prispieva k tvorbe pútavých vizuálnych efektov a zážitkov [5], [7].

1.3.1 Farebné mapovanie

Farebné mapovanie (color mapping) je technika používaná v rámci vizualizácie dát na grafické zobrazenie numerických hodnôt prostredníctvom farieb s cieľom zjednodušiť ich interpretáciu. Táto metóda je efektívna pri vizualizácii komplexných dátových súborov, ako sú 3D modely alebo vedecké dáta, pretože umožňuje rýchlo a intuitívne odozdávať informácie. V kontexte vizualizácie kvality 3D meshov je farebné mapovanie užitočné, pretože poskytuje okamžitú vizuálnu spätnú väzbu o rôznych metrikách kvality modelu, ako sú napríklad pomer strán, uhlová kvalita, hustota meshu a ďalšie. Farebné mapovanie funguje priradením rozsahu farieb k rozsahu numerických hodnôt. Typicky sa používajú farebné stupnice od teplých farieb (červená, oranžová) pre vyššie hodnoty, po studené farby (modrá, zelená) pre nižšie hodnoty. To uľahčuje rozlíšenie medzi rôznymi úrovňami kvality. Farebné mapovanie nie je len o estetike. Je to kľúčový nástroj pre efektívnu vizuálnu komunikáciu informácií. Vo vedeckých a inžinierskych aplikáciách pomáha rýchlo identifikovať oblasti záujmu alebo problémové oblasti, čím zvyšuje efektívnosť pri analýze a revízii komplexných 3D modelov a dátových súborov [20].

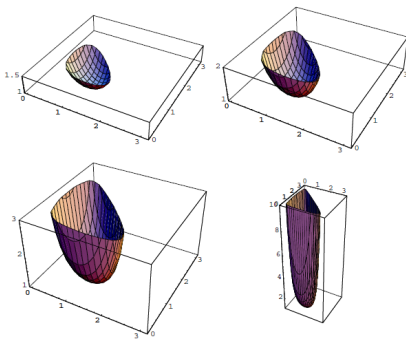
1.3.2 Histogramy

Histogramy sú nástroje na vizualizáciu, ktoré efektívne sumarizujú distribúciu dát pomocou súvislých stĺpcov, ktoré reprezentujú frekvenciu výskytu rôznych rozsahov hodnôt v datasete. V kontexte analýzy kvality 3D meshov, histogramy sú užitočné na ilustráciu, ako sú rozložené geometrické a kvalitatívne metriky cez celý mesh. Tento

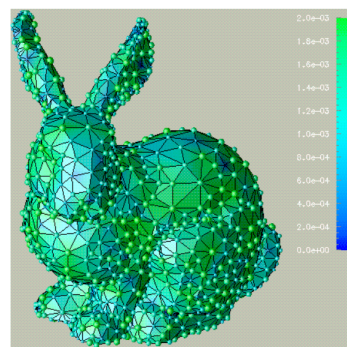
spôsob vizualizácie môže poskytnúť cenné informácie o všeobecnej kvalite meshu a pomôcť pri identifikácii oblastí, ktoré vyžadujú zlepšenie. Histogramy taktiež môžu ukazovať rozdelenie rôznych metrík kvality meshu, ako sú pomer strán, skosenosť, minimálne alebo maximálne uhly, hustota meshu, atď. Zobrazenie týchto metrík vo forme histogramu umožňuje rýchlo zistiť, či veľká časť meshu spĺňa určité požiadavky kvality alebo či existujú významné výkyvy, ktoré by mohli naznačovať výskyt problematických oblastí, ktoré vyžadujú pozornosť. Taktiež teda môžu byť použité na porovnanie kvality meshov pred a po aplikácii rôznych opravných alebo optimalizačných techník. Týmto spôsobom môžeme vizuálne zhodnotiť účinnosť zvolených metód na zlepšenie kvality meshu. Pre aplikácie, kde je rovnomerné rozloženie prvkov meshu žiaduce (napríklad v simulačných modeloch), histogramy môžu poslúžiť na overenie, či sú elementy meshu distribuované rovnomerne. V konečnom dôsledku sú histogramy silné nástroje pre analýzu a vizualizáciu kvality meshov vďaka svojej schopnosti poskytovať rýchle a vizuálne zrozumiteľné prehľady distribúcie dát [6], [20],[2].

1.3.3 Vizualizácia P. Pébay a T. Baker

Philippe Pébay a Timothy Baker sa zaoberajú vizualizáciou vzťahu medzi veľkosťou pomeru opísanej a vpísanej kružnice a veľkosťou najmenšieho a najväčšieho uhla v trojuholníku. Tento vzťah prezentujú prostredníctvom grafu, kde os x predstavuje hodnotu najmenšieho uhla, os y hodnotu najväčšieho uhla a funkčná hodnota, ktorá je vyjadrená pozdĺž osi z, ilustruje veľkosť pomeru medzi opísanou a vpísanou kružnicou (obr. 1.5). Táto vizualizácia je primárne určená na demonštráciu a poskytuje názorný príklad spôsobu, akým môže byť geometrická kvalita trojuholníkov vyjadrená graficky, avšak s obmedzenou praktickou aplikáciou. Rozširuje pochopenie o tom, ako geometrické charakteristiky trojuholníkov ovplyvňujú ich kvalitu a poukazuje na dôležitosť vizualizácie pri analýze a optimalizácii 3D modelov [19].



Obr. 1.5: Vizualizácia podľa P. Pébay a T. Baker



Obr. 1.6: A. Guéziec, Vizualizácia č. 1

1.3.4 Vizualizacia od André Guéziaca

André Guéziec priniesol do oblasti počítačovej grafiky a 3D modelovania dve pokročilé vizualizačné metódy na hodnotenie a prezentáciu kvality trojuholníkov v 3D modeloch, ktoré poskytujú inovatívny pohľad na analýzu a optimalizáciu meshov:

1. Farebné kódovanie a sférické značky:

Táto metóda zobrazuje 3D model, kde každý trojuholník je farebne označený na základe hodnoty chyby v objeme, čo umožňuje rýchlu vizuálnu identifikáciu oblastí s potenciálnymi problémami (obr. 1.6). Okrem toho sú sférické značky s polomerom proporcionálnym k veľkosti chyby umiestnené na vrcholoch trojuholníkov, čím poskytujú vizuálnu spätnú väzbu o topologických nepresnostiach. Tento prístup umožňuje intuitívne rozpoznanie a hodnotenie kvality meshu.

2. Histogramová analýza:

Druhá metóda predstavuje analýzu prostredníctvom histogramu s predom definovanými hranicami. Vytvára "vejár" trojuholníkov, ktorý umožňuje skúmanie, ako posunutie vrcholov pozdĺž priamky s rôznymi stupňami ovplyvňuje kvalitu. Histogram pred a po znížení počtu trojuholníkov poskytuje prehľad o vplyve redukcie na kvalitu meshu. Tento prístup umožňuje hlbšie pochopenie vplyvu geometrických zmien na kvalitu meshu.

Guéziecove vizualizačné metódy predstavujú významný prínos v oblasti počítačovej grafiky a modelovania, poskytujúc nástroje pre efektívnejšie a presnejšie hodnotenie kvality 3D meshov. Ich aplikácia v praxi umožňuje dizajnérom a inžinierom identifikovať a riešiť problémy s meshmi skôr v procese dizajnu a vývoja, čím sa zvyšuje celková kvalita a efektivita 3D modelovania [10], [26].

1.3.5 Kontúrové čiary

Kontúrové čiary, alebo izolínie, sa často používajú v rôznych oblastiach vizualizácie dát na zobrazenie konštantných hodnôt pre určitú oblasť alebo povrch. Bežne sa používajú v geografii, meteorológii a iných disciplínach. V kontexte 3D modelovania a analýzy meshov kontúrové línie poskytujú grafické znázornenie rovnakých hodnôt na povrchu meshu. Kontúrové línie môžu byť použité na zobrazenie geometrických vlastností meshu, ako sú výškové úrovne alebo uhlové sklonenia. To pomáha identifikovať oblasti s potenciálnymi problémami, ako sú prudké zmeny v geometrii. Vďaka schopnosti zobrazovať konštantné hodnoty cez mesh, kontúrové línie umožňujú ľahko detekovať oblasti, ktoré potrebujú zvláštnu pozornosť, napríklad oblasti s nedostatočnou hustotou meshu alebo s nevhodnou topológiou. Izolínie teda poskytujú priamy spôsob na pochopenie dátových charakteristík rozptýlených po trojrozmerných povrchoch [13], [24], [21].

1.4 Existujúce softvérové riešenia

V dnešnej dobe je k dispozícii viacero softvérových riešení zameraných na hodnotenie a vizualizáciu kvality 3D meshov. Tieto nástroje ponúkajú rôzne techniky a metódy, ktoré sa líšia vo svojej unikátnosti, presnosti a efektívnosti. Umožňujú tak používateľom efektívne sa prispôbiť špecifickým požiadavkám ich projektov. Každé z týchto riešení prináša inovatívne prístupy k analýze a optimalizácii meshov, či už prostredníctvom pokročilých algoritmov na detekciu a opravu geometrických nedostatkov, alebo cez sofistikované vizualizačné techniky, ktoré umožňujú hlbšie porozumenie štruktúre a kvalite 3D modelov.

1.4.1 MeshLab – Quality Mapper

MeshLab je open-source softvér, ktorý je špeciálne navrhnutý na spracovanie a editovanie 3D triangulovaných sietí (meshov) a bodových mračien. MeshLab sa široko využíva v oblastiach ako sú 3D tlač, 3D skenovanie a počítačová grafika. Jednou z kľúčových vlastností MeshLabu je jeho schopnosť vizualizovať a analyzovať kvalitu meshov. Quality Mapper v programe MeshLab je sofistikovaný nástroj, ktorý umožňuje vizualizáciu kvality 3D meshov prostredníctvom farebného mapovania. Tento nástroj priraďuje farby vrcholom/plochám trojuholníkov na základe vybranej miery kvality, ako sú vzdialenosti od každého vrcholu k najbližšej hranici (k najbližšej hrane alebo hranici siete), plochy trojuholníkov, euklidovskej vzdialenosti vrcholov od zvoleného bodu a rôznymi ďalšími. Takéto farebné kódovanie poskytuje rýchly a efektívny spôsob na vizuálnu identifikáciu oblastí s potenciálnymi problémami a na posúdenie celkovej uniformity a kvality meshu. Používaním Quality Mapperu môžu používatelia rýchlo rozpoznať problematické oblasti v meshi, ako sú oblasti s nízkou hustotou vrcholov, oblasti s veľkými plochami, čo umožňuje používateľom zamerať sa na ich zlepšenie. Tento nástroj je neoceniteľný pre procesy, ako sú zjednodušovanie meshov, analýza kvality a príprava modelov na 3D tlač alebo iné špecifické aplikácie. Vďaka svojej flexibilita a širokému spektru metrík, ktoré Quality Mapper podporuje, sa tento nástroj stal neoddeliteľnou súčasťou pracovného toku mnohých profesionálov a výskumníkov v oblasti počítačovej grafiky, digitálnej fabrikácie a ďalších [20], [16].

1.4.2 MerQuit

MerQuit je softvér určený pre analýzu a vizualizáciu kvality nepravidelných trojuholníkových sietí. Tento program, ktorý bol vyvinutý ako súčasť diplomovej práce na Univerzite Komenského v Bratislave, je implementovaný v jazyku C# s využitím rozsiahlej knižnice OpenGL a umožňuje používateľom v krátkom čase vypočítať rôzne miery kvality pre zadaný 3D model. Ako vstupný formát sa používa štandardný ply

formát. Výstupom programu je vizualizácia vybraných mier kvality v špeciálnom 3D grafe, ktorý zobrazuje súradnicové osi euklidovského trojrozmerného priestoru. Základným prvkom tohto grafu je bod so súradnicami (x, y, z) , pričom (x, y) predstavuje bod nachádzajúci sa oproti najdlhšej strane trojuholníka (alebo vybraný špeciálny bod trojuholníka) a jednorozmernej miery z , ktorá odpovedá meraniu vybranému používateľom. Vďaka možnosti farebného označenia bodov podľa ich hustoty a toho, či spĺňajú určené uhlové limity nastavené používateľom, graf poskytuje trojrozmerný prehľad o celkovej kvalite zadaného meshu. Tento trojrozmerný pohľad umožňuje používateľom lepšie zhodnotiť a porozumieť kvalite a geometrickým vlastnostiam meshu a taktiež identifikovať problematické oblasti v používateľových modeloch [26].

1.4.3 Blender

Blender je výkonný open-source nástroj pre 3D modelovanie, animáciu, renderovanie, post-produkciu a herný dizajn, ktorý tiež ponúka rôzne nástroje na analýzu a vizualizáciu kvality meshov. Tieto nástroje sú zvlášť užitočné pre 3D umelcov a dizajnérov pri kontrole a zlepšovaní ich modelov. V Blenderi sa vizualizácia kvality meshov zameriava hlavne na prípravu modelov pre 3D tlač a poskytuje používateľom nástroje na identifikáciu a diagnostiku potenciálnych problémov, ktoré by mohli ovplyvniť kvalitu a realizovateľnosť tlače. Tieto nástroje sú prístupné v Edit Mode a v Solid Viewport shadingu, čo umožňuje podrobnú analýzu v reálnom čase. Blender poskytuje nasledovné vizualizačné režimy:

1. Overhang (Previs):

Tento režim zobrazuje oblasti s previsom, ktoré môžu byť problémové pre extrúzne 3D tlačiarne kvôli ich limitom na maximálny uhol previsu. Používatelia môžu nastaviť rozsah uhlov a os, na základe ktorých sa vyhodnocuje previs. Tieto oblasti sú následne farebne označené v závislosti od ich hodnôt.

2. Thickness (Hrúbka):

Testuje a zobrazuje oblasti meshu, ktoré majú obmedzenú stenu hrúbky, ktoré nemusia byť tlačiteľné vzhľadom na technické obmedzenia tlačiarne. Tento test používa ray casting na určenie hrúbky geometrie s možnosťou nastavenia minimálnej a maximálnej hrúbky a počtu vzoriek pre výpočet.

3. Intersections (Pretnutia):

Identifikuje pretnutia medzi povrchmi, kde vnútornú a vonkajšiu stranu modelu nie je možné spoľahlivo rozoznať. Blender zobrazuje tieto prekrývania ako kritické oblasti, kde môže dôjsť k chybám pri tlači.

4. Distortion (Deformácie):

Zobrazuje časti 3D meshu, ktoré sú deformované. Triangulácia deformovaných častí nie je jednoznačná. Deformácia sa meria na základe stien, ktoré nie sú rovné, pričom časti steny smerujú rôznymi smermi.

5. Sharp Edges (Ostré hrany):

Podobne ako hrúbka stien, ostré hrany môžu vytvoriť tvary, ktoré sú príliš tenké na to, aby boli tlačiteľné. Tento režim umožňuje zobrazenie oblastí s ostrými hranami na základe nastaveného rozsahu uhlov.

Tieto nástroje a funkcie umožňujú používateľom Blendera efektívne diagnostikovať a opravovať bežné problémy s meshmi. Sú významné hlavne pre tých, ktorí pracujú na 3D modeloch určených pre tlač, keďže umožňujú používateľom identifikovať a riešiť potenciálne problémy skôr, než pristúpia k samotnej tlači [17].

1.4.4 ParaView

ParaView je open-source nástroj pre vizualizáciu vedeckých dát, ktorý sa často používa na analýzu a vizualizáciu výsledkov z výpočtových simulácií vrátane tých, ktoré zahŕňajú 3D meshe. ParaView využíva VTK (Visualization Toolkit) pre spracovanie a zobrazenie dát, čo umožňuje detailnú analýzu a hodnotenie kvality 3D meshov. Tento softvér je vybavený viacerými nástrojmi na analýzu kvality meshu, ktoré sú kľúčové pre rôzne inžinierske a vedecké aplikácie. Jeho schopnosť analyzovať a vizualizovať geometrické a topologické vlastnosti meshov umožňuje používateľom identifikovať a riešiť potenciálne problémy vstupného 3D meshu, čo zvyšuje efektívnosť simulácií a analýz. ParaView ponúka nasledujúce nástroje a metódy na analýzu meshu:

1. Vizualizácia problémových oblastí:

ParaView umožňuje vizualizáciu rôznych charakteristík meshu, ako sú skosenosť, pomer strán alebo uhlová kvalita jednotlivých elementov. Používatelia môžu ľahko identifikovať oblasti, kde mesh nespĺňa určité kritériá kvality, čo je kľúčové pre zabezpečenie stability a presnosti v simuláciách.

2. Vizualizácia metrick kvality:

ParaView ponúka rozsiahle možnosti výberu a nastavení typov vizualizácií, ktoré umožňujú zobraziť kvalitatívne hodnoty meshu na základe rôznych kritérií, ako je napríklad pomer vpísanej a opísanej kružnice, plošná metrika a iné. Tieto metriky sú zobrazené väčšinou pomocou farebných prelnajúcich sa škál, kde rôzne farby indikujú rôzne úrovne kvality.

3. Analýza zložitosti meshu:

ParaView môže analyzovať a zobraziť zložitosť meshu, čo pomáha používateľom pochopiť, ako hustota alebo zložitosť meshu ovplyvňuje výpočtovú náročnosť a efektívnosť simulácií. Toto je obzvlášť užitočné pri optimalizácii meshov pre vysoko výkonné výpočty.

4. Interaktívne nástroje na úpravu meshu:

ParaView poskytuje interaktívne nástroje na úpravu meshu, ktoré umožňujú používateľom v reálnom čase upravovať a optimalizovať ich meshe na základe výsledkov analýzy.

5. Použitie filtrov na identifikáciu problémov:

ParaView obsahuje rozsiahly súbor filtrov, ktoré môžu byť použité na identifikáciu a riešenie špecifických problémov s meshmi. Tieto filtre umožňujú používateľom analyzovať meshe na základe rôznych kritérií, vrátane geometrických defektov, nekonzistentnej hustoty meshu, alebo nevhodných topologických štruktúr.

Softvér ParaView teda poskytuje robustné riešenie pre analýzu kvality meshu, ktoré sa dá efektívne využiť vo viacerých technických a vedeckých aplikáciách. Jeho schopnosť spracovávať veľké súbory a komplexné dátové štruktúry, spolu s možnosťou individuálnych prispôbení a rozširovania funkcionalít, robí z ParaView významný nástroj pre každého, kto potrebuje detailne analyzovať a vizualizovať 3D meshe [8], [18].

1.5 Zhrnutie a sumarizácia

V tejto kapitole sme sa zamerali na problematiku 3D sietí (meshov), kvality trojuholníkov a vizualizácií, ktoré sú kľúčové v digitálnom modelovaní a simuláciách.

Najprv sme definovali 3D mesh ako štruktúru zloženú z polygónov, najčastejšie trojuholníkov, ktorá slúži na reprezentáciu trojrozmerných objektov. Zdôraznili sme, že kvalita jednotlivých trojuholníkov priamo ovplyvňuje vizuálnu pôsobivosť, výkonnosť a presnosť modelov, a preto je dôležité mať presné a intuitívne nástroje na vizualizáciu týchto kvalít.

Analyzovali sme rôzne metriky kvality trojuholníkov vrátane uhlovej metriky, pomeru strán, plošnej metriky, metriky podielu polomerov vpísanej a opísanej kružnice a min-max výškovej metriky. Každá z týchto metrík má svoje špecifické výhody a použitie, ktoré nám umožňujú identifikovať a korigovať geometrické nedostatky trojuholníkov.

Okrem toho sme sa zamerali na metódy vizualizácie, ako sú farebné mapovanie a histogramy. Farebné mapovanie sa nám celkovo pozdáva ako veľmi efektívna metóda vizualizácie, pretože umožňuje rýchlu identifikáciu problémových oblastí v meshi.

Preskúmali sme aj existujúce softvérové riešenia, ako sú MeshLab, Blender a ParaView, ktoré ponúkajú rôzne nástroje na vizualizáciu a analýzu kvality 3D modelov. Celkovo sa nám pozdáva možnosť použitia filtrov na rozšírenie funkcionality softvérových nástrojov, ako to demonštrujú MeshLab a ParaView.

Všetky napísané definície, detailne analyzované metriky a opísané softvérové riešenia nám poskytli cenné poznatky, ktoré sme následne využili pri návrhu a implementácii nášho pluginu pre MeshLab.

Kapitola 2

Špecifikácia projektu

Kapitola Špecifikácia projektu poskytuje podrobný prehľad o vstupných a výstupných dátach, ako aj o technických a funkčných požiadavkách projektu. Táto časť je dôležitá pre porozumenie, ako je náš plugin MetricCombVis navrhnutý a ako bude fungovať.

2.1 Požiadavky

Hlavným cieľom nášho projektu je navrhnuť a implementovať plugin do MeshLabu, ktorý umožní vizualizáciu vlastností digitálnych modelov. Naše softvérové dielo je určené hlavne pre grafikov, ktorí potrebujú rýchlo a efektívne posudzovať kvalitu 3D modelov alebo ich porovnať. Plugin bude poskytovať používateľom nástroje na efektívnu analýzu a vizualizáciu kvalitatívnych parametrov 3D modelov. Zároveň je navrhnutý tak, aby splnil tri zásadné požiadavky, ktoré sú dôležité pre efektívne spracovanie a analýzu 3D meshov:

1. Detekcia a vizualizácia kritických hodnôt:

Plugin bude schopný identifikovať kritické hodnoty geometrických a štrukturálnych vlastností meshov, ktoré by mohli indikovať potenciálne problémy s ich kvalitou alebo stabilitou. Na základe zistených hodnôt budú následne intuitívne zobrazené prislúchajúce problémové oblasti, aby používatelia mohli rýchlo identifikovať a zaoberať sa týmito oblasťami vo svojich 3D modeloch.

2. Optimalizácia výpočtových metód:

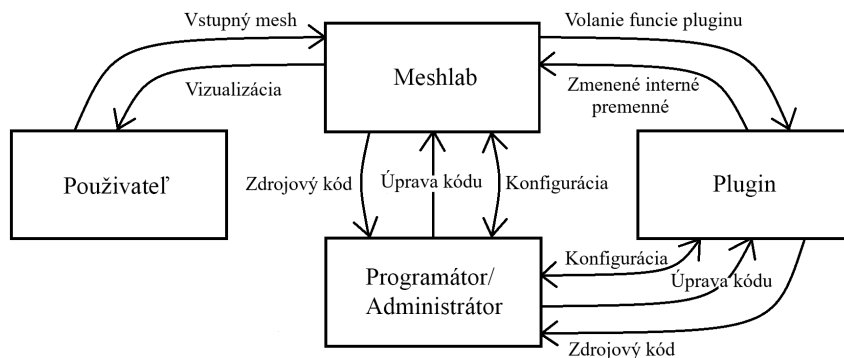
S ohľadom na často veľké objemy dát a zložitosť 3D modelov, je dôležité, aby boli výpočtové procesy čo najefektívnejšie. Preto bude tiež jedným z hlavných cieľov nášho pluginu implementácia efektívnych a rýchlych algoritmov pre výpočty kvality. Implementácia efektívnych algoritmov zaručí, že analýza sieťových štruktúr prebehne veľkou rýchlosťou. To poskytne používateľom možnosť pra-

covať s maximálnou efektivitou a zaručí optimálnu odozvu prostredia aplikácie MeshLab.

3. Vizualizácia celkovej kvality 3D modelu:

Plugin poskytne možnosti pre hodnotenie celkového stavu meshu z hľadiska vybranej kvalitatívnej metriky. To zahŕňa vizualizáciu rozloženia kvalitatívnych metrik na povrchu objektu, čo pomôže používateľom získať rýchly prehľad o celkovej kvalite a uniformite meshu. Týmto spôsobom plugin poskytne používateľom rýchly prehľad o kvalite celého meshu, čo je obzvlášť užitočné pri práci s rozsiahlymi alebo komplexnými 3D modelmi.

Hlavným cieľom nášho softvéru bude teda poskytnúť používateľom MeshLab pokročilé nástroje pre vizualizáciu a analýzu kvality 3D modelov, ktoré umožnia lepšie pochopenie charakteristík modelov. Používatelia budú môcť interaktívne prehliadať modely, meniť vizualizačné nastavenia. Softvér bude taktiež podporovať import dát z rôznych formátov a bude navrhnutý tak, aby bol intuitívny a ľahko použiteľný aj pre menej skúsených používateľov. Na obrázku 2.1 môžeme vidieť diagram nášho projektu. Tento diagram ilustruje, ako MeshLab komunikuje s používateľom a programátorom/administrátorom, a tiež ako prebieha interakcia medzi MeshLabom a jeho pluginom.



Obr. 2.1: Kontextový diagram projektu

2.2 Opis navrhnutých vizualizácií

V tejto časti sa zameriame na popis rôznych vizualizačných techník, ktoré sme navrhli a následne aj implementovali v našom pluginu pre MeshLab. Naše rozšírenie bude pridávať dva typy vizualizácie kvality:

1. Vizualizácia pomocou základných metrik a ich kombinácií:

Prvý typ vizualizácií, ktorý bude náš plugin obsahovať sú vizualizácie metrik kvality, ktoré sme opísali v predchádzajúcej časti našej práce, presnejšie v časti 1.2.

Konkrétne pôjde o vizualizácie založené na plošnej metrike, pomere polomerov vpísanej a opísanej kružnice, pomere výšok, uhlovej metrike a pomere strán.

Používateľ bude mať možnosť vybrať si jednu alebo viacero metrick pre vizualizáciu. V prípade výberu viacerých metrick dôjde k ich kombinácii. Kombinácia viacerých metrick má svoje výhody. Poskytuje komplexnejší prehľad o celkovej kvalite meshov a pomáha identifikovať špecifické problémy, ktoré by mohli byť prehliadnuté, ak by sa použila len jedna metrika.

Vizualizácia týchto metrick spočíva vo farbení jednotlivých stien 3D modelu na základe vypočítaných hodnôt. V tomto momente sa hodí podotknúť, že MeshLab už disponuje nástrojom, ktorý umožňujú vizualizáciu na základe pomeru polomerov vpísanej a opísanej kružnice a obsahu plôch trojuholníkov. Avšak existujúce, nemenné nastavenia týchto vizualizácií, nie sú vždy ideálne prispôsobené potrebám užívateľov. Napríklad v prípade vizualizácie pomeru polomerov sú rovnostranné trojuholníky, ktoré sú často považované za optimálne, často zobrazené modrou farbou. Táto farba v farebnom mapovaní typicky indikuje odchýlku od ideálneho stavu, čo môže viesť k nesprávnym interpretáciám dát získaných z vizualizácie. Zelené odtiene, ktoré sú naopak používané na označenie optimálnych vlastností, by boli vhodnejšie na označenie týchto trojuholníkov. Tento nedostatok náš plugin odstráni a poskytne používateľovi intuitívnejší spôsob vizualizácie.

Používateľ využívajúci náš plugin bude mať taktiež na výber spôsob, akým budú steny meshu zafarbené. Bude si môcť vybrať, či budú metriky zobrazené pomocou špecifických, optimálnych hodnôt pre danú metriku, alebo využije farebné mapovanie, ktoré vychádza z rozsahu vypočítaných hodnôt pre celý mesh. V prípade druhej možnosti, teda farebného mapovania vychádzajúceho z rozsahu, používateľ môže dodatočne určiť, či za optimálnu hodnotu považuje priemer hodnôt z vypočítaného rozsahu alebo strednú hodnotu rozsahu. Táto flexibilita vo výbere umožňuje užívateľom prispôsobiť vizualizáciu podľa ich analytických potrieb a preferencií.

2. Vizualizácia odchýlky uhlov v okolí vrcholov od optimálneho, 60-stupňového uhla:

Druhý typ vizualizácie, ktorý bude náš plugin obsahovať je vizualizácia odchýlok uhlov v okolí vrcholov od ideálneho 60-stupňového uhla. Tento nástroj poskytne vizuálny prehľad o tom, ako sa jednotlivé uhly okolo vrcholov odlišujú od ideálneho uhla rovnostranného trojuholníka.

Podobne ako aj pri prvom type vizualizácie v našom rozšírení, bude aj táto vizualizácia založená na farebnom mapovaní, presnejšie na farebnom mapovaní vrcholov. Používatelia si taktiež budú môcť vybrať, akým spôsobom budú vrcholy zafarbené. Môžu si vybrať, či vrcholy budú zafarbené podľa priemeru uhlov, ktoré

sa v okolí vrcholov nachádzajú alebo či sa vrcholy zafarbia podľa maximálnej odchýlky od 60-stupňového uhla.

Vizualizácia odchýlky uhlov v okolí vrcholov prinesie používateľom viaceré výhody, ktoré zlepšujú analýzu a prácu s 3D modelmi. Táto vizualizácia umožní používateľom okamžite identifikovať a hodnotiť geometrické vlastnosti vrcholov a susediacich stien. Pomocou farebného mapovania týchto odchýlok pomôžeme používateľom nájsť oblasti v ich 3D sieťach, ktoré obsahujú trojuholníky s príliš ostrými alebo tupými uhlami, ktoré môžu následne viesť k technickým komplikáciám a problémom.

Celkovo náš plugin a jeho vizualizácie sú navrhnuté tak, aby poskytli používateľom širokú škálu možností pre prispôsobenie podľa ich špecifických potrieb. Každá vizualizácia môže byť detailne nastavená, čo umožní používateľom presne definovať, ako chcú vizuálne reprezentovať kvalitu a charakteristiky svojich 3D sietí. Zároveň tieto nástroje sú navrhnuté tak, aby boli čo najviac intuitívne a používateľsky prívetivé, čím sa zabezpečuje, že aj bez rozsiahlych technických vedomostí môže používateľ efektívne využívať všetky funkcie, ktoré plugin ponúka.

2.3 Používateľské rozhranie

Používateľské prostredie MeshLab je navrhnuté tak, aby sa v ňom jednoducho spravovali a analyzovali 3D modely. Skladá sa z niekoľkých základných komponentov, pričom každý komponent má intuitívne ovládanie a špecifickú funkčnosť. Nižšie sú uvedené a opísané hlavné časti používateľského rozhrania:

1. Menu:

Na vrchu okna aplikácie MeshLab sa nachádza hlavné menu, ktoré poskytuje prístup k rôznym funkciám, ako sú import a export dát, úpravy meshov, aplikácia filtrov, nástroje a nastavenia aplikácie. Taktiež sa tu nachádza špeciálne menu pre prístup k dokumentácii a informáciám o aktuálne nainštalovaných rozšíreniach. Menu je štruktúrované do logických sekcií, ktoré zodpovedajú rôznym aspektom spracovania 3D modelov a nastaveniam aplikácie.

2. Panel nástrojov:

Pod položkami menu sa nachádza panel nástrojov, ktorý poskytuje rýchly prístup k širokému spektru funkcií, ako sú import/export súborov, zmeny zobrazenia a nástrojov na úpravu meshov. Tieto nástroje sú taktiež rozdelené do logických skupín podľa ich funkcionality, čo uľahčuje orientovanie sa medzi nimi.

3. Pracovná plocha:

Hlavnou a najväčšou časťou okna MeshLabu je pracovná plocha, kde sú zobrazené 3D modely. Používatelia tu môžu interaktívne manipulovať s meshmi v reálnom čase za použitia myši či dotykových plôch (touchpadov).

4. Bočný panel:

Bočný panel sa používa na zobrazenie zoznamu aktuálne otvorených modelov a zobrazenie dodatočných informácií o aktuálne vybranom 3D modeli ako napríklad počet vrcholov a počet stien. Taktiež sa používa na prístup k špecifickým nastaveniam pre vybrané nástroje a filtre.

5. Log:

Zobrazuje informácie o vykonaných operáciách, chybových hláseniach a ďalších systémových správach.

Náš plugin bude pridávať nové možnosti do sekcie Filters > Quality Measure v hlavnom menu MeshLabu. Zároveň toto rozšírenie bude interagovať a spolupracovať s už existujúcimi komponentami používateľského rozhrania ako log a bočný panel. Táto integrácia nášho pluginu do existujúceho rozhrania je navrhnutá tak, aby bola čo najviac intuitívna, pričom nové funkcie, ktoré plugin pridáva, budú prirodzene zapadať do štruktúry menu a budú prístupné rovnakým spôsobom ako štandardné funkcie MeshLabu.

2.4 Formáty vstupu a výstupu

V tejto časti uvedieme, s akými typmi vstupných dát bude náš plugin vedieť efektívne pracovať a aké výstupné dáta bude náš plugin generovať. Taktiež uvedieme príklady formátov 3D modelov podporované aplikáciou MeshLab a opíšeme, ako sa tieto dáta konvertujú do interných štruktúr potrebných pre náš nástroj.

2.4.1 Vstupné dáta

Vstupné dáta pre tento projekt zahŕňajú všetky formáty 3D modelov, ktoré sú podporované aplikáciou MeshLab. To zahŕňa širokú škálu štandardných formátov používaných v 3D modelovaní, ako sú OBJ, STL, PLY, 3DS a GLTF. Dôležitým kľúčom pri spracovaní týchto dát je ich konverzia do univerzálneho formátu, ktorý je reprezentovaný cez interné dátové štruktúry MeshLabu, a to sú MeshModel a MeshDocument. Transformácia všetkých vstupných formátov do jednotnej internej štruktúry umožňuje jednotné a efektívne spracovanie bez ohľadu na pôvodný zdroj dát.

Proces transformácie:

1. Import dát:

Keď používateľ importuje 3D model do systému MeshLab, softvér automaticky prevedie vstupný súbor z jeho pôvodného formátu do internej dátovej štruktúry známej ako MeshModel. Tento proces zahŕňa načítanie geometrických a vizuálnych informácií z modelu, vrátane vrcholov, hrán, plôch, textúr a farieb. MeshLab zanalyzuje štruktúru importovaného modelu, identifikuje a extrahuje nevyhnutné atribúty, ako sú súradnice vrcholov, informácie o hrách a plochách, ktoré sú základom pre tvorbu a analýzu 3D meshu.

2. Uloženie v MeshDocument:

Každý MeshModel je následne uložený v MeshDocument, ktorý slúži ako kontajner pre jeden alebo viacero MeshModelov. Tento dokument uľahčuje správu viacerých modelov súčasne, čo je ideálne pre projekty, kde je potrebné vykonávať operácie medzi viacerými rôznymi modelmi.

Po tejto konverzii sú modely pripravené na ďalšie spracovanie pomocou rozličných nástrojov a filtrov, ktoré MeshLab priamo poskytuje, ale aj tých, ktoré si používatelia sami naprogramujú.

2.4.2 Výstupné dáta

Náš plugin bude modifikovať premenné objektov vrcholov a stien, ktoré sú reprezentované triedami CVertexO a CFaceO, a taktiež dátové masky v triede MeshModel. Tieto zmeny sa následne po zapnutí náhľadu alebo aplikácii filtra prejavia na meshi zobrazovanom v pracovnej ploche MeshLabu, čím vznikne výsledná vizualizácia kvality.

1. Vizualizácia pomocou farbenia stien:

Pre prvú možnosť vizualizácie plugin pridá do triedy MeshModel informácie o farbe stien. Táto úprava nám následne umožní pridávať a modifikovať farby jednotlivých stien 3D siete. Na základe tejto úpravy plugin budeme môcť nastaviť atribút farby každej steny, ktorá je reprezentovaná prostredníctvom triedy CFaceO. Tieto steny sú prístupné cez objekt triedy CMeshO, ktorý je súčasťou triedy MeshModel.

2. Vizualizácia pomocou farbenia vrcholov:

Pre druhú možnosť vizualizácie, ktorú náš plugin bude ponúkať, plugin pridá do dátovej masky objektu triedy MeshModel informáciu o farbe vrcholov. Podobne

ako pri stenách, po nastavení dátovej masky budeme môcť ľubovoľne pridať a nastaviť farbu ktorémukoľvek vrcholu. Plugin po vykonaní potrebných výpočtov následne zmení farbu jednotlivým vrcholom, ktoré sú reprezentované cez triedu `CVertexO`. Ku jednotlivým vrcholom máme opäť prístup cez triedu `CMeshO`.

2.5 Výber programovacieho jazyka

Na vytvorenie nášho vlastného pluginu pre softvér MeshLab sme sa rozhodli využiť C++ z niekoľkých dôvodov. C++ ponúka výborné riadenie systémových zdrojov, vrátane pamäti a procesora. To je výhodné pre efektívne spracovanie veľkých a zložitých 3D dát a ťažkých matematických výpočtov. Jeho výkon zároveň umožňuje rýchlejšie spracovanie dát, čo zlepšuje odozvu programu. Ďalšou výhodou je jeho kompatibilita s množstvom existujúcich knižníc zameraných na grafické spracovanie a matematické výpočty, ktoré MeshLab intenzívne využíva. Použitie C++ zabezpečuje lepšie prepojenie s týmito knižnicami. C++ je taktiež podporovaný na všetkých hlavných platformách, čo zabezpečuje, že MeshLab a náš plugin môže byť ľahko využívaný medzi rôznymi operačnými systémami. Naše znalosti a skúsenosti s C++ sú tiež nezanedbateľným faktorom, keďže využívanie jazyka, v ktorom máme skúsenosti, zvyšuje efektívnosť a znižuje čas potrebný na vývoj. Všetky uvedené aspekty spolu robia tento jazyk ideálny pre našu tvorbu pluginu pre MeshLab.

2.6 Formát pluginu a vloženie do Meshlabu

Formát súboru pluginu pre MeshLab sa líši v závislosti od operačného systému, na ktorom je aplikácia používaná. Keďže náš plugin vyvíjame a testujeme na operačnom systéme Windows 10, je potrebné, aby bol plugin vo formáte DLL, aby sme ho mohli na tejto platforme používať. V nasledujúcej časti podrobne opíšeme, ako funguje formát DLL, ako sa bude plugin vkladať do Meshlabu, ako je možné plugin kompilovať aj na iných platformách, ktoré podporujú jazyk C++ a aplikáciu MeshLab.

2.6.1 DLL formát

Na platforme Windows sú pluginy pre Meshlab zvyčajne distribuované vo forme DLL súborov. DLL, alebo Dynamic-Link Library (Dynamicky linkovaná knižnica), je formát súboru používaný v operačných systémoch Windows na zdieľanie funkcií medzi viacerými aplikáciami. DLL súbory obsahujú kód a dáta, ktoré sa dynamicky načítajú do pracovnej pamäte na vyžiadanie. Výhodou DLL súborov je to, že umožňujú viacerým aplikáciám zdieľať rovnaký kód v pamäti, čím znižujú redundanciu a šetria pamäť. Ďalšou výhodou je to, že aplikácie môžu byť navrhnuté modulárne. Tieto princípy využíva

aj MeshLab. Pluginy sú vyvíjané oddelene od hlavnej aplikácie, používajú len vopred dohodnuté rozhrania a do pamäte sa načítajú len v prípade volania špecifickej funkcie, ktorú obsahujú. Tento prístup oddelenia zároveň uľahčuje aktualizáciu jednotlivých častí aplikácii, keďže následne nie je nutné preinštalovanie celej aplikácie [14].

2.6.2 Predkompilovaný Plugin

V prípade použitia predkompilovaného pluginu vo formáte .dll na operačnom systéme Windows existujú dva efektívne spôsoby, ako možno daný plugin vložiť do prostredia softvéru MeshLab.

1. Vloženie pluginu do predvoleného priečinka:

Prvý a najjednoduchší spôsob vloženia pluginu spočíva v jeho umiestnení do špecifického priečinka určeného pre pluginy v MeshLab. Tento priečinok sa obvykle nachádza na nasledujúcej adrese C:\Program Files\VCG\MeshLab\plugins v prípade systémov s operačnými systémami Windows. Po vložení .dll súboru pluginu do tohto priečinka sa toto rozšírenie automaticky načíta pri štarte aplikácie.

2. Načítanie cez užívateľské rozhranie MeshLab:

Druhý spôsob umožňuje načítať plugin priamo z používateľského rozhrania MeshLab. Tento spôsob spočíva v pár krokoch:

- Otvoríme MeshLab a prejdeme do sekcie Help v hlavnom menu.
- Vyberieme možnosť Plugin Info. Následne sa otvorí okno s prehľadom aktuálne načítaných pluginov.
- V tomto okne sa ponúka tlačidlo Load Plugins, ktoré spustí štandardné dialógové okno Windows pre výber súborov.
- Prostredníctvom tohto dialógu vyhľadáme .dll súboru pluginu, vyberieme ho a potvrdíme jeho otvorenie.

Tieto metódy poskytujú rýchlu a jednoduchú integráciu bez potreby zložitého nastavenia. Avšak, v niektorých prípadoch tieto spôsoby nemusia zaručne fungovať, keďže funkčnosť DLL súborov môže závisieť od špecifického systémového nastavenia a konfigurácií. To môže spôsobiť, že DLL plugin, ktorý bol vygenerovaný na jednom konkrétnom počítači, nemusí správne fungovať na iných systémoch. Problémy môžu nastať už pri načítaní pluginu, kde sa môže objaviť chybové hlásenie, alebo môže dôjsť k pádu programu pri volaní funkcie z pluginu. V prípade, že DLL súbor pluginu nefunguje, je potrebné skompilovať tento súbor priamo zo zdrojových kódov na cieľovom počítači. To zabezpečí, že DLL bude kompatibilný s konkrétnym operačným systémom a jeho špecifickými nastaveniami.

2.6.3 Kompilácia pluginu zo zdrojového kódu

Kompilácia pluginu do MeshLabu zo zdrojových kódov je proces, ktorý vyžaduje základné znalosti o programovaní v jazyku C++ a použití nástroja CMake, ktorý sa používa pre konfiguráciu a generovanie buildov systémov. Na začiatok si treba stiahnuť zdrojové kódy MeshLabu, vrátane všetkých submodulov. Tieto zdrojové kódy sa nachádzajú na oficiálnom GitHub repozitári. Pred začatím kompilácie je nutné, aby používateľ mal nainštalované nasledujúce nástroje:

- C++ kompilátor: napríklad MSVC pre Windows, GCC pre Linux.
- CMake: treba si nainštalovať verziu podľa platformy zariadenia.
- Qt SDK: používateľ sa musí uistiť, že má nainštalovanú kompatibilnú verziu Qt s konkrétnou verziou MeshLabu. Pre najnovšie vydanie MeshLabu, označované ako MeshLab-2023.12, ktoré bolo najaktuálnejšie počas písania práce, je treba použiť Qt vo verzii 5.15.

Po úspešnom vykonaní všetkých týchto krokov je potrebné vytvoriť priečinok pre naše rozšírenie v `meshlab\src\meshlabplugins` a do tohoto priečinka nakopírovať zdrojové kódy. Následne treba upraviť súbor `CMakeLists.txt` v adresári `meshlab\src`. V tomto súbore pridáme do sekcie `Filter plugins` informáciu o novom priečinku so zdrojovými kódmi, čo umožní nástroju CMake identifikovať a zahrnúť nové rozšírenie do buildu projektu. Keď sú zdrojové kódy na mieste a CMake konfigurácia je aktualizovaná, tak sa môžeme pustiť do generovania buildu projektu. Pomocou programu CMake spustíme spracovanie súboru `meshlab\CMakeLists.txt`, čo následne vytvorí build projektu. Ten potom môžeme otvoriť a skompilovať pomocou integrovaného vývojového prostredia (IDE) ako je Visual Studio alebo Qt Creator. V tomto kroku sa môžeme rozhodnúť, či chceme generovať finálnu release verziu alebo verziu pre ladenie (debug). Výsledný DLL súbor pluginu sa po kompilácii nachádza v priečinku `meshlab\build\src\distrib\plugins` v prípade systému Windows (na iných platformách sa DLL nachádza v build priečinku pod sekciou `plugins`). Tento súbor potom môžeme vložiť do inštalácie MeshLabu podľa postupov popísaných v sekcii 2.6.2.

Kapitola 3

Implementácia a experimenty

Kapitola Implementácia a experimenty sa zameriava na konkrétne aspekty realizácie nášho projektu, vrátane detailného popisu technických rozhodnutí a použitých metód, ktoré boli aplikované pri vývoji softvéru. Ďalej sa v tejto kapitole venujeme aj experimentom, ktoré boli vykonané na overenie funkčnosti implementovaných riešení.

3.1 Implementácia

V tejto sekcii sa budeme zaoberať implementáciou pluginu. Predstavíme základnú architektúru, vrátane hlavných tried a funkcií, ktoré zabezpečujú jeho integráciu do Meshlabu a rozoberieme funkcie, ktoré slúžia na manipuláciu a analýzu 3D meshov.

3.1.1 Štruktúra pluginu

Náš plugin pre MeshLab je štruktúrovaný okolo dvoch základných súborov: `plugin.h` a `plugin.cpp`. Tento prístup zabezpečuje prehľadnosť a modularitu, ktorá je nevyhnutná pre efektívny vývoj a údržbu softvéru v rámci rozsiahleho projektu ako je MeshLab. Štruktúra nášho rozšírenia pozostáva z nasledujúcich súborov:

- Hlavičkový súbor: `plugin.h`

Tento súbor definuje triedu `Plugin`, ktorá je odvodená od triedy `FilterPlugin` poskytovanej systémom MeshLab. Hlavičkový súbor taktiež obsahuje deklarácie potrebných metód a premenných, ktoré sú špecifické pre funkcie pluginu, metódy pre spracovanie dát a aplikáciu filtrov. Zároveň definuje enumerácie, ktoré slúžia ako identifikátory pre jednotlivé vizualizácie implementované v plugine.

- Implementačný súbor: `plugin.cpp`

Tento súbor obsahuje konkrétne implementácie všetkých metód a funkcií deklarovaných v hlavičkovom súbore, ako sú funkcie na inicializáciu pluginu, spracovanie

parametrov a poskytnutie informácií o týchto filtroch. V súbore je definovaný konštruktor triedy, ktorý inicializuje potrebné akcie pre jednotlivé filtre a nastavuje interné štruktúry potrebné pre správne fungovanie filtrov. Zároveň je tu implementovaná funkcia `applyFilter()`, ktorá zodpovedá za aplikáciu konkrétnych transformácií na 3D modely na základe vybraného filtra a používateľských nastavení.

Trieda `Plugin` dedí z `FilterPlugin`, čo je základná trieda pre všetky filtračné pluginy v MeshLab. Toto dedenie umožňuje triede `Plugin` hladko sa integrovať do existujúcej architektúry MeshLabu a využívať rozhrania definované v `FilterPlugin` pre interakciu s hlavnou aplikáciou. Trieda `Plugin` taktiež dedí z triedy `QObject`. To umožňuje efektívnu správu objektov a signálov v rámci Qt frameworku. Softvér MeshLab využíva Qt pre implementáciu svojho grafického používateľského rozhrania a pre manipuláciu s udalosťami. To zahŕňa spracovanie akcií používateľa a interakciu medzi rôznymi komponentmi systému. Dedenie z `QObject` poskytuje triede `Plugin` prístup k týmto dôležitým funkcionalitám a tým umožňuje integráciu s hlavným aplikačným oknom a ostatnými aspektmi Qt prostredia.

3.1.2 Funkcie potrebné pre vytvorenie pluginu

`Plugin`, ktorý sme vyvinuli pre MeshLab, obsahuje rad základných aj rozšírených funkcií implementovaných v súboroch `plugin.h` a `plugin.cpp`, ktoré sme opísali v časti 3.1.1. Tieto funkcie umožňujú pluginu správne fungovať v rámci ekosystému MeshLab, spracovávať 3D modely a poskytovať používateľovi rozhranie pre interakciu s týmito modelmi. Nižšie sú opísané kľúčové funkcie obsiahnuté v našom kóde, ktoré slúžia na integráciu do prostredia meslabu:

Konštruktor:

- `Plugin()`:

Táto funkcia inicializuje plugin a jeho základné nastavenia, vrátane registrácie rôznych filtrov, ktoré plugin ponúka. Tieto filtre sú identifikované prostredníctvom enumerátorov ako `FP_FIRST_VIS`, `FP_SECOND_VIS`.

Informačné Funkcie:

- `filterName(ActionIDType filterId)`:

Táto funkcia poskytuje krátky, ale výstižný popis každej filtračnej akcie. Vracia reťazec, ktorý je používaný nielen pre popis konkrétnej funkcionality filtra, ale aj pre definíciu položky v menu MeshLabu.

- `filterInfo(ActionIDType filterId):`

Táto funkcia poskytuje podrobný popis každej akcie filtra. Text, ktorý funkcia vracia, je zobrazený na vrchu okna s parametrami pri aktivácii konkrétneho filtra a je dizajnovaný tak, aby poskytoval rozsiahle informácie o použití filtra.

Konfiguračné Funkcie:

- `initParameterList(QAction *a, MeshDocument & md):`

Táto funkcia inicializuje a definuje zoznam parametrov, ktoré sú potrebné pre vykonanie vybraného filtra. Parametre sú potom zobrazené v dialógovom okne, kde ich môže používateľ upraviť pred aplikovaním filtra.

- `filterArity(QAction *act):`

Určuje, či filter pôsobí na jednotlivý mesh model alebo na viacero modelov súčasne. Toto je dôležité pre správne spracovanie vstupov v závislosti od typu filtra.

Pomocné Funkcie::

- `postCondition(const QAction *filter):`

Definuje, ktoré atribúty mesh modelu sú zmenené po aplikácii filtra, umožňujúc MeshLabu správne aktualizovať svoje zobrazenie a dáta.

Aplikačná Funkcia:

- `applyFilter(QAction* action, Mesh-Document & md...):`

Toto je hlavná funkcia pluginu, ktorá aplikuje logiku filtra na mesh model. Funkcia zohľadňuje nastavené parametre a používa rôzne techniky (napr. zmeny farby, úpravy geometrie) na modifikáciu alebo analýzu 3D modelu. Výsledky sú potom reflektované v modeli a prípadne sú používateľovi zobrazené štatistiky alebo ďalšie relevantné informácie.

3.1.3 Farbenie hrán a vrcholov

Aby sme mohli priradovať farby vrcholom alebo stranám, implementovali sme funkciu `GetColorForValue()`. Táto funkcia je základom nášho farebného mapovania. Funkcia nám vracia farby typu `Color4b`, ktoré následne priradíme do príslušného atribútu farby jednotlivým stranám a vrcholom. Trieda `Color4b` je súčasťou knižnice `VCG` (`Visualization and Computer Graphics Library`), ktorú `Meshlab` aktívne využíva, pretože poskytuje nástroje na spracovanie a vizualizáciu geometrických objektov. Trieda

Color4b predstavuje farbu s RGBA kanálmi. RGBA farba je definovaná štyrmi zložkami: červená (red), zelená (green), modrá (blue) a alfa (alpha). Každý kanál je reprezentovaný jedným bajtom a teda hodnota každého kanála je v rozsahu 0 až 255.

Farby priraďujeme objektom na základe ich hodnôt vzhľadom na minimálnu, optimálnu a maximálnu hodnotu. Funkcia `GetColorForValue()` zabezpečuje, že objekty s hodnotou pod minimom budú červené, nad maximom modré a optimálne hodnoty budú zelené. Pre hodnoty medzi týmito bodmi sa farba interpoluje. Pre výpočet interpolovaných farieb sme implementovali funkciu `InterpolateColor()`, ktorá interpoluje medzi dvoma farbami (medzi červenou a zelenou alebo zelenou a modrou) na základe zadaného faktora. Faktor nám určí, ako blízko sa výsledná farba bude nachádzať ku ktorejkoľvek z dvoch vstupných farieb. Farby a výpočet hodnoty faktora, ktoré táto funkcia dostane na vstup závisia od hodnoty, ktorú má funkcia `GetColorForValue()` vizualizovať a od toho, či je táto hodnota menšia alebo väčšia ako optimálna.

3.1.4 Výpočet 1. metriky - Obsah trojuholníkov

Na výpočet obsahu trojuholníkov sme navrhli a implementovali funkciu `CalculateTriangleArea()`. Na výpočet obsahov využíva základné princípy vektorovej algebry, konkrétne vektorový súčin a normu vektora. Funkcia pracuje nasledovne:

- Vrcholové body trojuholníka:

Funkcia najprv z `FaceIterátora` (objektu, ktorý reprezentuje jednu stranu meshu), ktorý dostane na vstupe, získa súradnice vrcholov $v_1 = (x_1, y_1, z_1)$, $v_2 = (x_2, y_2, z_2)$ a $v_3 = (x_3, y_3, z_3)$.

- Vektory hrán:

Následne funkcia vypočíta 2 vektory, reprezentujúce dve hrany trojuholníka. Vypočítame vektor $edge1 = v_2 - v_1 = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$, ktorý reprezentuje hranu z bodu v_1 do v_2 a taktiež vektor $edge2 = v_3 - v_1 = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$, čo reprezentuje hranu z v_1 do v_3 .

- Vektorový súčin:

Vektorový súčin dvoch vektorov (hrán) vytvára nový vektor, kolmý na rovinu, ktorú tieto dva vektory tvoria. Tento vektor obsahuje informáciu o ploche rovnobežníka, ktorého strany sú tieto dva vektory. Funkcia preto následne vypočíta vektorový súčin vektorov $edge1$ a $edge2$. Na výpočet sme použili špeciálny operátor $\hat{\cdot}$, ktorý sa používa na výpočet vektorových súčinov a pracuje podľa vzorca $a \hat{\cdot} b = (a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x)$.

- Výpočet plochy:

Následne funkcia počíta plochu trojuholníka. Plocha trojuholníka je polovica plochy rovnobežníka tvoreného vektormi *edge1* a *edge2*, čo môžeme matematicky zapísať ako $Plocha = 0.5 \times \| a \wedge b \|$. Na výpočet veľkosti tohto rovnobežníka sme použili funkciu `Norm()`, ktorá počíta veľkosť vektoru podľa vzorca $\| c \| = \sqrt{c_x^2 + c_y^2 + c_z^2}$, kde $c = (c_x, c_y, c_z)$

Týmto postupom funkcia efektívne vypočíta plochu trojuholníka v 3D priestore pomocou vektorovej algebry. Túto funkciu používame nielen pre výpočet plošnej metriky, ale aj pri výpočtoch výšok pre min-max výškovú metriku a pre výpočet polomerov vpísaných a opísaných kružníc trojuholníkov.

3.1.5 Výpočet 2. metriky - Podiel polomerov

Pre výpočet metriky podielu polomerov vpísanej a opísanej kružnice sme implementovali funkciu `CalculateCircleRadii()`. Podobne ako predchádzajúca funkcia, na vstup dostane objekt typu `FaceIterator`, reprezentujúci jeden trojuholník, presnejšie jednu stenu meshu. Funkcia počíta nasledovne:

- Vrcholy trojuholníka a veľkosť hrán:

Funkcia na začiatku získa súradnice jednotlivých vrcholov a pomocou funkcie `CalculateDistance()` vypočíta všetky tri strany trojuholníka. Funkcia dostane dva body, reprezentujúce vrcholy trojuholníka a pomocou vzorca $|AB| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$ vypočíta ich vzdialenosť.

- Výpočet polomerov:

Následne funkcia vypočíta jednotlivé polomery. Na výpočet polomeru vpísanej kružnice použije vzorec $r = \frac{2S}{a+b+c}$ a na výpočet polomeru opísanej kružnice vzorec $R = \frac{a \times b \times c}{4S}$, pričom S je obsah trojuholníka. V oboch prípadoch na výpočet obsahu trojuholníka použijeme funkciu `CalculateTriangleArea(..)`, ktorú sme opísali v časti 3.1.4.

- Výpočet pomeru:

Funkcia na záver ošetrí prípady, kedy by mohlo dôjsť k prípadu delenia nulou a vráti výsledný pomer, teda vypočíta $ratio = \frac{r}{R}$.

3.1.6 Výpočet 3. metriky - Uhlová metrika

Na zistenie minimálneho respektívne maximálneho uhla sme implementovali funkciu `CalculateMinMaxAngles()`. Taktiež ako u predchádzajúcich funkcií dostane na

vstup objekt trojuholníka triedy `FaceIterator`. Celkovo je funkcia založená na nasledujúcom, jednoduchom nápade:

- Zistenie súradníc, strán a výpočet uhlov:

Na začiatku zo súradníc vrcholov vypočítame dĺžky jednotlivých strán. Na výpočet dĺžok použijeme opäť funkciu `CalculateDistance()`, ktorú sme opísali v časti 3.1.5. Následne za pomoci kosínusovej vety vieme podľa vzorcov $\alpha = \arccos\left(\frac{b^2+c^2-a^2}{2bc}\right)$, $\beta = \arccos\left(\frac{a^2+c^2-b^2}{2ac}\right)$ a $\gamma = \arccos\left(\frac{a^2+b^2-c^2}{2ab}\right)$ vypočítať všetky vnútorné uhly trojuholníka. Keďže ale C++ funkcia `acos()`, ktorá počíta arkus kosínus, vracia na výstup radiány, vynásobíme jednotlivé výsledky konštantou $\frac{180}{\pi}$.

- Výpočet najväčšieho rozdielu:

Následne funkcia nájde za pomoci funkcií `min()` a `max()` najmenší a najväčší uhol. Zároveň skontroluje, či sú tieto uhly v rozsahu od 0° do 180° a pomocou porovnania vráti uhol, ktorý má najväčšiu odchýlku od 60° .

3.1.7 Výpočet 4. metriky - Pomer strán

Funkcia `CalculateMaxMinSideRatio()` je najjednoduchšou funkciou zo všetkých implementovaných funkcií, ktoré rátaajú jednotlivé metriky. Funkcia vypočíta pomer medzi najdlhšou a najkratšou stranou. Zo vstupného objektu trojuholníka zistí potrebné súradnice vrcholov a pomocou funkcie `CalculateDistance()` zistí rozmery strán. Následne pomocou `min` a `max` funkcie nájdeme najmenšiu a najväčšiu stranu. Následne vráti výsledný pomer strán, teda hodnotu $ratio = \frac{minLength}{maxLength}$. Funkcia pred vykonaním delenia ešte skontroluje a ošetrí prípadné delenie nulou.

3.1.8 Výpočet 5. metriky - Min-Max výšková metrika

Na výpočet Min-Max výškovej metriky, ktorá sa zaoberá pomerom minimálnej a maximálnej výšky, sme implementovali funkciu `CalculateMinMaxHeightsRatio()`. Funkcia postupuje nasledovne:

- Obsah trojuholníka a dĺžky strán:

Funkcia najprv vypočíta obsah trojuholníka a jednotlivé dĺžky strán. Následne funkcia vypočíta pomocou vzorcov $v_a = \frac{2S}{a}$, $v_b = \frac{2S}{b}$ a $v_c = \frac{2S}{c}$ jednotlivé výšky.

- Nájdenie minima a maxima, výpočet pomeru:

Nakoniec pomocou `min()` a `max()` funkcií nájde najmenšiu a najväčšiu výšku a vypočíta výsledný pomer ako $\frac{minHeight}{maxHeight}$. Samozrejme funkcia ošetrí prípadné delenie nulou.

3.1.9 Kombinácia metrík

Prvá vizualizácia ma za úlohu vizualizovať kombináciu jednej alebo viacerých metrík podľa príslušných nastavení od používateľa. Pre jednoduchšiu implementáciu kombinovania sme vytvorili funkciu `FP_FIRST_VIS_Apply()`. Táto funkcia prijíma ako vstup mriežku (mesh), parameter špecifikujúci metriku, ktorú má vypočítať, parameter určujúci typ farebného mapovania a informáciu o tom, či už bola táto funkcia volaná.

Pre potreby implementácie kombinácií metrík sme navrhli a implementovali dátovú štruktúru, ktorá nám uľahčí ukladanie a prenos výsledkov z výpočtov jednotlivých základných metrík. Táto štruktúra obsahuje:

- `double sum`:

Uchováva súčet všetkých vypočítaných hodnôt metrík. Používame ju na výpočet priemeru týchto hodnôt v prípade, že používateľ chce vizualizovať rozsah hodnôt.

- `int counter`:

Premenná `counter` počítá počet spracovaných trojuholníkov, pre ktoré bola metrika vypočítaná. Taktiež ju využívame na výpočet priemeru hodnôt.

- `double min`:

Premenná `min` uchováva minimálnu hodnotu metriky z rozsahu metriky a využívame ju pre funkciu `GetColorForValue()`, opísanú v časti 3.1.3.

- `double max`:

Premenná `max` uchováva maximálnu hodnotu metriky vypočítanú pre všetky plôšky trojuholníkov. Pomáha určiť najvyššiu hodnotu pre funkciu `GetColorForValue()`.

- `mapu map<CMesh0::FaceIterator, double>`:

Kľúčom do tejto mapy je iterátor na tvár (`FaceIterator`). Mapa pre jednotlivé kľúče uchováva hodnoty z vypočítaných metrík. Používame ju pre finálne farbenie plôšok trojuholníkov meshu.

Funkcia `FP_FIRST_VIS_Apply()` pracuje nasledovne:

1. Kontrola parametra farebného mapovania:

Funkcia najprv skontroluje hodnotu parametra `averageOfSpecificOrUseRanges`, ktorý určuje typ farebného mapovania. Ak je parameter rovný 0, bude funkcia volať funkciu `GetColorForValue()`, do ktorej ako parametre použije špecifické, optimálne hodnoty a následne bude tieto farby priradovať a kombinovať

s farbami jednotlivých trojuholníkov. V prípade, že je parameter rovný 1 alebo 2 vypočítané hodnoty metriky vloží do štruktúry opísanej na predchádzajúcej strane.

2. Výpočet jednotlivých základných metrik:

Potom ako funkcia vie aký typ farebného mapovania si používateľ vybral, skočí do odpovedajúcej časti kódu. Následne pomocou príkazu `switch()` a hodnoty parametra `metric` určí, či a prípadne akú metriku používateľ vybral a pomocou cyklu preiteruje cez všetky tváre 3D siete. V rámci tohto cyklu volá funkcie na výpočet metrik pre jednotlivé tváre. Tieto funkcie na výpočet metrik sú detailne opísané v častiach 3.1.4 až 3.1.8.

Funkcia `FP_FIRST_VIS_Apply()` je volaná iba vo funkcii `applyFilter()`, čo je hlavná funkcia nášho rozšírenia. Funkcia `applyFilter()` načíta hodnoty zadané v dialógovom okne používateľom, aby zistila, ktoré metriky chce skombinovať a aký režim farbenia chce použiť. Na základe týchto parametrov potom až päťkrát zavolá funkciu `FP_FIRST_VIS_Apply()` s príslušnými hodnotami, keďže umožňujeme vytvoriť kombináciu až piatich metrik.

Ak si používateľ zvolil použitie špecifických hodnôt na farbenie tvárí 3D siete, funkcia `applyFilter()` sa nemusí zaoberať výsledným farbením mriežky, pretože funkcia `FP_FIRST_VIS_Apply()` to zaistí. V prípade, že sa používateľ rozhodol použiť rozsah hodnôt pre farbenie, funkcia `applyFilter()` pomocou hodnôt uložených v dátovej štruktúre a cyklu prechádzajúceho cez všetky tváre zabezpečí finálne ofarbenie jednotlivých trojuholníkov.

3.1.10 Výpočet uhlov pre farebné mapovanie vrcholov

Druhá vizualizácia má za úlohu farebne označiť jednotlivé vrcholy 3D siete. Na výpočty uhlov v okolí jednotlivých vrcholov a výsledné aplikovanie farieb sme implementovali funkciu `FP_SECOND_VIS_Apply()`. Táto funkcia na vstup dostane 3D objekt a parameter `mixColorsMode`, ktorého hodnota závisí od možnosti, ktorú si používateľ vybral v dialógovom okne. Ak je hodnota parametra 0, funkcia bude vizualizovať priemer uhlov vrcholov a ak je 1, bude vizualizovať maximálny rozdiel od 60 stupňov. Inakšie hodnoty parametre neakceptuje a vypíše výnimku, ktorá upozorní používateľa. Funkcia pracuje ďalej nasledovne:

- Režim priemerovania uhlov (`mixColorsMode == 0`):

Na začiatku tohoto režimu sa vytvorí mapa, ktorej kľúč je pointer na vrchol a bude ukladať farby jednotlivých vrcholov.

Následne sa pomocou cyklu pre každý trojuholník/face meshu vypočítajú dĺžky strán a pomocou kosínusovej vety sa vypočítajú uhly v trojuholníku.

Na základe veľkosti uhlov sa vypočítajú prislúchajúce farby, spriemerujú sa a uložia do mapy.

Po spracovaní všetkých tvári sa pomocou cyklu vypočítané farby priradia vrcholom mriežky.

- Režim Maximálneho Rozdielu od 60 Stupňov (`mixColorsMode == 1`):

Podobne ako v prvom režime sa vytvorí mapa, ktorej kľúčom je `VertexPointer` a budú sa v nej ukladať veľkosti uhlov vrcholov.

Pre každý trojuholník sa následne vypočítajú dĺžky strán a uhly pomocou kosínusovej vety a aktualizuje sa mapa uhlov tak, aby každý vrchol v mape obsahoval uhol s najväčším rozdielom od 60 stupňov.

Po spracovaní všetkých trojuholníkov sa vypočítané uhly premenia na farby a priradia vrcholom.

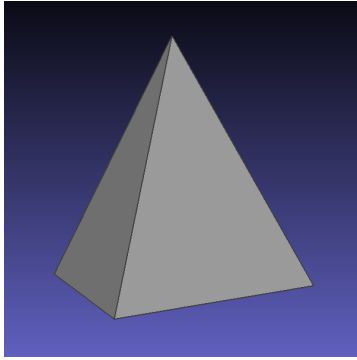
Funkcia `FP_SECOND_VIS_Apply()` je taktiež volaná iba v hlavnej funkcii filtra `applyFilter()`. Funkcia `applyFilter()` načíta hodnoty zadané v dialógovom okne, aby zistila, ktorý režim farbenia sa má použiť. Na základe zistených parametrov potom zavolá funkciu `FP_SECOND_VIS_Apply()`.

3.2 Experimenty

V tejto sekcii sa zameriame na opis a vykonanie experimentov, ktoré demonštrujú funkčnosť a efektivitu nášho pluginu `MetricCombVis`. Najprv popíšeme testovacie objekty, ktoré sme použili na overenie schopností nášho nástroja. Následne vykonáme časové merania, aby sme zistili, ako rýchlo a efektívne dokáže plugin spracovať a vizualizovať metriky na 3D meshoch. Na záver použijeme nástroj `Metro` na porovnanie dvojíc objektov a vyhodnotenie nášho prístupu v porovnaní s referenčnými modelmi.

3.2.1 Testovacie objekty

Na dôkladné otestovanie funkčnosti a efektivity nášho pluginu `MetricCombVis` sme vybrali rôznorodé testovacie objekty. Každý z týchto objektov má rôzne úrovne zložitosti a štruktúry. To nám umožní overiť, ako náš nástroj zvláda široké spektrum reálnych scenárov, s ktorými sa používatelia môžu stretnúť.



Obr. 3.1: Pravidelný štvorsten

Pravidelný štvorsten:

- Počet vrcholov: 4
- Počet trojuholníkov: 4

Pravidelný štvorsten je základným geometrickým tvarom, ktorý sa používa v digitálnom modelovaní kvôli jeho symetrii.

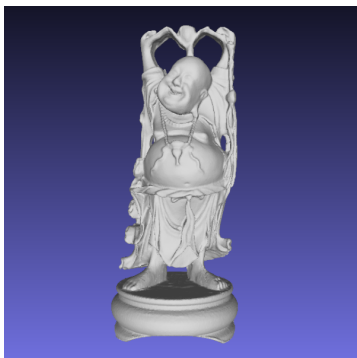


Obr. 3.2: Stanford Bunny

Stanford Bunny:

- Počet vrcholov: 35 947
- Počet trojuholníkov: 69 451
- Zdroj: Stanford Repository [25]

Stanford Bunny je známy 3D model, ktorý sa bežne používa na účely testovania.

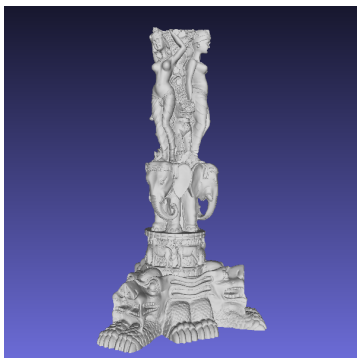


Obr. 3.3: Happy Buddha

Happy Buddha:

- Počet vrcholov: 543 652
- Počet trojuholníkov: 1 087 716
- Zdroj: Stanford Repository [25]

Podobne ako Stanford Bunny je Happy Buddha ďalším modelom, ktorý sa používa na účely testovania nových programov.



Obr. 3.4: Thai Statue

Thai Statue:

- Počet vrcholov: 4 999 996
- Počet trojuholníkov: 10 000 000
- Zdroj: Stanford Repository [25]

Thai Statue je veľmi veľký mesh, ktorý využijeme na testovanie efektivity pluginu.

3.2.2 Časová efektivita

Časová efektivita je dôležitým faktorom pri vývoji a používaní softvérových nástrojov na analýzu a vizualizáciu 3D meshov. Pri spracovaní 3D modelov je bežné, že objemy dát môžu byť veľmi veľké. Preto je dôležité, aby nástroje nielen správne analyzovali a vizualizovali dáta, ale aby to robili aj v rozumnom čase. Rýchlosť spracovania priamo ovplyvňuje produktivitu používateľov, najmä pri veľkých a zložitých modeloch, kde dlhé časy spracovania môžu výrazne spomaliť pracovný tok. Kvôli dôležitosti časovej efektivity sme sa rozhodli otestovať naše rozšírenie aj z hľadiska časovej náročnosti.

Všetky testy sme vykonali na počítači s nasledujúcimi špecifikáciami:

- Procesor: Intel i7 1165G7 2,8GHz
- Pamäť: 16GB RAM DDR4 3200 MHz
- Úložisko: M.2 NVMe SSD

Meranie časov sme vykonávali nasledovne:

1. vizualizácia: Kombinácia metrík a farebné mapovanie:

Pri prvej vizualizácii sme náhodne vybrali kombináciu dostupného počtu metrík a režim farebného mapovania. Počas testovania sme spustili plugin viackrát a zaznamenali sme časy potrebné na dokončenie úloh. Tieto časy sme získali z logu MeshLab, ktorý zaznamenáva čas potrebný na aplikáciu filtrov, teda v našom prípade zaznamenáva čas potrebný na aplikáciu vizualizácie. Na základe týchto údajov sme vyhodnotili priemerné časy potrebné na vykonanie týchto vizualizácií.

2. vizualizácia: Farebné mapovanie vrcholov podľa veľkosti uhlov:

Pri druhej vizualizácii sme niekoľkokrát spustili režim, ktorý farbí vrcholy podľa priemeru uhla v okolí vrchola alebo podľa maximálnej odchýlky od 60 stupňov. Časy sme opäť merali pomocou logu MeshLab.

Jednotlivé namerané časy je možné vidieť v tabuľkách 3.1 až 3.3. Z týchto meraní sme zistili, že časová náročnosť vizualizácií sa môže líšiť v závislosti od zvolenej kombinácie metrík a spôsobu farebného mapovania. Pre menšie modely a jednoduchšie metriky boli časy vykonania relatívne krátke, zatiaľ čo pre väčšie modely a komplexnejšie kombinácie metrík sa časy vykonania predĺžili. Napriek tomu sme dosiahli uspokojujúce výsledky, ktoré ukazujú, že plugin MetricCombVis je schopný efektívne spracovať a vizualizovať kvalitatívne vlastnosti 3D meshov v rozumnom čase.

vizualiz. \ meranie	1	2	3	4	5	6	7	8	priemer
1. viz. - 1 metrika	63	91	75	74	70	69	72	77	73,875 ms
1. viz. - 2 metriky	86	84	90	103	99	94	96	89	92,625 ms
1. viz. - 3 metriky	95	104	92	99	100	104	101	102	99,625 ms
1. viz. - 4 metriky	113	108	111	112	106	103	99	101	106,625 ms
1. viz. - 5 metrik	116	109	102	104	99	108	110	107	106,875 ms
2. viz. - priemer uhlov	139	125	123	125	126	130	128	120	127,000 ms
2. viz. - max uhol	118	122	117	125	122	126	120	119	121,125 ms

Tabuľka 3.1: Čas vizualizácie v milisekundách pre model Stanford Bunny

vizualiz. \ meranie	1	2	3	4	5	6	7	8	priemer
1. viz. - 1 metrika	391	387	687	425	400	613	574	512	466,000 ms
1. viz. - 2 metriky	757	458	699	513	594	487	541	492	567,625 ms
1. viz. - 3 metriky	761	487	507	748	684	712	511	532	617,750 ms
1. viz. - 4 metriky	838	605	580	799	754	740	675	690	653,250 ms
1. viz. - 5 metrik	804	882	905	879	926	841	899	863	874,875 ms
2. viz. - priemer uhlov	935	933	918	930	957	960	942	981	944,500 ms
2. viz. - max uhol	740	752	732	736	733	711	735	732	733,875 ms

Tabuľka 3.2: Čas vizualizácie v milisekundách pre model Happy Buddha

vizualiz. \ meranie	1	2	3	4	5	6	7	8	priemer
1. viz. - 1 metrika	3,20	3,39	5,84	3,49	3,21	4,41	3,84	4,10	3,935 s
1. viz. - 2 metriky	6,01	3,28	5,58	3,81	4,26	4,71	5,01	3,96	4,578 s
1. viz. - 3 metriky	4,17	6,87	3,74	5,81	4,92	4,31	5,10	4,32	4,905 s
1. viz. - 4 metriky	4,27	6,56	5,34	5,74	7,26	6,71	5,98	6,98	6,094 s
1. viz. - 5 metrik	7,38	7,76	7,81	6,94	8,31	7,52	7,64	7,47	7,603 s
2. viz. - priemer uhlov	8,64	8,23	8,32	9,12	8,74	8,61	8,42	8,37	8,556 s
2. viz. - max uhol	7,12	7,38	7,08	7,51	7,82	7,40	7,29	7,45	7,381 s

Tabuľka 3.3: Čas vizualizácie v sekundách pre model Thai Statue

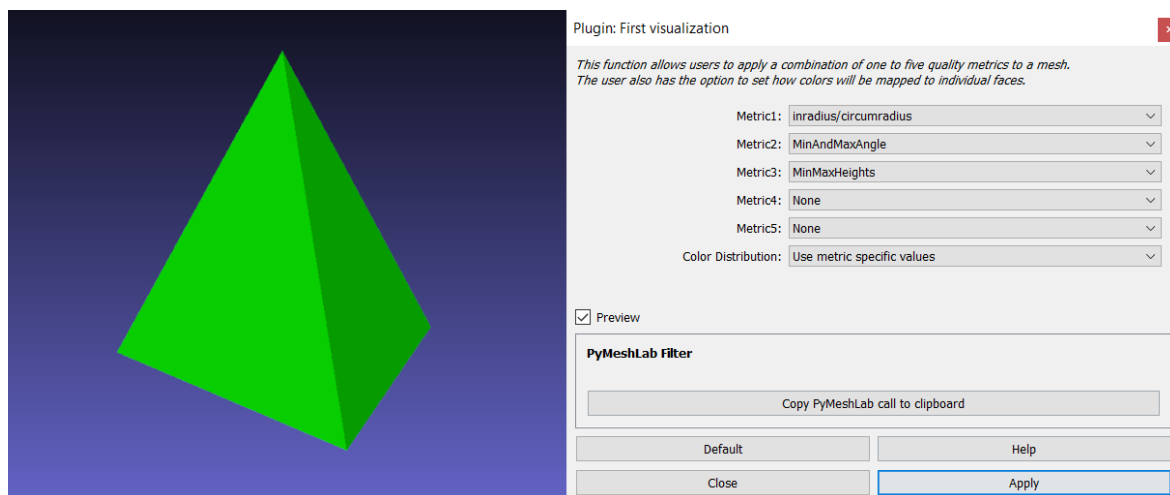
3.2.3 Príklady použitia rozšírenia

Aby používatelia pluginu a čitatelia práce pochopili, ako sa s pluginom narába a čo dokáže, uvedieme pár praktických príkladov. Tieto príklady ilustrujú možnosti a funkcionality nášho rozšírenia pre MeshLab, vrátane toho, ako vyzerá rozhranie, v ktorom môže používateľ vyberať a kombinovať rôzne metriky pre analýzu 3D modelov.

1. príklad - pravidelný štvorsten:

Na obrázku 3.5 môžeme vidieť príklad vizualizácie, ktorá kombinuje tri rôzne metriky: pomer polomerov, pomer výšok a uhlovú metriku. V prípade použitia tejto alebo ľubovolnej inej kombinácie metrik a možnosti farbenia sa vždy všetky štyri

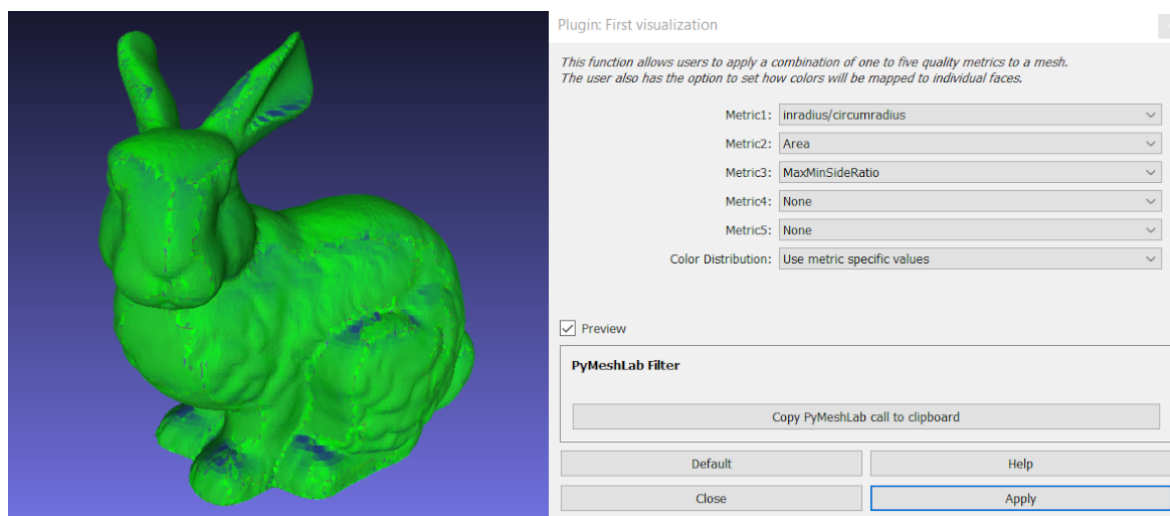
trojuholníky zafarbia na zeleno, pretože tento mesh obsahuje iba rovnostranné trojuholníky, ktoré považujeme za optimálne. Taktiež na priloženom obrázku môžeme vidieť rozhranie pluginu v MeshLab, ktoré umožňuje vytvárať a aplikovať kombinácie metrík. Používateľ jednoducho pre možnosti Metric1 až Metric5 vyberie z rozbaľovacieho zoznamu metriky, ktoré chce skombinovať a pre možnosť Color Distribution vyberie zo zoznamu spôsob, ktorým sa vypočítajú farby. Po zapnutí možnosti náhľadu (preview) alebo stlačení tlačítka Apply sa výsledná vizualizácia premietne na mesh.



Obr. 3.5: Vizualizácia kombinácie metrík na štvorstene

2. príklad - kombinácia metrík na Stanford Bunny:

Na obrázku 3.6 vidíme ďalší príklad kombinácie plošnej metriky, pomeru polomerov a metriky pomeru strán na modeli Stanford Bunny. Táto vizualizácia umožňuje lepšie identifikovať problémové oblasti v okolí uší, spoji tela a stehna, a na predných a zadných labkách.

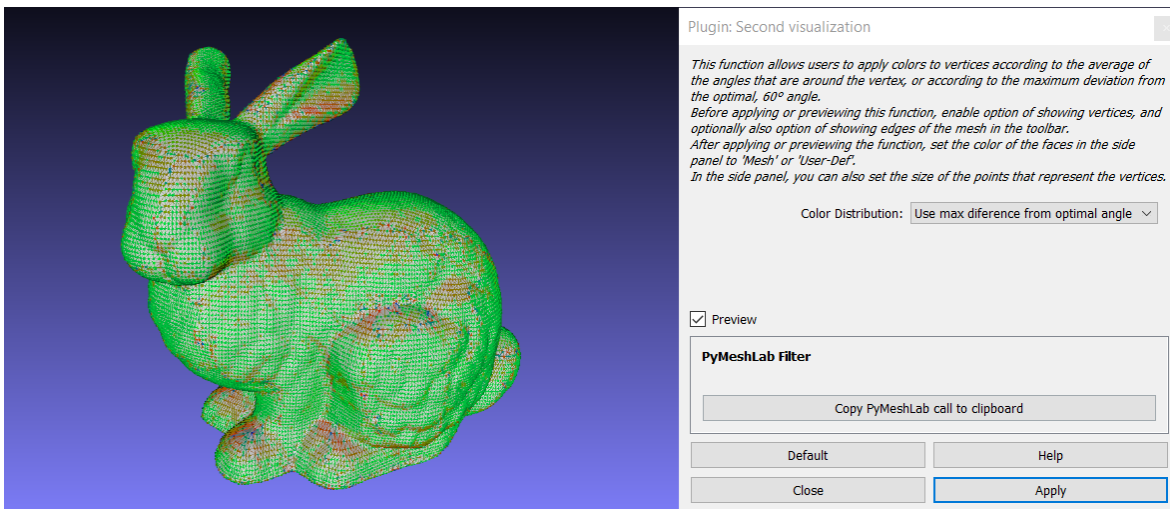


Obr. 3.6: Vizualizácia kombinácie metrík na Stanford Bunny

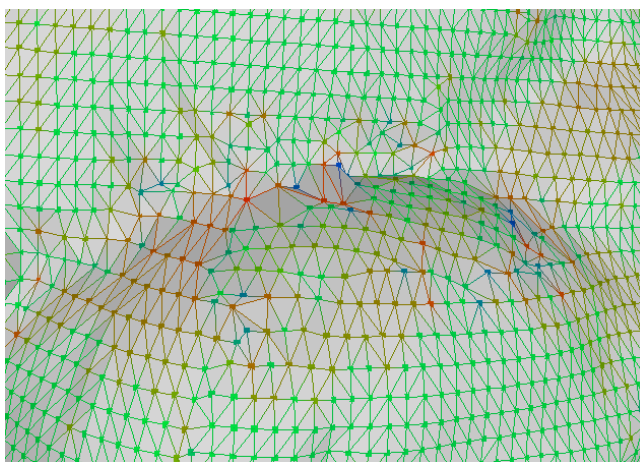
3. príklad - Farebné mapovanie vrcholov na Stanford Bunny:

Na obrázku 3.7 vidíme vizualizáciu modelu Stanford Bunny, ktorá využíva farbenie vrcholov na základe maximálnej odchýlky uhlov v okolí vrchola od optimálneho 60-stupňového uhla. Vizualizácia maximálnej odchýlky nám identifikuje oblasti s príliš veľa ostrými alebo tupými trojuholníkmi. Na obrázku môžeme vidieť veľké oblasti na ušiach, hlave, vrchu tela a labkách s ostrejšími trojuholníkmi, ktoré môžu nepriaznivo pôsobiť na kvalitu.

Tento typ vizualizácie má podobné rozhranie ako predchádzajúce vizualizácie kombinácii metrick. V dialógovom okne však používateľ nájde len jednu nastaviiteľnú položku Color Distribution, v ktorej si môže vybrať, či chce použiť priemer uhlov alebo maximálnu odchýlku od optimálneho uhla pre svoju vizualizáciu. Vizualizácia sa na modeli zobrazí po zapnutí náhľadu (preview) alebo stlačení tlačítka Apply, rovnako ako v predchádzajúcich príkladoch.



Obr. 3.7: Farebné mapovanie vrcholov na Stanford Bunny



Obr. 3.8: Detailný pohľad na farebné mapovanie vrcholov

Na obrázku 3.8 môžeme vidieť detailnejší pohľad na farbenie vrcholov podľa maximálnej odchýlky od optimálneho 60° uhla. Môžeme vidieť, že vrcholy, ktoré majú v okolí iba rovnostranné trojuholníky sú zafarbené na zeleno. Vrcholy s príliš ostrými uhlami sú zafarbené do červena a vrcholy s príliš tupými uhlami sú zafarbené do modra.

3.2.4 Nástroj Metro

Metro bol nástroj vyvinutý na meranie rozdielov medzi dvojicami trojuholníkových sietí (meshov). Metro bol pôvodne malý, open-source príkazový program, ktorý sa stal štandardným nástrojom v oblasti zjednodušovania povrchov.

Metro pracuje na princípe výpočtu vzdialeností medzi zodpovedajúcimi bodmi dvoch 3D objektov. Pomocou tejto metódy dokáže určiť, ako blízko sú jednotlivé vrcholy a hrany jedného modelu k druhému. Výsledkom analýzy je množina numerických hodnôt, ktorá reprezentuje tieto vzdialenosti a možnosť vizualizácie, ktorá farebne označuje oblasti s väčšími odchýlkami.

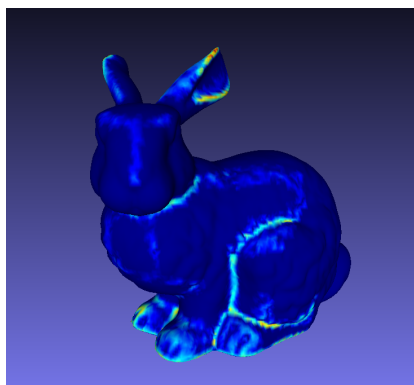
V MeshLab bola funkcionálna Metro integrovaná do filtra Sampling->Hausdorff Distance. Tento filter umožňuje používateľom porovnávať dve 3D siete tým, že vypočíta Hausdorffovu vzdialenosť medzi nimi. V MeshLab sa tento výpočet vykonáva pomocou vzorkovania bodov na povrchu jednej siete a hľadania najbližších bodov na druhej sieti [16], [23].

My tento nástroj použijeme na zobrazenie rozdielov medzi pôvodnou a upravenou 3D sieťou.

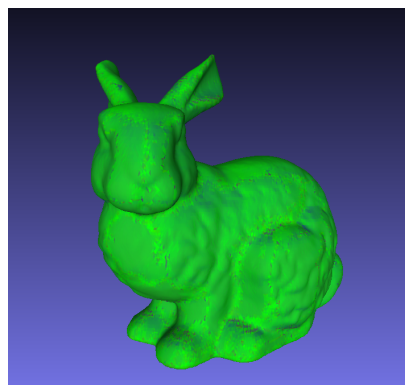
1. príklad - vylepšenie Stanford Bunny:

V časti 3.2.3 v príklade dva sme ukázali, že náš plugin dokázal identifikovať problémové oblasti v okolí uší, spoji tela a stehna a na predných a zadných labkách.

Na tieto časti sme následne aplikovali filter Subdivision Surface - Loop, ktorý slúži na zvýšenie hustoty a zlepšenie kvality povrchu trojuholníkovej siete. Potom sme použili filter Laplacian Smooth, ktorý tieto miesta vyhladil. Ovplyvnené časti môžeme vidieť na obrázku 3.9. Najtmavšie modré časti reprezentujú nezmenený mesh, zatiaľ čo najsvetlejšia farba zobrazuje najviac pozmenené oblasti.



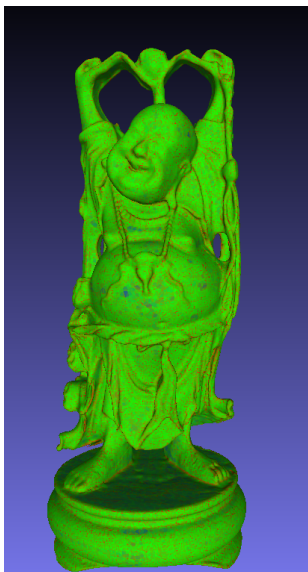
Obr. 3.9: Výstup z filtra Hausdorff Distance



Obr. 3.10: Upravený Stanford Bunny

Na obrázku 3.10 môžeme vidieť vizualizáciu s rovnakým nastavením ako v časti 3.2.3. Po aplikovaní spomenutých filtrov sa celková kvalita výrazne zlepšila, zatiaľ čo veľkosť meshu narástla minimálne. Počet trojuholníkov sa zvýšil o 1503, čo predstavuje nárast o 2,16% a počet vrcholov sa zvýšil o 752, teda o 2,1%.

2. príklad - vylepšenie Happy Buddha:

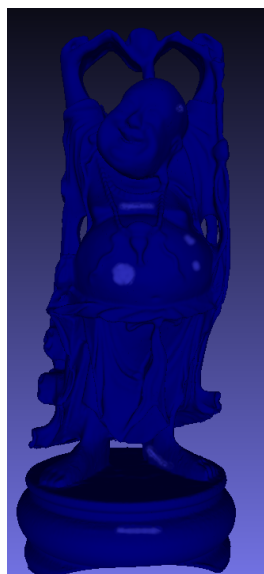


Obr. 3.11: Vizualizácia Happy Buddha

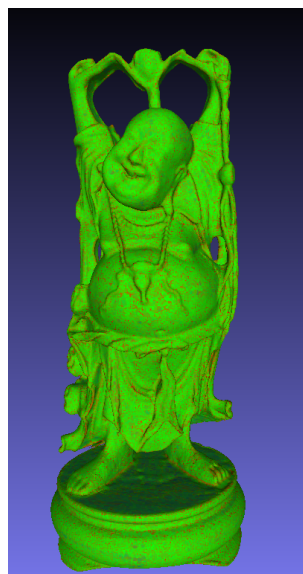
Na obrázku 3.11 môžeme vidieť vizualizáciu kombinácie metriky pomeru polomerov a plošnej metriky. Táto kombinácia na modeli Happy Buddha odhalila problémové oblasti v okolí brucha, hlavy, nôh, chrbta a podstavy.

Tieto oblasti sme rovnako ako v predchádzajúcom príklade ošetrili pomocou filtra Subdivision Surface - Loop a Laplacian Smooth.

Na obrázku 3.12 môžeme vidieť pozmenené oblasti a na obrázku 3.13 výslednú kvalitu upraveného 3D modelu.



Obr. 3.12: Výstup z Hausdorff Distance



Obr. 3.13: Upravený Happy Buddha

Rovnako ako v predchádzajúcom príklade boli zásahy minimálne, pričom sa niektoré ani neprejavili na výstupe filtra Hausdorff Distance. Počet trojuholníkov sa zvýšil o 13 414, čo predstavuje nárast o 1,23% a počet vrcholov sa zvýšil o 6 713, teda sa tiež zvýšil o 1,23%.

V obidvoch príkladoch sme ukázali schopnosť nášho rozšírenia odhaliť problémové miesta na 3D modeloch, ktoré je možné následne odstrániť pomocou minimálnych zásahov vďaka zabudovaným nástrojom v MeshLab. Týmto spôsobom sme ukázali, že MetricCombVis je plnohodnotné rozšírenie, ktoré efektívne zlepšuje kvalitu meshov.

Záver

V závere našej bakalárskej práce môžeme s hrdosťou konštatovať, že sa nám podarilo vytvoriť rozšírenie pre MeshLab, ktoré ponúka približne 300 jedinečných vizualizácií kvality 3D meshov. Tento plugin je zároveň navrhnutý tak, aby bol intuitívny a prispôbil sa rôznym potrebám používateľov.

Jednou z hlavných výhod nášho rozšírenia je jeho jednoduchosť použitia. V prípade použitia najnovšej verzie MeshLabu na operačných systémoch Windows 10 a 11 môže používateľ tento plugin nainštalovať behom niekoľkých sekúnd a bez ďalších zbytočných nastavovaní sa pustiť do vizualizácie svojich 3D modelov. Ďalšiu výhodou je efektívna implementácia nášho rozšírenia. Plugin taktiež dokáže v priebehu niekoľkých sekúnd vizualizovať kvalitu aj veľmi veľkých sietí, ktorých veľkosť presahuje aj 5 miliónov vrcholov a 10 miliónov trojuholníkov. Táto rýchlosť a efektivita sú kľúčovými faktormi, ktoré z nášho pluginu robia atraktívny nástroj.

Naša práca má tiež veľký potenciál pre budúci vývoj. V prípade pokračovania na práci v budúcnosti plánujeme rozšíriť plugin o ďalšie možnosti vizualizácie, ako je napríklad farebné mapovanie hrán, ktoré by poskytovalo ešte detailnejší pohľad na kvalitu 3D meshov. Ďalej by sme chceli navrhnuť a implementovať systém, ktorý by umožňoval používateľom jednoducho a intuitívne pridať vlastné metriky. Tento krok by výrazne zvýšil použiteľnosť nášho nástroja, čím by sa stal ešte cennejším pre širšie spektrum používateľov, od začiatočníkov v oblasti 3D sietí, cez akademikov až po profesionálov.

Veríme, že naša práca predstavuje významný príspevok do projektu MeshLab a že náš plugin bude mať pozitívny vplyv na zlepšenie kvality digitálnych modelov a podporí ďalší výskum a vývoj v oblasti 3D modelovania.

Literatúra

- [1] Inc. ANSYS. Report of mesh statistics, 4 2008. <https://romeo.univ-reims.fr/documents/fluent/tgrid/ug/chp15.pdf>.
- [2] Atlassian. A complete guide to histograms. <https://www.atlassian.com/data/charts/histogram-complete-guide>.
- [3] Mario Botsch, Leif Kobbelt, and Mark Pauly. Polygon mesh processing, 2010. dostupné na <http://www.pmp-book.org/>.
- [4] Cadcam. Mesh quality analysis: Aspect ratio, 7 2023. <https://wiki.cadcam.com.my/knowledgebase/mesh-quality-analysis-aspect-ratio/>.
- [5] Ed Chi. A taxonomy of visualization techniques using the data state reference model. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, pages 69–75, 2000.
- [6] Comsol documentation, the mesh statistics window. https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/comsol_ref_mesh.15.15.html.
- [7] Data visualization. <https://www.tableau.com/learn/articles/data-visualization>.
- [8] ParaView Developers. Paraview usersquied mesh quality, 2020. <https://docs.paraview.org/en/latest/UsersGuide/filteringData.html#mesh-quality>.
- [9] GeeksforGeeks. Polygon mesh in computer graphics, 11 2023. <https://www.geeksforgeeks.org/polygon-mesh-in-computer-graphics/>.
- [10] André Guézic. Surface simplification inside a tolerance volume. Research Report RC20440, IBM Research, 1997.
- [11] Altair HyperMesh. Element quality: How it's calculated, 2022. https://2022.help.altair.com/2022.1/hwdesktop/hm/topics/pre_processing/meshing/element_quality_how_its_calculated_c.htm#element_quality_calculations_hypermesh_r_reference_e1w_q11_z3b__element_quality_calculations_hypermesh_r_dlentry_eh4_dzy_yfb.

- [12] Inscribed & circumscribed right triangles. <https://easingthehurrrysyndrome.wordpress.com/2014/03/14/inscribed-circumscribed-right-triangles/>.
- [13] Luděk Krtička. Contour lines, 10 2022. <https://omapwiki.orienteeering.sport/glossary/contour-lines/>.
- [14] Ben Lutkevich. Techtarget - dynamic link library. <https://www.techtarget.com/searchwindowserver/definition/dynamic-link-library-DLL>.
- [15] Mechead. Mesh quality checking in ansys workbench, 2023. <https://www.mechead.com/mesh-quality-checking-ansys-workbench/>.
- [16] Meshlab. <http://meshlab.sourceforge.net/>.
- [17] Blender org. Blender mesh analysis, 4 2024. https://docs.blender.org/manual/en/latest/modeling/meshes/mesh_analysis.html.
- [18] Paraview. <https://www.paraview.org/>.
- [19] P. Pébay and T. Baker. A comparison of triangle quality measures. *Mathematics of Computation*, 72(244):1817–1839, October 2003.
- [20] Qualitymapper, 2019. <https://stackoverflow.com/questions/58593101/what-does-the-equalizer-do-in-meshlabs-qualitymapperdialog>.
- [21] Jonathan Shewchuk. What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures. 2002. dostupné na <https://people.eecs.berkeley.edu/~jrs/papers/elemj.pdf>.
- [22] Simscale mesh quality, 9 2022. <https://www.simscale.com/docs/simulation-setup/meshing/mesh-quality/>.
- [23] MeshLab Stuff. Measuring the difference between two meshes, 1 2010. <https://meshlabstuff.blogspot.com/2010/01/measuring-difference-between-two-meshes.html>.
- [24] Terramodeler user guide - contours, 2024. <https://terrasolid.com/guides/tmodel/displaycontours.html>.
- [25] The stanford 3d scanning repository. <http://graphics.stanford.edu/data/3Dscanrep/>.
- [26] Veronika Weissová. Analýza a vizualizácia mier kvality nepravidelných trojuholníkových sietí. Diplomová práca, Univerzita Komenského v Bratislave, 2014.

Príloha A: obsah elektronickej prílohy

Elektronická príloha obsahuje zip súbor s:

1. release verziou rozšírenia
2. debug verziou rozšírenia
3. zdrojovými kódmi pluginu
4. readme súbormi
5. vybranými 3D modelmi

Obsah elektronickej prílohy je taktiež možné nájsť na stránke <https://github.com/Martin-SK-PD/Kosovsky-Martin---bachelor-thesis-files> .