



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

REALIZACE 3D SKENERU NA BÁZI STRUKTUROVANÉHO SVĚTLA

REALIZATION OF A STRUCTURED LIGHT 3D SCANNER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Matouš Přikryl

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Roman Adámek

BRNO 2021

Zadání bakalářské práce

| | |
|-------------------|--|
| Ústav: | Ústav mechaniky těles, mechatroniky a biomechaniky |
| Student: | Matouš Přikryl |
| Studijní program: | Aplikované vědy v inženýrství |
| Studijní obor: | Mechatronika |
| Vedoucí práce: | Ing. Roman Adámek |
| Akademický rok: | 2020/21 |

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Realizace 3D skeneru na bázi strukturovaného světla

Stručná charakteristika problematiky úkolu:

3D skenování je v současné době často používanou metodou pro digitalizaci objektů i rozměrných celků. Cílem této práce bude navrhnout a sestavit 3D skener využívající ke své funkci strukturované světlo. Tento 3D skener bude doplněn otočným stolem, na kterém bude skenovaný objekt umístěn, a bude schopen vytvořit 3D model daného objektu.

Cíle bakalářské práce:

- Provedte rešerši na téma 3D skenerů využívajících strukturované světlo. V rámci rešerše popište, mimo jiné, různé typy používaného strukturovaného světla, metody zpracování snímků a generování 3D modelu z naměřených dat.
- Na základě dostupného vybavení v Mechatronické laboratoři, sestavte a zprovozněte HW skeneru využívající strukturované světlo spolu s otočným stolem pro skenovaný objekt.
- Vytvořte vlastní nebo použijte již existující SW pro 3D skener, který bude schopen generovat strukturované světlo, snímat obraz z kamery, vyhodnocovat jej a vytvářet 3D model snímaného objektu.
- Porovnejte funkčnost 3D skeneru při využití různých typů strukturovaného světla, okolního osvětlení a různých povrchů a členitosti snímaného objektu.

Seznam doporučené literatury:

TRUCCO, Emanuele a Alessandro VERRI. Introductory techniques for 3-D computer vision. Upper Saddle River, NJ: Prentice Hall, c1998. ISBN 0132611082.

SOLEM, Jan Erik. Programming computer vision with Python. Sebastopol, CA: O'Reilly, 2012. ISBN 1449316549.

KAEHLER, Adrian a Gary R. BRADSKI. Learning OpenCV 3: computer vision in C++ with the OpenCV library. Sebastopol, CA: O'Reilly Media, [2017]. ISBN 1491937998.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

V úvodu se práce zabývá různými druhy 3D skenerů. Následně se zaměřuje na funkci 3D skeneru využívajícího ke své funkci strukturované světlo. Zabývá se celým procesem skenování od kalibrace, přes optickou triangulaci, až po generování struktury skenovaného objektu. V druhé části práce popisuje realizaci jednoduchého 3D skeneru s využitím vybavení Mechatronické laboratoře a jeho schopnosti v různých okolních podmínkách.

Summary

The beginning of the thesis presents 3D scanner types. Following part describes how the structured light 3D scanners works. It takes into the account whole scanning process and address problems like calibration, optical triangulation, or surface reconstruction of 3D objects. The second half of the thesis is focused on the implementation of simple 3D scanner, using Mechatronics laboratory hardware. It also describes scanners behaviour in different conditions.

Klíčová slova

3D skener, strukturované světlo, projektor, kamera, 3D model, shluk bodů

Keywords

3D scanner, structured light, projector, camera, 3D model, pointcloud

Bibliografická Citace

PŘIKRYL, M. *Realizace 3D skeneru na bázi strukturovaného světla*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2021. 43 s., Vedoucí diplomové práce: Ing. Roman Adámek.

Prohlašuji, že jsem bakalářskou práci *Realizace 3D skeneru na bázi strukturovaného světla* vypracoval samostatně pod vedením Ing. Romana Adámka, s použitím materiálů uvedených v seznamu literatury.

Matouš Příkryl

Brno

.

Děkuji tímto Ing. Romanovi Adámkovi za vedení práce a cenné rady k jejímu vypracování. Také děkuji všem, kteří svými radami, připomínkami a motivací přispěli k dokončení práce.

Matouš Přikryl

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 9 |
| 2 | Cíl práce | 10 |
| 3 | Rozdělení 3D skenerů | 11 |
| 4 | 3D skenery a strukturované světlo | 14 |
| 4.1 | Camera obscura | 14 |
| 4.2 | Kalibrace | 15 |
| 4.3 | Optická triangulace | 16 |
| 4.4 | Filtrace bodů | 20 |
| 4.5 | Vytváření 3D modelu | 21 |
| 4.5.1 | Využití základních tvarů | 21 |
| 4.5.2 | Ball pivoting | 22 |
| 4.5.3 | Poisson surface reconstruction | 22 |
| 5 | Realizace | 23 |
| 5.1 | Hardware | 23 |
| 5.2 | Software | 24 |
| 5.2.1 | Ovládání otočného stolu | 25 |
| 5.2.2 | Scanner.py | 25 |
| 6 | Zhodnocení výsledků práce | 28 |
| 6.1 | Geometrická přesnost | 28 |
| 6.2 | Vliv strukturovaného světla | 29 |
| 6.3 | Vliv okolního osvětlení | 32 |
| 6.4 | Vliv materiálu objektů | 34 |
| 7 | Závěr | 37 |
| | Literatura | 38 |
| | Seznam využitých knihoven | 40 |
| | Seznam zkratk | 41 |

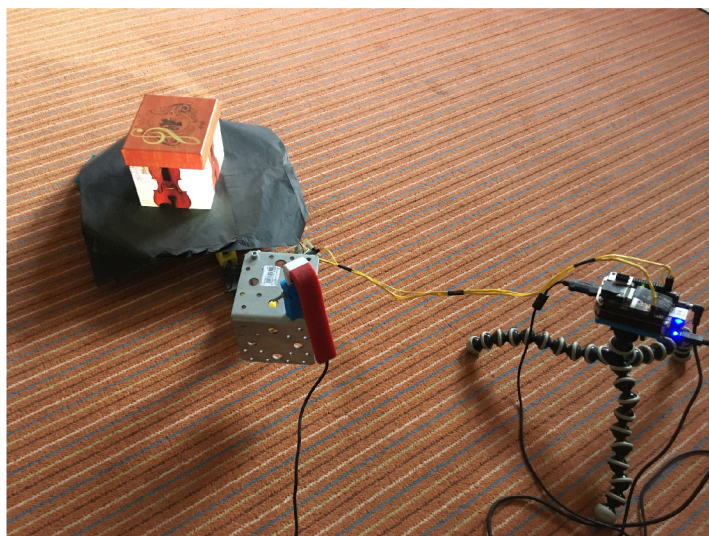
| | |
|----------------|----|
| Seznam obrázků | 42 |
| Seznam tabulek | 43 |

1 Úvod

3D skenování je proces získávání digitálního 3D modelu již existující věci. Existuje mnoho druhů 3D skenerů, které se často liší významně nejen konstrukcí, ale i v oblasti použití. Pro využití v průmyslu je potřeba mnohonásobně vyšší přesnosti, než při domácím skenování jednoduchých předmětů, jejichž tvar chceme zdigitalizovat. Od toho se následně odvíjí kvalita provedení, ale také princip, který skener k získání modelu využívá. Různé technologie nabízí také různé rychlosti skenování.

Využití v současné době rozšiřuje především rozmach 3D tisku. Běžní uživatelé jsou tak schopni jednoduše tisknout věci, které již vlastní a potřebují tvarově stejný objekt. Není tedy problém tisknout ani předměty, které jsou rozlomeny na několik částí, ale lze je spojit tak, aby mohl být získán jejich model. Velký rozvoj lze ovšem sledovat také v průmyslu a jiných odvětvích, například medicíně, nebo archeologii. Vzhledem ke zvětšujícímu se výpočetnímu výkonu běžných procesorů se začínají nejrůznější technologie 3D skenování objevovat i v běžné spotřebitelské elektronice a mohou tedy být součástí, jak mobilních telefonů, tak i např. automobilů.

Tato práce se zabývá především realizací jednoduchého 3D skeneru využívajícího strukturovaného světla na základě vybavení Mechatronické laboratoře. Finální verzi lze vidět na obr. 1.1. Pro jeho konstrukci využíváme jedné kamery, projektoru a otočného stolu. V následujících kapitolách bude vysvětlena funkce skeneru a průběh skenování, který obsahuje mnoho dílčích problémů, jako např. kalibraci soustavy, nebo optickou triangulaci využívanou pro získávání bodů. Navrhnut bude také software potřebný ke zpracování obrazových dat. Nakonec také popíšeme geometrickou přesnost a vlivy okolního osvětlení na kvalitu skenování.



Obrázek 1.1: Realizovaný 3D skener

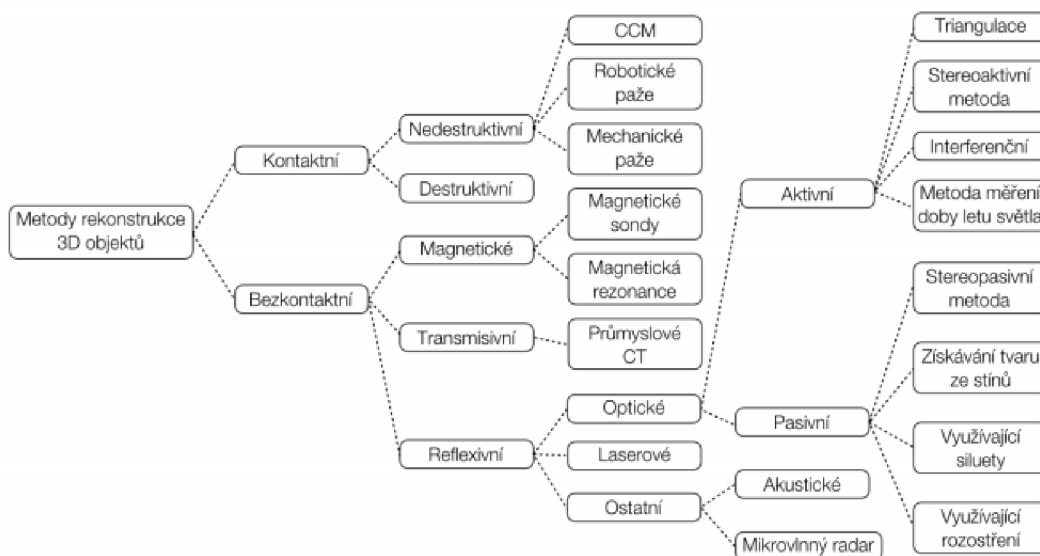
2 Cíl práce

Hlavním cílem práce je realizovat 3D skener využívající strukturované světlo, schopný vytvořit 3D model objektu a následně porovnat funkčnost skeneru za různých podmínek. Cíl práce můžeme rozdělit na jednotlivé podcíle:

- Proveďte rešerši na téma 3D skenerů využívajících strukturované světlo. V rámci rešerše popište, mimo jiné, různé typy používaného strukturovaného světla, metody zpracování snímků a generování 3D modelu z naměřených dat.
- Na základě dostupného vybavení v Mechatronické laboratoři, sestavte a zprovozněte HW skeneru využívající strukturované světlo spolu s otočným stolem pro skenovaný objekt.
- Vytvořte vlastní nebo použijte již existující SW pro 3D skener, který bude schopen generovat strukturované světlo, snímat obraz z kamery, vyhodnocovat jej a vytvářet 3D model snímaného objektu.
- Porovnejte funkčnost 3D skeneru při využití různých typů strukturovaného světla, okolního osvětlení a různých povrchů a členitosti snímaného objektu.

3 Rozdělení 3D skenerů

Existuje nepřehledné množství různých 3D skenerů. V této kapitole se zaměříme na nejpožívanější druhy, které se využívají hojně v mnoha částech naší společnosti. Na obrázku 3.1 také vidíme jedno z mnoha možných rozdělení.



Obrázek 3.1: Dělení 3D skenerů [4]

- **Kontaktní 3D skenery**

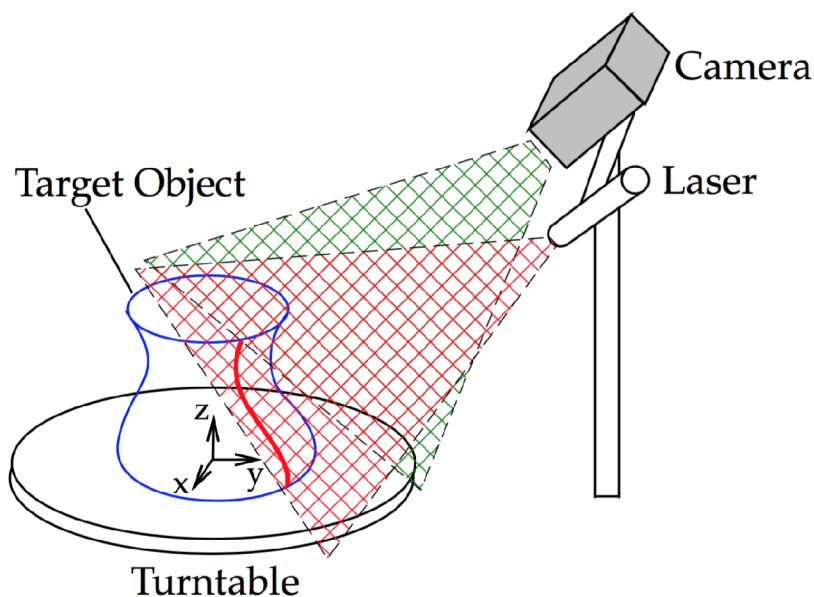
3D skenery založené na dotyku [4] se skenovaným objektem mají pohyblivou část, která snímá svou polohu, a hrot, případně jinou část citlivou na tlak, která sleduje kontakt se skenovaným předmětem. V praxi můžeme nelézt mnoho variant, od ručně ovládaných, po plně automatické. V průmyslu také narazíme často na jednoduché aplikace, které mají za úkol měřit odchylky vyrobených dílů na nějaké definované přímce nebo ploše. Nevytváří tedy celkový model, ale měří například rozměry profilu v jednom směru. Velkou výhodou dotykových 3D skenerů je schopnost skenovat skleněné a lesklé předměty bez problémů, které se objevují u optických skenerů. Nevýhodou může být například možnost rekonstrukce pouze malých předmětů, nebo také relativně pomalé získávání modelu, které může trvat i desítky minut.

- **Fotogrammetrie**

Fotogrammetrie funguje na principu zpracování fotek z mnoha pohledů. Jediné co je potřeba, je dostatek kvalitních obrázků zachycující objekt, který chceme převést na 3D model, a program, který dokáže obrazová data zpracovat. Fotky musí být pořízeny tak, aby byl objekt viditelný ze všech stran a každá část byla viditelná minimálně na dvou z nich. Software následně zpracuje soubory, a na základě významných bodů objektu, které se snaží najít na každé fotce, skládá jednotlivé pohledy k sobě. Následně je schopen převést obrazová data na výsledný 3D model. Proces zpracování zachycených dat je výpočetně velmi náročný. Je proto vhodné provádět je na počítačích s velkým grafickým výkonem. Software je dostupný ke stažení na internetu. Většina z pokročilejších programů je placená, existují ale i bezplatné programy, jako například Colmap.

- **Laserové 3D skenery**

Laserové skenery jsou dvojího typu. První typ využívá laserový proužek [1], který promítá na skenovaný objekt. Ten je snímán kamerou u které známe její pozici v prostoru a následně pomocí triangulace počítáme umístění daných bodů proužku v prostoru. Princip funkce můžeme vidět na obrázku 3.2. Druhý typ využívá k získávání informací o vzdálenosti rychlost světla. Zařízení vyšle daným směrem bodový paprsek a snímá čas, za jaký se odraz laserového paprsku vrátí zpět. Snímač tedy získá vzdálenosti objektu na všech místech, na které vyšle světlo a je schopen poskládat 3D model objektu. Tato varianta se využívá především pro měření velkých objektů a velkých vzdáleností. Rychlost světla je velmi vysoká¹ a problémem je zachytit přesný čas od vyslání paprsku do vrácení jeho obrazu. I při rychlostech, kterých jsou dnešní snímače schopny dosáhnout, nedokážeme dosáhnout přesnosti, která je možná s využitím ostatních typů skenerů.



Obrázek 3.2: Laserový 3D skener využívající triangulaci [1]

¹ve vzduchu dosahuje světlo rychlosti téměř 300 000 km/s

- **Skenery využívající strukturované světlo**

Skenování s využitím strukturovaného světla [1] je varianta, kterou se budeme zabývat v této práci. Strukturované světlo je obrazec, který má strukturu uspořádanou tak, abychom byli schopni rozlišit jednotlivé jeho části. Jedná se tedy často o pruhy o různých šířkách, nebo barvách. Mohou se ale objevit i méně typické druhy, které mohou využívat unikátních rozdělení barev ve složitých barevných obrazcích. Vidíme-li klidně jen část obrazu, stále jsme schopni určit, ve které části jsme. Skener potřebuje zdroj strukturovaného světla, jímž bývá projektor. Dále potřebuje kameru, která snímá skenovaný nasvětlovaný objekt. To nám dává informaci o tom, na kterou část obrazu promítaného z projektoru je kamera nasměřována. Když víme, jaká část obrazu promítaného projektorem vidíme na fotce z kamery, jsme schopni využít triangulace (kapitola 4.3) k získání bodů v prostoru. Výhodou je dobrá dostupnost potřebného hardware, kde můžeme využít i běžně dostupných projektorů a kamer. Stejně tak jako u ostatních optických metod skenování má velký vliv na výsledný sken okolní osvětlení a povrch skenovaného předmětu.

- **Skenery využívající rentgen**

Výpočetní tomografie, používaná převážně v medicíně je také druh 3D skeneru. Skenovaný objekt se nachází mezi zářičem rentgenového záření a detektorem rentgenového záření. Zářič s detektorem postupně krouží kolem skenovaného objektu a po každém posunutí o určitý úhel provedou snímek. Z každé celé otáčky kolem skenovaného objektu získá počítač pomocí specializovaného software sken jedné vrstvy objektu. Následně můžeme jednotlivé vrstvy poskládat k sobě a získáme výsledný tvar. Můžeme tak, s využitím rentgenu, neinvazně vytvářet modely orgánů živých organismů.

4 3D skenery a strukturované světlo

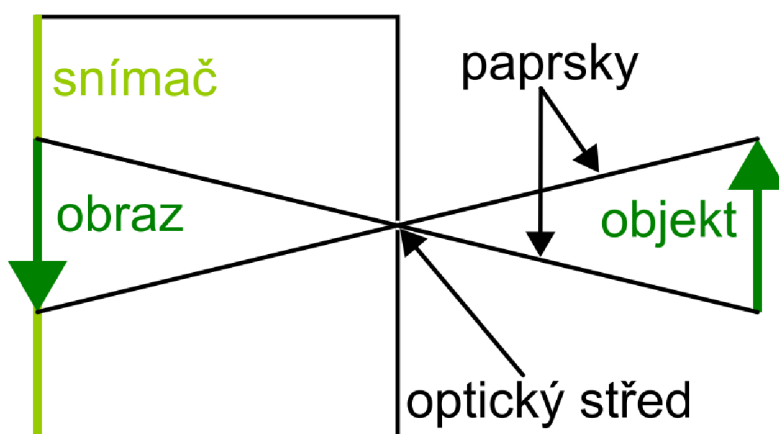
V této kapitole se budeme zabývat funkcí 3D skeneru využívajícího strukturované světlo. Zaměříme se na teoretické poznatky, které platí všeobecně pro tento druh skenerů.

4.1 Camera obscura

Dříve než začneme rozebírat funkci samotného 3D skeneru, definujeme model kamery, který nám následně usnadní práci v následujících kapitolách.

Reálné kamery využívají optických prvků, jako jsou čočky, aby dosáhly kvalitního obrazu. Kvalitní kamery využívají složité optické soustavy, které lámou světlo na fotografický čip. Ten snímá dopadající světlo a vytváří tak obraz. Počítání s takovými soustavami by pro účely 3D skeneru bylo velmi složité a museli bychom pro každou kameru zcela předělat rovnice popisující prostup světla na čip. Zavádíme proto zjednodušený model kamery Camera obscura. Někdy také nazývaný dírková komora, nebo pinhole [1].

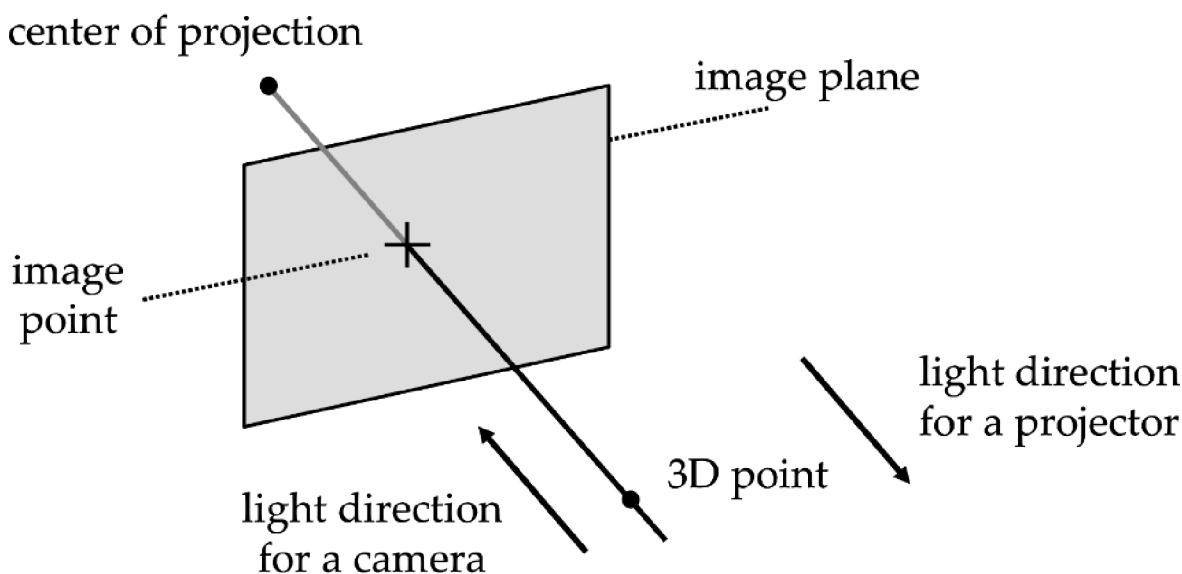
Vycházíme z toho, že všechny paprsky prochází optickým středem, který můžeme v reálném světě vyjádřit dírkou s velmi malým průměrem. Pro naše výpočty budeme brát optický střed jako jeden bod v prostoru. Za tímto bodem máme plochu, na kterou se zobrazuje obraz. Na každý bod této zobrazovací plochy se zobrazí bod, jehož světlo, které daný objekt vyzářil nebo odrazil, prošlo optickým středem. Získáváme tak převrácený obraz skutečného objektu. Tento efekt je ukázán na obrázku 4.1.



Obrázek 4.1: Zobrazení obrazu v camera obscura

V našem případě je ale lepší si tento model představit tak, že zobrazovací plochu dáme mezi optický střed a objekt, který zobrazujeme na kameře. V tomto případě zobrazovací plocha přesně reprezentuje fotografický čip. Tato zobrazovací rovina má jednotlivé body, které reprezentují jednotlivé pixely kamery. Optický střed nám následně svou vzdáleností od zobrazovací roviny určuje optickou vzdálenost kamery.

U kamery, podle obrázku 4.2, počítáme s tím, že světlo letí od bodu v trojrozměrném prostoru skrz zobrazovací rovinu do optického středu. U projektoru můžeme využít stejný model, pouze otočíme směr, kterým putuje světlo. Projektor tedy vysílá světlo z jeho optického středu. To prochází skrz zobrazovací rovinu a dopadá na bod v prostoru.



Obrázek 4.2: Využití camera obscura v 3D skeneru [1]

Následně rozdělíme zobrazovací rovinu na jednotlivé pixely kamery a projektoru, tak, že dostaneme rovinu s počtem bodů odpovídajícím rozlišení kamery, potažmo projektoru. Získáváme tedy polopřímky, které vychází z optického středu a prochází jednotlivými obrazovými body. Tyto polopřímky následně využijeme v kapitole 4.3.

4.2 Kalibrace

Cílem kalibrace kamery je získat informace o tom, jak kamera zkresluje obraz. Tedy jak kamera vidí rovné linie v reálném světě. Tyto informace následně můžeme použít k tomu, abychom obraz narovnali a zbavili se tak zkreslení. Tedy abychom na výsledné, zpracované fotografii viděli rovné hrany opravdu rovné, a ne prohnuté vlivem různých nedokonalostí reálné kamery [7].

Ke kalibraci používáme fotografie objektu, u kterého známe přesné rozměry jednotlivých obrazců. V praxi je nejpoužívanější kalibrační šachovnice s černými čtverci na bílém podkladu. Software nalezne na fotografii rohy čtverců šachovnice, zjistí jejich souřadnice na fotografii a spočítá posunutí čtverců oproti očekávané pozici ve čtvercové síti. Následně použije transformační matice pro odstranění zkreslení. Čím více fotografií šachovnice získáme, tím přesnější kalibraci kamery získáme. Doporučuje se použití alespoň deseti fotografií z různých úhlů tak, aby šachovnice pokryla co největší část obrazu.

Z kalibrace také získáme, kromě parametrů popisujících zkreslení, takzvanou matici vnitřních parametrů [7]

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

kteřá určuje optické vlastnosti kamery a je specifická pro danou kameru. To znamená, že jakmile kameru jednou zkalibrujeme, můžeme tuto kalibraci použít pro jakékoli další fotografie pořízené touto kamerou. V matici 4.1 reprezentují f_x a f_y ohniskové vzdálenosti v daných směrech a c_x a c_y reprezentují optické středy, které nemusí korespondovat se středem souřadnicového systému kamery.

Stejně tak, jako kalibrujeme kameru, potřebujeme kalibrovat projektor, který také promítá nedokonalý, zkreslený obraz. Navíc také pro účely optické triangulace, viz. kapitola 4.3, musíme zjistit polohu projektoru vůči kameře. Využijeme tedy šachovnice a zároveň promítáme strukturované světlo projektoru na šachovnici. Jsme tak schopni získat natočení a posunutí projektoru vůči kameře, neboli vnější parametry projektoru a kamery. Také jsme schopni získat matici vnitřních parametrů projektoru, která je tvarem stejná jako matice vnitřních parametrů kamery.

Posunutí kamery vůči projektoru popisuje translační vektor, který má tvar

$$\mathbf{T} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (4.2)$$

Rotaci kamery vůči projektoru popisuje rotační matice. Rotační matice je matice 3x3, která je složená ze 3 složek, každá popisující rotaci kolem jedné z os. Z kalibrace kamery s projektorem dostaneme přímo celkovou rotační matici ve tvaru

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (4.3)$$

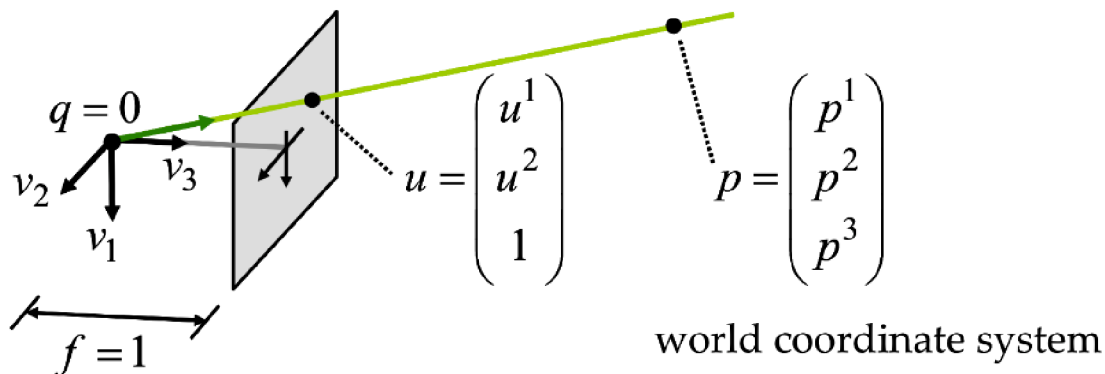
4.3 Optická triangulace

Principy optické triangulace využíváme k získání bodů v prostoru. V našem případě získáváme body povrchu skenovaného objektu. Jak napovídá název metody, k získání bodu v prostoru využíváme trojúhelníku, jehož vrcholy tvoří projektor, kamera a samotný skenovaný předmět.

Nejdříve definujeme souřadnicové systémy [1], které nám zjednoduší práci s vyjádřením přímek a bodů v prostoru. Počátek prvního souřadnicového systému umístíme do optického středu kamery. Tento souřadnicový systém budeme dále nazývat souřadnicovým systémem kamery. Počátek druhého souřadnicového systému umístíme do optického středu projektoru a budeme jej nazývat souřadnicovým systémem projektoru. Souřadnicové systémy natočíme v prostoru tak, aby jejich třetí souřadnice (z-tová složka) byla normálovou vzdáleností optického středu od zobrazovací plochy a první a druhá složka

(x a y) korespondovaly s natočením pixelů zobrazovací plochy. Umístění souřadnicového systému je znázorněno na obr. 4.3.

camera coordinate system



Obrázek 4.3: Popis polopřímky a bodů v prostoru [1]

Pro projektor vytvoříme počet polopřímek, který koresponduje s počtem pixelů projektoru. Polopřímky vytvoříme tak, že počátek polopřímky umístíme do optického středu projektoru a polopřímku natočíme tak, aby procházela právě jedním pixelem, který je určen přesně pro tuto polopřímku [1]. Zjistíme tak, kam míří světlo daného pixelu projektoru. Polopřímky popisující trajektorii světla je možné popsat rovnicí

$$P = \{\mathbf{p} = \mathbf{q} + \lambda \mathbf{v} : \lambda \geq 0\} \quad (4.4)$$

kde \mathbf{q} je bod v prostoru, v našem případě optický střed. \mathbf{v} je vektor, který popisuje směr paprsku a lze jej spočítat podle obrázku 4.3 jako

$$\mathbf{v} = \mathbf{u} - \mathbf{q} \quad (4.5)$$

bod \mathbf{q} bude v případě projektoru roven

$$\mathbf{q} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (4.6)$$

U výpočtu polopřímek pro kameru budeme postupovat stejným způsobem. Nejdříve však musíme body v souřadnicovém systému kamery transformovat do souřadnicového systému projektoru, který zvolíme jako hlavní souřadnicový systém. K transformaci využijeme vnější parametry, které jsme získali z kalibrace kamery a projektoru v kapitole 4.2.

Pro jednoduchost prostorové transformace využijeme homogenní souřadnice [11], které se hojně využívají v 3D počítačové grafice a je možné díky nim velmi jednoduše zapsat prostorové transformace v maticovém tvaru. Sestavíme tedy transformační matici, s jejíž pomocí můžeme transformovat bod nebo vektor z souřadnicového systému kamery do

námi zvoleného hlavního souřadnicového systému projektoru

$$\mathbf{M} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.7)$$

Bod v homogenních souřadnicích je ve tvaru

$$\mathbf{u} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (4.8)$$

Transformaci následně provedeme jako

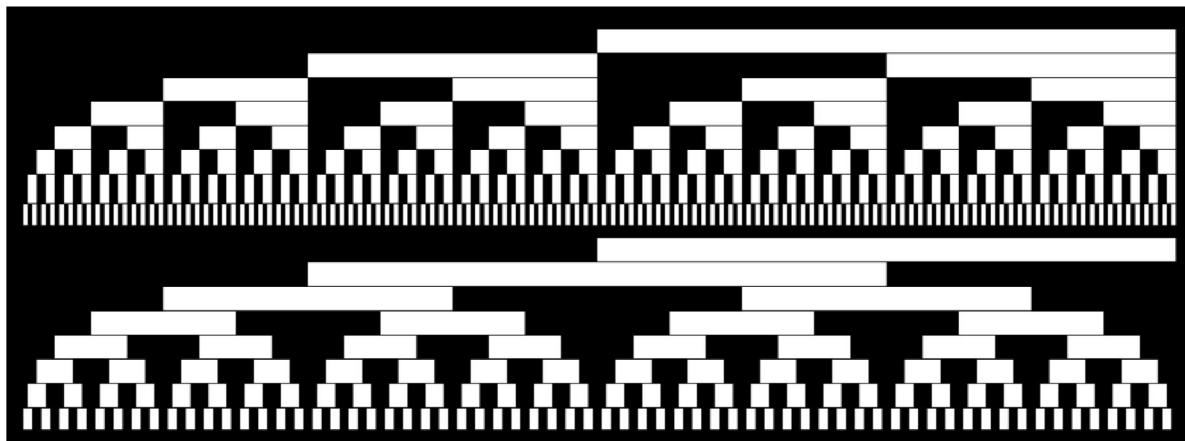
$$\mathbf{u}' = \mathbf{M}\mathbf{u} \quad (4.9)$$

Kde \mathbf{u} je bod v souřadnicovém systému kamery, \mathbf{M} transformační matice a \mathbf{u}' bod v souřadnicovém systému projektoru.

Transformované body následně využijeme pro výpočet polopřímky popisující trajektorii světla mířícího do kamery. Postupujeme stejně jako v případě projektoru dle rovnic 4.4, 4.5 a obr. 4.3.

Máme tedy rovnice popisující jak světlo z daných pixelů postupuje prostorem. Dále je potřeba zjistit na jakém pixelu kamery se zobrazí jaký paprsek vycházející z projektoru [1]. K tomu, abychom zjistili tyto závislosti mezi projektorem a kamerou využijeme právě strukturované světlo, které promítáme projektorem. Můžeme využít mnoho druhů strukturovaného světla. Nejjednodušší varianty jsou založené na promítání bílých a černých pruhů, které v sobě nesou informaci o pozici daného promítaného paprsku. Nejpoužívanější je binární a greyovo kódování. Abychom dostali informaci, kde je bílý a kde černý pruh, promítáme vždy dvojici obrázků, která má navzájem invertované barvy jednotlivých pruhů. Následně porovnáme světlost jednotlivých míst na dvou fotkách a určíme, kde je černý a bílý pruh. Na základě této informace přidělíme bílému pixelu hodnotu 1. Černému pixelu přidělíme hodnotu 0. Postupně promítneme projektorem tolik pruhů různých šířek tak, abychom byli schopni přesně určit, který pixel projektoru vidíme na jakém pixelu kamery.

Na obrázku 4.4 je srovnání binárního a greyova kódování. K tomu, abychom spolehlivě určili který paprsek vysílaný z projektoru vidíme na jakém pixelu kamery, musíme při použití binárního kódování promítat i nejjemnější pruhy, které mají šířku jednoho pixelu projektoru. Takto úzké pruhy je velmi náročné přesně rozeznat na snímači kamery a dochází tedy často k nepřesnému přiřazení pixelů. Navíc se nemění pozice přechodů mezi bílými a černými pruhy a mírné nepřesnosti na hranách mohou vést k výrazné nepřesnosti výsledného skenu. Těmto problémům se můžeme vyhnout použitím greyova kódování. K určení všech pixelů v tomto případě nepotřebujeme šířku slouce o jednom pixelu projektoru. Dosahujeme toho posouváním přechodů bílých a černých pruhů. Greyovo kódování následně můžeme převést na binární kód s pomocí převodních vztahů dle obr. 4.5.



Obrázek 4.4: Srovnání binárního (nahore) a greyova kódování (dole) [1]

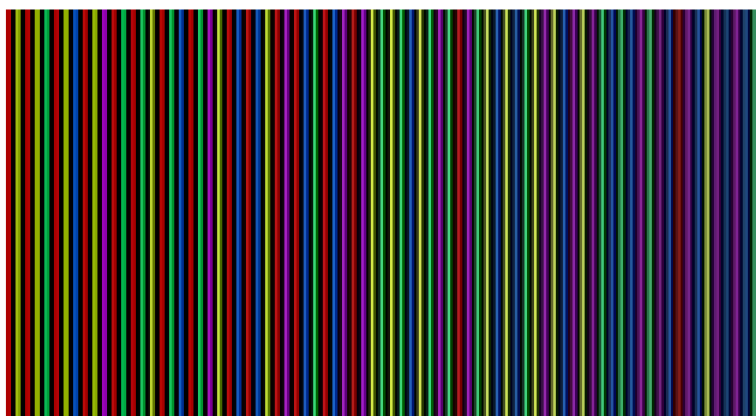
```

GRAY2BIN( $G$ )
1   $n \leftarrow \text{length}[G]$ 
2   $B[1] \leftarrow G[1]$ 
3  for  $i \leftarrow 2$  to  $n$ 
4    do  $B[i] \leftarrow B[i - 1] \text{ xor } G[i]$ 
5  return  $B$ 

```

Obrázek 4.5: Převod greyova kódování do binárního kódování [1]

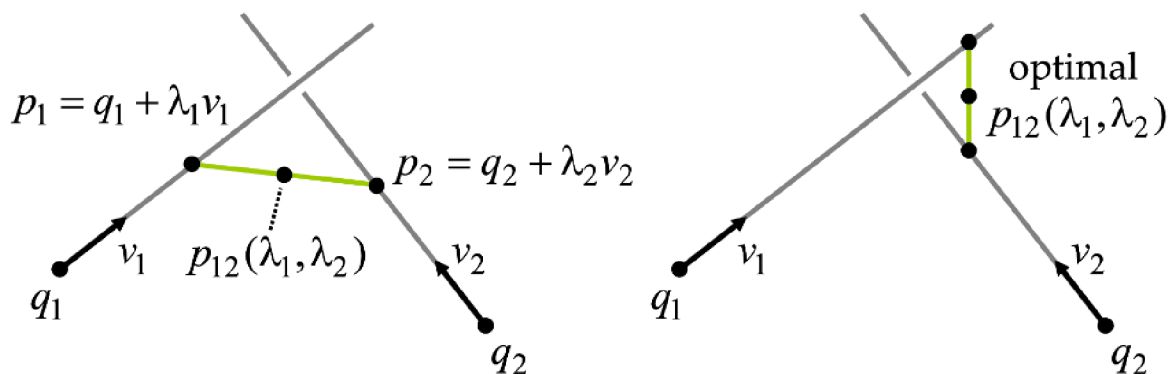
Je možné využít také různých barevných vzorů strukturovaného světla [3]. Barevné strukturované světlo je navrženo tak, aby bylo podle okolí daného bodu poznat kde se v daném vzoru nacházíme. Nejčastěji se proto setkáváme s barevnými pruhy, jejichž pruhy jsou poskládány v každém místě do jedinečného uspořádání. Častou výhodou u barevného světla může být, že k provedení skenu stačí promítnutí jediného vzoru. Naopak nevýhodou bývají problémy při skenování vícebarevných objektů. Jednou z možných barevných variant jsou barevné pruhy De Bruijn zobrazené na obrázku 4.6.



Obrázek 4.6: De Bruijn barevné pruhy [3]

Víme-li jaký pixel projektoru koresponduje s jakým pixelem kamery, můžeme vzít polopřímky vytvořené pro tyto pixely a hledáme jejich průsečík. Průsečík těchto dvou polopřímek je bod, který leží na povrchu skenovaného předmětu.

Přímky se ovšem v prostoru v reálných aplikacích nikdy neprotnou. Proto použijeme přibližné průsečíky [1], pro aproximaci reálných bodů. Přibližné průsečíky hledáme v polovině nejkratší vzdálenosti, mezi korespondující polopřímkou z kamery, a polopřímkou z projektoru. Vyjádříme tedy vzdálenost mezi dvěma body jako vzdálenost mezi bodem na jedné a druhé polopřímce a hledáme její minimum pomocí parciálních derivací prvního řádu, které musí být v minimu rovny nule. To vidíme na obrázku 4.7. Vlevo máme vyjádření rovnicemi s průsečíkem v obecném bodě mezi paprsky. Na pravé části obrázku vidíme již minimální vzdálenost mezi polopřímkami, s optimálním průsečíkem v polovině vzdálenosti. Úsečka, která reprezentuje nejkratší možnou vzdálenost mezi dvěma paprsky je vždy kolmá na obě polopřímky v prostoru. Vypočítáme-li průsečíky pro všechny dvojice polopřímek, dostaneme celý sken povrchu z daného pohledu kamery.



Obrázek 4.7: Přibližný průsečík dvou polopřímek [1]

4.4 Filtrace bodů

V rámci skenování vznikají chyby, které generují body v místech, kde se žádná část reálného objektu nenachází. Tyto chyby vznikají především na hranách skenovaných objektů, nebo případně vlivem odrazů světla. Také může docházet ke špatnému určení dvojic paprsků, které následně vygenerují nechtěný bod. Tyto chyby jsme schopni eliminovat za použití určitých metod filtrace bodů, především porovnáním umístění bodu v porovnání s ostatními body.

První možností filtrace, kterou odfiltrujeme především body vzniklé špatným přiřazením dvojice paprsků, je filtrování bodů, které mají příliš velkou minimální vzdálenost mezi paprsky.

Další možností je filtrace na základě počtu sousedních bodů [6]. Okolo každého bodu z mraku bodů vytvoříme kouli o daném poloměru. Následně zkoumáme kolik jiných bodů je v této kouli, a všechny body, které nemají dostatek sousedních bodů v dané vzdálenosti odfiltrujeme. Tato metoda je závislá na daném poloměru a při skenování např. velkých objektů se může stát že žádný z bodů nebude mít dostatek sousedních. Naopak při skenování malého předmětu se může stát, že všechny body budou kritérium splňovat, a filtrování tedy nebude efektivní.

Jednou z dalších možností je také odfiltrování bodů, které jsou statisticky dále od ostatních, než zbytek bodů. Je tedy potřeba určit kolik sousedních bodů je potřeba vzít v úvahu při výpočtu vzdálenosti od ostatních. Dalším parametrem je agresivita filtru, která určuje kolik bodů z celkového množství má být odfiltrováno. Takovýto druh filtru není závislý na velikosti skenu a je jej možné využít poměrně univerzálně.

Kombinací různých způsobů filtrování jsme schopni dosáhnout vyfiltrování i velkého množství bodů, které do výsledného skenu nepatří.

4.5 Vytváření 3D modelu

Prozatím jsme získali shluky bodů, které jsme naskenovali z jednotlivých pohledů kamery. Pro vytvoření 3D modelu skenovaného objektu je potřeba tyto pohledy sloučit do jednoho, celkového. Sloučení je jednou z náročnějších částí 3D skenování a pro naše účely je možné využít buď ručního, nebo automatického složení pohledů. Ruční skládání závisí na vstupu uživatele, který vybere na každém z pohledů nejméně 3 významné body. Na základě výběru se následně body přiřadí k sobě a poskládá se výsledný 3D sken. Automatické možnosti jsou poněkud náročnější a pro jednoduché aplikace jako náš 3D skener přichází v úvahu pouze možnost kalibrace otočného stolu.

Kalibrace otočného stolu probíhá na základě kalibračního obrazce, který umístíme na otočný stůl. Otáčíme otočným stolem s kalibračním obrazcem a získáváme po zpracování osu, kolem které se stůl v prostoru otáčí. Otáčení kolem osy následně využíváme při skládání jednotlivých pohledů.

Jakmile máme mraky bodů poskládané do jednoho výsledného, můžeme vygenerovat samotnou 3D strukturu skenovaného objektu. K generování 3D struktury je popsáno mnoho metod, které popisují možnosti jak generování provést. Základními možnostmi se budeme dále zabývat.

Některé metody využívají ke své funkci normály. Normálou myslíme vektor, který míří ven z bodu. Normály můžeme vypočítat s pomocí dostupných funkcí nejrůznějších software. Také je můžeme určit při skenování povrchů předmětů.

4.5.1 Využití základních tvarů

Metoda využití základních tvarů [6] vytvoří velký 3D objekt kolem všech bodů, které v prostoru máme. Následně probíhá odebírání materiálu tam, kde nejsou žádné body, které strukturu definují. Vytváření struktury si můžeme představit jako bychom vzali nástroj pro odebírání materiálu, tedy jakousi lžici. Touto lžicí odebereme materiál všude tam, kde se dostaneme bez toho, abychom se se lžicí potkali s jakýmkoli bodem, kolem kterých strukturu vytváříme. Základním parametrem metody je parametr alfa, který udává velikost nástroje, kterým materiál odebíráme. Máme-li tedy velký parametr alfa, máme i velký nástroj pro odebírání materiálu a odebereme malé množství materiálu. S velkým parametrem alfa tedy dostaneme málo detailní struktury, které jsou velmi ovlivnitelné chybnými body nacházejícími se mimo náš skenovaný objekt. Máme-li však malý parametr alfa, můžeme bez kolize s body odebrat velké množství materiálu. Dostáváme tedy detailnější model. Při zvolení příliš malého parametru alfa naopak máme příliš malý nástroj, takže můžeme odebrat i materiál mezi body povrchu skenovaného objektu a dostáváme následně strukturu s velkým počtem děr. Metoda je také velmi obtížně použitelná pro shluky bodů, které mají v různých svých místech různou bodovou hustotu. Pro takové shluky bodů je možné, že v místech s větší hustotou bodů bude struktura málo detailní, a v místech s

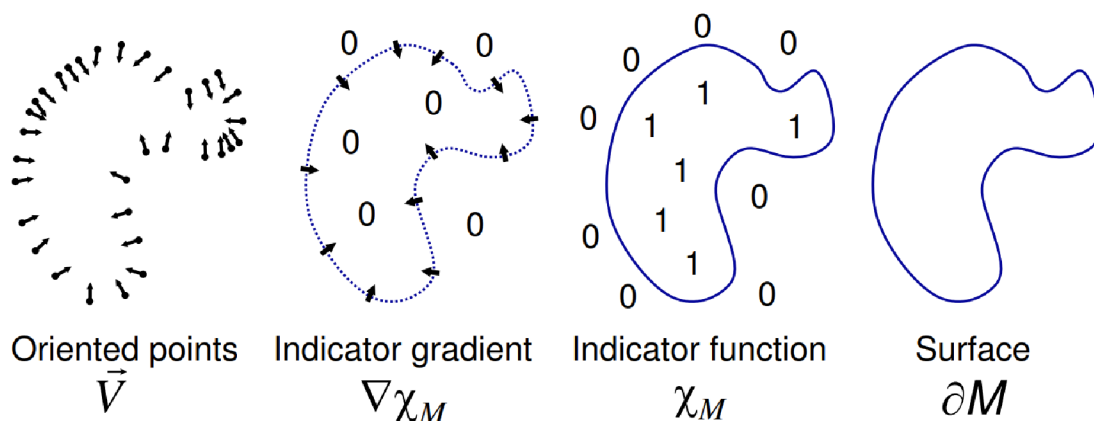
menší hustotou bude struktura plná děr.

4.5.2 Ball pivoting

Metoda ball pivoting [12] generuje koule o daném poloměru, které používá k hledání trojic bodů. Mezi trojicí bodů je následně vygenerována trojúhelníková struktura. Nejdříve metoda najde tři body, které se vejdu do vygenerované koule a vytvoří trojúhelník. Následně postupujeme od již vygenerovaného trojúhelníku. Na každý vrchol trojúhelníku vygenerujeme další kouli a hledáme další trojice bodů. Jsou-li body moc daleko od sebe koule propadne mezi body a nevytvoří se žádný další trojúhelník, který by doplnil strukturu. Vznikají tedy díry v struktuře stejně jako u metody s využitím základních tvarů.

4.5.3 Poisson surface reconstruction

Na základě normál shluku bodů jsme schopni určit, jaký bod v prostoru je uvnitř skenovaného objektu, případně mimo skenovaný objekt [2]. Všem bodům v prostoru přiřadíme hodnotu, podle toho, kde se nachází. Bodům uvnitř objektu přiřazujeme hodnotu 0, bodům mimo objekt přiřazujeme hodnotu 1. Získáváme tedy trojrozměrné skalární pole. Toto pole má hodnotu téměř na všech místech 1, pouze uvnitř objektu má hodnotu 0. Z tohoto skalárního pole můžeme získat vektorové pole, popisující změnu hodnoty skalárního pole v okolí daného bodu. Takto získáme vektorové pole s vektory mířícími do objektu na všech místech, kde dochází v skalárním poli ke změně hodnoty. To je na povrchu skenovaného objektu. Vektorové pole dále popisuje indikační funkci, kterou dále převedeme na běžnou Poissonovu rovnici, jejíž řešením je povrch skenovaného objektu. Postup práce se skalárním a vektorovým polem je názorně zobrazen na obrázku 4.8.



Obrázek 4.8: Znázornění funkce Poisson surface reconstruction v 2D [2]

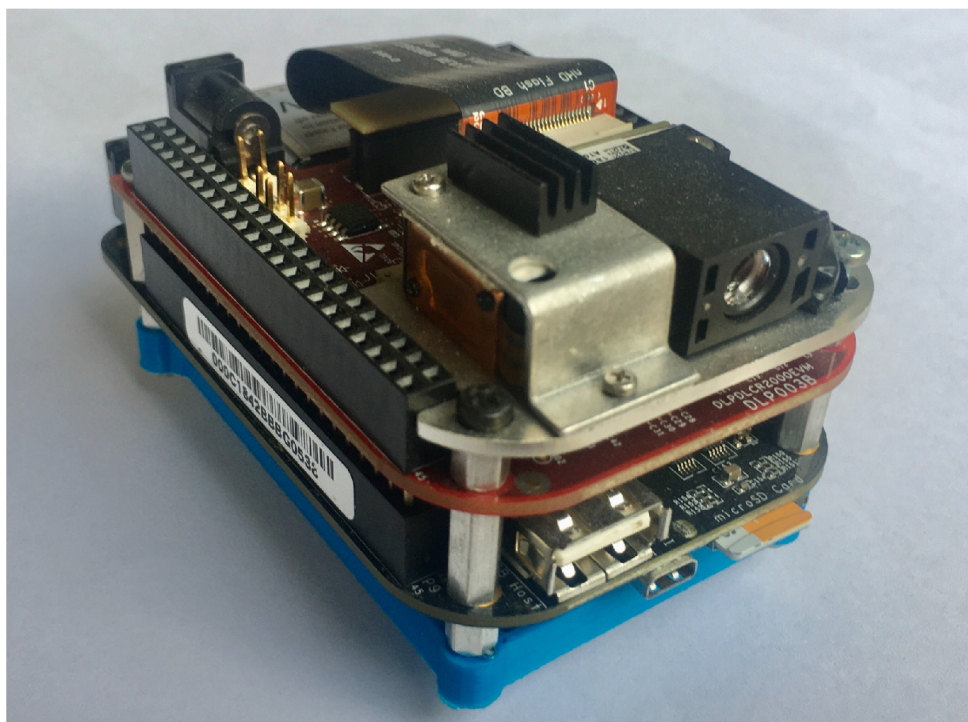
Metoda na rozdíl od výše zmíněných generuje hladký povrch, z principu nemůže generovat díry v struktuře a není příliš citlivá na změnu bodové hustoty shluku bodů. Také je schopna minimalizovat vliv šumu v shluku bodů, protože na rozdíl od dříve zmíněných metod nemusí být body shluku bodů přímo součástí struktury 3D objektu.

5 Realizace

5.1 Hardware

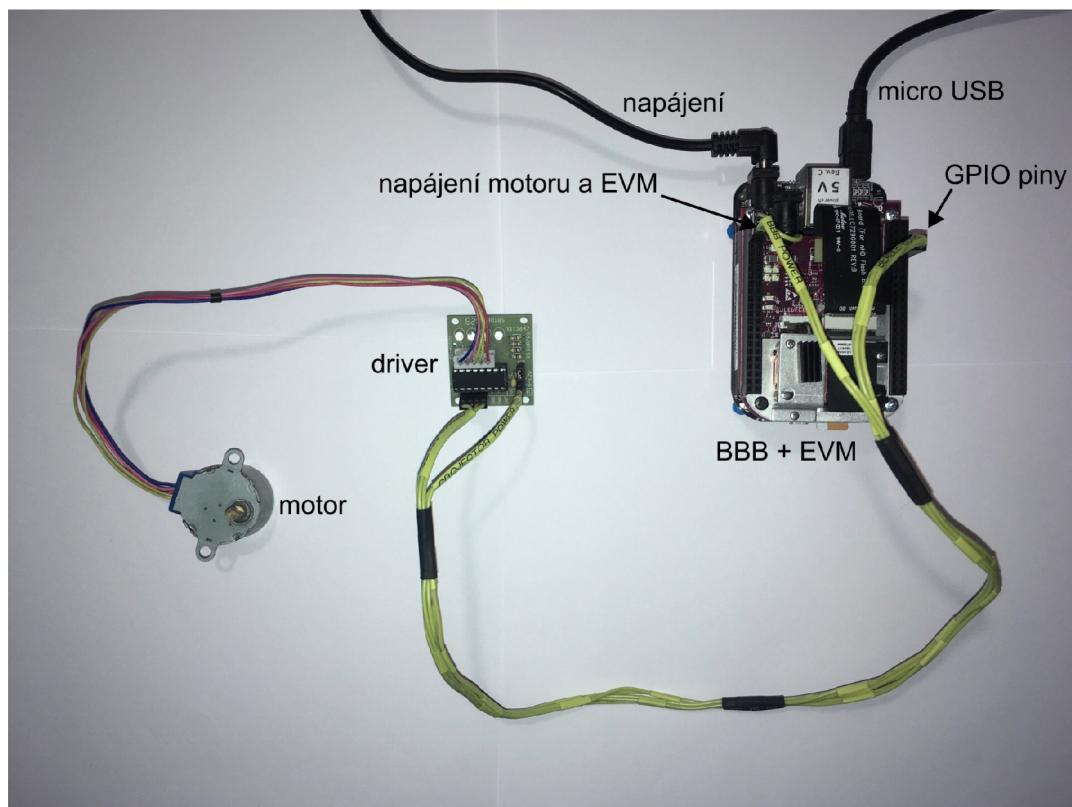
Pro sestavení jednoduchého 3D skeneru bylo využito vybavení z Mechatronické laboratoře. Finální podoby skeneru, kterou si dále popíšeme, si bylo možné všimnout v úvodu práce na obrázku 1.1.

Hlavní část skeneru je realizována pomocí BeagleBone Black (BBB) [10] s rozšířením DLP LightCrafter Display 2000 EVM (EVM) [9]. BBB je jednodeskový počítač (SBC), který je využíván především jako výpočetní a vývojářský modul. Na BBB je nainstalován linux, na kterém probíhají potřebné procesy. Rozšíření EVM se stará, pomocí optické soustavy, o promítání strukturovaného světla. BBB s EVM jsou zobrazeny na obrázku 5.1. BBB je napájen pomocí síťového adaptéru. Komunikace s počítačem probíhá přes micro USB konektor. EVM je napájeno skrz BBB z 5V výstupu (obr. 5.2).



Obrázek 5.1: BeagleBone Black s DLP LightCrafter Display 2000 EVM

Další součástí soustavy skeneru je otočný stůl, který je sestaven z desky a krokového motoru 28BYJ-48 - 5V. Krokový motor je napájen přes BBB z výstupního napětí 5V. Ovládání krokového motoru obstarávají čtyři GPIO piny, jejichž signál putuje do motoru přes ovládací jednotku (driver) krokového motoru ULN2003. Propojení je znázorněno na obrázku 5.2



Obrázek 5.2: Propojení motoru, driveru a EVM s BBB

Pro získání obrazových dat je využita USB kamera Hercules 4780733 HD Twist Mini (obr. 5.3) připojená pomocí USB k běžnému stolnímu počítači.



Obrázek 5.3: USB kamera Hercules 4780733 HD Twist Mini

5.2 Software

Bylo sestaveno několik programů, které se starají o jednotlivé funkce skeneru. Všechny soubory byly naprogramovány v programovacím jazyce python.

Dva jednoduché programy se starají o získávání fotografií pro kalibraci a samotné skenování. Přes SSH se z počítače připojíme na BBB, kde přepínáme pomocí xdttool promítané obrázky z prohlížeče obrázků Eye Of GNOME (eog). Po každém přepnutí pořídíme fotku kamerou. Na BBB je také spuštěn skript pro otáčení stolu, jehož řízení

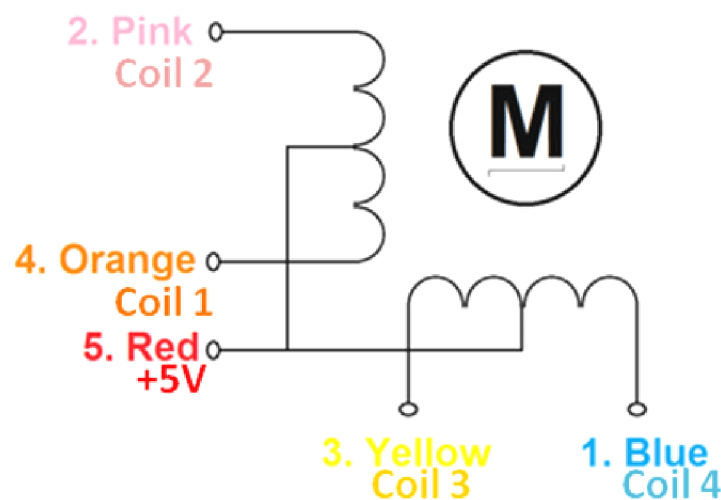
vysvětlíme v kapitole 5.2.1. Umožňuje nám to získat obrazová data z více stran objektu.

Dále byl naprogramován soubor generující sady obrázků s binárním kódem, podle potřebného rozlišení projektoru.

Generování struktury a zpracování obrazových dat je realizované v programu scanner.py, který popíšeme v kapitole 5.2.2

5.2.1 Ovládání otočného stolu

Otočný stůl je ovládán z BBB s využitím GPIO pinů GPIO2_7, GPIO2_8, GPIO2_9, GPIO2_10. GPIO piny jsou nastaveny jako výstup a logickou jedničkou spínají tranzistory v řídicím obvodu ULN2003. Sepnutý tranzistor spojí přívod do motoru se zemí a cívkou protéká proud ze zdroje napětí. Nastavíme-li na výstup logickou nulu, tranzistor rozpojí obvod a cívkou proud neteče.



Obrázek 5.4: Zapojení cívek ve statoru krokového motoru [5]

Krokový motor řídíme s plným krokem dle následující tabulky 5.1. Barvy znázorňují přiřazení pinů k jednotlivým cívkám motoru dle obr. 5.4.

| PIN | Krok 1 | Krok 2 | Krok 3 | Krok 4 | Krok 5 | Krok 6 | Krok 7 | Krok 8 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|
| GPIO2_8 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| GPIO2_10 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| GPIO2_7 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| GPIO2_9 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Tabulka 5.1: Řízení motoru s plným krokem

Součástí motoru jsou také převody, které převodují otáčení do pomala s převodovým poměrem $i=64$. Na každou otáčku rotoru připadá 8 kroků. Pro celou otočku výstupní hřídele musíme tedy provést 512 kroků.

5.2.2 Scanner.py

Program scanner.py obsahuje funkce potřebné pro zpracování obrazových dat na 3D sken. Ovládání, ukázané na obrázku 5.5, probíhá prostřednictvím příkazového řádku, kde jsou

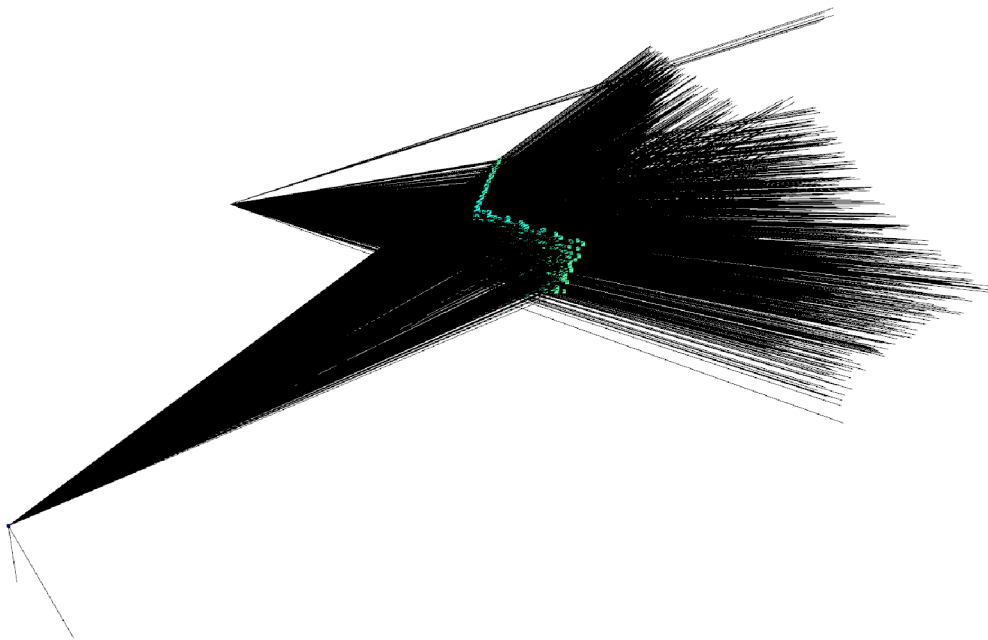
také vypisovány stručné informace o průběhu probíhajících a možnostech dalších akcí. Software je možno použít i pro jiný, než námi realizovaný, 3D skener.

```
choose action:
calibrate, new_pcl, combine_pcl, additional_scans, pcl_to_mesh, all, end, treshold, visualization
new_pcl
folder: /cropped
Save as: Box_tma_1
```

Obrázek 5.5: Ukázka ovládání z příkazového řádku

První částí je kalibrace kamery a projektoru, kterou jsme popsali v kapitole 4.2. Ke kalibraci je využito knihovny pro počítačové vidění, OpenCV [7]. Pro základ kalibračního algoritmu jsou použity funkce `calibrateCamera()` a `stereoCalibrate()`, které jsou doplněny dalšími funkcemi pro zpracování a úpravu obrazových dat. Údaje o kalibraci jsou následně uloženy do souboru `calibration.yalm` a jsou využívány pro další úkony vykonávané programem.

Další částí je triangulace a získávání bodů. Triangulace je sestavena na základě znalostí z kapitoly 4.3 a z ostatních zdrojů uvedených v seznamu literatury. Na obr. 5.6 můžeme vidět grafické znázornění triangulace implementované v našem software. Polopřímky vycházejí z projektoru a kamery. Průsečíky jsou na obr. 5.6 zelené body mezi paprsky.



Obrázek 5.6: Vykreslení polopřímek z projektoru a kamery

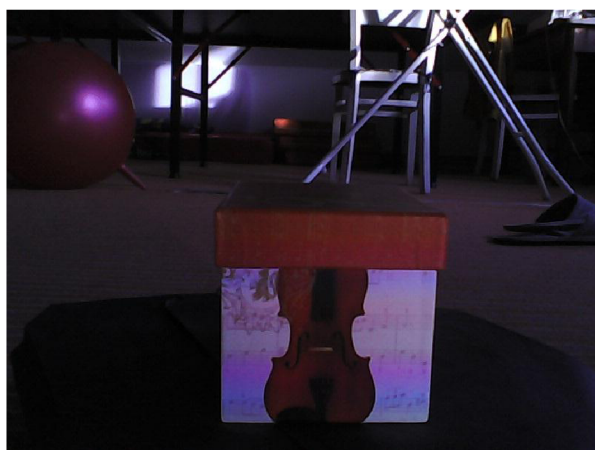
K zobrazení a další práci se získanými body je využita knihovna Open3D [6], zabývající se zpracováním a prací s 3D objekty a body. Z Open3D jsou také implementovány funkce k filtraci bodů a generování struktury předmětů.

Po dokončení každé jedné akce v programu dojde k uložení výsledku. Modely, i shluky bodů jsou ukládány ve formátu `.ply`, který je určen pro uchovávání 3D objektů. Tyto soubory je následně možné upravovat s pomocí běžného software pro práci s trojrozměrnými

objekty.

Pro správnou funkci je potřeba v úvodu programu nastavit umístění souborů v počítači a název skenovaného objektu. Dále je možné upravit rozlišení kamery a projektoru nebo důležité parametry pro chod programu, jako jsou velikost kalibrační šachovnice a práh osvětlení. Pomocí práhu osvětlení nastavujeme potřebnou míru osvětlení objektu projektoem. Na jeho základě se následně rozhodneme, svítí-li na danou část objektu projektor, nebo je daná část obrazu neosvětlená. Rozlišujeme tak body, které jsou součástí výsledného skenu. Máme-li vysoký práh osvětlení, ve shluku bodů se objeví pouze body silně osvětlené projektoem (obr. 5.8a). Při nízké hodnotě naopak získáme i málo osvětlené body (obr. 5.8b). V tomto případě ovšem můžeme získat také body, které jsou osvětleny například odrazem světla a nejsou žádané. Osvětlené pixely jsou na obrázcích bílé.

Pro názornost byly přiloženy fotografie pořízené kamerou skeneru, které využíváme k získání osvětlených pixelů. Na obrázku 5.7a se nachází objekt osvětlený projektoem. Na obrázku 5.7b potom objekt neosvětlený, tedy bez promítání.

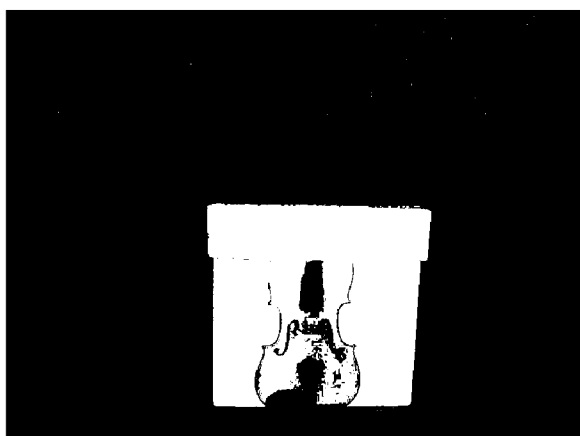


(a) Osvětlený skenovaný objekt

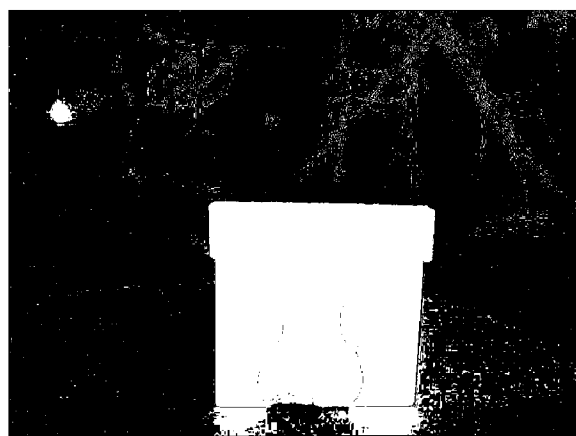


(b) Neosvětlený skenovaný objekt

Obrázek 5.7: Obrázky pro výpočet osvětlených pixelů



(a) Práh osvětlení 20



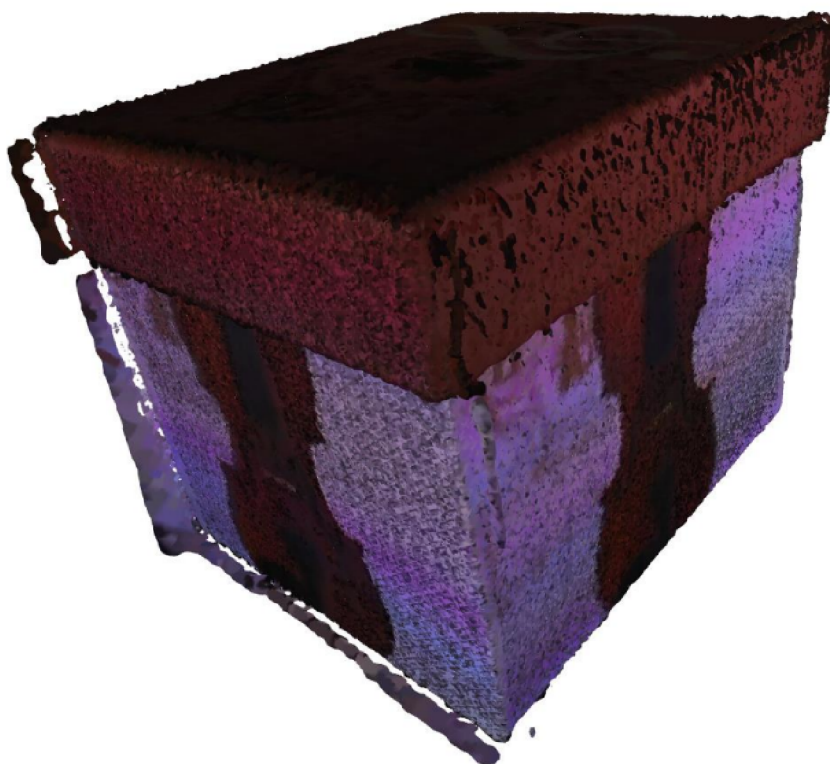
(b) Práh osvětlení 5

Obrázek 5.8: Osvětlené pixely pro různý práh osvětlení

6 Zhodnocení výsledků práce

V kapitole projdeme kvalitu modelů, které získáváme použitím postupů popsaných v předchozích částech práce. Zaměříme se na přesnost a schopnosti námi realizovaného 3D skeneru. Rozebereme podrobněji především skenování v problematických podmínkách a vliv různých povrchů a materiálů na výsledný model. Objekty, které jsou použity jako testovací, jsou vyfoceny pro srovnání se skeny na obr. 6.13.

Výsledný sken krabičky lze vidět na obr. 6.1 reálná podoba je zobrazena na obr. 6.13a.

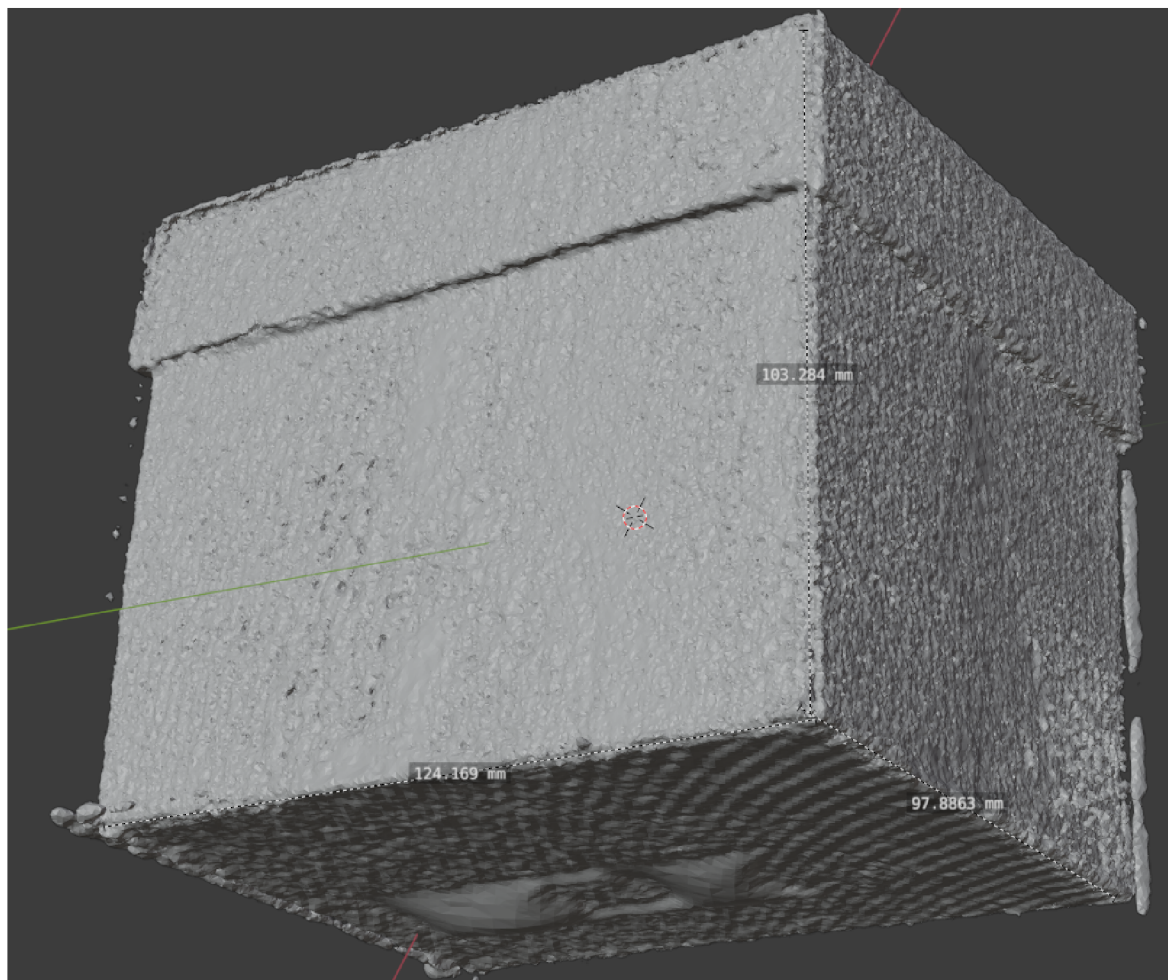


Obrázek 6.1: Finální model krabičky

6.1 Geometrická přesnost

Geometrická přesnost modelů, které získáme skenováním je přirozeně jedna z nejdůležitějších věcí, kterou od skeneru požadujeme. Zaměříme se proto na jednoduchý objekt, jehož rozměry lze jednoduše změřit a porovnáme je se skutečnými rozměry. Na obrázku 6.2 lze vidět vygenerovanou strukturu krabičky, vygenerovanou metodou Poisson surface reconstruction (kapitola 4.5.3), kterou měříme. Skutečné rozměry podstavy krabičky jsou

100 mm x 125 mm. Výška krabičky je 103 mm. K měření byl využit Blender [8].



Obrázek 6.2: Geometrická přesnost krabičky

Dostáváme tedy poměrně přesný model skenovaného objektu. Odchytky jsou způsobeny především nepřesným ručním skládáním jednotlivých pohledů k sobě. Rozměry podstavy jsou ovlivněny skládáním více než rozměry ostatních stran, protože model skládáme po obvodu a až následně přidáváme spodní a horní stranu krabice. U hran objektu lze také pozorovat nechtěné struktury generované na základě bodů, které se nepodařilo vyfiltrovat. Při nahrávání modelů do jiných programů musíme dávat pozor na rozměr. Soubory .ply ukládají rozměry jako bezrozměrnou veličinu. V našem software počítáme v milimetrech, musíme proto pohlídat, aby i nahraný soubor byl ve stejných jednotkách.

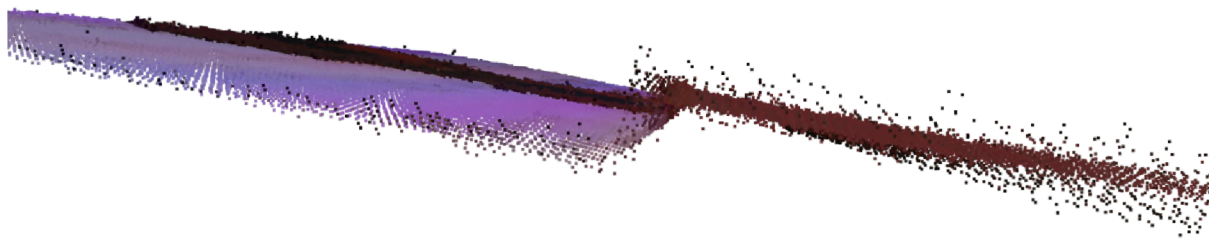
Přesnost by bylo možné vylepšit změnou postupu skládání pohledů. Kalibrace otočného stolu je jedna z možností, která by zároveň urychlila proces získávání skenu.

6.2 Vliv strukturovaného světla

Strukturované světlo popsané v kapitole 4.3 má velký vliv na průběh skenování. Na obrázcích 6.3 a 6.4 je zobrazen boční pohled na body povrchu jedné strany skenovaného objektu, kterým je krabice z obrázku 6.13a. Vidíme, že při použití greyova kódování, je povrch rovná plocha, u které jsou body umístěny na velmi přesných pozicích. Zato u binár-

ního kódování jsou body rozptýlené v blízkosti optimální plochy. Tato nepřesnost vzniká špatným přiřazením pixelů projektoru na pixely kamery, která vzniká u příliš úzkých promítaných pruhů. Problém je popsán více v kapitole 4.3.

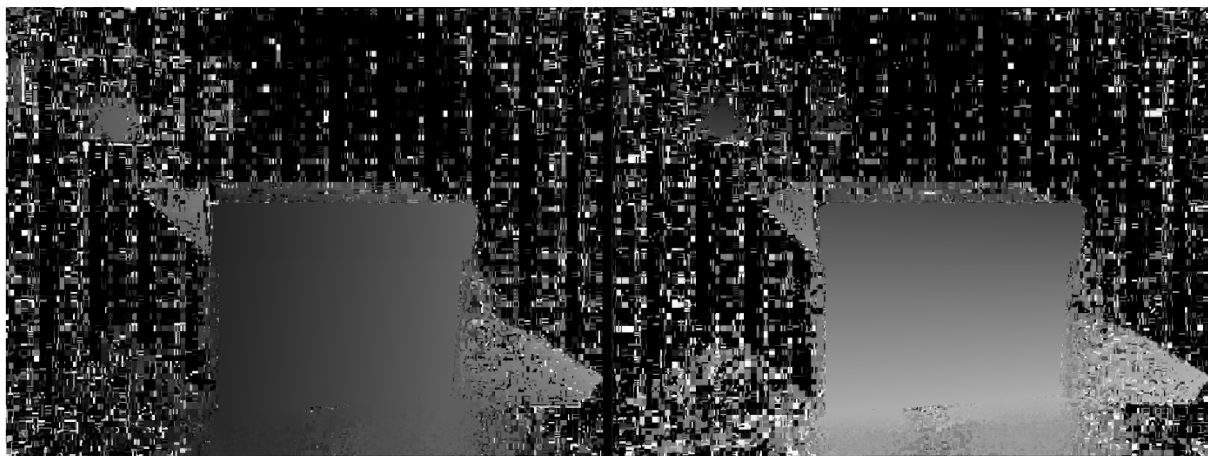
Obrázky 6.5 a 6.6 zobrazují samotné přiřazení pixelů projektoru a kamery. Na druhém z nich, zobrazujícím přiřazení u binárního kódování, vidíme pruhy pixelů, které jsou nejasně přiřazené. Tyto nejasnosti vznikají na místech, kde je při promítání přechod černého pruhu na bílý. Na prvním nic takového vidět není. To potvrzuje předpoklad z kapitoly 4.3, který již na základě teoretické úvahy uvádí greyovo kódování jako vhodnější možnost pro náš skener. K srovnání byl využit pohled na přední stranu krabíčky (obr. 6.13a).



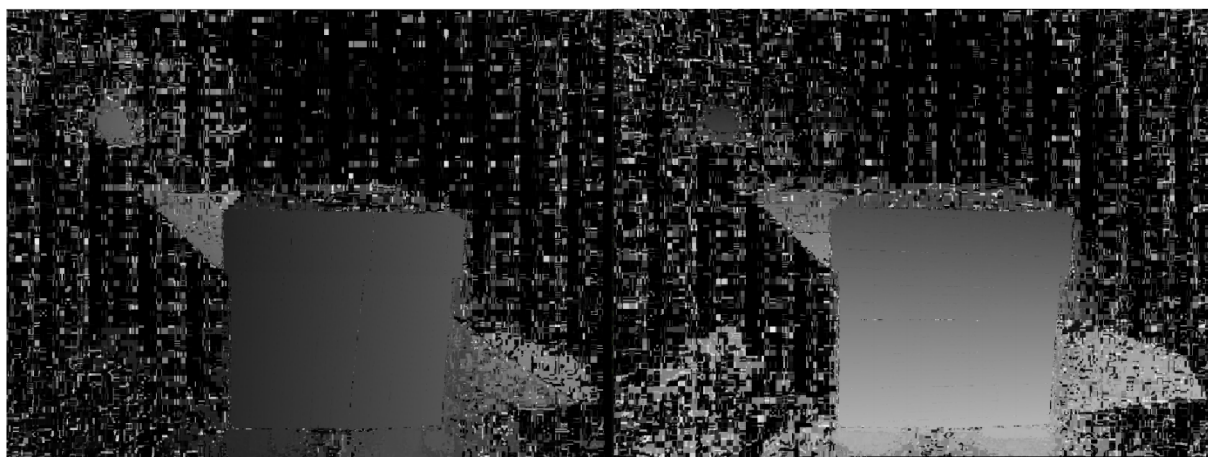
Obrázek 6.3: Body na povrchu skenu při použití greyova kódování



Obrázek 6.4: Body na povrchu skenu při použití binárního kódování



Obrázek 6.5: Přiřazení pixelů kamery k pixelům projektoru (greyovo kódování)

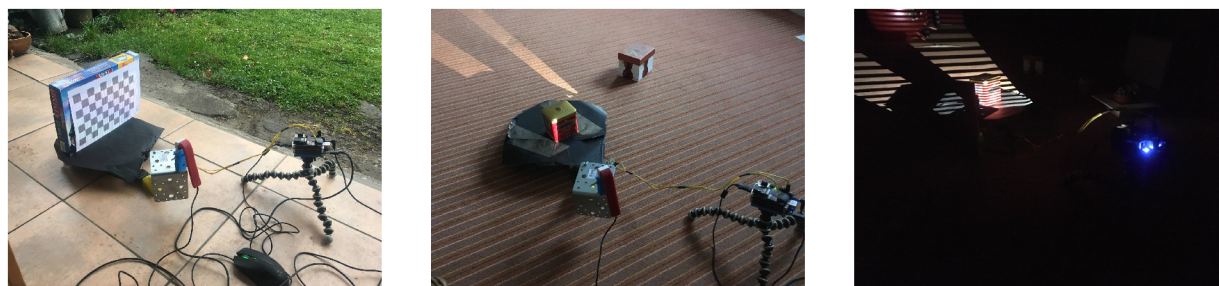


Obrázek 6.6: Přiřazení pixelů kamery k pixelům projektoru (binární kódování)

6.3 Vliv okolního osvětlení

Vliv okolního osvětlení může být velmi velký. Ve tmě nemůžeme kalibrovat, protože na fotografiích nenalezneme rohy kalibrační šachovnice. Zato skenujeme-li v příliš velkém světle, nemusíme rozpoznat místa kam svítí projektor, protože světlo vyslané z projektoru bude mít příliš malou intenzitu vůči okolí.

Na obrázku 6.9 jsou zobrazeny podmínky, ze kterých budeme dále porovnávat výsledky skenování objektů.



(a) Venku (silné světlo)

(b) Místnost (mírné světlo)

(c) Tma

Obrázek 6.7: Srovnání okolního osvětlení

Pro porovnání výsledků jsme vybrali květinu postavenou na bločku papíru. Na obrázcích 6.8a a 6.8b, které byly pořízeny za tmy a mírného osvětlení, nelze najít výrazné rozdíly v kvalitě skenů. Velký rozdíl lze ovšem vidět při porovnání s daty získanými ze skenování při silném okolním osvětlení (obr. 6.8c). Okolní světlo umožňuje věrnější zachycení barev objektu, ale zároveň také způsobuje problémy s rozeznáním osvětlených částí povrchu. Takový povrch je následně plný děr. Tento problém lze pozorovat především na bloku papíru, na kterém je květináč postaven.

Intenzita dopadajícího světla je také silně ovlivněna natočením věci kterou skenujeme. Skenujeme-li plochu kolmou k projektoru (obr. 6.9a), vysílané světlo se jeví silnější. To pozitivně ovlivňuje spolehlivost algoritmu. Plochy pod úhlem jsou osvětleny méně (obr. 6.9b) a může vznikat problém s příliš velkou intenzitou okolního světla.

Barvy u skenů by bylo možné ovlivnit např. přisvětlením objektu pro účel získání barvy. Toho by mohlo být dosaženo externím osvětlením ovládaným přes BBB. Jako další možnost vylepšení se nabízí kontrola expozice kamery při pořizování potřebných fotografií.



(a) Květina skenovaná ve tmě

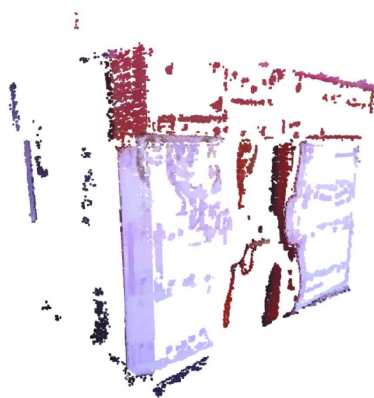


(b) Květina skenovaná v místnosti



(c) Květina skenovaná venku

Obrázek 6.8: Skeny dle okolního osvětlení



(a) Špatně rozpoznaná nasvětlená místa



(b) Správně rozpoznaná nasvětlená místa

Obrázek 6.9: Srovnání natočeného a nenatočeného povrchu při silném okolním osvětlení

6.4 Vliv materiálu objektů

Materiál objektů výrazným způsobem ovlivňuje funkčnost skeneru. Vyzkoušíme proto, jsme-li schopni skenovat i materiály, které mají vlastnosti výrazně ovlivňující chování světla.

Matné objekty jsou nejméně problematickými a již jsme ukázali na obrázku 6.1, že s jejich skenováním nejsou spojeny výraznější problémy.

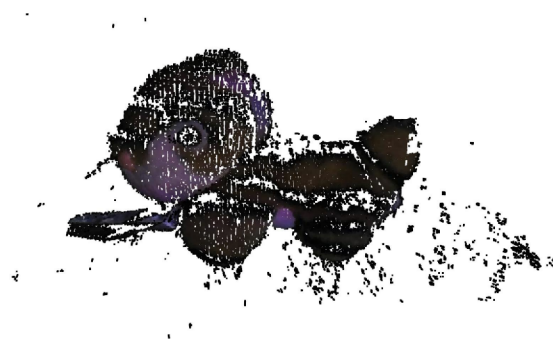
Náročnější může být získávání modelů lesklých objektů, které zvláště v náročných světelných podmínkách mohou vytvářet nejrůznější odlesky světla. V běžných podmínkách není skenování příliš náročné a dosahujeme výsledků srovnatelných s matnými povrchy. Sken lesklé plechové nádoby (obr. 6.13d) je zobrazen na obrázku 6.10.



Obrázek 6.10: Lesklá plechová krabička

Velmi pohltivé materiály jako plyš nebo látka představují další materiály, ovlivňující výrazně skenování. Na obrázku 6.11a je zobrazen sken plyšového tygra (obr. 6.13b). Vidíme, že především černé pruhy pohltivého materiálu se nenaskenovaly, protože změna osvětlení daného místa byla velmi nízká. Tento problém lze odstranit pomocí nastavení nižšího prahu světelnosti (kapitola 5.2.2). Získáme tak výsledek dle obr. 6.11b, kde již máme naskenovány i černé pruhy. Kolem se ovšem začíná objevovat výrazný šum, který již nedokážeme vyfiltrovat algoritmy popsány v kapitole 4.4 a musí být ořezán ručně.

Velmi problematické je skenování průhledných objektů a s našim skenerem takové skenování není možné. Strukturované světlo prostupuje skrz celý skleněný objekt a narušuje tak rozpoznání správného přiřazení pixelů. Vznikne tak shluk bodů na místě skleněného objektu, který má pouze malé množství bodů na povrchu objektu. Také vzniká velké množství šumu okolo. Dosažitelný výsledek skenování takovýchto objektů zobrazuje obr. 6.12. Skenování skleněných objektů jsme testovali na skleničce 6.12b.



(a) Plyšový tygr s vysokým prahem osvětlení (b) Plyšový tygr s nízkým prahem osvětlení
Obrázek 6.11: Skenování pohltivých materiálů



(a) Sken (b) Skleněný objekt
Obrázek 6.12: Skenování průhledného objektu



(a) Krabička



(b) Plyšový tygr



(c) Květina



(d) Plechová krabička

Obrázek 6.13: Skenované objekty

7 Závěr

V rámci bakalářské práce byl s využitím vybavení z Mechatronické laboratoře sestaven a zprovozněn 3D skener využívající strukturované světlo. Hardware doplnil software, který byl naprogramován v rámci práce a stará se o chod celé soustavy. Jsme tedy schopni využít vlastní software pro všechny procesy, od zachycení obrazových dat, přes promítání strukturovaného světla, až po generování struktury 3D modelů reálných věcí.

Funkčnost celého skeneru byla pečlivě popsána v kapitole 4. Software byl navrhnut tak, aby byl schopen generovat přesné výsledky. Hodně pozornosti proto bylo věnováno kalibraci skeneru a získávání bodů, ze kterých generujeme výsledný model.

Z Mechatronické laboratoře jsme využili k sestavení skeneru BeagleBone Black a rozšíření EVM s projektorem. Dále jsme využili krokový motor, s jehož pomocí jsme schopni otáčet reálné objekty a skenovat z více stran bez zásahu uživatele. Soustava je doplněná USB kamerou připojenou k počítači. Počítač také synchronizuje činnost kamery s projektorem a otočným stolem pomocí SSH.

Podářilo se tedy splnit cíle práce a ověřit schopnost vytvoření 3D skeneru i s pomocí levného, běžně dostupného vybavení, kterým kamera i projektor je. Zároveň jsme tak vytvořili základ skeneru, který je dále možno rozvíjet a vylepšovat.

Úpravy mohou pokračovat jak na straně hardware, kde můžeme zvolit výkonnější projektor a kvalitnější kameru, tak na straně software, kde je možnost automatizovat skládání pohledů na výsledný sken, nebo optimalizovat a zkvalitňovat postupy zpracování obrazových dat. Velkým vylepšením by také mohlo být vytvoření GUI, které by zpříjemnilo, zjednodušilo a urychlilo ovládání.

Literatura

- [1] TAUBIN, Gabriel, MORENO, Daniel a LANMAN, Douglas. 3D scanning for personal 3D printing. In: *ACM SIGGRAPH 2014 Studio on - SIGGRAPH '14* [online]. New York, USA: ACM Press, 2014, s. 1-66. [cit. 17.5.2021]. ISBN 9781450329774. Dostupné z: https://www.researchgate.net/publication/266660058_3D_scanning_for_personal_3D_printing_Build_your_own_desktop_3D_scanner
- [2] KAZHDAN, Michael, BOLITHO, Matthew a HOPPE, Hugues. Poisson surface reconstruction. In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June 26-28, 2006, ACM International Conference Proceeding Series* [online]. Ročník 256, Eurographics Association, 2006, s. 61–70. [cit. 17.5.2021] Dostupné z: <https://doi.org/10.2312/SGP/SGP06/061-070>
- [3] SALVI, Joaquin, FERNANDEZ, Sergio a PRIBANIC, Tomislav a LLADOA, Xavier. A state of the art in structured light patterns for surface profilometry. In: *Pattern Recognition* [online], ročník 43, č. 8, 2010: s. 2666–2680. [cit. 17.5.2021] ISSN 0031-3203. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S003132031000124X>
- [4] ČERMÁK, Jan. *Metody 3D skenování objektů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 44 s. Vedoucí bakalářské práce Ing. Tomáš MARADA, Ph.D.
- [5] 28BYJ-48 - 5V Stepper Motor [online]. [cit. 17.5.2021]. Dostupné z: <https://components101.com/motors/28byj-48-stepper-motor>
- [6] ZHOU, Qian-Yi, PARK, Jaesik a KOLTUN, Vladlen. Open3D: A Modern Library for 3D Data Processing. *ar-Xiv:1801.09847*, 2018.
- [7] BRADSKI, Gary, KAEHLER, Adrian. Learning OpenCV: Computer vision with the OpenCV library. "O'ReillyMedia, Inc.", 2008.
- [8] Blender Online Community. Blender - a 3D modelling and rendering package. *Blender Foundation, Stichting Blender Foundation*, Amsterdam, 2018. Dostupné z: <http://www.blender.org>
- [9] TI DLP LightCrafter Display 2000 EVM User's Guide. In: *ti.com* [online]. [cit. 17.5.2021]. Dostupné z: https://www.ti.com/lit/ug/dlpu049c/dlpu049c.pdf?ts=1605872648776&ref_url=https%253A%252F%252Fwww.google.com%252F&fbclid=IwAR1-lTg_EQee-X0Vqp5bHn84tAh1Er5qg0LI350uxaTzgjZaHokJBQil0j8

- [10] COLEY, Gerald. BeagleBone Black System Reference Manual [online]. [cit. 17.5.2021]. Dostupné z: https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf?fbclid=IwAR11Q9Uwxq9t0CtSzybXL4vPZZvqGL0ysNQ_p0fHybFIz_tsRH4q16a1yvc
- [11] Spatial Transformation Matrices. In: *Brainvoyager.com* [online]. [cit. 19.5.2021]. Dostupné z: https://www.brainvoyager.com/bv/doc/UsersGuide/CoordsAndTransforms/SpatialTransformationMatrices.html?fbclid=IwAR2Q0X_fTTos6rp7EZ1gPhcJJyB8cb8Pq0D5pn1Fv9kLjjMouQpdsnMEy1E
- [12] BERNARDINI, Fausto, MITTELMAN, Joshua, RUSHMEIER, Holly, SILVA, Cláudio a TAUBIN, Gabriel. The Ball Pivoting Algorithm for Surface Reconstruction. In: *IEEE Transaction on Visualization and Computer Graphics. Vol. 5., No. 4. October-December 1999.* [online]. s. 349-359. Electronic ISSN: 1941-0506. [cit. 19.5.2021]. Dostupné z: <https://vgc.poly.edu/~csilva/papers/tvcg99.pdf?fbclid=IwAR1aIUacWMDi5lZFPQSF7wR-UDojTjJ021qbT5m8xAeehEakTS1tY7awa1I>

Seznam využitých knihoven

Programovací jazyk:

- Python 3.8.6

Knihovny:

- OpenCV 4.4.0.46
- Open3D 0.11.2
- numpy 1.19.4
- PyYALM 5.3.1
- pexpect 4.8.0
- fswebcam 20140113
- xdotool 3.20160805.1
- Adafruit_BBIO 1.2.0
- GNOME Image Viewer (eog) 3.28.4

Seznam zkratek

- 3D** Trojrozměrný
- BBB** BeagleBone Black
- eog** Eye Of GNOME
- EVM** DLP LightCrafter Display 2000 EVM
- GPIO** General-purpose input/output (Univerzální vstupní/výstupní pin)
- GUI** Graphical users interface (Grafické uživatelské rozhraní)
- SBC** Single-board computer (Jednodeskový počítač)
- SSH** Secure Shell (Jeden z komunikačních protokolů)
- USB** Universal Serial Bus

Seznam obrázků

| | | |
|------|---|----|
| 1.1 | Realizovaný 3D skener | 9 |
| 3.1 | Dělení 3D skenerů | 11 |
| 3.2 | Laserový 3D skener využívající triangulaci | 12 |
| 4.1 | Zobrazení obrazu v camera obscura | 14 |
| 4.2 | Využití camera obscura v 3D skeneru | 15 |
| 4.3 | Popis polopřímky a bodů v prostoru | 17 |
| 4.4 | Srovnání binárního (nahore) a greyova kódování (dole) | 19 |
| 4.5 | Převod greyova kódování do binárního kódování | 19 |
| 4.6 | De Bruijn barevné pruhy | 19 |
| 4.7 | Přibližný průsečík dvou polopřímek | 20 |
| 4.8 | Znázornění funkce poisson surface reconstruction v 2D | 22 |
| 5.1 | BeagleBone Black s DLP LightCrafter Display 2000 EVM | 23 |
| 5.2 | Propojení motoru, driveru a EVM s BBB | 24 |
| 5.3 | USB kamera Hercules 4780733 HD Twist Mini | 24 |
| 5.4 | Zapojení cívek ve statoru krokového motoru | 25 |
| 5.5 | Ukázka ovládání z příkazového řádku | 26 |
| 5.6 | Vykreslení polopřímek z projektoru a kamery | 26 |
| 5.7 | Obrázky pro výpočet osvětlených pixelů | 27 |
| 5.8 | Osvětlené pixely pro různý práh osvětlení | 27 |
| 6.1 | Finální model krabičky | 28 |
| 6.2 | Geometrická přesnost krabičky | 29 |
| 6.3 | Body na povrchu skenu při použití greyova kódování | 30 |
| 6.4 | Body na povrchu skenu při použití binárního kódování | 30 |
| 6.5 | Přiřazení pixelů kamery k pixelům projektoru (greyovo kódování) | 31 |
| 6.6 | Přiřazení pixelů kamery k pixelům projektoru (binární kódování) | 31 |
| 6.7 | Srovnání okolního osvětlení | 32 |
| 6.8 | Skeny dle okolního osvětlení | 33 |
| 6.9 | Srovnání natočeného a nenatočeného povrchu při silném okolním osvětlení | 33 |
| 6.10 | Lesklá plechová krabička | 34 |
| 6.11 | Skenování pohltivých materiálů | 35 |
| 6.12 | Sken průhledného objektu | 35 |
| 6.13 | Skenované objekty | 36 |

Seznam tabulek

| | | |
|-----|--|----|
| 5.1 | Řízení motoru s plným krokem | 25 |
|-----|--|----|