

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019

Marek Lampáš



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## SPOJOVÁNÍ DÍLČÍCH 3D MODELŮ POVRCHU DO CELKOVÉHO MODELU

JOINING PARTIAL 3D SURFACE MODELS INTO THE OVERALL MODEL

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Marek Lampáš

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Adam Chromý, Ph.D.

BRNO 2019



# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**  
Ústav automatizace a měřicí techniky

**Student:** Marek Lampáš

**ID:** 195378

**Ročník:** 3

**Akademický rok:** 2018/19

**NÁZEV TÉMATU:**

## Spojování dílčích 3D modelů povrchu do celkového modelu

**POKYNY PRO VYPRACOVÁNÍ:**

Cílem práce je vytvořit a implementovat algoritmy pro sesazování několika dílčích skenů do jediného celku, a to na základě znalosti navzájem korespondujících bodů nebo na základě znalosti přesných absolutních souřadnic. Výsledky práce budou nasazeny v systému RoScan určeném pro medicínské skenování lidského těla.

Zadání:

1. Seznamte se s robotickým multispektrálním systémem RoScan a jeho stávající implementací.
2. Navrhněte modul pro spojování několika dílčích skenů, které mají shodný souřadný systém.
3. Navrhněte modul pro transformaci souřadných systémů na základě znalosti korespondujících bodů. Tento modul bude aplikován na skeny s rozdílnými souřadnými systémy ještě před aplikací spojovacího algoritmu.
4. Implementujte navržené moduly do systému RoScan užitím programovacího jazyka C#.
5. Realizujte experimenty, které prokáží korektní fungování implementovaných modulů.

**DOPORUČENÁ LITERATURA:**

[1]F. Yu, Z. Lu, H. Luo, and P. Wang, Three-Dimensional Model Analysis and Processing. Springer Science & Business Media, 2011.

**Termín zadání:** 4.2.2019

**Termín odevzdání:** 20.5.2019

**Vedoucí práce:** Ing. Adam Chromý, Ph.D.

**Konzultant:**

**doc. Ing. Václav Jirsík, CSc.**  
*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Táto práca sa zaoberá spájaním čiastkových 3D modelov povrchu, ktoré sa na určitej ploche prekrývajú, do celkového výsledného modelu. Cieľom je popísať a vysvetliť algoritmus, ktorý som v rámci predchádzajúcej semestrálnej práce navrhol a jeho následná implementácia v jazyku C# v systéme RoScan. Uvodná časť je venovaná krátkemu priblíženiu systému RoScan. Ďalej je podrobne popísaný princíp algoritmu, ktorý som neskôr implementoval a otestoval na rôznych druhoch vstupných dát. Pri spájaní môžu nastať dva prípady- buď čiastkovým skenom sedia absolútne súradnice, alebo nesedia a skeny sa musia najskôr dopasovať pomocou užívateľom vybraných odpovedávúcich si bodov. Táto práca sa zaoberá oboma prípadmi.

## KLÚČOVÉ SLOVÁ

Spájanie čiastkových 3D modelov, absolútne súradnice, odpovedajúce si body, transformačné matice, homogénne matice, prekrývajúce sa modely, dopasovanie 3D modelov, triangulácia povrchu modelu, pretriangulovanie povrchu

## ABSTRACT

This bachelor's thesis deals with joining partial 3D surface models, that overlap on certain area, into the overall model. The aim is to describe and explain the algorithm I proposed in the previous semestral thesis and its subsequent implementation in the C# language in RoScan system. Introductory part contains short approach of the RoScan system. Next, there is detailed description of the algorithm's principle, that I later implemented and tested on various types of input data. There are two cases that can occur during the connection- either the partial models fits in the absolute coordinates or they do not and they must be fitted first using user-selected corresponding points. This thesis deals with both cases.

## KEYWORDS

Joining partial 3D models, absolute coordinates, corresponding coordinates, transform matrix, homogeneous matrix, overlapping models, fitting 3D models, model surface triangulation, surface retriangulation

LAMPÁŠ, Marek. *Spojování dílčích 3D modelů povrchu do celkového modelu*. Brno, 2019, 55 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedúci práce: Ing. Adam Chromý, Ph.D.

## VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Spojování dílčích 3D modelů povrchu do celkového modelu“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora

## POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Adamovi Chromému, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno .....

.....

podpis autora

# Obsah

Úvod	9
<b>1 Teoretická časť študentskej práce</b>	<b>11</b>
1.1 Vyhотовovanie modelov . . . . .	11
1.2 Spájanie modelov . . . . .	15
1.2.1 Dopasovávanie modelov v priestore . . . . .	15
1.2.2 Singulárny rozklad (SVD) . . . . .	17
1.2.3 Spojenie povrchov modelov . . . . .	19
<b>2 Výsledky študentskej práce</b>	<b>22</b>
2.1 Návrh riešenia . . . . .	22
2.1.1 Vstupné dáta . . . . .	22
2.1.2 Transformácia modelu . . . . .	23
2.1.3 Priemerovanie povrchu . . . . .	26
2.1.4 Triangulácia . . . . .	31
2.2 Implementácia riešenia . . . . .	34
2.2.1 Prostredie projektu ModelViewer . . . . .	34
2.2.2 Programové riešenie . . . . .	39
2.2.3 Postup spájania čiastkových modelov . . . . .	44
<b>3 Záver</b>	<b>52</b>
<b>Literatúra</b>	<b>53</b>
<b>4 Obsah príloženého CD</b>	<b>55</b>

# Zoznam obrázkov

1	Schématický postup pri spájaní čiastkových 3D modelov. . . . .	9
1.1	Príklad vstupných fotiek pre modelovanie. . . . .	11
1.2	Príklad výstupného modelu zo série fotiek. . . . .	12
1.3	Príklad problematických objektov a scén pre modelovanie zo série fotiek. . . . .	12
1.4	Príklad výzoru modelu s rôznou hustotou bodov. . . . .	14
1.5	Vizuálny príklad funkcie ICP algoritmu. . . . .	16
1.6	Vizuálny príklad erózie povrchu v prekrývajúcej časti. . . . .	19
1.7	Vizuálny príklad "zazipovania"medzery medzi povrchmi. . . . .	20
2.1	Príklad časti setu bodov. . . . .	22
2.2	Príklad modelov s nepasujúcimi absolútnymi súradnicami. . . . .	23
2.3	Príklad modelov posunutých do počiatku sústavy. . . . .	24
2.4	Príklad dopasovaných modelov v priestore. . . . .	26
2.5	Príklad normálových vektorov trojuholníkov a z nich vypočítaných smerníc bodu. . . . .	27
2.6	Ukážka bodov a smerníc pred ich posunutím. . . . .	28
2.7	Ukážka priesečníku smernice bodu s prouholníkom druhého modelu. . . . .	28
2.8	Ukážka po posunutí bodov prvého z modelov. . . . .	29
2.9	Pohľad z vrchu na modely pred pretiangulovaním. . . . .	31
2.10	Pohľad z vrchu na pretiangulovaný povrch. . . . .	32
2.11	Ukážka výsledného povrchu bez priemerovania bodov. . . . .	32
2.12	Ukážka vyhladenia povrchu po priemerovaní. . . . .	32
2.13	Príklad možnej línie povrchu bez trojuholníkov. . . . .	33
2.14	Príklad prvého nájdeného trojuholníka v nevyplnenej medzere. . . . .	33
2.15	Príklad dvoch možností nájdenia druhého trojuholníka v nevyplnenej medzere. . . . .	34
2.16	Príklad triangulácie nevyplnenej línie. . . . .	34
2.17	Užívateľské prostredie projektu ModelViewer- záložka ViewSettings. . . . .	35
2.18	Užívateľské prostredie projektu ModelViewer- záložka ExtraTools. . . . .	36
2.19	Výberové polia v časti Select Models v záložke Extra Tools. . . . .	36
2.20	Zapnutie-vypnutie časti Align Selected Models. . . . .	37
2.21	Sprístupnené tlačidlá v časti Align Selected Models. . . . .	37
2.22	Tabuľka zvolených odpovedajúcich si bodov. . . . .	37
2.23	Prvý zvolený bod v tabuľke odpovedajúcich si bodov. . . . .	38
2.24	Prvý zvolený pár bodov v tabuľke odpovedajúcich si bodov. . . . .	38
2.25	Okno na potvrdenie zmazania odpovedajúcich si bodov v tabuľke. . . . .	38
2.26	Časť Merge Selected Models v záložke Extra Tools. . . . .	39
2.27	Medzera medzi spojenými čiastkovými modelmi. . . . .	44



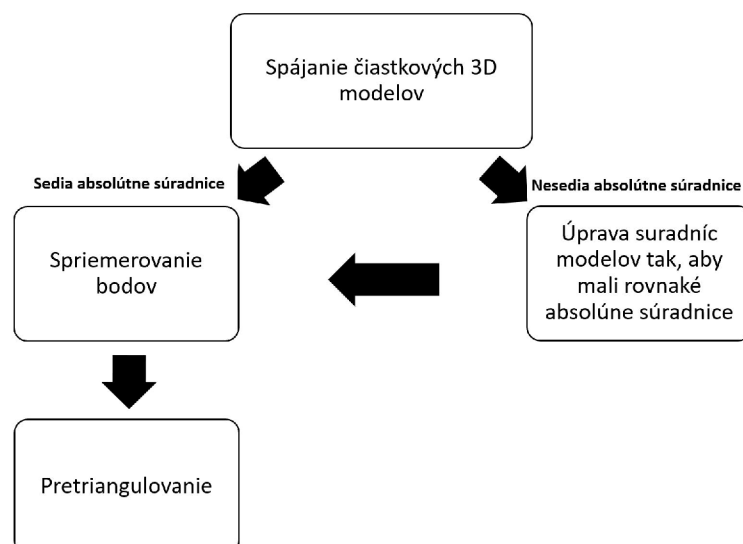
2.28 Príklad vstupných čiastkových modelov s nepasujúcimi absolútnymi súradnicami. . . . .	45
2.29 Príklad výberu modelov v Extra tools na dopasovanie. . . . .	46
2.30 Príklad modelov so zvolenými štyrmi párami odpevedajúcich si bodov. . . . .	46
2.31 Dopasované modely na základe odpevedajúcich si bodov. . . . .	47
2.32 Odpovedajúce si páry bodov na dopasovaných čiastkových modeloch. . . . .	48
2.33 Medzera medzi povrchni dopasovaných čiastkových modelov. . . . .	48
2.34 Dvojitý povrch dopasovaných čiastkových modelov. . . . .	49
2.35 Celkový model vytvorený spojením dvoch čiastkových modelov. . . . .	49
2.36 Miesto kde odstával model po spojení čiastkových modelov. . . . .	50
2.37 Blízky pohľad na povrch v mieste pôvodného prekrytia čiastkových modelov. . . . .	50
2.38 Nevhodné priemerovanie bodov pri spájaní čiastkových modelov. . . . .	51

# Úvod

Skenovanie viacrozmerných modelov ľudstvo pozná od druhej polovice 20. storočia. Nie je to teda žiadna novinka. Od svojho vynájdenia si skenovanie našlo uplatnenie v mnohých oblastiach ako napríklad v priemysle, reverznom inžinierstve, dizajne, architektúre, baníctve a v mnohých ďalších. Využitie si našlo aj v zdravotníctve, kde sa využíva pri viacerých vyšetreniach. Avšak v niektorých oblastiach má isté nedostatky. Napríklad všetky súčasné termografické riešenia využívajú 2D termálne snímky. Práve týmto nedostatkom sa zaoberá systém RoScan. Jedná sa o multispektrálny 3D skenovací systém, ktorý dokáže snímať viacrozmerný model s farebnou vrstvou a informáciou o teplote s veľkým rozlíšením.[10, 11]

Princíp tohto systému je založený na senzorickej hlave, ktorá sa pomocou robotického manipulátora pohybuje okolo skenovaného objektu po predom definovanej trajektórii. Systém RoScan je však limitovaný v možnej veľkosti skenov a preto je potrebné väčšie objekty snímať po častiach a získané čiastkové skeny následne spojiť do výsledného celkového modelu.

Ponúkajú sa dve možnosti ako získať žiadané čiastkové modely. Buď sa pred snímaním každého skenu nastaví nová trajektória a so skenovaným objektom sa vôbec nehýbe, alebo sa preddefinuje jedna trajektória a tá sa použije na získanie medelov všetkých častí tak, že sa zakaždým vhodne nastaví snímaný objekt. Prvá možnosť je samozrejme užívateľsky viac náročná, no jej výhoda je v tom, že všetkým čiastkovým skenom sedia absolútne súradnice a teda takéto dáta sú jednoduchšie na spracovanie. Postup spracovania vysvetľuje táto jednoduchá schéma na obr.1.



Obr. 1: Postup pri spájaní čiastkových 3D modelov.

Z tejto schémy je jasné, že celý algoritmus spracovania modelov s pasujúcimi absolútnymi súradnicami je použitý v spracovaní skenov, ktoré boli snímané tak, že sa hýbalo objektom a teda čiastkové modely na seba v absolútnych súradniciach nesedia. V tomto prípade je spracovanie rozšírené o transformáciu modelov tak, aby sa zaistila ich nadväznosť. Toto dopasovanie je založené na posunutí a pootočení jedného zo skenov pomocou aplikovania transformačnej matice, ktorá je vypočítaná zo znalosti navzájom si odpovedajúcich bodov v oboch skenoch.

Z dôvodu, že sa oba možné prípady spájania líšia iba v transformácii bodov, bude najskôr vysvetlený tento krok a zvyšok práce bude teda zhodný pre obe varianty získania čiastkových modelov.

# 1 Teoretická časť študentskej práce

V posledných desaťročiach zažilo viacrozmerné skenovanie veľký rozmach a dnes je už celkom bežné. Avšak táto oblasť techniky so sebou prináša ďalšie výzvy, ako napríklad potrebu získané modely spájať. Posledný rok som sa venoval práve tejto téme. V práci zhrňam získané poznatky, aké sú možnosti spájania a ako sa rieši v už existujúcich implementáciách.

Problém spájania dvoch čiastkových modelov je veľmi obsiahly pojem. Možností je veľmi veľa pre najrôznejšie požiadavky. Tiež veľmi záleží na type vyhotovovaných trojrozmerných modelov a na type predmetov, ktoré sa snímajú.

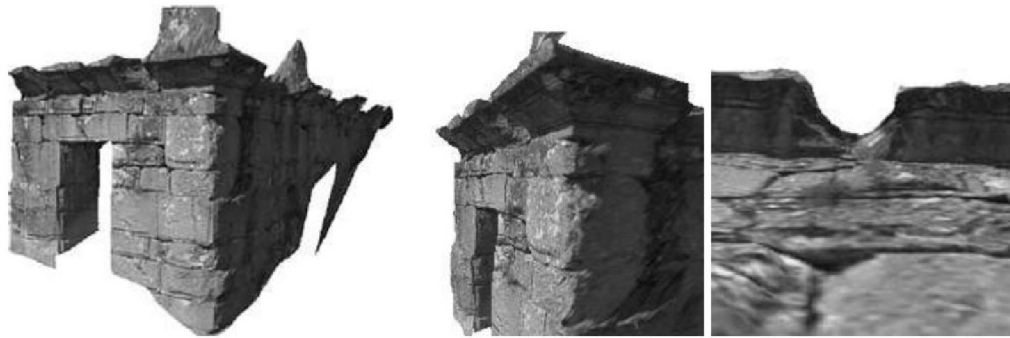
## 1.1 Vyhотовovanie modelov

Modelovanie viacrozmerných modelov je možné robiť rôznymi postupmi. Medzi tieto postupy patrí napríklad rekonštrukcia trojrozmerného modelu zo série obrazových snímok. Túto problematiku rieši vo svojej práci Marc Pollefeys, asistent profesora z univerzity University of North Carolina - Chapel Hill. [1] Ako sa v tejto práci píše, problémom zachytávania objektov fotoaparátmi, teda ich interpretáciou v dvojrozmernej ploche je, že pri tomto postupe sa strácaja informácia o hĺbke predmetu, ktorá je pre trojrozmerné modelovanie zásadná. Aby sa tento problém vyriešil a modelovanie z obrazov bolo možné, musí byť pre tento postup vyhotovených viac snímok modelovaného objektu z roznych uhlov, v ktorých sa hľadajú spojitosti, na základe ktorých je možné túto hĺbku obnoviť.

Jedná sa o veľmi rozšírený postup trojrozmerného modelovania, lebo nevyžaduje žiadne špeciálne zariadenia. Postačujúcim je fotoaparát a software, v ktorom sa fotky objektu spracujú.



Obr. 1.1: Príklad vstupných fotiek pre modelovanie z práce [1].



Obr. 1.2: Príklad výstupného modelu zo série fotiek z práce [1].

Príklad vstupnej série fotiek a z nich vytvoreného modelu ukazujú obrázky obr1.1 a obr1.2.

Marc Pollefeys vo svojej práci tiež informuje, že táto metóda má aj veľké obmedzenia, čo sa týka snímaných objektov. Ideálnym predmetom na použitie pri trojrozmernom modelovaní zo série fotiek je statický objekt, ktorý sa v čase nemení. Tiež by nemal obsahovať príliš zložité štruktúry. Príkladom vhodného objektu môže byť socha, architektonická pamiatka alebo iné pevné objekty.

Objekty, s ktorými má tento postup modelovania najväčšie problémy a výrazne ho limitujú, naznačuje obrázok obr1.3. Sú to teda pohybujúce sa objekty, zložité scény s mnohými diskontinuitami alebo objekty obsahujúce odrazy predmetov alebo scén.



Obr. 1.3: Príklad problematických objektov a scén pre modelovanie zo série fotiek z práce [1].

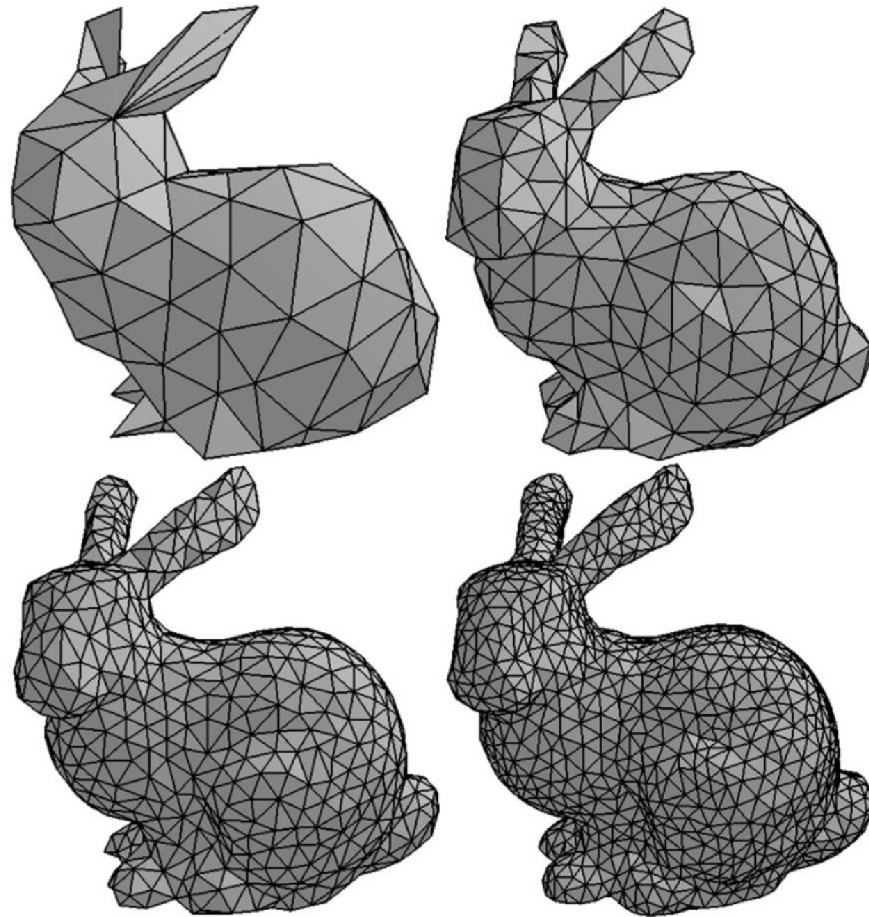
Druhý veľmi rozšírený postup trojrozmerného modelovania využíva, na rozdiel od predchádzajúceho postupu, zachytávanie hĺbky v priestore v určitých bodoch, z ktorých sa neskôr rekonštruuje povrch snímaného objektu. To znamená, že na vytvorenie trojrozmerného modelu stačí jeden snímok skenerom. Keďže pri tomto postupe skener zachytáva priamo hĺbku priestoru, je spracovanie takto získaných dát značne jednoduchšie, než v predchádzajúcom prípade.

Body ale nie sú dostatočné na vykreslenie celého povrchu. Pre vymodelovanie celistvého povrchu je nutné špeciálnym postupom pospájať body do trojuholníkov, ktoré tvoria sieť povrchu. Ako sa môžete dočítať v práci zo Západočeskej univerzity v Plzni s názvom *An Algorithm Recreating 3D Triangle Mesh Faces from Its Edges*, na vytvorenie tvárí, ktoré tvoria jednotlivé výplne siete modelu, nepotrebujeme poznať trojuholníky, ako celok tvorený tromi stenami, ale stačia nám dve spojnice bodov, ktoré majú jeden bod spoločný. [2]

Pri tvorení týchto tvárí je dôležité, aby spojnica dvoch bodov vždy patrila maximálne dvom trojuholníkom a aby sa spojnice bodov, tvoriace steny trojuholníkov, medzi sebou nekrižovali. To sa dosiahne tým, že su pospájané medzi sebou vždy susedné body. Teda vo väčšine prípadov patrí spojnica dvoch bodov práve dvom trojuholníkom. Výnimku tvoria steny trojuholníkov modelov, ktoré tvoria okraje modelu, ak objekt nie je snímaný zo všetkých strán. Avšak algoritmus popísaný v citovanej práci očakáva ako vstup čiastkové modely, ktoré po spojení tvoria uzavretý model.

Vyplnením vzniknutej siete vznikne teda celistvý povrch, ktorý definujú jednotlivé plochy trojuholníkov. Ako bolo spomínané, týmto vyplneným trojuholníkom sa hovorí tváre modelu. Čím viac bodov je pri snímaní povrchu zaznamenaných, čím je myslené viac bodov pri rovnakej ploche povrchu, tým menšie plochy obsahuje sieť trojuholníkov, čo je pochopiteľné, keďže sú body snímané hustejšie a spojnice susedných bodov sú kratšie. Čím sú jednotlivé tváre modelu menšie, tým prirodzenejšie a vyhladenejšie model vyzerá. Pri veľmi riedko snímaných bodoch sa totiž stáva, že na spojniciach tvárí, s rôznymi uhlami otočenia, je zreteľný zlom. A tiež sa na objekte zachytí menej detailov, čo môže objekt zkrasliť. Lepšiu predstavu tohto problému získate pri pohľade na obrázok obr1.4.

Skenovanie modelov povrchu, systémom snímania jednotlivých bodov, je možné v dnešnej dobe robiť v zásade dvoma spôsobmi. Jeden z nich je postup, kedy je skener statický a čiastkové modely sa zhotovujú tak, že sa pred skenerom rôzne otáča snímaný objekt. Rovnaký efekt má aj prenosný skener, ktorým sa naskenuje objekt z rôznych strán. Výsledkom tohto spôsobu snímania modelov je, že modely sú rôzne rozmiestnené v trojrozmernom súradnicovom priestore a neskôr sa musia jednotlivé skeny spojiť do celkového modelu. Výhodou je však, že tieto skenovacie systémy sú celkom lacné a na väčšinu aplikácií pri potrebe spájania sú dostačujúce.



Obr. 1.4: Príklad výzoru modelu s rôznou hustotou bodov (zdroj: [http://www.cmap.polytechnique.fr/~peyre/images/test\\_remeshing.jpg](http://www.cmap.polytechnique.fr/~peyre/images/test_remeshing.jpg)).

Druhým spôsobom, ktorý je výrazne cenovo nákladnejší je, že skener, snímajúci trojrozmerné modely, využíva nejaký spôsob sledovania svojej polohy a prispôbuje týmto údajom súradnice snímaných modelov. Toto využívajú napríklad skenery terénu v prírode, ktoré podľa GPS polohy prepočítavajú polohu skenu. Výsledkom je, že jednotlivé trojrozmerné snímky na seba v trojrozmernom súradnom systéme nadväzujú a ich dodatočná transformácia nie je nutná.

Aplikácie, pri ktorých sa skenujú menšie predmety, využívajú často na skenovanie v absolútnych súradniciach roborické ramená. Skener je pri tejto aplikácii umiestnený na tomto ramene, ktoré vždy pozná svoju polohu a smer natočenia. Tento spôsob pracovania s absolútnymi súradnicami je veľmi presný. Na projekte, kde sa na skenovanie používa roborické rameno, pracuje aj vedúci mojej práce Ing. Adam Chromý, Ph.D. [11] [10]

## 1.2 Spájanie modelov

### 1.2.1 Dopasovanie modelov v priestore

Pri potrebe spájania jednotlivých získaných skenov sa ponúka veľmi veľa možností. Jedno z možných rozdelení postupov je, či sa jedná o automatický proces spájania na základe podobnosti v jednotlivých modeloch, alebo je potrebná asistencia užívateľa, ktorý v jednotlivých modeloch podobnosti vyznačí. Automatickému spájaniu modelov sa venujú aj Daniel F. Huber a Martial Hebert z robotického inštitútu *The Robotics Institute Carnegie Mellon University* v práci "Fully automatic registration of multiple 3D data sets", v ktorej popisujú princíp spájania jednotlivých modelov, ktoré boli snímané stacionárnym skenerom, ktorým nasníмали objekt z rôznych strán. [3]

V tejto práci sa venujú automatizovaniu dopasovania modelov, ktoré zachytávajú časti jedného objektu. Základom ich práce je použitie algoritmu ICP (iterative closest point), ktorý je založený na prechádzaní bodu za bodom dvoch spájaných modelov, na hľadanie dvojíc najbližších bodov a na minimalizovaní vzdialeností medzi nimi. Algoritmus ICP vymysleli Paul Besl and Neil McKay a predstavili ho svetu v práci s názvom *A method of registration of 3-D shapes*. [4] Táto práca, publikovaná v roku 1992, sa stala v podstate základom spája častkových modelov. Predpokladom pre funkčnosť ich postupu je, že modely sa na častiach prekrývajú.

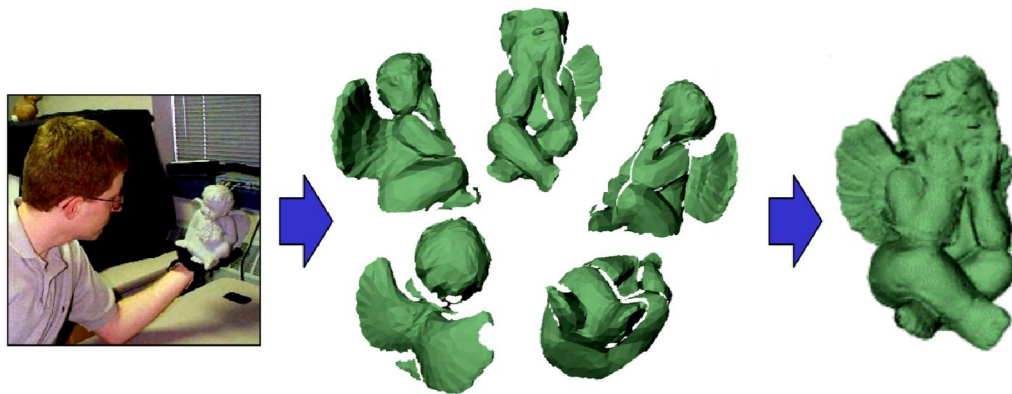
Vstup ICP algoritmu vyžaduje počiatočný odhad transformácie, ktorá je potrebná na dopasovanie častkových modelov. Po vykonaní tejto počiatočnej transformácie, ktorá nie je úplne presná, pokračuje algoritmus v počítaní ďalších transformácií potrebných na úplné dopasovanie. Nejedná sa tak o úplne autonómny spôsob dopasovania.

Algoritmus ICP je tvorený v podstate tromi krokmi, ktoré sa dookola opakujú po počiatočnej transformácii na základe odhadu užívateľa. Týmito krokmi sú:

- Pre všetky body v zdrojových modeloch sa vypočíta najbližší bod, ktorý sa mu priradí.
- Stanoví sa potrebná rotácia a translácia vedúca k tomu, že sa vzdialenosti medzi najbližšími bodmi skrátia. Tento krok využíva na stanovenie presnosti spojenia metódu najmenších štvorcov.
- Na základe stanovených transformácií sa model, ktorý sa dopasováva, upraví. A pokračuje sa krokom jedna, kým nie sú splnené kritéria spojenia definované na počiatku.

Príklad vstupných častkových modelov a z nich spojeného celkového modelu, ktorý bol spojený spomínaným upraveným autonómny algoritmom, ukazuje obrázok obr1.5.





Obr. 1.5: Vizuálny príklad funkcie ICP algoritmu z práce [3].

Veľkou výhodou algoritmu popísaného v práci Daniela F. Huber a Martiala Herbert [3] je, že eliminuje ľudský faktor a na dopasovanie dvoch častí objektu, ktoré sú vyhodnotené ako prekrývajúce sa, využíva podobnosti čiastkových modelov a metódu najmenších štvorcov, čo znamená, že dopasovanie je veľmi presné. Avšak ako sa zmieňujú v závere, algoritmus nemá stopercentnú úspešnosť. Občas sa totiž stáva, že postup vyhodnotí prekrývajúce sa časti objektu nesprávne. Táto chyba sa však vyskytla iba v jednom z desiatich prípadov, preto má tento postup v celku dobré výsledky.

Od publikácie tejto práce sa objavilo mnoho vylepšení a rozšírení, ktoré mali za úlohu zlepšiť je funkčnosť a zefektívniť výpočty, ktoré sa v nej prevádzajú.

Na spominatej práci stávali aj H. Hügli a T. Jost v práci s názvom *A match and merge method for 3D modeling from range images*, ktorý pôvodný algoritmus upravili tak, aby minimalizovali chyby v hľadaní podobností. [5] To dosiahli tým, že definovali maximálnu vzdialenosť prípustnú medzi dvoma najbližšími bodmi, do ktorej sa považujú za odpovedajúce si a majú váhu pri výpočtoch. Celkovo dosiahli dobré výsledky, no ako sa zmieňujú v tejto práci, na získanie optimálnych výsledkov je potrebné, aby prekrývajúca sa časť tvorila aspoň 30 až 50 percent povrchu modelov.

V súčasnej dobe už existuje viac možností ako čiastkové modely spojiť automaticky, bez zásahu užívateľa, no ich veľkým problémom je časová a výpočetná náročnosť a tiež to, že niektoré aplikácie spájania vyžadujú spoľahlivosť algoritmu. Preto je vhodné zvážiť aj omnoho jednoduchšiu, no možno o trochu menej presnú, možnosť volenia podobností v čiastkových modeloch užívateľom.

Postupy využívajúce asistenciu užívateľa sú rozšírenejšou implementáciou spájania čiastkových modelov. Človek veľmi jednoducho nájde podobnosti medzi dvoma čiastkovými modelmi, ktoré sa na časti prekrývajú. Preto môže užívateľ v procese

dopasovávaní častí modelu významne pomôcť algoritmu tým, že označí na jednotlivých modeloch body alebo časti, ktoré sú na oboch čiastkových modeloch rovnaké.

Ak je algoritmus schopný z definovaných odpovedajúcich si častí získať dostatočné množstvo informácií, je potom schopný dopasovať dané dva modely iba na základe týchto zvolených bodov.

## 1.2.2 Singulárny rozklad (SVD)

Najviac využívaný postup pri spájaní trojrozmerných modelov v priestore, na základe korešpondencie bodov, je SVD (singulárny rozklad - Singular Value Decomposition). Singulárny rozklad je použitý aj v spomínanom algoritme s názvom *Iterative closest point*, čo potvrdzuje jeho dôležitosť pri dopasovaní setov bodov. Tento postup má rôzne použitia či už v dvojrozmernom alebo trojrozmernom priestore.

Aplikácie, ktoré využívajú singulárny rozklad, často vyžadujú počítanie projekcie alebo vzdialenosti od matice. Veľkou výhodou je, že tento postup nepredpokladá stopercentnú zhodu spájaných modelov a počíta s predpokladom, že vstupné dáta obsahujú určité nepresnosti, alebo inak povedané šum.

Ako sa môžeme dočítať vo veľmi zaujímavej práci s názvom *Least-Squares Fitting of Two 3-D Point Sets* od autorov K. S. Arun, T. S. Huang a S. D. Blostein [13], dva sety odpovedajúcich si bodov vyjadrených v maticiach  $\mathbf{A}$  a  $\mathbf{B}$ , sú totožné s malými odchýlkami v podobe šumu a maticu  $\mathbf{B}$ , môžeme vyjadriť ako maticu  $\mathbf{A}$  transformovanú rotačnou a tlaňáchnou maticou v priestore. Túto rovnosť vyjadruje nasledujúci vzťah vzťah.

$$\mathbf{B}_i^t = \mathbf{R} * \mathbf{A}_i^t + \mathbf{T} + \mathbf{N}_i$$

kde

- $\mathbf{R}$  je rotačná matica rozmeru  $3 \times 3$
- $\mathbf{T}$  je tlaňáčný vektor ( $3 \times 1$  stĺpcová matica)
- index  $i$  značí riadok v maticiach odpovedajúcich si bodov  $\mathbf{A}$  a  $\mathbf{B}$ , v ktorých nájdeme vždy na rovnakom riadku polohy dvoch odpovedajúcich si bodov
- $\mathbf{N}_i$  je vektor šumu
- index  $t$  v  $\mathbf{B}_i^t$  a  $\mathbf{A}_i^t$  značí transponovanú maticu

Princípom využitia singulárneho rozkladu je vyjadrenie presnej rotačnej a tlaňáčnej matice, ktoré sú potrebné na transformáciu jedného z modelov tak, aby súradnice oboch modelov v priestore na seba pasovali. Táto metóda využíva pre čo najväčšiu presnosť metódu najmenších štvorcov. Taktiež je jej veľkou výhodou rýchlosť kalkulácie, ktorá je však veľmi závislá na rozmeroch vstupných matíc.

Spomínaná práca navrhuje päť krokový postup, ako z uvedenej rovnosti medzi dvoma modelmi určiť danú rotačnú a translačnú maticu. Princíp rotácie spočíva v afinnej matici, ktorá využíva rotáciu okolo jednotlivých osí.

*Krok 1:* Najskôr sa musí spočítať centroid oboch modelov podľa vzťahu:

$$\mathbf{CentroidA} = \frac{1}{N} \sum_{i=1}^N \mathbf{A}_i$$

kde

- $\mathbf{A}_i$  značí bod na  $i$ -tom riadku matice  $\mathbf{A}$
- $N$  je počet riadkov v matici  $\mathbf{A}$

Analogicky sa spočíta aj pre body matice  $\mathbf{B}$  a oba modely sa posunú v priestore tak, aby sa ich centroidy nachádzali v počiatku súradnej sústavy.

$$\mathbf{QA}_i = \mathbf{A}_i - \mathbf{CentroidA}$$

$$\mathbf{QB}_i = \mathbf{B}_i - \mathbf{CentroidB}$$

*Krok 2:* Spočíta sa 3 x 3 matica.

$$\mathbf{H} = \sum_{i=1}^N \mathbf{QA}_i^{t*} \mathbf{QB}_i$$

*Krok 3:* Vypočíta sa singulárny rozklad matice  $\mathbf{H}$ .

$$\mathbf{H} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^t$$

kde

- $\mathbf{U}$  a  $\mathbf{V}$  sú 3 x 3 ortogonálne matice
- $\mathbf{\Lambda}$  je 3 x 3 diagonálna matica s nezápornými hodnotami

*Krok 4:* Nakoniec sa dopočíta rotačná matica  $\mathbf{R}$ .

$$\mathbf{R} = \mathbf{V} \mathbf{U}^t$$

*Krok 5:* Posledný krok slúži na overenie správnosti výsledku postupu. Spočíva vo vypočítaní determinantu matice  $\mathbf{R}$ .

- Ak  $\det(\mathbf{R}) = +1$ , tak bol postup úspešný a matica  $\mathbf{R}$  je hľadaná rotačná matica
- Ak  $\det(\mathbf{R}) = -1$ , tak postup zlyhal

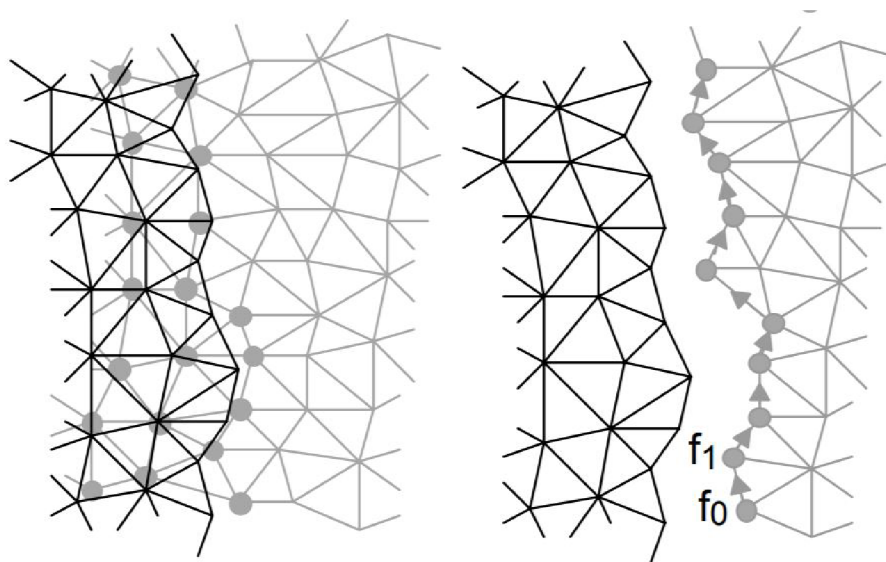
Ak bol postup úspešný, aplikuje sa vypočítaná rotačná matica  $\mathbf{R}$  na všetky body modelu, ktorý sa má v priestore dopasovať.

### 1.2.3 Spojenie povrchov modelov

Pri snímaní modelov veľmi často nastáva situácia, kedy jednotlivé čiastkové modely obsahujú rovnaké úseky snímaného objektu a preto sa na častiach povrchu prekrývajú. Presne snímané nadväzujúce modely, ktoré sa neprekrývajú, je možné snímať v podstate iba pri použití postupu skenovania pracujúceho v absolútnych súradniciach.

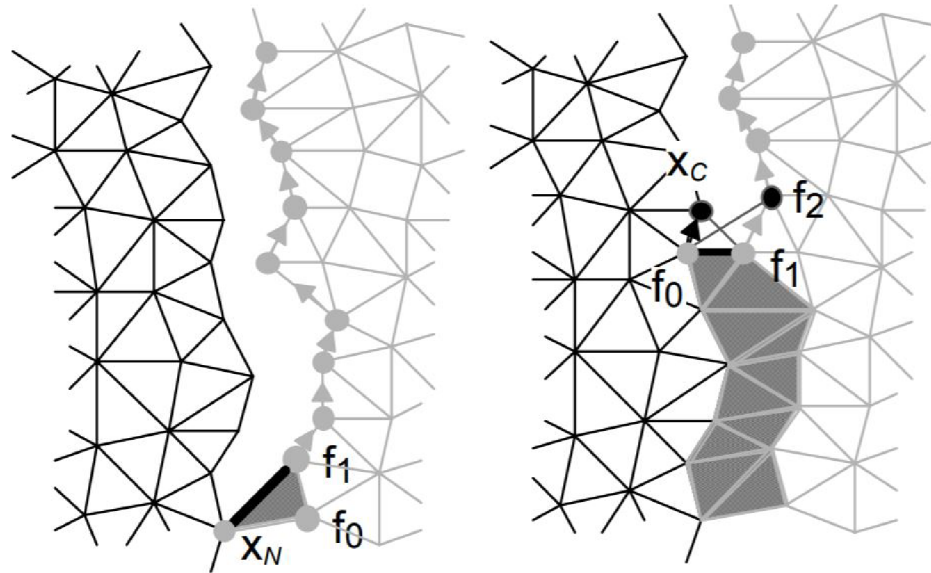
Ale predsa len aj tieto povrchy, tvorené pospájanými troholníkmi, ktoré definujú jednotlivé tváre povrchu, musia byť dodatočne medzi sebou prepojené. Inak by medzi modelmi mohla vzniknúť medzera, kde sú síce definované body modelov blízko seba, no nie sú medzi sebou tieto krajné body z oboch čiastkových modelov prepojené spojnicami bodov a netvorí tak tváre modelu. To, že v tomto páse nie sú definované tváre povrchu znamená, že pri vykreslení modelov by v danom mieste bez tváre vznikla diera v povrchu.

Rovnaký problém sa ale rieši aj vo väčšine aplikácií, kde sú vstupom čiastkové modely, ktoré sa na častiach povrchu prekrývajú. Aplikácie tohto typu veľmi často fungujú na princípe, že prekrývajúca časť sa z jedného z čiastkových modelov odstráni a zvyšok povrchu modelu sa pripojí rovnako, ako v prípade, keď boli modely snímané bez prekrytia. Tento princíp aplikácie a postup prepojenia čiastkových modelov vysvetľuje aj už spomínaná práca s názvom *A match and merge method for 3D modeling from range images*. [5] Princíp odstránania prekrývajúceho sa povrchu modelu ukazuje nasledujúci obrázok obr1.6. Tento princíp erózie odstráni všetky body trojuholníkov, ktoré ležia v prekrývajúcej časti.



Obr. 1.6: Vizualný príklad erózie povrchu v prekrývajúcej časti z práce [5].

Väčšina vedeckých prác, využívajúcich tento postup nakladania s prekrývajúcimi sa povrchni, využíva postupnú eróziu povrchu, kým prekrytie nezmizne. [6] [7] Po odstránení celej prekrývajúcej sa časti, sú potom povrchy prepojené takpovediac "zazipovaním", ktoré spočíva v pospájaní najbližších bodov tvoriacich okraj tejto medzery, ktorá medzi nimi po erózii vznikne. Jeden z možných spôsobov vyplnenia medzery je vizuálne naznačený na obrázku obr1.7.



Obr. 1.7: Vizuálny príklad "zazipovania" medzery medzi povrchni z práce [5].

Postup z obrázku obr1.7, ktorý je popísaný v práci *A match and merge method for 3D modeling from range images*, [5] je založený na znalosti okrajových bodov medzery. Body tvoriace okraje sú identifikované v procese erózie. Identifikácia okrajových bodov je pri kroku zazipovania tou obťažnejšou časťou. V momente, keď sú tieto body známe, je postup už jednoduchý a dá sa zhrnúť do týchto krokov.

- Zvolí sa spojnica dvoch bodov, ktorá je obsiahnutá na začiatku okraju erodovaného modelu.
- Nájde sa pre body, spojené touto spojnicou, najbližší bod z protilahlého okraju medzery.
- Tým vznikne prepojavací trojuholník prepojujúci modely s dvoma stenami prechádzajúcimi medzerou.
- Nájde sa najbližší bod z okrajov, k bodom spojeným stenou vzniknutého trojuholníka a vytvorí sa ďalší trojuholník tvoriaci tvár povrchu.
- Tento krok sa opakuje, kým sa medzi sebou neprepoja všetky okrajové body.

Niektoré iné práce pre zmenu využívajú na spojenie dvoch čiastkových modelov pretriangulovanie celého povrchu, alebo prekrývajúcej sa časti. [8] [9] Pri tomto

riešení teda nie sú odstraňované žiadne body, iba sú medzi sebou znovu pospájané spôsobom, kedy sa ich spojnice nekrižujú a tváre modelu tvoria celistvý povrch.

Jednoznačne určiť, ktorý postup je lepší sa všeobecne nedá. Záleží to na konkrétnej implementácii. Každopádne toto sú len základné postupy, na ktorých stavia množstvo ďalších publikácií, ktoré sa snažia jeden z postupov vylepšiť. Príkladom je práve aj práca s názvom *Mesh Integration Based on Co-Measurements*, ktorej autorom je R. Pito. [6] Jeho práca využíva postup erózie, ale snaží sa výpočet zefektívniť tým, že prekrývajúcu sa časť modelov počíta už v procese dopasovania daných čiastkových modelov a tým spájanie urýchluje.

Môj subjektívny názor je, že postup spájania erodovaním prekrývajúcich sa častí, je vhodnejší pre objekty, ktoré majú svoj pevný tvar. Teda objekty, pri ktorých sa predpokladá, že jednotlivé čiastkové modely po transformácii na seba presne sedia. Naopak, ak sa skenujú a modelujú trojrozmerné modely predmetov, ktoré v krátkom čase medzi získaním jednotlivých čiastkových modelov môžu aspoň nepatrne zmeniť svoj tvar, tak je podľa mňa vhodnejšie zvoliť postup pretrojholníkovania a tak oba povrchy spriemerovať.

Cielom mojej práce je navrhnúť algoritmus spájania čiastkových modelov ľudskeho tela. Jedná sa teda o objekty, ktoré môžu z rôznych dôvodov veľmi rýchlo zmeniť svoj tvar. Dôvodom môže byť napríklad, že sa na skenovanej časti tela napne nejaký sval, alebo na povrchu navrie žila. Preto tomuto faktoru prisposobím návrh postupu v praktickej časti.

## 2 Výsledky studentské práce

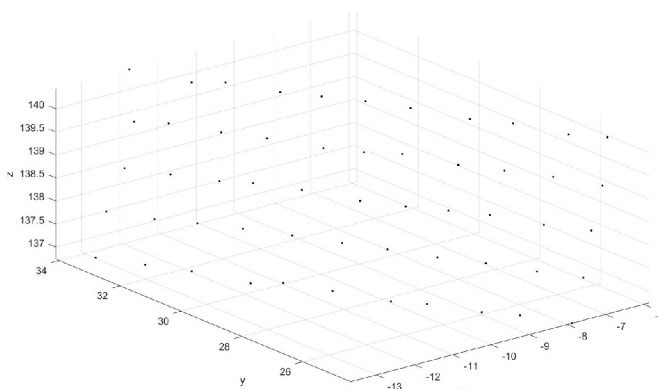
### 2.1 Návrh riešenia

#### 2.1.1 Vstupné dáta

Algoritmus, ktorý budem popisovať, dostáva ako vstup dva modely triedy **RoScanModel**, teda modely, s ktorými štandardne pracuje celý systém RoScan. Algoritmus využije potrebné informácie oboch čiastkových modelov bez zmeny vstupných dát a vráti nový model, ktorý je výslednou kombináciou.

Zo vstupných dát triedy **RoScanModel** nás budú najviac zaujímať list bodov čiastkového modelu **points**, list polôh odpovedajúcich si bodov **FeaturePointList**, ktorý je pri vstupe prázdny, a pole celočíselných hodnôt **indices**, reprezentujúce indexy bodov v liste **points**, kde vždy trojica indexov popisuje trojuholník tvoriaci povrch modelu.

- List **points** obsahuje N bodov triedy **ModelPoints** tvoriacich model. Táto trieda obsahuje v podstate všetky základné údaje modelu- informáciu o polohe bodu, o farbe bodu a informáciu o bodovej normále povrchu modelu, ktorá bude vysvetlená ďalej v práci.
- List **FeaturePointList** je pri vstupe prázdny a v prípade, že modelom, ktoré sa budú spájať nepasujú absolútne súradnice, sa naplní bodmi, ktoré budú užívateľom zvolené ako odpovedajúce si na oboch modeloch. Aby bolo dopasovanie modelov možné, musí užívateľ zvoliť minimálne štyri body, pričom horná hranica nie je stanovená.
- Pole **indices** obsahuje M indexov bodov listu **points**, kde rovnaký index môže byť použitý viackrát, lebo jeden bod takmer vždy tvorí vrchol viac ako jednému trojuholníku.



Obr. 2.1: Časť setu bodov.

## 2.1.2 Transformácia modelu

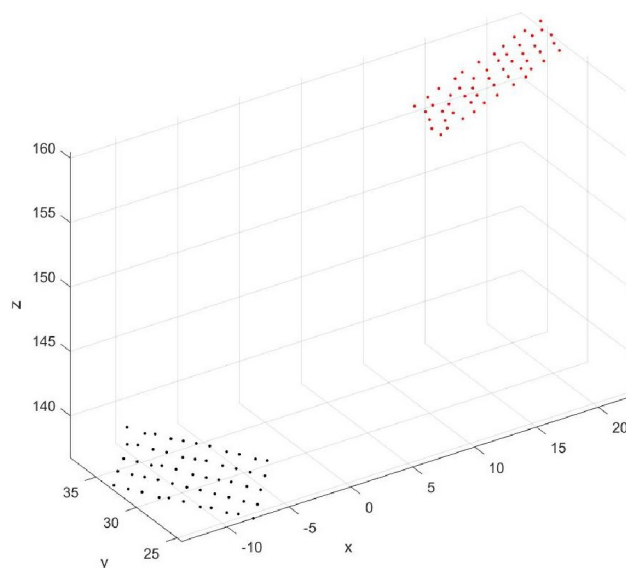
Pokiaľ naskenovaným modelom nesedia absolútne súradnice (obr.2.2), musíme jeden z nich vhodne otočiť a posunúť, tak aby sme toto pasovanie súradníc zaručili. To zabezpečíme aplikovaním transformačných matic, ktoré zistíme pomocou SVD (singulárny rozklad – Singular Value Decomposition)[12]. Transformáciu môžeme vypočítať podľa nasledujúceho postupu:[13]

Dostali sme dva čiastkové modely, kde oba modely obsahujú list polôh bodov **FeaturePointList**. Kvôli singulárnemu rozkladu sa musia tieto listy vyjadriť ako matice veľkosti  $\mathbf{R} \times 3$ , kde každý riadok matice obsahuje  $x$ ,  $y$  a  $z$  polohu jedného bodu. Pre rozlíšenie označme maticu tvorenú bodmi listu **FeaturePointList** prvého modelu **PointsA** a maticu tvorenú bodmi listu **FeaturePointList** druhého modelu **PointsB**. Pre tieto dve matice bodov platí nasledujúci vzťah

$$\mathbf{PointsB}_i^t = \mathbf{R} * \mathbf{PointsA}_i^t + \mathbf{T} + \mathbf{N}_i$$

kde

- $\mathbf{R}$  je rotačná matica rozmeru  $3 \times 3$
- $\mathbf{T}$  je translačný vektor ( $3 \times 1$  stĺpcová matica)
- index  $i$  značí riadok v maticiach odpovedajúcich si bodov **PointsA** a **PointsB**, v ktorých nájdeme vždy na rovnakom riadku polohy dvoch odpovedajúcich si bodov
- $\mathbf{N}_i$  je vektor šumu
- index  $t$  v  $\mathbf{PointsB}_i^t$  a  $\mathbf{PointsA}_i^t$  značí transponovanú maticu



Obr. 2.2: Modely ktorým nepasujú absolútne súradnice.



*Krok 1:* Najskôr si musíme spočítať centroid oboch modelov podľa vzťahu

$$\mathbf{CentroidA} = \frac{1}{N} \sum_{i=1}^N \mathbf{PointsA}_i$$

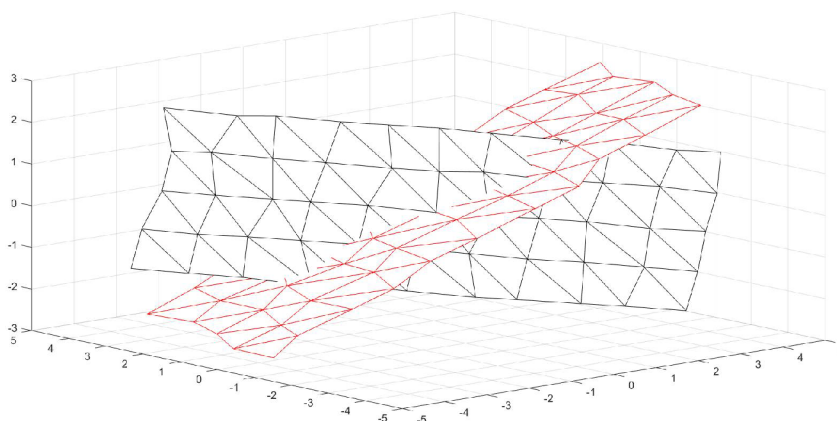
kde

- $\mathbf{PointsA}_i$  značí bod na  $i$ -tom riadku matice  $\mathbf{PointsA}$
- $N$  je počet riadkov v matici  $\mathbf{PointsA}$

Analogicky ho spočítame aj pre body matice  $\mathbf{PointsB}$  a oba modely posunieme v priestore tak, aby sa ich centroidy nachádzali v počiatku súradnej sústavy ako je vidieť na obrázku obr2.3.

$$\mathbf{QA}_i = \mathbf{PointsA}_i - \mathbf{CentroidA}$$

$$\mathbf{QB}_i = \mathbf{PointsB}_i - \mathbf{CentroidB}$$



Obr. 2.3: Sety s centroidmi posunutými do počiatku sústavy.

*Krok 2:* A spočítame 3 x 3 maticu

$$\mathbf{H} = \sum_{i=1}^N \mathbf{QA}_i^{t*} \mathbf{QB}_i$$

*Krok 3:* Vypočítame si transformačné matice pomocou SVD.

$$\mathbf{H} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^t$$

kde

- $\mathbf{U}$  a  $\mathbf{V}$  sú 3 x 3 ortogonálne matice
- $\mathbf{\Lambda}$  je 3 x 3 diagonálna matica s nezápornými hodnotami

*Krok 4:* Nakoniec dopočítame maticu rotačnú maticu  $\mathbf{R}$

$$\mathbf{R} = \mathbf{V}\mathbf{U}^t$$

*Krok 5:* Vypočítame determinant matice  $\mathbf{R}$

- Ak  $\det(\mathbf{R}) = +1$ , tak bol postup úspešný a matica  $\mathbf{R}$  je naša hľadaná rotačná matica
- Ak  $\det(\mathbf{R}) = -1$ , tak postup zlyhal

Ak bol postup úspešný, aplikujeme vypočítanú rotačnú maticu  $\mathbf{R}$  na všetky body modelu, ktorý chceme dopasovať. To urobíme jednoducho tak, že každý bod čiastkového modelu vynásobíme rotačnou maticou  $\mathbf{R}$ .

Docielili sme, že oba modely sú teraz rovnako otočené a stačí nám vypočítať o koľko sa musí jeden z modelov posunúť, aby sa prekrývali, ako je možné vidieť na obr2.4. To môžeme jednoducho urobiť tak, že si zistíme priemernú vzdialenosť medzi odpovedajúcimi si bodmi, ktorých polohy obsahujú matice  $\mathbf{PointsA}$  a  $\mathbf{PointsB}$  a všetky body jedného zo skenov, ktoré, ako bolo spomínané vyššie, sa nachádzajú v liste bodov  $\mathbf{points}$ , posunieme práve o túto vzdialenosť.

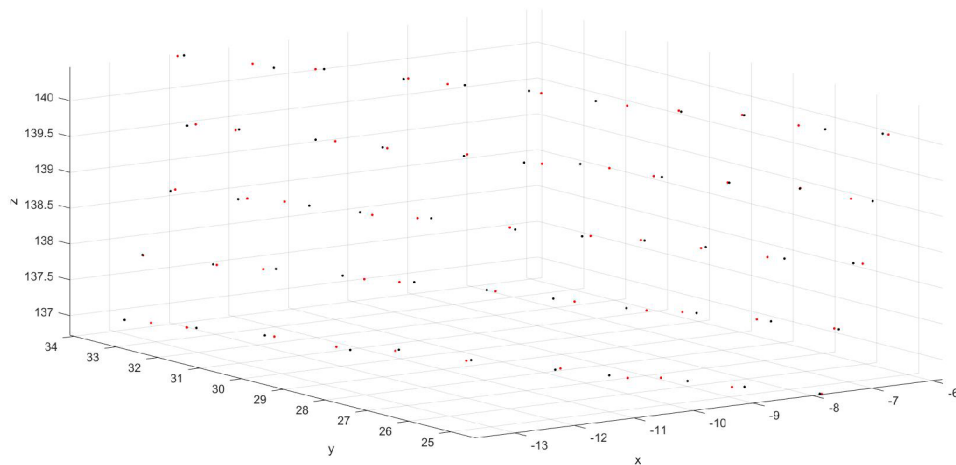
$$\mathbf{TranslationB} = \frac{1}{N} \sum_{i=1}^N (\mathbf{PointsA}_i - \mathbf{PointsB}_i)$$

$$\mathbf{points}_i = \mathbf{points}_i + \mathbf{TranslationB}$$

kde

- $\mathbf{PointsA}_i$  a  $\mathbf{PointsB}_i$  značia odpovedajúce si body na  $i$ -tom riadku matíc  $\mathbf{PointsA}$  a  $\mathbf{PointsB}$
- $N$  je počet bodov v liste  $\mathbf{FeaturePointList}$
- $\mathbf{points}_i$  je  $i$ -ty bod modelu, ktorý dopasovávame

Presnejší postup dopasovania by spočíval v posunutí bodov modelu pomocou metódy najmenších štvorcov tak, že kvadráty vzdialeností medzi odpovedajúcimi bodmi budú najmešie možné. Avšak je potrebné si uvedomiť, že body, ktoré sa použijú na posunutie modelu, sú zvolené užívateľom a preto samotné odpovedajúce si body nie sú vybrané so stopercentnou presnosťou. Preto je možno posunutie bodov pomocou metódy najmenších štvorcov zbytočne náročné a úplne postačí posunutie o priemernú vzdialenosť.



Obr. 2.4: Dopasované modely.

### 2.1.3 Priemerovanie povrchu

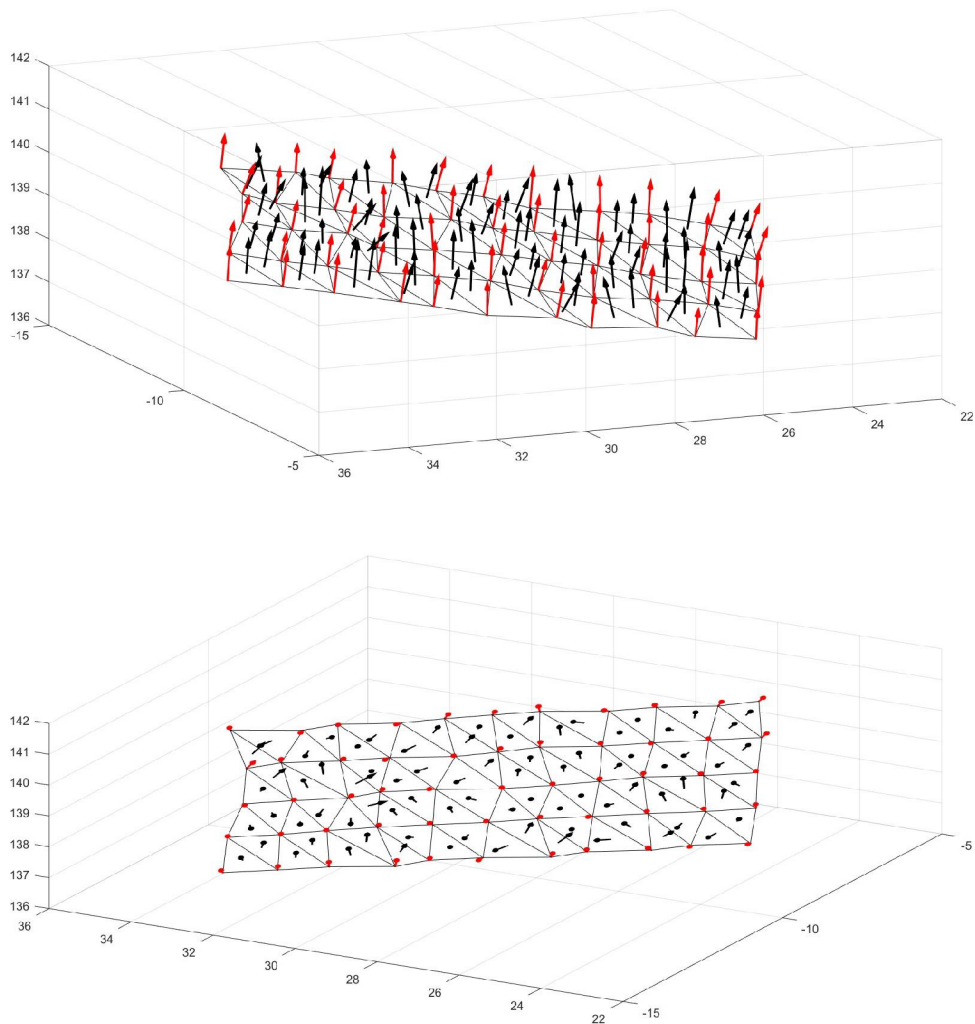
Pred tým ako začnem opisovať túto kapitolu by som rád definoval nový pojem "smernica bodu", ktorý je pre priemerovanie povrchu čiastkových modelov kľúčový a bude v tejto časti algoritmu často spomínaný.

Každý trojuholník, ktorý tvorí povrch modelu, sa skladá vždy z troch bodov, ktoré definujú rovinu v priestore, ktorej časť tvorí plocha daného trojuholníka. Jeden bod, až na výnimočné prípady, tvorí vrchol viacerých trojuholníkov. Výnimky sú naozaj ojedinelé a môžu to byť napríklad body v ostrých rohoch modelu. Trojuholníky sú tvorené tak, že sa vždy práve 2 trojuholníky dotýkajú stenami, aby vytvorili celistvý povrch. Teda s výnimkou stien trojuholníkov, ktoré tvoria okraje modelu a teda patria iba jednému trojuholníku. Každá rovina definovaná týmito trojuholníkmi má svoju normálu, ktorá je kolmá na danú rovinu. Aritmetický priemer všetkých normál trojuholníkov, ktorých jeden z vrcholov tvorí daný bod, bude ďalej definovaný ako smernica bodu. Smernicu bodu teda vypočítame tak, že sčítame jednotlivé súradnice vektorov trojuholníkov priliehajúcich k bodu a vydělíme počtom týchto vektorov.

Nemusíme teda poznať normály rovín definovaných trojuholníkmi, lebo ak daný bod tvorí vrchol trojuholníka, je zrejme že v tomto bode začínajú úsečky dvoch strán trojuholníka, ktoré smerujú k ďalším dvom bodom, definujúcim vrcholy jedného trojuholníka. Tieto dve úsečky, ktoré začínajú v jednom bode, nám stačia na definovanie roviny priliehajúceho trojuholníka.

Pre lepšie pochopenie môžete na nasledujúcich obrázkoch obr.2.5 vidieť časť povrchu modelu, na ktorom sú čiernou farbou vyznačené normály plôch trojuholníkov

a červenou farbou z nich vypočítané smernice bodov.

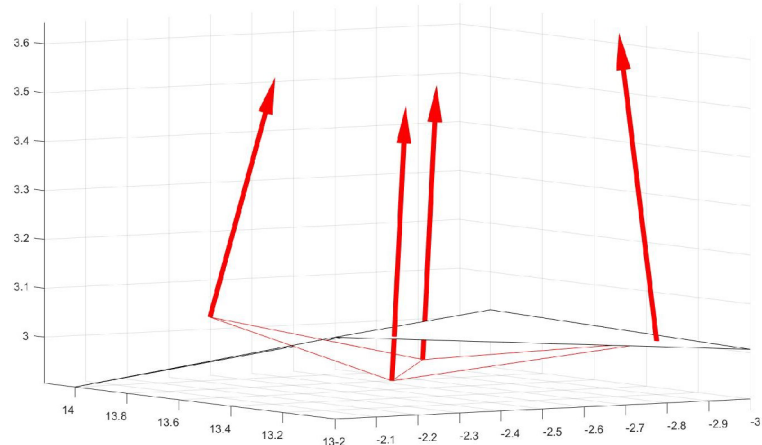


Obr. 2.5: Normálové vektory trojuholníkov a z nich vypočítané smernice bodov.

Modely sa skladajú z množiny bodov, ktoré sa na určitej ploche prekrývajú. S istotou však môžeme povedať, že tieto navzájom si korešpondujúce body z oboch modelov nie sú totožné. Napríklad si predstavme, že skenujeme model predlaktia a smímame dva modely, ktoré chceme spojiť. Toto skenovanie zaberie čas a teda sa môže stať, že na ruke navrie nejaká žila, alebo sa ruka trochu pohne. Takto naskenované modely by nám na seba teda nepasovali a vznikol by nám dvojitý povrch ako na obrázku obr2.6. Musíme preto nájsť priemer týchto modelov, ktorý bude čo najpresnejšie vykreslovať snímaný objekt.

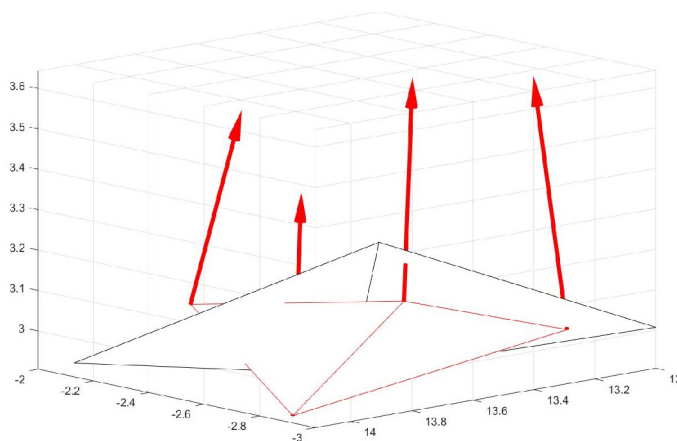
Body modelu sú snímané v určitých miestach povrchu a teda v týchto bodoch vieme ako daný snímaný objekt vyzerá úplne presne. No nevieme s istotou určiť, čo sa

nachádza medzi týmito bodmi. Preto korešpondujúce body nebudeme priemerovať. Zanecháme všetky body oboch modelov. Budeme sa len snažiť spriemerovať tieto povrchy tak, že povrch v prekrývajúcej sa časti modelov pretrianglujeme a pokúsime sa ho čo najviac vyhladiť s čo najpresnejším zachovaním oboch pôvodných modelov. Preto ak budeme bod posúvať, budeme to robiť iba po smernici tohto bodu.



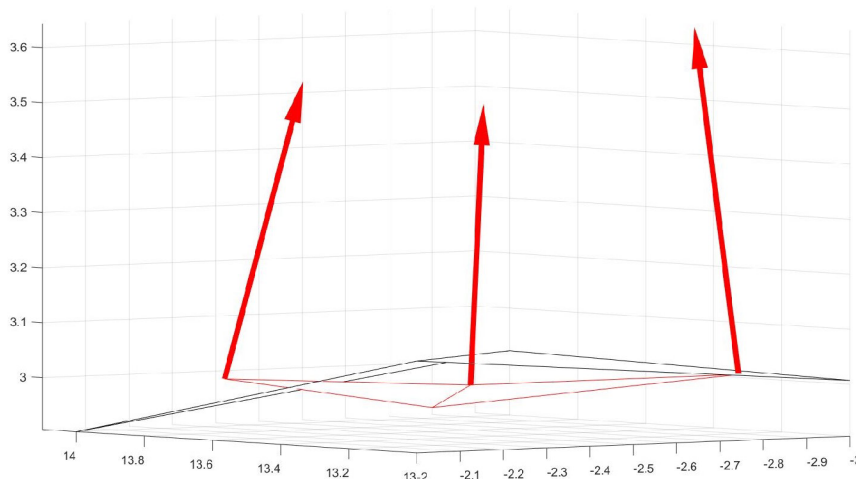
Obr. 2.6: Body pred posunutím.

V mieste, kde sa dva modely prekrývajú nastáva, že smernica bodu z jedného skenu pretína v určitom mieste druhý model (obr.2.7). Toto miesto nám určuje, kde by sa daný bod mal nachádzať v druhom skene. Ak teda priesečník smernice bodu z jedného modelu s povrchom druhého modelu neleží v mieste daného bodu, znamená to, že sa skenovaný objekt medzi snímaním jednotlivých modelov zmenil. Musíme teda bod prispôbiť tak, aby sme tieto nepresnosti kompenzovali.



Obr. 2.7: Priesečník smernice bodu s plochou druhého modelu.

Posunieme preto tento bod do polovice vzdialenosti medzi daným bodom a priesečníkom s druhým modelm (obr.2.8). Aplikujeme tento postup na všetky body oboch modelov. Získame tak nový model so spriemerovaným a vyhladeným povrchom so zachovaním informácie o pôvodných bodoch.



Obr. 2.8: Po posunutí bodov prvého z modelov.

### Určenie priesečníku

Povrch modelu sa skladá z množstva trojuholníkov, ktoré sú definované trojicami bodov modelu. Tieto trojuholníky sú vlastne výseky z rovín, ktoré definujú jednotlivé trojice. Keď chceme teda vypočítať priesečník smernice bodu s druhým modelom, musíme určiť priesečníky s týmito rovinami a následne zistiť, v ktorom prípade leží priesečník s rovinou vo vnútri daného trojuholníku, ktorý je súčasťou tejto roviny. To môžeme zistiť pomocou nasledujúcich rovníc.[14]

Priesečník  $P$  bodu  $X$ , pohybujúceho sa po jeho smernici, s rovinou definovanou tromi bodmi  $A$ ,  $B$  a  $C$  vypočítame tak, že si určíme parameter, ktorým musíme vynásobiť vektor  $\mathbf{V}$ , definovaný smernicou bodu  $X$  tak, že keď ho pripočítame k bodu, bude bod  $X$  ležať v definovanej rovine.

$$P = D + \mathbf{V} * n$$

$$D + \mathbf{V} * n = A + |AB| * m + |AC| * k$$

Rovnicu si môžeme upraviť na tvar

$$|AB| * m + |AC| * k - \mathbf{V} * n = |AD|$$

Budeme teda určovať celkom tri parametre. Každý bod v priestore a aj každý vektor sú popísané pomocou  $x$ ,  $y$  a  $z$  súradníc. Naše hľadané parametre sú pre tieto tri

súradnice rovnaké. Preto, aby sme zistili tieto tri neznáme, sa musí rovnica rozpísať na sústavu troch rovníc (jedna pre  $x$ , jedna pre  $y$  a jedna pre  $z$ ), z ktorých určíme jednotlivé neznáme.

$$(B_x - A_x) * m + (C_x - A_x) * k - \mathbf{V}_x * n = (D_x - A_x)$$

$$(B_y - A_y) * m + (C_y - A_y) * k - \mathbf{V}_y * n = (D_y - A_y)$$

$$(B_z - A_z) * m + (C_z - A_z) * k - \mathbf{V}_z * n = (D_z - A_z)$$

kde

- body  $A$ ,  $B$  a  $C$  sú vrcholmi jedného z trojuholníkov definovaných polom indexov bodov **indices** v modeloch
- bod  $D$  je jeden z bodov z listu **FeaturePointList** druhého modelu
- vektor  $\mathbf{V}$  má počiatok v bode  $D$  a jeho smer je definovaný smernicou tohto bodu
- index  $x$  v  $A_x$ ,  $B_x$ ,  $C_x$ ,  $D_x$  a  $\mathbf{V}_x$ ) a rovnako tak indexy  $y$  a  $z$ , značia veľkosť v danej súradnici.

Keď už máme vypočítaný priesečník, musíme zistiť či leží v trojuholníku vymedzenom bodmi  $A$ ,  $B$  a  $C$ . To vieme znova zistiť pomocou vektorov strán trojuholníka.

$$|\mathbf{AD}| = |\mathbf{AB}| * g + |\mathbf{AC}| * h$$

Znovu máme dva neznáme parametre, ktorými keď vynásobíme odpovedajúce vektory, tak dostaneme bod  $A$  na miesto bodu  $P$ . Musíme preto túto rovnicu znova rozpísať na sústavu rovníc.

$$-A_x = (B_x - A_x) * g + (C_x - A_x) * h - P_x$$

$$-A_y = (B_y - A_y) * g + (C_y - A_y) * h - P_y$$

$$-A_z = (B_z - A_z) * g + (C_z - A_z) * h - P_z$$

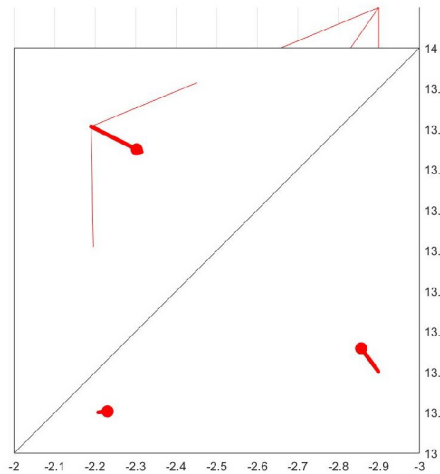
Aby ležal bod  $P$  v trojuholníku definovanom bodmi  $A$ ,  $B$  a  $C$ , musí byť

- $g + h \leq 1$
- $g \geq 0$
- $h \geq 0$

Pri niektorých zložitejších modeloch sa môže stať, že smernica bodu bude prechádzať viacerými trojuholníkmi. V tomto prípade zvolíme ako správny, ten ktorý je najbližšie a je orientovaný rovnakým smerom. To znamená že uhol medzi normálou trojuholníka a smernicou bodu musí byť  $\leq \pm 90^\circ$ .

### 2.1.4 Triangulácia

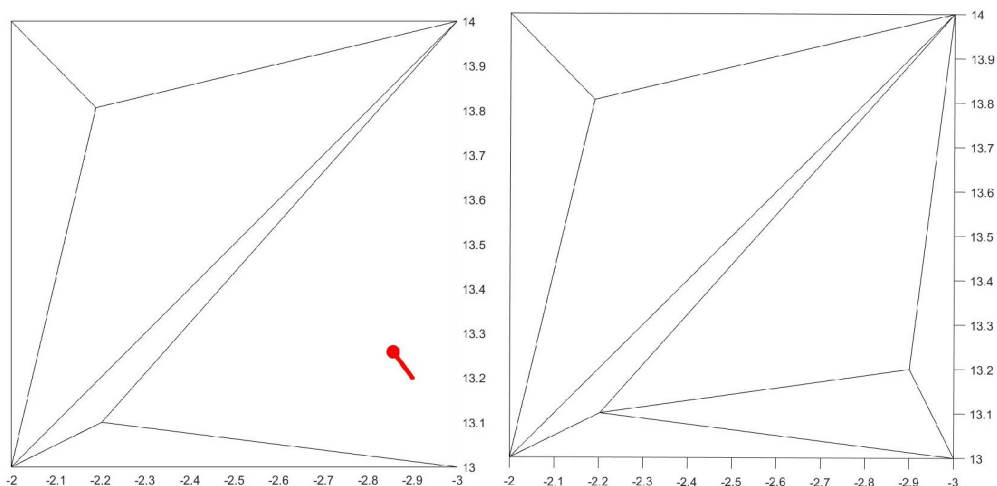
Aby sme čo najviac zachovali pôvodné čiastkové modely, nebudeme meniť trojuholníky, ktoré neležia v časti, ktorá sa prekrýva s druhým modelom. Zameriame sa tak iba na trojuholníky, v ktorých sme našli aspoň jeden priesečník so smernicou bodu druhého modelu alebo naopak na trojuholníky, z ktorých aspoň jeden bod tvoriaci vrchol má na svojej smernici aspoň jeden priesečník s pruchom prvého modelu. (obr2.9).



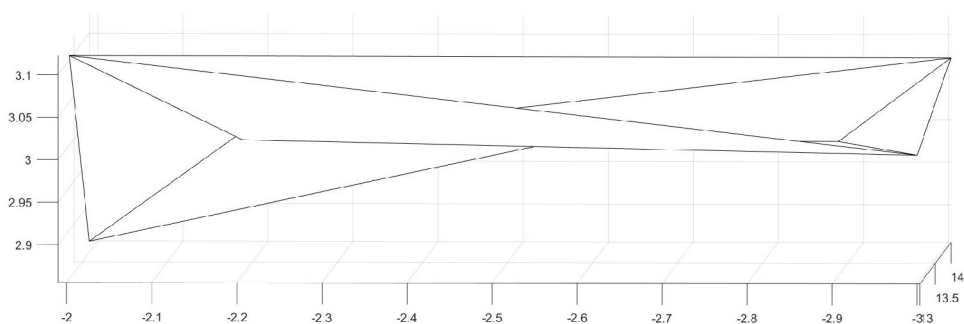
Obr. 2.9: Povrchy pred pretriangulovaním.

Aby sme vytvorili súvislý povrch, musíme jeden z povrchov zvoliť ako referenčný a všetky nájdené trojuholníky obsahujúce body podľa predchádzajúceho kroku v nerereferenčnom modeli odstrániť a body týchto trojuholníkov použijeme na upravenie povrchu referenčného. Na vytvorenie nového povrchu v mieste prekrytia použijeme bodov z oboch modelov dokopy upravíme referenčný povrch tak, že za pomoci informácií o priesečníkoch smerníc bodov nereferenčného modelu v prekrývajúcej sa časti s trojuholníkmi referenčného povrchu rozdelíme každý trojuholník, v ktorom leží priesečník na tri menšie trojuholníky s jedným spoločným vrcholom v bode priesečníku. Každý trojuholník referenčného povrchu, v ktorom leží priesečník so smernicou bodu, sa rozpadne na tri menšie trojuholníky a informácia o pôvodnom trojuholníku tak zanikne (obr2.10). Ak v jednom trojuholníku leží týchto priesečníkov viac, rozdelí sa trojuholník podľa jedného z bodov a potom sa zistí, v ktorom z novovzniknutých trojuholníkov z pôvodného trojuholníku leží priesečník so smernicou ďalšieho z týchto bodov a znova za rovnako tento menší trojuholník rozpadne na ďalšie tri trojuholníky.

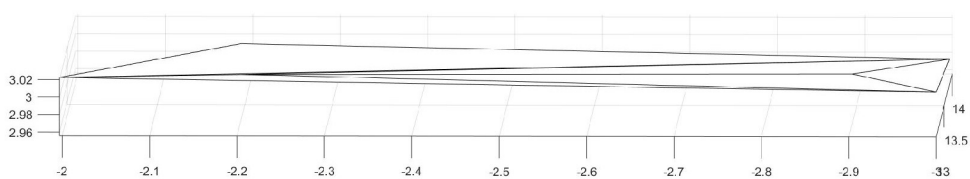




Obr. 2.10: Pretiangulovaný povrch.



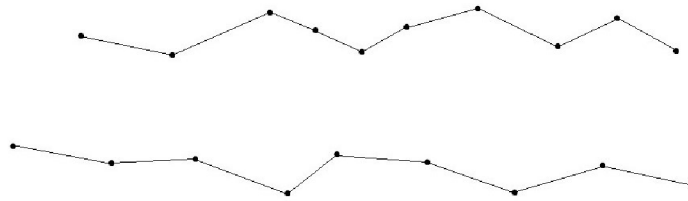
Obr. 2.11: Ukážka výsledného povrchu bez priemerovania bodov.



Obr. 2.12: Ukážka vyhladenia povrchu po priemerovaní.

Týmto postupom vznikne celistvý povrch až na miesto, kde končí model, ktorého trojuholníky sme použili ako referenčné, a začína druhý model. V tomto mieste nám vznikne úzky pás, kde nebudeme mať definované žiadne trojuholníky. Musíme teda ešte zaplniť tento pás. To urobíme tak, že si nájdeme body druhého modelu, ktoré boli zapojené v trojuholníkoch, ktoré ležali v prekrývajúcej sa časti, ale nie sú

zapojené v novovzniknutých trojuholníkoch. Tieto body tvoria jeden okraj nezaplnenej medzery. Body, ktoré tvoria druhý okraj medzery nájdeme podľa toho, že v novovzniknutých trojuholníkoch tvoria trojice s bodmi, ktoré tvorili trojice s bodmi, ktoré teraz tvoria prvý nájdený okraj. Nie však všetky takto nájdené body sú okrajové, ale určite sa okrajové nachádzajú v tejto množine bodov. Preto v nej nájdeme dvojice bodov, ktoré tvoria v spojenom modeli stenu iba jedného trojuholníka. A toto sú už naše hľadané body tvoriace okraj medzery, ktorých príklad môžeme vidieť na obr2.13.



Obr. 2.13: Lína bez definovaných trojuholníkov.

Nápomocné pri spájaní bude aj znalosť toho, že body tvorili trojuholník a teda vždy existuje pár bodov ktoré tvorili stenu. Vyberieme si teda jednu zo stien definovanú bodmi  $A$  a  $B$ , ktoré tvoria okraj, a nájdeme si k nemu najbližší bod z druhého okraja  $X$  podľa vstahu

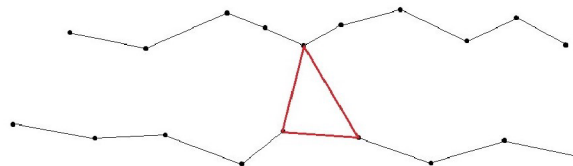
$$d = |AX| + |BX|$$

kde

- $d$  značí súčet vzdialeností bodov  $A$  a  $B$  od bodu  $X$

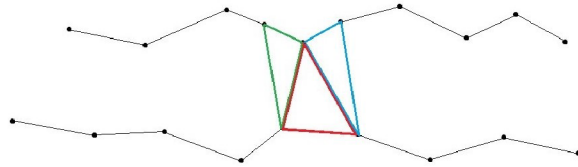
Náš hľadaný bod má súčet týchto vzdialeností najmenší.

Vytvoríme teda nový trojuholník z týchto troch bodov (obr2.14).



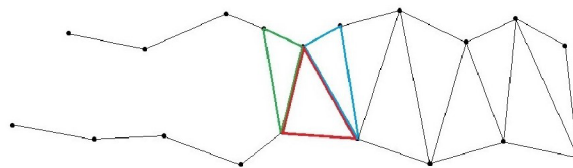
Obr. 2.14: Možnosť prvého trojuholníka.

Potom si nájdeme stenu trojuholníka, ktorá je tvorená jedným z bodov z oboch okrajov a vytvoríme ďalší trojuholník s jedným z dvoch bodov, ktoré tvoria stenu okraja s bodom  $X$ . Teda s tým, ktorý je bližšie k našej zvolenej stene (obr2.15).



Obr. 2.15: Možosti druhého trojuholníka.

Teraz už teda máme dva trojuholníky tvorené štyrmi bodmi (dvoma z každého okraja). Ďalšie trojuholníky sa už budú tvoriť jednoduchšie, lebo jedna zo stien posledného nájdeného bodu je už v trojuholníku použitá a teda nemusíme nič počítať a vieme, že bod, ktorý hľadáme tvorí s týmto bodom druhú stenu a má tvoriť ďalší trojuholník. Zoberieme si vždy stenu posledného vytvoreného trojuholníka, ktorá je tvorená bodom z oboch okrajov medzery a je stenou iba jedného trojuholníka a na striedačku budeme tvoriť trojuholníky s bodmi tvoriacimi stenu s bodmi tvoriacimi našu stenu, ku ktorej hľadáme trojuholník a ešte sme s nimi trojuholník nevytvorili (obr2.16).



Obr. 2.16: Celistvý povrch v predtým nezaplnenej línii.

Takto vyplníme celú líniu postupne z oboch strán prvého nájdeného trojuholníka a vznikne vám výsledný model dvoch spojených čiastkových modelov.

## 2.2 Implementácia riešenia

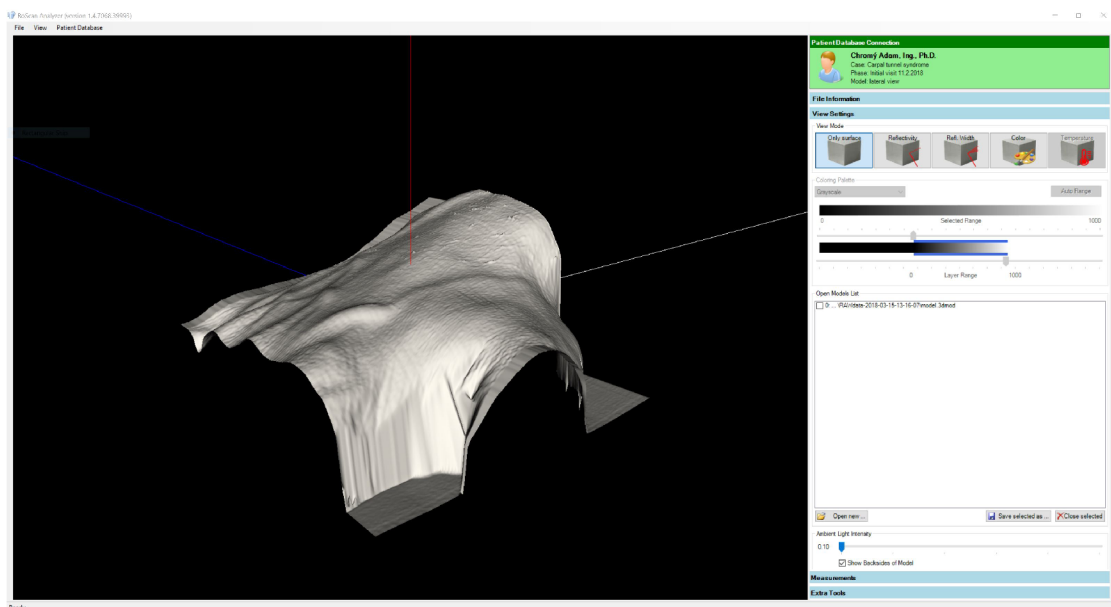
### 2.2.1 Prostredie projektu ModelViewer

Systém RoScan, na ktorom vedúci mojej práce pracuje už niekoľko rokov, sa skladá z viacerých dielčích projektov, ktoré spolupracujú. Moja práca, ktorá sa ktorá sa zaoberá spájaním dvoch čiastkových modelov do jedného celkového modelu, spadá do projektu ModelViewer. Tento projekt sa stará o zobrazovanie už naskenovaných modelov pre ich prezeranie. Projekt tiež obsahuje nástroje pre rôzne merania.

Na pravej strane obrazovky je umiestnená ovládací lišta, na ktorej sa nachádzajú štyri záložky- *File Information*, *View Settings*, *Measurements* a *Extra Tools*. Po kliknutí na jednu zo záložiek sa rozbalí ponuka s danými informáciami a nástrojmi. Na spodu obrazovky sa nahádza úzka lišta, ktorá informuje o aktuálnom stave projektu alebo o práve vykonávanom úkone. Celý zvyšok obrazovky je venovaný prehliadaču, v ktorom si užívateľ môže prezerať model.

Kvôli spájaniu modelov musel byť tento projekt upravený tak, aby bolo možné naraz otvoriť a zobráť viac ako jeden model. Dôvodom bolo to, aby mohol užívateľ čo najpresnejšie zvoliť odpovedajúce si body na oboch modeloch, ktoré sa budú dopasovávať.

Mojou úlohou bolo tento projekt rozšíriť o už spomínanú možnosť dopasovania a spájania dvoch modelov do jedného celkového modelu podľa popísaného algoritmu. Prostredie projektu ModelViewer môžete vidieť na nasledujúcom obrázku obr2.17.



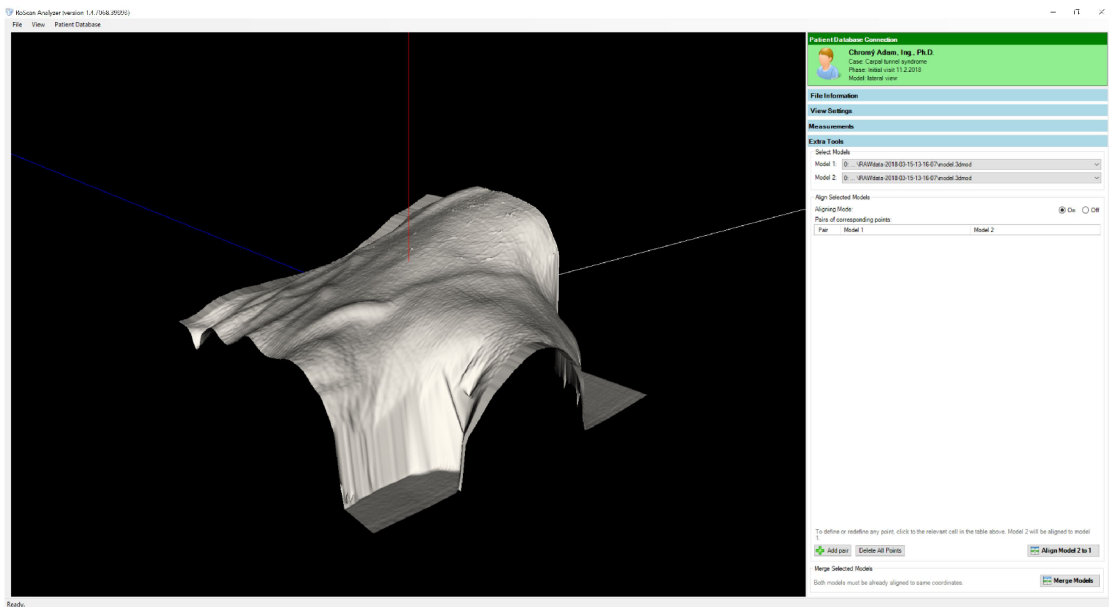
Obr. 2.17: Uživatelské prostredie projektu ModelViewer - záložka ViewSettings.

Pri spájaní modelov sa užívateľ stretne so záložkami *View Settings* a *Extra Tools*. Hlavnou záložkou pre moju prácu, v ktorej sa skrývajú nástroje pre dopasovanie a spájanie, je práve záložka *Extra Tools*.

### Záložka Extra Tools

Po rozkliknutí záložky *Extra Tools* sa rozbalí ponuka nástrojov, ktorá slúži na dopasovanie čiastkových modelov a ich spájanie do celkového modelu. Ponuka má tri časti- *Select Models*, *Align Selected Models* a *Merge Selected Models*. Záložku *Extra*

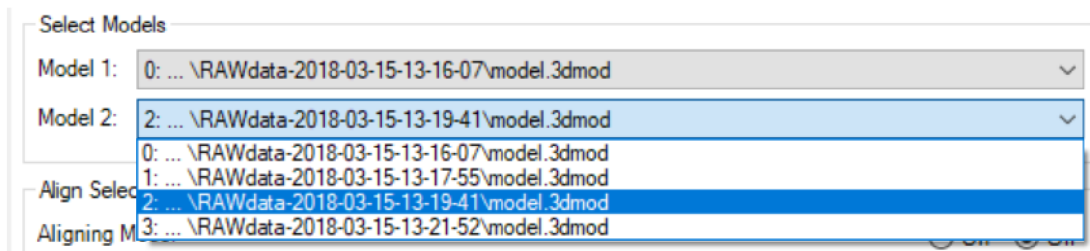
Tools môžete vidieť na obrázku obr2.18. V tejto časti bude popísané, na čo slúžia jednotlivé časti a ako sa používajú.



Obr. 2.18: Užívateľské prostredie projektu ModelViewer - záložka ExtraTools.

## Select Models

Táto časť, ktorá sa nachádza v hornej časti záložky *Extra Tools* obsahuje dve polia (obr2.19). Po kliknutí na jedno z polí sa rozbalí ponuka práve otvorených modelov v projekte ModelViewer. Na každom riadku výberového okna je cesta k otvorenému modelu spolu s jeho názvom. Obsah týchto polí je dynamický a mení sa automaticky vždy, keď užívateľ v záložke *View Settings* otvorí alebo zatvorí nejaký model.



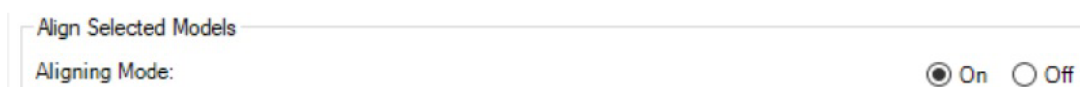
Obr. 2.19: Výberové polia v časti Select Models.

Polia sú dôležité pretože v projekte môže byť otvorených viac ako dva modely naraz. Preto tieto polia slúžia na definovanie práve dvoch modelov, ktoré sa budú doposovávať alebo spájať do jedného.

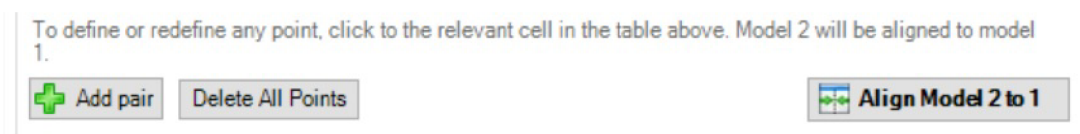
## Align Selected Models

Časť *Align Selected Models* slúži k dopasovávaniu modelov v priestore, ak zvoleným dvom modelom, ktoré sa budú spájať, nepasujú absolútne súradnice.

V pravom hornom rohu tejto časti sa nachádza voľba On/Off (obr2.20), ktorou zapíname alebo vypneme možnosť výberu odpovedajúcich si bodov tlačidlom *Add pair* na zvolených čiastkových modeloch a tiež sprístupňujeme tlačidlo na dopasovanie modelov *Align Model 2 to 1* na základe zvolených bodov. Tieto tlačidlá, spolu s posledným tlačidlom *Delete All Points*, sa nachádzajú na spodu časti *Align Selected Models* a vidieť ich môžeme na obr2.21. Aby bolo tlačidlo *Align Model 2 to 1* aktívne a dopasovanie bolo možné vykonať, je potrebné zvoliť minimálne štyri páry odpovedajúcich si bodov.



Obr. 2.20: Zapnutie-vypnutie časti Align Selected Models.



Obr. 2.21: Sprístupnené tlačidlá v časti Align Selected Models.

Odpovedajúce si body modelov sa dajú voliť po jednotlivých pároch tak, že po kliknutí na tlačidlo *Add pair* sa v tabuľke *Pairs of corresponding points* objaví nový prázdny riadok (obr2.22). Tabuľka obsahuje tri stĺpce. Prvý stĺpec obsahuje poradové číslo zvoleného páru odpovedajúcich si bodov. Druhý a tretí stĺpec obsahujú súradnice zvolených bodov.

Pairs of corresponding points:		
Pair	Model 1	Model 2
1		

Obr. 2.22: Tabuľka zvolených odpvedajúcich si bodov.

Tabuľka odpovedajúcich si bodov sa plní tak, že ak chceme myšou voliť bod na prvom zvolenom modeli, klikneme na dané políčko v tabuľke *Pairs of corresponding*

*points*. Zvolené políčko zmodrá a je pripravené na výber bodu daného modelu. Výber bodu sa takisto prevádza kliknutím myši na vybrané miesto modelu. Tabuľku s prvým zvoleným bodom ukazuje obrázok obr2.23.

Pairs of corresponding points:		
Pair	Model 1	Model 2
1	{X:-37.00019 Y:506.1505 Z:61.4161}	

Obr. 2.23: Prvý zvolený bod v tabuľke.

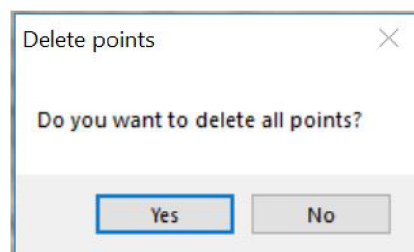
Po zvolení bodu na oboch modeloch je možné pridať ďalší riadok tabuľky kliknutím na tlačidlo *Add pair*, ako ukazuje obrázok obr2.24.

Pairs of corresponding points:		
Pair	Model 1	Model 2
1	{X:-37.00019 Y:506.1505 Z:61.4161}	{X:-47.00002 Y:486.45 Z:16.64999}
2		

Obr. 2.24: Prvý zvolený pár bodov v tabuľke.

Keď je v tabuľke vybraných viac ako jeden bod, je možné upraviť ktorýkoľvek bod jednoducho tak, že znovu klikneme na dané miesto v tabuľke a zvolíme kliknutím myši na model nový bod. V prípade, že chceme body odstrániť, môžeme tak urobiť kliknutím na tlačidlo *Delete All Points*. Toto tlačidlo vymaže všetky odpovedajúce si body modelov a je možné zvoliť iný pár modelov na dopasovanie v časti *Select Models* a vyberať odpovedajúce si body novozvolených modelov.

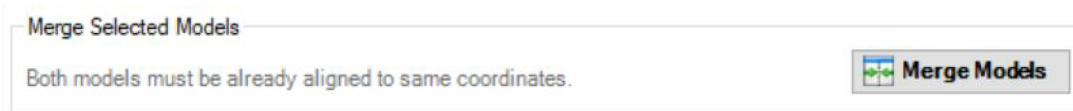
Aby sa predišlo nechcenému zmazaniu celej tabuľky odpovedajúcich si bodov, sa po kliknutí na toto tlačidlo zobrazí varovná hláška, ktorá nám dáva na výber, či chceme body naozaj zmazať alebo ich chceme zanechať (obr.2.25).



Obr. 2.25: Okno na potvrdenie zmazania odpovedajúcich si bodov v tabuľke.

## Merge Selected Models

Poslednou časťou záložky *Extra Tools* je *Merge Selected Models*, ktorá sa nachádza úplne na spodu tejto záložky. Ako je vidieť na obrázku obr2.26, táto časť obsahuje iba jedno tlačidlo.



Obr. 2.26: Časť Merge Selected Models v záložke Extra Tools.

Stlačením tohto tlačidla sa spustí samotný algoritmus priemerovania povrchu a spájania daných dvoch modelov zvolených v časti *Select Models*.

Táto časť práce sa zaoberala popisovaním a vysvetlovaním vizuálnej stránky projektu, s ktorou sa stretne užívateľ. Tento vzhľad som však v rámci mojej bakalárskej práce vôbec netvoril ani neupravoval. V rámci mojej bakalárskej práce som len už vytvoreným tlačidlám pridal funkčnosť a preto si v ďalšej časti tejto práce vysvetlíme, čo všetko sa za týmito tlačidlami a tabuľkou na dopasovávanie a spájanie modelov skrýva a ako to všetko funguje na programovej úrovni.

### 2.2.2 Programové riešenie

Všetky projekty systému RoScan sú písané v jazyku C#. Vráťane projektu *ModelViewer*, ktorý som v rámci bakalárskej práce rozširoval. Preto som pochopiteľne musel implementovať algoritmus dopasovania a spájanie dvoch čiastkových modelov, popísaný v teoretickej časti, takisto v jazyku C#, pretože využíva mnohé funkcie a premenné ostatných tried tohto projektu.

Ako bolo zmienené v úvode práce, pri spájaní dvoch čiastkových modelov do jedného celkového modelu môžu nastať dva prípady. Buď boli modely skenované spôsobom, kedy im pasujú absolútne súradnice a ich dopasovanie nie je potrebné, alebo boli skenované spôsobom, kedy na seba v priestore nenadväzujú a je pred ich spájaním potrebné jeden z modelov vhodne otočiť a posunúť, aby sme pasovanie absolútnych súradníc zaručili. To teda znamená, že algoritmus je pre oba prípady rovnaký až na výnimku, že jeden z postupov potrebuje na začiatku krok navyše. Preto budem postupovať rovnako ako v teoretickej časti a najskôr vysvetlím tento krok, kedy sa modely s nepasujúcimi absolútnymi súradnicami dopasujú.



## Dopasovávanie modelov

Hneď v úvode by som chcel povedal, že časť kódu, ktorá sa stará o spojenie čiastkových modelov som nepísal sám. Pri prehľadávaní zdrojov na internete som narazil na webovej stránke GitHub.com na projekt s názvom *Transform3DBestFit*, ktorý sa venoval práve tomuto problému.[15] Zhodou náhod bol ten projekt písaný priamo v jazyku C# a po preštudovaní jeho implementácie som zistil, že dokonca funguje na rovnakom princípe ako som popísal v teoretickej časti tejto práce. A keďže som zistil, že tento projekt je chránený licenciou MIT (<https://opensource.org/licenses/MIT>), rozhodol som sa tento projekt otestovať a po malej úprave oboch projektov som ho implementoval do projektu *ModelViewer*. Aby tento projekt fungoval, musel som do projektu *ModelViewer* pridať aj OpenSource matematickú knižnicu tvorenú pre C# s názvom *Math.NET Numerics* (<https://numerics.mathdotnet.com/>), ktorá je takisto chránená MIT licenciou.

Výsledky, ktoré boli výstupom projektu *Transform3DBestFit*, boli veľmi presné a hlavne tento projekt pracoval veľmi rýchlo. Preto som sa rozhodol ho použiť do finálnej verzie mojej práce, samozrejme so všetkými podmienkami, ktoré požaduje licencia MIT.

Samozrejme okrem tohto algoritmu je potrebná na spojenie čiastkových modelov aj informácia o odpovedajúcich si bodoch, aby mal spomínaný algoritmus čo počítať. V minulej kapitole bolo vysvetlené, ako sa odpovedajúce si body modelov vyberajú a kam sa ukladajú v užívateľskom rozhraní. Teraz bude vysvetlené, čo sa skrýva za tabuľkou odpovedajúcich si bodov *Pairs of corresponding points* a tlačidlami *Add pair* a *Align Model 2 to 1*.

### Tlačidlo *Add pair* a tabuľka *Pairs of corresponding points*

Na začiatku teoretickej časti práce bolo spomenuté, že systém RoScan pracuje s modelmi triedy **RoScanModel**. Rovnako bolo napísané, že z tejto triedy sú pre dopasovávanie a spájanie modelov dôležité hlavne

- list všetkých bodov čiastkového modelu **points**
- list polôh odpovedajúcich si bodov **FeaturePointList**
- pole celočíselných hodnôt **indices**, reprezentujúce indexy bodov v liste points, kde vždy trojica indexov popisuje trojuholník tvoriaci povrch modelu

Vstupné dáta boli podrobnejšie podrobnejšie popísané v odstavci 1.1 s názvom *Vstupné dáta*.

**FeaturePointList** je teda miesto, kde sa ukladajú vybrané odpovedajúce si body a tento list je zdrojom pre plnenie tabuľky odpovedajúcich si bodov *Pairs of corresponding points*.

Po stlačení tlačidla *Add pair* sa vykoná funkcia *butAddPair\_Click*, v ktorej sa overia dve podmienky, aby sa zistilo, či je možné pridať do tabuľky nový riadok. Prvou podmienkou je, že vo výberových poliach v časti *Select Models* sú zvolené rozdielne modely. Druhá podmienka je, že aktuálny počet riadkov tabuľky je zhodný s počtom bodov v listoch **FeaturePointList** oboch modelov. To znamená, že páry bodov sa musia voliť postupne po jednom a nový riadok je možné pridať až po zvolení predchádzajúceho páru. Avšak pridanie riadku do tabuľky neovplivní dĺžku listov **FeaturePointList**. Do týchto listov sa body pridávajú až v momente, keď ich užívateľ zvolí.

Dôvod pre volenie bodov po jednom je ten, že body sa do listu **FeaturePointList** pridávajú na index, ktorý je zvolený kliknutím na riadok v tabuľke *Pairs of corresponding points*. Ak chce teda užívateľ prepísať už vybratý bod v liste odpovedajúcich si bodov, jednoducho klikne na príslušný riadok v tabuľke *Pairs of corresponding points* a nový zvolený bod prepíše pôvodný. A ak sa zapisuje do nového prázdneho riadku, to znamená, že počet riadkov tabuľky odpovedajúcich si bodov je o jedna väčší ako počet polôh bodov v listoch *FeaturePointList*, rozšíri sa tento list odpovedajúceho modelu, teda toho modelu, v ktorom stĺpci sa volí bod, o jeden a zvolený bod sa zapíše na toto nové miesto.

Výber bodov modelu kliknutím myši bol už v projekte implementovaný, kvôli možnosti merania. Preto som využil príslušné funkcie projektu *LeftMouseClicked* a *NearestPointToRay*.

Keď teda užívateľ zvolí miesto v tabuľke odpovedajúcich si bodov a klikne na model zobrazený v prehladači modelov, spustí sa funkcia *LeftMouseClicked*, ktorá overí že v tabuľke *Pairs of corresponding points* je aspoň jeden riadok. Ak je podmienka vyhodnotená ako pravdivá, zavolá táto funkcia funkciu *LeftMouseClickedAlign* príslušného modelu, na ktorý užívateľ klikol. Táto funkcia ako prvé zavolá funkciu *NearestPointToRay*, ktorá vráti index najbližšieho bodu ku kurzoru myši v liste všetkých bodov čiarkového modelu **points**, alebo hodnotu *-1*, ak užívateľ klikol mimo modelu. Potom funkcia *LeftMouseClickedAlign* overí, či index modelu, na ktorý užívateľ klikol je zhodný s indexom stĺpca, ktorý obsahuje jeho hodnoty bodov z listu *FeaturePointList* a že funkcia *NearestPointToRay* vrátila nezápornú hodnotu. Ak áno, rozhodne, či ide o zapísanie nového bodu do listu, alebo o prepísanie už existujúceho.

## Tlačidlo **Align Model 2 to 1**

Stlačením tlačidla *Align Model 2 to 1* sa spustí funkcia *butAlign\_Click*, ktorá overí, či boli zvolené aspoň štyri páry odpovedajúcich si bodov a že počet polôh bodov v listoch *FeaturePointList* oboch modelov je rovnaký.

Ak je podmienka pravdivá, predajú sa triede **ModelMerging** oba čiastkové

modely a zavolá sa funkcia tejto triedy s názvom *AlignModels*. Tá pripraví všetky potrebné matice pre výpočet rotačnej a translačnej matice postupom singulárneho rozkladu ako bolo vysvetlená v teoretickej časti práce, zavolá funkciu *CalcTransform* spomínaného projektu *Transform3DBestFit* a podľa získaných transformačných matíc upraví čiastkový model.

## Spájanie modelov

Či už boli modely skenované spôsobom, kedy im od začiatku pasovali absolútne súradnice, alebo boli po naskenovaní dopasované, postup spájania dvoch čiastkových modelov do celkového modelu je pre všetky prípady rovnaký. Všetko, čo sa stane v programe po kliknutí na tlačidlo *Merge Models*, bude vysvetlené v tejto časti práce.

## Nájdenie prekrývajúcich sa častí modelov

Po stlačení tlačidla *Merge Models*, sa spúští funkcia *butMerge\_Click*, ktorej úlohou je predať triede **ModelMerging** dva modely, ktorým pasujú absolútne súradnice a zavolať funkciu tejto triedy s názvom *MergeModels*. Táto funkcia spracuje oba vstupné modely a vráti ako výstup nový model, ktorý je kombináciou dvoch čiastkových vstupných.

Funkcia *MergeModels* ako prvé zistí, ktoré body modelov ležia v časti, ktorou sa predané dva modely prekrývajú. Na to využíva už predtým implementovanú funkciu *RayTriangleIntersection*, ktorá podľa troch bodov definujúcich trojuholník v priestore a smernice bodu zistí, či daná smernica bodu pretína plochu trojuholníka a ak áno, vráti vzdialenosť od bodu, ktorého smernicu testovala, k bodu priesečníku smernice bodu s plochou trojuholníka.

Pôvodne to funkcia *MergeModels* určovala tak, že prechádzala bod za bodom a pre každý bod zisťovala, či jeho smernica pretína nejaký trojuholník povrchu druhého modelu. Tento postup bol však výpočetne a časovo veľmi náročný, lebo modely, ktoré skenuje systém RoScan môžu mať aj niekoľko stotisíc bodov. Testovacie modely, ktoré som používal mali cez 250 000 bodov a ich povrch tvorilo cez 500 000 trojuholníkov, čo znamená, že funkcia *RayTriangleIntersection* sa pre jeden model volala viac ako 125 miliárd krát, čo by na mojom notebooku odhadom trvalo možno až niekoľko desiatok hodín.

Preto som po konzultácii s vedúcim práce využil už implementovaný algoritmus **Octree**, ktorý rozdelí povrch na menšie časti, ktoré obsahujú menej trojuholníkov povrchu a výpočet sa tým výrazne zrýchli. [16] Vďaka tomuto postupu sa doba výpočtu prekrývajúcej sa časti jedného testovacieho modelu skrátila na približne minútu.

## Priemerovanie a spájanie prekrývajúcich sa častí modelov

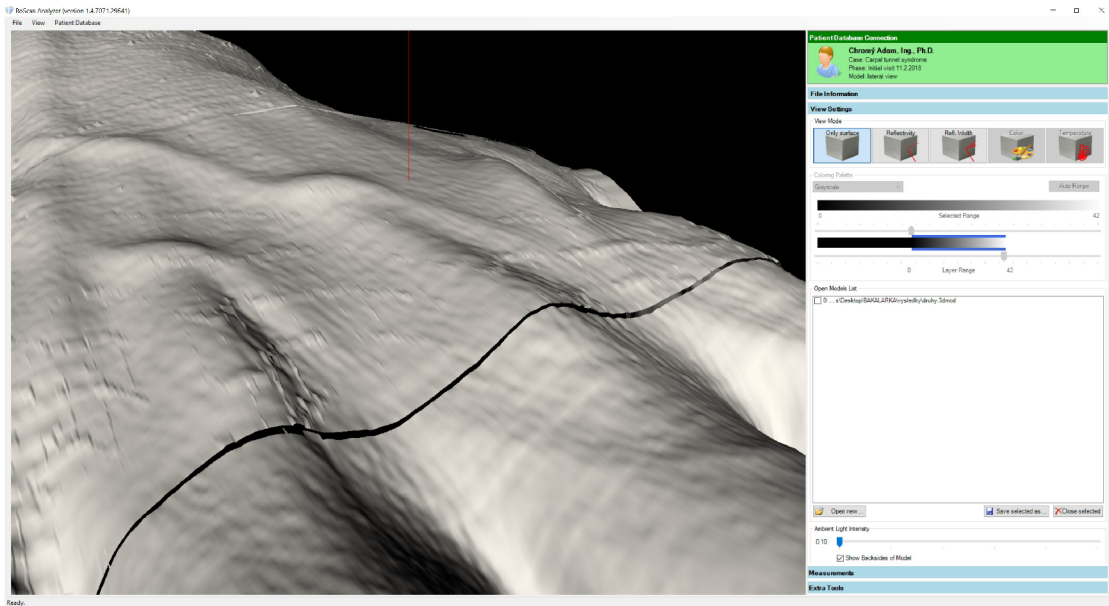
Pri hľadaní priesečníkov smeníc bodov s trojuholníkmi povrchu druhého modelu, algoritmus ukladá štruktúru obsahujúcu index bodu, index trojuholníka, ktorý jeho smernica pretína a vzdialenosť od daného bodu k priesečníku do listov **overlap1** a **overlap2**. List **overlap1** obsahuje indexy bodov druhého modelu a indexy trojuholníkov prvého modelu, teda trojuholníky povrchu, ktoré budú referenčné. List **overlap2** obsahuje indexy bodov prvého modelu a indexy trojuholníkov druhého modelu, teda trojuholníky ktoré sa musia odstrániť, aby sa vyriešil problém s dvojitém povrchom modelu v prekrývajúcej sa časti modelov. Po tom ako, algoritmus prejde všetky body oboch modelov, využije informácie z týchto listov a posunie všetky body, ktorých smernica sa pretína s povrchom druhého modelu o polovicu zistenej vzdialenosti. Následne vymaže všetky trojuholníky povrchu druhého modelu v prekrývajúcej sa časti, teda trojuholníky na indexoch trojuholníkov z listu *overlap2*. Aby bol tento krok poistený, nemaže algoritmus iba trojuholníky, ktoré pretína smernica nejakého z bodov referenčného modelu, ale aj všetky trojuholníky, ktorých aspoň jedna smernica vrcholového bodu pretína povrch referenčného modelu.

Následne sa list bodov *points* nového modelu triedy **RoScanModel** naplní všetkými bodmi referenčného modelu a pridajú sa aj všetky body druhého modelu, čo znamená, že ak ich bude chcieť algoritmus indexovať v tomto novom liste, bude musieť počítať s tým, že ich index sa zvýšil o počet bodov referenčného modelu. Preto sa upraví indexy bodov druhého modelu v liste *overlap1* tak, že sa k nim pripočíta počet bodov referenčného modelu. Tak isto sa upraví aj indexy bodov definujúce trojuholníky nereferenčného modelu, ktoré neležia v prekrývajúcej sa časti modelov a pridajú sa k trojuholníkom referenčného modelu.

Posledný krok je pretrojuholníkovanie povrchu v prekrývajúcej sa časti modelov podľa informácií v liste *overlap1*. To sa robí presne ako bolo popísané v teoretickej časti práce. Každý trojuholník, ktorý pretína smernica bodu nereferenčného modelu sa rozpadne na tri menšie trojuholníky s jedným spoločným vrcholom v bode, ktorého smernica pretína tento trojuholník. Ak jeden trojuholník pretínalo viac bodov, zistí sa, v ktorom z novovzniknutých trojuholníkov leží priesečník s ďalším bodom a zvoja sa tento nový trojuholník rozpadne na tri menšie trojuholníky.

## Vyplnenie úzkej medzery medzi spojenými čiastkovými modelmi

Po spriemerovaní a spojení dvoch čiastkových modelov zostane na konci povrchu referenčného modelu, v mieste kde začína časť povrchu tvorená neprekrývajúcou sa časťou povrchu nereferenčného modelu, úzka medzera medzi týmito časťami, ktorá sa tiahne cez celý model (obr2.27).



Obr. 2.27: Medzera medzi spojenými čiastkovými modelmi.

Žiaľ popísaný postup v teoretickej časti na vyplnenie tejto medzery zlyhal. Po implementovaní tohto postupu a snahe indentifikovať okrajové body som zistil, že tento postup na testovacích modeloch našiel iba 118 okrajových bodov, čo je veľmi málo, pretože testovací model má na šírku 1280 bodov, čo znamená, že okrajových bodov musí byť omnoho viac.

Otestoval som ho aj na iných modeloch, no výsledok bol vždy podobný a z dôvodu nedostatku času som tento postup nestihol premyslieť a implementovať do prjektu. Takže výstupom mojej práce sú spojené čiastkové modely, ktoré žiaľ obsahujú túto medzeru.

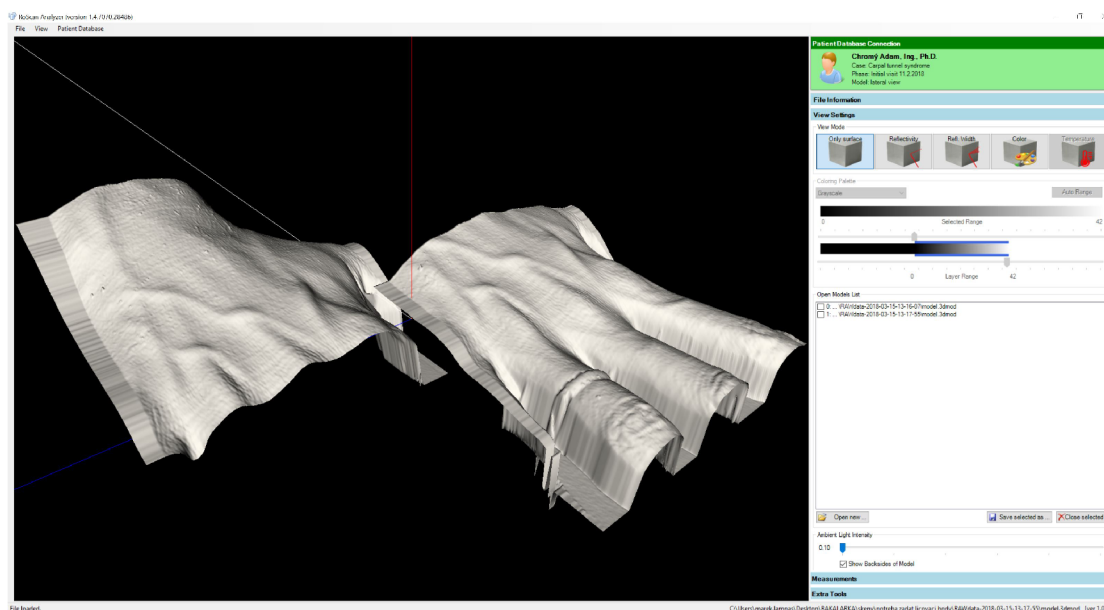
V tomto vidím najväčší nedostatok mojej práce a v budúcnosti projektu **ModelViewer** vnímam potrebu doplniť tento krok.

### 2.2.3 Postup spájania čiastkových modelov

V poslednej časti tejto práce by som rád uviedol príklad použitia projektu **ModelViewer** s implementovaným algoritmom na dopasovávanie a spájanie dvoch čiastkových modelov. Pretože chcem predviesť celý postup, použijem modely, ktorým na začiatku nepasujú absolútne súradnice.

Užívateľ v projekte **ModelViewer** v záložke *View Settings* otvorí modely, ktoré chce spájať. Všetky aktuálne otvorené modely sa zobrazia v prehliadači modelov a cesty k ich umiestneniu spolu s ich názvami sa zobrazia na pravej strane v okne *Open Models List*. Na začiatku každého riadku s cestou k modelu je aj jeho index

otvoreného modelu. Toto číslo predstavuje poradové číslo otvoreného modelu a pripisuje sa modelom automaticky po otvorení. Otvorené dva modely s nepasujúcimi súradnicami môžeme vidieť na obrázku obr2.28.



Obr. 2.28: Príklad vstupných čiastkových modelov s nepasujúcimi absolútnymi súradnicami.

Po otvorení modelov sa užívateľ prepne do záložky *Extra Tools*, v ktorej sa nachádzajú nástroje potrebné na dopasovanie a spojenie čiastkových modelov.

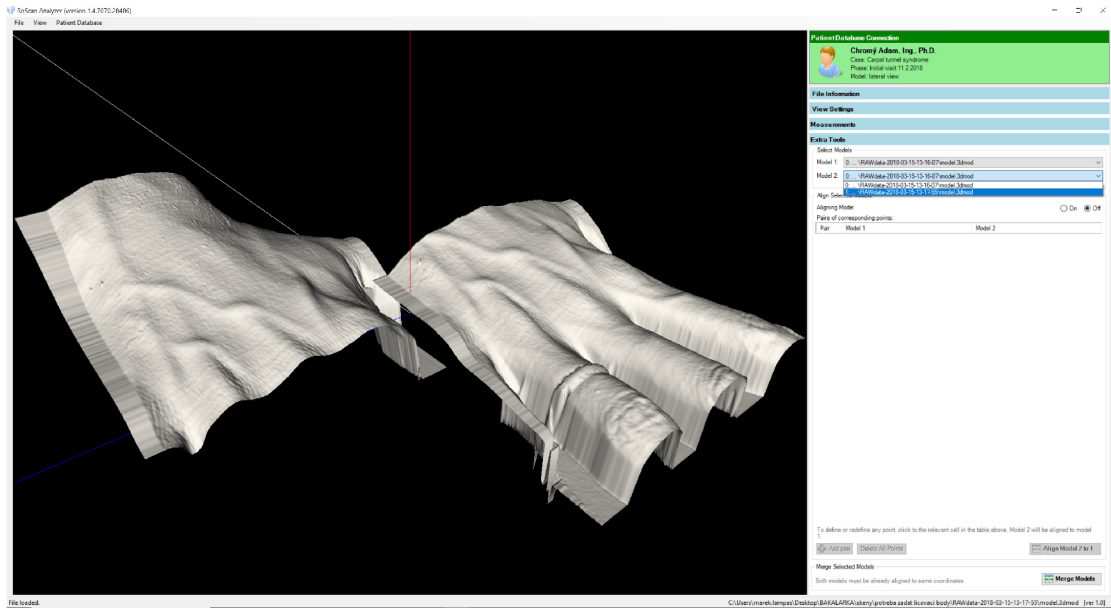
V hornej časti s názvom *Select Models*, zvolí užívateľ z ponuky aktuálne otvorených modelov dva, ktoré sa budú dopasovávať alebo spájať. Algoritmus funguje tak, že sa upravuje druhý zvolený model, teda model 2. Spôsob výberu modelov, kedy chceme dopasovať a potom pripojiť model s indexom 1 k modelu s indexom 0, môžeme vidieť na obrázku obr2.29.

Po výbere modelov je užívateľ schopný prepnúť vypínač časti *Align Selected Models* do stavu *On*, čím sa sprístupnia všetky tlačidlá v tejto časti.

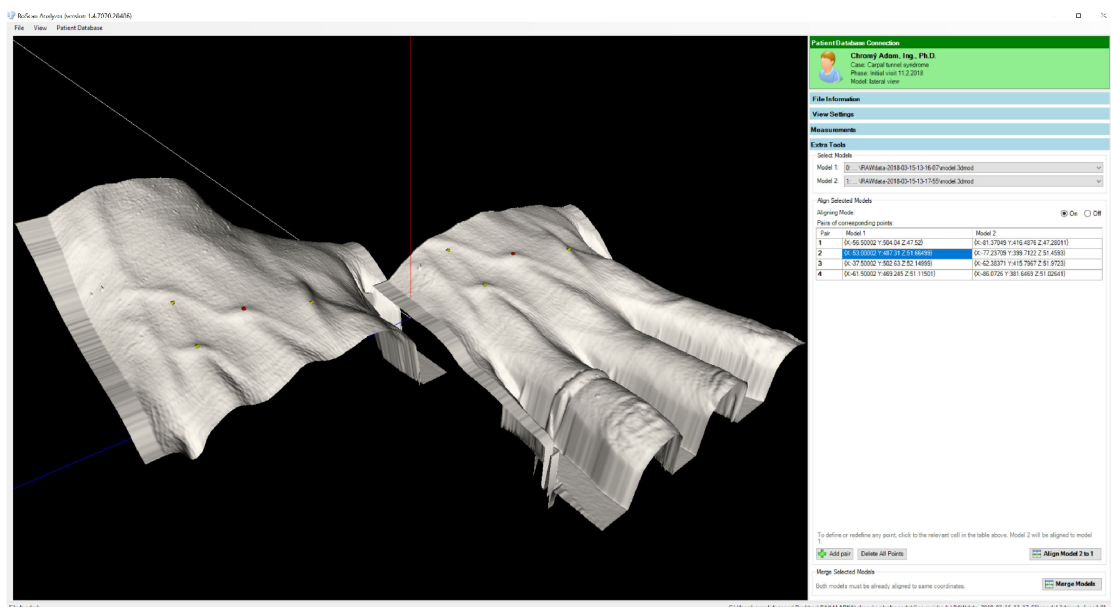
Tlačidlom *Add pair* sa pridá do tabuľky *Pairs of corresponding points* prvý riadok a užívateľ môže zvoliť prvý pár odpovedajúcich si bodov. Podrobnejší popis ako to urobiť je v kapitole 2.1 s názvom *Prostredie projektu ModelViewer*.

Tabuľku s vybranými štyrmi párami odpovedajúcich si bodov zobrazuje obrázok obr2.30.

Na tomto obrázku je taktiež možno vidieť, ako sú zvolené body vyznačené v prehliadači modelov. Aktuálne je v tabuľke označený druhý riadok, takže druhý zvolený pár bodov je v prehliadači vyznačený červenou farbou, a ostatné páry odpovedajúcich si bodov sú vyznačené v prehliadači modelov žltou farbou.



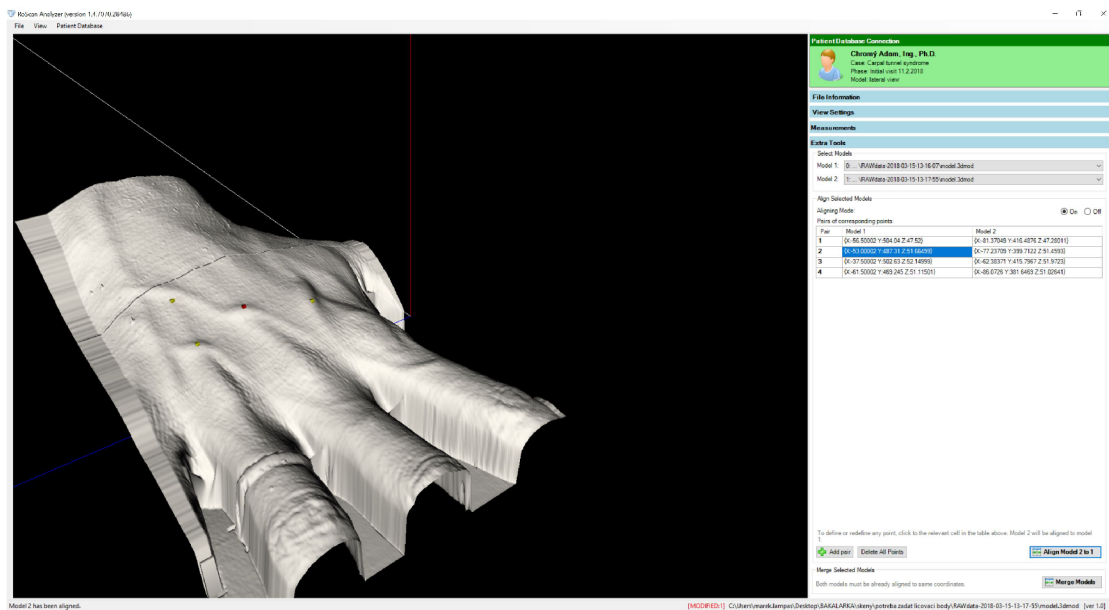
Obr. 2.29: Príklad výberu modelov v Extra tools na dopasovanie.



Obr. 2.30: Príklad modelov so zvolenými štyrmi párami odpevďajúcich si bodov.

Po zvolení minimálne štyroch párov odpovedajúcich si bodov sa stane aktívne aj tlačidlo *Align Model 2 to 1*. Zvoliť sa však môže aj viac bodov a keďže sa týmto postupom body s veľkou pravdepodobnosťou nezvolia úplne presne, je vhodné zvoliť čo najviac bodov, aby sa chyba výberu párov normalizovala. Stlačením tlačidla *Align Model 2 to 1* sa model, ktorý je zvolený ako model 2, dopasuje na základe zvolených

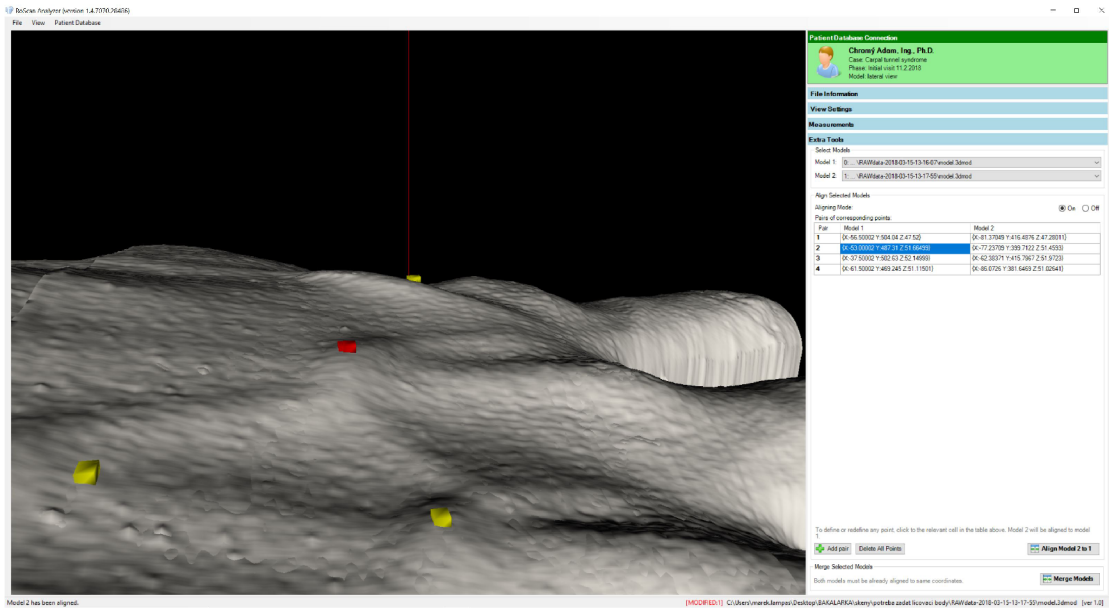
bodov. Takto dopasované modely môžeme vidieť na obrázku obr2.31.



Obr. 2.31: Dopasované modely na základe odpevedajúcich si bodov.

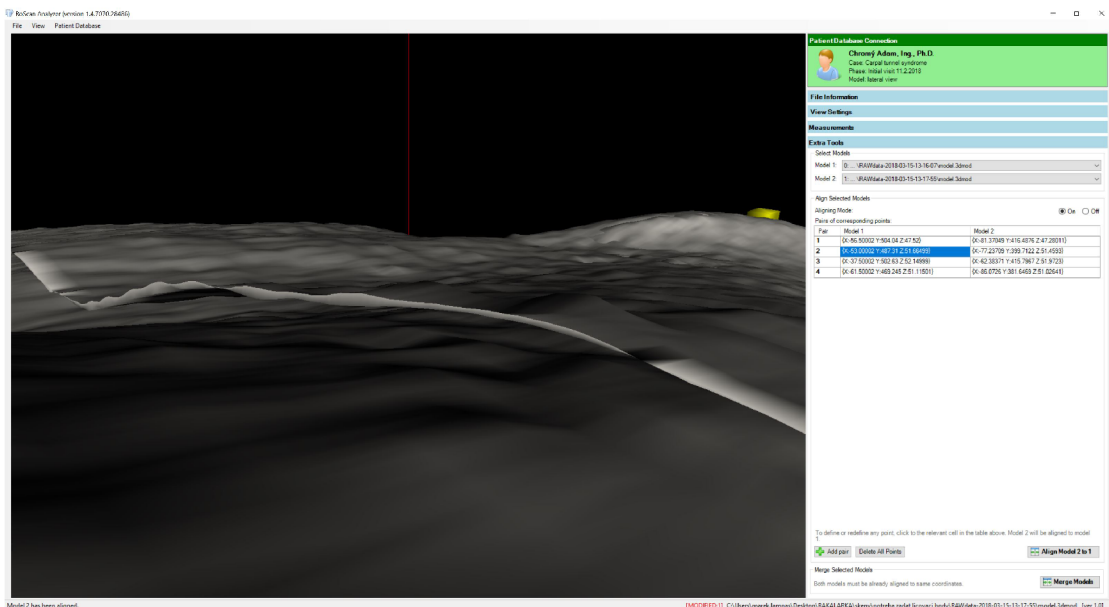
Keď sa pozrieme bližšie na zvolené páry odpovedajúcich si bodov v prehliadači modelov, ako ukazuje obrázok obr2.32, môžeme vidieť ako presne sa zvolené dva čiastkové modely dopasovali. Lenže ako bolo spomínané v úvode práce, povrchy dvoch čiastkových modelov nie sú totožné, z dôvodu, že sa neskenujú pevné telesá. Preto sa skenovaný objekt medzi dvoma skenovaniami môže trochu zmeniť. Na obrázku obr2.33 a na obrázku obr2.34 môžeme vidieť nepresnosti dopasovania dvoch povrchov z dôvodu zmeny povrchu medzi vyhotovením modelov.



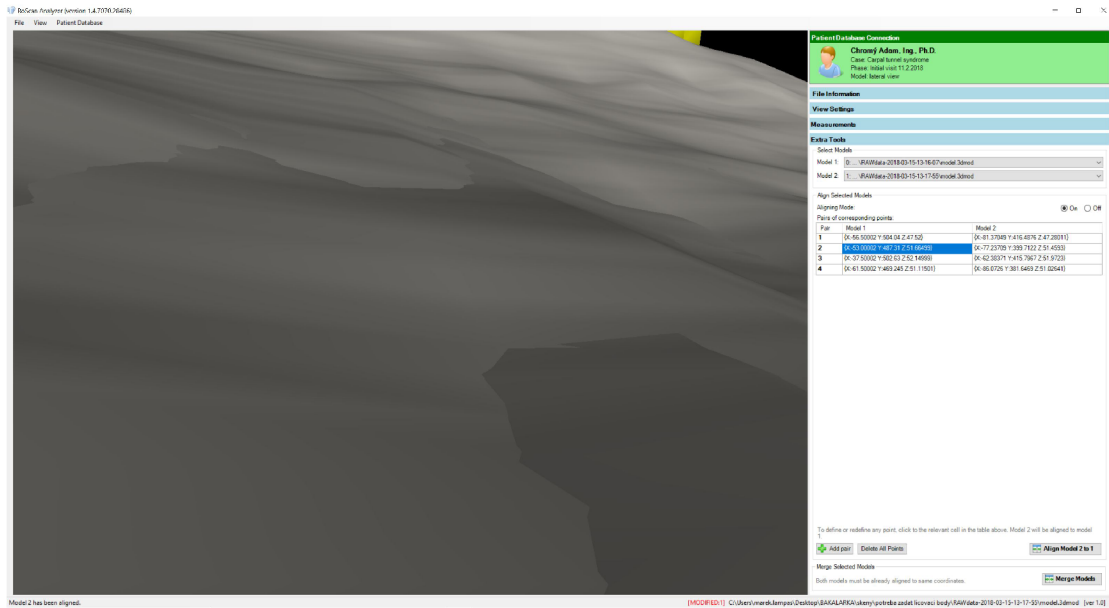


Obr. 2.32: Odpovedajúce si páry bodov na dopasovaných čiastkových modeloch.

Na obrázku obr2.33 je zreteľné ako na kraji jeden z modelov odštvá a je medzi povrchmi medzera. Na obrázku obr2.34 môžeme vidieť ako oba povrchy cez seba prestupujú a tvoria tým dvojitý povrch v mieste prekrytia modelov. Kvôli tomuto problému je ešte potrebné povrchy zpriemerovať a spojiť do jedného. Na to slúži tlačidlo *Merge Models*.

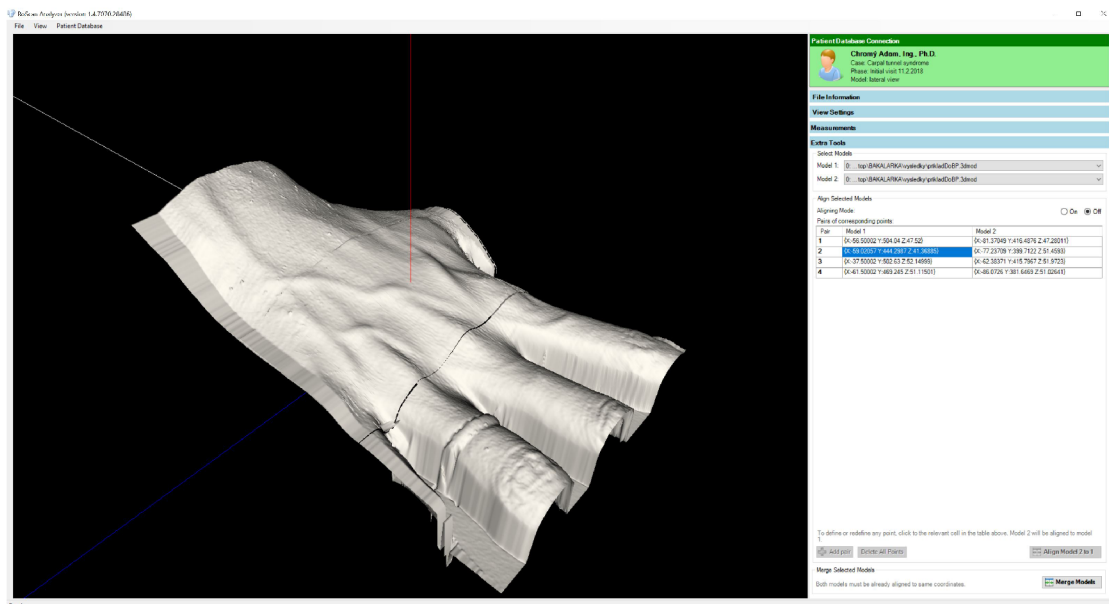


Obr. 2.33: Medzera medzi povrchmi dopasovaných čiastkových modelov.



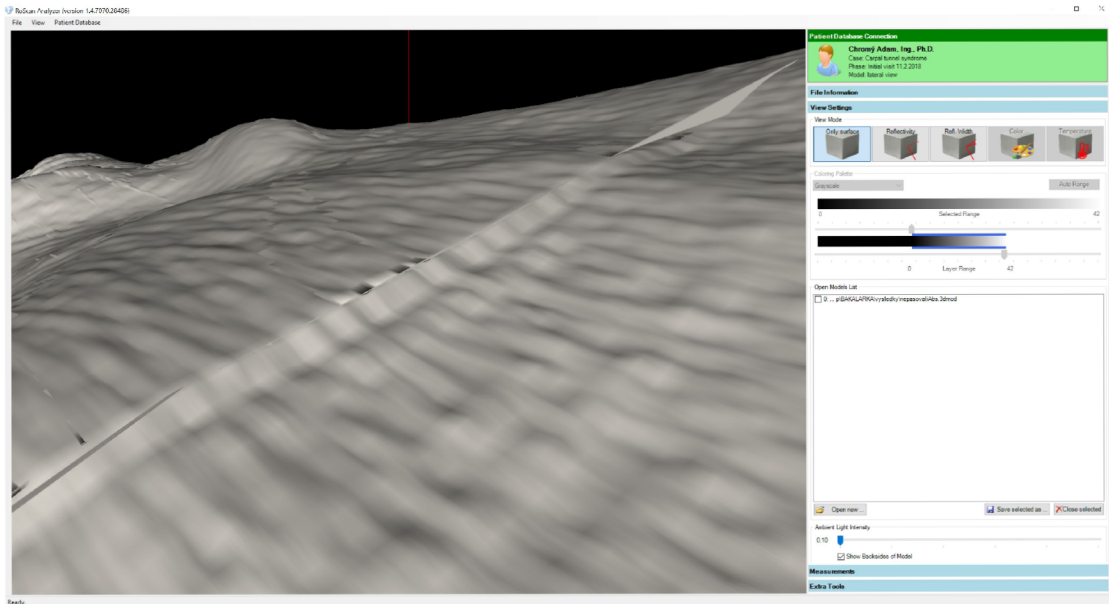
Obr. 2.34: Dvojitý povrch dopasovaných čiastkových modelov.

Pokiaľ modelom od začiatku pasovali absolútne súradnice a dopasovávanie nebolo potrebné, užívateľ celú časť s výberom bodov preskočí a stlačí rovno tlačidlo *Merge Models*. Po stlačení tohto tlačidla sa podľa algoritmu popísaného v teoretickej časti povrchy v prekrývajúcej časti spriemerujú a na základe bodov oboch modelov spoja a vytvoria tak jeden súvislý povrch, ako môžeme vidieť na obrázku obr2.35.

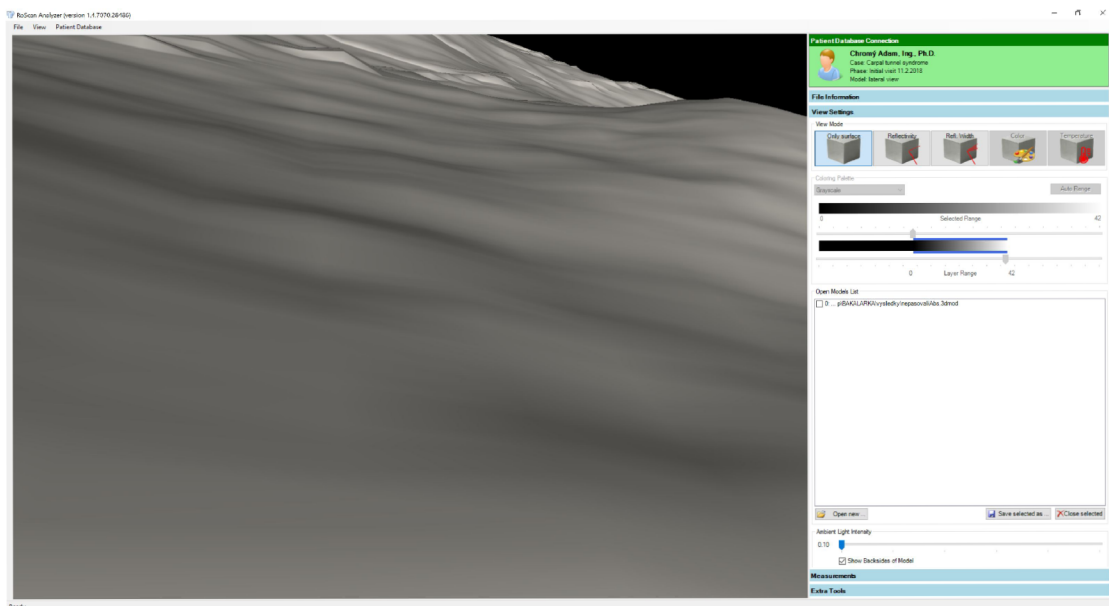


Obr. 2.35: Celkový model vytvorený spojením dvoch čiastkových modelov.

Na tomto vzdialenom pohľade možno nie je vidieť tak veľkú zmenu, no keď sa znovu pozrieme bližšie na miesto, kde model odstával (obr.2.36) a z blízka na povrch v pôvodne prekryvajúcej sa časti modelov (obr.2.37), môžeme vidieť, ako si s týmito problémami algoritmus poradil.

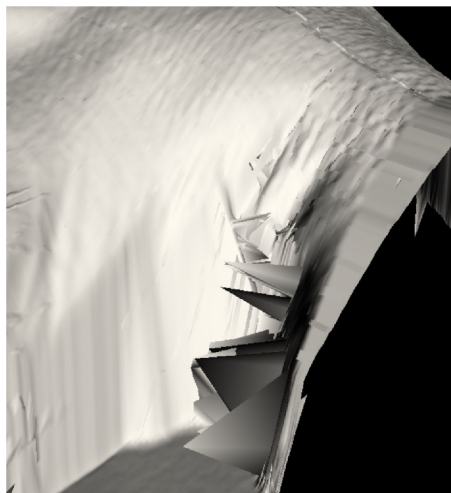


Obr. 2.36: Miesto kde odstával model po spojení čiasťových modelov.



Obr. 2.37: Blízky pohľad na povrch v mieste pôvodného prekrytia čiasťových modelov.

Tento algoritmus samozrejme nie je dokonalý a aj priemerovanie povrchu má určité nedostatky. Napríklad ako ukazuje obrázok obr2.38, kde sa body posunuli aj keď nemali, pretože algoritmus posunie vždy bod o polovičnú vzdialenosť k najbližšiemu priesečníku, čo je napríklad v tomto prípade nevhodné.



Obr. 2.38: Nevhodné priemerovanie bodov pri spájaní čiastkových modelov.

Ďalším veľkým problémom tohto algoritmu je jeho časová náročnosť. Pri testovaní trval výpočet celého spojenia modelov, v závislosti od vstupných dát, od približne pol hodiny, čo bol najrýchlejší výpočet, až do viac ako hodiny a pol. Preto v implementácií vidím ešte viac priestoru na optimalizáciu. Jeden z faktorov, ktorý môže výrazne ovplyvňovať čas výpočtu, je zisťovanie indexu trojuholníka pri hľadaní priesečníkov smerníc bodov s trojuholníkmi tvoriacimi povrch modelu. Súčasná implementácia pomocou Octree vracia totižto indexy bodov modelu, ktoré tvoria trojuholník, ktorým smernica prechádza a potom sa musí náročne hľadať index daného trojuholníku podľa týchto bodov.

Viac príkladov spojených modelov môžete nájsť na priloženom CD, kde sa nenachádzajú iba ďalšie obrázky výsledných modelov, ale aj inštalačný software prehliadača modelov a príklady spojených modelov vo formáte .3dmod.

## 3 Záver

V tejto práci bol podrobne rozobratý návrh algoritmu, ktorý z dvoch čiastkových trojrozmerných modelov vytvorí jeden celistvý model. Algoritmus bol navrhnutý pre systém RoScan a v rámci tejto práce aj implementovaný v jeho projekte ModeViewer.

Boli popísané možnosti systému RoScan a dva typy vstupných dát, ktoré môže systém RoScan poskytnúť a tiež postupy, ako sa tieto vstupy spracovávajú. Princíp algoritmu bol ukázaný a overený v systéme Matlab na vzorových dátach a neskôr implementovaný v jazyku C#.

Použitý postup dopasovania čiastkových modelov, ktoré boli skenované postupom, kedy im nepasujú absolútne súradnice, je veľmi užívateľsky prívetivý a navyše sú jeho výsledky veľmi presné.

Pri testoch na reálnych modeloch sa preukázala funkčnosť navrhnutého algoritmu spájania pri oboch typoch vstupných dát. Postup je schopný minimalizovať rozdiely medzi povrchmi oboch modelov ich spriemerovaním a následne spojiť oba povrchy pretriangulovaním referenčného modelu na základe bodov modelu, ktorý sa má k referenčnému pripojiť. Získané výsledky prekonalí moje očakávania.

Výsledný povrch, ktorý je produktom implementovaného algoritmu na spojenie, má však nedostatky v podobe vynechaných trojuholníkov pri detekcii prekrývajúcich sa častí dvoch čiastkových modelov, alebo pri posúvaní niektorých bodov pri priemerovaní. Navyše kvôli zlyhaniu popísaného postupu pre vyplnenie zostávajúcej úzkej medzery medzi spojenými čiastkovými modelmi je povrch neúplný. Medzera je však veľmi úzka a nie je až takým výrazným nedostatkom. V budúcnosti ale bude potrebné tento problém vyriešiť. Napriek tomu je však výsledný model veľmi kvalitný a použiteľný k ďalšej analýze užívateľom.

Za najväčší nedostatok implementovaného algoritmu považujem jeho časovú náročnosť, ktorá sa pri testoch, aj po optimalizácii niektorých výpočtov, v závislosti na vstupných dátach pohybovala približne v intervale pol hodiny až hodiny a pol. V tomto smere vidím stále možnosť výraznej optimalizácie.

## Literatúra

- [1] POLLEFEYS, M.: *Visual 3D Modeling from Images* [online]. Dostupné z URL: <<https://www.cs.unc.edu/~marc/tutorial.pdf>>.
- [2] ZABRAN, M.: *An Algorithm Recreating 3D Triangle Mesh Faces from Its Edges* [online]. Dostupné z URL: <<https://cescg.org/wp-content/uploads/2018/04/Zabran-An-Algorithm-Recreating-3D-Triangle-Mesh-Faces-from-Its-Edges.pdf>>.
- [3] HUBER, F., HEBERT, M.: *Fully automatic registration of multiple 3D data sets* [online]. 1992, Dostupné z URL: <[https://www.ri.cmu.edu/pub\\_files/pub3/huber\\_daniel\\_f\\_2001\\_1/huber\\_daniel\\_f\\_2001\\_1.pdf](https://www.ri.cmu.edu/pub_files/pub3/huber_daniel_f_2001_1/huber_daniel_f_2001_1.pdf)>.
- [4] BESL, P., McKAY, N.: *A method of registration of 3-D shapes*. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992, vol. 14, no. 2, p. 239-256.
- [5] HUGLI, H., JOST, T.: *A match and merge method for 3D modeling from range images*. [online]. Dostupné z URL: <<http://members.unine.ch/heinz.hugli/parlab2013/pub/pdf/2002-ICSP.pdf>>.
- [6] PITO, R.: *Mesh Integration Based on Co-Measurements*. In *IEEE International Conference on Image Processing*. 1996, p. 397-400.
- [7] TURK, G., LEVOY, M.: *Zippered Polygon Meshes from Range Images*. In *Proceedings ACM Siggraph '94*. 1994, p. 311-318.
- [8] HILTON, A., STODDART, A., ILLINGWORTH, J., WINDEATT, T.: *Marching triangles: range image fusion for complex object modelling*. In *IEEE International Conference on Image Processing*. 1996, p. 381-384.
- [9] RUTISHAUSER, M., STRICKER, M., TROBINA, M.: *Merging Range Images of Arbitrarily Shaped Objects*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1994, p. 573-580.
- [10] CHROMÝ, A.: *Multispectral 3D Surface Scanning System RoScan and its Application in Inflammation Monitoring and Quantification*. In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2018) - Volume 2: BIOIMAGING*, pages 106-113.

SCITEPRESS – Science and Technology Publications, Lda. ISBN: 978-989-758-278-3

- [11] CHROMÝ, A., ZALUD, L.: *Novel 3D Modelling System Capturing Objects with Sub-Millimetre Resolution*. In *Advances in electrical and electronic engineering. 2014, vol. 12, no. 5, p. 476-487*.
- [12] Carnegie Mellon University: *Singular Value Decomposition (SVD)* [online]. Dostupné z URL:   
<<https://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/book-chapter-4.pdf>>.
- [13] ARUN, K. S., HUANG, T. S., BLOSTEIN, S. D.: Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 1987, vol. PAMI-9, no. 5, s. 698–700.
- [14] CURLESS, B.: *Ray-triangle intersection* [online]. 2006, poslední aktualizace 9. 12. 2009. Dostupné z URL:   
<[https://courses.cs.washington.edu/courses/cse457/09au/lectures/triangle\\_intersection.pdf](https://courses.cs.washington.edu/courses/cse457/09au/lectures/triangle_intersection.pdf)>.
- [15] ORF, D.: *Transform3DBestFit* [online]. Posledá aktualizácia 23. 4. 2016. Dostupné z URL:   
<<https://github.com/danielorf/Transform3DBestFit>>.
- [16] AYALA, D., BRUNET, P., JUAN, R., NAVAZO, I.: *Object representation by means of nonminimal division quadtrees and octrees* In *ACM Transactions on Graphics. 1985, vol. 4, no. 1, p. 41-59*.

## 4 Obsah priloženého CD

Na priloženom CD nájdete príklady modelov vo formáte .3dmod a aj obrazové príklady výstupných modelov vo formáte .jpeg. CD tiež obsahuje zdrojové súbory celého projektu RoScan a inštalačné súbory, s návodom ako ich nainštalovať, aby bol program spustiteľný na novších verziách operačného systému Windows. Pre prehliadanie priložených modelov môžete využiť program RoScanAnalyzer.msi, ktorý obsahuje strašiu verziu projektu ModelViewer.

```
/ ..... koreňový adresár priloženého CD
├── Modely ..... príklady vstupných a výstupných modelov
│   ├── Bez dopasovania ..... vstupne modely s pasujúcimi súradnicami
│   │   ├── model1.3dmod ..... referenčný model
│   │   ├── model2.3dmod ..... pripájaný model
│   │   └── vysledok.3dmod ..... výsledný model
│   ├── Obrazky ..... obrazové ukážky výstupných modelov
│   │   ├── vysledok1.jpeg
│   │   ├── vysledok2.jpeg
│   │   ├── vysledok3.jpeg
│   │   ├── vysledok4.jpeg
│   │   ├── vysledok5.jpeg
│   │   └── vysledok6.jpeg
│   ├── S dopasovanim ..... vstupne modely s nepasujúcimi súradnicami
│   │   ├── model1.3dmod ..... referenčný model
│   │   ├── model2.3dmod ..... dopasovaný a pripájaný model
│   │   └── vysledok.3dmod ..... výsledný model
│   ├── Vstupne modely ..... ďalšie príklady vstupných modelov
│   │   ├── model1.3dmod
│   │   ├── model2.3dmod
│   │   ├── model3.3dmod
│   │   └── model4.3dmod
│   ├── Vystupne modely ..... ďalšie príklady výstupných modelov
│   │   ├── vysledok1.3dmod
│   │   └── vysledok2.3dmod
│   └── XnaForVS2017 ..... potrebné instalácie pre Visual Studio
│       ├── DirectX ..... priečinok s inštalačným súborom
│       ├── XNA Framework 4.0 Redistribution ..... priečinok s inštalačným súborom
│       ├── XNA Game Studio 4.0 Platform Tools ... priečinok s inštalačným súborom
│       ├── XNA Game Studio 4.0 Shared ..... priečinok s inštalačným súborom
│       ├── README.txt ..... návod na inštaláciu
│       └── XNA Game Studio 4.0.vsix
├── zdrojaky ..... zdrojové súbory systému RoScan
│   └── 04_3D_scanner_software
│       ├── ...
│       └── ...
└── RoScanAnalyzer.msi ..... prehliadač modelov typu .3dmod
```