

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

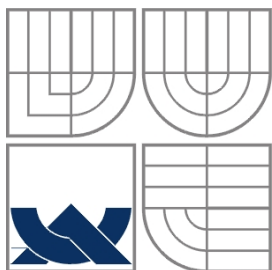
ZPRACOVÁNÍ 3D MODELŮ SCÉNY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

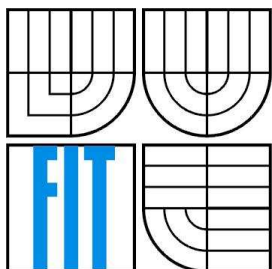
AUTOR PRÁCE
AUTHOR

Bc. LUKÁŠ ZDRÁHAL

BRNO 2008



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

ZPRACOVÁNÍ 3D MODELŮ SCÉNY

PROCESSING OF 3D SCENE MODELS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. LUKÁŠ ZDRÁHAL

VEDÚCÍ PRÁCE
SUPERVISOR

Doc. Dr. Ing. PAVEL ZEMČÍK

BRNO 2008

Abstrakt

Tento dokument má za úlohu oboznámiť čitateľa s princípmi digitalizácie a spracovania polygonálnych modelov trojrozmerných objektov. V práci je popísaný prehľad prakticky používaných zariadení, ich fyzikálny princíp a niekoľko rôznych metód merania. V ďalšej časti práca popisuje prehľad základných metód pre spracovanie modelov ako vyhladzovanie a decimácia, ktoré sú nevyhnutné z hľadiska spracovania modelu scény, ako tiež popis ich implementácie a vytvorenie užívateľského rozhrania pre publikáciu modelov prostredníctvom WWW stránok.

Časť práce ktorá sa týka digitalizácie, je úzko spojená s konkrétnym zariadením - digitizérom od spoločnosti Minolta VIVID-700, ktorý je umiestnený na našej fakulte, preto sú v dokumente uvedené taktiež praktické skúsenosti, demonštračné ukážky a technické parametre.

Záverom dokumentu je uvedené zhodnotenie výsledkov, ich porovnanie a ďalší predpokladaný postup projektu.

Kľúčové slová

3D skenovanie, 3D digitizér, metódy skenovania, Minolta VIVID 700, rekonštrukcia, decimácia, vyhladzovanie, www publikácia.

Abstract

Purpose of this document is acquaint reader with basic principles of 3D model digitalization. This work describes general overview of 3D scanning devices, their physical principle and measurements methods. Next part of this document describes basic method for polygonal mesh processing as smoothing and decimation which are necessary for 3D model processing. This document contains also algorithms description of implementation, user interface and publication part through WWW. Fundamental essence of this diploma thesis will be introduction with general principles of 3D scanning and working with Minolta VIVID-700 3D digitizer which is placed on our faculty.

At the end are mentioned results evaluation, demonstration examples and next supposed project advancement.

Keywords

3D scanning, 3D digitizer, digitalization methods, Minolta VIVID 700, reconstruction, decimation, smoothing, www publication

Citácia

Lukáš Zdráhal: Spracovanie 3D modelu scény, diplomová práca, FIT VUT v Brne, 2008.

Spracovanie 3D modelov scény

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením Doc. Dr. Ing. Pavla Zemčíka.

Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Lukáš Zdráhal
24. 4. 2008

PodĎakovanie

Týmto by som chcel poďakovať vedúcemu semestrálneho projektu Pavlovi Zemčíkovi a Stanislavovi Sumcovi za pomoc, ktorá mi veľmi pomohla pochopiť problematiku v oblasti práce s 3D skenermi.

Taktiež by som rád spomenul Pavla Tišnovského za ochotu pri poskytovaní informácií pre získanie strateného softvéru zariadenia Minolta VIVID 700.

Táto práca vznikla ako školné dielo na Vysokom učení technickom v Brne, Fakulte informačných technológií. Práca je chránená autorským zákonom a jej užitie bez udelenia oprávnenia autorom je nezákonné, s výnimkou zákonom definovaných prípadov.

Obsah

1	Úvod.....	2
2	Metódy digitalizácie.....	3
2.1	Rozdelenie digitizérov podľa princípu.....	3
2.2	Metódy merania vzdialenosti	10
3	Spracovanie modelu scény.....	14
3.1	Získanie vstupných dát.....	14
3.2	Laplaceovo vyhladzovanie polygonálnej siete.....	16
3.3	Zjednodušovanie povrchov	17
3.4	Quadric error metrics	19
4	Popis vlastností skeneru.....	23
4.1	Popis zariadenia.....	23
4.2	Dodávaný softvér a rekonštrukcia objektu.....	24
4.3	Problémy pri skenovaní.....	26
5	Návrh publikačného systému	27
6	Popis implementácie	28
6.1	Načítanie vstupných dát	30
6.2	Laplaceovo vyhladzovanie polygonálnej siete.....	31
6.3	Decimácia polygonálnych modelov	32
6.4	Popis užívateľského rozhrania	33
6.5	Popis publikačnej časti	36
7	Namerané výsledky a vlastnosti aplikácie	38
7.1	Vyhladzovanie polygonálnej siete.....	38
7.2	Decimácia polygonálnej siete.....	40
7.3	Zhodnotenie nameraných výsledkov.....	41
7.4	Použité prostriedky.....	42
8	Záver	43

1 Úvod

Vďaka obrovskému technologickému pokroku v posledných rokoch na poli výpočtovej techniky došlo k expanzii aplikácií pracujúcimi s modelmi fyzických objektov. Výsledkom je, že dnes sa počítačová grafika a modelovanie objektov uplatňuje v mnohých odvetviach priemyslu. V súčasnej dobe preto rastú požiadavky na rýchlosť získavania 3D polygonálnych modelov scény. Predmetom tejto diplomovej práce bude teda uvedenie čitateľa do problematiky spracovania modelov pomocou 3D digitizérov, ktoré sa vďaka dnešným trendom na poli informačných technológií značne rozšírili.

Získavanie tvaru objektov pomocou 3D skenerov je v poslednej dobe jeden z najdôležitejších zdrojov pre generovanie komplexných modelov pozostávajúcich z polygonálnej siete. Z historického hľadiska sa jedná o relatívne staršiu technológiu, ktorej masívny rozmach nastal s príchodom kvalitnejších snímacích čipov, zvýšením výkonu výpočtovej techniky a potrebami užívateľa ako aj typického vývojára v oblasti počítačovej grafiky. Typickým príkladom použitia 3D digitizérov v praxi by mohol byť vývoj počítačových hier, strojársky priemysel resp. rôzne oblasti v medicíne.

Digitálny model je v dnešnej dobe možné vytvoriť dvoma rozdielnymi spôsobmi:

- Použitím 3D modelovacieho softvéru. Príkladom môže byť najznámejší 3D Studio MAX, Maya, Blender alebo množstvo ďalších, ktoré sa využívajú v strojárskom priemysle. Takéto „ručné“ modelovanie prináša radu nevýhod hlavne z hľadiska rýchlosti a presnosti. Príkladom je modelovanie objektov v počítačových hrách, ktorých zložitosť je vo väčšine prípadov neúnosná a tým pádom veľmi zdĺhavá.
- Digitalizáciou (skenovaním) fyzického objektu pomocou 3D digitizéru. Takýto model, ktorý je vernou kópiou originálu je možné v ďalších fázach upravovať rôznymi algoritmami pracujúcimi s polygonálnou sieťou (vyhladenie, decimácia polygonálnej siete, triangulácia vzniknutých dier, atď.), resp. modelovacím softvérom uvedenom v predchádzajúcom bode. Jedná sa totiž o prístup, ktorý sa z praktického hľadiska javí jednoznačne efektívnejší.

Cieľom tejto práce je uvedenie čitateľa do problematiky spracovania modelu scény pomocou 3D skeneru, popisom niektorých najpoužívanejších metód skenovania, princípov, ich výhod a nevýhod, ako tiež niektorých základných algoritmov potrebných pre finálne spracovanie po digitalizácii. Súčasťou bude taktiež návrh užívateľského rozhrania pre publikáciu a prehliadanie modelov.

Táto diplomová práca je rozdelená do siedmich hlavných častí (bez tohto úvodu). V prvej časti práce by som popísal prehľad 3D skenerov, v ktorom budú popísané všeobecne dostupné a v dnešnej dobe prakticky využívané zariadenia vrátane dostupných metód merania vzdialenosti. Ďalej by som uviedol popis niektorých algoritmov, potrebných pre úpravy polygonálnej siete a ich význam v celkovom procese spracovania modelu.

Keďže táto práca sa týka digitalizácie predmetov, jej dôležitou časťou je práca s 3D skenerom umiestneným na našej fakulte. V tejto tretej časti sú uvedené praktické poznatky a skúsenosti pri práci s digitalizačným zariadením ako tiež metódy dôležité pre túto diplomovú prácu. Štvrtá a piata časť sa bude zaoberať návrhom a popisom implementačnej časti diplomovej práce, uvedením do problematiky a komentárom nameraných výsledkov.

Koniec dokumentu tvorí časť zhodnocujúca aktuálny stav a naznačujúca ďalší vývoj projektu. Demonštračnými ukážkami, ako tiež užívateľskou príručkou sa zaoberá posledná časť – jednotlivé prílohy.

2 Metódy digitalizácie

V tejto kapitole by som čitateľovi popísal všeobecné metódy digitalizácie, ktoré sú využívané v praxi. U každého typu by som rád uviedol vlastnosti jednotlivých metód a zariadení, ich výhody, nevýhody a tiež zhodnotenie z hľadiska použiteľnosti v praxi.

V jednoduchosti povedané, digitalizácia 3D modelu zahŕňa nasnímanie priestorových súradníc objektu, spracovanie nameraných dát a ich prevod na plošnú, resp. objemovú reprezentáciu, s ktorou je možné ďalej pracovať napríklad v CAD systéme.

2.1 Rozdelenie digitizérov podľa princípu

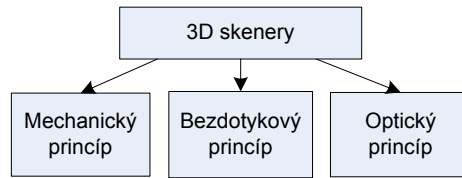
Skenery je možné rozdeliť do troch skupín. Do prvej skupiny patria mechanické sondy, ktoré pracujú na princípe priameho kontaktu s objektom. Táto metóda je najpomalšia, ale umožňuje snímať vnútorne dutiny telesa, ktoré nie je možné získať napríklad pomocou laserového bezdotykového princípu. Jedná sa o najpomalší spôsob merania. Digitalizácia je možná tiež pomocou optických zariadení, ktoré pracujú na princípe počítačového videnia. Tento spôsob je v praxi využívaný, hlavne kvôli dostupnosti a spôsobu vyhodnotenia.

Ako najpraktickejšia tretia možnosť, je použitie bezdotykových skenerov pracujúcich na laserovom princípe. Na trhu je v dnešnej dobe celá rada prístrojov pracujúcich so známymi formátmi dát, pri veľmi uspokojivej rýchlosti a presnosti.

Medzi najväčšiu výhodu bezdotykového snímania pomocou laseru patrí vlastnosť, že skenovaný objekt môže byť uložený na ľubovoľnom mieste a objekt nie je potrebné vkladať do

špeciálneho zariadenia (ako v prípade roentgenového, resp. CT/MR snímania). Preto táto metóda rozširuje použiteľnosť pre digitalizáciu objektov, ktoré nie je možné presunúť do laboratória.

Typickým príkladom môžu byť stavby, historické pamiatky, časti áut alebo ľubovoľná geometria u ktorej by meranie a modelovanie mohlo byť veľmi problematické.



Obr.2.1.: Rozdelenie digitizérov podľa princípu skenovania.

Ku konkrétnejšiemu popisu jednotlivých spôsobov sa dostanem v nasledujúcich častiach tejto kapitoly. Bude tu uvedený stručný prehľad, ale vzhľadom k tomu, že predmetom diplomovej práce bude práve práca s laserovým 3D skenerom, rád by som tento spôsob popísal detailnejšie v samostatnej časti.

2.1.1 Mechanické 3D skenery

Skenery pracujúce na tomto princípe patria na trhu medzi tie najlacnejšie. Ich princíp je založený na tom, že skenovaný objekt „ručne“ prejdeme hrotom, ktorý je umiestnený na špeciálnom ramene. Toto rameno má v každom kĺbe umiestnené senzory, ktoré spätne udávajú informáciu o polohe. Tento typ skeneru neumožňuje zistenie textúry objektu a je potrebné ju dodať explicitne pomocou dodatočného optického zariadenia.

Vzhľadom k potrebe manuálnej obsluhy je skenovanie touto formou časovo náročnejšie, ale oproti ostatným skenerom umožňuje snímanie zložitejších objektov aj so slepými miestami tzv. dutinami.



Obr.2.2: Skenovanie mechanickým 3D skenerom. Prevzaté z [2].



Obr.2.3.: Mechanický skener Microscribe MX. Prevzaté z [8]

Na obrázku 2.3 je uvedený mechanický skener spoločnosti Microscribe, ktorý je pripojiteľný cez sériový port počítača a dokáže snímať až 1000 bodov za sekundu.

Tento skener je tiež možné využiť ako “myš“(navigátor)”, ktorá na rozdiel od tej klasickej pracuje v trojrozmernom priestore. Toto je možné využiť pri umiestňovaní objektov, svetiel v modelovacom softvéri, bez zložitého a zdĺhavého prepínania osí. Skenery bývajú dodávané v rôznom prevedení, pričom najčastejšie sa líšia hlavne v dĺžke ramien a presnosti merania. [8]

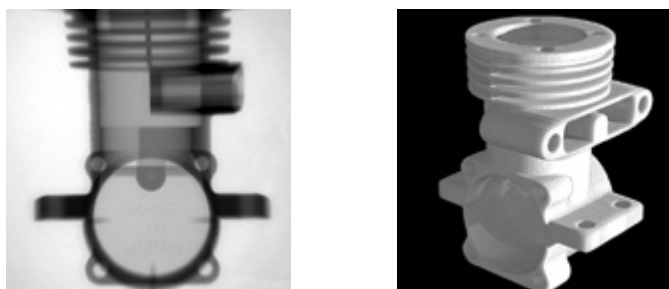
Mechanické skenery nie sú ovplyvniteľné typom povrchu, a sú veľmi dobre podporované v moderných modelovacích systémoch ako Rhinoceros, 3DStudio Max a pod.

2.1.2 Röntgenové skenery

Tento spôsob digitalizácie funguje na princípe bezkontaktného snímania. Röntgenové 3D skenery pracujú na princípe získavania informácií o vnútornej geometrii použitím röntgenového žiarenia, tak ako klasické röntgeny, používané v zdravotníctve s tým rozdielom, že pri tomto je používaná intenzita žiarenia vyššia.



Obr.2.4.: Röntgenový skener, určený pre mikroskopické merania s presnosťou 1 μ m. Prevzaté z [8].



Obr.2.5.: Príklad skenovaného objektu (vľavo) a vytvoreného polygonálneho modelu (vpravo). Prevzaté z [8].

Systemy pre skenovanie pomocou röntgenu sú väčšinou dodávané ako kompaktné riešenie, ktoré vďaka svojej objemnosti postrádajú výhodu prenositeľnosti. Zariadenie uvedené na obrázku 2.4 dosahuje oproti ostatným typom skenerov omnoho väčšej presnosti (až $1\mu\text{m}$), ale jeho váha – 300 kg [8] obmedzuje použitie iba na laboratórne podmienky, v ktorom býva najčastejšie umiestnené.

Skenovanie objektov pomocou röntgenu býva často zameniteľné s tzv. defektoskopiou, kde sa žiarenie využíva k odhaleniu skrytých väd materiálu (napr. v metalurgii).

2.1.3 Ultrazvukové skenery

V prípade skenovania ultrazvukom je meranie vykonávané ultrazvukovou sondou tvaru pištole s kovovým hrotom, ktorý prikladáme k povrchu a stlačením spúšte dôjde k vyslaniu ultrazvukového signálu. Tento signál je pomocou ultrazvukových čidiel dekódovaný do priestorových súradníc. Nevýhodou zariadenia je jeho relatívne malá presnosť.



Obr.2.6.: Ultrazvukový skener FastTrak spoločnosti Cortech. Prevzaté z [10].

2.1.4 Deštruktívne 3D skenery

Jedná sa o typ zariadenia, ktoré má schopnosť digitalizovať vonkajší povrch aj vnútornú geometriu. Princíp digitalizácie spočíva vo vyplnení dutín a okolia kontrastným materiálom. Takto pripravený objekt je uložený do frézovacieho stolu, ktorý po tenkých plátkoch odstraňuje jednotlivé vrstvy. Jednotlivé vrstvy sa naskenujú pomocou optického zariadenia a odošlú sa k ďalšiemu spracovaniu. Skenovaný objekt je síce pri digitalizácii zničený, ale hlavné výhody prináša v presnosti, ktorá závisí od hrúbky odfrézovaného plátku. U tejto metódy sa chyba približuje k 1 μ m.

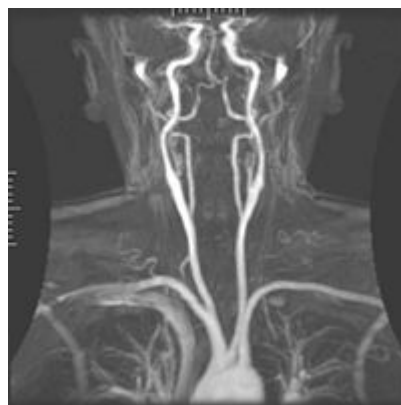


Obr.2.7.: Deštruktívny skener CCS-3000. Prevzaté z [13].

2.1.5 Digitalizácia pomocou magnetickej rezonancie

Je zobrazovacia technika používaná predovšetkým v zdravotníctve. Umožňuje získať rezy skenovaného objektu, pomocou kombinácie magnetického a elektromagnetického poľa ktoré je možné ďalej spracovať do výsledného 3D modelu. Tento spôsob z fyzikálnej podstaty neumožňuje skenovanie objektov z ľubovoľného materiálu, preto je využívaná iba v tejto špecifickej oblasti.

Spracovaním získaných dát (najčastejšie objemových) je pomocou špecifických metód (marching cubes) získať polygonálnu reprezentáciu. Objemové dáta je možné prezerat' aj priamo napríklad pomocou ray-castingu, resp. voxel splattingu.



Obr.2.8.: Zariadenie pre digitalizáciu pomocou MR (vľavo). Obráz získaný pomocou MR po podaní kontrastnej látky (vpravo). Prevzaté z [11].

2.1.6 Optické 3D skenery

Tento bezdotykový druh skenerov sníma skenovaný objekt z niekoľkých pohľadov, pričom uhol otočenia je nastaviteľný ručne, alebo pomocou otočného stola riadeného počítačom. Objekt, ktorý je „odfotený“ z niekoľkých uhlov, je odoslaný do zariadenia pre spracovanie, kde sa pomocou vhodnej metódy rekonštruje. Uplatňuje sa princíp stereovidenia.



Obr.2.9.: Optický skener Atos II. Prevzaté z [12].

Kvalita výsledného objektu sa dá ovplyvniť množstvom získaných snímok. Táto metóda vyžaduje aby objekt bol umiestnený proti kontrastnému pozadiu, inak by nebolo možné výsledný objekt vysegmentovať z obrazu. Metóda neumožňuje digitalizáciu miest, ktoré sú pre skener „slepé“ tj. najčastejšie dutín a priehlbín. Preto sa ako každé z uvedených zariadení používa iba pre špecifické aplikácie.

Medzi výhody optických skenerov patrí hlavne mobilita, flexibilita, rýchlosť merania, presnosť a vysoká hustota výsledného mračna bodov.

2.1.7 Laserové 3D skenery

Ako bolo v predošlej časti spomenuté, predmetom tejto diplomovej práce je použitie laserového skeneru umiestneného na našej fakulte. Preto by som bol pri tejto metóde konkrétnejší a uviedol tiež niektoré metódy merania. V prílohe A sú pre čitateľa k dispozícii technické parametre a informácie k dodávanému zariadeniu a softvéru skeneru Minolta VI-700.

Laserové skenovacie systémy umožňujú bezkontaktné určovanie priestorových súradníc, 3D modelovanie a vizualizáciu stavieb, konštrukcií, interiérov, terénov i drobnejších predmetov. Závisí hlavne od poskytovaných možností použitého zariadenia.

Naskenovaný objekt môže byť príslušným softvérom zobrazený ako „mračno bodov“ – (point cloud) - výšková mapa (viď obr.2.11), z ktorej je možné použitím špeciálneho softvéru dodávaného k zariadeniu rekonštruovať 3D model použiteľný v ďalších aplikáciách. Je potrebné si uvedomiť, že ako každá iná, má aj táto metóda svoje fyzikálne a technologické obmedzenia, ktoré môžu meranie znehodnotiť.

Medzi najväčšie výhody laserovej digitalizácie patrí:

- automatický a systematický zber bodov
- vysoká hustota bodov a rýchlosť ich zmerania (státisíce bodov za sekundu, čo závisí od typu, resp. modelu použitého zariadenia)
- vypočítanie súradníc v reálnom čase
- okamžité zobrazenie naskenovanej výškovej mapy na prehľadovom displeji zariadenia

Podľa druhu je možné laserové skenery rozdeliť do nasledujúcich kategórií:

- statické, pevne fixované na mieste napr. výrobné linky, napr. pre meranie veľkosti kvapiek lepidla (viď obr.2.11)
- mobilné systémy umiestňované na statívy pre meranie na krátke a stredné vzdialenosti (viď obr.2.10)
- systémy leteckého skenovania pre topografické aplikácie(GIS)

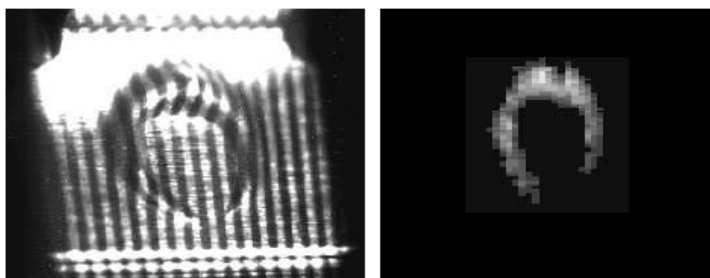
Z hľadiska použitej technológie:

- triangulačné skenery (napr. laboratórne, prenosné)
- laserové skenovacie systémy diaľkomerné (stacionárne, mobilné, letecké)

Výhodou laserových skenerov je ich vysoká presnosť, mobilita, rýchlosť a nenáročnosť na obsluhu, ale na druhej strane, ich vysoká cena omnoho prevyšuje ostatné typy skenovacích zariadení. Pre využitie v praxi má oproti ostatným najlepšie predpoklady.



Obr.2.10.: Zariadenie pre skenovanie vzdialených objektov (vľavo), pre letecký prieskum terénu pri využití v GIS (stred), v laboratórnych podmienkach (vpravo). Prevzaté z [11].



Obr.2.11.: Meranie objemu kvapky lepidla. Nasvietený obraz (vľavo) a jej ekvivalentná výšková mapa (vpravo). Prevzaté z [13].

2.2 Metódy merania vzdialenosti

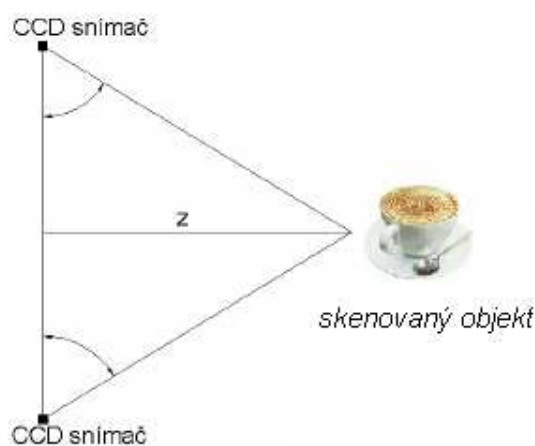
Optické snímače s ktorými pracujeme umožňujú získať iba dvojrozmerný obraz, ktorý pre našu potrebu skenovania 3D modelov nepostačuje. Preto je pri meraní potrebné využiť fyzikálne princípy svetla, ktoré nám pomôžu odhaliť hĺbku skenovaného objektu v podobe výškovej mapy.

Metódy pre meranie tretieho rozmeru je možné rozdeliť na metódy založené na triangulácii, alebo tie, ktoré využívajú vlnové vlastnosti svetla.

2.2.1 Pasívna triangulácia

Tento princíp je veľmi podobný ľudskému videniu, pri ktorom skenovaný objekt snímame dvoma kamerami, ktorých vzdialenosť je konštantná. Ľudské oko vytvára stredový obraz pozorovaného predmetu na sietnici. Pokiaľ pozorujeme predmet obidvoma očami, vidíme objekt priestorovo. Je to dané tým, že pravé a ľavé oko vytvára samostatný obraz, z ktorého ľudský mozog interpretuje informáciu o hĺbke.

U skenerov získavame pomocou dvojice CCD snímačov stereoskopické snímky (viď obr.2.12). Z uhlu snímaného združenými lúčmi (takzvaná uhlová paralaxa), je možné získať vzdialenosť povrchu od stredu pohľadu oboch snímačov.



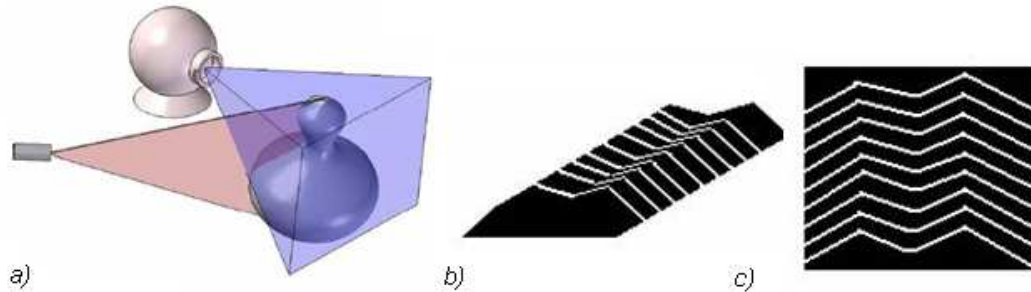
Obr.2.12.: Princíp pasívnej triangulácie

2.2.2 Aktívna triangulácia

Metóda skenovania, ktorá využívala aktívnu trianguláciu pri skenovaní laserom bola prvý krát využitá a publikovaná už v roku 1978. Technika aktívnej triangulácie spočíva v rekonštrukcii snímaného objektu, nasvietením jeho povrchu svetelným zdrojom a súčasným snímaním CCD snímačom.

Zdroj svetla spolu so snímačom a osvetleným bodom na skenovanom objekte tvorí tzv. triangulačný trojuholník. Spojnicu, svetelný zdroj – snímač, nazývame triangulačná báza. Na strane

zdroja je uhol zvieraný s triangulačnou bázou nemenný, ale na strane snímača je uhol určený premennou pozíciou vysvieteného bodu CCD snímača. Z veľkosti tohto uhla a na základe znalosti triangulačnej bázy je možné určiť z -ovú súradnicu objektu.

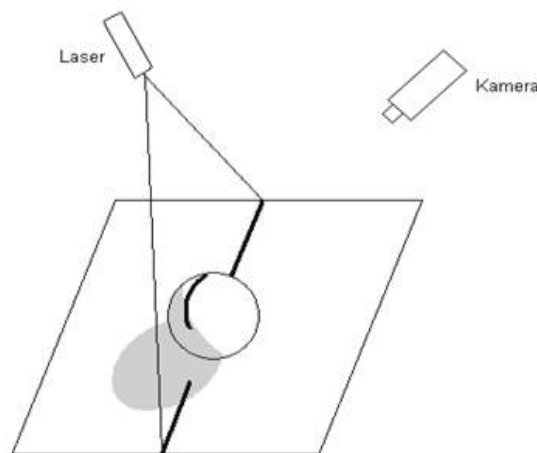


Obr.2.13.: Použitie aktívnej triangulácie pri využití svetelného lúča (a), objekt osvetlený lúčom (b), obraz vytvorený na CCD snímači (c) [2] [3].

Vzhľadom k tomu, akým typom lúča povrch nasvietime a skenujeme, trianguláciu môžeme rozčleniť do nasledujúcich kategórií:

- svetelný lúč (1D triangulácia),
- svetelný pruh (2D triangulácia),
- štruktúrovaný svetelný zväzok (3D triangulácia).

Medzi hlavnú nevýhodu triangulačnej metódy patrí fakt, že, vďaka konkavitám v skenovanom objekte nemusí byť premietaný bod, pruh či vzor vždy viditeľný, a teda nie je možné rekonštruovať povrch objektu. Na obrázku 2.6. je rovina vytvorená laserom a scénu sníma kamera. Šedé plôšky na obrázku znázorňujú oblasť, ktorú kamera nevidí. Dolná časť gule nie je osvetlená rovinou z laseru. [3]



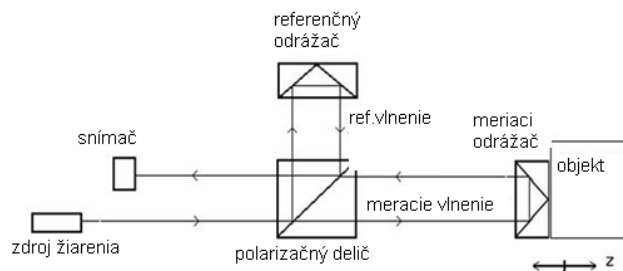
Obr.2.14.: Problém viditeľnosti u triangulačných metód [3].

2.2.3 Meranie doby letu svetla

Meranie vzdialenosti touto metódou využíva tzv. laserový diaľkomer. Diaľkomer meria čas za ktorý sa laserový pulz vráti od odrazeného (skenuvaného) objektu, naspäť k vysielaču. Pri meraní sa využije známa – konštantná rýchlosť svetla. Presnosť takéhoto skeneru je závislá od toho s akou chybou sme schopní čas letu svetla odmerať. Vzhľadom k tomu, že svetlo prejde jeden milimeter približne za 3.3 pikosekundy, je táto metóda pre skenovanie veľkých detailov objektu nie príliš presná. Využitie samozrejme nachádza tam, kde predmetom digitalizácie sú veľké objekty (pamiatky, budovy), kde presnosť v rádoch milimetra nehrá rolu.

2.2.4 Optická interferometria

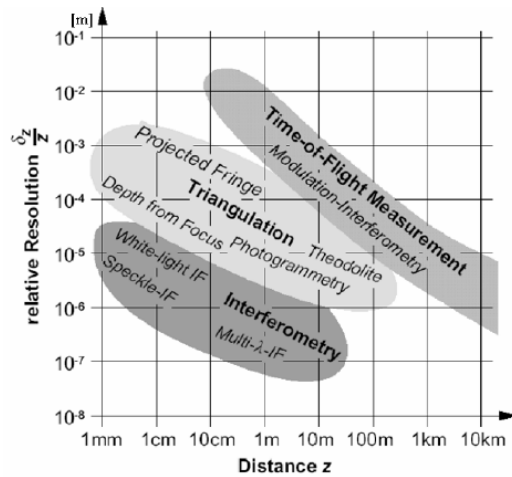
Metódy optickej interferometrie (viď obr.2.15.), sú založené na princípe merania času letu koherentného žiarenia. Vlnenie svetelného lúča je rozdelené na merané a referenčné. Obidve vlnenia spolu interferujú a výsledok je zachytený snímačom. Pri meraní sa využíva vlastnosť polarizačného deliča a dvoch odrážačov. Táto metóda umožňuje merať iba relatívne vzdialenosti skúmaných bodov. Meranie absolútnych vzdialeností je možné po kalibrácii.[3]



Obr.2.15.: Schéma optického interferometra.[3].

2.2.5 Presnosť jednotlivých metód

Nároky na presnosť skenovania sa môžu líšiť. V niektorých prípadoch hrá prvoradú úlohu portabilita zariadenia pred jeho rozlíšením. Na nasledujúcom obrázku je uvedené porovnanie presnosti jednotlivých metód (obr.2.16.) [7].

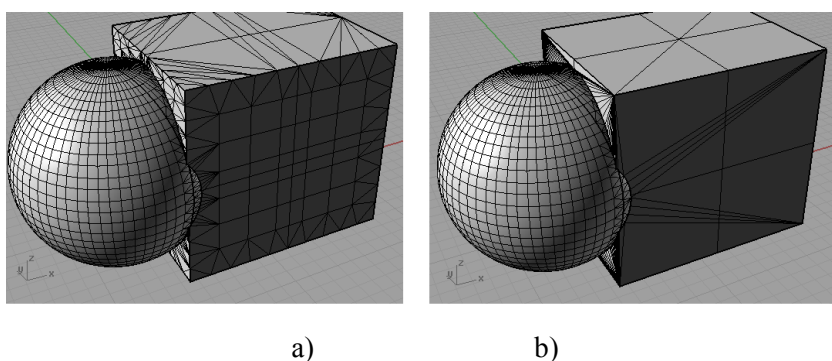


Obr 2.16.: Presnosť jednotlivých metód merania v závislosti na vzdialenosti.[7]

3 Spracovanie modelu scény

Geometrické modely, ktoré sú produktom skenovania, modelovania či akejkoľvek inej metódy, často nevyhovujú podmienkam pre zobrazenie v aplikácii. Povrchy, ktoré získavame z výškových máp sú často rovnomerne delené a hustota trojuholníkov býva rovnaká tak, ako pre ploché, tak aj zaoblené časti modelu (viď obr.3.1).

Väčšinou sa jedná o defekty spôsobené použitou metódou a preto je nutné polygonálny model pred použitím upraviť do požadovanej formy. Medzi základné algoritmy používané na modifikáciu polygonálnej siete patria vyhladzovanie a decimácia siete. Práve tieto ďalšie dva spôsoby, ktoré nájdú v tejto diplomovej práci využitie, popíšem v nasledujúcich častiach.



Obr.3.1.: Model pred (a) a po (b) odstránení prebytočných polygónov. Pri odstránení nedošlo k žiadnej zmene geometrie objektu.

Obrázok vygenerovaný pomocou aplikácie Rhinoceros 4.0

3.1 Získanie vstupných dát

K tomu aby bolo možné aplikovať algoritmy pre úpravu polygonálnej siete, je potrebné zvoliť vhodný spôsob uloženia polygonálnych modelov, s ktorými by spolupracovali v prípade použitia aj iné CAD systémy. Pre prezentácie prostredníctvom webového rozhrania sa nám javí najlepšie použiť formát VRML, ktorý je podporovaný rôznymi pluginmi v internetových prehliadačoch.

Tento formát nie je v našom prípade vhodný pre uloženie modelov, pretože jeho štruktúra je náchylná k chybám a vyžaduje rozsiahle validácie, ktoré by spomaľovali našu aplikáciu. Preto tento formát nebude určený pre internú reprezentáciu dát, ale až k jej finálnej prezentácii prostredníctvom WWW.

Pretože v tejto práci sa pracuje s istými vybranými formátmi súborov s polygonálnymi modelmi, rád by som ich v tejto časti popísal detailnejšie. Implementáciou načítania a spracovania sa bude zaoberať samostatná časť.

3.1.1 VRML - Virtual Reality Modelling Language

VRML je grafický formát založený na deklaratívnom programovacom jazyku, ktorý bol navrhnutý predovšetkým pre popis trojrozmerných scén obsahujúcich aktívne aj pasívne objekty pre použitie v aplikáciách virtuálnej reality. Vo VRML je scéna popísaná pomocou objektov, ktoré sú usporiadané v stromovej štruktúre. Pre nás je použitie tohto formátu veľmi výhodné hlavne z toho hľadiska, že v internetových prehliadačoch je možné doinštalovať plugin, ktorý umožňuje tieto súbory priamo prehliadať. Tento formát, ktorý je mimo iné definovaný normou ISO, je veľmi komplexný na to aby mohol byť použitý ako zdroj dát pre algoritmy spracúvajúce model scény. Preto tento formát príde do styku s užívateľom úplne na konci – pri prezentácii modelov.

3.1.2 STL – Stereolithography formát

Tento súborový formát je určený pre uloženie telies reprezentovanými polygónmi. Ako už názov napovedá, primárnym účelom tohto formátu bol prenos dát na zariadenie pre stereolitografiu (rapid prototyping pomocou laserom tvrdeného fotopolyméru). Tento formát je vzhľadom k svojej jednoduchosti používaný v mnohých CAD systémoch a so stereolitografiou nemusí súvisieť. Existujú dve verzie formátu :

- Textový (ascii)
- Binárny

Stl súbor neobsahuje žiadnu špecifikáciu povrchu objektu, ale iba vrcholy trojuholníkov a k nim korešpondujúce normály. Vzhľadom k veľkosti sa textové STL súbory moc nepoužívajú. Rozšírenejšia je ich binárna podoba, ktorá je úspornejšia. Použitie tohto súborového formátu v tejto diplomovej práci rozširuje jej všeobecnú použiteľnosť aj na iné modely, ktoré sú produkované inými CAD aplikáciami.

3.1.3 TRI – Interný formát knižnice VectorEntity

Štruktúra, ktorá zachytáva iba informáciu o tvare telesa je veľmi podobná spôsobu uloženia v súbore typu VRML, avšak bez rozširujúcich údajov. Súbor obsahuje zoznam vrcholov – ich súradnice a zoznam indexov. Trojice indexov reprezentujú jednotlivé trojuholníky polygonálnej siete.

Použitie interného formátu bolo podmienené z nasledujúceho dôvodu. Pri spracovaní modelov, nech už sa jedná o decimáciu alebo vyhladzovanie dochádza na polygonálnej sieti ku zmenám v takej presnosti, ktorá pri súborovom formáte STL nemusí byť zachytená. STL totiž využíva pri ukladaní polohy 4bajtový dátový typ float, kdežto implementácia TRI využíva 8bajtový double.

Ďalšou výhodou je fakt, že súbor tri korešponduje s dátovými štruktúrami reprezentujúce polygonálny model (zoznam trojuholníkov, ktorého vrcholy sa odkazujú zo zoznamu vrcholov).

Výhodou je jednoduché a rýchle načítanie bez rozsiahlych validácií, ako je tomu u VRML.

3.2 Laplaceovo vyhladzovanie polygonálnej siete

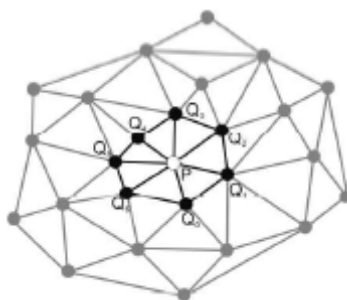
V predošlej časti bol popísaný spôsob získania polygonálnej reprezentácie dát. V ďalších častiach by som uviedol algoritmy, ktoré upravujú geometrickú alebo topologickú časť objektu.

Laplaceova [7] metóda, alebo tiež filter, je najjednoduchšia a najčastejšie používaná metóda na vyhladzovanie povrchov 3D modelov. Využíva topologické okolie bodu a produkuje kvalitne vyhladené modely už v niekoľkých prvých iteráciách. Táto metóda nastavuje pozíciu každého vrcholu do geometrického stredu susedných vrcholov.

Laplaceov filter obsahuje dva parametre – počet iterácií a váhový faktor λ , ktorý reguluje dopad susedných vrcholov na novo spočítanú pozíciu.

Nový bod p' je spočítaný zo starej pozície bodu p a jeho susedov q , tak ako je to uvedené v nasledujúcom výraze:

$$p' = p + \frac{\lambda}{n} \sum_{i=0}^{n-1} (q_i - p) \quad (3.1)$$



Obrázok 3.2: Vplyv okolitých vrcholov na novú pozíciu. Prevzaté z [7]

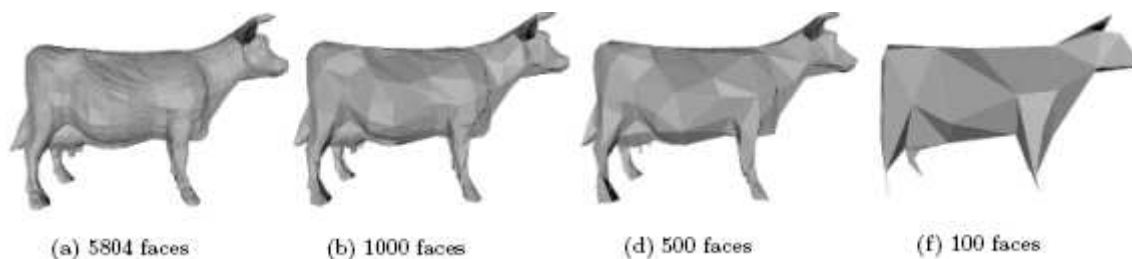
Metóda nie je náročná na výpočtový výkon počítača, avšak nezaručuje zlepšenie kvality obzvlášť pri spracovávaní elementov ktoré zvierajú s inými uzlami siete príliš veľký, alebo príliš malý uhol. Tento problém rieši modifikácia Laplaceovho algoritmu, ktorá overuje, či po úprave uzlu sa zlepši kvalita siete. Pokiaľ sa kvalita nezlepší, uzol je z daného výpočtu vypustený.

3.3 Zjednodušovanie povrchov

Hlavným dôvodom k znižovaniu počtu trojuholníkov je potreba optimalizovať reprezentáciu povrchu. Menej zložitá reprezentácia je menej náročná na pamäťové prostriedky, ako aj na celkový výkon počítača. Metódy zjednodušovania povrchu môžeme deliť podľa toho, či zachovávajú, alebo nezachovávajú topológiu siete, alebo podľa toho či nová reprezentácia ponecháva podmnožinu pôvodných vrcholov alebo je vytvorená prevzorkovaním z úplne nových bodov.

Bežné modely reálnych objektov pozostávajú z pevne daného počtu trojuholníkov. V prípade že máme model ktorý je od pozorovateľa vzdialený, nám stačí na veľmi verné zobrazenie originálu iba zlomok pôvodných polygónov. Práve toto je úloha decimačných algoritmov – nájsť najlepšiu aproximáciu povrchu pri zachovaní maximálnej miery detailov.

Prvý krát sa tejto vlastnosti využilo v oblasti leteckých simulátorov. Často sa toho doteraz dosahuje spôsobom, že model obsahuje niekoľko úrovní detailov a podľa vzdialenosti sa určuje, ktorý model sa má pre zobrazenie použiť.

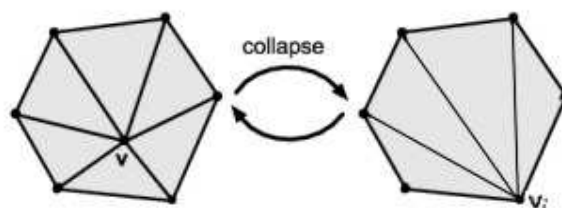


Obr. 3.3: Rôzne úrovne detailov modelu. Ukážka simplifikačného algoritmu. Prevzaté z [5]

V nasledujúcej časti uvediem prehľad tých najzákladnejších kategórií metód pre redukciu polygonálnych sietí.

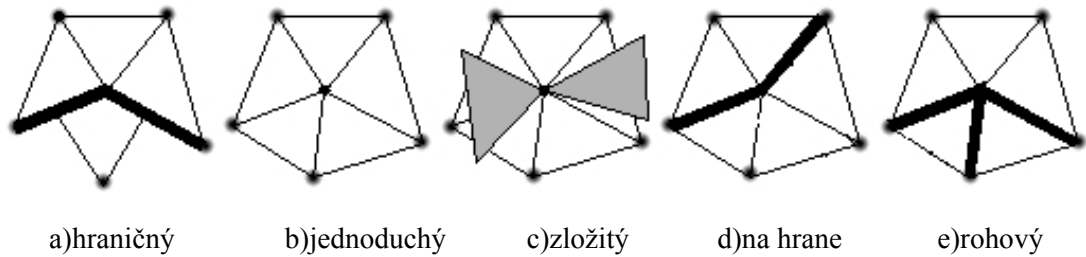
3.3.1 Redukcia vrcholov

Asi najznámejší [4] decimačný algoritmus zachovávajúci topológiu aj množinu pôvodných vrcholov vytvoril W. Schroeder (1992). Algoritmus prechádza celú polygonálnu sieť, odstraňuje vrcholy podľa decimačného kritéria a takto vzniknuté diery trianguluje. Odstraňované sú vrcholy s najmenšou dôležitosťou.



Obrázok 3.4: Odstránenie vrcholu a jeho následná triangulácia. Prevzaté z[4]

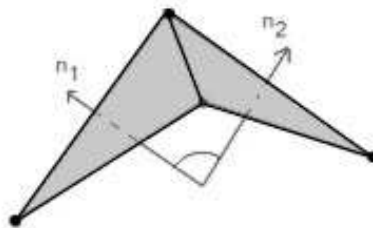
V prvom priechode algoritmu polygonálnou sieťou sú vrcholy klasifikované do piatich kategórií (viď. obr. 3.5).



Obrázok 3.5.: Klasifikácie vrcholov. Prevzaté z[4]

Vrchol, ktorý je obklopený trojuholníkmi tak, že všetky hrany z neho vychádzajúce sú zdieľané práve dvoma trojuholníkmi, je označený ako jednoduchý (simple). Vrchol, ktorý je na hranici siete, označujeme ako hraničný (boundary). Pokiaľ je nejaká hrana spoločná s viacerými než dvoma trojuholníkmi, alebo sa vo vrchole dotýka trojuholník, ktorý má s ostatnými trojuholníkmi okolo tohto vrcholu spoločný práve bod, je vrchol označený ako zložitý (complex).

Pre posledné dve kategórie klasifikácie musíme od jednoduchých vrcholov nejako odlíšiť. Významnou hranou rozumieme takú hranu, ktorej príslušné dva trojuholníky, resp. ich normály, zvierajú priestorový uhol väčší než zadaný medzný uhol.



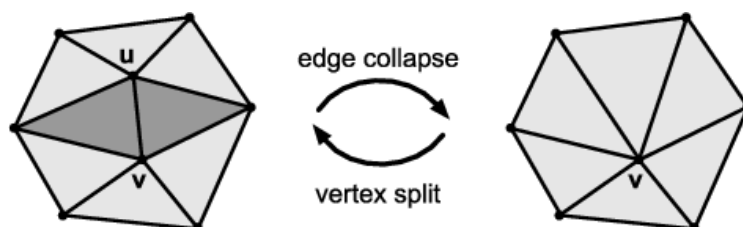
Obrázok 3.6.: Klasifikácia hraničných vrcholov. Prevzaté z[4]

Pre vypočítanie decimálneho kritéria jednoduchých vrcholov, tj. zistenie dôležitosti vrcholu, sa používajú rôzne heuristiky. Ohodnotenie vrcholu znamená, akú geometrickú chybu zanesie do modelu prípadné odstránenie vrcholu.

Pôvodný Schroederov algoritmus využíva heuristiku zvanú Average Normal, ktorá určuje mieru dôležitosti vrcholu ako hodnotu dĺžky priemernej normály vypočítanej z normál trojuholníkov obklopujúcich vrchol. Medzi ďalšie heuristiky patria Height difference, Absolute Binary a ich kombinácie často vytvorené pre špecifický účel.

3.3.2 Kontrakcia hrán

Rodina týchto metód je založená na eliminácii celých hrán. Hrany sú eliminované kontrakciou, kde jeden vrchol nahradzuje obidva koncové body hrany (obr. 3.7). Operácia nevyžaduje aby bolo okolie typu manifold, tj. topologicky zhodné s „diskom“.



Obrázok 3.7.: Kontrakcia hrany. Prevzaté z [4].

Možnosť kontrakcie hrán pre simplifikáciu polygonálneho modelu po prvý krát využil Hopp v roku 1993 a následne Garland v roku 1997. Pred vlastnou kontrakciou je potrebné vrcholy (hrany) ohodnotiť z hľadiska dôležitosti.

Jedným z hlavných problémov pri kontrakcii hrán je zachovanie topológie siete. Najčastejším problémom býva tzv. preklopenie siete. Pred každou kontrakciou je preto nutné urobiť ešte tzv. kontrolu konzistencie novej siete, kde zisťujeme či nedochádza k degenerácii.

Najznámejšiu metódu, ktorú vytvoril Garland využíva pre ohodnotenie kvadriky. Keďže táto metóda, bude s najväčšou pravdepodobnosťou predmetom implementácie, rád by som ju v nasledujúcej časti prebral detailnejšie.

3.4 Quadric error metrics

[4] Tento algoritmus bol publikovaný v roku 1997 Michaelom Garlandom a Paulom Heckbertom. Algoritmus tak ako bolo spomenuté pracuje na princípe iteratívnej kontrakcie hrán. Pre svoju jednoduchosť, rýchlosť a robustnosť je jedným z najviac atraktívnych algoritmov.

Garlandov algoritmus používa pre ohodnotenie hrán práve kvadriky, ktoré dosahujú výborné výsledky pri aproximácii originálu. Na rozdiel od predošlého Schroederova algoritmu vstupom tohto algoritmu je namiesto kontajneru vrcholov kontajner hrán. Hranou rozumieme dvojicu vrcholov, ktoré sú navzájom previazané.

Algoritmus prechádza každú jednotlivú hranu, ohodnotí ju pomocou kvadriky. Po určení dôležitosti môže nastať iteratívne odstraňovanie hrán, až pokiaľ sa nedosiahne žiadaného efektu.

3.4.1 Ohodnotenie vrcholov pomocou kvadrik

Kvadrika je symetrická matica s rozmerom 4x4, ktorá zachycuje pre daný vrchol súčet štvorcových vzdialeností k ostatným rovinám.

Garland[5] používa pre zápis kvadriky alternatívnu homogénnu formu, kde kvadrika je definovaná ako trojica $Q = (A, b, c)$, v ktorej A je matica rozmeru 3x3, b je vektor v R^3 a c je skalár.

$$Q = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix} \quad (3.1)$$

V našom prípade môžeme asociovať množinu rovín, ktoré obklopujú daný vrchol a tým pádom definovať chybu vrcholu vzhľadom k tejto množine (súčtu jeho štvorcových vzdialeností k rovinám). Výsledkom spočítania kvadriky teda dostaneme informáciu, ktorá zachycuje susediace roviny :

$$\Delta(v) = \Delta([v_x v_y v_z 1]^T) = \sum_{p \in \text{planes}(v)} (p^T v)^2 \quad (3.2)$$

kde $p = [a \ b \ c \ d]^T$ reprezentuje rovinu definovanú vzťahom $ax+by+cz+d=0$. Vzťah 3.2 môžeme prepísať do kvadratickej formy :

$$\Delta(v) = \sum_{p \in \text{planes}(v)} (pv^T)(p^T v) = \sum_{p \in \text{planes}(v)} v^T (pp^T) v = v^T \left(\sum_{p \in \text{planes}(v)} K_p \right) v \quad (3.3)$$

kde K_p je elementárna matica, ktorá je symetrická podľa hlavnej diagonály.

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (3.4)$$

Výslednú kvadriku teda dostaneme súčtom všetkých elementárnych matíc, v ktorej premenné a, b, c predstavujú súradnice normálových vektorov rovín.

Kvadrika Q priradzuje hodnotu $Q(v)$ bodu v v priestore podľa nasledujúceho vzťahu.

$$Q(v) = v^T A v + 2b^T v + c \quad (3.5)$$

Po vyhodnotení dostaneme súčet štvorcov vzdialeností vrcholu k rovinám.

$$Q(v) = (v^T Q v) =$$

$$q_{11} x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x +$$

$$q_{22}y^2 + 2q_{23}yz + 2q_{24}y +$$

$$q_{33}z^2 + 2q_{34}z +$$
(3.6)

V prípade, že prevedieme kontrakciu do vrcholu v_k nad dvoma vrcholmi v_i a v_j , ku ktorým náležia kvadriky Q_i a Q_j , tak výsledná kvadrika Q_k sa bude rovnať súčtu týchto kvadrík.

Chyba, ktorá kontrakciou vznikne sa rovná $(v_k^T Q v_k)$, reprezentuje súčet vzdialeností bodu v_k , k všetkým rovinám zahŕňajúce obidve kvadriky Q_i a Q_j , následovne:

$$Q(\bar{v}) = Q_i(\bar{v}) + Q_j(\bar{v})$$
(3.7)

3.4.2 Nájdenie optimálnej pozície nového vrcholu

Prevzaté z [5]. V tomto bode už vieme spočítať aká je vzniknutá chyba pri kontrakcii hrany. Pre získanie najlepšej aproximácie potrebujeme vedieť, kde umiestniť výsledný bod, aby chyba, ktorá takto vznikne bola najmenšia. Pri eliminácii hrany $(v_1, v_2) \rightarrow v$ môžeme buď vybrať bod, ktorý má lepšie ohodnotenie, alebo nájsť optimálnu pozíciu pre nový bod. Tento spôsob je časovo náročnejší, ale dosahuje lepšie výsledky. Vychádzame z funkcie $Q(v)$, ktorá reprezentuje vzniknutú chybu je minimálna vtedy, keď platí podmienka:

$$\frac{\delta \Delta}{\delta x} = \frac{\delta \Delta}{\delta y} = \frac{\delta \Delta}{\delta z} = 0$$
(3.8)

Po spočítaní parciálnych derivácií dostaneme gradient Q , ktorý je:

$$\nabla Q(v) = 2Av + 2b$$
(3.9)

Vyriešením rovnice $\nabla Q(v) = 0$ dostaneme optimálnu pozíciu nového vrcholu:

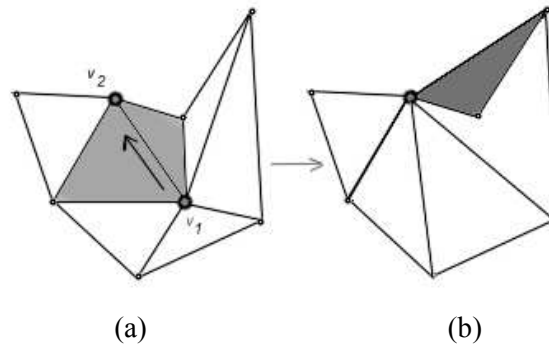
$$\bar{v} = -A^{-1}b$$
(3.10)

$$\bar{v} = - \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}^{-1} \begin{bmatrix} q_{14} \\ q_{24} \\ q_{34} \end{bmatrix}$$
(3.11)

Pokiaľ matica nie je invertovateľná (je singulárna), pre kontrakciu sa vyberie vrchol ktorý má menšie ohodnotenie, resp. jeho stredová hodnota. Táto situácia môže nastať napríklad vtedy, keď všetky vrcholy ležia v jednej rovine.

3.4.3 Kontrola konzistencie siete

Jedným z hlavných problémov pri kontrakcii hrán je zachovanie topológie siete. Pri postupnej kontrakcii hrán, môže nastať situácia, kde dôjde k degenerácii trojuholníkovej siete.(viď obr. 3.8)



Obr.3.8.: Preklopenie siete pri postupnej kontrakcii hrán pred(a), po(b). Prevzaté z [5]

Tento problém je možné vyriešiť porovnaním normálových vektorov rovín pred a po úprave siete. Pokiaľ prekročí zmena normálového vektoru roviny danú prahovú hodnotu, hrana je z decimálneho procesu vyradená.

4 Popis vlastností skeneru

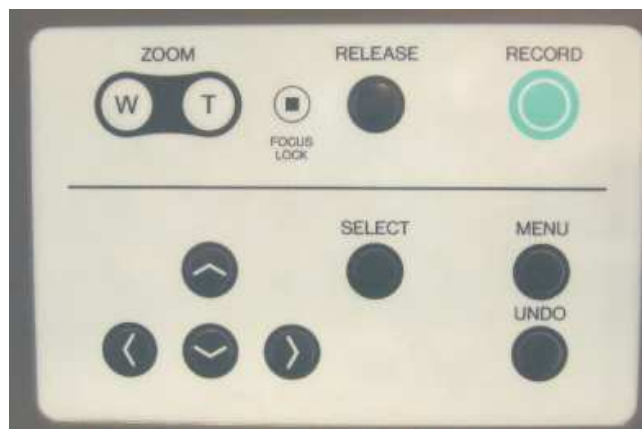
V tejto časti by som rád čitateľovi rád uviedol praktické skúsenosti, s ktorými som sa stretol pri experimentálnej práci so skenerom.

4.1 Popis zariadenia

Samostatný skener má vzhľadom k svojej veľkej hmotnosti na spodnej strane maticu pre uchopenie na statív (obr 4.1), čím je veľmi uľahčená manipulácia a jeho prenášanie, resp. nastavenie pozorovacieho uhla pod ktorým skenujeme. K skeneru je potrebný zdroj 220V, pretože model Minolta Vi700 neobsahuje pre napájanie batérie.

K skeneru nie je pripojený riadiaci počítač, takže všetky nastavenia je potrebné realizovať cez jeho ovládací panel a LCD displej. Pomocou tohto panelu je možné prezerat', mazať jednotlivé snímky, približovať a oddaľovať skenovaný predmet.

K skeneru existuje otočný stolček, ktorý je pripojený cez sériový port počítača, takže pokiaľ používame skener v teréne, použitie stolčeka nie je možné. Mimo iné, pri samotnom skenovaní je potrebné otvoriť sklápacie dvierka, ktoré chránia optiku laseru pred mechanickým poškodením. Pokiaľ dvierka zabudneme otvoriť, skener nás na túto skutočnosť upozorní na prehľadovom displeji.



Obr.4.1. Skener Minolta Vi-700 umiestnenom na statíve(vpravo) a jeho ovládacie prvky skeneru(vľavo)

Skenované snímky sú ukladané na pamäťovú kartu typu PCMCIA(PC-CARD) s kapacitou 40 MB, takže pokiaľ nepoužívame rozhranie SCSI(obr 4.5.), je možné dané snímky nahrat' do počítača pre ďalšie spracovanie priamo z karty. Karta sa po zasunutí chová ako samostatný disk a snímky sú uložené v jej koreňovom adresári. Na túto kartu je možné uložiť približne 35 snímkov, v prípade, že

zaznamenávame textúru objektu (veľkosť snímku cca 1,2Mb). V prípade, že požadujeme iba informáciu o tvare, je možné na kartu uložiť až 100 snímkov (veľkosť snímku cca 300Kb).

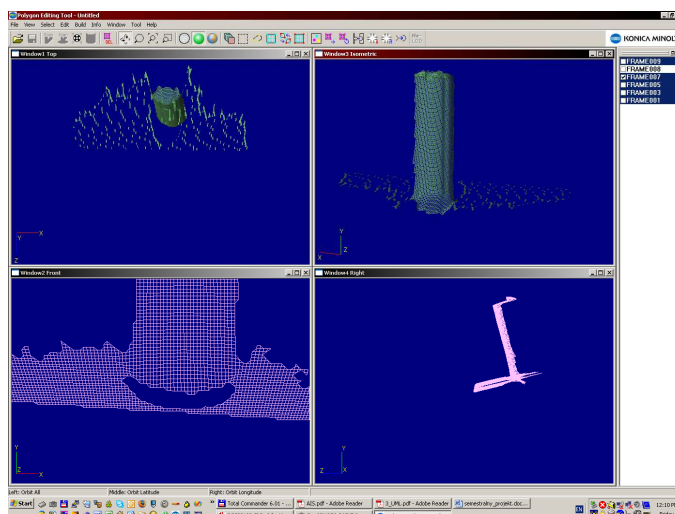


Obr.4.2.: Rozhrania skeneru (vľavo) a jeho optický snímač v dolnej časti a sklápacie dverka s laserom (vpravo).

Aj keď o skenovanie modelov pomocou 3D skeneru môže zdať jednoduchou záležitosťou, cena tohto skenera, ktorý bol našou fakultou zaobstaraný v roku 1999, vrátane všetkých doplnkov prevyšovala 1,3 milióna českých korún.

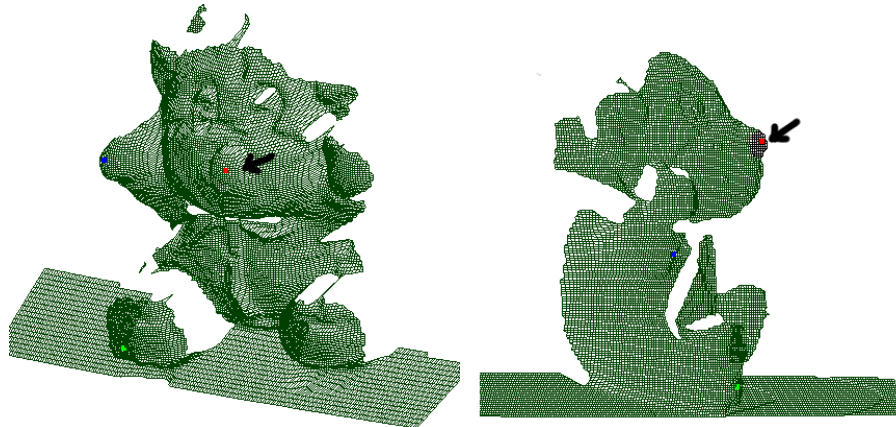
4.2 Dodávaný softvér a rekonštrukcia objektu

Pre vyskúšanie práce s 3D skenerom som mal k dispozícii demonstračnú verziu programu Polygon Editing Tool (ďalej P.E.T.) verzia 2.20, dodávaného výrobcom. Keďže sa nejedná o plnú verziu programu, tak tento softvér neumožňuje pokročilejšie operácie nad polygonálnou sieťou ani ukladanie alebo export do známych formátov.

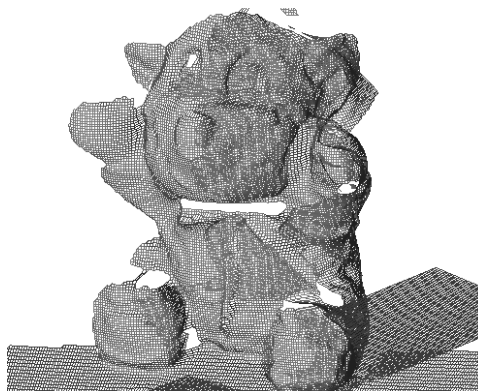


Obr.4.3.: Experiment zo skenovaním objektu (magnetická stierka) z jednej strany. Pod stierkou je vidieť nežiaduce zoskenovanie podložky, ktorú je v editačnom softvéri potrebné odstrániť.

Pre registráciu bodov a spájanie jednotlivých pohľadov do celistvého modelu prebieha nasledujúcim spôsobom. V prvom kroku nainportujeme do aplikácie P.E.T. jednotlivé CAM súbory, ktoré sú uložené na PCMCIA karte. V zozname súborov(umiestnenom na pravej strane obrazovky aplikácie) vyberieme modely určené k registrácii. V prípade manuálnej registrácie sa nám otvorí okno s týmito dvoma modelmi(vid' obr.4.4), na ktorých kliknutím vyberieme minimálne 3 korešpondujúce body. Po výbere bodov softvér automaticky nájde vhodnú transformáciu a informuje užívateľa o vzniknutej chybe(vid' obr.4.6 a 4.7). Ako ukážkový model, som vybral plyšového medveďa, pretože odrazovými vlastnosťami jeho povrchu veľmi dobre vyhovuje skeneru.

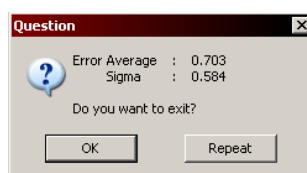


Obr.4.4.:Výber bodov pre manuálnu registráciu. Farby boli pre zvýraznenie kontrastu oproti aplikácii upravené.



Obr.4.5.: Model rekonštruovaný z dvoch pohľadov.

Keďže demonštračná verzia programu P.E.T. neumožňuje ukladanie modelov, nebolo možné v tejto demonštrácii ukázať kvalitne vyrenderovaný výstup, tak ako je to napríklad v prílohe B.1.



Obr.4.6.:Dialógové okno aplikácie P.E.T. informujúce užívateľa o chybe registrácie.

4.3 Problémy pri skenovaní

Vo všeobecnosti platí, že zo zvyšujúcim počtom snímok z rôznych uhlov, sa znižuje počet slepých miest obrazu. Kvalita výsledných snímok, je závislá jednak od vzdialenosti od žiarica laseru ako aj svetelných podmienok. Pre úspešné skenovanie je potrebné, aby objekt bol najlepšie matný a laserový paprsok sa od objektu neodrážal.

V prípade skenovania lesklých objektov, je potrebné na predmet naniest' špeciálnym difúznym sprejom prášok, ktorý zlepšuje kvalitu výslednej výškovej mapy. Tento prášok je možné po ukončení skenovacej procedúry bez problémov odstrániť jemnou látkou.

Vzhľadom k vplyvu okolitého osvetlenia a farbe objektu je niekedy potrebné meniť energiu laserového paprsku. Táto zmena je možná prostredníctvom ovládacieho panelu. Pokiaľ má lúč malú energiu, nemusí ju optický čip zaznamenať, ale pokiaľ by bola energia lúča príliš veľká, dochádzalo by k chybnéj interpretácii tvaru objektu. Preto je potrebné zvoliť určitý kompromis. Vďaka prehľadovému displeju skenera, je toto možné veľmi dobre korigovať.

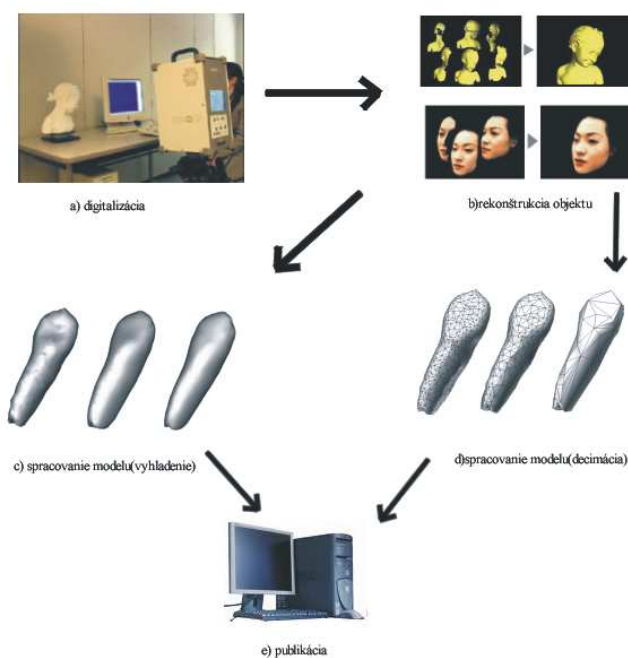
5 Návrh publikačného systému

Proces vzniku a jeho následného spracovania modelu s kombináciou publikácie prostredníctvom www stránok by bolo možné rozdeliť do nasledujúcich častí. V prvej fáze (obr.5.1.a) je potrebné získať digitálnu reprezentáciu objektu. Túto prvú časť nám zabezpečuje digitalizačné zariadenie, ktorého výstupom je najčastejšie výšková mapa. Model, ktorý je naskenovaný z rôznych uhlov je potrebné previesť do jedného kompaktného modelu. O rekonštrukciu sa stará softvér dodávaný k digitalizačnému zariadeniu, ktorý automaticky spojí pomocou vhodnej transformácie dostupné výškové mapy. Keďže formát výškových máp(CAM súbory) nie je známy a nebolo ho možné odhadnúť(obsahuje okrem výškovej mapy aj iné dáta – informácie o textúre a pod.) je nezbytné nutné použiť softvér dodávaný výrobcom. Takýto model, ktorý je zdigitalizovaný a rekonštruovaný (obr.5.1.b) postupuje do ďalšej fáze spracovania.

Takýto model obsahuje veľké množstvo rovnomerne rozložených trojuholníkov, ktorých počet je konštantný pre rovné aj zaoblené časti objektu. Z tohto hľadiska je veľmi žiadané, aby tie trojuholníky, ktoré sú prebytočné boli zdecimované niektorým s ďalej uvedených algoritmov (viď. obr.5.1d).

V tejto fáze môžeme nad modelom aplikovať ľubovoľný algoritmus, ktorý nám umožní zvýšiť jeho kvalitu, príkladom môže byť triangulácia „slepých miest“ skeneru, vyhladenie povrchu objektu, alebo odstránenie objektov, ktoré do scény nepatria.

Takýto model, ktorý je upravený je možné využiť pre ďalšie ľubovoľné účely. V našom prípade to bude publikácia prostredníctvom HTML stránok.



Obr.5.1.: Proces spracovania modelu. Ilustrácia.

6 Popis implementácie

Jedna z hlavných úloh práce bola implementácia metód pre spracovanie polygonálnych modelov a ich publikácia prostredníctvom webového servera. Ako prostriedok pre implementáciu spracovania modelov bola využitá knižnica VectorEntity, ktorá obsahuje množinu tried a operácií pre sofistikovanú úpravu a navigáciu v polygonálnej sieti.

Ako implementačný jazyk som si zvolil kombináciu 3 jazykov : C#, C++ a PHP a preto implementačnú stránku diplomovej práce je možné rozdeliť do troch častí:

1. Spracovanie modelov scény (C++)
2. Užívateľské rozhranie pre vizualizáciu a ich publikáciu (C#)
3. Implementácia publikačného servera (PHP)

Zvolenie kombinácie 3 rozdielnych platforiem, bolo motivované na základe predchádzajúcich skúseností ako tiež vhodnosti ich použitia. Pre spracovanie modelov scény bola vytvorená aplikácia v jazyku C++, hlavne z dôvodu jej rýchlosti a vhodnosti pre použitie v oblasti počítačovej grafiky.

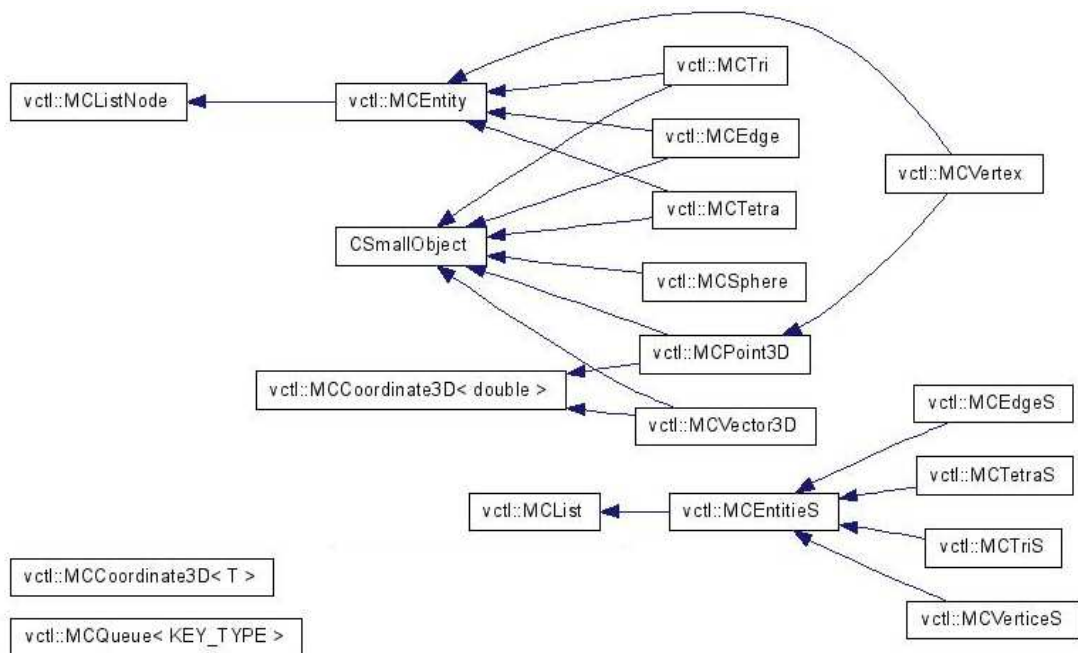
Táto aplikácia ktorá sa stará o spracovanie modelu je spúšťaná ako externý proces z užívateľského rozhrania, ktoré umožňuje zobrazenie, organizáciu pre publikáciu, vytváranie projektov atp.

Táto aplikácia je implementovaná v jazyku C#, hlavne kvôli svojej vhodnosti pre tvorbu užívateľských rozhraní. Tento bod by mohol byť implementovaný taktiež v jazyku C++, ale hlavným dôvodom zavedenia tejto platformy do diplomovej práce bola motivácia porovnania výkonnosti týchto dvoch rozdielnych implementácií.

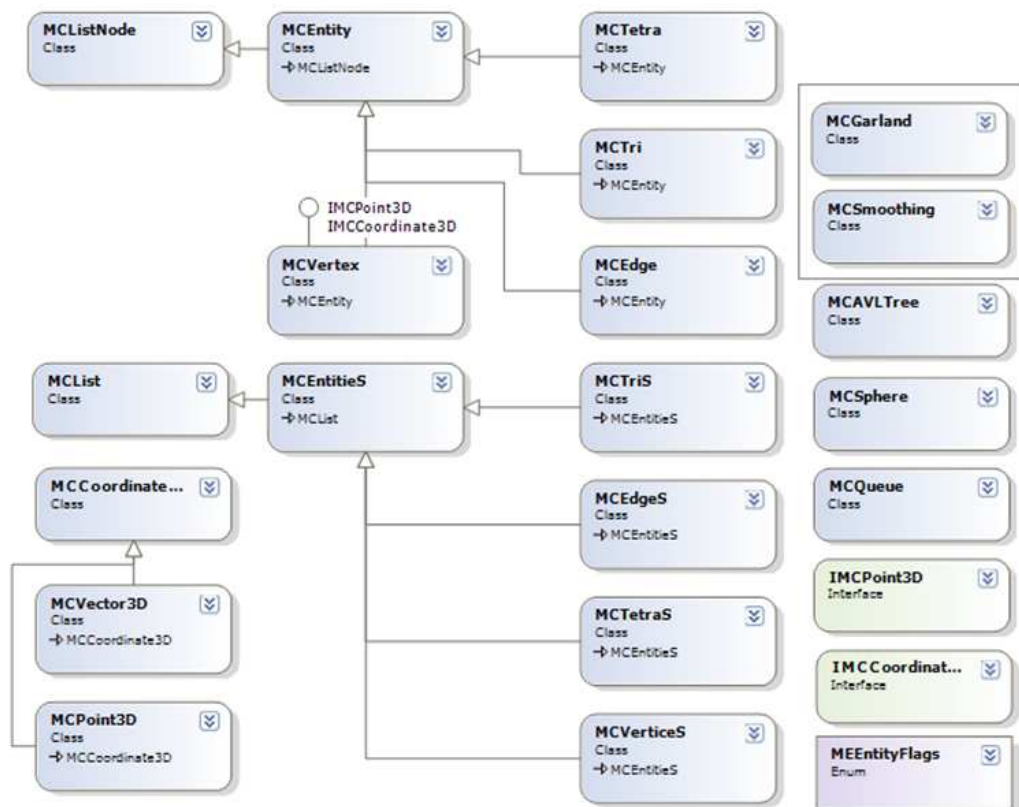
Keďže pôvodná knižnica VectorEntity bola implementovaná využitím jazyka C++, bolo nutné celú knižnicu prepracovať do cieľového jazyka. Medzi základné implementačné obmedzenia jazyka C# patria neprítomnosť knižnice STL, rozdielna práca s pamäťou, neprítomnosť mnohonásobnej dedičnosti (pre porovnanie rozdielov viď priložené diagramy tried. Obr 6.1,6.2) a množstvo iných „zaujímavostí“, ktoré bolo nutné pre vytvorenie užívateľského rozhrania vyriešiť.

Implementácia zámerne využíva bezpečný kód, pretože jeden z mojich ďalších cieľov, je otestovanie výkonnosti danej platformy a tým tiež overenie vhodnosti použitia pre riešenie problémov v oblasti počítačovej grafiky.

V prílohe B sa nachádza 17 tried, ktoré obsahujú implementáciu knižnice Vectorentity v jazyku C# (obr. 6.2), ako aj implementáciu tried pre vyhladenie a decimáciu polygonálnych modelov v jazyku C++ (obr. 6.1).



Obr. 6.1: Diagram výslednej implementácie knižnice Vectorentity v jazyku C++.
Generované pomocou dokumentačnej aplikácie doxygen.



Obr. 6.2: Diagram výslednej implementácie knižnice Vectorentity v jazyku C#.
Náhľad bol vygenerovaný pomocou podporných pomôcok Visual Studia.

V pôvodnej knižnici (C++) sa nenachádza trieda MCAVLTree, ktorá zapuzdruje operácie nad vyváženým binárnym stromom. Tento strom je implementovaný ako náhrada multimapy knižnice STL, ktorá slúži k radeniu prvkov do fronty (MCQueue).

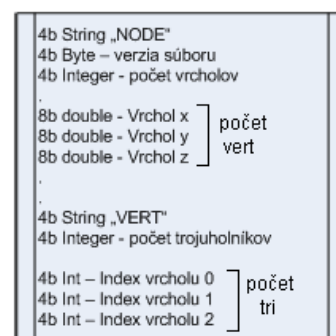
6.1 Načítanie vstupných dát

Pokiaľ chceme s nejakým modelom v aplikácii pracovať, musíme vybrať vhodnú reprezentáciu pre jej uloženie. Naša aplikácia by mala zvládnuť prácu minimálne s takým formátom dát, ktorý je veľmi dobre podporovaný aj inými CAD systémami. Preto okrem interného formátu VectorEntity – TRI je potrebné zaistiť možnosť práce aj s iným formátom. V našom prípade STL. V tejto časti by som popísal základné rozdiely medzi týmito dvoma formátmi a spôsob ich začlenenia do aplikácie.

6.1.1 Získanie dát z TRI súboru

Súbor TRI, ktorý je interným formátom knižnice VectorEntity je možné rozdeliť do troch častí. Prvá obsahuje hlavičku a základné informácie o súbore. Druhá časť dáta reprezentujúca vrcholy polygonálneho modelu a tretia reprezentujúca indexy jednotlivých trojuholníkov.

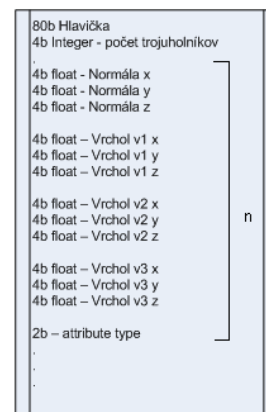
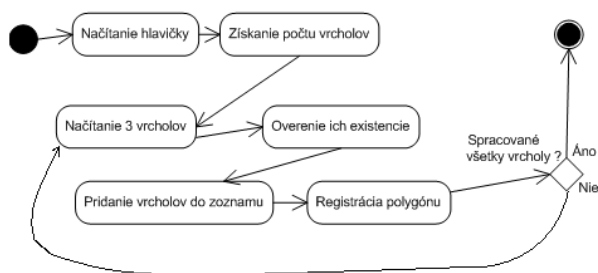
Implementačne je to riešené tak, že najprv sa do rezervovaného poľa načítajú vrcholy modelu a potom sa na základe indexov vytvoria väzby v dátových štruktúrach. Tak vznikne plnohodnotný model, s ktorým je ďalej možné pracovať.



Obr.6.3.:Štruktúra súboru TRI

6.1.2 Získanie dát z STL súboru

Pri načítaní binárneho súboru je situácia o niečo komplikovanejšia. Súbor miesto samostatných súradníc vrcholov a indexov obsahuje normály a vrcholy jednotlivých trojuholníkov. Pri načítaní vrcholov je potrebné testovať, či už sa daný vrchol v zozname nenachádza. Pokiaľ by sa v zozname vrchol, nachádzali dva rovnaké vrcholy, mohlo by dôjsť k vzniku nekonzistentnej siete a algoritmy pracujúce s topologickou časťou objektu by nefungovali. Preto pri vytváraní trojuholníku v dátových štruktúrach testujem existenciu vrcholu. Pokiaľ sa načítavaný vrchol už v dátovej štruktúre nachádza, tak miesto vytvorenia nového objektu novo vznikajúcemu (vrcholu) trojuholníku jednoducho predám ukazateľ na tento už existujúci vrchol.



Obr.6.4.: Stavový diagram načítania STL modelu (vľavo) a štruktúra súboru STL (vpravo)

6.2 Laplaceovo vyhladzovanie polygonálnej siete

Táto metóda bola zo všetkých troch implementovaných najjednoduchšia. Algoritmus je implementovaný ako samostatná trieda, pri ktorej je nutné, aby sme vedeli počet iterácií, váhový faktor a kontajner trojuholníkov MCTriS nad ktorým má byť daná operácia prevedená.

Algoritmus prechádza za sebou všetky vrcholy v kontajneri a pracuje iba s geometrickou časťou modelu. Je možné ho zhrnúť do nasledujúcich častí.:

1. Získanie súradníc okolitých bodov.
2. Vypočítanie súradnice vrcholu, z priemeru pozície ostatných vrcholov.
3. Uloženie novej pozície do dátových štruktúr
4. Presun na ďalší vrchol.

Výsledky, ktoré som dosiahol boli vynikajúce už v niekoľkých prvých iteráciách. Rýchlosťou a pojednaním o dosiahnutých výsledkoch sa zaoberá samostatná časť.

6.3 Decimácia polygonálnych modelov

Implementáciu môžeme rozdeliť do troch hlavných častí, v ktorých by som popísal postup prevedenia celého algoritmu kontrakcie hrán.

a) Získanie hrán

V prvej časti potrebujeme získať reprezentáciu modelu, ktorý nám umožňuje prevádzať operácie vyhovujúce Garlandovmu algoritmu. Modely, ktoré sú výsledkom predošlých algoritmov sú v trojuholníkovej reprezentácii, preto je pred prevádzaním vlastného algoritmu nutné vytvoriť reprezentáciu, ktorá pozostáva z hrán. Hrana je definovaná dvoma vrcholmi. K uloženiu hrán som použil triedu MCEdgeS, ktorá tento kontajner hrán zapuzdruje. Kontajner je naplnený na základe

Kontajneru trojuholníkov MCTriS, ktorý obsahuje model určený k decimácii.

b) Klasifikácia hrán

V tejto fáze algoritmu sa rozhoduje o dôležitosti kontrahovaných hrán. Implementovaný algoritmus prechádza celý kontajner hrán a ohodnocuje ich pomocou kvadrík. Ohodnotené hrany sú ukladané do fronty, ktorá je implementovaná pre rýchle vyhľadávanie pomocou dátovej štruktúry – multimapy.

Pre každú jednotlivú hranu si zistím, akú chybu nám zavedie jej odstránenie. Táto hodnota je pre nás v prípade rozhodovania dôležitosti smerodajná.

c) Kontrakcia hrán

Pri odstraňovaní si vyberám z fronty tú hranu, ktorá mi zavedie najmenšiu geometrickú chybu. Toto je opakované pokiaľ sa nedosiahne odstránenia požadovaného počtu vrcholov. Po odstránení aktualizujem chyby susedných vrcholov.

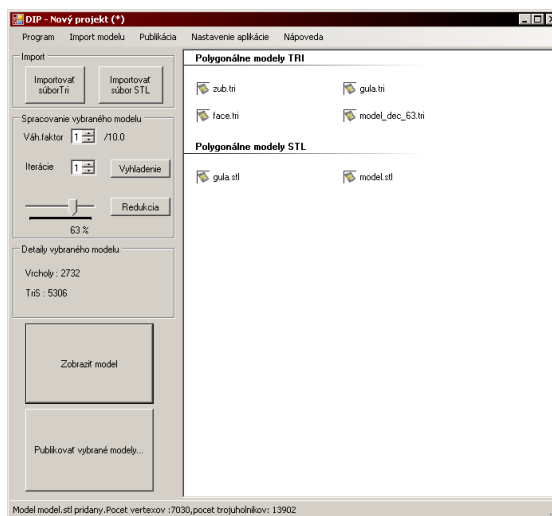
Pri kontrakcii hrany, najprv odstránim trojuholníky, ktoré sú k nej registrované. Pokiaľ hrana nemá dva susedné trojuholníky, môže byť z topologického hľadiska označená ako hranová, alebo komplexná. Takáto hrana je vyradená z fronty a decimačného procesu sa naďalej nezúčastní.

Výsledky a namerané hodnoty sú uvedené v kapitole 7. Grafické ukážky modelov, na ktoré som dané tri algoritmy aplikoval sú uvedené v prílohe C.

6.4 Popis užívateľského rozhrania

V tejto časti by som rád čitateľovi popísal jednotlivú funkcionálnosť aplikácie pre správu a publikáciu modelov. Úlohou tejto časti nie je popísať ovládanie programu, ale popis hlavných myšlienok tvoriace implementáciu. Užívateľská príručka bude uvedená v samostatnej prílohe (viď príloha D).

Hlavná myšlienka aplikácie spočíva vo vybratí modelov z disku počítača, ich načítaní a publikácii. Na obrázku 6.5 vidíme hlavné okno aplikácie, ktoré obsahuje zoznam modelov, vpravo možnosti úpravy, prezerania a publikácie, dole časť informujúca o aktuálnom stave a hore menu pre ovládanie projektov, nastavenia a nápovedy.



Obr.6.5.: Hlavné okno aplikácie pre správu a publikáciu modelov.

6.4.1 Vytvorenie, načítanie a ukladanie projektu

K tomu aby užívateľ mohol vybraný zoznam modelov aj v prípade zavretia aplikácie vrátiť, môže združovať modely do projektov, tak ako je to bežné v iných programoch. Projekt ako taký je reprezentovaný C# triedou **Project**. Táto trieda obsahuje zoznam jednotlivých modelov (trieda **ProjectItem**) v kolekcii typu **ArrayList**. Tieto dve majú nastavený príznačnosť serializovateľnosti. Vďaka tomu je jednoduché ich previesť na súbor, ktorý obsahuje všetky potrebné informácie a uložiť ich do konfiguračného súboru pre ďalšie použitie.

6.4.2 Načítanie a uloženie konfiguračných dát

Konfiguračné dáta sú uložené v triede **Settings**, ktorá je tak isto, ako projekt serializovateľná.

Pôvodný zámer bolo vytvoriť konfiguračný súbor, ktorý by sa po spustení aplikácie automaticky načítal, a pri zavretí uložil. Tým pádom by ostala aplikácia v takom nastavení ako ju užívateľ zanechal. Vďaka tomu, že som s nastavením viackrát experimentoval, bolo výhodnejšie aby aplikácia bola po spustení v určitom stave .

Preto je po spustení aplikácie inicializované implicitné nastavenie a o ukládanie a načítanie sa stará sám užívateľ. Pokiaľ by to bolo potreba zmeniť tak, ako to som spomenul, stačilo by pridať obsluhu dvoch udalostí pri načítaní a ukončení aplikácie, čo by predstavovalo cca. 6 riadkov kódu.

6.4.3 Nastavenia externých aplikácii pre úpravy siete.

Keďže aplikácia využíva pri spracovaní modelov scény externé aplikácie, ktorých implementácia bola popísaná v predchádzajúcej časti, je potrebné aby bolo možné nastaviť cestu k spustiteľným súborom. Túto možnosť som niekoľkokrát využil pri porovnávaní rôznych implementácií a hodnotení ich výkonnosti.

6.4.4 Nastavenie pripojenia k publikačnému serveru

Pokiaľ by došlo k zmene umiestnenia servera, ktorý sa stará o publikáciu a administráciu modelov a galérií, je možné v ponuke nastavení toto URL zmeniť. V menu pre nastavenie je tiež k dispozícii položka ktorá otestuje spojenie s vybraným serverom. V niektorých prípadoch je možné týmto spôsobom odhaliť príčinu zlyhania spojenia, pretože pri vzniknutej výnimke je užívateľ informovaný pomocou vyskakovacieho okna tzv. „MessageBoxu“.

6.4.5 Nastavenia proxy servera

Spojenie s publikačným systémom zabezpečuje .NET trieda z balíka System.Net WebClient, ktorá obsahuje množstvo rozširujúcich nastavení vytváraného HTTP spojenia. Túto možnosť som do tejto časti pridal práve preto, lebo neprítomnosť nastavenia proxyservera mi mnohokrát spôsobovala nefunkčnosť častí aplikácie.

6.4.6 Importovanie modelov

Pri importe modelov nedochádza k uložení modelov do triedy zapuzdrujúcej celý projekt, ale iba získaní informácií o ceste, počtu elementov modelu a vygenerovaní náhľadov pre ďalšie použitie v publikačnej časti.

Aplikácia umožňuje import súborov typu STL a TRI. Formáty typu VRML nie sú podporované, pretože tieto súbory vo svojej textovej podobe vyžadujú rozsiahle validácie a obsahujú z nášho hľadiska veľa redundantných informácií. VRML neslúži totiž iba pre popis geometrie objektu, ako je tomu u STL ale aj k popisu textúr a animácií.

Aplikácia ale umožňuje export súborov VRML do vybraného umiestnenia pomocou kontextového menu, ktoré je prístupné po kliknutí pravého tlačítka na vybraný importovaný model.

6.4.7 Zobrazenie modelu pomocou Direct3D

K renderovaniu a interaktívnej práci pri prezeraní modelu je použitá knižnica DirectX. Samotné vykresľovanie prebieha tak, že sa postupne prechádza kontajner trojuholníkov (MCTriS) a naplňa dva buffery. V jednom prípade sa jedná o buffer zahŕňajúci vrcholy (tzv. vertex buffer) v druhom prípade sa jedná o buffer zahŕňajúci indexy jednotlivých vrcholov (aby bolo možné určiť, ktorému trojuholníku patrí vybraný vrchol).

Zobrazenie je implementované pomocou rozhrania usercontrol. Tým je jednoducho umožnená interakcia s užívateľom pomocou myši. Z dôvodu, že Direct 3D a jej vývojový kit nie je štandardne obsiahnutý v súbore balíkov .NET, je potrebné, aby pri kompilácii boli k dispozícii potrebné knižnice. Dll súbory sú obsiahnuté v tzv. Direct X SDK. Tento kit ale nie je potrebný úplne celý, ale postačujú iba vybrané súbory, ktoré sú obsiahnuté na priloženom CD (v archíve obsahujúci spustiteľný projekt). S projektom je teda k dispozícii potrebná knižnica, a po kompilácii, v prípade spustenia na inom PC sa ku kompilátu automaticky pridáva.

6.4.8 Generovanie náhľadov

Náhľady, ktoré sú určené aj pre publikáciu sú vytvorené nasledujúcim spôsobom. Postupne prechádzam kontajner trojuholníkov a po prepočte pridávam do predpripravenej bitmapy. Pixely sú vysvecované na základe vzdialenosti. Pri počítaní viem maximálnu aj minimálnu z-ovú vzdialenosť modelu, preto nie je problém farbu pixelu normalizovať do rozsahu 0-255.

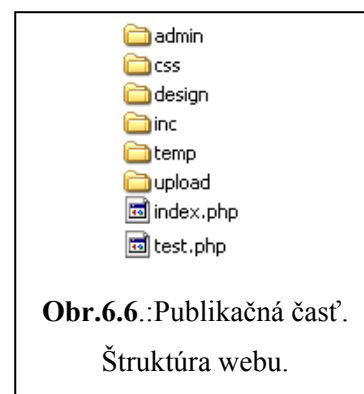
6.5 Popis publikačnej časti

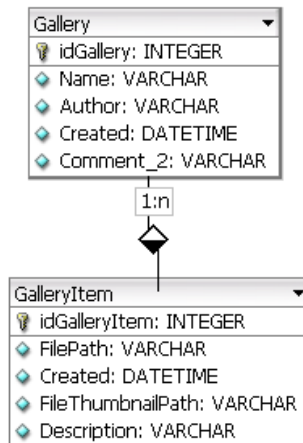
Časť, ktorá sa stará o publikovanie vybraných modelov prostredníctvom aplikácie popísanej v predošlej časti je implementovaná pomocou jazyka PHP. Táto časť, ktorá je umiestnená na danom serveri (v tomto prípade som použil školský – stud.fit.vutbr.cz), komunikuje na porte 80(HTTP) s aplikáciou pomocou jednoduchých POST dotazov. Informácie o publikovaných modeloch, galériách a pod. sú ukladané pomocou databázy My SQL, ktorá je na našej fakulte voľne dostupná všetkým študentom.

Aby nedochádzalo k výraznej záťaži databázového systému, je potrebné aby modely, ktoré v niektorých prípadoch môžu mať aj niekoľko megabajtov boli uložené iným spôsobom. Preto sa tieto modely a ich náhľady ukladajú v adresári, ktorý je na to určený a databáza je iba pridaný záznam o umiestnení súbora.

Aplikácia, ktorá je umiestnená na webovom serveri, je logicky rozdelená do jednotlivých častí, preto by som uviedol význam a funkciu jednotlivých adresárov a súborov:

- **admin** – súbory zahŕňajúce funkcionality pre vytváranie galérií, mazanie galérií a publikáciu modelov. Volané z aplikácie pre správu modelov uvedenom v predchádzajúcom bode.
- **css** – definície kaskádových štýlov. Z hľadiska implementácie nepodstatné.
- **design** – grafika webovej stránky (obrázky, ikony...) Z hľadiska implementácie taktiež nepodstatné.
- **inc** – súbory, ktoré sa starajú o zobrazenie jednotlivých sekcií webovej aplikácie.
- **temp** – adresár pre dočasné ukladanie súborov.
- **upload** – v tomto adresári sa ukladajú náhľady a jednotlivé modely, ktoré sú výsledkom publikácie. Pre správnu funkciu je potrebné aby skripty, ktoré tento adresár používajú mali nastavené práva pre zápis.
- **index.php** – hlavný súbor webovej časti, do ktorého sa podľa užívateľom vybranej stránky vkladajú pomocou php príkazu include stránky z adresára „inc“. Predávanie parametrov s informáciou, ktorá stránka sa má zobraziť je vyriešené pomocou GET, resp. POST v prípade odosielania formulárov.
- **test.php** – súbor, ktorý vráti reťazec „OK“. Aplikácia pre správu modelov ho využíva k otestovaniu funkčnosti pripojenia.





Obr.:6.7.: Databázový model použitý pre reprezentáciu galérií a ich súborov (vytvorené pomocou pomôcky DBDesigner 4).

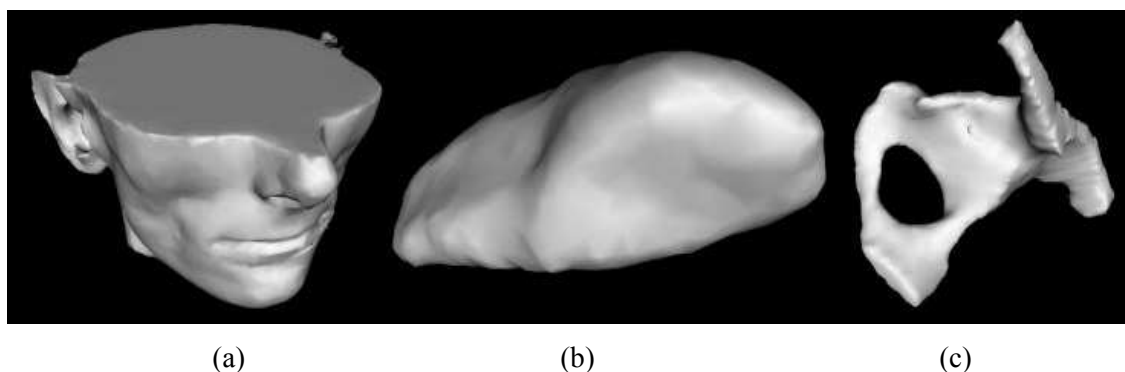


Obr.:6.8.:Publikačná časť. Na obrázku vidíme zoznam modelov v galérii. Náhl'ady je možné zvolit' ľubovoľne zo súboru. Napravo v menu je možné vidiet' ďalšie možnosti.

7 Namerané výsledky a vlastnosti aplikácie

Pre testovanie algoritmov pre úpravu polygonálnej siete by bolo najlepšie použiť modely, ktoré boli výsledkom skenovania. Keďže som nemal prostriedky (stratený softvér) k tomu, aby som použil výstupné CAM súbory k rekonštrukcii, použil som buď už hotové geometrické modely vo formáte TRI (obr. 7.1), alebo som si vygeneroval pomocou aplikácie Rhinoceros súbory STL (viď príloha B.1). Všetky súbory, ktoré slúžili ako zdroj dát, ako tiež výsledky aplikácie algoritmov, sú uvedené na priloženom disku (v prílohe C).

Keďže implementácia algoritmov pre spracovanie polygonálnych modelov je implementovaná v jazyku C++, naskytla sa mi možnosť porovnať výkon so svojou predchádzajúcou implementáciou v jazyku C#, ktorá bola predmetom mojej predošlej bakalárskej práce, na ktoré táto práca istým spôsobom nadväzuje.



Obrázok 7.1: Modely použité pre testovanie algoritmov a)face.tri b)zub.tri c)lopata.tri

Názov Modelu	Počet vrcholov	Počet trojuholníkov	Počet hrán
face.tri	4995	9998	14997
zub.tri	1001	1998	2997
lopata.tri	8333	16670	25003

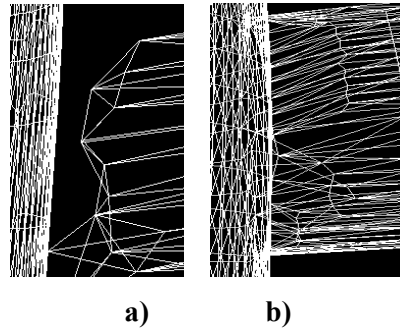
Tabuľka 1: Detaily spracovávaných modelov.

7.1 Vyhladzovanie polygonálnej siete

Tento algoritmus, ktorý je z danej skupiny implementovaných algoritmov najjednoduchší, sa javí ako najrýchlejší. Výsledky, ktoré sa po aplikácii algoritmu na objekte javili ako veľmi uspokojujúce som dosiahol už v niekoľkých prvých iteráciách. V prípade, kde som opakoval pokus, a počet iterácií som postupne zvyšoval, som zistil chybu, ktorá vyplýva z podstaty algoritmu. Jedná sa o zmenšenie

celého objektu. Tento problém nie je skoro vôbec badateľný, pokiaľ vyhladzujeme práve jeden objekt.

Pokiaľ ale vyhladzujeme model(napr. lietadlo), ktorý sa skladá s viacerých objektov(trup, krídla...) je toto zmenšenie viditeľné a nežiaduce(vid' obr.7.2.).



Obr.7.2.:Chyba - oddelenie vzniknuté pri vyhladzovaní nekompaktnej siete(a), vyhladenie správne vytvoreného modelu, kde vrcholy na seba nadväzujú(b)

V nasledujúcej tabuľke 2 sa nachádza využitie času procesora algoritmom postupne, pri zvyšovaní počtu iterácií. Zmena váhového faktoru na rýchlosť nemala žiadny vplyv. Z tabuľky je vidieť lineárnu časovú zložitosť.

Počet iterácií	zub.tri(C++)	zub.tri(C#)
10	220ms	6124ms
20	461ms	12221ms
30	701ms	18514ms
40	921ms	24652ms
50	1001ms	30515ms

Tabuľka 2: Výsledky získané vyhladzovaním objektov pomocou Laplaceovho operátora, porovnanie implementácii C# a C++.

Ďalšie demonštračné ukážky modelov, z ktorých je omnoho ľahšie poznať výsledky sa nachádzajú v prílohe B(vid' obr. B.2,B.3).

7.2 Decimácia polygonálnej siete

V nasledujúcej tabuľke sú uvedené namerané časy pri decimácii polygonálneho modelu na výslednú úroveň. Doba potrebná k ohodnoteniu siete ostáva rovnaká. Dobu potrebnú k načítaniu modelu boli v rámci niekoľkých milisekúnd, preto tento údaj neuvádzam. Časy, ktoré sú uvedené pri odstraňovaní sú závisle na počte odobratých vrcholov. Uvedené sú dve tabuľky a graf ktorý porovnáva dve rozdielne implementácie v rôznych jazykoch.

Redukcia vrcholov (%)	Zub.tri			Face.tri			Lopata.tri		
	Počet vrcholov	Čas ohodnotenia	Čas redukcie	Počet vrcholov	Čas ohodnotenia	Čas redukcie	Počet vrcholov	Čas ohodnotenia	Čas redukcie
0	1001	-	-	4995	-	-	8333	-	-
20	800	931 ms	2s 253ms	3995	26s 541ms	1m 53s	6667	1m 24s	5m 14s
40	600		3s 825ms	2995		3m 12s	5001		8m 12s
60	400		4s 746ms	1995		4m 3s	3335		11m 25s
80	200		5s 375ms	995		4m 45s	1679		14m 37s
90	100		5s 547ms	500		5m 12s	837		15m 44s

Tabuľka 3: Výsledky získané decimáciou polygonálnych modelov pomocou Garlandovho algoritmu kontrakcie hrán(C#).

Redukcia vrcholov (%)	Zub.tri			Face.tri			Lopata.tri		
	Počet vrcholov	Čas ohodnotenia	Čas redukcie	Počet vrcholov	Čas ohodnotenia	Čas redukcie	Počet vrcholov	Čas ohodnotenia	Čas redukcie
0	1001	-	-	4995	-	-	8333	-	-
20	800	1s 241 ms	761ms	3995	6s 419ms	3s375ms	6667	11s 166 ms	6s 900ms
40	600		1s 503ms	2995		7s611ms	5001		12s 902ms
60	400		2s 254ms	1995		11s677ms	3335		18s 707ms
80	200		3s 045ms	995		15s612ms	1679		25s 296ms
90	100		3s 375ms	500		17s225ms	837		29s 92ms

Tabuľka 4: Výsledky získané decimáciou polygonálnych modelov pomocou Garlandovho algoritmu kontrakcie hrán(C++).

7.3 Zhodnotenie nameraných výsledkov

Algoritmy, ktoré boli výsledkom tejto práce, boli postavené na platforme C++. K tomu aby bolo možné naimplementovať algoritmy pre úpravy polygonálnej siete, bolo potrebné obstarat' knižnicu VectorEntity, ktorá je momentálne súčasťou Medical Data Segmentation Toolkitu (MDSTK).

Aj keď som oproti pôvodnej implementácii v jazyku C# dosiahol niekoľkonásobné zvýšenie rýchlosti algoritmom. Jednalo sa hlavne o odstránenie nežiaducich cyklických závislostí garbage collectoru. Odstránenie cyklických závislostí je vidieť najlepšie vidieť na modeloch s väčším počtom polygónov(vid' tabuľka 3, tabuľka 4). Porovnanie výkonnosti C# a C++ je možné hlavne vďaka tomu, že výsledky platformy C# boli získané pri tvorbe mojej bakalárskej práce, na ktorú táto práca istým spôsobom nadväzuje.

Ďalšie zvýšenie výkonu mojej implementácie v C++, by mohlo byť použitím knižnice SmallObjectLoki. Táto knižnica obsahuje sadu tried, ktoré sú využívané pri alokácii veľkého množstva malých objektov, na ktorých je polygonálna sieť postavená. Elementy polygonálnej siete zaberajú najväčšiu časť pamäte a práve táto sada tried sa stará, že nedochádza k alokácii nadbytočného množstva pamäti. Týmto by bola urýchlená časť, s ktorou sa najviac pracuje – vytváranie o odstraňovanie objektov ako sú vrcholy, trojuholníky, hrany, atď.

Toto zvýšenie by mohlo byť užitočné hlavné pri spracovaní rozsiahlych objektov. V mojom prípade som chcel mať možnosť porovnania a aj tak som mal k dispozícii iba reálne objekty, ktorých veľkosť bola maximálne 10000 vrcholov. Sadu tried SmallObjectLoki som teda musel zo zdrojových kódov „manuálne“ odstrániť, pretože je štandardnou súčasťou MDSTK.

Čo sa týka aplikácie pre správu modelov, je postavená naproti algoritmom na jazyku C#, platforme .NET. V krátkosti môžeme povedať, že sa jedná o platformu, nad ktorou sa spúšťajú programy napísané nielen v jazyku C#, ale i Java, C++, Visual Basic, Fortran a množstvo iných jazykov. .NET Framework je možné prirovnať k Java Runtime, ale samozrejme existuje mnoho väčších, či menších rozdielov.

Podobný je napríklad prístup k správe pamäti, kde sa rovnako ako v jazyku Java používa algoritmus Garbage Collection. Tento algoritmus patrí do veľkej množiny funkcií a služieb, ktoré táto platforma poskytuje. Garbage Collector sa stará o uvoľňovanie nepotrebných programových objektov z pamäti. Vďaka tomuto spôsobu sa vývojár nemusí starať o priradzovanie alebo uvoľňovanie operačnej pamäti a odpadá tak riziko nekorektnej práce s ňou, ktorá vo väčšine situácií končí pádom aplikácie.

Tento fakt je veľmi výhodný pri programovaní napr. formulárových aplikácií, kde počet použitých objektov nedosahuje také veľké množstvo ako pri použití v počítačovej grafike, a tým pádom nedochádza k uvoľňovaniu pamäti tak často.

Vďaka tomu, že aplikácia využíva, pre načítanie informácií o modeloch prepracovanú knižnicu v jazyku C#, dochádza pri načítavaní modelov k určitému spomaleniu. Toto spomalenie pre modely, ktoré som mal k dispozícii nebolo príliš veľké (do 2 s) .

Problém rýchlosti načítania modelov, je možné vyriešiť použitím generických kolekcii Frameworku verzie 2.0. resp. najnovšej 3.0.

7.4 Použité prostriedky

Implementácia, ktorej výsledky sú uvedené v predošlých častiach som testoval na osobnom počítači. Následne sú uvedené komponenty, ktoré mali na rýchlosť výslednej implementácie najväčší vplyv:

- Procesor Pentium Centrino 1,8 GHz,
- Grafický chipset Intel Extreme Graphics 82852 GM/GME
- Pamäť 1GB
- Operačný systém MS Windows XP Professional SP2
- Použité prostredie Visual Studio .NET 2005, Framework 2.0
- Microsoft DirectX SDK (august 2007)

8 Záver

Pre splnenie tejto práce bolo nutné naštudovať literatúru na téma snímania predmetov, ako tiež zoznámiť sa s existujúcim snímačom umiestnenom na našej fakulte. Z tohto hľadiska boli v tejto práci čitateľovi uvedené jednotlivé metódy digitalizácie, ich výhody, nevýhody, použitie s praktického hľadiska a princípy merania vzdialenosti.

Boli tu uvedené praktické skúsenosti zo skenerom, jeho vlastnosti a výstupy demonštrované vhodnými ukázkami. K skeneru bola tiež vyčlenená samostatná kapitola, ktorá pojednáva o praktických problémoch pri skenovaní, ktoré ma v prípade tak sofistikovaného zariadenia dosť prekvapili. V práci som taktiež popísal algoritmy, ktoré sú využívané pri procese spracovania modelov scény, popis rekonštrukcie objektov, ďalej popis užívateľského rozhrania pre prezeranie a publikovanie skenovaných modelov.

Publikačnému rozhraniu, ktorá bola jedna podstatných častí diplomovej práce bol vyčlenený priestor, v ktorom boli vysvetlené jej hlavné myšlienky implementácie.

Súčasťou prílohy sú demonštračné ukážky jednotlivých algoritmov, ako tiež užívateľská príručka implementovaných programov.

Pre demonštráciu publikačného rozhrania a spracovania modelov nebolo možné použiť priamo výstupy skeneru, tak ako to bolo pôvodne plánované. Dôvodom toho bolo, že softvér, ktorý sa stará o rekonštrukciu polygonálneho modelu nebol na fakulte k dispozícii. K dispozícii bola iba demonštračná verzia softvéru, ktorý síce umožňoval vytvorenie modelu, ale neumožňoval jeho uloženie. Preto som na ukážky použil už existujúce modely, alebo som si ich vytvoril sám pomocou modelovacieho softvéru.

Z tohto hľadiska by som poznamenal, že práca mala v istých fázach vďaka stratenému softvéru veľmi investigatívny charakter. Posledný známi užívateľia skeneru, ktorý pracovali so spomenutým zariadením už neboli v dobe písania tejto diplomovej práce zamestnancami fakulty. Po spolupráci s p.Zemčíkom sa podarilo získať na tieto osoby kontakt a zistiť niektoré informácie, ktoré aj po dôkladnom prehľadani fakultného skladu nevedli k nájdeniu vyššie spomínaného softvéru.

Keďže licencia softvéru, ako tiež kompaktné disky s aplikáciami pre rekonštrukciu výškových máp neboli na fakulte tým pádom k dispozícii, bolo nutné nadviazať kontakt so zástupcami výrobcu. Nanešťastie komunikácia so zástupcom bola veľmi zdĺhavá a až približne po cca 5-6 mailoch som dostal z českého zastupiteľstva nejakú odpoveď.

Vzhľadom k tomu, že fakulta má záujem do strateného softvéru investovať(jeho hodnota je v porovnaní s cenou skeneru – cca 1,3 milióna CZK zanedbateľná) všetky potrebné údaje, týkajúce sa objednávky tohto špecifického problému boli predané p. Zemčíkovy. Z tohto hľadiska by mohol byť skener plne využiteľný už v nasledujúcom semestri. Bez tohto softvéru sa totiž jedná o prakticky nepoužiteľné zariadenie.

Z hľadiska ďalšieho vývoja tohto projektu by bolo možno pokračovať v dvoch rozdielnych smeroch. V prvom prípade by bolo možné rozšíriť aktuálnu implementáciu o ďalšie funkcie a algoritmy. Táto časť by sa mohla týkať všetkých troch častí - aplikácie pre správu modelov, publikačnej časti ako aj pridaním ďalších algoritmov. V prípade aplikácie správy by bolo možné rozšíriť spôsob komunikácie na zabezpečenú úroveň, pridať rozlíšenie užívateľských úloh, alebo rozšíriť modely a databázu o ďalšie informácie, ktoré by bolo potreba. Túto časť by som považoval za najviac rozšíriteľnú, pretože závisí práve od cieľového užívateľa, aké chovanie programu by potreboval.

V druhom prípade rozširovania projektu, by mohlo byť vytvorenie jednotného užívateľského rozhrania, ktoré by umožňovalo prehliadanie, publikáciu a interaktívne aplikovanie algoritmov pre úpravu polygonálnych modelov. Pokiaľ by sme ostali pri implementácii v jazyku C# bolo by potrebné optimalizovať algoritmy a knižnicu VectorEntity do tej úrovne, aby sa výkonnosť tejto platformy priblížila implementácii C++. Toto by bolo možné uskutočniť buď použitím generických kolekcí frameworku 2.0, alebo naprogramovaním VectorEntity pomocou „unsafe“ blokov. Toto by prinášalo odstránenie výhod resp. nevýhod garbage collectoru. Preprogramovaním by sme ale prišli o možnosť zrovnania výkonu platforiem.

Literatúra

- [1] TIŠNOVSKÝ, Pavel. Bezkontaktní digitalizace předmětů pomocí 3D scanneru Minolta Vivid VI-700. *Elektrorevue*, 2003, roč. 2003, č. 13, s. 1-25, ISSN 1213-1539.
- [2] Žara J., Beneš B., Felkel P.: Moderní počítačová grafika. 1. vyd. Praha, Computer press 1998, 448 s., ISBN 80-7226-049-9
- [3] Robinson S., Cornes O. a kol.: C# Programujeme profesionálně. 1. vyd. Praha, Computer press 2003., ISBN 80-251-0085-5
- [4] Luebke D., Reddy M.: Level of detail for 3D graphics. San Francisco, Morgan Kaufmann Publishers 2003, ISBN 1-55860-838-9
- [5] Garland M.: Quadric-based polygonal surface simplification, Ph.D. Thesis, Technical Report CMU-CS-99-105, Computer Science Department, Carnegie Mellon University, 1999, Dokument dostupný na URL: <http://graphics.cs.uiuc.edu/~garland/research/thesis.html>
- [6] Inspetion services, 3d measurements specialist, Dokument dostupný na URL: www.gks.com
- [7] VOJÁČEK, Antonín. *PMD senzor & 3D měření vzdálenosti - 1. část "Princip"* [online]. 2006-07-03 [cit. 2007-04-05]. <<http://automatizace.hw.cz/view.php?cisloclanku=2006070301>>.
- [8] 3D Grafika, Mechanické skenery řady Microscribe, Dokument dostupný na URL: www.3dgrafika.cz
- [9] 3D X-Ray Microscopic CT Scanner, Dokument dostupný na URL: <http://www.yamato-net.co.jp/english/products/bio/tdm1000/tdm1000.htm>
- [10] Cortech Solutions, Dokument dostupný na URL: <http://cortechsolutions.com>
- [11] 3D Digitalizace, Dokument dostupný na URL: <http://robo.hyperlink.cz/re-teorie/main04.html>
- [12] Riegl USA, Long range laser scanning, Dokument dostupný na URL: <http://www.riegl.com>
- [13] CGI, Capture Geometric Internally, Dokument dostupný na URL: http://www.cgiinspection.com/products_css-3000.cfm

Príloha A

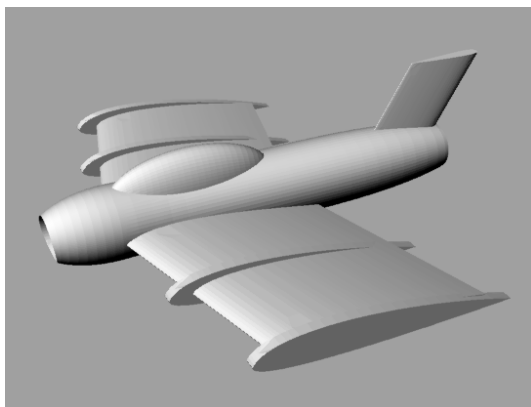
Technické parametre skeneru Minolta VIVID-700

Vzdialenosť objektívu	0.6-2.5 metra
Zorné pole	70x70 - 1100x1100 mm
Rozlíšenie v osách XY	0.35-5.5 mm
Rozlíšenie v ose Z	0.11-7.23 mm
Veľkosť textúry	400x400 pixlov
Formát pixlu	RGB
Veľkosť geometrie	200x200 geometrických bodov
Maximálny počet bodov	40 000 bodov
Rýchlosť snímania	0,6 sec
Rýchlosť prenosu	2,0 sec
Výmenný objektív	nie
Váha	9 kg
Rozmery skenera	210x367x326 mm

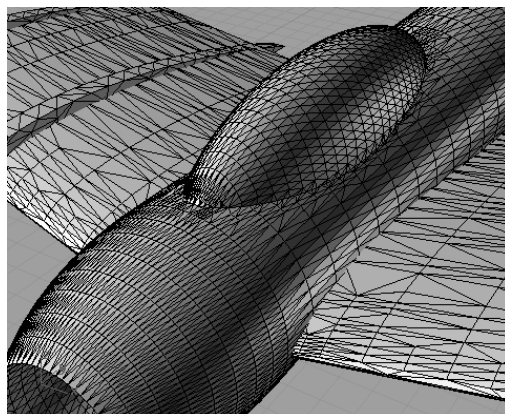


Obr.A.1.:Skener Minolta VIVID-700 so stanicou Silicon Graphics.[1]

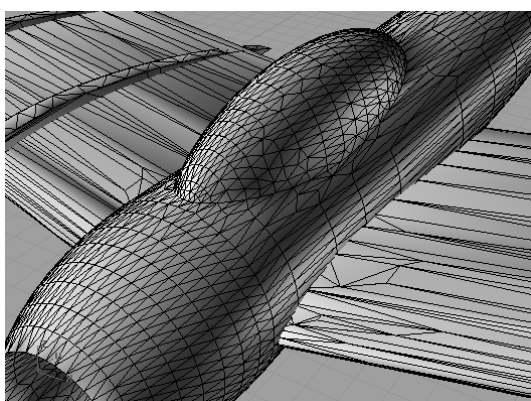
Príloha B



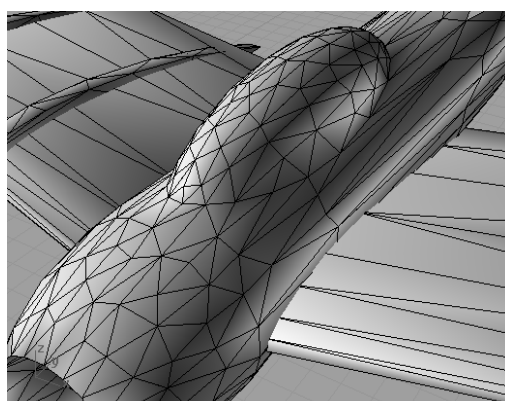
a)



b)

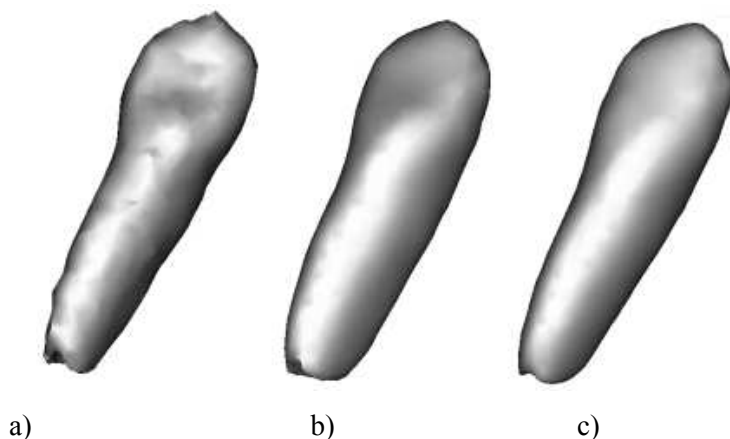


c)

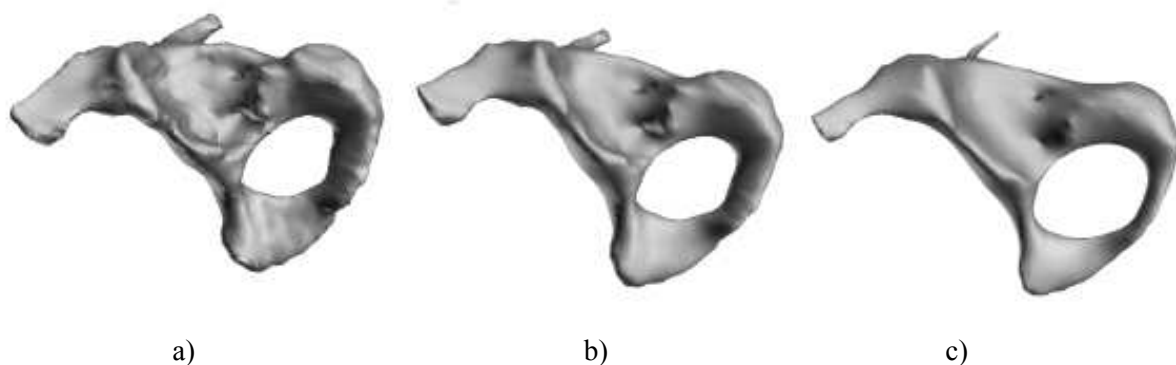


d)

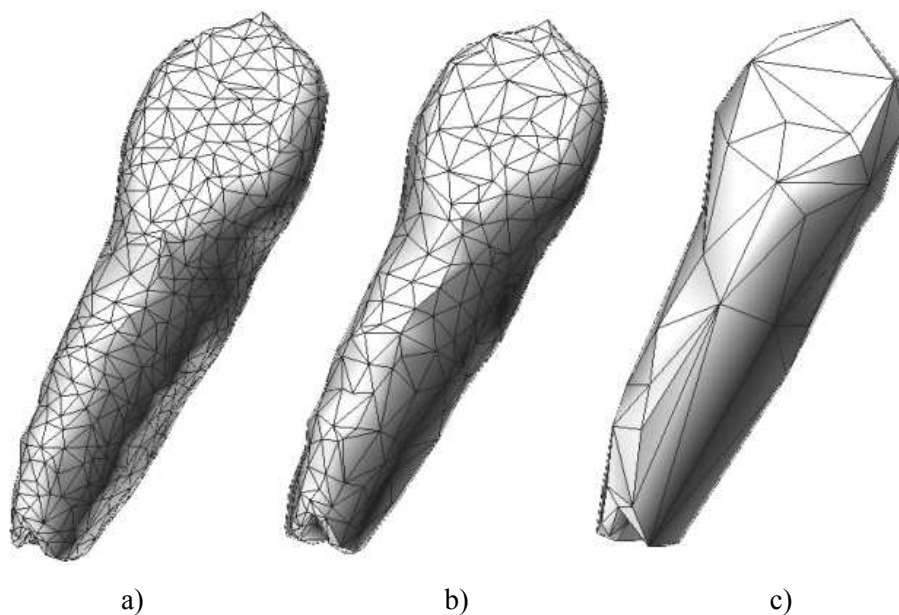
Obr.B.1.: Aplikácia decimácie na polygonálny model obsahujúci 14231 vrcholov(a), 10104 vrcholov(b), 6 437 vrcholov(c), 1260 vrcholov(d). Náhľady boli vyrenderované pomocou trial verzie prehliadača ModelPress Reader. Pôvodný model bol vytvorený v programe Rhinoceros a vyexportovaný do STL.



Obrázok B.2: aplikovanie vyhladzovania na model zub.tri. (a)Pôvodný model, (b)Model vyhladený pomocou Laplaceova operátora, váhový faktor 0.5,počet iterácií 5, (c) Model vyhladený pomocou Laplaceova operátora váhový faktor 0.5,počet iterácií 10.



Obrázok B.3: Aplikovanie vyhladzovania na model lopata.tri. (a)Pôvodný model, (b)Model vyhladený pomocou Laplaceova operátora, váhový faktor 0.5,počet iterácií 10, (c) Model vyhladený pomocou Laplaceova operátora váhový faktor 0.9,počet iterácií 15



Obrázok B.4: Garlandov algoritmus pre decimáciu aplikovaný na model zub.tri. (a)Pôvodný model,1001 vrcholov,(b)500 vrcholov(c) 100 vrcholov

Príloha C

Treťou časťou prílohy je priložené CD, ktoré obsahuje:

- Navigáciu obsahu CD vo formáte *HMTL*.
- Zdrojové kódy
- Príklady generovaných objektov vo formáte *TRI*, *STL*, *VRML* a *JPG*.
- Technickú správu vo formáte *PDF* a jej zdrojový tvar v *MS WORD*.
- Programovú dokumentáciu projektu a diagram použitých tried.

Na priloženom disku je umiestnená html navigácia (`$ROOT/readme.html`), ktorá umožňuje prehľadné listovanie v jednotlivých kategóriách ukážok výsledkov algoritmov, implementácií, dokumentácie apod.

Na disku sú k dispozícii zdrojové súbory a kompiláty týchto aplikácií:

1. Implementácia algoritmov pre úpravu polygonálnej siete.
2. Užívateľské rozhranie pre správu a publikáciu vybraných modelov.
3. Webová časť pre zobrazenie a administráciu publikovaných modelov.

V prvom prípade sa jedná sa o konzolovú aplikáciu, ktorá po spustení žiada užívateľa o výber algoritmu pomocou parametra. Aplikácia ďalej požaduje po užívateľovi vstupný súbor (*TRI*, *STL*) pre spracovanie a výstup v internom formáte *TRI*. S konzolovou aplikáciou ale užívateľ nepríde do styku, pretože generovanie parametrov je automatické na základe informácií získaných s druhého „okenného rozhrania.“

V druhom prípade je k dispozícii rozhranie umožňujúce komplexnú správu polygonálnych modelov, ich konverzie, import, generovanie náhľadov a ich publikáciu. Táto aplikácia volá externé spustiteľné súbory implementujúce algoritmy.

K spustení už kompilovaného projektu bez prítomnosti inštalovaného Visual Studia, je nutné mať nainštalovaný .NET Framework 2.0, ktorý je štandardne obsiahnutý už v MS Windows XP SP2.

Aplikácie sú na disku umiestnené vo forme projektu ako aj výsledného kompilátu (Výnimkou je bod 3, ktoré obsahujú iba php – zdrojové súbory)

Príloha D

Predmetom tejto prílohy bude popis užívateľského rozhrania s pohľadu ovládania(nie funkcionality) tzv. užívateľská príručka. Tejto časti budú vysvetlené postupy ovládania programu sprevádzané ilustračnými ukážkami.

D.1.Ovládanie publikačného rozhrania

Modely, ktoré boli určené na publikáciu sú k dispozícii prostredníctvom vygenerovaných www stránok. Po otvorení adresy(pre účely diplomovej práce www.stud.fit.vutbr.cz/~xzdrah04/DIP) sa nám zobrazí úvodná stránka podobná obrázku D.1. Táto stránka obsahuje základné informácie o aplikácii a pre zjednodušenie – tiež čiastočné vysvetlenie funkcionality. Ďalej je možné pokračovať výberom z užívateľského menu v pravej časti obrazovky.



Obr.D.1.:Otvorenie publikačnej časti prostredníctvom internetového prehliadača (Firefox).

D.1.1.Užívateľské menu

Užívateľ môže pokračovať výberom z menu. Toto menu sa mení iba v prípade prihlásenia užívateľa. Pod ponukou je umiestnený ikona, ktorá overuje validitu stránok(nemá vplyv na funkcionality). Jednotlivé položky ponuky budú vysvetlené v nasledujúcom texte.



Obr.D.2.:Užívateľské menu po prihlásení(vľavo) a normálne(vpravo)

D.1.2.Prehľady galérii

Aplikácia umožňuje listovať v galériách, ktoré sú zoradené podľa času vytvorenia. U každej galérie je vidieť popis, autora, čas vytvorenia a počet objektov. Obrázky, ktorý je uvedený v pravej časti(Obr.D.3.), patria vždy prvému modelu danej galérie.



Obr.D.3.:Obrazovka so zoznamom galérií.

D.1.3.Zobrazenie položiek galérie

Pri prehliadaní galérií(ak galéria má nejaké modely) užívateľ môže kliknúť na ikonu zobrazenia galérie(Obr.D.4.)



Obr.D.4.:Ikona pre zobrazenie modelov galérie.

D.1.4. Nápoveda

Táto časť je iba informačná, a neobsahuje užívateľskú príručku. V tejto časti je k dispozícii skrátený zoznam funkcionality a referencia na plugin pre prehliadanie VRML a STL modelov.

D.1.5.Prihlásenie administrátora

Publikačná časť umožňuje základné možnosti administrácie galérií a modelov. K tomu aby bolo možné administráciu vykonávať, je potrebné sa prihlásiť. S dočasných (z dôvodu diplomového projektu) je užívateľské meno a heslo dočasne predvyplnené(DIP,DIP).

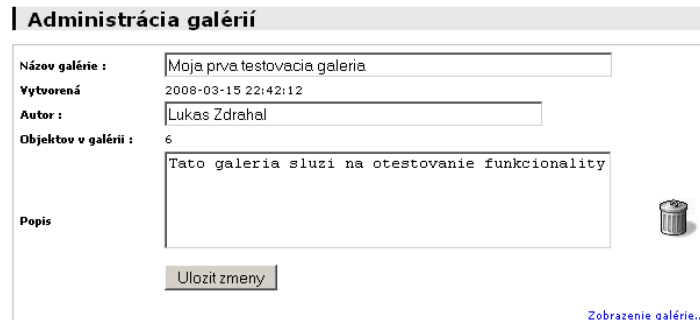
Prihlasovacie meno	<input type="text" value="DIP"/>
Heslo *)	<input type="text" value="DIP"/>
<input type="button" value="Ok"/>	

*)Prihlasovacie meno a heslo je dočasne predvyplnené.

Obr.D.5.:Obrazovka s prihlasovacím dialógom.

D.1.6.Správa galérií

Po úspešnom prihlásení je užívateľ automaticky presmerovaný na stránku obsahujúca zoznam galérii(Obr D.6.). v tejto časti má užívateľ možnosť meniť jej popis, autora a názov, ako tiež celú ju odstrániť.



Obr.D.6.:Obrazovka so zoznamom galérií.

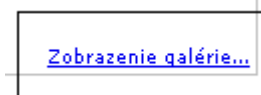
Pre zmazanie modelu, alebo celej galérie je možné kliknúť na ikonu v tvare koša(Obr.D.7)



Obr.D.7.:Ikona pre odstránenie galérie, resp. modelu.

D.1.7.Správa modelov

Po kliknutí v administrátorskej (D.8.) sa objaví stránka podobná(D.6.), ktorá sa týka administrácie už konkrétnej galérie. Ovládanie je vcelku podobné



Obr.D.8.:Ikona pre administráciu už konkrétnej galérie.

D.1.8.Štatistiky

V tejto časti sú k dispozícii informácie, o zmenách a obsadenosti galérie (počet modelov, galérií, ich celková priestorová náročnosť).

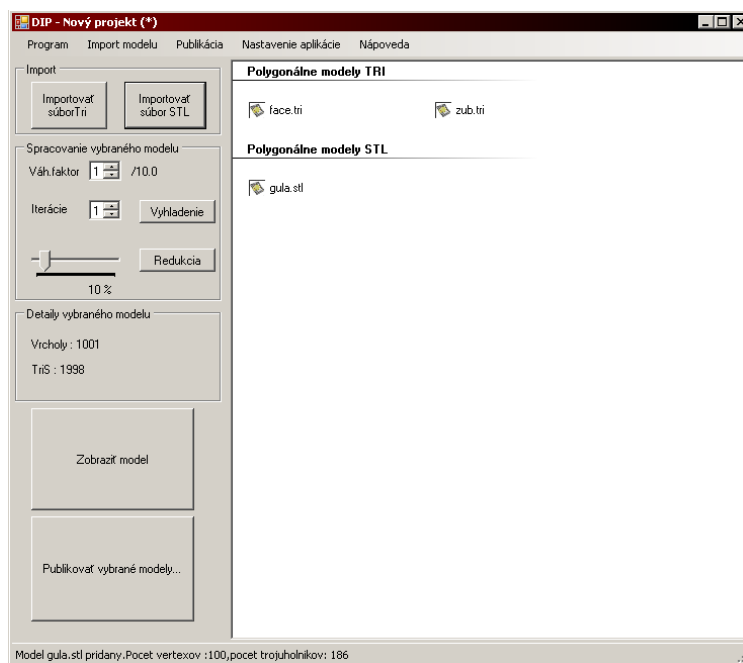
Štatistiky	
Galérie	
Celkový počet galérií	2
Celkový počet súborov v galériách	8
Počet vytvorených galérií	9
Posledná pridaná galéria	2008-04-08 12:37:49
Modely	
Celkový počet modelov	8
Počet pridaných súborov v histórii	52
Posledný pridaný súbor	2008-04-08 12:38:13
Celková veľkosť náhľadov	62256 Bajtov
Celková veľkosť súborov	150294 Bajtov

Copyright ©2008 Bc.Lukáš Zdrahal

Obr.D.9.:Stránka s informáciami o dátach galérie

D.2. Aplikácia pre správu a publikáciu modelov

Po spustení projektu, alebo už kompilovanej aplikácie sa otvorí obrazovka podobná nasledujúcemu obrázku(Obr.D.10). Okno by bolo možné rozdeliť na 4 časti: horné menu, spodný status bar, ovládacie prvky na ľavej strane a zoznam naimportovaných modelov na strane pravej.



Obr.D.10.: Aplikácia pre správu a publikáciu modelov.

D.2.1. Import polygonálnych modelov

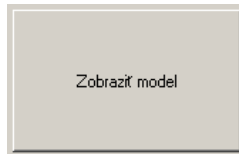
Aplikácia umožňuje import polygonálnych modelov vo formáte TRI a STL. Import je možný kliknutím na položke v hornom menu, alebo priamo z ľavej časti hlavného okna aplikácie(obr. D.11.). Po importovaní prebieha načítanie modelu a zistenie informácií ako tiež vygenerovanie náhľadu. Časová náročnosť načítania je závislá od veľkosti načítavaného modelu.



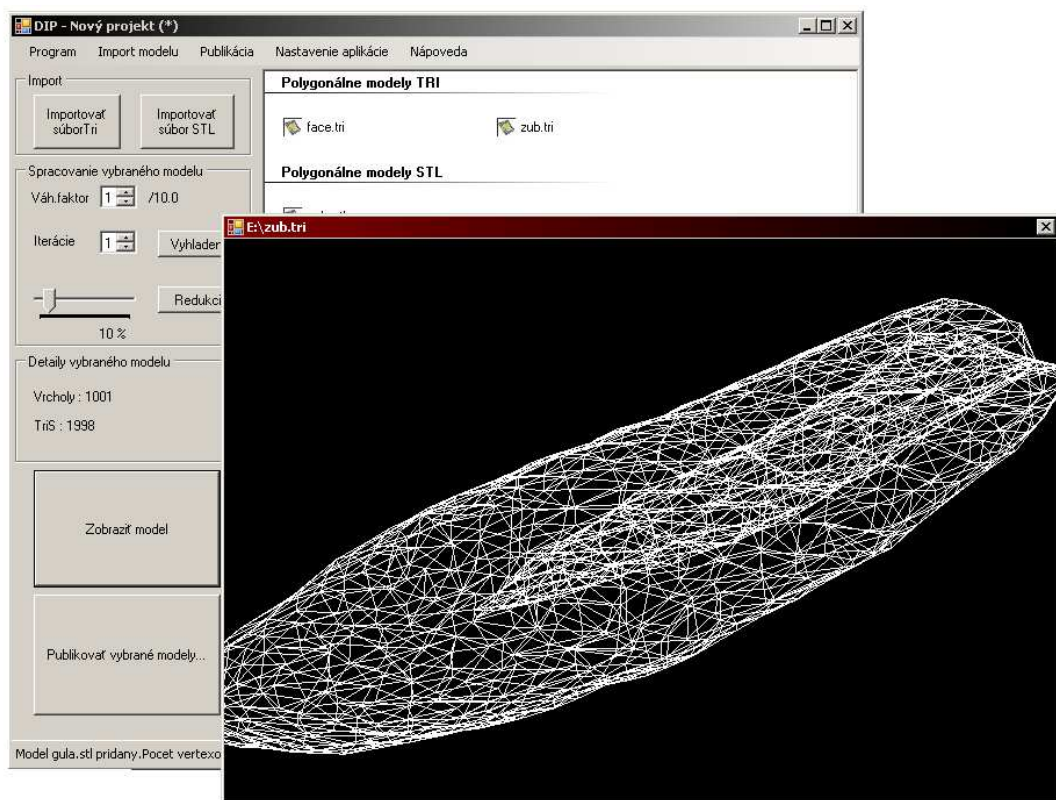
Obr.D.11.: Ikony pre import modelov z hlavného okna(vľavo) a z menu(vpravo).

D.2.2. Vizualizácia polygonálneho modelu

Zobrazenie polygonálneho modelu prostredníctvom DirectX je možné po vybratí modelu z pravej časti (zoznamu modelov) a kliknutí na ikonu „Zobraziť model“. Po otvorení vykreslovacieho okna užívateľ môže ľavým tlačítkom myši rotovať a pravým približovať model.



Obr.D.12.: Ikony pre zobrazenie modelu.



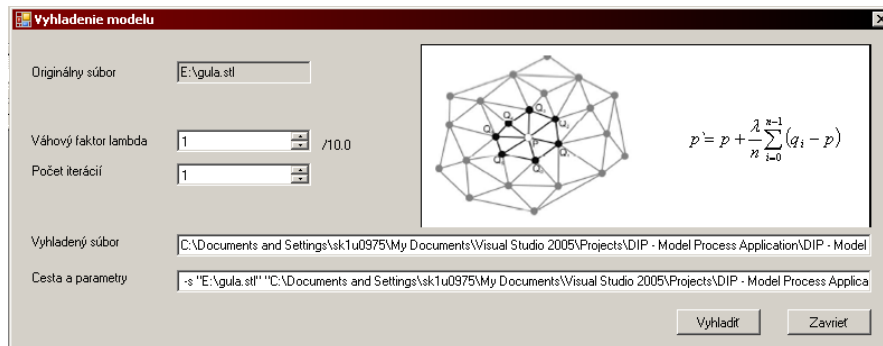
Obr.D.13.: Okno pre vizualizáciu polygonálneho modelu.

Pretože aplikácia by nemala obsahovať v pamäti naraz všetky nainportované modely, pri kliknutí na zobrazenie modelu dochádza k jeho načítaniu zo súboru, definovanou štruktúrami, ktoré pri prvotnom importe vznikajú.

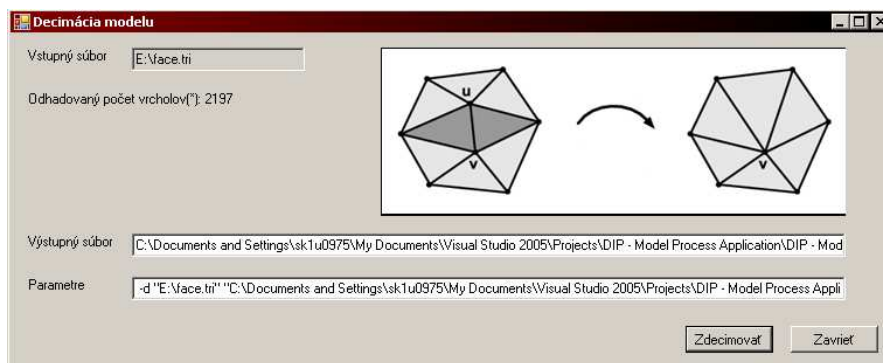
V prípade, že by súbor neexistoval (užívateľ ho odstránil, premenoval, alebo zmazal), užívateľ je informovaný o chybe hlásením v podobe messageboxu.

D.2.3. Úprava polygonálnej siete

Po vybratí polygonálneho modelu určeného k úprave (smoothing, decimácia) je zobrazené okno, ktoré umožňuje užívateľovi zmeniť nastavenia decimácie, prípadne parametrov spúšťaného externého programu. Možnosť nastavenia parametrov externej aplikácie pre úpravu siete rozširuje jej použitie o ľubovoľný externý program.



Obr.D.14.: Okno pre vyhladzovanie polygonálneho modelu.

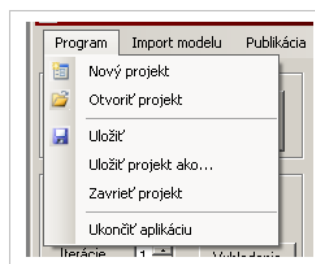


Obr.D.15.: Okno pre redukciju polygonálneho modelu..

D.2.6. Správa modelov v podobe projektu

V prípade potreby je možné uložiť aktuálny stav aplikácie do podoby projektu. Každú zmenu projektu môže vidieť tak ako je to v iných aplikáciách. V názve okna po zmene pribudne hviezdička.

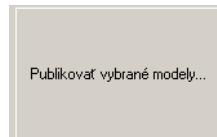
V prípade, že sa užívateľ snaží zavrieť neuložený projekt, je o tejto skutočnosti informovaný a vyzvaný vyskakovacím oknom k uloženiu. Súbory s konfiguráciou projektu uchovávajú informácie o uložení súborov, nie ich dátový obsah (modely môžu byť príliš veľké).



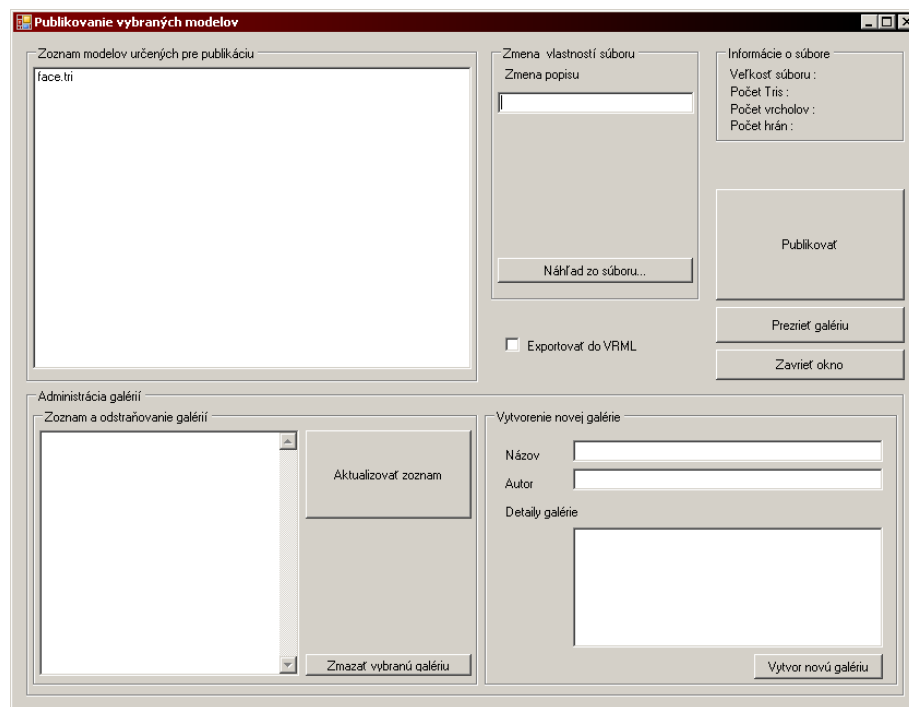
Obr.D.16.: Možnosti pre prácu s projektom.

D.2.7. Publikácia modelov

Po naimportovaní modelov je ich možné predať ďalej do publikačnej časti. Modely publikujeme tak, že ich v hlavnej aplikácii označíme a klikneme na tlačítko(Obr.D.17). V prípade, že nebudú vybrané žiadne modely, publikačné okno sa neotvorí(Obr.D.18). Pri otváraní okna je potrebné byť pripojený na internet, pretože aplikácia sa dotazuje na webový server pomocou jednoduchých HTTP dotazov. Hneď pri otvorení je automaticky získaný zoznam galérií(ekvivalentné s použitím tlačítka „Aktualizovať zoznam“).



Obr.D.17.:Tlačítko pre publikáciu modelov.



Obr.D.18.:Publikačná časť, po výbere modelov.

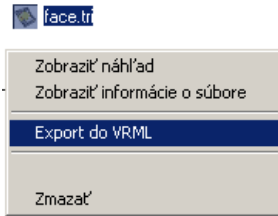
V okne máme k dispozícii rozhranie pre vytváranie a mazanie galérií, úpravy popisov, patriace k jednotlivým modelom a hromadný export do VRML(pokiaľ chceme, aby všetky modely boli v tomto formáte).

Zmeny popisiek modelu meníme tak, že v zozname modelov vyberieme položku a vpravo sa nám zobrazí aktuálna informácia, ktorú môžeme ľubovoľne zmeniť. Ukladanie prebieha automaticky. Ďalšou možnosťou je zmena náhľadu. Náhľad súboru môžeme ľubovoľne zmeniť na akýkoľvek iný, ktorý načítame zo súboru.

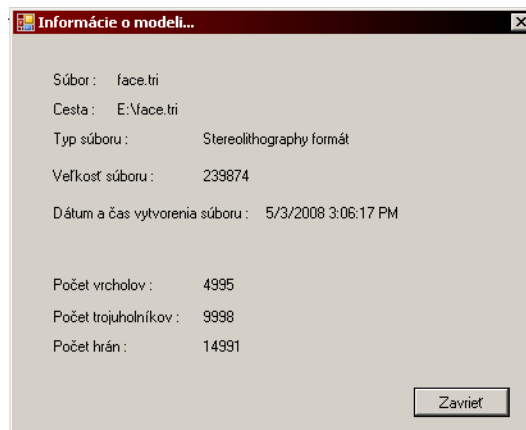
Pre prezretie galérie on-line môžeme kliknúť na ikonu „Prezrieť galériu“. Kliknutím sa nám otvorí prehliadač, s prednastavenou adresou webovej galérie(možné zmeniť v konfigurácii).

D.2.8. Kontextové menu

Aj keď detaily(počty vrcholov) sú k dispozícii v hlavnom okne, je možnosť prezrieť všetky ostatné informácie o modeli(obr. D.20). Toto menu je možné vyvolať kliknutím pravého tlačítka na naimportovaný model. V tomto menu je možné zobraziť informácie o uložení súboru, zobrazenie jeho náhľadu, exportovanie do VRML, alebo zmazanie modelu.



Obr.D.19.:Menu, ktoré sa zobrazí pri kliknutí pravým tlačítkom na model.



Obr.D.20.:Okno zobrazujúce detaily modelu.

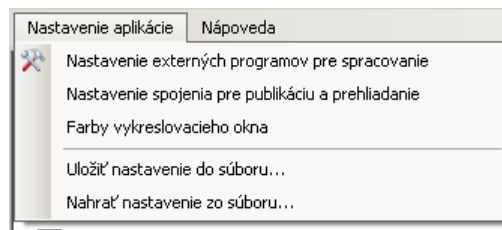


Obr.D.21.:Okno zobrazujúce náhľad generovaný na základe polygonálnej siete.

D.2.9. Nastavenia aplikácie

Pretože množstvo parametrov, s ktorými aplikácia pracuje sa môže časom meniť, je k dispozícii množstvo nastavení prístupné z horného menu(Obr.D.22.).

- **Nastavenia externých programov pre spracovanie** – Obsahuje názvy a cesty aplikácii externých aplikácii pre vyhladzovanie a decimáciu.
- **Nastavenie spojenie pre publikáciu** – Obsahuje umiestnenie galérie a nastavenie proxyservera
- **Farby vykreslovacieho okna** – Použité pri vykresľovaní(viď D.2.3.)
- **Ukladanie a načítanie nastavenia do súboru** – pre uloženie predošle uvedeného nastavenia do súboru.



Obr.D.22.:Menu s možnosťami nastavenia aplikácie.