

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

Návrh systému na tvorbu počítačového modelu
skutečného objektu vhodného pro následný 3D tisk

PLZEŇ, 2024

JAKUB GAIER

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jakub GAIER**
Osobní číslo: **A21B0370P**
Studijní program: **B0714A150005 Kybernetika a řídicí technika**
Specializace: **Automatické řízení a robotika**
Téma práce: **Návrh systému na tvorbu počítačového modelu skutečného objektu vhodného pro následný 3D tisk**
Zadávající katedra: **Katedra kybernetiky**

Zásady pro vypracování

- Provedte rešerši v oblasti snímání objektu pomocí jedné či více kamer za účelem tvorby počítačového modelu snímaného předmětu.
- Provedte rešerši metod Struktury z pohybu (Structure from Motion) a Simultánní Lokalizace a Mapování (SLAM) vzhledem k tématu práce.
- Navrhněte systém pro tvorbu počítačového modelu snímaného objektu s možností následného tisku na 3D tiskárně.

Rozsah bakalářské práce: **30-40 stránek A4**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

- Schonberger, J. L., & Frahm, J. M. (2016). Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4104-4113).
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., ... & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6), 1309-1332.
- Helle, R. H., & Lemu, H. G. (2021). A case study on use of 3D scanning for reverse engineering and quality control. *Materials Today: Proceedings*, 45, 5255-5262.

Vedoucí bakalářské práce: **Ing. Petr Neduchal, Ph.D.**
Výzkumný program 1

Datum zadání bakalářské práce: **17. října 2023**
Termín odevzdání bakalářské práce: **20. května 2024**



Doc. Ing. Miloš Železný, Ph.D.
děkan



Doc. Dr. Ing. Vlasta Radová
vedoucí katedry

V Plzni dne 17. října 2023

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 20. 5. 2024

.....
vlastnoruční podpis

Abstrakt

Tato práce se zabývá návrhem systému na tvorbu počítačového modelu skutečného objektu vhodného pro následný 3D tisk. Cílem práce je seznámit se se základními principy 3D skenování a tvorbou 3D modelu z naměřených dat až po jeho úpravu pro 3D tisk. Následně je dle prostudované literatury navržen vlastní systém pro tvorbu počítačového modelu. Pro pořízení dat je použita hloubková kamera Intel RealSense D435i a data jsou zpracována na notebooku. Dosažené výsledky jsou prezentovány na vytisknutém modelu.

Klíčová slova

3D skenování, 3D rekonstrukce, 3D tisk, Reverzní inženýrství, Rapidní prototypování, Mračno bodů, Iterative closest point algoritmus

Abstract

This work deals with the design of a system for creating a computer model of a real object suitable for subsequent 3D printing. The aim of this thesis is to get acquainted with the basic principles of 3D scanning and 3D model creation, from measured data to its modification for 3D printing. Subsequently, according to the studied literature, an own system for creating a computer model is designed. An Intel RealSense D435i depth camera is used for data acquisition and the data is processed on a laptop. The achieved results are presented on a printed model.

Key Words

3D scanning, 3D reconstruction, 3D printing, Reverse engineering, Rapid prototyping, Point Cloud, Iterative closest point algorithm

Poděkování

Tímto bych chtěl poděkovat panu Ing. Petru Neduchalovi Ph.D. za vedení mé bakalářské práce, podporu během jejího vypracovávání a ochotu při zodpovídání mých dotazů. Dále bych chtěl poděkovat výzkumnému centru katedry kybernetiky za zapůjčení hloubkové kamery použité k 3D skenování.

Obsah

1	Úvod	1
2	Převod skutečného objektu do počítačového modelu	2
2.1	Skenování objektu	2
2.1.1	Hardware 3D skenování	2
2.1.2	Metody optického měření vzdálenosti	5
2.2	Datové reprezentace modelu v počítači	9
2.2.1	Bodové modely	9
2.2.2	Povrchové modely	10
2.2.3	Objektové modely	12
2.3	Zpracování získaných dat	13
2.3.1	Filtrace dat	13
2.3.2	Registrace a integrace dat	15
2.3.3	Rekonstrukce povrchu	18
2.4	Vyspělé systémy 3D rekonstrukce	19
2.4.1	Structure from Motion	19
2.4.2	Simultaneous Localization and Mapping	20
2.5	Zpracování rekonstruovaného modelu pro 3D tisk	21
3	Návrh systému	23
3.1	Popis hloubkové kamery Intel RealSense D435i	24
3.2	Pořízení dat z hloubkové kamery	25
3.2.1	Popis scény experimentálního měření	26
3.3	Návrh systému pro tvorbu modelu	27
3.3.1	Registrace a integrace naměřených dat	27
3.3.2	Převod Point Cloudu do tisknutelného formátu	32
3.4	3D tisk	33
3.5	Uživatelské rozhraní	34
4	Závěr	37

Kapitola 1

Úvod

3D skenování je forma měření vzdálenosti za účelem tvorby počítačových modelů reálných objektů. První 3D skenery byly vytvořeny za rozmachu počítačových technologií během třetí průmyslové revoluce s účelem automatizovat dokumentaci objektů. Největšího využití v té době našli souřadnicové měřicí stroje pro měření vyrobených součástí. 3D skenování ve strojírenství slouží ke kontrole kvality součástí a nebo vytvoření 3D modelu objektu, k němuž není dostupná dokumentace. S vývojem bezkontaktních 3D skenerů se rozšířily i možnosti jejich uplatnění.

Reprezentace reálných objektů pomocí 3D modelů v počítači umožňuje vytvářet virtuální světy a provádět nad nimi různé simulace. Z tohoto důvodu nachází 3D skenování široké uplatnění. Laserové skenery se využívají například k mapování povrchu Země. Autonomní roboti a manipulátory vytváří 3D modely objektů ze svého okolí pomocí svých senzorů, aby mohli s nimi manipulovat. Automobilový průmysl uplatňuje 3D skenování v systémech autonomní jízdy. V zábavním průmyslu se objekty reálného světa přetváří 3D skenováním do jejich virtuálních podob. Za účelem tvorby mapy okolí byly vytvořeny systémy Simultaneous Localization and Mapping (SLAM) a Structure from Motion (SfM).

3D skenování společně s 3D tiskem spadá do kategorie rapidního prototypování. Příkladem takového procesu je vytvoření koncepčního modelu z modelíny, vytvoření počítačového modelu objektu pomocí 3D skeneru, ruční úprava parametrů a detailů modelu a zhotovení na 3D tiskárně. Rapidní vývoj je užitečný například při výrobě ergonomických součástí, protože ergonomie se těžko posuzuje z počítačového modelu a vývoj takovéto součásti zpravidla zahrnuje spoustu iterací.

Tato bakalářská práce se zabývá návrhem systému pro tvorbu počítačového modelu z dat získaných z hloubkové kamery Intel RealSense D435i. K reprezentaci dat během zpracování je zvolena forma mračna bodů. K registraci jednotlivých snímků je použit algoritmus Iterative Closest Point. Z mračna bodů je vytvořena trojúhelníková síť reprezentující povrch modelu a model je nakonec vytištěn na 3D tiskárně. Pro obsluhu navrženého systému je vytvořeno uživatelské rozhraní. Veškerý zdrojový kód navrženého systému 3D rekonstrukce i s uživatelskou aplikací je veřejně přístupný v GitHub repositáři na adrese https://github.com/jgaier/3D_Object_Scanner.

Kapitola 2 rozebírá dostupné 3D skenery a jejich principy fungování, možné způsoby reprezentace modelu v počítači, některé algoritmy a postupy pro tvorbu modelu a zpracování pro 3D tisk. Kapitola 3 se již zabývá samotným návrhem systému a popisuje použité zařízení, proces vývoje systému pro tvorbu modelu, dosažené výsledky s ukázkou vytisknutého modelu a vytvořené uživatelské rozhraní.

Kapitola 2

Převod skutečného objektu do počítačového modelu

Problematika rekonstrukce skutečného objektu do počítačového modelu je v odborné literatuře nazývána jako Structure from Motion, Reverse Engineering a Three-Dimensional Object Reconstruction. Problém spočívá v tom, že je třeba vytvořit počítačový model objektu/prostředí bez apriorní informace o tvaru a struktuře. Jmenované vědní disciplíny tedy zkoumají zejména algoritmy, které z dat pořízených ze 3D skenerů vytvoří jednu ucelenou reprezentaci objektu/prostředí [1, 2]. Celý proces tvorby počítačového modelu lze rozdělit na části, jimiž jsou 3D skenování, filtrace dat, registrace a integrace dat, rekonstrukce povrchu, zjednodušení a vyhlazení modelu. Počítačový model může být dále použit k segmentaci dat, detekci vlastností modelu a nebo porovnání součásti s výkresem [3]. Uvedené části procesu tvorby modelu budou popsány samostatně v dalších kapitolách.

2.1 Skenování objektu

Skenování objektu nebo terénu za účelem tvorby 3D modelu je žádáno v odvětvích jako je geodézie, fotogrammetrie, stavitelství, strojírenství, herní průmysl, počítačové vidění, ale najde uplatnění i v mnoha dalších aplikacích jako rekonstrukce historických památek, modelování měst a interiérů budov, monitorování archeologických nalezišť a medicíně [4].

3D skenování objektu je rozsáhlý pojem zejména kvůli odlišným požadavkům různých aplikačních domén. Existuje proto i mnoho principiálně odlišných senzorů a volba jednoho konkrétního závisí právě na dané aplikaci. Primárně se u senzorů pro 3D skenování porovnávají vlastnosti, jimiž jsou maximální vzdálenost měření, přesnost, velikost zorného pole, rychlost měření a cena. Podle místa nasazení může být vyžadován i konkrétní fyzikální princip měření senzoru. V následující kapitole jsou uvedeny příklady hardwaru pro 3D skenování, který se v posledních letech používá.

2.1.1 Hardware 3D skenování

Hardware 3D skenování lze rozdělit do dvou kategorií a to na kontaktní a bezkontaktní skenery. Kontaktní skenery mají nejdelsí dobu měření, ale dosahují zdaleka nejlepších přesností [1]. Bezkontaktní skenery jsou pak jejich protějšky s opačnými vlastnostmi. Kontaktní skenery ze svého principu nemohou přesně měřit měkké a deformovatelné

objekty. U bezkontaktních skenerů je na druhou stranu problém měřit objekty s lesklým či průhledným materiálem. Značnou výhodou bezkontaktních skenerů je možnost měřit na větší vzdálenosti, a právě ve spojení s rychlým ziskem dat se proto staly vhodnou volbou pro univerzální 3D skenování.

Bezkontaktní skenery využívají fyzikálních principů optiky, akustiky, ale také magnetismu nebo rentgenu [4, 5]. Lze je dělit na pasivní a aktivní. Aktivní skenery vysílají energii do okolí, nejčastěji laser, který se odrazí zpět do senzoru. Pasivní skenery zachycují pouze okolní energii, nejčastěji světlo.

Příklady používaných skenerů

Nejběžnějším zástupcem kontaktních skenerů je takzvaný souřadnicový měřicí stroj (CMM). Jedná se o manipulátor s dotykovou sondou, viz obrázek 2.1a. Manipulátor je typicky tvořen třemi posuvnými rameny a sonda má až dva rotační klouby. CMM má uplatnění zejména ve strojírenství, kde se používá hlavně ke kontrole kvality vyrobených součástí, ale také ke skenování součástí k nimž neexistuje dokumentace [5, 6]. Měřit úplně neznámý objekt kontaktní sondou vyžaduje vhodně naplánovat trajektorii pohybu, aby se sonda nepoškodila. Trajektorie může být naprogramována ručně, nebo lze vygenerovat z předchozího bezkontaktního měření [5, 1].

K bezkontaktnímu skenování lze využít i obyčejnou kameru. Pokud je k dispozici pouze jedna kamera, je třeba vyfotit objekt z několik míst. Ideální by bylo znát přesnou polohu a natočení kamery vůči pevně zvolenému systému souřadnic, aby bylo možné z 2D projekcí vytvořit 3D model na základě triangulace. Když není poloha kamery k dispozici, lze využít metod počítačového vidění k extrakci výrazných bodů a hledání transformační matice mezi snímky [3, 6]. Při použití více kamer najednou je postup podobný, ale je možné pořídit snímky ve stejný okamžik a eliminovat tak problém s proměnlivým osvětlením či pohybem objektu. K odhadu vzdálenosti pomocí kamery je třeba pořídit minimálně dva snímky z různých úhlů [3, 5, 2].

Laserové skenery jsou všechny skenery emitující laserový paprsek a lze je dělit podle fyzikálních principů měření vzdálenosti. Na krátké vzdálenosti se používá princip triangulace a strukturované světlo, na větší vzdálenosti se využívá metody měření doby letu (ToF) a fázového posunu laserového pulsu. V průmyslu jsou velmi oblíbené ruční laserové skenery, viz obrázek 2.1d, které umožňují rychle a flexibilně pořídit 3D data.

Light Detection and Ranging nebo zkráceně LiDAR je laserový skener, který pomocí zrcadel vychylujících laserový paprsek umožní skenovat ve 2D nebo 3D [7]. Příklad 2D LiDARu je na obrázku 2.1c. LiDAR nejčastěji využívá principu ToF pro měření vzdálenosti a je často používán v mobilní robotice, autonomní jízdě, měření výšky povrchu země, ale i v mnoha dalších oblastech [4, 7].

Totální stanice (TS) je pozemní laserový skener používaný hlavně v geodézii. Na obrázku 2.1b je zobrazena TS na stativu. Pomocí elektro-optického zařízení se měří úhly mezi body zájmu a pomocí laseru se měří vzdálenost objektu. TS dosahuje velmi přesných výsledků, ale není vhodná pro mobilní skenování, protože musí být pevně umístěna na známém bodu v prostoru. [8]

První cenově dostupné zařízení pro 3D skenování vydala firma Microsoft pod názvem Kinect, viz obrázek 2.2a. Původně se jednalo o příslušenství k herní konzoli Xbox 360, jež umožnilo pohybem celého těla ovládat herní postavu. Tento skener funguje na bázi strukturovaného světla a je vybaven i barevnou kamerou [9]. O pár let později vyšla druhá verze Kinectu nabízející větší rozlišení a zorné pole. Kinect v2 také změnil

způsob měření vzdálenosti na metodu ToF a celkově dosahuje větší přesnosti měření než jeho předchůdce [9, 7].

Na trhu se postupně začalo objevovat více komerčně dostupných 3D skenerů. Jedním z nich je kamera Intel RealSense D435, viz obrázek 2.2b, která je mimo jiné použita v této práci k získání dat. Oproti Kinectu v2 je D435 menší a lehčí, což umožňuje ji upevnit na drobnější autonomní zařízení, a také má vyšší rozlišení a snímkovou frekvenci [10]. D435 používá primárně stereovizi k určení vzdálenosti, ale je vybavena i pomocným infračerveným projektorem, který pomáhá při měření hladkých objektů. Veškeré výpočty ke zpracování obrázků ze stereovize se provádí na vlastním integrovaném čipu a skener tak produkuje odhad vzdálenosti v reálném čase.



(a) Souřadnicový měřicí stroj



(b) Totální stanice



(c) LiDAR



(d) Ruční laserový skener

Obrázek 2.1: Příklady průmyslových 3D skenerů



(a) Microsoft - Xbox 360 Kinect



(b) Intel RealSense D435

Obrázek 2.2: Příklady cenově dostupných 3D skenerů

2.1.2 Metody optického měření vzdálenosti

Nejčastěji používané bezkontaktní skenery fungují na bázi optiky. Světlo je nejrychlejším přenosovým médiem a objekt zájmu je ve většině případů viditelný. Optické skenery dosahují ve většině případů dostatečné přesnosti měření a jsou nejlepšími kandidáty pro 3D skenování neznámých objektů. Optické skenery mají problémy s měřením lesklých a průhledných materiálů, protože se většina světla od nich rozptýlí do okolí [3]. Dnes existuje spousta optických 3D skenerů pro měření na malé i velké vzdálenosti s uspokojivou kvalitou. Hardware optického 3D skenování byl již značně vylepšen, nynějším středem zájmu jsou samotné algoritmy pro zpracovávání dat [5].

Alternativou optických skenerů jsou například magnetické skenery. Ovšem ty lze použít jen na magnetické materiály a jejich maximální vzdálenost je značně omezena. Další alternativou jsou akustické skenery. Ty dnes nacházejí využití spíše pod vodou, kde oproti optickým skenerům umožňují skenovat na větší vzdálenosti. Akustické skenery se také ukazují být nejlepšími pro mikroskopické skenování povrchu [11]. Rentgenové skenery jsou velmi drahé a uplatnění nacházejí spíše v medicíně.

Metody měření aktivními skenery

Aktivní skenery se skládají z emitoru a snímače. Emitor vysílá signál do okolí a snímač měří jeho odraz. Do této kategorie spadají laserové skenery a skenery s vlastním osvětlením. Laserové skenery mají potíže s měřením hran objektu, protože jedna část laserového signálu se odrazí od hrany a druhá část od plochy za hranou [12, 13]. Signál se tak vrací do snímače ze dvou míst najednou a dojde ke zprůměrování výsledku [12].

A. Doba letu

Metoda měření doby letu (anglicky Time of Flight - ToF), měří dobu, za jakou světlo urazí cestu k objektu a zpět. Skener vyšle krátký laserový puls a začne odpočítávat čas. Puls letí rychlostí světla k objektu, odrazí se od něj a letí zpět do snímače, kde je odměřena doba celkového letu [3, 12]. Na základě tohoto měření je vypočítána vzdálenost objektu. ToF metoda měří z principu vzdálenost pouze jednoho bodu. Pro měření celých ploch se nejčastěji využívá zrcátek pro vychýlení laserového pulsu, protože jejich lehká váha umožňuje s nimi přesně otáčet [12].

Jelikož je rychlost světla známá, přesnost těchto zařízení závisí na přesnosti měření času a na odrazových vlastnostech měřeného objektu [3, 12, 1]. Skenery s tímto principem měření dosahují až milimetrové přesnosti měření a až kilometrové maximálně měřitelné vzdálenosti [12].

Podobně jako ToF funguje princip měření fázového posunu. Odražený signál se vrátí do snímače s fázovým zpožděním, jež odpovídá vzdálenosti objektu. Aby nedocházelo k nejednoznačné interpretaci měření kvůli periodicitě fáze, je měření provedeno pomocí vícero laserových signálů s různou frekvencí. Tato metoda dosahuje menších přesností a menších měřitelných vzdáleností než metoda měření doby letu. [12]

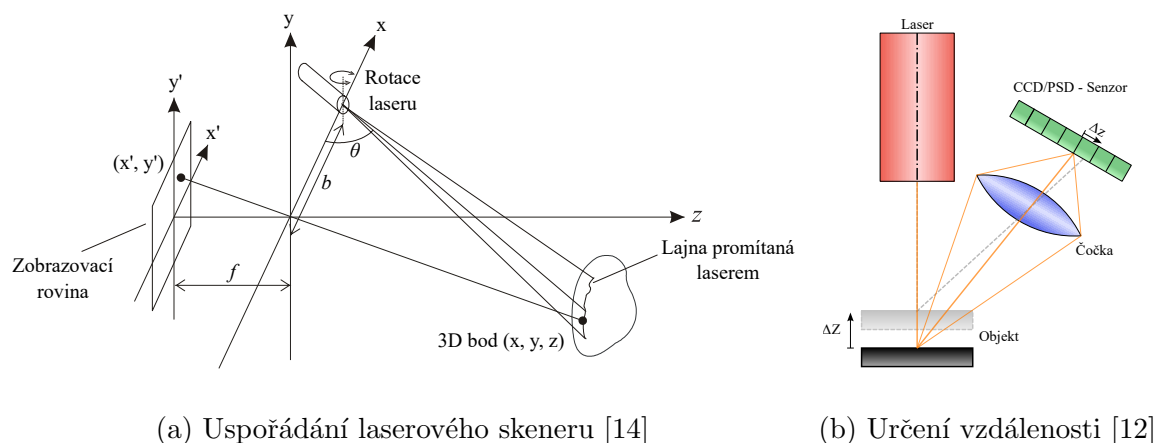
B. Triangulace

Princip měření metodou triangulace, jak už z názvu vyplývá, spočívá v geometrických vlastnostech trojúhelníku. Laserový projektor, čočka snímače a měřený objekt tvoří trojúhelník, viz obrázek 2.3a, v němž je známá vzdálenost mezi zdrojem paprsku a středem čočky a úhly přilehlé této straně [12, 14, 5]. Zobrazovací

rovina kamery je umístěna do ohniska čočky, jejíž vzdálenost je také známa [14, 5]. Ze všech těchto informací lze dopočítat reálnou vzdálenost objektu.

Přístroj je předkalibrován na vzdálenost, kde se odražený paprsek zobrazí do prostředí zobrazovací roviny senzoru. Variace ve vzdálenosti objektu způsobuje posun bodu v zobrazovací rovině, viz obrázek 2.3b, na základě čehož je určen reálný úhel čočky vůči objektu. Pro měření více bodů v prostoru je třeba otáčet laserovým projektorem. Stejně jako u ToF skenerů se nejčastěji používají zrcátka k vychýlení laseru. Natočení zrcátka je měřeno enkodérou [14]. Ve většině případů emituje laserový projektor linii paprsků najednou, což vede ke zrychlení akvizice dat [12].

Skenery používající princip triangulace dokáží přesně měřit jen do vzdálenosti několika metrů, ale oproti ToF skenerům mají daleko větší přesnost měření, která se pohybuje v řádech desítek mikrometrů [12].



Obrázek 2.3: Princip laserové triangulace

C. Strukturované světlo

Metoda projekce strukturovaného světla vychází z metody triangulace, ale namísto jedné lajny se projektuje rovnou celý 2D vzor [12, 5]. Vzor se vlivem perspektivy a variace hloubky objektu zdeformuje. Kamera zachytí zdeformovaný vzor a analýzou obrázku se extrahuje hloubka povrchu sledovaného objektu [12, 5].

Vzory mohou být jednoduché rovnoběžné lajny a mřížky, ale mohou mít i speciální tvary nebo různou intenzitu osvětlení [5]. Podle volby vzoru se odvíjí rychlost a komplexita analýzy obrázku a kvalita měření hloubky povrchu [5]. Akvizice dat pomocí strukturovaného světla je rychlejší než u triangulace, protože se analyzuje celý povrch najednou [12]. Ze stejného důvodu je i tato metoda odolná vůči zkreslení způsobené pohybem kamery nebo objektu [12].

Metody měření pasivními skenery

Pasivní skenery nevysílají žádný užitečný signál do okolí. Oproti aktivním skenerům jim tedy chybí emitor a musí si vystačit pouze se snímači. Optické pasivní skenery jsou zejména kamery a fotoaparáty sledující viditelné spektrum světla [12]. Jejich výhodami jsou nízká spotřeba energie a nízká pořizovací cena [12]. Na druhou stranu

jsou citlivé na okolní osvětlení a obecně mají problém s měřením hladkých povrchů bez výrazných rysů [13]. Samotný získání dat je velmi rychlý, ale analýza obrázků k určení vzdálenosti je časově náročná [5, 13, 2]. Zpracováním obrázků se zabývají vědní disciplíny počítačového vidění a fotogrammetrie.

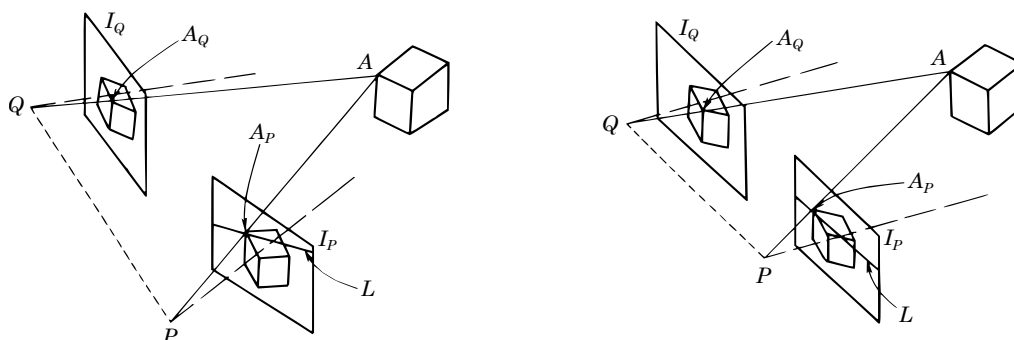
A. Stereovize

Metoda stereovize vychází z principu triangulace. Dvě kamery pozorují stejný objekt z různých míst a část jejich projekcí obrazu se překrývá. Nutnou podmínkou triangulace je, že je známá vzájemná poloha a směr natočení kamer, aby bylo možné určit vzdálenost bodů pozorovaných z obou míst. Ovšem určení vzdálenosti bodů není triviální, protože orientace objektu není známá a musí se určit, kterému bodu v prvním obrázku odpovídá bod z druhého obrázku. [15]

K nalezení párového bodu není třeba prohledávat celý obrázek, stačí bod hledat na projekci paprsku L , který vychází ze středu kamery Q a prochází vybraným bodem A_Q , v zobrazovací rovině I_P druhé kamery, viz obrázek 2.4a. Když se kamery postaví do jedné roviny tak, aby jejich osy otáčení byly rovnoběžné, projekcí paprsku bude horizontální čára, viz obrázek 2.4b, což zjednoduší hledání párových bodů. Ovšem ani toto zjednodušení nezaručí jejich nalezení. Například u objektu s hladkým povrchem nebo s pravidelnými vzory bude těžké rozeznat jeden bod od jeho okolí. K párování se proto vybírají pouze body s lokálně odlišitelnými rysy, jako je například velká změna intenzity světla na hranách objektu. [15]

Pouhé nalezení výrazných rysů nestačí, protože i ty si bývají podobné. Ačkoliv tedy platí, že párové body leží ve stejné rovině, je třeba analyzovat všechny rysy najednou. Ke globální analýze se často využívá těchto tří vlastností. Právý pár odpovídajících si rysů si je podobný pro různé velikosti zvoleného okolí. Rysy zachovávají horizontální i vertikální pořadí. Pokud rysy stejného typu leží na jedné vertikální přímce v prvním obrázku, tak leží i na jedné vertikální přímce v druhém obrázku. Ani při znalosti těchto vlastností se nepovede vždy jednoznačně určit pár odpovídajících si rysů. Často se v těchto situacích provádí odhad, který se v budoucnu opraví na základě nových pozorování. [15]

Uspořádání kamer má vliv i na schopnost hledání odpovídajících si rysů. Často používané horizontální uspořádání zjednodušuje hledání párů na vertikálních přímkách, ale již hůře se hledají páry na přímkách horizontálních. Tento nedostatek lze vyřešit přidáním další kamery nad jednu ze stávajících. [15]



(a) Kamery obecně v prostoru

(b) Kamery v horizontální rovině

Obrázek 2.4: Princip stereovize [15]

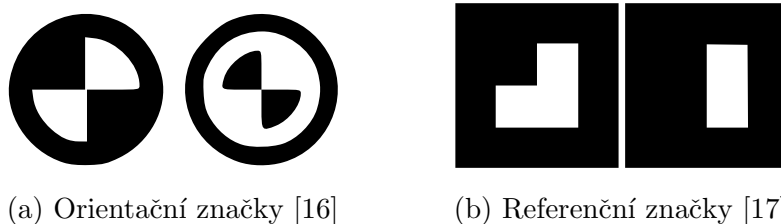
B. Pohyb kamery

Principu triangulace lze využít i s jednou kamerou. Tou je třeba pohybovat, aby zachytila objekt z více míst. Místa a natočení kamery jsou předem známé, aby byly splněny podmínky pro triangulaci. Mezitím co se kamera pohybuje z jednoho místa do druhého, pořizuje průběžně snímky objektu. Díky malé změně v poloze je jednoduché hledat odpovídající si body. Poté co kamera dorazí do nové pozice, okamžitě ví, které body si odpovídají a může jednoznačně určit jejich vzdálenost. Stejného výsledku lze docílit i pohybem objektu namísto kamery. Objekt může být například umístěn na rotační podložku, u které je měřeno natočení. [15]

V reálných situacích často není poloha kamery známá a tudíž nelze použít princip triangulace přímo. V tomto případě je třeba polohu kamery odhadnout [2]. Řešení tohoto problému se schovává v sadě složitých nelineárních rovnic [15]. Ve většině případů nelze jednoznačně najít řešení a místo toho jsou aplikovány numerické algoritmy hledající lokální minimum. Touto problematikou se zabývá technika Structure from Motion, která je popsána detailněji v kapitole 2.4.1.

C. Sledování značek

Metoda sledování značek se uplatňuje tehdy, když poloha kamery není známá a objekt je možné osadit značkami. Značky musí být vůči objektu jednoduše rozpoznatelné, typicky mají na sobě nějaký černobílý vzor. Podle způsobu využití mohou být značky děleny na dva typy, orientační a referenční. Orientační značky mohou sloužit jen jako umělé rysy k usnadnění párování, v tomto případě je třeba pořádit více snímků objektu z různých úhlů pohledu [16]. Pokud jsou na objekt umístěny alespoň tři orientační značky v nepravidelném uspořádání, je možné určit polohu kamery v prostoru již z jednoho snímku [15]. Na obrázku 2.5a jsou zobrazeny kruhové orientační značky u nichž je snadné najít jejich střed. Referenční značky mají na sobě složitější vzor, který umožňuje jednoznačně určit polohu a natočení kamery z jednoho snímku [17]. Příklad primitivních referenčních značek je na obrázku 2.5b.



Obrázek 2.5: Příklad značek pro 3D skenování

Výhody kombinování různých metod 3D skenování

Každá metoda má své výhody a nevýhody, proto se vyplatí metody vhodně kombinovat, aby byly potlačeny jednotlivé nedostatky a celkově se navýšila kvalita a nebo rychlost měření.

Ukázkovým příkladem je kombinace kontaktního CMM s bezkontaktním skenerem. V této kombinaci se využívá rychlosti bezkontaktních skenerů, jako je například laserový skener, k naplánování optimální trajektorie pohybu pro sondu CMM, když je tvar objektu neznámý [5]. Bezkontaktní měření značně urychlí proces identifikace objektu,

zatímco kontaktní sonda poskytne velmi přesné měření parametrů. Tato kombinace skenerů se používá v oblasti reverzního inženýrství [1].

Dalším ukázkovým příkladem je kombinace laserového skeneru s kamerou, respektive kombinace aktivního a pasivního skeneru. Laserový skener produkuje husté mračno bodů s velkou přesností, ale má problém s měřením hran [13]. Kamera na druhou stranu dokáže přesně měřit hrany, ale má problém párovat body na rovných plochách a analýza obrázků je příliš zdlouhavá a náročná [13]. Spojením obou metod se využije jejich silných stránek a celkově se zvýší kvalita skenování. Této kombinace využívají i hloubkové kamery Intel RealSense.

2.2 Datové reprezentace modelu v počítači

Počítačový model objektu lze realizovat mnoha způsoby a výběr reprezentace závisí především na účelu použití. V počítačové grafice se využívá polygonální síť (Polygonal Mesh), protože grafické karty jsou optimalizované na vykreslování trojúhelníků [5]. Reverzní inženýrství potřebuje vytvořit ideálně parametrický model, stejný jako kdyby byl navržen v CAD softwaru [12]. Zatím se nepodařilo vytvořit algoritmy, jež by zrekonstruovaly libovolně sofistikovaný objekt pomocí parametrizovaných objektů. Dnes se v reverzním inženýrství využívá převážně povrchového popisu pomocí křivek a ploch [1, 5]. Ve vyspělých systémech 3D rekonstrukce, kde se využívají kamery k vytvoření mapy okolí, se rozvinula reprezentace pomocí mapy významných bodů (Feature Map) k úsporné reprezentaci dat [18]. Novější SLAM systémy adaptují nový způsob reprezentace scény pomocí Volumetric Truncated Signed Distance Function (TSDF), která popisuje povrch objektu po částech spojitou funkcí [18, 19]. Detaily jednotlivých reprezentací jsou popsány v následujících podkapitolách.

2.2.1 Bodové modely

Skenery jsou určeny primárně pro měření vzdálenosti a jejich výstupem je nějaká forma diskretních bodů v prostoru. Diskretní body nastíní obrys modelu, ale těžce se s nimi pracuje. V mnoha aplikacích, jako je například reverzní inženýrství, je třeba tyto data zpracovat do vyšší formy reprezentace a vytvořit z nich model vhodný pro použití v CAD softwarech [1].

Point Cloud

Data ze skenerů jsou typicky body popsány souřadnicemi x , y , z , jež dohromady tvoří obrys modelu nazývaný mračno bodů, neboli Point Cloud. Kromě polohových souřadnic může mít každý bod navíc informaci o barvě, normálovém vektoru a míře nejistoty odhadu polohy. Laserové skenery jsou typickým příkladem skenerů, které poskytují data ve formě Point Cloudu. Body v této podobě nemají mezi sebou žádnou souvislost, je tedy těžké na základě nich odvozovat vlastnosti modelu [18].

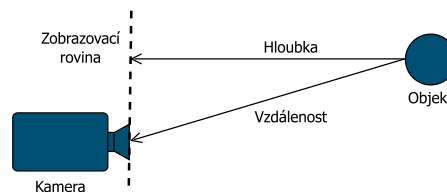
RGB-D obrázek

Druhý nejběžnější popis nezpracovaných dat je RGB-D obrázek, který produkují kamery ze stereovize. Klasický RGB obrázek je doplněn o hloubku (D), což je nejmenší

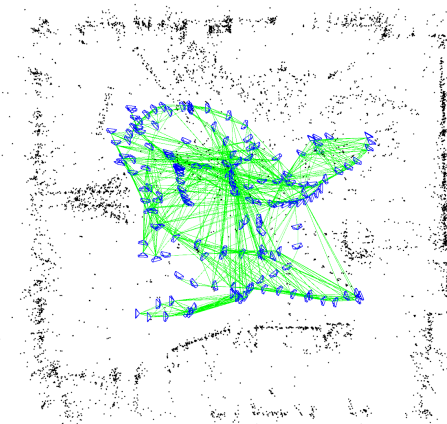
vzdálenost do zobrazovací roviny, viz obrázek 2.6. Hloubka je přiřazena každému pixelu na pořízeném snímku. Pokud není možné hloubku odhadnout, je danému pixelu přiřazena vzdálenost nula. RGB-D obrázek může být dále zpracováván v této podobě, a nebo se ze znalosti vnitřních parametrů kamery jednoduše převede na Point Cloud.

Feature Map

Mapa významných bodů (Feature Map) se nejčastěji využívá v systémech, kde se zpracovávají obrázky. Mapa obsahuje významné body extrahované ze snímků a pózy kamer, které dané body pořídily. Na základě podobnosti významných bodů se určí závislost mezi pózami kamer, což umožní vytvořit mapu okolí s reálnými vzdálenostmi. V případě, kdy je použita pouze jedna kamera s neznámými vnitřními parametry, dochází totiž ke zkreslení tvaru objektu. Mapa významných bodů se používá v SfM a SLAM systémech k vytvoření Point Cloud modelu ze snímků okolí. Na obrázku 2.7 je příklad, jak může Feature mapa vypadat. Významné body jsou černé, modré trojúhelníčky reprezentují pózy kamer a související kamery jsou propojené zelenou čarou [20]. Feature mapa může být také reprezentována jako graf póz (Pose Graph). Rozdíl je v tom, že graf póz je 2D zobrazení mapy pomocí grafu a kamery jsou zde propojeny jen s bezprostředními sousedy.



Obrázek 2.6: Rozdíl mezi hloubkou a vzdáleností od senzoru



Obrázek 2.7: Příklad mapy významných bodů [20]

2.2.2 Povrchové modely

Povrchové modely popisují tvar objektu pomocí rovin, křivek nebo ploch a tvoří tak spojitý popis modelu. V systémech 3D skenování jsou vytvářeny z naměřených bodů. Spojitý popis se hodí pro fyzikální simulace a plánování, jako je manipulace s objekty a předcházení kolizí [18]. Reprezentace modelu povrchem je dostačující a nejběžněji používaná reprezentace pro rekonstrukci skenovaných součástí [1].

Polygon Mesh

Nejjednodušší způsob jak vytvořit povrch modelu je pospojovat sousední body tak, aby vytvořili malé mnohoúhelníky neboli polygony. Polygon Mesh je síť polygonů tvořící model objektu. Nejpoužívanějším polygonem je trojúhelník, protože je to nejmenší mnohoúhelník pomocí něhož se dají vytvořit všechny ostatní mnohoúhelníky.

Polygon Mesh se většinou nemodeluje přímo, ale je vytvořen například z parametrických ploch. 3D modely jsou převáděny na trojúhelníkovou síť, protože grafické karty v počítačích jsou optimalizovány pro vykreslování trojúhelníků a pro výpočty spojené s nimi. Opačný proces, kde se z mračna bodů tvoří trojúhelníky, je mnohem složitější. Vygenerovaná síť má v sobě díry, osamocené trojúhelníky a nebo trojúhelníky, které se protínají, což v následujícím zpracování působí potíže [5]. Existují algoritmy, které tyto defekty eliminují, vyhladí povrch a nebo zmenší počet potřebných trojúhelníků k reprezentaci modelu [3]. Modifikace tvaru modelu v této podobě je složitá a Mesh zabírá spoustu paměti [5]. Pro účely 3D modelování se místo Meshes častěji používají parametrizované křivky a plochy.

B-Spline a NURBS

Basis Spline (B-Spline) je hladká parametrizovaná křivka definovaná kontrolními body. B-Spline vychází z Bézierovy křivky, která je definovaná polynomiální funkcí, ale oproti ní je definována po částech [21]. Části jsou rozděleny pomocí rovnoměrně rozprostřených uzlů na polynomiální funkce nižších řádů s vlastností, že sousední části na sebe hladce navazují. Na obrázku 2.8a je B-Spline se stupněm polynomu 3. Křivka může být buď aproximační nebo interpolační [22]. V druhém případě prochází křivka všemi kontrolními body, zatím co v prvním nemusí. Naměřené body jsou kontaminovány šumem, proto se aproximační křivky typicky odhadují metodou nejmenších čtverců.

Non-Uniform Rational Basis Spline (NURBS) je zobecněná forma B-Spline. Non-Uniform znamená, že uzly nejsou rovnoměrně rozprostřené po křivce a Rational znamená, že kontrolní body mají váhy, jak moc k sobě přitahují křivku [22]. Rovnoměrným rozložením křivek do čtvercové mřížky se vytvoří plocha ve 3D, viz obrázek 2.8b.

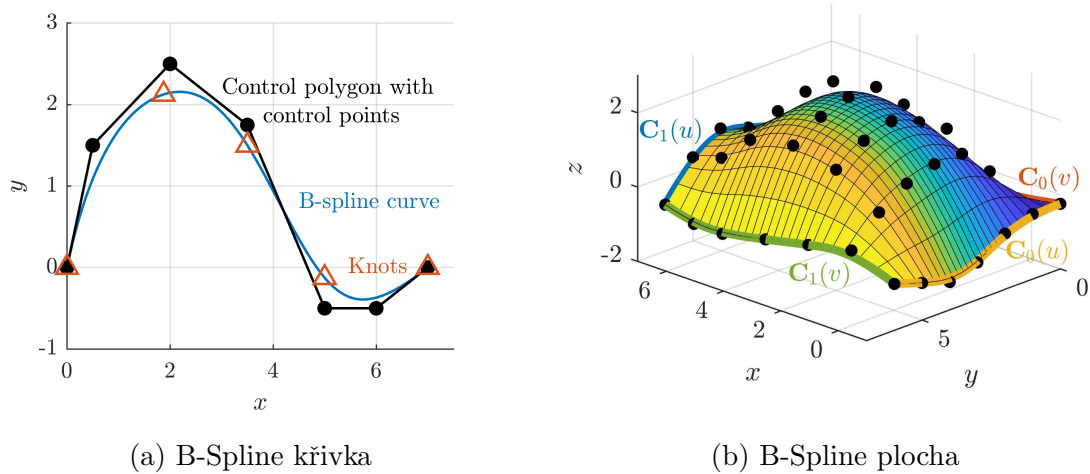
Výhodou těchto parametrických křivek je lokální kontrola tvaru. Posun jednoho bodu způsobí pouze lokální změnu tvaru křivky [22, 21]. NURBS mají i své nevýhody, například obtížně se zjišťuje náležitost bodu křivce a hledání jejich průsečíků [22].

Truncated Signed Distance Function

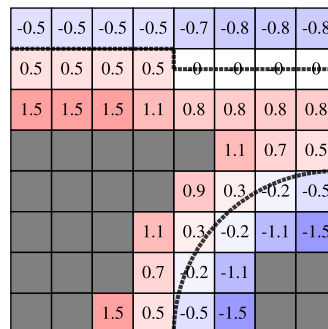
Volumetric Signed Distance Function (SDF) reprezentuje model pomocí vzdálenostní funkce. Prostor je rozdělen do pravidelné mřížky na takzvané voxely (krychle) a každý voxel v sobě uchovává vzdálenost od povrchu objektu ke svému středu a váhu tohoto měření. Vzdálenost od povrchu objektu je vypočítána pomocí paprsku vycházejícího z kamery, jež je kolmý na zobrazovací rovinu [19]. Váha reprezentuje nejistotu měření, ta je například malá na okrajích objektu. Reálná poloha povrchu se zpětně určí z voxelů, které vůči svým sousedům mění znaménko, jinak řečeno, hledá se poloha nuly, viz obrázek 2.9.

Pro vykreslení celého objektu se musí spojit SDF z vícero úhlů pohledu. Kvůli tomuto problému se zavádí Truncated Signed Distance Function (TSDF), která omezí rozsah vzdálenosti od povrchu na $\pm\mu$. Voxely před objektem jsou omezeny na maximální hodnotu $+\mu$ a voxelům za povrchem se vzdáleností menší než $-\mu$ se nepřičítají

hodnota, aby neovlivnila rekonstrukci povrchu z druhé strany objektu. Díky těmto omezením se povrch modelu vypočítá jednoduše jako vážený aritmetický průměr všech TSDF. [19]



(a) B-Spline křivka (b) B-Spline plocha
Obrázek 2.8: Příklad aproximační B-Spline křivky a plochy [21]



Obrázek 2.9: TSDF mřížka ve 2D [23]

2.2.3 Objektové modely

Objektové modely používají elementární prostorové útvary, jimiž jsou například koule, kvádry a kužele, k popisu složitějších útvarů s libovolným tvarem. Toho docílí například skládáním jednotlivých elementů a nebo jejich transformací a Booleových operací [18]. Objektové reprezentace zachycují hlubší podstatu modelu a umožňují měnit větší část objektu pouhou změnou jednoho parametru, například poloměrem koule [18]. Takovéto modely se nejvíce přibližují k ručně vytvořeným modelům v CAD softwarech, kde návrhář modeluje součást pomocí vazeb s nějakým záměrem [12]. Oproti povrchovým modelům jsou i paměťově kompaktnější, protože objekty jsou charakterizovány menším počtem parametrů než křivky a Meshes [18]. Objektové modelování ve 3D skenování je teprve na počátku svého vývoje, ale má velký potenciál do budoucna.

2.3 Zpracování získaných dat

V předchozí kapitole byly popsány různé způsoby reprezentace modelu. Ovšem získání těchto modelů v procesu 3D skenování není jednoduché, protože jsou data kontaminována šumem měření a outliery. Tato kapitola rozebírá jednotlivé části procesu zpracování získaných dat ze skeneru k vytvoření povrchového modelu. Vzhledem k tomu, že zpracování dat závisí na datové reprezentaci a rozebírání všech metod by bylo vyčerpávající, je tato kapitola zaměřena pouze na zpracování dat ve formě mračna bodů.

Proces zpracování mračna bodů začíná odfiltrováním šumu měření a outlierů. Poté následuje časově nejnáročnější část, registrace dat z jednotlivých úhlů pohledu na sebe. Integrací registrovaných dat vznikne model celého objektu. V závislosti na úspěšnosti filtrace na začátku procesu se po integraci mohou objevit další outliery. Ty se ještě před vytvořením spojitého povrchu musí odstranit, aby vytvořený povrch neměl nežádoucí artefakty. Podle potřeby může být model dále vyhlazován a ručně upravován.

2.3.1 Filtrace dat

Filtrace je nedílnou součástí každého zpracovávání dat při použití reálných senzorů. V systémech pro 3D skenování se typicky filtruje šum měření a takzvané outliery. Outliery jsou odlehle body, které nereprezentují měřený model a jejich výskyt je způsobený nahodilou chybou, jako je odraz světla od optického snímače, nebo v případě stereo-vize částečné zakrytí modelu z pohledu druhé kamery. Šum je vysokofrekvenční signál obsažený v měření a způsobuje malou odchylku od reálné hodnoty u všech bodů. Žádný povrch tedy nebude naskenován jako hladký. Šum je způsoben samotným snímačem, ale může být zesílen například pohybem skeneru. Šum měření se filtruje vždy na začátku zpracování dat, kdežto filtrace outlierů se může provádět i několikrát během zpracování. Filtrací dat lze nazvat jakýkoliv proces, kde dochází k vyhlazování nebo selekci. Spadá sem tedy i prostý výběr podmnožiny bodů, anglicky Subsampling. Ten je důležitý zejména kvůli snížení výpočetní doby v algoritmech pro následující zpracování.

Subsampling

Subsampling, neboli výběr podmnožiny či vzorkování, je častou praktikou při 3D rekonstrukci. 3D skenery jsou schopny pořídit desítky až stovky tisíců bodů v každém snímku, ale aktuálně používané algoritmy by tak obrovské množství dat zpracovávali příliš pomalu. Zejména aplikace reálného času si nemohou dovolit zpracovávat takové množství dat. Vzorkování se zabývá redukcí množství dat, ale snaží se, aby vybraná podmnožina stále věrohodně reprezentovala model. Metody náhodného výběru nejsou žádoucí.

Příklad algoritmu pro zmenšení počtu bodů v Point Cloudu je objemové vzorkování, anglicky Voxel Downsampling. Tato metoda rozdělí prostor na voxely (stejně velké krychle) o definované velikosti a všechny body nacházející se v daném voxelu jsou nahrazeny jediným bodem. Tento bod je aritmetickým průměrem všech bodů v daném voxelu, přesněji poloha, barva a další vlastnosti bodů jsou průměrovány zvlášť. Oproti voxelové reprezentaci v TSDF zde body nemusí ležet uprostřed voxelů. Voxelové vzorkování je efektivní volbou co se týče doby výpočtu a zachování struktury původního Point Cloudu. Na druhou stranu zde dochází ke ztrátě lokálních detailů. [24]

Další příklad jednoduchého výběru podmnožiny je výběr každého k -tého prvku. Tento algoritmus je rychlý a prostý, ale není zaručeno rovnoměrné rozložení bodů a je pravděpodobnější, že dojde ke ztrátě významných bodů oproti předchozí metodě. Tento přístup je na hranici mezi náhodným a strategickým výběrem dat, což se projevuje na jeho kvalitě.

Algoritmus hledání hran přímo hledá významné body a dosahuje velké věrohodnosti modelu ku počtu zachovaných bodů. Tyto algoritmy jsou často používané ve stereovizi při zpracovávání obrázků, ale existují i varianty pro mračna bodů. K nalezení hran je třeba odhadnout normálové vektory bodů, pomocí nichž se hledají pravé a ostré úhly, které jsou pak klasifikovány jako hrany. Algoritmus není efektivní při použití na hladkých a kulatých objektech, protože nemají skoro žádné ostré hrany. [25]

Prostorové filtry

Prostorové filtry se zaměřují na potlačení šumu analýzou svého okolí. Jednoduché filtry aplikují metody, jako je klouzavý průměr, který zprůměruje polohu aktuálního bodu s body ze svého okolí, nebo obousměrný exponenciální klouzavý průměr, který přiřazuje větší váhu bližším hodnotám. Oba zmíněné filtry vyhladí šum měření, ale zároveň vyhladí ostré hrany, které se ve scéně objevují. Hrany jsou důležitou součástí 3D objektů a je třeba zachovat jejich tvar pro úspěšnou rekonstrukci. Tento problém může být vyřešen omezením okolí klouzavého průměru pouze na ty body, jejichž rozdíl s aktuální hodnotou je menší než stanovená mez. Vzhledem k tomu, že chyba měření roste s kvadrátem vzdálenosti objektu od skeneru, musely by se parametry filtru měnit v závislosti na aktuální vzdálenosti. Tento problém lze vyřešit převodem vzdálenosti na disparitu, což je převrácená hodnota vzdálenosti. [26]

Prostorové filtry se také mohou zabývat vyplňováním děr, které se při měření objevují. Zda-li je žádoucí díry vyplnit záleží na dané aplikaci, špatný odhad může způsobit zanesení nepravdivé informace, která způsobí více škod než užitku. Prostorové filtry se snaží díry vyplnit na základě svého aktuálního okolí, například průměrem bezprostředních sousedů, nebo vybráním nejmenší hodnoty ze svého okolí. [26]

Dočasné filtry

Dočasné filtry plní stejné funkce jako prostorové, ale využívají data z minulosti. Průměrováním v čase lze potlačit výskyt šumu měření, ale také vyplnit nově vzniklé díry, které v minulých snímcích nebyly pozorovány. Problémy s průměrováním v čase vznikají při pohybu skeneru nebo objektu. V tomto případě nejsou jednotlivé body na stejném místě jako předtím. Tento problém lze potlačit stejnými principy popsanych u prostorových filtrů. Druhou možností je odhadovat pohyb scény ze snímků, ale jedná se o výpočetně náročnější variantu. [26]

RANSAC

Random Sample Consensus (RANSAC) je metoda pro odhad parametrů modelu z dat s velkým procentuálním zastoupením outlierů. Princip této metody spočívá ve výběru modelu, který mají data reprezentovat, a poté se náhodně vybere minimální počet bodů definující daný model (například rovina je definována třemi body). Pro náhodně vybrané body se vypočítají parametry odhadovaného modelu a všechny ostatní body jsou vůči tomuto modelu porovnány. Pokud je Euklidovská vzdálenost bodu od modelu

větší než stanovené ε , tak je bod označen jako outlier. Tento proces se opakuje, dokud se nenaleznou parametry modelu, pro které bude procento zastoupení outlierů menší než ρ , nebo se nedosáhne maximálního počtu iterací N . Žádný přesný postup jak volit parametry ε , ρ a N neexistuje. Spousta adaptací RANSAC algoritmu se snaží optimalizovat volbu těchto parametrů a samotný algoritmus s cílem dosáhnout rychlejší konvergence k řešení. [27]

V oblasti 3D skenování se RANSAC algoritmus používá zejména k detekci základních tvarů v objektu a ke sledování výrazných bodů při pohybu skeneru. Při detekci geometrických tvarů se objekt rozkládá na základní struktury, jimiž jsou rovina, koule, kvádr, kužel a další. Klasifikace částí objektu pomocí RANSAC algoritmu se využívá například v robotických aplikacích. Při pohybu skeneru je třeba odhadnout transformační matici mezi snímky, to lze nejnadhěji provést sledování výrazných bodů. Algoritmus pro odhad transformační matice se snaží minimalizovat vzdálenost mezi odhadnutými párovými body mezi snímky. Přítomnost nepravdivých párů, outlierů, značně ovlivňuje přesnost tohoto odhadu a je pro správný odhad je třeba outliery odstranit. SLAM, SfM a další systémy využívají k tomuto účelu RANSAC algoritmy, protože poskytují relativně velkou robustnost vůči jejich rychlému výpočtu. [27]

Statistické filtry

Statistické filtry se zaměřují na odstranění outlierů za pomoci statistických veličin. Nejčastěji se outliery určují podle Euklidovské vzdálenosti od regresního modelu, nebo od ostatních bodů. Typicky používané statistiky jsou střední hodnota, medián a směrodatná odchylka vzdáleností. Při hledání regresního modelu, například metodou nejmenších čtverců [28], jsou pak za outliery považovány body, jejichž vzdálenost nevyhovuje dané statistice. Například vzdálenost od modelu je větší než dva a půl násobek směrodatné odchylky. Statistické určování outlierů může být provedeno i bez modelu, v tomto případě se body rozdělují do shluků. Pro opravdová data platí, že jednotlivé body tvoří jeden kompaktní shluk, v němž se vyskytuje velké množství bodů [28]. Oproti tomu outliery tvoří menší shluky, jež bývají od ostatních shluků vzdálené [28]. K určení outlierů je třeba stanovit minimální počet bodů definující pravý shluk a také určit minimální vzdálenost, která odděluje rozdílné shluky.

Statistické filtry oproti RANSAC algoritmu používají pro své výpočty všechny dostupná data, což je dělá pro velké množství dat časově náročné. Z tohoto důvodu jsou implementovány statistické filtry na bázi okolí, kde je problém určení outlierů rozdělen na několik lokálních problémů.

2.3.2 Registrace a integrace dat

Registrace Point Cloudových dat se zabývá hledáním transformační matice mezi dvěma částečně se překrývajícími mračky bodů. Ze znalosti transformačních matic pro všechny pořízené data lze složit, integrovat, jednotlivá data do jednoho celku tvořící model objektu. Mračna bodů mohou být pořízená buď ze stejného 3D skeneru, nebo z vícero 3D skenerů s jinými parametry. Registrace bodů pořízených z různých skenerů je náročnější, protože každý skener obsahuje jiný šum měření a Point Cloudy jsou často různě husté [29]. Metody registrace dat lze rozdělit na tři hlavní skupiny: optimalizace, učení vlastností a end-to-end učení [29]. Poslední dvě zmíněné metody využívají hlubokých neuronových sítí k natrénování registračního modelu. Použití neuronových sítí nabízí velkou přesnost a rychlost registrace, ale objevují se zde typické problémy, jako

je například selhání modelu na datech velmi odlišných od těch trénovacích a potřeba velkého množství trénovacích dat [29]. Tato kapitola je zaměřena na optimalizační metodu iterativního hledání nejbližších bodů, anglicky Iterative Closest Point (ICP), další metody jsou uvedeny v článku [29].

Iterative Closest Point

ICP je iterativní algoritmus pro hledání transformační matice mezi dvěma mračny bodů. Je řazen do optimalizačních metod, neboť postupně minimalizuje zvolenou metriku. ICP algoritmus se skládá ze dvou hlavních částí. V první části se hledají odpovídající si body a v druhé části se počítá transformační matice mezi Point Cloudy [29]. Pokud by jedna z částí byla známá perfektně, tak řešení druhé části by bylo triviální [29]. ICP, jako každý iterativní proces, vyžaduje spouštěcí parametry, konkrétně odhad transformační matice. Odhad lze získat například algoritmy globální registrace, které jsou obtížnější variantou problému registrace, protože nemají k dispozici počáteční odhad [30]. Pokud data pořídil mobilní robot, lze počáteční odhad získat pomocí jeho odometrie (odhad polohy v čase z různých snímačů) [18]. Výsledek ICP algoritmu silně závisí na počátečním odhadu transformační matice. Ze špatného odhadu může algoritmus dojít pouze k lokálně optimálnímu řešení, jež je daleko od globálně optimálního.

První část ICP algoritmu páruje body z prvního snímku na odpovídající body z druhého snímku. Hledání párových bodů je obtížné, protože se snímky překrývají jen částečně, a tudíž některé body nemají párový bod už z principu. Druhá potíž je, že rozdílná hustota bodů na stejném místě způsobí, že jednomu bodu je přiřazeno vícero párů. Poslední komplikaci v této části způsobuje šum měření, kvůli němuž se mohou vytvořit špatné páry. [29]

Základní metoda určování párů vychází z minimální Euklidovské vzdálenosti bodů mezi snímky [31]. Nechtě $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ je množina bodů transformovaného Point Cloudu a $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M\}$ je množina bodů výchozího Point Cloudu, body $\mathbf{p}_i, \mathbf{q}_j \in \mathbb{R}^3$, matice $\mathbf{R}(k) \in \mathbb{R}^{3 \times 3}$ je rotační matice mezi \mathbf{p}_i a \mathbf{q}_j v kroku k a $\mathbf{t}(k) \in \mathbb{R}^3$ je translační vektor mezi \mathbf{p}_i a \mathbf{q}_j v kroku k . Pak je k bodu \mathbf{p}_i vybrán jeho párový bod $\hat{\mathbf{q}}_i$ v koku k podle rovnice (2.1).

$$\hat{\mathbf{q}}_i = \underset{\mathbf{q}_j}{\operatorname{argmin}} \|\mathbf{R}(k)\mathbf{p}_i + \mathbf{t}(k) - \mathbf{q}_j\|^2 \quad (2.1)$$

Často se matice rotace a vektor translace zapisují pomocí jedné transformační matice $\mathbf{T} \in \mathbb{R}^{4 \times 4}$, viz rovnice (2.2). Pokud jsou snímky pořízeny z různých 3D skenerů, tak je třeba měnit i parametr měřítka v pravém dolním rohu matice \mathbf{T} , pro stejné skenery je roven jedné [29].

$$\mathbf{T} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (2.2)$$

V článku [31] jsou popsány i další způsoby párování bodů a následně mezi sebou porovnány. Ve zmíněném článku bylo experimentálně dokázáno, že pouze metoda párování podle nejmenší Euklidovské vzdálenosti mezi body konverguje k optimálnímu řešení, avšak hledání minimální vzdálenosti pro každý bod je výpočetně náročné. Proto se v případech, kdy jsou Point Cloudy málo odlišné, vyplatí používat jiné, daleko rychlejší metody. [31]

Určování párů podle nejmenší vzdálenosti je typický algoritmus typu každý s každým, neboli má výpočetní složitost $\mathcal{O}(n^2)$. Doba výpočtu roste s kvadrátem počtu bodů

a je tedy logické, že zmenšením objemu dat se značně zvýší rychlost algoritmu. Výběr podmnožiny bodů z celého Point Cloudu lze udělat náhodně, rovnoměrně a nebo systematicky, například že výběr bude obsahovat co nejvíce bodů s různými normálovými vektory, nebo body, jejichž intenzita barev se výrazně mění [31]. Cíl systematických metod výběru podmnožiny je redukovat počet bodů tak, aby byly stále zachovány podstatné rysy objektu a zároveň tím nebyla obětována výpočetní rychlost.

Kvalitu odhadu párových bodů a rychlost konvergence ICP algoritmu lze zlepšit exkluzí některých nalezených párů a případně i jejich vážením. Nejjednodušší způsob vyřazení potenciálně mylných párů je stanovení prahu maximální vzdálenosti mezi párovými body. Jelikož se Point Cloudy ve většině případů překrývají pouze částečně, vytvoří se spousta nepravdivých párů na krajích mračen. Vyloučení těchto párů vede téměř vždy k zlepšení kvality odhadu. Přiřazování vah jednotlivým párům ovlivňuje celkový výsledek pouze nepatrně a s ohledem na rychlost algoritmu se vyplatí váhy nepřiiřazovat. [31]

Druhá část ICP algoritmu používá nalezené páry v první části k výpočtu transformační matice, která minimalizuje zvolenou metriku podobnosti. Metrika podobnosti bývá většinou nějaká funkce Euklidovské vzdálenosti v prostoru [31]. Vyplatí se ale využívat i barevné složky bodů k výpočtu druhé metriky, například jako kvadrát rozdílů intenzit párových bodů [30]. V tomto případě jsou metriky kombinovány v určitém poměru. ICP algoritmy se typicky dělí podle zvolené minimalizační metriky. Nejrozšířenějšími metrikami jsou vzdálenost mezi jednotlivými body (point-point), nebo vzdálenost bodu od tečné roviny (point-plane) [29, 31].

Nechť \mathcal{L} je množina L párových bodů $\mathbf{p}_i, \hat{\mathbf{q}}_i$ v kroku k , \mathbf{T} je transformační matice skládající se z rotační matice \mathbf{R} a translačního vektoru \mathbf{t} . Pak výpočet transformační matice pomocí point-point metriky je dán rovnicí (2.3).

$$\mathbf{T}(k+1) = \operatorname{argmin}_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^L \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \hat{\mathbf{q}}_i\|^2 \quad (2.3)$$

Dále nechť \mathbf{n}_{q_i} je normálový vektor tečné roviny příslušící bodu $\hat{\mathbf{q}}_i$. Pak výpočet transformační matice pomocí point-plane metriky je dán rovnicí (2.4).

$$\mathbf{T}(k+1) = \operatorname{argmin}_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^L \|\mathbf{n}_{q_i} \cdot (\mathbf{R}\mathbf{p}_i + \mathbf{t} - \hat{\mathbf{q}}_i)\|^2 \quad (2.4)$$

Tyto metriky vedou na použití metod nejmenších čtverců pro nalezení řešení. Výhodou point-point metriky je, že její řešení lze nalézt přímo, například metodou singularního rozkladu (Singular Value Decomposition - SVD). Na druhou stranu point-plane metrika nemá přímé řešení a k jeho nalezení se používá linearizace nebo Levenberg–Marquardtův algoritmus. Ačkoliv nalezení řešení point-plane metriky je obtížnější, jejím použitím se dosáhne rychlejší konvergence ICP algoritmu k řešení, které bývá i blíže ke globálnímu optimu, ve srovnání s point-point metrikou. [31]

Spojením point-point, point-plane a plane-plane metrik se vytvořila takzvaná generalizovaná forma ICP algoritmu [29]. Zde se plně využívá informace o normálách bodů, což se přibližuje faktu, že skenovaný objekt je tvořen plochami a ne diskretními body [29]. Většina moderních 3D skenerů je již schopna pořídit mračna bodů i s barvami a jedná se tak o další informaci, kterou lze využít k nalezení optimálnějšího řešení [30]. Informace o barvě výrazně zlepšuje výsledek pro rovné plochy bez výrazných rysů, které by jinak nebylo možné přesně na sebe registrovat [30].

2.3.3 Rekonstrukce povrchu

Metody pro vytvoření povrchu z Point Cloudu se dají rozdělit na mnoho typů, zde budou popsány tři rozvinuté typy, které se objevují v literatuře. První typ metod vychází z Delaunayovy triangulace a vytváří síť trojúhelníků přímo z naměřených bodů. Druhý typ metod hledá implicitní funkce, které rozdělují povrch na vnitřek a vnějšek. Třetí typ metod parametrizuje křivky, aby co nejlépe aproximovaly data.

Delaunayova triangulace

Zástupci prvního typu metod jsou algoritmy vycházející z Delaunayovy triangulace, která vytváří trojúhelníky z bodů, jejichž opsaná kružnice neobsahuje žádné jiné body. Ve 3D tvoří trojúhelníky čtyřstěny jejichž opsaná koule neobsahuje žádné jiné body. Body jsou postupně propojovány čtyřstěny, které sdílí stěny se svými sousedy. Nakonec se celý model zabalí do takzvaného konvexního obalu tvořeného těmito čtyřstěny. Avšak většina stěn neleží na povrchu modelu a pro 3D rekonstrukci je tento základní algoritmus upravován. Alpha Shapes a Ball Pivoting jsou algoritmy zastupující kategorii Delaunayovy triangulace. Alpha Shapes nejprve zabalí model do konvexního obalu a pak odstraňuje ty čtyřstěny, jejichž vepsaná koule má větší poloměr než uživatelem zvolený parametr α . Pokud se α zvolí dobře, zbudou jen stěny na povrchu modelu, ale často nelze podchytit všechny detaily objektu. Ball Pivoting na druhou stranu omezuje poloměr ρ opsané koule již při tvorbě trojúhelníkových stěn. Tento algoritmus funguje na principu, že pokud by koule nepropadla mezi třemi body, tak se tam vytvoří trojúhelníková stěna. [32]

Algoritmy vycházející z Delaunayovy triangulace fungují nejlépe, pokud jsou data rovnoměrně rozložena. V opačném případě jsou v povrchu díry, nebo nejsou data reprezentována věrohodně. Jak algoritmy dopadnou také záleží na parametru, který volí uživatel. Parametr je typicky volen náhodně a ze všech vygenerovaných modelů se ručně vybere ten nejlepší. Jelikož tyto algoritmy pracují s naměřenými body přímo, je třeba použít nějaký vyhlazovací proces před nebo po použití těchto algoritmů.

Některé algoritmy přerozdělí naměřená data do více rovnoměrného rozdělení, aby se vylepšil proces Delaunayovy triangulace. Typicky se data rozdělí do voxelů a buď se body uvnitř nahradí jedním bodem, nebo se vytvoří nové body tak, aby mezi všemi byla přibližně stejná vzdálenost a zároveň se zachovala struktura povrchu. [33]

Implicitní funkce

Implicitní funkce jsou funkce více proměnných definované jako $f(x,y,z) = 0$. Tam, kde je funkce nulová, je povrch modelu a podle znaménka se pak rozhoduje, zda se bod nachází vně nebo uvnitř. Nalezení jedné implicitní funkce pro celý model by bylo příliš náročné, protože obecný model může mít libovolně složitý tvar. Proto se implicitní funkce definují po částech, které se pak spojí tak, aby na sebe hladce navazovaly. Typicky je potřeba k výpočtu implicitní funkce znát normálové vektory daných bodů a shodně je orientovat [33].

Výsledné, po částech definované implicitní funkce aproximují naměřená data a jsou robustní vůči šumu měření. Na druhou stranu aproximace vyhledá lokální detaily. Kvalita výsledného modelu těchto algoritmů je vysoce podmíněna kvalitou odhadu a orientace normálových vektorů. [33]

Parametrizované křivky

Převod z Point Cloudu na parametrizované plochy lze provádět dvěma způsoby: interpolací a aproximací. Interpolace se snaží najít křivky procházející všemi pozorovanými body, kdežto aproximace se snaží najít křivku nejlépe vystihující pozorovaná data. V aplikacích 3D skenování se kvůli velkému množství dat, které jsou zároveň kontaminována šumem měření, používají k vytvoření povrchu výhradně aproximační techniky [22]. Velké popularity dosáhly B-Spline a NURBS parametrické křivky a plochy, kvůli možnosti lokálně měnit jejich tvar. B-Spline křivka je definována bázovou funkcí, stupněm polynomu, kontrolními body a uzly rozdělující křivku. NURBS křivka pak přidává váhy jednotlivým kontrolním bodům a uzly jsou nerovnoměrně rozloženy.

Parametrické plochy se skládají ze sítě parametrických křivek a jejich vytvoření je o něco složitější, pro jednoduchost zde bude popsán pouze proces parametrizace křivek. Proces přizpůsobení NURBS křivky na změřený Point Cloud začíná zvolením některých parametrů podle naměřených dat a zbylé parametry se získají optimalizačním procesem. Nejprve se zvolí stupeň polynomu křivky, který ovlivňuje počet kontrolních bodů a jak moc se křivka přizpůsobí naměřeným datům. Na základě vzdáleností mezi body se odhadne počet a rozmístění uzlů křivky. Ze všech dosavadních informací se vypočítají optimální pozice kontrolních bodů a jejich váhy, například metodou nejmenších čtverců. Tento proces je typicky iterační, kdy se zkouší volit různé stupně polynomů a metody rozmístění uzlů a získané modely se pak vůči sobě porovnávají. [21]

2.4 Vyspělé systémy 3D rekonstrukce

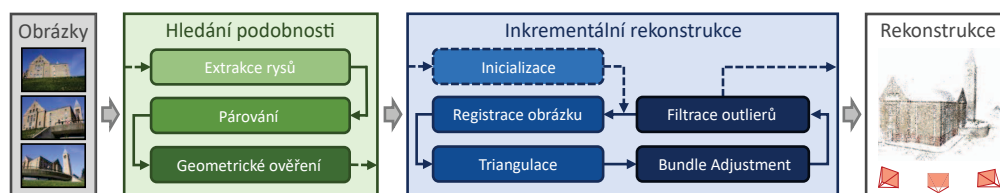
Tvorba systému pro 3D rekonstrukci začíná prvotní volbou snímače či snímačů. Rozhoduje se hlavně podle dané aplikace. Poté je třeba zvolit způsob reprezentace modelu a algoritmy pro jejich zpracování. Jejich výběr ovlivňuje kvalitu rekonstruovaného modelu a rychlost samotné rekonstrukce, často jdou tyto dva požadavky proti sobě. Vyspělé systémy obsahují i nějakou formu samoopravy, která zabrání driftování rekonstruovaného modelu od reálné podoby. Drift je způsobený akumulací chybných odhadů a propagací šumu měření. V této kapitole jsou uvedeny dvě praxí podložené formy pro tvorbu robustních systémů 3D rekonstrukce.

2.4.1 Structure from Motion

Structure from Motion (SfM) je systém pro rekonstrukci 3D modelu z neuspořádaných obrázků zachycujících stejný objekt z různých úhlů pohledu. Obrázky mohou být pořízeny z kalibrovaných i nekalibrovaných kamer a fotoaparátů. Systém SfM lze rozdělit na dvě hlavní části: hledání odpovídajících si obrázků a inkrementální rekonstrukce modelu, viz obrázek 2.10. [2]

První část SfM hledá a páruje obrázky s překrývající se projekcí objektu/scény. Nejprve se extrahují body s výraznými rysy z každého obrázku. Poté následuje párování obrázků na základě podobnosti jejich bodů. Nakonec je ověřena geometrická věrohodnost jednotlivých párů. Pár obrázků je považován za geometricky ověřený, pokud je nalezena transformace mapující dostatečný počet bodů na sebe. Na konci této části je scéna popsána grafem, kde obrázky jsou vrcholy a jejich překrytí je popsáno hranami. [2]

Druhá část SfM využívá výstupu první části k rekonstrukci 3D modelu z 2D obrázků. Proces rekonstrukce je inkrementální, to znamená, že se model postupně rozrůstá. Takový proces je třeba nejprve inicializovat vhodně vybraným párem obrázků, z něhož bude vytvořen základ modelu. Z registrovaných párů se vybere nejlepší kandidát na rozšíření modelu a na základě odhadované transformace z jednoho obrázku do druhého se odhadne póza a vnitřní parametry kamery, která daný obrázek pořídila. Pózou kamery se rozumí její poloha a natočení v prostoru, neboli šest stupňů volnosti. Znalost odhadnutých póz kamer umožní provést triangulaci překrývajících se pixelů (bodů) a určit jejich polohu v prostoru. Každé nové pozorování jednoho bodu zvyšuje redundantnost jeho odhadu polohy. Minimalizací projekční chyby vycházející z redundantních pozorování se zabývá Bundle Adjustment (BA). Bez BA se velmi rychle akumuluje chyba odhadu a rekonstruovaný model driftuje od reálné podoby. Dalšího zpřesnění odhadu polohy lze docílit filtrací outlierů při registraci nového obrázku. Výstupem této části je mračno bodů reprezentující model objektu/scény. [2]



Obrázek 2.10: Postup metody Structure from Motion [2]

2.4.2 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) je systém pro odhadování polohy robota v prostoru pomocí mapy, kterou si robot sám vytváří prozkoumáváním svého okolí. Tato technika je zaměřená především na zpracovávání dat ze sensorů k tvorbě 3D mapy, plánování pohybu autonomního robota tento systém neřeší. Problém SLAMu lze rozdělit na dvě části. První část asociuje právě získaná data s mapou okolí a postupně ji tak rozšiřuje. Druhá část se zabývá odhadem reálné polohy robota a pozorovaných objektů na základě aktuálního stavu robota a nových měření. Úlohou druhé části je napomáhat první části při vytváření ucelené mapy, protože bez apriorní znalosti struktury mapy si robot myslí, že je svět jeden dlouhý koridor. Rozpoznání dříve navštívených míst umožní vytvořit uzavřené smyčky z onoho zdánlivě nekonečného koridoru a zachytit tak skutečnou topologii svého okolí. [18]

SLAM je do jisté míry rozšířený proces SfM, kde oproti SfM není limitován pouze na zpracování snímků z kamer, ale využívá navíc měření z dalších sensorů, jako jsou inerciální pohybový sensor, LiDAR nebo odometrie z kol robota, které napomáhají při odhadu aktuální polohy robota. Druhým rozdílem SLAMu od SfM je, že mapa se rozšiřuje inkrementálně tak, jak robot prozkoumává okolí, kdežto SfM má k dispozici celou kolekci snímků už od začátku. [18]

SLAM využívají především autonomní roboti v prostorech, kde není předem známá mapa okolí, v které má robot operovat. Robot si musí mapu okolí vytvořit za běhu, aby se v ní mohl lokalizovat a vykonat svou práci. Za tímto účelem je vybaven kombinací sensorů, jež dohromady zvyšují robustnost odhadu polohy. SLAM se hodí k prozkoumávání budov a uzavřených prostorů, kde bývá dostatek výrazných artefaktů,

pomocí kterých se robot orientuje. Pro venkovní aplikace je z důvodu řídkého počtu objektů výhodné použít k odhadu polohy i GPS. [18]

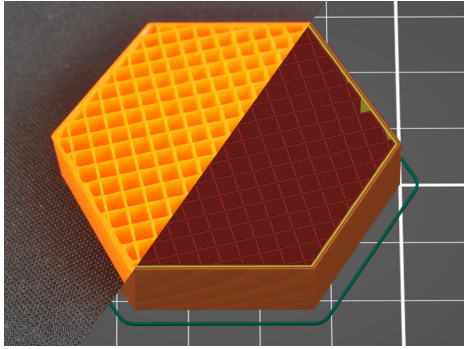
2.5 Zpracování rekonstruovaného modelu pro 3D tisk

3D tisk je forma aditivní výroby, která přidává materiál po vrstvách. 3D tisk byl původně vyvinut pro rychlé prototypování součástí, ale s vývojem technologií 3D tisku došlo ke zlepšení vlastností tisknutých součástí [34]. Oproti běžné subtraktivní výrobě, kdy je materiál postupně odebírán například CNC obráběním, je 3D tisk charakteristický rychlejší a modulárnější výrobou a menší materiálovou ztrátou. Na druhou stranu běžné 3D tiskárny používají plastové materiály a vyrobené součásti nejsou tolik odolné jako při výrobě sléváním a obráběním kovů, avšak pro některé účely je kvalita dostačující a vytisknuté součásti se používají jako plnohodnotné výrobky [34].

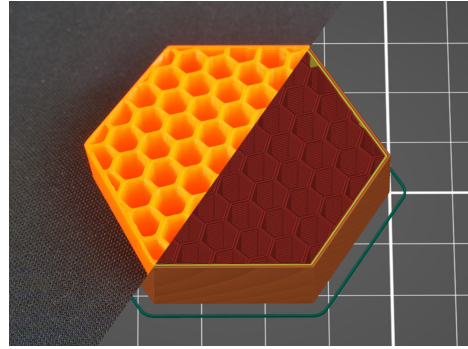
Mezi nejrozšířenější typy 3D tisku patří stereolitografie (SL) a metoda taveného nanášení (Fused Deposition Modelling - FDM). SL vytváří model z tekutého fotopolymery (UV resin), který pod září UV laseru tvrdne. Po vytisknutí se součást vyjme z nádrže, usuší a vloží do UV pece pro finální ztvrdnutí materiálu. FDM zahřívá a tlačí termoplast zvaný filament skrze trysku, který po nanesení rychle tvrdne. Filament je nanášen na vyhřívanou podložku, aby se jednotlivé vrstvy snáze spojily. Pro běžné účely se používá filament z kyseliny polymléčné (PLA) nebo akrylonitrilbutadienstyrenu (ABS). Součásti vyrobené technikou FDM mají vidět jednotlivé vrstvy naneseného filamentu, a proto je jejich povrch následně vyhlazován. Jelikož oba procesy tvoří součásti po vrstvách, vyžadují, aby součástí modelu byly nosné podpory, které zpevní převislé stěny a dočasně vznášející se části během tisku. Podpory jsou po vytisknutí odříznuty. Součásti vyrobené pomocí SL mají hladký povrch a tisk je mnohem rychlejší než při použití FDM. Na druhou stranu FDM tiskárny mají levnější a odolnější materiály pro tisk. [34]

Aby mohla 3D tiskárna model vytisknout, je potřeba model nahrát do CAM softwaru (Computer Aided Manufacturing), který převede model na G kód, což jsou instrukce pro danou 3D tiskárnu. Jelikož je 3D tisk forma aditivní výroby po vrstvách, jsou CAM softwary pro 3D tisk také nazývány jako slicery. Slicery přijímají soubory ve formátech STL (Standard Tessellation Language), OBJ (Object), AMF (Additive Manufacturing) a 3MF (3D Manufacturing Format) [35]. Tyto soubory reprezentují model pomocí trojúhelníků, což v procesu 3D skenování znamená, že je potřeba rekonstruovat povrch modelu.

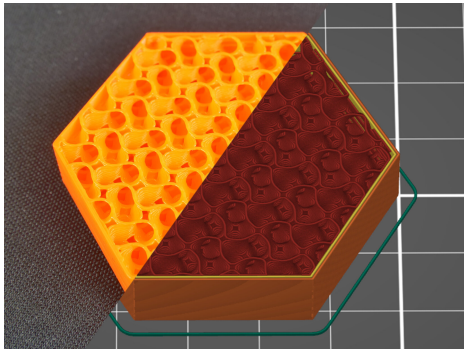
Slicery kromě generování G kódu poskytují i další funkce, jako jsou automatické generování nosných podpor, úprava parametrů tisknutého filamentu a úprava pohybu nástroje. Typické parametry, které uživatel specifikuje při použití FDM tiskárny, jsou teplota trysky a podložky v závislosti na použitém filamentu, výška a šířka naneseného filamentu, šířka vertikálních stěn a výška horizontálních stěn. Dále si uživatel může zvolit z kolika procent má být model vyplněn a vzor výplně. Vzory mohou být trojúhelníkové, čtvercové nebo šestiúhelníkové mřížky, nebo to mohou být neprotínající se křivky, jako je Hilbertova křivka, a další. Některé příklady vzorů výplně poskytnuté v *PrusaSliceru* jsou zobrazeny na obrázku 2.11. Všechny tyto volitelné parametry ovlivňují mechanické vlastnosti vytisknutého modelu, rychlost tisku a množství spotřebovaného filamentu. [36]



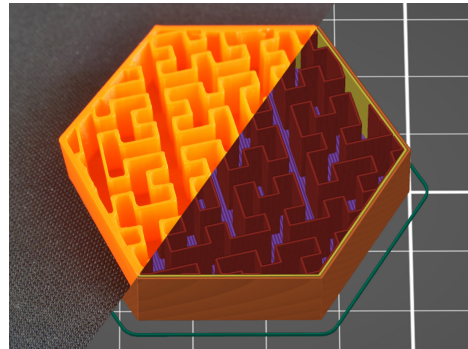
(a) Rektilineární mřížka



(b) Plástev



(c) Gyroid



(d) Hilbertova křivka

Obrázek 2.11: Příklady vzorů pro výplň 3D modelu v PrusaSliceru

Kapitola 3

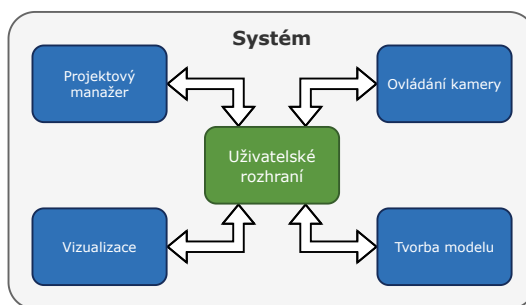
Návrh systému

Tato kapitola popisuje, jak probíhal návrh systému pro tvorbu počítačového modelu skutečného objektu a následně jsou prezentovány dosažené výsledky. Systém byl vytvářen se zaměřením na zpracovávání dat od jejich pořízení ze 3D skeneru až po vytvoření modelu vhodného pro 3D tisk.

Pro akvizici dat byla zvolena hloubková kamera Intel RealSense D435i, která poskytuje snímky v RGB-D formátu. Vývojáři této kamery vytvořili k její obsluze sadu instrukcí v programovacích jazycích C++, MATLAB, ROS, Python a další. Pro zpracování dat se vybrala open source knihovna *open3d*, která poskytuje své instrukce v C++ a v balíčku pro Python. Vzhledem k rozsahu práce a cíli prozkoumat základní koncept 3D rekonstrukce, se pro vývoj systému zvolil jazyk Python, protože je snáze pochopitelný. Je nutné podotknout, že Python je oproti C++ interpretovaný jazyk a je tedy známo, že programy v Pythonu běží trochu pomaleji. Pro návrh systému pro komerční použití se zaměřením na rychlost programu by bylo vhodné použít jazyk C++, neboť algoritmy spojené s 3D rekonstrukcí jsou obecně výpočetně náročné.

Navržený systém funguje tak, že kamera je připojena k notebooku, kde běží program pro zaznamenávání videa, a skenovaný objekt je položený na pevné podložce. Uživatel s kamerou v ruce musí objekt obejít kolem dokola a zaznamenat jednu nepřetržitou nahrávku. S objektem nesmí být během nahrávání nijak manipulováno, neboť je uvažováno, že je scéna statická. Požadavek na nepřetržitou nahrávku bez manipulace objektem způsobuje, že není možné skenovat objekt zespoda.

Součástí této práce je i uživatelská aplikace zapouzdřující veškeré skripty pro ovládní kamery, vytváření a ukládání nahrávek, zpracování naměřených dat a jejich vizualizaci do jednoho celku. Aplikace byla vytvořena se záměrem zpřístupnit výsledky této práce lidem, kteří by chtěli systém pro 3D rekonstrukci použít, ale nejsou znalí programování. Aplikace byla vytvořena pomocí vývojového balíčku *PySide6*, což je distribuce multiplatformní vývojové aplikace *Qt* pro tvorbu uživatelského grafického rozhraní. Na obrázku 3.1 je vidět vnitřní topologie navrženého systému.



Obrázek 3.1: Vnitřní topologie navrženého systému pro tvorbu počítačového modelu reálného objektu

3.1 Popis hloubkové kamery Intel RealSense D435i

Hloubková kamera Intel RealSense D435i je 3D skener vyráběný společností Intel. Jedná se o jednu z cenově dostupných kamera na trhu, která poskytuje vysokou kvalitu měření v porovnání s její cenou. Často se kvůli těmto vlastnostem využívá v robotických aplikacích. [37]

Kamera se skládá z mikroprocesoru, jedné RGB kamery, dvou infračervených (IR) kamer a IR laserového projektoru strukturovaného vzoru. Její základní koncept měření vzdálenosti vychází ze stereo vize. Snímky z obou IR kamer jsou pořízeny ve stejný okamžik a na vestavěném mikroprocesoru je vypočítána vzdálenost jednotlivých pixelů. Jako reference se používá levá IR kamera, protože je blíže k RGB kameře, a pravá slouží k přiřazení vzdálenosti. Toto je třeba brát v potaz, protože okluze části objektu způsobí, že pixely za hranami na levé straně nemusí mít přiřazenou vzdálenost, neboli vzniknou díry [26]. IR projektor je zde využit k obohacení textury plochých objektů, které by jinak bylo téměř nemožné rekonstruovat pouze z obrázků. Na konec je hloubkové měření z levé IR kamery spojeno se snímkem z RGB kamery a je vytvořen RGB-D obrázek. [37]

Kamera používá globální závěrku (Global Shutter), to znamená, že všechny pixely kamery jsou pořízeny ve stejný okamžik. Oproti rolovací závěrce má výhodu při pořizování snímků dynamických objektů, protože nedojde ke zkreslení vlivem postupného snímání jednotlivých pixelů [37]. Použití IR kamer umožňuje měřit vzdálenost i ve špatném osvětlení. Maximální rozlišení IR kamer je 1280×720 pixelů se zorným polem (Horizontální \times Vertikální \times Diagonální): $85^\circ \times 58^\circ \times 90^\circ$. Rozsah měřitelné vzdálenosti je 0,2 až 10 metrů, ale největší přesnosti dosahuje do vzdálenosti jednoho metru, kdy je chyba měření menší než 5 milimetrů [10]. RGB kamera má maximální rozlišení 1980×1080 pixelů a zorné pole o velikosti (H \times V \times D): $91^\circ \times 65^\circ \times 100^\circ$. V závislosti na rozlišení kamer se odvíjí i maximální snímková frekvence, respektive při největším rozlišení je možné pořídit pouze 6 snímků za vteřinu. Maximálně pak RGB kamera umožňuje pořídit snímky s frekvencí 60 Hz a IR kamera s frekvencí až 90 Hz, ovšem při rozlišeních 424×240 pixelů a 256×144 pixelů respektive.

Je vhodné podotknout, že pro aplikace 3D skenování za účelem rekonstrukce modelu je mnohem žádoucí mít vysoké rozlišení kamer nežli velké snímkové frekvence, protože vysoké rozlišení umožní zachytit více detailů a proces snímání objektu nemusí být prováděn ve velké rychlosti.

Model kamery D435i má navíc oproti standardní verzi D435 zabudovaný inerciální měřicí jednotku (IMU) s gyroskopem a akcelerometrem. Tato IMU jednotka je vyrobena technologií Micro Electro Mechanical Systems (MEMS). IMU jednotky vyrobené touto technologií jsou známy svou nižší kvalitou měření. IMU jednotka umožňuje odhadnout pózu kamery, ale je nutné filtrovat unášení gyroskopu a odečítat vliv gravitace z akcelerometru. Pro odhad orientace kamery byl využit Madgwickův filtr z knihovny *AHRS*. Odhad měl zrychlit proces registrace dat, ale ukázalo se, že při pomalém pohybu kamery, ke kterému dochází při skenování, je odhad orientace nepřesný a poměrně rychle driftuje od reality. Vzhledem k následující kapitole 3.3.1, kde bude popsán způsob mapování bodů na sebe, nebyl odhad orientace z filtru přínosný a IMU nebyla nakonec využita.

3.2 Pořízení dat z hloubkové kamery

K pořízení dat z kamery Intel RealSense D435i lze využít uživatelské aplikace Intel RealSense Viewer, kde je zároveň vizualizován přenos dat. Lze tam i upravit možné parametry kamery a zvolit způsob filtrace dat v reálném čase. Nahrávky v této aplikaci kromě samotného záznamu hloubky a obrazu obsahují všechny informace o senzoru a jeho nastavení, a navíc má každý snímek své vlastní parametry. Vzhledem k informacím, které navrhovaný systém potřebuje k 3D rekonstrukci, jsou soubory zbytečně velké. Proto byl návrh systému pojat už od samotného nahrávání a ukládání dat. Navržené uživatelské rozhraní ukládá pouze potřebná data do hierarchického uspořádání adresářů formou projektů.

Systém pro ovládání kamery funguje následujícím způsobem. Nejprve se zjistí, jestli je k počítači připojená kamera, která je dále v programu identifikována svým sériovým číslem. Před zahájením přenosu se musí definovat konfigurace přenosu, neboli informace, jaké datové kanály a s jakými parametry otevřít. Datový kanál je definovaný typem informace, například hloubka nebo barva, jeho formátem (pro hloubku z16 a pro barvu rgb8), rozlišením a snímkovou frekvencí. Konfigurace použitá v navrženém systému obsahuje sériové číslo kamery, datový kanál pro hloubku a druhý datový kanál pro barevný obraz. Konfigurace se předá do pipeline, která otevře datové kanály specifikované kamery a zahájí přenos.

Pipeline vrací snímky v pravidelném intervalu a existují tři způsoby jak ke snímkům přistupovat. První způsob vybírá snímky v nekonečné smyčce, která čeká, než přijde snímek a pak ho zpracuje. Nevýhoda tohoto způsobu je, že program během čekání na nový snímek nic nedělá. Druhý způsob je vytvořit si frontu, kam bude pipeline snímky odkládat, a ty se budou vybírat v pravidelném intervalu. Tento způsob odstranil problém s čekáním na snímky, ale zároveň je třeba vytvořit logiku k obsluze fronty. Třetí, nejpohodlnější způsob je přiřadit k pipeline obslužní rutinu, která se automaticky zavolá po příchodu nového snímku a přeruší na chvíli chod hlavní programové smyčky. Výhodou tohoto způsobu je, že není třeba řešit logiku, kdy vyprázdnit frontu. Pro tento systém byl zvolen třetí způsob obsluhy pipeline a čas nestrávený čekáním na nové snímky je využit na vizualizaci dat z kamery, viz kapitola 3.5.

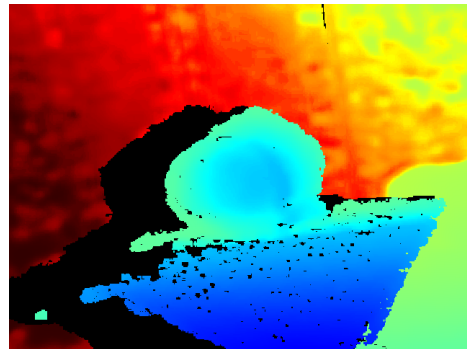
Proces zpracování snímku započne hned po jeho obdržení z pipeline. Navržený systém je nastaven tak, že oba datové kanály mají vždy stejné rozlišení a snímkovou frekvenci, ale nemuselo tomu být tak. RGB kamera a IR kamera se fyzicky nacházejí na odlišných místech, proto se hloubková mapa nejprve spojí s barevným obrázkem. Z důvodu vizualizace se z hloubkové mapy zároveň vytvoří druhý barvený obrázek.

Aktuálně získaná data se uloží do proměnných a tím skončí obslužná rutina pipeline. Když aplikace uzná za vhodné, požádá systému pro ovládání kamery, aby převedl data do Point Cloudu a odstranil body vzdálenější než stanovená mez. Point Cloud a barevné obrázky se vizualizují a předají projektovému manažerovi, aby je uložil. Snímky jako takové se dále nezpracovávají, jsou ukládány spíše pro dokumentaci. Proto jsou ukládány jen jednou za čas.

Při experimentech s použitím IMU jednotky byla zjištěna tato vlastnost pipeline. Jelikož akcelerometr a gyroskop operují s mnohem větší vzorkovací frekvencí, není možné je obsluhovat pomocí stejné pipeline, protože ta vždy čeká na data s nejpomalejší vzorkovací frekvencí. Během čekání na data ze všech datových kanálů dojde ke ztrátě mnoha vzorků. Z toho vyplývá, že každý datový kanál s rozdílnou vzorkovací frekvencí musí mít vlastní pipeline.



(a) RGB obrázek



(b) Obarvená hloubková mapa



(c) Point Cloud z bodů do vzdálenosti 0,5 m



(d) Simulované osvětlení Point Cloudu

Obrázek 3.2: Vytvořený Point Cloud z naměřených dat

3.2.1 Popis scény experimentálního měření

Ke skenování byla zvolena plyšová želva o rozměrech (Délka×Šířka×Výška): $15 \times 13 \times 10$ cm, která má dominantní zelenou barvu, ale obsahuje i černé, bílé a červené prvky. Želva byla umístěna na oranžovou desku podloženou oranžovým válcem pro vyvýšení objektu. Hloubková kamera byla připojena k notebooku a ručně se s ní skenoval objektu kolem dokola, shora a zdola nebyl objekt skenován (díra uprostřed pomůže při validaci algoritmu). Vzhledem k tomu, že proces vytváření modelu nedosahuje bezchybných výsledků, nové měření i s hořejškem objektu nebylo realizováno.

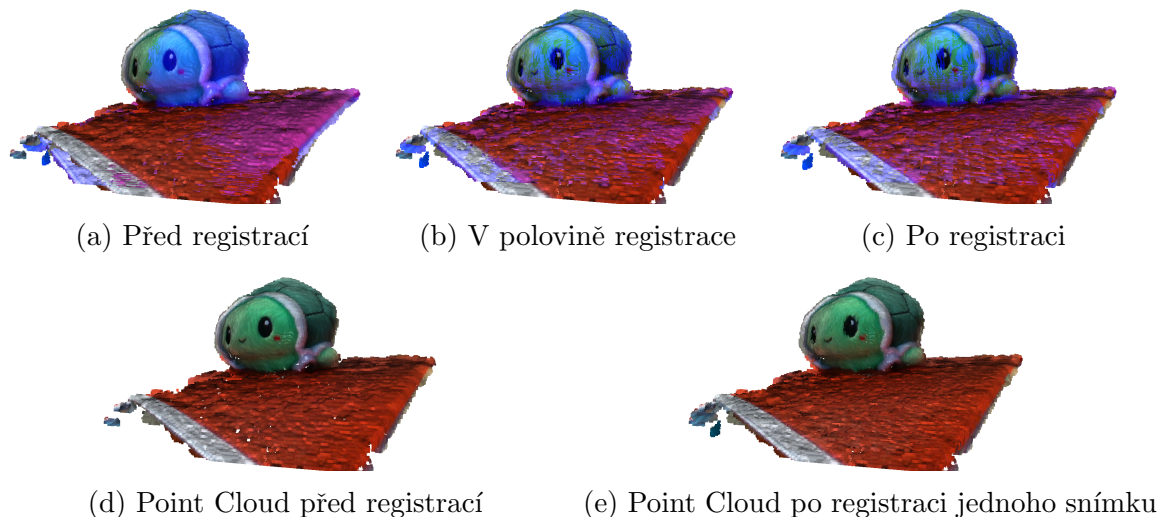
Držení kamery v ruce přináší nové obtíže, zejména otřesy kamery při chůzi a rychlé nerovnoměrné pohyby, které způsobují rozmazání nahrávky. Tímto faktorem bude postihnuta registrace dat za pomoci barevné složky. Objekt je osvětlen ze shora pomocí světelných zdrojů v místnosti a také ze předu denním světlem z okna. Během měření lze pozorovat stín pod objektem a tmavší zbarvení objektu ze zadní strany. Kvůli variaci v

osvětlení dojde opět ke komplikacím při registraci dat pomocí barevné složky. Ačkoliv by bylo potěšující rekonstruovat model objektu i se správným zbarvením, primárním cílem této práce je vytvořit tisknutelný 3D model. Data z nahrávání jsou vidět na obrázku 3.2. Pro natočení nahrávky bylo zvoleno rozlišení RGB a IR kamer 640×480 pixelů se snímkovou frekvencí 15 Hz. Nahrávka se skládá ze 406 snímků.

3.3 Návrh systému pro tvorbu modelu

Celý proces registrace a integrace dat je mnohem složitější než se může zdát. Kromě ladění spouštěcích parametrů pro ICP algoritmus je třeba vymyslet i jak data vhodně předzpracovat, aby program neběžel příliš dlouho a byla zlepšena kvalita rekonstrukce. Typicky se ICP spouští několikrát pro různé hustoty bodů, což vede ke zrychlení konvergence ke správnému řešení. Kolikrát spustit ICP a jako moc data vzorkovat nelze předem stanovit a během návrhu jsou tyto parametry laděny metodou pokus omyl.

Z bodového modelu je třeba odstranit pozadí a outliersy. Objekt zájmu byl v tomto případě vyříznut ručně a k odstranění outlierů se použil statistický filtr. Po těchto úpravách se značně snížil počet integrovaných bodů, což urychlilo výpočet trojúhelníkové sítě. Pro rekonstrukci povrchu byly vyzkoušeny celkem tři algoritmy.



Obrázek 3.3: Průběh ICP algoritmu

3.3.1 Registrace a integrace naměřených dat

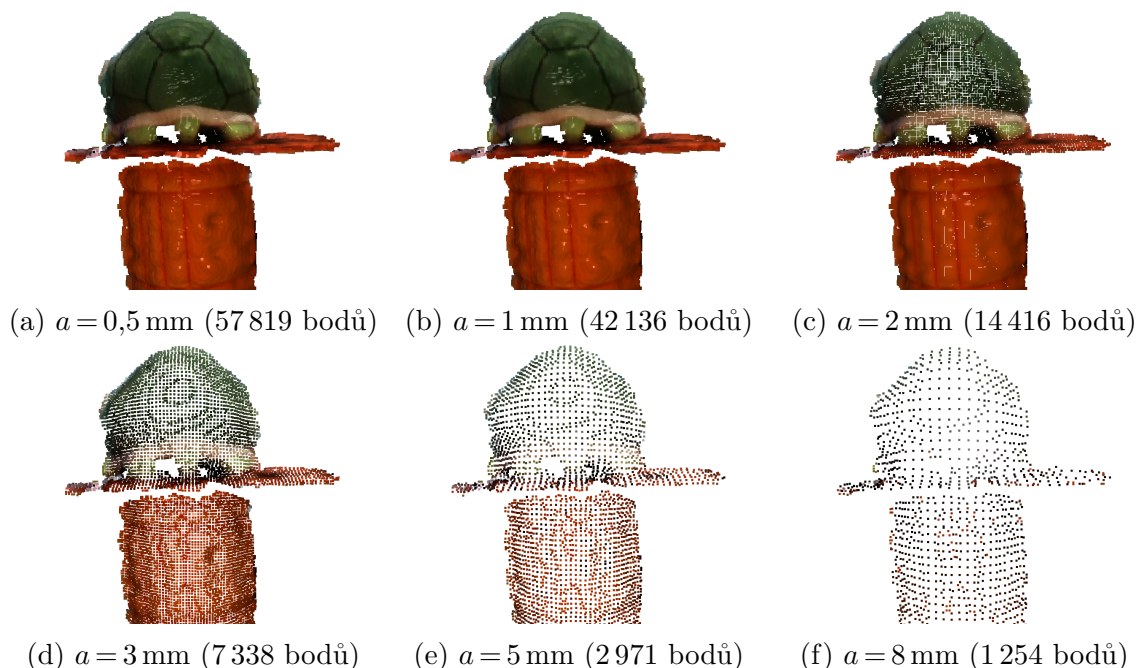
Proces vytváření modelu z naměřených dat začíná registrací a integrací dat. Pro registraci Point Cloudových dat byl zvolen ICP algoritmus s barevnou složkou, který implementuje knihovna *open3d* podle článku [30]. Jak probíhá ICP algoritmus je zobrazeno na obrázku 3.3. Ovšem před samotným spuštěním ICP algoritmu je data třeba vhodně zpracovat, jinak by šum a outliersy znemožnily správnou registraci dat. Analýzou pořízených dat se zpozoroval jev, že na hranách objektů dochází k přetečení pozadí do popředí, neboli body, které jsou ve skutečnosti za hranou objektu, jsou měřeny jako součást hrany. Také byli v několika snímcích zaznamenáni čistě nahodilé outliersy nacházející se daleko od měřeného objektu. Zřejmě došlo k odrazu světla od čočky kamery.

Některé snímky mají i černé body na hranách, kde se černá barva ve skutečnosti nenachází. K výskytu těchto bodů dochází nejspíše vlivem okluze nebo kvality osvětlení v místnosti. Mimo jiné jsou veškeré snímky kontaminovány šumem měření, což bylo očekáváno.

ICP algoritmus vyžaduje spoustu spouštěcích parametrů, které je třeba specifikovat. Nejdůležitějšími parametry je zdrojový Point Cloud, registrovaný Point Cloud a počáteční odhad transformační matice. ICP algoritmus v knihovně *open3d* umožňuje dále nastavit maximální vzdálenost mezi párem odpovídajících si bodů, poměr λ mezi klasickou point-plane metrikou a metrikou intenzity barev, maximální počet iterací, minimální změnu RMSE mezi iteracemi pro ukončení, a minimální shodu Point Cloudů pro ukončení iterací. (Například parametr $\lambda = 0,9$ znamená, že poměr metrik point-plane a intenzity bude 9:1.)

Aby bylo možné algoritmus spustit, musí mít body přiřazené normálové vektory. Ty nejsou z měření dostupné, a proto se musí odhadnout. K odhadnutí normálových vektorů je třeba kouknout na své sousedy a z jejich polohy najít tečnou rovinu daného bodu. *Open3d* již obsahuje funkci pro odhadnutí normálových vektorů. Jako vstupní parametry přijímá počet sousedů a jejich maximální vzdálenost od aktuálního bodu. Před odhadem normálových vektorů a spuštěním ICP algoritmu se vždy zmenší objem dat vzorkováním. Ke vzorkování je využita voxelové vzorkování, které rozdělí prostor na stejně velké voxely o dané velikosti. Všechny body uvnitř voxelu se nahradí jedním bodem, jejich průměrem, a tím se dosáhne téměř rovnoměrného rozložení dat. Voxelové vzorkování je již také implementováno v knihovně *open3d*.

Parametry pro odhad normálových vektorů potřebných pro ICP jsou během všech iterací návrhu systému nepozměněné. Maximální vzdálenost sousedů je dvojnásobkem velikosti voxelů při vzorkování a maximální počet sousedů je vždy 20. Zastavovací podmínka pro minimální RMSE a shodnost Point Cloudů je vždy nastavena na 10^{-6} , což v realitě nenastane nikdy.



Obrázek 3.4: Voxelové vzorkování

Na začátku návrhu byla porovnána hustota bodů po vzorkování pro různou velikost voxelů. Bylo zjištěno, že pro velikost voxelu $a = 5$ mm dojde ke zmenšení objemu přibližně o 95 % a zároveň body stále zachycují hrubý tvar objektu. Pro $a = 3$ mm dojde ke zmenšení objemu zhruba o 90 %, pro $a = 2$ mm o 80 % a pro $a = 1$ mm o 30 %. Pro velikost voxelu 0,5 mm již nedojde ke změně objemu dat. Voxelové vzorkování je zobrazeno na obrázku 3.4.

První verze

Na začátku byl navržen systém bez jakéhokoliv předzpracování dat, který představoval naivní přístup registrace. Tento naivní systém data pouze vzorkoval voxely o velikosti 3 mm, aby ICP netrval příliš dlouho. Počet iterací ICP na jeden snímek byl 6, z toho pouze poslední iterace využila barvy s parametrem $\lambda = 0,98$. Registrace se prováděla vždy vůči celému dosud integrovanému modelu.

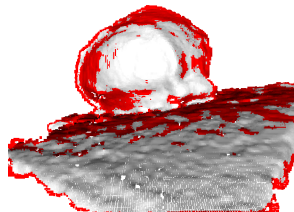
Po registraci a integraci všech dat je povrch modelu velmi hrbolatý a obsahuje nežádoucí artefakty v podobě černých bodů, které se na povrchu ve skutečnosti nevykytují. Rekonstruovaný model nereprezentuje skutečný objekt, protože během registrace došlo k driftu. Výsledek registrace je zobrazen v prvním řádku obrázku 3.6.



(a) Neupravený Point Cloud



(b) Point Cloud po voxelovém vzorkování ($a = 1$ mm)



(c) Body vybrané k odstranění ($r = 5$ mm, $k = 70$)



(d) Point Cloud po filtraci

Obrázek 3.5: Předzpracování Point Cloudu pro registraci

Druhá verze

Druhá verze systému se zaměřuje na zkrácení doby potřebné k registraci dat. Optimalizoval se zejména objem dat předávaný do ICP algoritmu, ale došlo i k vylepšení kvality rekonstrukce.

Nejpřirozenější způsob, jak zrychlit registraci dat, je zmenšit objem dat, pro které je registrace počítána. V druhé verzi systému se přistoupilo na tři klíčové aspekty pro zrychlení registrace. Za prvé se přeskočily snímky, které obsahovaly málo nové informace. Za druhé došlo k odstranění krajních bodů, které jsou náchylné na outliery. Za třetí se vytvořila dočasná paměť posledních několika registrovaných snímků,

vůči kterým se budou nové snímky registrovat. V ideální situaci by stačilo znát jenom jeden předchozí snímek k registraci nového, kvůli robustnosti je jich použito více. Registrace vůči všem dosud integrovaným bodům zpomaluje jak výpočet, tak i rychlost konvergence ke správnému řešení. Používání pouze posledních snímků navíc zachovává v průměru stejně dlouhou dobu registrace každého nového snímku. Proces předzpracování Point Cloudu je vidět na obrázku 3.5.

Pro přeskokování málo přínosných snímků byly otestovány dvě metody, výběr každého n -tého a shodnost Point Cloudů po jedné iteraci ICP algoritmu. Již z principu lze očekávat, že přeskokováním vždy stejného počtu snímků bude rychlejší, než pro každý snímek počítat shodnost s modelem. Na druhou stranu experimenty ukázaly, že testování shodnosti vede na kvalitnější rekonstrukci modelu, a proto je do systému zakomponovaná tato varianta. Odstranění bodů na okrajích bylo provedeno pomocí funkce pro odstranění outlierů podle kulového okolí. Tato funkce je implementována v knihovně *open3d* a odstraní všechny body, které mají ve vzdálenosti r méně než k sousedů.

Na začátku zpracování je každý Point Cloud vyhlazen pomocí voxelového vzorkování na voxely s délkou strany $a = 1$ mm. Tím je mírně potlačen šum měření, jehož propagace v registraci způsobuje driftování. Experimentálně bylo zjištěno, že registrace s barvenou složkou zhorší v tomto experimentu kvalitu rekonstruovaného modelu, a proto ji navržený systém nevyužívá.

Třetí verze

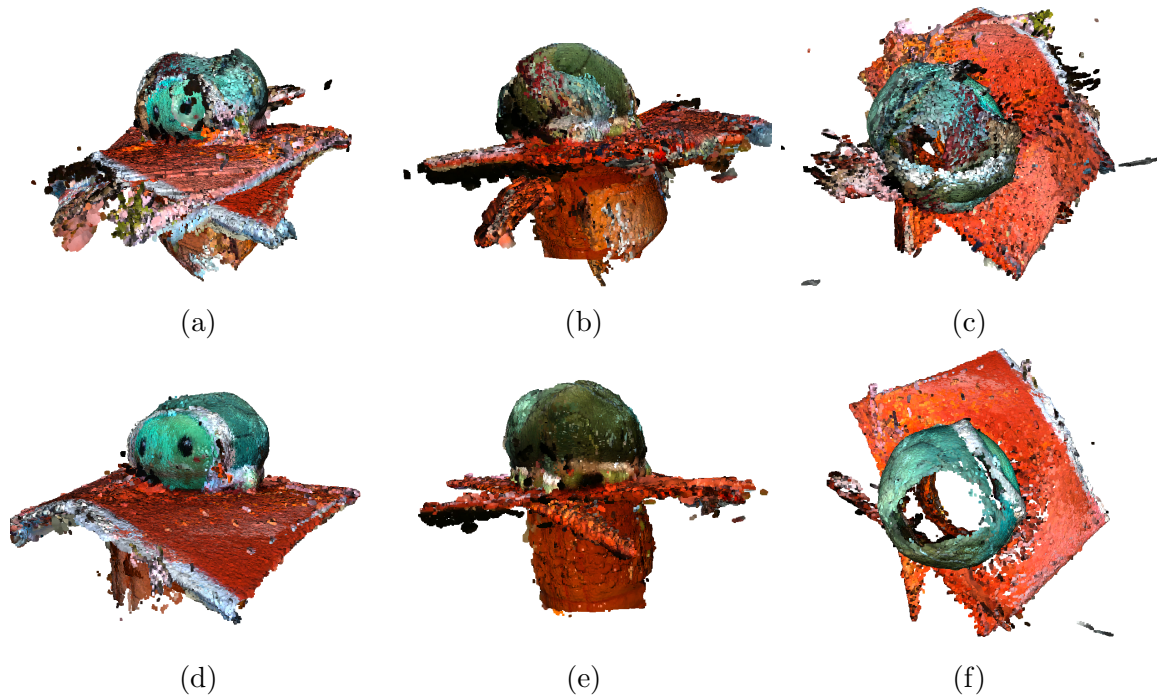
Třetí verze systému se zaměřuje na ladění parametrů a vyhlazení výsledného modelu. Celkem se ladilo 6 parametrů metodou pokus omyl. Vzhledem k nejasné závislosti mezi parametry byl v každém experimentu změněn pouze jeden parametr oproti minulému experimentu, aby byla podchycena kvalitativní změna v závislosti na daném parametru. Ovšem je možné, že se tímto postupem našlo pouze lokální optimum a nevyklučuje se, že úplně odlišná kombinace parametrů vede k vyšší kvalitě rekonstrukce. Kvalita rekonstrukce je posuzována pro každý experiment manuálně pozorováním klíčových aspektů modelu. Klíčové aspekty jsou hladkost povrchu, množství outlierů vůči zachovanému modelu, velikost driftu od reálného tvaru, zachování přibližných rozměrů objektu (smrštění nebo roztažení).

Finální parametry jsou následující: velikost paměťové buňky je 30 snímků (respektive Point Cloudů); Point Cloud je filtrován pomocí voxelového vzorkování s velikostí 1 mm; snímek je přeskočen pokud je po jedné iteraci ICP s velikostí voxelu 3 mm míra shodnosti větší než 0,96; body jsou označeny za outliery a odstraněny, pokud mají méně než 70 sousedů do vzdálenosti 5 mm; ICP se spouští v sekvenci (velikost voxelu - počet iterací): 5-1, 3-3, 2-1. Pro všechny ICP je parametr $\lambda = 1$ (barva bodů není využita).

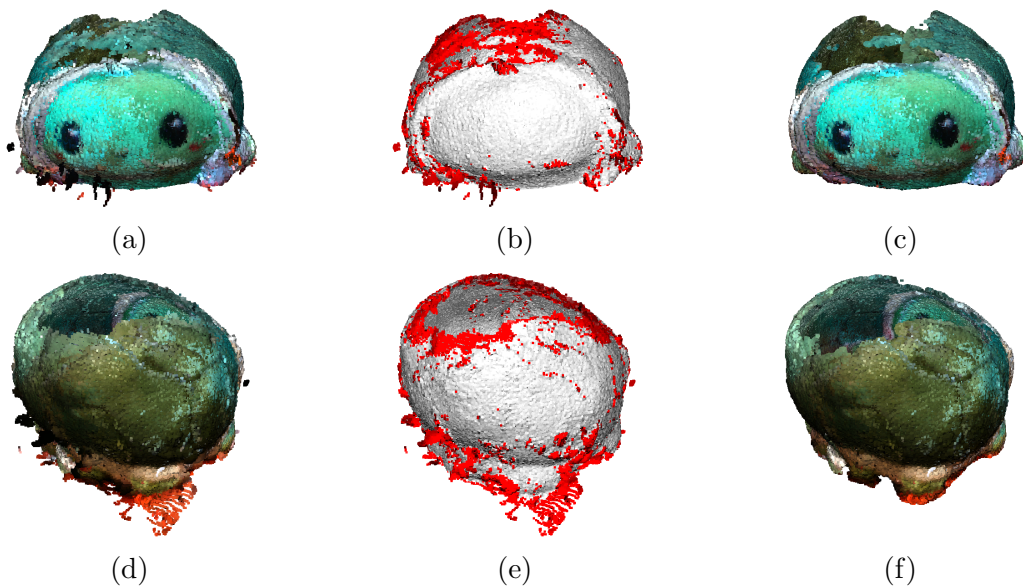
Registrace i s vizualizací každého kroku trvala 365 sekund a v porovnání s první verzí došlo k šestinásobnému zrychlení. Celkem bylo přeskočeno 136 snímků z původních 406. Po voxelovém vzorkování finálního modelu s velikostí voxelu 0,5 mm obsahuje model 2 516 073 bodů. Výsledek registrace je zobrazen v druhém řádku obrázku 3.6.

Z obrázku 3.6 je vidět, že se registrovaný model podobá reálnému objektu, ale nepovedlo se odstranit drift při registraci. Drift je obecně známým problémem při iterativní registraci a pro jeho odstranění byly vynalezeny sofistikované systémy, viz kapitola 2.4. Jejich návrh je komplikovaný a vyžaduje rozsáhlé znalosti v tomto oboru. Návrh takového systému je nad rámec této práce a bez aplikace složitějších filtrů již nelze dosáhnout lepšího výsledku.

Po registraci a integraci je třeba z modelu vyříznout pouze sledovaný objekt a následně odebrat zbylé outliery. Vyříznutí zde bylo provedeno ručně podle tvaru krychle, ale mohlo by být použita i sofistikovanější metoda, například vyříznutí podle rozdílné barvy podstavy a objektu. Po vyříznutí jsou outliery odebrány podle směrodatné odchylky vzdálenosti bodu od svých sousedů. Pokud průměrná vzdálenost bodu od svých k nejbližších sousedů je větší než p násobek směrodatné odchylky této hodnoty v celém Point Cloudu, pak bude bod odebrán. Pro tento případ se dosáhlo nejlepšího výsledku pro hodnoty $k = 50$ a $p = 2,5$. Po těchto operacích obsahuje model 934 738 bodů. Proces vyhlazení povrchu je zobrazen na obrázku 3.7.



Obrázek 3.6: Porovnání první a třetí verze systému



Obrázek 3.7: Vyhlazení povrchu modelu

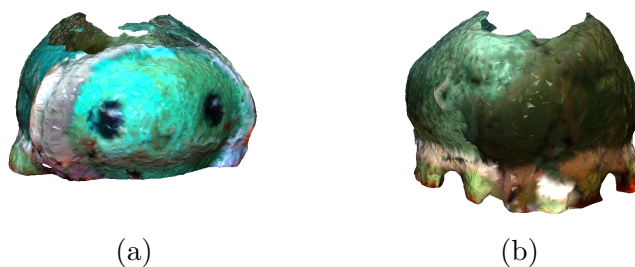
3.3.2 Převod Point Cloudu do tisknutelného formátu

Ačkoliv bodovému modelu chybí horní část, byly vyzkoušeny algoritmy pro převod Point Cloudu na Polygon Mesh. Konkrétně byly vyzkoušeny algoritmy Alpha Shapes, Ball Pivoting a Poissonův rekonstruktor, které jsou již implementované v knihovně *open3d*.

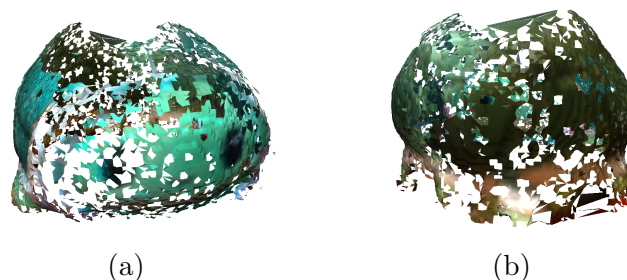
Algoritmy Alpha Shapes a Ball Pivoting se snaží vytvořit trojúhelníky mezi všemi body, a proto je vhodné opět zmenšit počet bodů před jejich spuštěním. Oproti tomu Poissonův rekonstruktor povrchu odhaduje implicitní funkci reprezentující daný model a je zde výhodné mít počet bodů větší. Poissonův rekonstruktor je výrazně rychlejší než předem zmíněné algoritmy při vytváření povrchu, ale pro jeho použití je třeba odhadnout a shodně orientovat normálové vektory všech bodů. Z tohoto důvodu je proces triangulace Poissonovým rekonstruktorem nejpomalejší ze zmíněných algoritmů.

Alpha Shapes vytvoří konvexní obal z čtyřstěnnů kolem modelu a poté odstraňuje všechny čtyřstěny, jejichž opsaná koule má poloměr větší než dané α . Pro tento model bylo zvolené voxelové vzorkování s velikostí 1 mm, což zmenšilo počet bodů na 202 393. Výsledek pro parametr $\alpha = 5$ mm je zobrazen na obrázku 3.8. Triangulace trvala 12 sekund. Povrch modelu je převážně hladký, ale hrany nejsou zachovány, protože dochází k vyhlazení ostrých úhlů. Tento algoritmus nedokáže vyplnit díry v modelu a zároveň zachovat detaily objektu.

Ball Pivoting vytváří trojúhelníky tam, kde jsou body dostatečně blízko u sebe, aby jimi nepropadla koule o poloměru ρ . Těchto poloměrů je zvoleno několik, aby se zaplnily díry z předchozí iterace. Bohužel se pomocí tohoto algoritmu implementovaného v *open3d* nedosáhlo uspokojujících výsledků. Pro různou kombinaci velikosti voxelů a sekvenci poloměrů se často stávalo, že program nedoběhl a nebo vytvořil povrch se spousty dírami. Na obrázku 3.9 je zobrazen výsledek pro velikost voxelu 2 mm a poloměry $\{3$ mm, 5 mm, 10 mm, 30 mm, 50 mm $\}$. Triangulace trvala 7 sekund, ale výsledek je nepoužitelný.



Obrázek 3.8: Rekonstruovaný povrch metodou Alpha Shapes



Obrázek 3.9: Rekonstruovaný povrch metodou Ball Pivoting

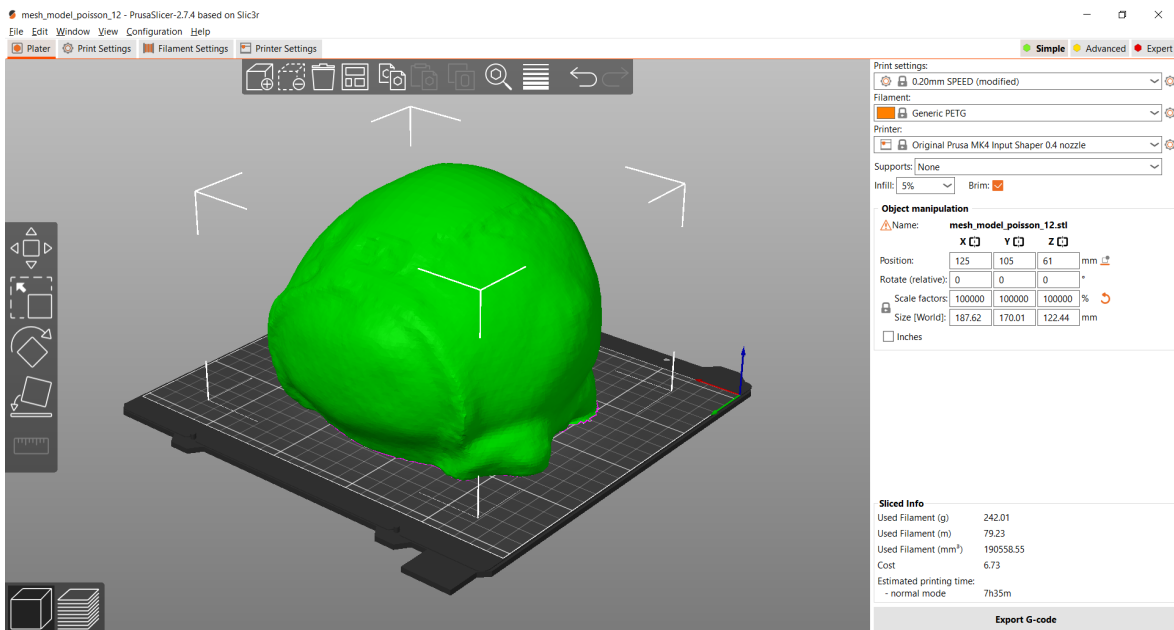


Obrázek 3.10: Rekonstruovaný povrch Poissonovým rekonstruktorem

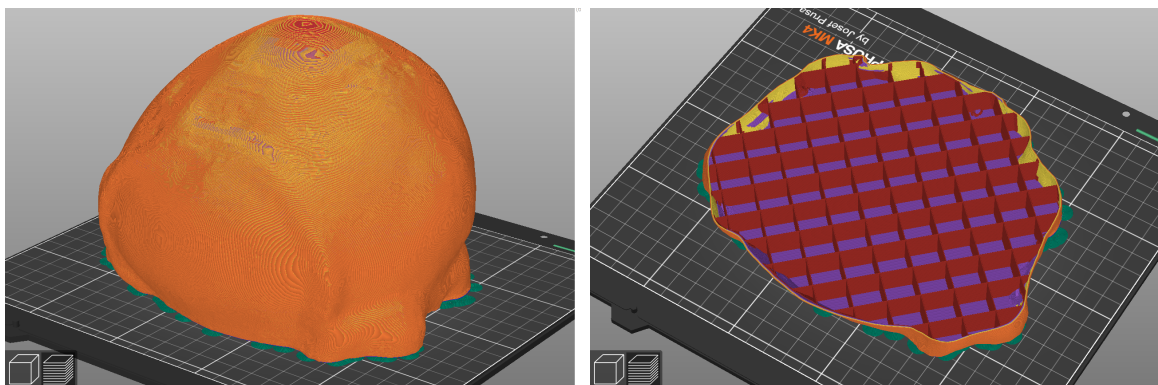
Poissonův rekonstruktor odhaduje implicitní funkci rozdělující model na vnitřek a vnějšek. Jelikož by odhad jedné funkce pro celý model byl příliš složitý, je model rozdělen na voxely pomocí oktalového stromu s hloubkou d . S hloubkou stromu se zvětšuje počet voxelů a tím i rozlišení modelu. Pro odhad normálových vektorů jejich shodnou orientaci jsou použity funkce v *open3d*. Pro odhad normálových vektorů byla zvolena maximální velikost okolí $r = 5$ mm a maximální počet použitých sousedů $k = 30$. Pro shodnou orientaci byl zvolen poměrně velký počet použitých sousedů $k = 100$, protože na orientaci normálových vektorů závisí kvalita Poissonovo rekonstruktoru. Na obrázku 3.10 je zobrazen výsledek pro hloubku $d = 7$. Proces triangulace trval pouhé 2 sekundy pro všechny body bez vzorkování, ale odhad normálových vektorů a následně jejich orientace trvala 164 sekund. Výsledný model obsahuje 67 393 trojúhelníků a povrch je relativně detailní a bez nežádoucích artefaktů. Výhodné je, že Poissonův rekonstruktor odhaduje tvar povrchu v místech, kde jsou díry, a to poměrně věrohodně. Jelikož model nemá žádné body ze spodní strany, vznikl dole takový plášť, který byl odříznut podle minimální polohy bodů na ose z bodového modelu.

3.4 3D tisk

Model vygenerovaný pomocí Poissonovo rekonstruktoru byl uložen ve formátu STL a nahrán do sliceru *PrusaSlicer*. Po nahrání do sliceru je třeba zvětšit tisíckrát měřítko, protože knihovna *open3d* používá jako jednotku vzdálenosti metr, kdežto *PrusaSlicer* má jednotky v milimetrech. Jak je vidět na obrázku 3.11, rekonstruovaný model má rozměry (D×Š×V): 18,8×17,0×12,2 cm, což znamená, že došlo přibližně k 25% zvětšení oproti realitě. Takovýto závěr byl očekáván, neboť při rekonstrukci došlo k driftu a konec registrace nebyl spojen se začátkem, viz obrázek 3.10b. Na obrázku 3.12 jsou zobrazené vrstvy filamentu s výškou 0,2 mm tak, jak by je tiskla 3D tiskárna. Také je na obrázku zobrazen průřez modelem, který má 5% rektilineární výplň. S touto konfigurací parametrů by bylo spotřebováno přibližně 0,25 kg filamentu. Aby se ušetřilo na spotřebovaném filamentu, byl model vytištěn v poloviční velikosti. Model želvy je dostupný v repositáři, jež je součástí této práce, a lze s ním volně experimentovat.



Obrázek 3.11: Modelu zobrazený v *PrusaSliceru*



Obrázek 3.12: Vygenerované vrstvy pro 3D tiskárnu

3.5 Uživatelské rozhraní

V rámci návrhu systému pro tvorbu počítačového modelu 3D skenováním bylo vytvořeno uživatelské rozhraní obsahující veškeré algoritmy pro nahrání a zpracování dat. Navržený systém pro 3D rekonstrukci je tak možné používat pomocí grafických komponent aplikace namísto spuštění programů v konzoli. Aplikace dále slouží k vizualizaci dat během nahrávání a následné tvorbě modelu a poskytuje prvky pro nastavování parametrů algoritmů.

Po spuštění aplikace se vykreslí grafické komponenty a naplní se výchozími hodnotami. Poté se spustí projektový manažer, který má na starost vytváření a otevírání projektů, což jsou hierarchicky uspořádané adresáře v paměti počítače, a manipulaci s daty, konkrétně ukládání dat do souborů a jejich čtení v rámci projektu. Nakonec se systém pro ovládání kamery dotáže počítače, jestli není připojena kamera od Intel RealSense, pokud ano a jedná se o model D435i, tak je automaticky zvolena pro nahrávání

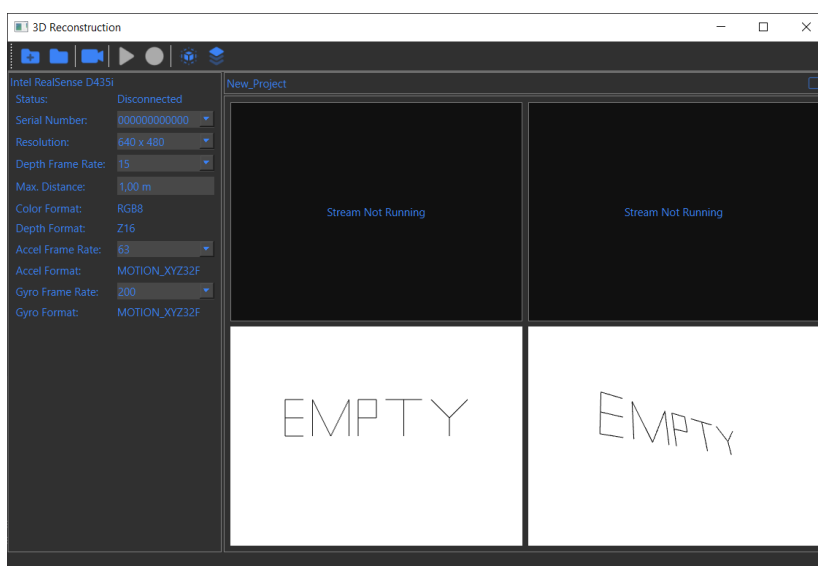
dat. Systém dále detekuje nově připojené a odpojené zařízení a automaticky na tyto situace reaguje. Například po odpojení kamery se automaticky ukončí nahrávání.

Na obrázku 3.13 je vidět výchozí stav aplikace. V horním okraji se nachází panel nástrojů s hlavními funkcemi, jimiž jsou vytváření a otevírání projektů, zahájení přenosu dat z kamery, nahrávání dat a spouštění jednotlivých algoritmů pro tvorbu modelu. V levé části se nachází panel s nastavitelnými parametry a základními informacemi. Zbytek aplikace tvoří prostor s vizualizačními okénky, ty lze přepínat mezi vizualizací dat při nahrávání a vizualizací modelu při rekonstrukci.

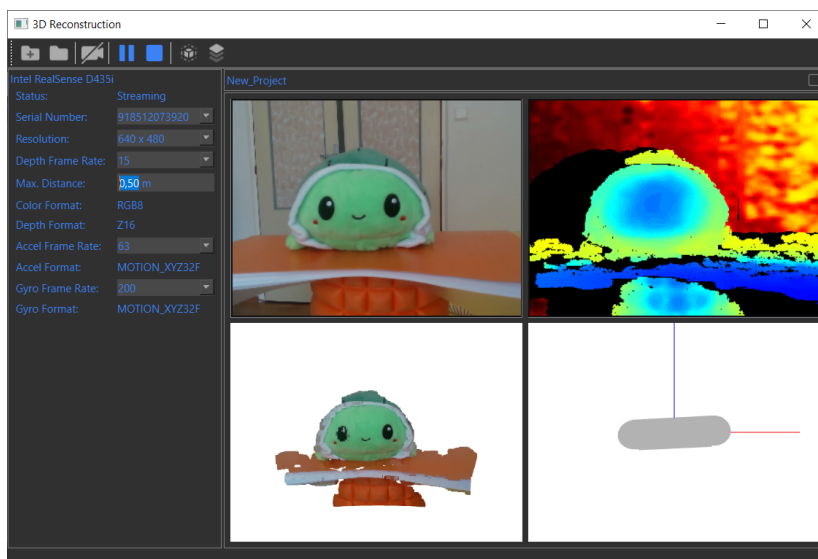
Při vizualizaci dat z kamery jsou k dispozici čtyři okénka, viz obrázek 3.14. Levé horní zobrazuje RGB snímky z kamery, pravé horní obarvenou hloubkovou mapu, levé dolní vygenerovaný Point Cloud z živého přenosu a pravé dolní odhad natočení kamery. Poslední okénko je pozůstatkem pokusu o využití IMU jednotky uvnitř kamery. Nakonec k využití těchto dat nedošlo, a proto ani vizualizace natočení kamery není upravena, aby odpovídala realitě.

Kliknutím na tlačítko vpravo nad vizualizačními okénky se přepne aplikace do stavu vizualizace procesů tvorby modelu. Jak je vidět na obrázku 3.15, vpravo se tentokrát nachází pouze jedno velké vizualizační okénko a parametry pro nahrávání se vyměnily za parametry pro algoritmy rekonstrukce. Během procesu registrace a integrace dat je aktuální stav modelu průběžně aktualizován. Tento proces ovšem zabere spoustu času. Pokud bude chtít uživatel proces přerušit, stačí aby podržel klávesu *Esc*. Během registrace je také možné modelem volně otáčet pomocí myši, ale vzhledem k tomu, že téměř celá aplikace běží na jednom vlákně, může chvíli trvat, než aplikace zareaguje.

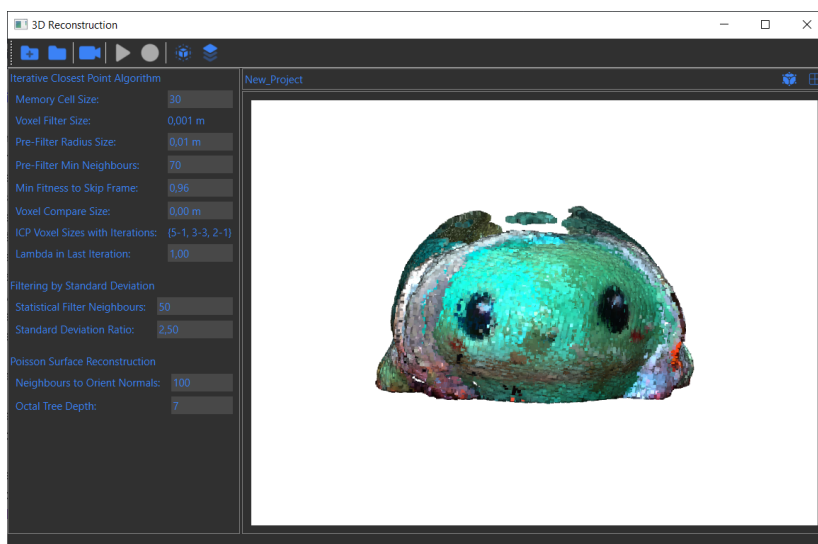
Na závěr je třeba zmínit některé kvalitativní vlastnosti navrženého uživatelského rozhraní. Při jeho návrhu bylo bráno v potaz především to, aby uživatel nemohl svou činností rozbít chod aplikace. Aktuální verze poskytuje pouze bezpečně funkční minimum pro práci s navrženým systémem, ale chybí ji některé uživatelsky přívětivé vlastnosti. Aplikace například neumožňuje zpětné přehrávání naskenovaných dat, které by uživatel mohl využít pro analýzu dat před tvorbou modelu. Aplikace také nedokáže zpětně vyvolat trojúhelníkový model, protože se při jeho nahrávání ze souboru vyskytne chyba, kterou se nepodařilo odstranit. Ovšem model se zdá být v pořádku, protože ho lze bez problémů otevřít v jiném CAM softwaru (PrusaSlicer).



Obrázek 3.13: Výchozí stav aplikace



Obrázek 3.14: Stav aplikace během nahrávání



Obrázek 3.15: Stav aplikace během tvorby modelu

Kapitola 4

Závěr

Cílem této práce bylo navrhnout vlastní systému pro tvorbu počítačového modelu vhodného pro následný 3D tisk. Za tímto účelem byly popsány principy měření pomocí 3D skenerů, algoritmy pro zpracování dat a algoritmy pro tvorbou tisknutelného modelu. Byl podrobněji představen Iterative Closest Point algoritmus pro registraci mračen bodů, který byl následně použit v navrženém systému.

Pro skenování objektu byla použita hloubková kamera Intel RealSense D435i, se kterou se ručně obcházel kolem objektu. Převod RGB-D snímků na mračno bodů a zbylé zpracování dat je provedeno na notebooku. Navržený systém uplatňuje jednoduché metody pro filtraci šumu a odstranění outlierů, které zlepšují registraci a integraci dat. Z rekonstruovaného bodového modelu je ručně vyříznuta část se skenovaným objektem a následně je model systémem převeden na síť trojúhelníků vhodnou pro 3D tisk. Pro snazší obsluhu systému bylo vytvořeno uživatelské rozhraní.

Navržený systém není schopen zcela přesně rekonstruovat skenovaný model, protože neobsahuje žádnou formu uzavření skenovací smyčky a model při registraci dat driftuje od reality. Avšak rekonstruovaný model se podobá reálnému objektu a došlo na jeho 3D tisk v poloviční velikosti. Ke zlepšení kvality rekonstruovaného modelu by bylo třeba aplikovat složitější algoritmy pro filtraci šumu a outlierů, aby bylo potlačeno driftování. Dalšího způsobu zlepšení lze dosáhnout použitím mechanismus pro autokorekci při nalezení uzavřené smyčky. Také se nabízí možnost použití jiné reprezentace dat a registračních algoritmů, které by mohly vést na lepší výsledek.

Seznam obrázků

2.1	Příklady průmyslových 3D skenerů	4
2.2	Příklady cenově dostupných 3D skenerů	4
2.3	Princip laserové triangulace	6
2.4	Princip stereovize [15]	7
2.5	Příklad značek pro 3D skenování	8
2.6	Rozdíl mezi hloubkou a vzdáleností od senzoru	10
2.7	Příklad mapy významných bodů [20]	10
2.8	Příklad aproximační B-Spline křivky a plochy [21]	12
2.9	TSDF mřížka ve 2D [23]	12
2.10	Postup metody Structure from Motion [2]	20
2.11	Příklady vzorů pro výplň 3D modelu v PrusaSliceru	22
3.1	Vnitřní topologie navrženého systému pro tvorbu počítačového modelu reálného objektu	24
3.2	Vytvořený Point Cloud z naměřených dat	26
3.3	Průběh ICP algoritmu	27
3.4	Voxelové vzorkování	28
3.5	Předzpracování Point Cloudu pro registraci	29
3.6	Porovnání první a třetí verze systému	31
3.7	Vyhlazení povrchu modelu	31
3.8	Rekonstruovaný povrch metodou Alpha Shapes	32
3.9	Rekonstruovaný povrch metodou Ball Pivoting	32
3.10	Rekonstruovaný povrch Poissonovým rekonstruktorem	33
3.11	Modelu zobrazený v <i>PrusaSliceru</i>	34
3.12	Vygenerované vrstvy pro 3D tiskárnu	34
3.13	Výchozí stav aplikace	35
3.14	Stav aplikace během nahrávání	36
3.15	Stav aplikace během tvorby modelu	36

Literatura

- [1] R. H. Helle and H. G. Lemu, “A case study on use of 3d scanning for reverse engineering and quality control,” *Materials Today: Proceedings*, vol. 45, pp. 5255–5262, 2021. Second International Conference on Aspects of Materials Science and Engineering (ICAMSE 2021).
- [2] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016.
- [3] Z. Bi and L. Wang, “Advances in 3d data acquisition and processing for industrial applications,” *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 5, pp. 403–413, 2010.
- [4] M. Daneshmand, A. Helmi, E. Avots, F. Noroozi, F. Alisinanoglu, H. S. Arslan, J. Gorbova, R. E. Haamer, C. Ozcinar, and G. Anbarjafari, “3d scanning: A comprehensive survey,” *arXiv preprint arXiv:1801.08863*, 2018.
- [5] Z. Geng and B. Bidanda, “Review of reverse engineering systems – current state of the art,” *Virtual and Physical Prototyping*, vol. 12, no. 2, pp. 161–172, 2017.
- [6] T. Várady, R. R. Martin, and J. Cox, “Reverse engineering of geometric models—an introduction,” *Computer-Aided Design*, vol. 29, no. 4, pp. 255–268, 1997. Reverse Engineering of Geometric Models.
- [7] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménéier, “An overview of depth cameras and range scanners based on time-of-flight technologies,” *Machine vision and applications*, vol. 27, no. 7, pp. 1005–1020, 2016.
- [8] A. Olawuyi, E. Okeniyi, and I. Osisanya, “Accuracy assessment of height difference using total station and levelling instrument,” *World Scientific News*, vol. 189, pp. 87–101, 01 2024.
- [9] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, “Kinect v2 for mobile robot navigation: Evaluation and modeling,” in *2015 international conference on advanced robotics (ICAR)*, pp. 388–394, IEEE, 2015.
- [10] M. S. Ahn, H. Chae, D. Noh, H. Nam, and D. Hong, “Analysis and noise modeling of the intel realsense d435 for mobile robots,” in *2019 16th International Conference on Ubiquitous Robots (UR)*, pp. 707–711, IEEE, 2019.
- [11] H. Yu, “Scanning acoustic microscopy for material evaluation,” *Applied Microscopy*, vol. 50, no. 1, p. 25, 2020.

- [12] M. A.-B. Ebrahim, “3d laser scanners’ techniques overview,” *Int J Sci Res*, vol. 4, no. 10, pp. 323–331, 2015.
- [13] M. Koch and M. Kaehler, “Combining 3d laser-scanning and close-range photogrammetry—an approach to exploit the strength of both methods,” in *Making History Interactive. Computer Applications and Quantitative Methods in Archeology Conference*, pp. 22–26, Willimsburg VA, March, 2009.
- [14] J. Franca, M. Gazziro, A. Ide, and J. Saito, “A 3d scanning system based on laser triangulation and variable field of view,” in *IEEE International Conference on Image Processing 2005*, vol. 1, pp. I–425, 2005.
- [15] K. Sugihara, “Three principles in stereo vision,” *Advanced Robotics*, vol. 1, no. 4, pp. 391–400, 1986.
- [16] S. Barone, A. Paoli, and A. V. Razonale, “Three-dimensional point cloud alignment detecting fiducial markers by structured light stereo imaging,” *Machine Vision and Applications*, vol. 23, pp. 217–229, 2012.
- [17] M. Maidi, J.-Y. Didier, F. Ababsa, and M. Mallem, “A performance study for camera pose estimation using visual marker based tracking,” *Machine Vision and Applications*, vol. 21, no. 3, pp. 365–376, 2010.
- [18] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [19] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136, 2011.
- [20] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular slam with map reuse,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [21] C. Harmening, *Spatio-temporal deformation analysis using enhanced B-spline models of laser scanning point clouds*. PhD thesis, Technische Universität Wien, 05 2020.
- [22] E. Dimas and D. Briassoulis, “3d geometric modelling based on nurbs: a review,” *Advances in Engineering Software*, vol. 30, no. 9, pp. 741–751, 1999.
- [23] K. Daun, S. Kohlbrecher, J. Sturm, and O. von Stryk, “Large scale 2d laser slam using truncated signed distance functions,” in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 222–228, 2019.
- [24] W. Lyu, W. Ke, H. Sheng, X. Ma, and H. Zhang, “Dynamic downsampling algorithm for 3d point cloud map based on voxel filtering,” *Applied Sciences*, vol. 14, no. 8, 2024.

- [25] D. Bazazian, J. Casas, and J. Ruiz-Hidalgo, "Fast and robust edge extraction in unorganized point clouds," in *2015 international conference on digital image computing: techniques and applications (DICTA)*, pp. 1–8, 11 2015.
- [26] A. Grunnet-Jepsen and D. Tong, "Depth post-processing for intel® realsense™ d400 depth cameras," *New Technologies Group, Intel Corporation*, vol. 3, 2018.
- [27] J. M. Martínez-Otzeta, I. Rodríguez-Moreno, I. Mendiáldua, and B. Sierra, "Ransac for robotic applications: A survey," *Sensors*, vol. 23, no. 1, p. 327, 2022.
- [28] Q. Zhao, X. Gao, J. Li, and L. Luo, "Optimization algorithm for point cloud quality enhancement based on statistical filtering," *Journal of Sensors*, vol. 2021, pp. 1–10, 2021.
- [29] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," *arXiv preprint arXiv:2103.02690*, 2021.
- [30] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [31] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, 2001.
- [32] R. Sitnik and M. Karaszewski, "Optimized point cloud triangulation for 3d scanning systems," *Machine Graphics and Vision*, vol. 17, no. 4, pp. 349–371, 2008.
- [33] C. Lv, W. Lin, and B. Zhao, "Voxel structure-based mesh reconstruction from a 3d point cloud," *IEEE Transactions on Multimedia*, vol. 24, pp. 1815–1829, 2022.
- [34] A. Ramya and S. L. Vanapalli, "3d printing technologies in various applications," *International Journal of Mechanical Engineering and Technology*, vol. 7, no. 3, pp. 396–409, 2016.
- [35] C. Iancu, "About 3d printing file formats.," *Annals of 'Constantin Brancusi' University of Targu-Jiu. Engineering Series/Analele Universității Constantin Brâncuși din Târgu-Jiu. Seria Inginerie*, 2018.
- [36] M. Qamar Tanveer, G. Mishra, S. Mishra, and R. Sharma, "Effect of infill pattern and infill density on mechanical behaviour of fdm 3d printed parts- a current review," *Materials Today: Proceedings*, vol. 62, pp. 100–108, 2022. International Conference on Advances in Materials and Mechanical Engineering.
- [37] V. Tadic, A. Toth, Z. Vizvari, M. Klincsik, Z. Sari, P. Sarcevic, J. Sarosi, and I. Biro, "Perspectives of realsense and zed depth sensors for robotic vision applications," *Machines*, vol. 10, no. 3, 2022.