VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií
Faculty of Information Technology

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

Brno, 2016 Vojtěch Minařík

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SEGMENTACE POLYGONÁLNÍHO MODELU

POLYGONAL MESH SEGMENTATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE VOJTĚCH MINAŘÍK

AUTHOR

VEDOUCÍ PRÁCE Ing. MICHAL ŠPANĚL, Ph.D.

SUPERVISOR

BRNO 2016

Abstrakt

Tato bakalářská práce se zabývá rozdělením 3D modelů na jednotlivé části reprezentující významové celky (segmenty). Metoda je založena na algoritmu zpracovaném v díle Protrusion-oriented 3D Mesh Segmentation. Realizované řešení je založeno na předpokladu, že 3D modely se skládají z těla a výběžků z těla vyčnívajících. Algoritmus zajišťuje aproximaci těla modelu, nalezení koncových bodů výběžků a rozdělení bodů do jednotlivých segmentů. Pro zajištění operací jako je načtení modelu a práce s jeho komponentami je využita knihovna CGAL [3].

Abstract

This bachelor's thesis deals with 3D model segmentation into particular meaningful segments. Method is based on algorithm described in article Protrusion-oriented 3D mesh segmentation The realized solution is based on assumption, that 3D models consist of a core and core attached protrusions. The segmentation algorithm consists of three main steps: core approximation, selection of salient points which indicate end of protrusion and assignment of points to appropriate segments. Implementation uses library CGAL [3] which is useful for loading and manipulating 3D meshes.

Klíčová slova

Segmentace, segment, síť bodů, model, 3D, polygon, tělo, koncový bod, výběžek, aproximace

Keywords

Segmentation, segment, mesh, model, 3D, polygon, core, salient point, protrusion, approximation

Citace

Minařík Vojtěch: Segmentace polygonálního modelu, bakalářská práce, Brno, FIT VUT v Brně, rok 2016

Segmentace polygonálního modelu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Vojtěch Minařík
13.5.2016

Poděkování

Tímto děkuji vedoucímu bakalářské práce Ing. Michalu Španělovi, Ph.D. za odborné vedení a užitečné rady, které zkvalitnily výslednou práci.

© Vojtěch Minařík, 2016.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

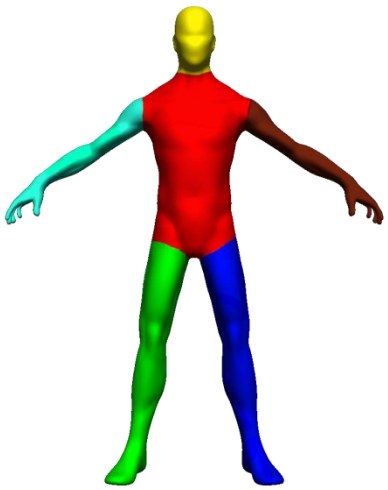
Obsah

1 Úvod ... 3
2 Zpracování a segmentace polygonálních modelů ... 4
2.1 Reprezentativní koncový bod ... 4
2.2 Geodetická vzdálenost ... 4
2.3 Nejkratší cesta ... 4
2.4 Povrchová reprezentace modelu ... 5
2.5 Ukládání polygonálních modelů ... 5
2.6 Formát OFF ... 5
3 Existující metody segmentace modelů ... 7
3.1 Watershed Segmentation ... 7
3.2 Shape Diameter Function ... 8
3.3 Metoda dostupná v knihovně CGAL ... 8
4 Návrh řešení segmentace ... 9
4.1 Popis metody ... 9
4.2 Ohodnocení bodů modelu ... 11
4.3 Výběr koncových bodů výběžků ... 11
4.4 Seskupení koncových bodů ... 11
4.5 Filtrace přebytečných koncových bodů ... 12
4.6 Aproximace těla ... 12
4.7 Rozřazení bodů do výběžků ... 13
4.8 Provedené úpravy algoritmu ... 13
5 Implementace řešení segmentace ... 15
5.1 Využité knihovny ... 15
5.2 Ohodnocení bodů modelu ... 15
5.3 Výběr koncových bodů výběžků ... 17
5.4 Seskupení koncových bodů ... 18
5.5 Filtrace přebytečných koncových bodů ... 20
5.6 Aproximace těla modelu ... 20
5.7 Odstranění malých rozptýlených částí těla ... 23
5.8 Rozřazení bodů do výběžků ... 24
5.9 Speciální implementované funkce ... 25
6 Testování ... 26
6.1 Testovací data ... 26
6.2 Využité nástroje ... 27

6.3 Naměřené hodnoty časové náročnosti segmentace polygonálních modelů 28
6.4 Dosažené výsledky segmentace 29
7 Pokračování do budoucna 31
8 Závěr 32
9 Reference 33
10 Seznam příloh 34
Příloha 1. CD 34

1 Úvod

Segmentace polygonálního 3D modelu může znatelně ulehčit další zpracování modelu tím, že po provedení segmentace lze další kroky provádět pro jednodušší části modelu, popřípadě pouze pro části modelu které nás zajímají. Segmentace modelů nachází využití při dělení modelů získaných 3D skenováním stejně jako v procesu reverzního inženýrství.



Obrázek 1.1: segmentovaný model.

Cílem této práce je implementovat program provádějící segmentaci polygonálních 3D modelů. Segmentací polygonálního modelu rozumíme rozdělení modelu na menší části reprezentující nějaký smysluplný objekt. Výsledný segmentovaný model může mít v ideálním případě podobu modelu na obrázku 1.1.

Zde popisovaná implementace vychází z algoritmu segmentace polygonálních modelů uvedeného v článku Protrusion-oriented 3D mesh segmentation [1]. U tohoto algoritmu však byly provedeny úpravy metody určení hranice těla modelu a výběžků.

Při realizaci řešení se podařilo dosáhnout uspokojivých výsledků segmentace modelů na menší významově samostatné (oddělitelné) části. Realizované řešení poskytuje nejlepší výsledky právě pro modely odpovídající myšlence, že se skládají z těla a výběžku k němu připojených. Do takové skupiny se nejčastěji řadí model lidí a zvířat.

Celá práce je rozdělená do několika kapitol. Z nichž druhá se zabývá teoretickými znalostmi nezbytnými pro pochopení metody segmentace polygonálního modelu. Kapitola číslo 3 se věnuje současnému stavu a existujícím řešením. Další kapitola (4) popisuje podrobně implementovanou metodu. Následují kapitoly věnující se popisu dosažených výsledků, testování, uživatelské rozhraní programu.

2 Zpracování a segmentace polygonálních modelů

Cílem segmentace polygonálních modelů je rozložení velkých, složitých modelů na menší jednodušší části. Jednodušší části usnadňují další zpracování.

2.1 Reprezentativní koncový bod

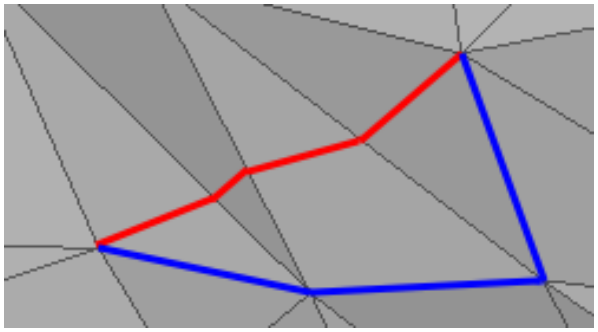
Reprezentativní koncový bod je bod s nejvyšším ohodnocením, který zastupuje skupinu koncových bodů. Výběr takových bodů je popsán v kapitolách 4.3 a 4.4. Reprezentativní koncový bod pak označuje konec výběžku modelu. Poměr mezi počtem reprezentativních koncových bodů a počtem výběžků modelu musí být 1:1.

2.2 Geodetická vzdálenost

Geodetická vzdálenost dvou bodů je délka nejkratší cesty mezi dvěma body na povrchu modelu. Cesta nemusí využívat pouze body a hrany modelu, ale může vést po povrchu stěny modelu. Geodetická vzdálenost je znázorněna na obrázku 2.1 červenou barvou.

2.3 Nejkratší cesta

Nejkratší cesta na rozdíl od geodetické vzdálenosti využívá pouze hrany a body modelu. Cesta je vybrána tak, aby byla co možná nejkratší. K tomu slouží například Dijkstrův algoritmus. Nejkratší cesta je vyznačena modrou barvou na obrázku 2.1.



Obrázek 2.1: porovnání geodetické vzdálenosti (červená) a nejkratší cesty (modrá).

2.4 Povrchová reprezentace modelu

K uchování informací o tvaru modelu je třeba tento tvar popsat. Reprezentace modelů můžeme rozdělit na objemovou a povrchovou reprezentaci. Záleží na typu modelu a na požadovaných operacích, které musí model umožňovat.

Objemová reprezentace modelu poskytuje pouze informace o vnitřní struktuře modelu, ale neposkytuje pouze informace o povrchu modelu.

Povrchová reprezentace modelů naopak poskytuje pouze informace o povrchu modelu. Takové modely jsou nejčastěji ukládány jako množina bodů a polygonálních ploch nebo hran. Model dále může obsahovat informace o barvě nebo textuře povrchu.

2.5 Ukládání polygonálních modelů

Existuje mnoho formátů pro ukládání polygonálních modelů. Ukládání modelů využívá formáty od *plain text* (prostý text), například formáty *obj* a *off* až po sofistikovanější metody, jako je formát STL využívající strukturovaného zápisu informací. U polygonálních modelů se ukládá množina bodů model a množina hran nebo ploch.

2.6 Formát OFF

Formát OFF (Object File Format) využívá formu prostého textu pro ukládání polygonálních modelů a je označován koncovkou *.off*. Tento formát nevyužívá žádnou speciální strukturu zápisu informací popisujících model, pouze každý objekt je zapsán na jeden řádek.

Soubor musí obsahovat text "OFF" bez uvozovek na prvním řádku, na druhém řádku počet vrcholů následovaný počtem polygonálních ploch a počtem hran modelu. Následuje výpis souřadnic jednotlivých bodů. Souřadnice všech bodů jsou uloženy ve tvaru `x y z` … a jsou od sebe odděleny znakem konce řádku. Hodnoty souřadnic mohou být jak celá čísla, tak i čísla s plovoucí desetinnou čárkou (implementace využívá přesnost datového typu `double`). Plochy modelu se ukládají ve formátu `n i j k`, kde `n` označuje počet vrcholů plochy, `i, j` a `k` označují indexy řádků, na kterých jsou uloženy body určující danou plochu (v tomto případě trojúhelníkovou plochu). Záznam plochy může obsahovat záznam udávající informace o barvě. Barvy jsou zaznamenány ve formátu `r g b` α, kde `r, g, b` reprezentují příslušné barevné složky aditivního barevného modelu a α určuje alfa složku barvy.

Příklad souboru ve formátu OFF pro model krychle:

```
OFF
8 6 0
-1.00000 -1.00000 1.00000
1.00000 -1.00000 1.00000
-1.00000 1.00000 1.00000
1.00000 1.00000 1.00000
-1.00000 1.00000 -1.00000
1.00000 1.00000 -1.00000
-1.00000 -1.00000 -1.00000
1.00000 -1.00000 -1.00000
4 0 1 3 2
4 2 3 5 4
4 4 5 7 6
4 6 7 1 0
4 1 7 5 3
4 6 0 2 4
```

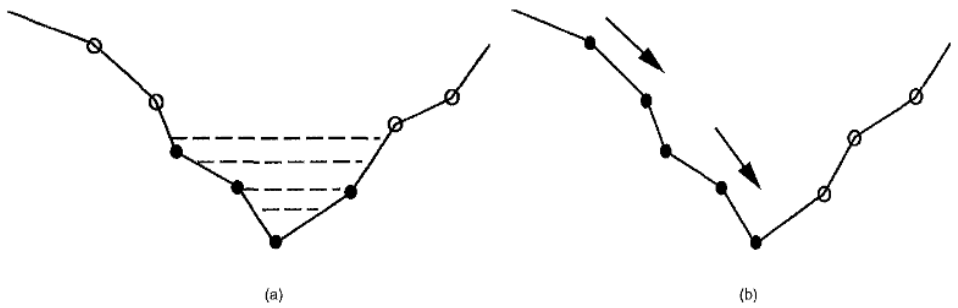
Ukázkový model je tvořen čtyřhrannými plochami. Proto je tento model čistě ilustrativní a nelze je použít v algoritmu segmentace, který počítá s modely tvořenými trojúhelníkovou sítí.

3 Existující metody segmentace modelů

V současné době existuje mnoho různých řešení tohoto problému využívajících různé přístupy k dosažení co možná nejlepšího výsledku. Metody segmentace využívají širokého rozsahu principů od využití vlastností povrchu modelu, jako jsou zakřivení, velikost jednotlivých polygonálních ploch atd., přes prokládání modelů geometrickými tvary až ke tvorbě vnitřní kostry modelu.

3.1 Watershed Segmentation

Tato metoda je popsána v článku Partitioning 3D Surface Meshes Using Watershed Segmentation [6]. Algoritmus této metody se zaměřuje na rozdělení povrchu modelu na konvexní části. Metoda umožňuje dva postupy, zdola nahoru a shora dolů, viz obrázek 3.1.



Obrázek 3.1: strategie algoritmu, a) postup metody zespoda nahoru, b) postup shora dolů, zdroj [6].

Princip algoritmu lze popsat kroky [6]:

1. výpočet křivosti pro každý bod
2. vyhledání lokálních minim a přiřazení unikátních značek
3. hledání všech rovinných ploch a jejich klasifikace jako minimum nebo jako plošinu
4. cyklické klesání všech plošin dokud se nenarazí na oblast s přidělenou značkou
5. všechny neoznačené vrcholy klesají, dokud nenarazí na označenou oblast
6. spojení oblastí, jejichž hloubka je nižší než přednastavená prahová hodnota

3.2 Shape Diameter Function

Tato metoda je popsána v dokumentaci knihovny CGAL [7]. SDF hodony se vypočítávají pro všechny plochy modelu. Pro každou plochu je vytvořeno několik polopřímek. Polopřímky jsou sestrojeny pomocí kuželů vytvořených pro každou plochu tak, že střed plochy je využit vrchol kužele a vnitřní normála plochy jako jeho osa. Polopřímky leží ve stěně kužele. Následně jsou polopřímky ukončeny v prvním průsečíku s protější stěnou modelu. Tím vzniknou úsečky, jejichž průměrná délka dává výslednou hodnotu SDF.

Následně jsou doplněny hodnoty pro plochy, které stále nemají SDF. Doplněné hodnoty odpovídají hodnotě SDF pro sousední plochy.

Dalším krokem je aplikování bilaterálního vyhlazování hodnot. Konečná podoba segmentovaného modelu se získá pomocí měkkého shlukování algoritmem *k-means* upraveným tak, aby bral v úvahu hodnoty SDF. Tento algoritmus se opakuje ve více iteracích a následně se vybere nejlepší výsledek.

3.3 Metoda dostupná v knihovně CGAL

Před započetím implementace vlastního algoritmu proběhly experimenty s řešením segmentace obsaženým v knihovně CGAL [7]. Princip této metody je popsán v předchozí kapitole (3.2). Metoda je obsažena i ve finální implementaci a je přístupná zadáním parametru *-d*.

Provedené experimenty výrazně přispěly pochopení problematiky segmentace modelů a poskytlo představu o možné podobě výsledné segmentace. Metoda se ukázala jako velmi rychlý způsob segmentace modelu. V porovnání s vlastní implementací dosahuje několikanásobně nižší časové složitosti. Ukázka dosažených výsledků je znázorněna na obrázku 3.3.1.

Obrázek 3.3.1: příklady segmentace implementované v knihovně CGAL.

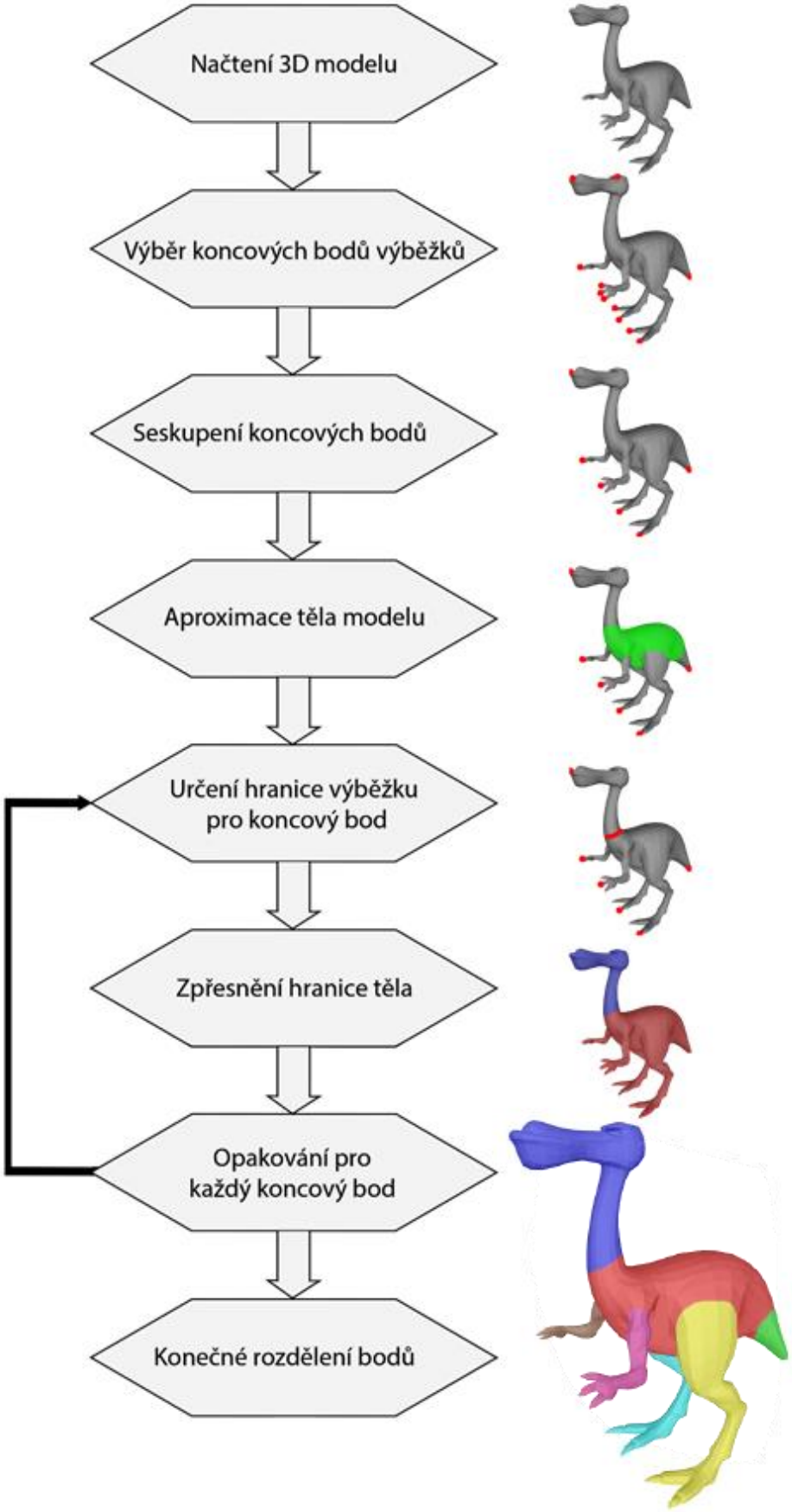
4 Návrh řešení segmentace

Cílem této kapitoly je přiblížit čtenáři zásadní principy metody řešení a způsob jejich realizace. Mezi hlavní části řešení patří detekce koncových bodů výběžků, aproximace těla modelu a přiřazení bodů příslušným částem modelu. Tento postup byl převzat ze článku Protrusion-orineted 3D mesh segmentation [1].

4.1 Popis metody

Cílem segmentace je rozdělení modelu na menší části reprezentující nějaké smysluplné celky. Problém segmentace modelu je rozdělen na několik menších úkolů: ohodnocení všech bodů hodnotící funkcí popsanou v podkapitole 4.2, seskupení blízkých koncových bodů, aproximace těla (jádra) modelu a konečné rozdělení bodů do jednotlivých částí. Schéma algoritmu si lze prohlédnou na obrázku 4.1.1.

Využití knihovny CGAL [3] spolu s návrhem metody vyžadují, aby formát vstupního souboru byl .off a všechny plochy modelu měly podobu trojúhelníku.



Obrázek 4.1.1: kroky algoritmu segmentace [1].

4.2 Ohodnocení bodů modelu

Po načtení modelu je třeba ohodnotit každý bod v modelu pomocí speciální funkce. Tato funkce vypočítá postupně pro každý bod modelu hodnotu, která mu zůstane přiřazena po zbytek procesu segmentace. Hodnotící funkce nevyužívá přímo informace o bodu, jako jsou jeho souřadnice, zakřivení povrchu (konkávnost nebo konvexnost) nebo počet hran z něho vycházejících. Hodnocení využívá informace o členitosti a struktuře povrchu v okolí bodu.

4.3 Výběr koncových bodů výběžků

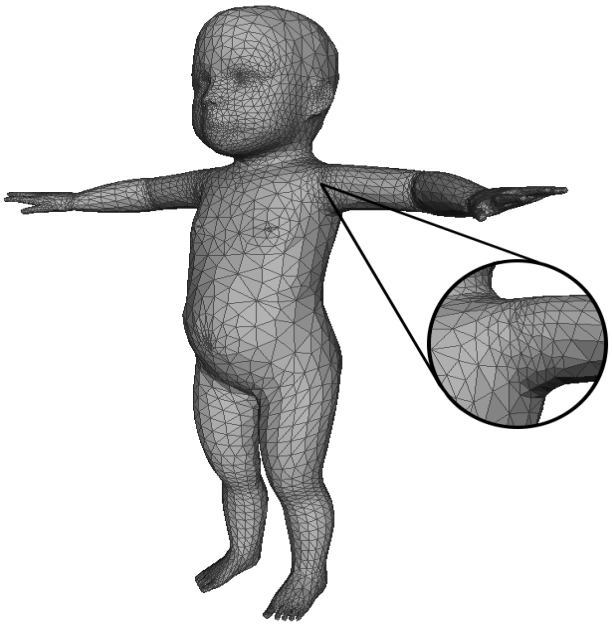
Po ohodnocení všech bodů jsou vybírány body, které budou označeny jako koncové body jednotlivých výběžků modelu. Tyto body jsou vybírány v závislosti na jejich ohodnocení. Aby byl bod vybrán jako koncový bod výběžku, musí být jeho ohodnocení vyšší než prahová hodnota. Takto je vybrána množina všech koncových modelů. Zde ještě neplatí nutnost poměru mezi počtem koncových bodů a počtem výběžků 1:1

4.4 Seskupení koncových bodů

Po výběru koncových bodů není zajištěna podmínka určující, že každý výběžek má právě jeden koncový bod. Tento krok je velmi důležitý, protože aby byla splněna hlavní myšlenka této metody, že model se skládá z těla a z výběžků z něho vyčnívajících ukončených vždy právě jedním koncovým bodem. Pokud by jeden výběžek obsahoval více než jeden bod, viz obr. 4.4.1, byla by narušena funkčnost metody, protože by bylo třeba rozdělit dvě části výběžku od sebe pomocí bodů přiřazených tělu modelu. Tím by došlo k závažné chybě provedení algoritmu. Problém kdy je opravdu nezbytné rozdělit jeden výběžek na více menších části, by bylo možné vyřešit rekurzivním použitím metody na jednotlivé segmenty vytvořené během předchozího kroku. Tato funkcionalita však není v současnosti zahrnuta v implementaci.

4.5 Filtrace přebytečných koncových bodů

Tento krok není nezbytný pro každý model, ale existují modely, které mají výrazně vyšší hustotu sítě bodů a hran i v oblastech, které nepatří do žádného výběžku. Výskyt takových oblastí se nejčastěji nachází v místech kolem kořene výběžku, obzvláště v oblastech kde se vyskytují hranice dvou výběžku v těsné blízkosti od sebe. Příklad tohoto jevu znázorňuje obrázek 4.5.2. Poněvadž takové oblasti nemohou patřit do žádného výběžku, naopak protože výběžky jsou od sebe vždy odděleny linií o šířce alespoň jednoho bodu přiřazeného tělu modelu, jsou vytříděny a již nejsou dále označeny jako koncové body.



Obrázek 4.5.2: Příklad modelu s neideálním rozložením bodů.

Z množiny reprezentativních koncových bodů jsou vyřazeny ty body, které leží přímo na nejkratší cestě mezi dvěma dalšími koncovými body, nebo v její těsné blízkosti.

4.6 Aproximace těla

V tomto kroku již máme jednoznačně určeny koncové body výběžků, ale stále není určena žádná hranice, která by je od sebe oddělovala. Tato hranice vzniká během aproximace těla, kdy se vytváří speciální část modelu, která neobsahuje žádný koncový bod. Tělo modelu je definováno jako středová část modelu oddělující všechny výběžky od sebe.

Protože se v této etapě řešení zpracovávají všechny body modelu, musí se zavést pravidlo jejich uspořádání v seznamu bodů, které umožní jejich výběr ve správném pořadí. Kdybychom nechali body v původním pořadí, nebylo by možné správně provést aproximaci těla modelu. Jako pravidlo

uspořádání se využívá seřazení všech bodů modelu podle jejich ohodnocení od nejmenšího k nejvyššímu. Vzestupné pořadí je určeno myšlenkou, že tělo jakožto hlavní část modelu nedisponuje členitostí srovnatelnou se členitostí výběžků. Díky nižší členitosti svého povrchu tělo obsahuje polygonální plochy s největším obsahem, nejdelšími hranami a hustota jeho bodů je nízká.

Princip aproximace těla spočívá v procházení seřazeného vektoru všech bodů a jejich postupné přidávání k tělu. Před zahájením se všem koncovým bodům nastaví příznak aktivity `active` na hodnotu `true` a přiřadí se jim hodnoty značící číslo segmentu. Čísla segmentů jsou přidělována z rozsahu 1 až *n*, přičemž *n* označuje počet všech reprezentativních koncových bodů výběžků modelu. Zvláštním případem je číslo segmentu 0, které je vyhrazeno pro body náležící tělu modelu. Proto se body musí inicializovat na zápornou hodnotu.

4.7 Rozřazení bodů do výběžků

V tomto kroku se pouze přiřazují body do jednotlivých částí s výjimkou těla modelu, které již nabývá své finální podoby po krocích popsaných v kapitole 4.6. Z algoritmu aproximace těla modelu vyplývá, že všechny výběžky od sebe musí být vždy odděleny linií bodů náležících tělu modelu o šířce alespoň jednoho bodu.

4.8 Provedené úpravy algoritmu

Metoda segmentace jako taková je převzata z díla Protrusion orinted 3D mesh segmentation [1]. V algoritmu však byly provedeny úpravy za účelem dosažení požadovaného výsledku.

Mezi hlavní úpravy patří zjednodušení algoritmu určení, vyhlazení hranice mezi jednotlivými výběžky modelu a tělem modelu. V původní podobě algoritmu se vyhlazení hran těla provádí výpočtem nejkratších cest mezi všemi reprezentativními koncovými body a všemi body povrchu modelu. K tomu se využívá Dijkstrova algoritmu a cena (ohodnocení) hran mezi body *u* a *v* je určena vzorcem:

$$cena(u,v) = \delta \frac{délka(u,v)}{prům_délka} + (1 - \delta) \frac{prot(u,v)}{prům_prot} \tag{4.8.1}$$

kde *prot(u,v)* je rovno absolutní hodnotě rozdílu ohodnocení bodů *u* a *v* popsaného v kapitole 4.2, *prům_délka* označuje průměrnou vzdálenost mezi všemi dvojicemi bodů povrchu modelu a *prům_prot* je průměrná hodnota prototypu všech hran povrchu modelu. δ je nastavena na hodnotu 0,4. Nechť *D*, je funkce délky přiřazená každému reprezentativnímu koncovému bodu. Pak je hranice dvou částí modelu vytvořena interpolací povrchu modelu iso-konturou vytvořenou pomocí konstantní prahové hodnoty D_c.

Iso-kontura C_{Dc} protíná hranu modelu, pokud je splněna nerovnice 4.8.2:

$$(\boldsymbol{D}(\boldsymbol{u}) - \boldsymbol{D_c}) \times \left(\boldsymbol{D_c} - \boldsymbol{D}(\boldsymbol{v})\right) > \boldsymbol{0} \tag{4.8.2}$$

a průsečík je:

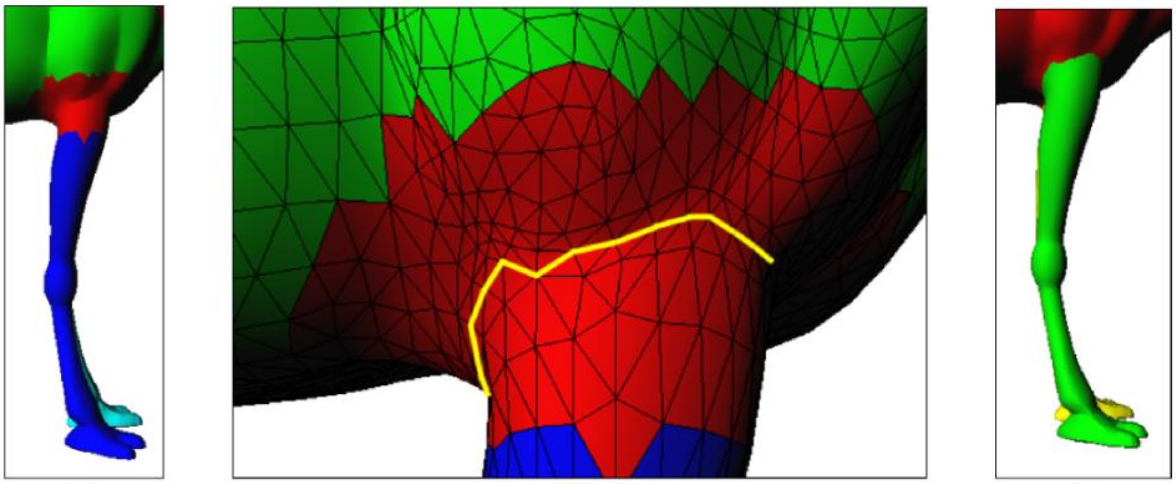
$$\boldsymbol{v_{int}} = (\boldsymbol{1} - \boldsymbol{\lambda})\boldsymbol{u} + \boldsymbol{\lambda v} \tag{4.8.3}$$

kde platí:

$$\boldsymbol{\lambda} = \frac{\boldsymbol{D_c} - \boldsymbol{D}(\boldsymbol{u})}{\boldsymbol{D}(\boldsymbol{v}) - \boldsymbol{D}(\boldsymbol{u})} \tag{4.8.4}$$

Vybráním všech trojúhelníkových ploch, jejichž stranu tvoří průsečná hrana, je možné zkonstruovat aproximaci této iso-kontury. Tato aproximace je v převážné většině případů hranicí výběžku modelu, zdroj [1].

Hranice modelu je nyní tvořena širokým pásem trojúhelníkových ploch, viz obrázek 4.8.1(a), v okolí reálné hranice. Reálná hranice je získána jako minimální řez tímto pásem ploch, obrázek 4.8.1(b). 4.8.1(c) ukazuje konečný výsledek.



(a) (b) (c)

Obrázek 4.8.1: kroky nalezení hranice výběžku, zdroj [1].

Řešení problému hranic výběžků ve vlastní implementaci je popsáno v kapitole 4.6.

Další úpravou oproti originálnímu popisu algoritmu [1] je seskupení koncových bodů výběžků a výběr reprezentativního koncového bodu pro každou skupinu. Tento algoritmus je založen na stejném principu, avšak v implementovaném řešení je využit ten stejný mechanismus ve dvou iteracích. Tento přístup se osvědčil během testování na identických modelech, jako jsou použity na ukázku v originálním článku [1].

5 Implementace řešení segmentace

Tato kapitola přibližuje detailní postup realizace segmentace polygonálních modelů. Teoretický popis této metody se nachází v kapitole 4. Jako implementační jazyk byl zvolen c++, protože poskytuje dostatečný počet knihoven pro práci s 3D modely.

5.1 Využité knihovny

Protože je zde nutná složitá manipulace s modelem, využívá se knihovna CGAL verze 4.7. Knihovna umožňuje kromě načtení modelu operace, které mají velmi dobré využití v této metodě, jako jsou funkce pro nalezení nejkratší cesty mezi dvěma body modelu. Dále se využívá knihovna *boost* a standardní knihovny jazyka *c++*.

5.2 Ohodnocení bodů modelu

Po načtení modelu je třeba ohodnotit každý bod v modelu pomocí speciální funkce. Tato funkce vypočítá postupně pro každý bod modelu hodnotu, která mu zůstane přiřazena po zbytek procesu segmentace. Hodnotící funkce nevyužívá přímo informace o bodu, jako jsou jeho souřadnice, zakřivení povrchu (konkávnost nebo konvexnost) nebo počet hran z něho vycházejících. Hodnocení využívá informace o členitosti a struktuře povrchu v okolí bodu.

Hlavní roli hraje nejkratší cesta mezi bodem a body v jeho okolí. Pro každý bod modelu je proto nezbytné zjistit nejkratší cestu ke všem ostatním bodům modelu. Zde se objevuje riziko vysoké redundance vypočítávání již známých hodnot. Implementace tento problém řeší zavedením pomocného bufferu pro ukládání již získaných hodnot. Délky nejkratších cest se ukládají do dvourozměrného pole na polohu určenou pořadím koncových bodů cesty v seznamu všech bodů modelu. Tím je eliminováno opakované vypočítáváni nejkratší cesty pro jednu dvojici bodů, nicméně stále není vyřešen výpočet nejkratších cest mezi stejnými body avšak opačným směrem. Proto jsou vypočtené hodnoty ukládány nejen na příslušné souřadnice bufferu, ale také na souřadnice převrácené.

Pomocný buffer implementuje třída `distbuffer`. Tato třída také řeší problém zápisu délky pro oba směry nejkratší cesty.

K výpočtu nejkratších cest jsou využity prostředky, které poskytuje knihovna CGAL verze 4.7 [3], dřívější verze knihovny tyto funkce nepodporují. Nejkratší cesta mezi dvěma body je nalezena pomocí Dijkstrova algoritmu. Protože je nutné nalézt nejkratší cestu (zjistit její délku), jedná se o časově i výpočetně nejnáročnější část celého algoritmu segmentace.

Hodnotící funkce využívá hodnoty uložené v bufferu k ohodnocení všech bodů modelu. Body jsou hodnoceny jeden po druhém, přičemž ohodnocení bodu je rovno součtu délek nejkratších cest ke všem bodům nalézajícím se v okolí bodu. Do okolí aktuálního bodu spadají všechny ostatní body, jejichž délka nejkratší cesty k aktuálnímu bodu je menší nebo rovna maximální uvažované vzdálenosti d_{max}. Pro d_{max} platí vztah:

$$\boldsymbol{d_{max}} = \sqrt{\mathbf{5} \times \mathbf{10^{-3}} \times \boldsymbol{S(M)}} \tag{5.2.1}$$

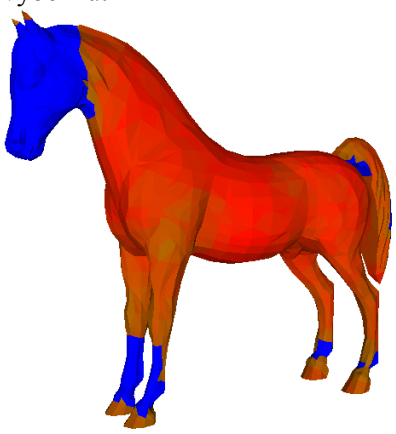
kde d_{max} je maximální vzdálenost bodu, který je brán v potaz při výpočtu ohodnocení bodu, M je vstupní model určený k segmentaci a $S(M)$ je celkový povrch modelu M (součet obsahů všech stěn modelu).

Pro výpočet hodnocení jednotlivých bodů pak platí, že hodnota je rovna součtu nejkratších cest ke všem bodům v blízkém okolí aktuálního bodu a lze jej vyjádřit vtahem:

$$\boldsymbol{h(v)} = \sum_{\boldsymbol{u_i} \in \boldsymbol{S}} \boldsymbol{g(v, u_i)} \tag{5.2.2}$$

kde v je bod, pro který je právě vypočítávána hodnotící funkce, u_i je iterátor přes všechny body povrchu modelu spadajících do okolí bodu v, g je nejkratší geodetická vzdálenost mezi body v a u_i, která však musí být menší než d_{max} , aby mohla být započítána do výsledné hodnoty. Tento postup se opakuje pro všechny body modelu.

Ze vztahu vyplývá, že závislost hodnocení bodu na hustotě bodů a hran v jeho okolí do maximální uvažované vzdálenosti je exponenciální. Na ohodnoceném modelu je patrný výrazný nárůst ohodnocení v oblastech, kde končí výběžky modelů, a hodnoty blížící se celkovému minimu v oblasti těla (jádra) modelu (modrá barva označuje koncové body). Tento jev je patrný na obrázku 5.2.1, který znázorňuje rozložení hustoty hodnocení bodů celého modelu. Obrázek také naznačuje reálný výskyt koncových bodů před jejich seskupováním na reálném modelu. Lze snadno zpozorovat, že rozložení nedosahuje kvality vyobrazené na ilustračních obrázcích, kde se nejvýše ohodnocené body modelu opravdu nachází na samém konci výběžků.



Obrázek 5.2.1: hustota hodnotící funkce, modrá barva označuje koncové body.

5.3 Výběr koncových bodů výběžků

Po ohodnocení všech bodů jsou vybírány body, které budou označeny jako koncové body jednotlivých výběžků modelu. Tyto body jsou vybírány v závislosti na jejich ohodnocení. Aby byl bod vybrán jako koncový bod výběžku, musí být jeho ohodnocení vyšší než prahová hodnota vypočtená ze vztahu 5.3.1.

$$\boldsymbol{h_t = \big(avg(v \in S) + \mu(v \in S)\big) \div 2} \tag{5.3.1}$$

kde h_t je prahová hodnota pro výběr koncových bodů (bod je vybrán jako koncový pokud je jeho hodnota hodnotící funkce vyšší též tato prahová hodnota), $v \in S$ jsou všechny body modelu, potom $avg(v \in S)$ značí průměrnou hodnotu ohodnocení a $\mu(v \in S)$ střední hodnotu ohodnocení všech bodů povrchu modelu. Takto lze zpracovávat pouze model, jehož hustota sítě povrchu je homogenně rozložena po celém povrchu modelu. Ve výjimečných případech kdy se hustota bodů koncentruje do jedné oblasti na úkor hustoty sítě ve zbytku modelu, existuje možnost zadat prahovou hodnotu pro výběr koncových bodů pomocí parametru `-t x`, kde *x* je celé číslo v uzavřeném intervalu od 0 do 100 a určuje procentuální pozici prahové hodnoty v celém oboru hodnot hodnotící funkce, které nabývá pro všechny nody modelu. Výsledná prahová hodnota je pak vyjádřena rovnicí:

$$\boldsymbol{h_t = \big(\max\big(h(v)\big) - \min\big(h(v)\big)\big) \times \frac{x}{100} + \min\big(h(v)\big)} \tag{5.3.1}$$

kde $max\big(h(v)\big)$ je maximální hodnota hodnotící funkce pro všechny body modelu, $min\big(h(v)\big)$ je minimální hodtona hodnotící funkce pro všechny body modelu. Ze vzorce je zřejmé, že manuálně zadaná hodnota `x` musí být vhodně zvolena. V případě zvolení hodnoty z krajních oblastí intervalu nemůže dojít ke správnému rozdělení modelu. V této situaci vznikají dvě možnosti:

1. zadaná hodnota je příliš nízká což způsobí výběr koncových bodů po celé ploše modelu a následné zredukování na pouze jeden koncový bod. Tím dojde k zanedbání přítomnosti všech výběžků, kromě výběžku, který obsahoval bod s nevyšším ohodnocením

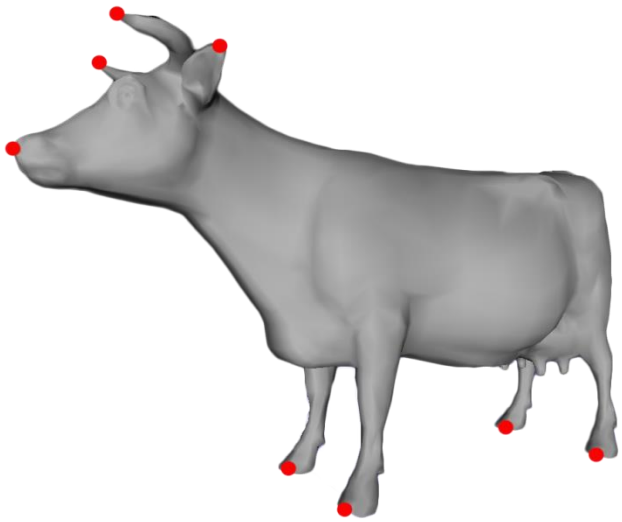
2. zadaná hodnota je příliš vysoká a nedojde ke spolehlivému nalezení všech koncových bodů výběžků, a proto mohou být některé výběžky zanedbány

3. teoreticky může existovat model, u něhož bude třeba extrémních hodnot, ale výskyt takového modelu je vysoce nepravděpodobný. Kvůli této možnosti nelze vstupní hodnotu omezit na menší rozsah, aby byl zajištěn kvalitní výstup při hodnotách z celého intervalu pro většinovou část modelů.

Předchozím postupem je vybrána množina koncových bodů, které však stále neindikují přítomnost výběžku jednoznačně, protože na jednom výběžku může být více koncových bodů, viz obrázek 2.1. Proto je třeba seskupit blízké koncové body a vybrat z nich jeden s nejvyšším ohodnocením, čímž je dosaženo požadovaného poměru počtu koncových bodů a počtu výběžků 1:1.

5.4 Seskupení koncových bodů

Po výběru koncových bodů není zajištěna podmínka určující, že každý výběžek má právě jeden koncový bod. Tento krok je velmi důležitý, protože aby byla splněna hlavní myšlenka této metody, že model se skládá z těla a z výběžků z něho vyčnívajících ukončených vždy právě jedním koncovým bodem. Pokud by jeden výběžek obsahoval více než jeden bod, viz obr. 5.4.1, byla by narušena funkčnost metody, protože by bylo třeba rozdělit dvě části výběžku od sebe pomocí bodů přiřazených tělu modelu. Tím by došlo k závažné chybě provedení algoritmu. Problém, kdy je opravdu nezbytné rozdělit jeden výběžek na více menších části, by bylo možné vyřešit rekurzivním použitím metody na jednotlivé segmenty vytvořené během předchozího kroku. Tato funkcionalita však není v současnosti zahrnuta v implementaci.



Obrázek 5.4.1: model bez seskupených koncových bodů

Aby nedocházelo k nekonzistentnímu chování metody je třeba koncové body seskupit a zredukovat jejich množství na počet odpovídající počtu o jedničku nižšímu než je celkový počet částí modelu. Seskupeny a následně sloučeny mohou být body nacházející se velmi blízko u sebe. Prahová hodnota vzdálenosti bodů, ve které je ještě možné body seskupit:

$$\boldsymbol{T_s} = \frac{\sum_{i=1}^{N_s-1} \sum_{j=i+1}^{N_s} g(s_i, s_j)}{N_s(N_s-1)} \tag{5.4.1}$$

kde N_s značí počet všech koncových bodů a $g(s_i, s_j)$ značí geodetickou vzdálenost mezi body s_i a s_j. Skupinu koncových bodů C potom lze definovat:

$$\boldsymbol{C = \{s_i \in S : \forall s_k \in C, g(s_i, s_j) < T_s\}} \tag{5.4.2}$$

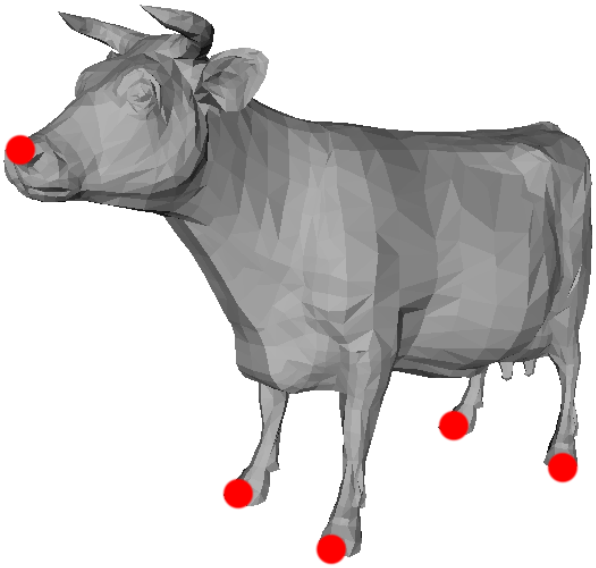
Když jsou body seskupeny podle vzájemných vzdáleností, vybere se reprezentativní koncový bod pro každou skupinu. Vybírá se bod s nejvyšším ohodnocením:

$$\boldsymbol{Rep(C) = \{s_i \in C : pf(s_i) > pf(s_k), \forall s_k \in C\}} \tag{5.4.3}$$

kde $Rep(C)$ reprezentativní bod skupiny, $pf(s_i)$ je hodnotící funkce pro bod s_i a $pf(s_k)$ je hodnotící funkce pro bod s_k.

Nicméně tento krok musel být oproti popisu metody segmentace popsané ve článku Protrusion-oriented 3D mesh segmentaion [1] pozměněn. V původní podobě metody se tento krok provádí pouze jednou, ale opakovanými testy byla zjištěna vyšší efektivita algoritmu v případě, kdy se celý krok zopakuje dvakrát po sobě.

Výsledkem úspěšného seskupení modelu a výběru jednoho reprezentativního bodu pro každou skupinu je model obsahující právě jeden koncový bod pro každý výběžek, viz obrázek 5.4.2. Tyto body nazýváme reprezentativní koncové body.



Obrázek 5.4.2: Model s reprezentativními koncovými body.

5.5 Filtrace přebytečných koncových bodů

Implementace provádí odstranění přebytečných bodů z množiny reprezentativních koncových bodů, jsou vyřazeny ty body, které leží přímo na nejkratší cestě mezi dvěma dalšími koncovými body, nebo v její těsné blízkosti. Ověření, zda má být bod smazán, se řídí výpočtem nerovnice 5.4.4.

$$\left| \left(\boldsymbol{g}(\boldsymbol{s_i}, \boldsymbol{s_k}) + \boldsymbol{g}(\boldsymbol{s_k}, \boldsymbol{s_j}) \right) - \boldsymbol{g}(\boldsymbol{s_i}, \boldsymbol{s_j}) \right| \leq \mathbf{acc} \tag{5.4.4}$$

kde s_k označuje právě kontrolovaný bod, s_i a s_j jsou iterátory přes všechny koncové body modelu kromě bodu s_k, $g(s_i, s_k)$ značí nejkratší cestu mezi body s_i a s_k, *acc* značí odchylku od délky nejkratší cesty mezi dvěma body, která určuje hraniční vzdálenost pro mazání koncových bodů. Přesnost v implementaci byla stanovena jako čtyřnásobek délky nejkratší cesty mezi dvěma body s nejmenší geodetickou vzdáleností. Tuto hodnotu lze upravit pomocí parametru `-a x`, kde *x* je hodnota určující řád násobku nejkratší cesty mezi dvěma nejbližšími body.
Pokud je podmínka z rovnice splněna, bod je odstraněn ze seznamu koncových bodů. Tento postup se opakuje pro všechny koncové body modelu. Tato operace upraví seznam koncových bodů do konečné podoby pro další kroky.

5.6 Aproximace těla modelu

Princip aproximace těla spočívá v procházení seřazeného vektoru všech bodů a jejich postupné přidávání k tělu. Před zahájením se všem koncovým bodům nastaví příznak aktivity `active` na hodnotu `true` a přiřadí se jim hodnoty značící číslo segmentu. Čísla segmentů jsou přidělována z rozsahu 1 až *n*, přičemž *n* označuje počet všech reprezentativních koncových bodů výběžků modelu. Zvláštním případem je číslo segmentu 0, které je vyhrazeno pro body náležící tělu modelu. Proto se body musí inicializovat na zápornou hodnotu.

Nechť $\hat{S} = \{\hat{s}_i, i = 1, \dots, N_c\}$ [1] je množina všech reprezentativních koncových bodů a N_c je počet výběžků nalezených v části 4.4, $\hat{s}_i$ značí reprezentativní koncový bod skupiny, která je v pořadí na místě *i*. Potom $P = \left\{P_{ij},\ i, j \in \{1, \dots, N_c\}\right\}$ [1] reprezentuje množinu všech nejkratších cest mezi body $\hat{S}$, kde P_{ij} označuje délku (cenu) nejkratší cesty mezi body $\hat{s}_i$ a $\hat{s}_j$ [1].

Proces aproximace těla probíhá pro všechny body v seřazeném seznamu od bodu s nejnižším ohodnocením. Proběhne kontrola, zda aktuální bod leží v *P*. Cesta P_{ij} zůstává aktivní, pokud je splněna nerovnice 4.5.1. Pro každý bod se vybírá nejbližší koncový bod. Pokud je tento bod aktivní, bod může

být přiřazen do množiny bodů těla. Pokud nejbližší bod není aktivní, zůstává nepřiřazen a pokračuje se zpracováním dalšího bodu. Po úspěšném přidání bodu do těla modelu je třeba ověřit, zda mají všechny reprezentativní koncové body zůstat aktivní. Ověření proběhne podle nerovnice 5.6.1 a kontrolou, zda stále v P existuje aktivní cesta P_{ij} pro tento bod.

$$\frac{\textbf{počet bodů těla v cestě}}{\textbf{počet bodů cesty celkem}} \leq \boldsymbol{t_c} \tag{5.6.1}$$

kde t_c je prahová hodnota určující, jak velká část nejkratších cest mezi reprezentativními koncovými body náleží tělu modelu. Pro tuto implementaci byla konkrétně použita hodnota 0.05, protože se osvědčila během testování pro naprostou většinu modelů. Tato hodnota je definována v hlavičkovém souboru `mesh_segmentation.h` pod názvem `CORE_THRESHOLD`.

V okamžiku, kdy nerovnice neplatí pro žádnou z cest některého reprezentativního koncového bodu, je bod deaktivován nastavením jeho příznaku `active` na hodnotu `false`. Jakmile se bod jednou stane neaktivním, již nemůže být znovu aktivován a zůstane pasivní po zbytek celého procesu neaktivní. Pokud již není žádný koncový bod aktivní, je aproximace těla u konce, tudíž další body již nejsou zpracovávány. Pseudokód aproximace těla je uveden na obrázku 5.6.1.

Tento přístup k aproximaci těla má dvě hlavní výhody oproti podobným metodám [1]:

a) Není zde třeba vícedimensionálních operací nad modelem, které se vyznačují vysokou časovou náročností. Místo toho se využívá nejkratších cest ke zjištění, zda již bylo tělo modelu dostatečně expandováno.

b) Využívá se procentuální vyjádření, kolik bodů těla se musí nacházet mezi dvěma různými výběžky, což zabezpečuje spolehlivé oddělení jednotlivých výběžků.

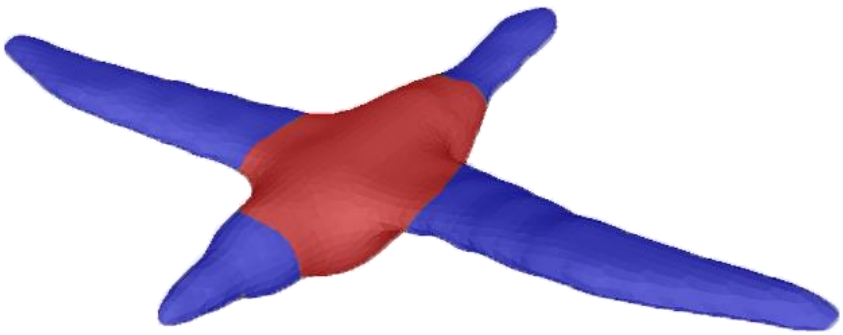
```
for all vertices v ∈ M do
    insert v in PFHeap with priority pf(v)
end for
StopGrowing = false
while !StopGrowing do
    pop a vertex v from PFHeap
    if vCanBeAdded then
        CoreList.add(v)
    end if
    for all P_ij ∈ P do
        if P_ij.active then
            if v ∈ P_ij then
                increment P_ij.counter
                if P_ij.counter / P_ij.SizeOfPath ≥ t_c then
                    P_ij.active = false
                end if
            end if
        end if
    end for
    for all ŝ_i ∈ Ŝ do
        if ŝ_i.active then
            ŝ_i.active = false
            for all ŝ_j ∈ Ŝ − ŝ_i do
                if P_ij.active then
                    ŝ_i.active = true
                end if
            end for
        end if
    end for
    //StopGrowing becomes true if all ŝ_i become
    non-active
end while
```

Obrázek 5.6.1: pseudokód aproximace těla modelu [1].

Na obrázku 5.6.1 *CoreList* označuje seznam bodů tvořící tělo modelu. V reálné implementaci je však nahrazen systémem přidělování čísel segmentů jednotlivým bodům. P_{ij}*.conuter* je počítadlo bodů nacházejících se v cestě P_{ij} přiřazených tělu modelu a P_{ij}*.SizeOfPath* značí celkový počet bodů, jimiž prochází cesta P_{ij}.

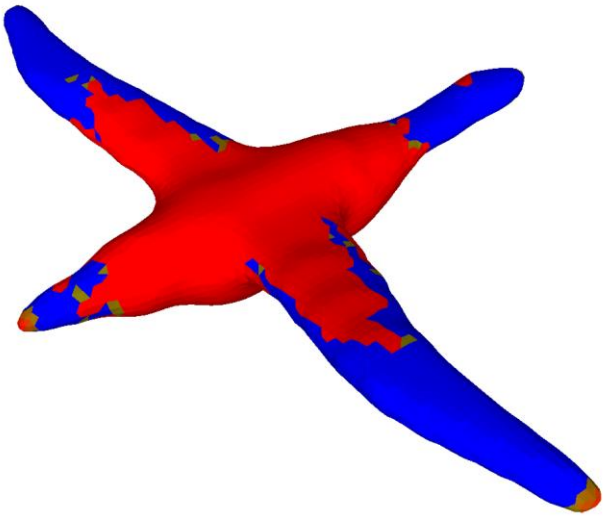
5.7 Odstranění malých rozptýlených částí těla

Po provedení aproximace těla nemusí být výsledný tvar ideální, viz obr. 5.7.1, nýbrž může obsahovat přebytečné body napojené se zbytkem těla, které narušují jednotvárnost výběžků. Tento jev znázorňuje obrázek 5.7.2.



Obrázek 5.7.1: ideální tvar aproximace těla modelu.

K dosažení lepšího výsledku jsou takové body odstraněny ze seznamu *CoreList*. Takovéto body se často vyskytují na koncích výběžků, viz obrázek 5.7.2, kde se reprezentativní koncový bod nachází blíže k tělu, než je reálný konec výběžku. Body na koncích výběžku je snadné detekovat, protože jejich vzdálenost je od všech koncových bodů ostatních výběžků větší než vzdálenost reprezentativního koncového bodu pro daný výběžek.



Obrázek 5.7.2: reálná aproximace těla bez následné úpravy.

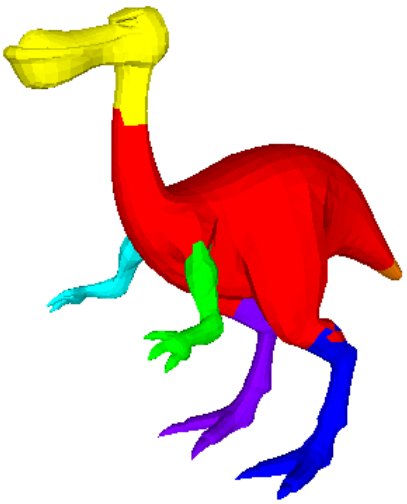
5.8 Rozřazení bodů do výběžků

V tomto kroku se přiřazují body pouze do jednotlivých částí s výjimkou těla modelu, které již nabývá své finální podoby po krocích popsaných v kapitolách 5.6 a 5.7. Z algoritmu aproximace těla modelu vyplývá, že všechny výběžky od sebe musí být vždy odděleny linií bodů náležících tělu modelu o šířce alespoň jednoho bodu.

Během této etapy algoritmu se prochází všechny segmentovaného body modelu kromě bodů přidělených tělu a reprezentativních koncových bodů všech výběžků. Pro každý bod se prochází vektor reprezentativních koncových bodů výběžků a pro každý se kontroluje, zda nemůže být aktuální bod přiřazen k výběžku obsahujícímu aktuální koncový bod. Pakliže neexistuje bod označený jako bod těla modelu ležící na nejkratší cestě z kontrolovaného bodu do odpovídajícího bodu výběžku.

Během přiřazování bodů do výběžků se může vyskytovat zvláštní situace, kdy některý z bodů (i více bodů) nelze přiřadit žádnému z výběžků. Z toho vyplývá, že bod je od všech koncových bodů oddělen nějakou částí těla a tudíž aktuální bod také patří k tělu modelu. Tím jsou doplněny případné "díry" v těle modelu.

Přiřazením všech bodů končí segmentace modelu. Tento výsledek je konečný, viz obrázek 5.8.1, a již neprobíhají žádné další úpravy. V tomto stavu proběhne uložení modelu do výstupního souboru.



Obrázek 5.8.1: reálný výsledek celého algoritmu.

5.9 Speciální implementované funkce

Za účelem testování byly implementovány speciální funkce, které se přímo nepodílejí na provedení segmentace modelu, ale umožňují výstup dílčích kroků řešení. Cílem těchto funkcí je zjednodušení zjištění chyb, jejich lokalizace ve zdrojovém kódu a jejich odstranění. Tyto speciální funkce také sehrály významnou roli při ladění programu.

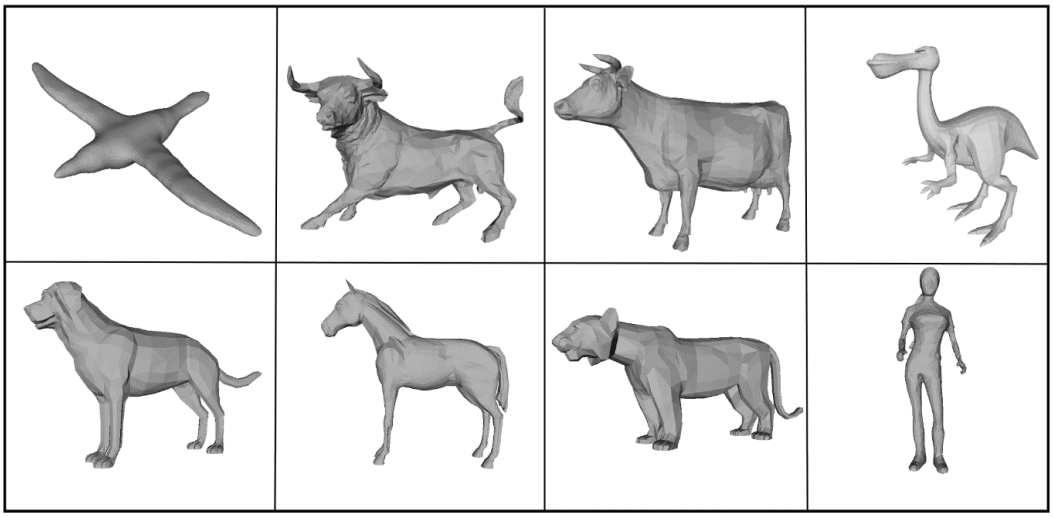
Funkce `getpfcolor` poskytuje na výstupu model obarvený podle hustoty hodnotící funkce. Barvu každé trojúhelníkové plochy určuje výběrem nejvyššího ohodnocení ze tří bodů, které tuto plochu vymezují. Funkci lze předat jako první parametr výstup funkce `pf_max_of_facet`. tento parametr určuje následující chování funkce. Oblasti, které obsahují kandidátní body na koncové body výběžků, jsou zde obarveny maximální intenzitou barevné složky, nikoliv podle jejich skutečného hodnocení. Pokud je jako první parametr předána návratová hodnota funkce `does_facet_have_sp`, na výstupu je model obsahující pouze barevně odlišené reprezentativní koncové body výběžků.

6 Testování

Testování implementovaného programu je nedílnou součástí vývoje, protože bylo třeba ověřit správnost jednotlivých kroků řešení již od samého počátku implementace. Za účelem testování byly vyvinuty speciální funkce, které při realizace konečné podoby segmentace nemají žádné využití. Takové funkce vznikly především s cílem výstupu dílčích kroků řešení. Proto finální řešení obsahuje metody, při jejichž použití program neprodukuje segmentovaný model, ale například model s obarvením povrchu odpovídajícím hustotě hodnotící funkce všechny body modelu. Tyto funkce jsou popsány v kapitole 4.10.

6.1 Testovací data

Jako testovací data posloužily 3d modely volně dostupné na internetu. Hlavním zdrojem modelů se stal web Princetonské university http://shape.cs.princeton.edu/benchmark/ [2]. Tento web poskytuje početnou sadu testovacích modelů obsahující modely mnoha různých tvarů, úrovní detailů a složitosti. Proto byla většina testovacích modelů vybrána právě z této databáze. Zdrojem dalších modelů se stala testovací data obsažená v repozitáři knihovny CGAL [4]. Obrázek 5.3.1 znázorňuje ukázku modelů, na kterých proběhlo testování.



Obrázek 6.1: ukázka testovacích modelů.

6.2 Využité nástroje

Testování probíhalo na operačním systému Ubuntu 14.04 LTS Desktop v prostředí terminálu. Celý běh programu byl sledován pomocí výpisů na standardní výstup. Stejným způsob je řešen i výstup informující o tom, jak velká část algoritmu již proběhla. Kromě terminálu použitého ke spouštění programu a získávání chybových hlášení byl využit program MeshLab [4]. Program MeshLab byl využit především k vizualizaci výsledných modelů, protože vizualizace jako taková není zadáním práce a v implementaci programu není obsažena. Program také posloužil jako nástroj pro získání snímků výstupních úprav modelů.

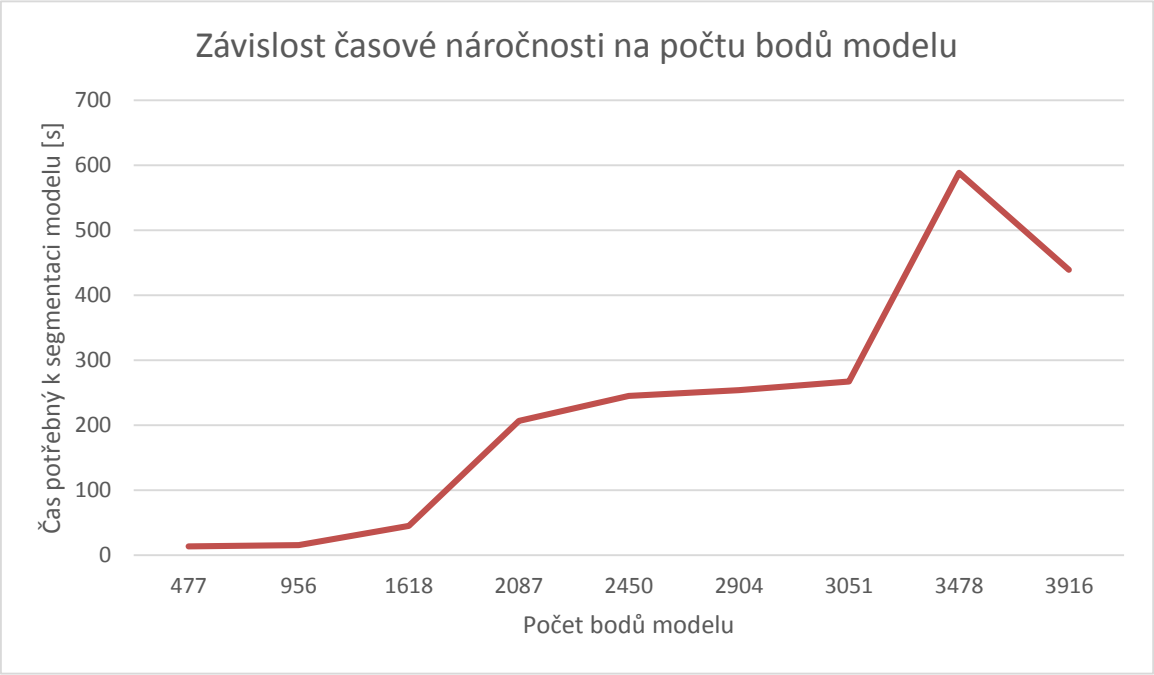
6.3 Naměřené hodnoty časové náročnosti segmentace polygonálních modelů

Během testování programu na testovacích datech byla shromážděna statiska informací o časové náročnosti segmentace modelu a jeho složitosti. Doba zpracování modelu je znázorněna na v tabulce 6.3.1. Tabulka obsahuje informace o počtu bodů modelu a čas, který byl potřeba pro vykonání segmentace. Názvy modelů v prvním sloupci označují název souboru, ve kterém jsou modely uloženy na přiloženém CD.

Tabulka 6.3.1: naměřené hodnoty časové náročnosti segmentace testovacích modelů.

název modelu	počet bodů	počet ploch	čas segmentace
dino	3916	7828	7m19.319s
bird	3478	6952	9m48.727s
cow	2904	5804	4m14.225s
bull	2087	4170	3m26.667s
dog	1618	3220	0m45.081s
fish	477	950	0m13.534s
horse	3051	6052	4m27.421s
horse2	2450	4896	4m5.147s
lion	956	1908	0m15,4s

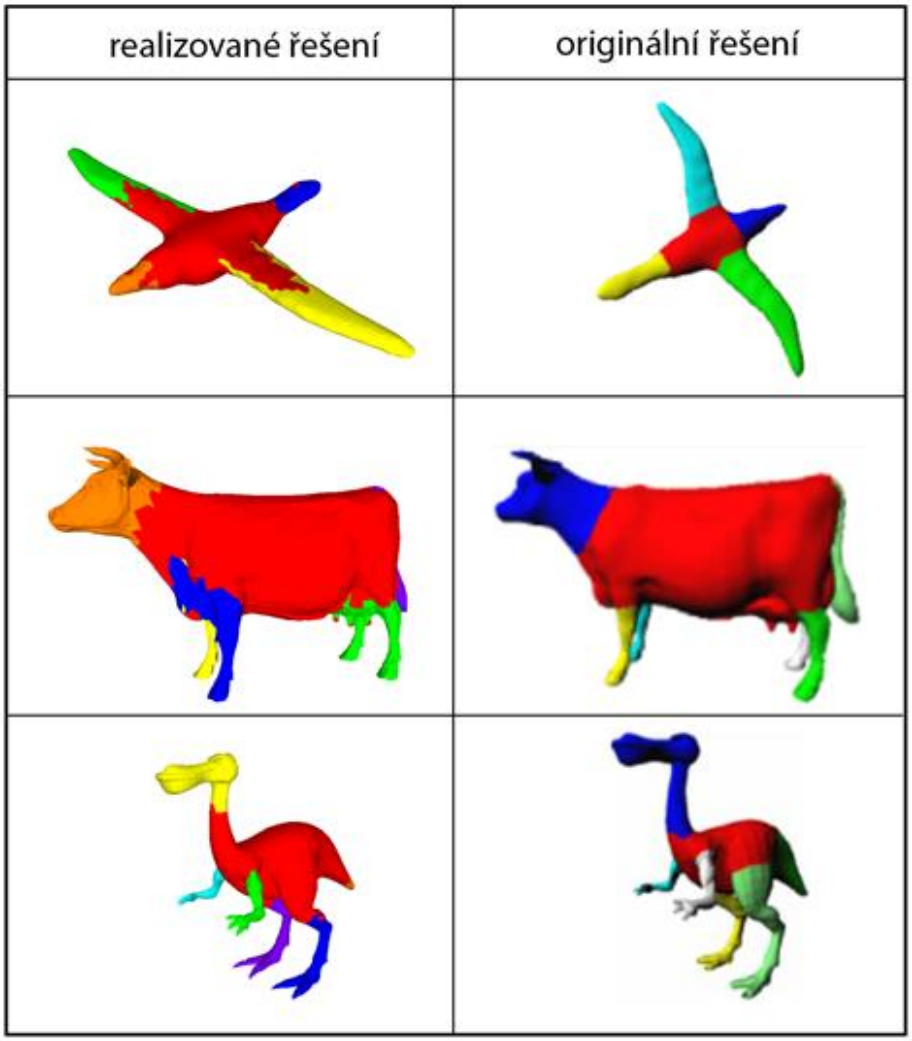
Hodnoty uvedené v tabulce 6.3.1 jsou naneseny do následujícího grafu. Z grafu je patrná prudce rostoucí časová náročnost segmentace při rostoucím počtu bodů modelu. Graf také naznačuje, že doba segmentace není závislá pouze na počtu bodů modelu, ale také na počtu trojúhelníkových ploch modelu a na struktuře povrchu modelu.



Graf 6.3.1: znázornění závislosti délky zpracování na počtu bodů modelu.

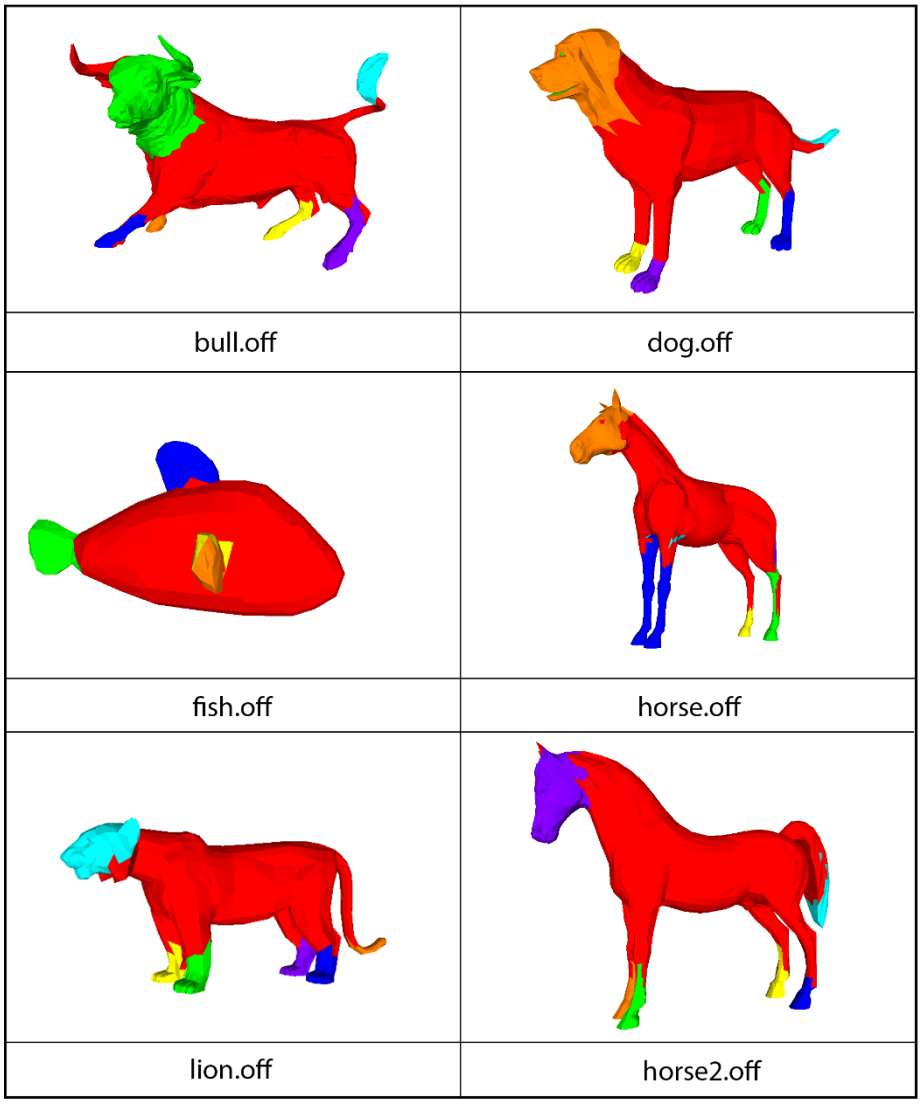
6.4 Dosažené výsledky segmentace

Cílem programu je segmentace 3D polygonálních modelů na menší významově oddělitelné celky. Vytvořený program tento cíl naplňuje především pro modely odpovídající myšlence, že model se skládá z těla a výběžků k němu připojených. Kvalitní výsledky se osvědčili především u modelů zvířat a lidí, které lze významově rozdělit na tělo, hlavu a končetiny (tímto modely splňují předchozí myšlenku přístupu k segmentaci modelu). Přestože realizované řešení dosahuje kvalitních výsledků segmentace, nedosahuje stejné přesnosti, jaká je prezentována ve článku popisující originální metodu [1]. Do kvality řešení se znatelně promítly provedené úpravy metody, především zjednodušení algoritmu hledání hranice mezi jednotlivými výběžky a tělem modelu. Porovnání těchto výsledků ukazuje obrázek 6.4.1.



Obrázek 6.4.1: porovnání výsledků realizovaného řešení a výsledků řešení uvedených v článku [1].

Počet modelů k porovnání s originálním návrhem metody [1] je značně omezený, protože se nepodařilo najít více identických modelů. Výsledky pro další modely jsou znázorněny na obrázku 6.4.2. Popisky u jednotlivých modelů označují název souboru, ve kterém jsou uloženy v příloze na CD.



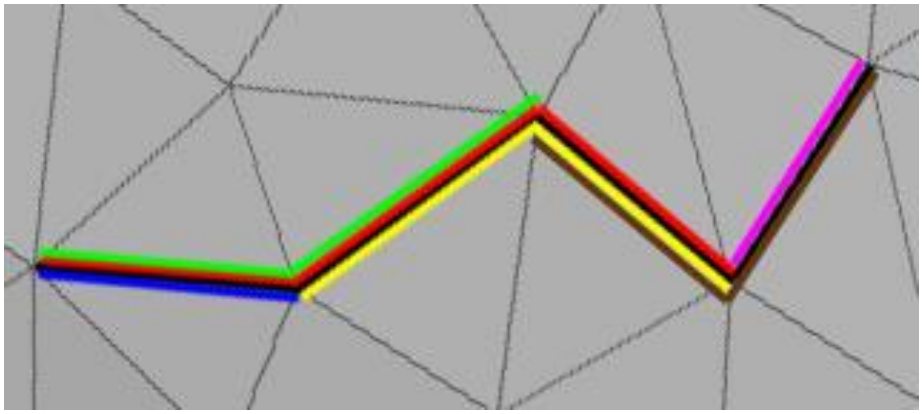
Obrázek 6.4.2: ukázka výsledků segmentace polygonálních modelů, uvedeny názvy souboru, ve kterých jsou uloženy původní modely.

7 Pokračování do budoucna

Práce v aktuálním stavu již poskytuje uspokojivé výsledky, avšak naprosté dokonalosti ještě nedosahuje. Mezi hlavní dosavadní nedostatky patří rychlost algoritmu a také je třeba zrealizovat přesnější metodu určení hranice jednotlivých částí modelu.

Časově nejnáročnější část zpracování modelu je ohodnocení všech bodů modelu, respektive nalezení nejkratších cest mezi všemi dvojicemi modelů. V současné implementaci jsou odstraněny pouze redundantní výpočty nejkratších cest opačnými směry. Tím je odstraněna polovina výpočtů vzhledem k řešení bez optimalizací.

Ke značnému zrychlení algoritmu by došlo, pokud by se při nalezení jedné nejkratší cesty neukládala pouze délka celé cesty, ale také délka všech cest obsažených v ní obsažených. Tím by se razantně snížila časová náročnost výpočtu, protože zejména v případě dlouhých cest (z jednoho konce modelu na druhý), by se vypočítala délka velkého počtu „podcest“, které již nebude třeba znova hledat. Tímto dojde k naplnění bufferu uchovávajícího délky všech cest během mnohem menšího počtu kroků než při stávajícím stavu implementace. Znázornění cesty a jejích dílčích cest ukazuje obrázek 7.1. Obrázek neukazuje všechny dílčí cesty, ale jen některé. Nechť P je množina všech bodů v cestě C, potom množina všech dílčích cest (jejich koncových bodů) M_c cesty C je tvořena kartézským součinem



Obrázek 7.1: Zobrazení některých dílčích cest.

$P \times P$. M_c pak obsahuje všechny dílčí cesty.

Dalším aspektem je úprava hledání hranice těla a výběžků. V reálné implementaci byl algoritmus značně zjednodušen a je popsán v kapitole 4.6. Během dalšího vývoje by mělo dojít k implementaci plné verze metody. Tato verze je popsána v kapitole 4.8 zabývající se provedenými úpravami.

8 Závěr

Jako řešení této bakalářské práce na téma Segmentace polygonálního modelu byla implementována terminálová aplikace provádějící segmentaci 3D modelů s povrchovou reprezentací (polygony). Cílem segmentace je rozdělení složitých modelů na jednodušší a menší části. Implementovaná metoda vychází ze článku Protrusion oriented 3D mesh segmentation [1].

Realizovaná metoda je založena na myšlence, že se model skládá z těla a výběžků z těla vystupujících. Z toho vyplývá nutnost oddělení jednotlivých výběžků od sebe nějakou částí těla modelu. Tento přístup má za cíl rozdělit model na části, které i v reálném světě reprezentují smysluplné segmenty modelu. Takové dělení modelu se osvědčilo především u modelů lidských bytostí a zvířat, protože oba druhy modelu lze vždy rozdělit na hlavu a končetiny, u zvířat ocas.

Podařilo se dosáhnout uspokojivých výsledků segmentace právě pro modely zvířat a lidí. Výsledná segmentace rozděluje model na významové menší části. Části jsou ve výstupních souborech rozlišeny různou barvou. Během dalšího vývoje bude implementované řešení více přiblíženo originálnímu návrhu hledání hrany těla modelu v Protrusion-oriented 3D Mesh segmentation s cílem dosažení ještě lepších výsledků.

9 Reference

1. AGATHOS, Alexander, Ioannis PRATIKAKIS, Stavros PERANTONIS a Nickolas S. SAPIDIS. Protrusion-oriented 3D mesh segmentation. *The Visual Computer* [online]. 2010, **26**(1), 63-81 [cit. 2016-04-12]. DOI: 10.1007/s00371-009-0383-8. ISSN 0178-2789. Dostupné z: http://link.springer.com/10.1007/s00371-009-0383-8

2. Princeton Shape Benchmark. *Princeton Shape Benchmark* [online]. Princeton University, 2003 [cit. 2016-05-08]. Dostupné z: http://shape.cs.princeton.edu/benchmark/

3. *The Computational Geometry Algorithms Library* [online]. 2016 [cit. 2016-05-09]. Dostupné z: http://www.cgal.org/

4. GitHub, Inc. *The CGAL Project* [online]. 2016 [cit. 2016-05-09]. Dostupné z: https://github.com/CGAL

5. SourceForge. *MeshLab* [online]. [cit. 2016-05-09]. Dostupné z: http://meshlab.sourceforge.net/

6. MANGAN, A.P. a R.T. WHITAKER. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics* [online]. **5**(4), 308-321 [cit. 2016-05-09]. DOI: 10.1109/2945.817348. ISSN 10772626. Dostupné z: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=817348

7. CGAL 4.8. *Triangulated Surface Mesh Segmentation* [online]. 2016 [cit. 2016-05-09]. Dostupné z: http://doc.cgal.org/latest/Surface_mesh_segmentation/index.html

10 Seznam příloh

Příloha 1. CD

Obsah přiloženého cd:

/src …….. zdrojové kódy programu
/lib ……... knihovna CGAL
/data …… ukázkové testovací modely
/demo …….. demonstrační skript spouštějící program segmentace
/doc ……. technická zpráva ve formátu .docx (MS Word 2016) a .pdf
README …... manuál k překladu a spuštění programu