

Zpracování obrazu z 3D kamery

Bc. Patrik Láník

Diplomová práce
2024



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Patrik Láník**
Osobní číslo: **A21476**
Studijní program: **N0714A150006 Automatické řízení a informatika v průmyslu 4.0**
Forma studia: **Kombinovaná**
Téma práce: **Zpracování obrazu z 3D kamery**
Téma práce anglicky: **3D camera image processing**

Zásady pro vypracování

1. Proveďte rešerši přístupů pro optickou lokalizaci objektu ve 3D prostoru.
2. Seznamte se s knihovnou SDK Realsense pro kamery Intel Realsense.
3. Vytvořte aplikaci, která je schopná určit polohu předdefinovaného objektu v prostoru pomocí 3D kamery.
4. Vyhodnotte přesnost lokalizace a vliv externích parametrů na přesnost.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. SZELISKI, Richard. *Computer vision: algorithms and applications*. Texts in computer science (Springer). London: Springer, 2011. ISBN 978-184-8829-343.
2. HOWSE, Joseph. *Learning OpenCV 4 Computer Vision with Python 3*. Third edition. Packt Publishing, 2020. ISBN 978-1-78953-161-9.
3. WOHLER, Christian. *3D Computer Vision, Efficient Methods and Applications*. Second edition. Springer, 2013. ISBN 978-1-4471-4149-5.
4. CYGANEK, Boguslaw a SIEBERT, J. Paul. *An Introduction to 3D Computer vision techniques and algorithms*. John Wiley and Sons, 2009. ISBN 978-0-470-01704-3.

Vedoucí diplomové práce:

Ing. Jakub Novák, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání diplomové práce: **8. prosince 2023**

Termín odevzdání diplomové práce: **27. května 2024**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 8. prosince 2023

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 24.5.2024

Patrik Láník, v.r.

ABSTRAKT

Diplomová práce se zabývá využitím dat z 3D kamery Intel RealSense D415 pro lokalizaci objektů ve 3D prostoru. V teoretické části je nejprve vysvětlen pojem Bin Picking, dále jsou představeny technologie a rozdělení 3D kamer a několik populárních 3D kamer je blíže popsáno. Následně se práce zabývá popisem technologií a přístupy zpracování obrazu, používané v kontextu 3D lokalizace objektů. V praktické části se práce věnuje konkrétní implementaci aplikace pro lokalizaci objektů pomocí 3D kamery Intel RealSense D415. Je popsáno vytvořené testovací pracoviště a použité knihovny v python programu. Následně jsou popsány jednotlivé kroky programu, jak je obraz ve scéně zpracován a v krátkosti je představeno grafické okno aplikace. V závěru se praktická část věnuje testování a validaci výsledků, kde je diskutována přesnost lokalizace a vliv externích parametrů na tuto přesnost.

Klíčová slova: 3D kamera, Intel RealSense, Stereo vidění, Mračna bodů, ICP, Python, Lokalizace

ABSTRACT

The diploma thesis deals with the use of data from the 3D camera Intel RealSense D415 for object localization in 3D space. The theoretical part explains the concept of Bin Picking, then the technologies and classifications of 3D cameras are introduced, and several popular 3D cameras are described in more detail. The last section of the theoretical part describes the technologies and image processing approaches used in the context of 3D object localization. The practical part of the thesis focuses on the specific implementation of an application for object localization using the 3D camera Intel RealSense D415. The created testing environment and used libraries in the Python program are described. Then the steps of the program, how the scene image is processed and a brief introduction to the application's graphical interface are presented. The conclusion of the practical part addresses the testing and validation of the results, discussing the localization accuracy and the influence of external parameters on this accuracy.

Keywords: 3D camera, Intel RealSense, Stereo vision, Point Cloud, ICP, Python, Localization

Děkuji Ing. Jakubu Novákovi, Ph.D. za konzultace a odborné vedení diplomové práce.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 BIN PICKING	10
2 3D KAMERY	12
2.1 AKTIVNÍ KAMERY	12
2.1.1 Strukturované světlo.....	13
2.1.2 Time of Flight	14
2.1.3 Laserové skenování	15
2.1.4 LiDAR skenování.....	16
2.2 PASIVNÍ KAMERY	17
2.2.1 Stereo vidění.....	17
2.2.2 Deep Learning	18
2.3 3D KAMERY NA TRHU	19
2.3.1 Microsoft Kinect v2	19
2.3.2 MESA SwissRanger SR4000	20
2.3.3 Intel RealSense D415	21
3 ZPRACOVÁNÍ OBRAZU A 3D DAT	22
3.1 ZPRACOVÁNÍ OBRAZU	22
3.1.1 Barevné prostory	22
3.1.2 Preprocessing	23
3.1.3 Segmentace obrazu.....	23
3.1.4 Postprocessing	24
3.2 FEATURE POINT DETEKCE	25
3.3 VYKRESLOVÁNÍ	25
3.3.1 Rasterizace	25
3.3.2 Ray tracing	26
3.3.3 Wireframe rendering	27
3.4 DYNAMICKÉ 3D REKONSTRUKČNÍ METODY	27
3.4.1 Dynamická 3D rekonstrukce založená na RGB-D kameře.....	27
3.4.2 3D Gaussian Splatting (3DGS)	28
3.5 METODY 3D REKONSTRUKCE ZALOŽENÉ NA STROJOVÉM UČENÍ	28
3.5.1 Statistické učící metody	28
3.5.2 Deep learning metody	28
3.5.2.1 Neural Radiance Fields (NeRF).....	28
3.6 HLOUBKOVÉ MAPY (DEPTH MAPS)	29
3.7 MRAČNA BODŮ – POINT CLOUD	29
3.7.1 Datový formát PCD.....	30

3.8	VOXEL SAMPLING	31
3.9	TEMPORAL FILTR.....	31
3.10	SPATIAL FILTR.....	32
3.11	ODSTRANĚNÍ OUTLIERŮ	32
3.12	ICP – ITERATIVE CLOSEST POINT.....	32
II	PRAKTICKÁ ČÁST	33
4	POPIS ZAŘÍZENÍ	34
4.1	ARUCO ZNAČKY.....	35
5	INTEL REALSENSE D415.....	36
6	KNIHOVNA SDK REALSENSE – LIBREALSENSE2.....	38
7	DALŠÍ POUŽITÉ KNIHOVNY	39
7.1	OPENCV	39
7.2	OPEN3D	39
7.3	CUSTOMTKINTER	39
8	VÝBĚR PŘEDMĚTŮ	40
9	VYGENEROVÁNÍ POINT CLOUDŮ HLEDANÝCH OBJEKTŮ VE FORMÁTU PCD	41
9.1	FUNKCE VÁLCE	42
9.2	FUNKCE KVÁDRU	43
10	KROKY PROGRAMU	44
11	POPIS APLIKAČNÍHO PROSTŘEDÍ.....	53
12	TESTOVÁNÍ A VALIDACE	55
12.1	VÝPOČETNÍ ZAŘÍZENÍ.....	55
12.2	VYHODNOCENÍ PŘESNOSTI LOKALIZACE	55
12.2.1	Pozice papírové ruličky.....	56
12.2.2	Pozice papírové krabice	57
12.2.3	Úhel papírové ruličky.....	58
12.2.4	Úhel papírové krabice	59
12.3	VYHODNOCENÍ VLIVU EXTERNÍCH PARAMETRŮ NA PŘESNOST LOKALIZACE	60
	ZÁVĚR	61
	SEZNAM POUŽITÉ LITERATURY.....	62
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	66
	SEZNAM OBRÁZKŮ	68
	SEZNAM TABULEK.....	70
	SEZNAM PŘÍLOH.....	71

ÚVOD

Zpracování 3D obrazu je v současnosti jedním z nejvýznamnějších a nejdynamičtěji se rozvíjejících témat, neboť nachází široké uplatnění v mnoha oblastech, jako je robotika, průmyslová automatizace, virtuální a rozšířená realita či medicína. Lokalizace objektů ve 3D prostoru nám umožňuje přesně určit jejich polohu, což je klíčové pro efektivní fungování aplikací v těchto oblastech. Díky této technologii mohou například robotické systémy manipulovat s předměty s vysokou přesností, průmyslové stroje automatizovat výrobní procesy nebo lékařské přístroje provádět složité operace.

V teoretické části této práce je nejprve vysvětlen pojem Bin Picking, což je technologie používaná v průmyslové automatizaci pro vybírání a manipulaci s náhodně rozmístěnými objekty. Tento koncept je zásadní pro automatizované systémy, které musí efektivně pracovat s neuspořádanými předměty. Dále jsou představeny technologie a rozdělení 3D kamer a několik populárních 3D kamer je blíže popsáno. Následně se práce zabývá popisem technologií a přístupy zpracování obrazu, používané v kontextu 3D lokalizace objektů.

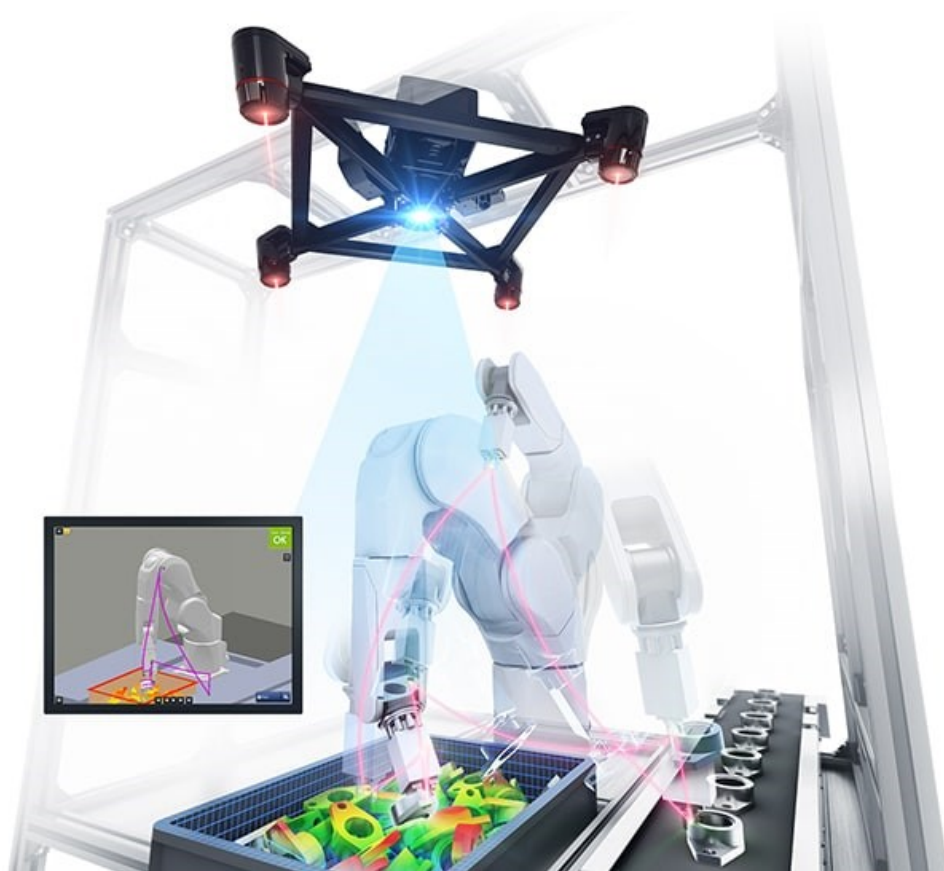
Praktická část práce se věnuje konkrétní implementaci aplikace pro lokalizaci objektů pomocí 3D kamery Intel RealSense D415. Nejprve je popsáno vytvořené testovací pracoviště a použité knihovny v python programu, mezi které patří librealsense2, která poskytuje rozhraní pro práci s kamerou Intel RealSense, dále CustomTkinter pro tvorbu grafického uživatelského rozhraní a knihovny OpenCV a Open3D pro zpracování a analýzu 3D obrazu. Následně jsou popsány jednotlivé kroky programu, jak je obraz ve scéně zpracován a v krátkosti je představeno grafické okno aplikace. V závěru praktické části práce se věnujeme testováním a validací výsledků, kde je diskutována přesnost lokalizace a vliv externích parametrů na tuto přesnost.

Diplomová práce je vypracována tak, aby bylo možné ji použít pro budoucí bakalářské a diplomové práce, které mohou využít lokalizované objekty pro další aplikace.

I. TEORETICKÁ ČÁST

1 BIN PICKING

Bin Picking je robotický proces, který se využívá v průmyslové automatizaci. Pomocí kamer a senzorů umožňuje lokalizovat a uchopit náhodně uložený objekt v kontejneru, zásobníku nebo například na paletě a s tímto objektem manipulovat. Pomocí 3D kamery tedy získáváme 3D souřadnice a orientaci daných objektů, pomocí kterých robota navigujeme a vybíráme tedy optimální pořadí vybírání objektů, tak abychom ostatní objekty neovlivnili [2].



Obrázek 1. Ukázka Bin Picking aplikace [4]

Bin Picking je důležitý pro zefektivnění a zrychlení výrobního automatizovaného procesu. Pokud do výrobního procesu Bin Picking implementujeme, zautomatizujeme části, které dříve obstarával člověk z důvodu jejich variability a náročnosti pro roboty. Jedná se například o smíšené typy objektů, náhodně uložené objekty nebo kompletace objednávek. Využitím této technologie snížíme náklady na pracovní sílu a minimalizujeme lidské chyby [2,3].

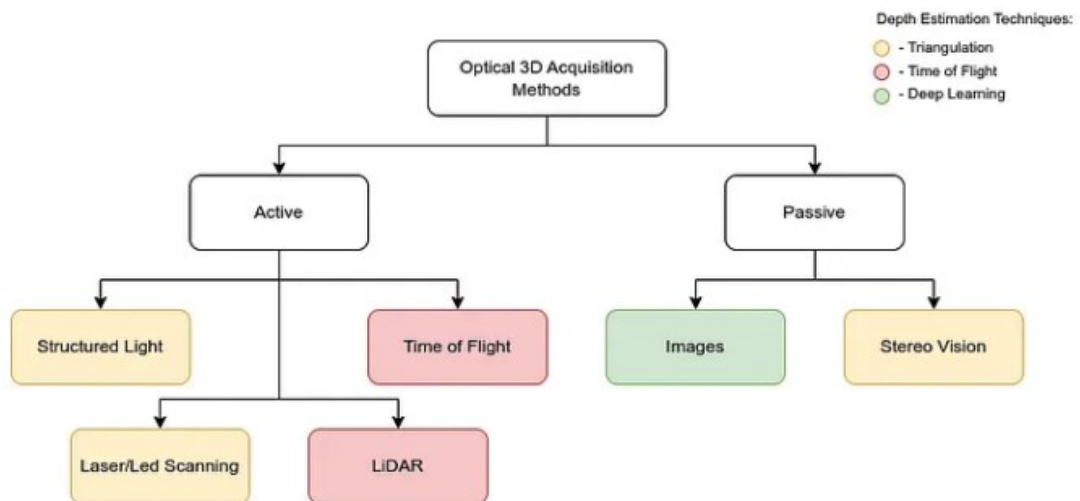
Další snížení časových prostojů můžeme eliminovat díky modulárnímu sensorovému systému, kdy dva senzory mohou skenovat dva kontejnery současně a jakmile robot jeden kontejner vyprázdní, může začít vyprazdňovat druhý bez nutnosti zastavení procesu [5].

Bin Picking proces pro svou funkčnost potřebuje integraci několika technologií, jako jsou strojové vidění, umělá inteligence, zpracování zpětné vazby ze sensorů a mechanické manipulátory. Právě strojovým viděním, přesněji optickou lokalizací objektů ve 3D prostoru se zabývá tato diplomová práce.

2 3D KAMERY

3D kamery jsou speciálním typem kamer, které nezachycují pouze obraz, ale také informace o hloubce bodů daného obrazu. Díky této technologii jsme schopni rekonstruovat 3D modely scény, která je před kamerou.

V následujícím Obrázek 2 je znázorněno, jak rozdělujeme 3D kamery podle způsobu snímání jako aktivní nebo pasivní a následně dělíme podle jednotlivých technologií, které budou v této části představeny. Techniku Triangulace používají metody Strukturované světlo, Laserové skenování a Stereo vidění, Time of Flight techniku ToF kamery a LiDAR. Hluboké neuronové sítě dokáží odhadovat hloubku také z jednoho snímku.



Obrázek 2. Rozdělení 3D kamer podle způsobu snímání obrazu a jednotlivých technologií [1]

2.1 Aktivní kamery

Aktivní kamery používají pro snímání hloubky vlastní zdroj světla, obvykle infračervené světlo, které se u techniky ToF vysílá do prostoru a měříme čas návratu do kamery po odrazu. U techniky triangulace vypočítáváme pozici na základě deformace vysílaného vzoru na objektu. Tyto kamery mají vysokou přesnost a fungují dobře i za špatných světelných podmínek. Na druhou stranu jsou nákladnější a mohou být ovlivněny okolním infračerveným rušením [11, 12].

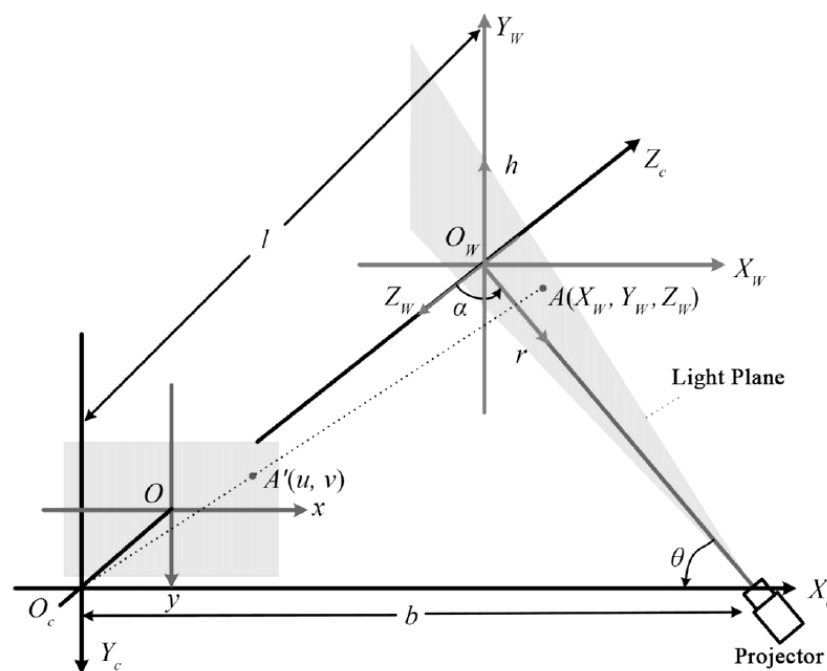
2.1.1 Strukturované světlo

Princip strukturovaného světla je založen na projekci speciálně navrženého světelného vzoru na scénu a následném snímání deformace tohoto vzoru pomocí kamery. Tato deformace je způsobena geometrickými vlastnostmi objektů ve scéně a umožňuje odhadovat jejich prostorovou strukturu.

Při použití strukturovaného světla je běžnou praxí použít projektor k vyzařování světelného vzoru na scénu. Tento vzor může mít různé formy, jako jsou mřížky, čárové kódy, nebo dokonce náhodně generované vzory. Poté je snímán obraz scény pomocí kamery. Deformace vzoru na obrazech způsobená prostorovou strukturou objektů ve scéně je analyzována a použita k rekonstrukci jejich 3D tvaru [11,14].

Tento proces je založen na principu triangulace, viz. Obrázek 3. Projekce světla a jeho deformace na povrchu objektu umožňuje vypočítat pozici každého bodu ve 3D prostoru. Na rozdíl od laserových skenerů, které pracují po řádcích, strukturované světlo umožňuje zachytit celou oblast najednou, což značně urychluje proces skenování [11,14].

Kamery strukturovaného světla vyžadují přesnou kalibraci, kvalita obrazu může být ovlivněna vlastnostmi povrchu objektu, jako vysokou odrazivostí anebo průhledností. Přímé sluneční záření může kvalitu obrazu také ovlivnit, proto se pro nejlepší výsledky doporučuje používat kameru uvnitř [1].



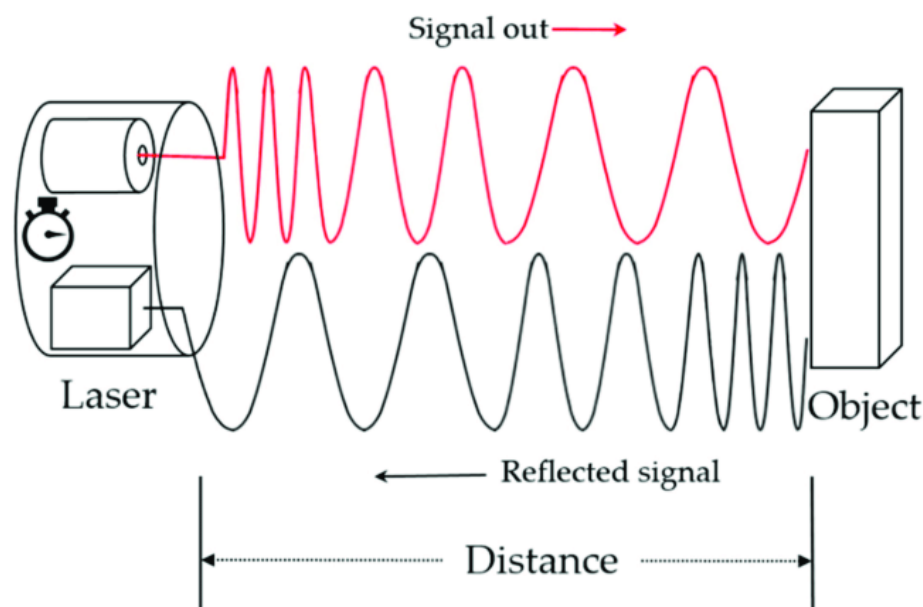
Obrázek 3. Princip triangulace strukturovaného světla [11]

2.1.2 Time of Flight

Princip časového letu pulzů (Time-of-Flight, TOF) v optické lokalizaci objektů ve 3D prostoru spočívá v měření času, který trvá světlu (běžně infračervené světlo) projít od zdroje k cíli a zpět k detektoru. Na základě tohoto naměřeného času je možné vypočítat vzdálenost od kamery k objektu, viz. Obrázek 4 [1,11].

TOF kamery obvykle využívají různé technologie pro generování světelných pulzů a detekci jejich odrazu. Mezi tyto technologie patří například modulace amplitudy světelného signálu, modulace frekvence nebo použití laserových pulzů. Po vyslání světelného signálu senzor detekuje odraz od povrchu objektu a měří dobu, která uplynula od vyslání signálu do jeho detekce. Tato doba je přímo úměrná vzdálenosti od kamery k objektu. Vytvořená hloubková mapa je 2D reprezentací 3D struktury scény a může být zkombinována s dalšími daty, například RGB obrazem z jiné kamery [1,11].

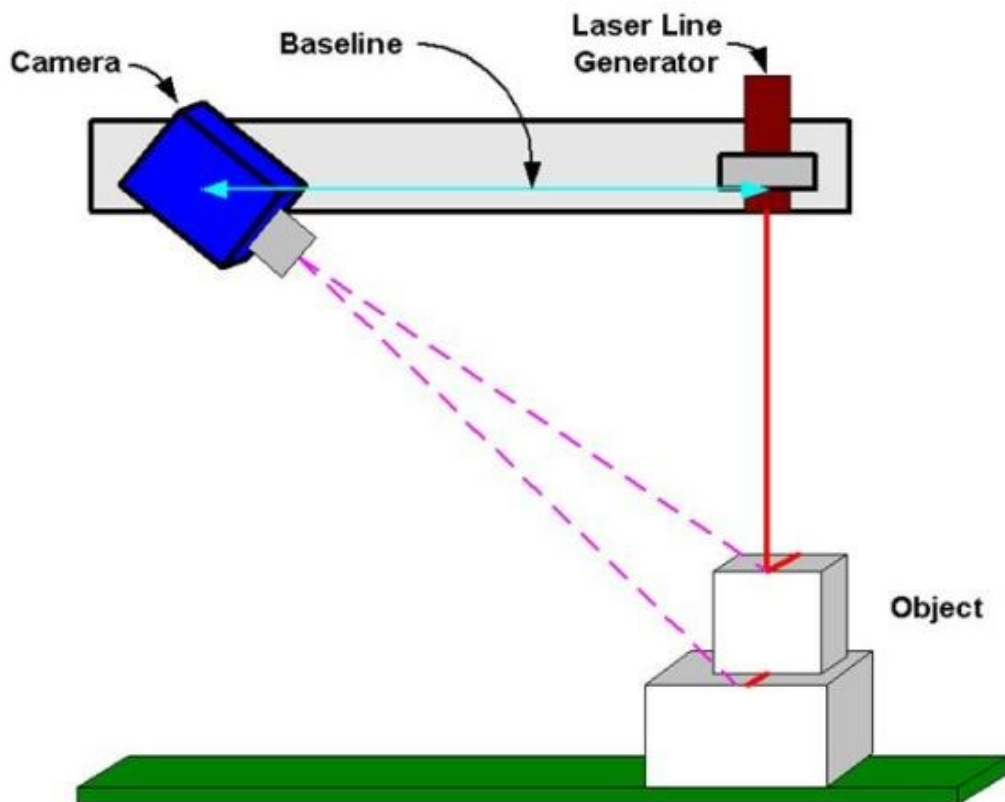
TOF kamery se běžně používají pro autonomní řízení, virtuální realitu nebo venkovní 3D skenování. Některé kamery jsou schopny pracovat při 60 snímcích za sekundu a jsou tedy ideální pro snímání v reálném čase. Další z výhod může být velká vzdálenost od snímaného objektu, která je až 20 metrů. Nevýhodou těchto kamer je nižší přesnost, rozlišení a horší výsledky u lesklých a průhledných materiálů snímaných objektů. Aktuálně mezi nejoblíbenější TOF kamery patří Kinect v2 od společnosti Microsoft a MESA SwissRanger SR4000 [1,11].



Obrázek 4. Diagram principu ToF [15]

2.1.3 Laserové skenování

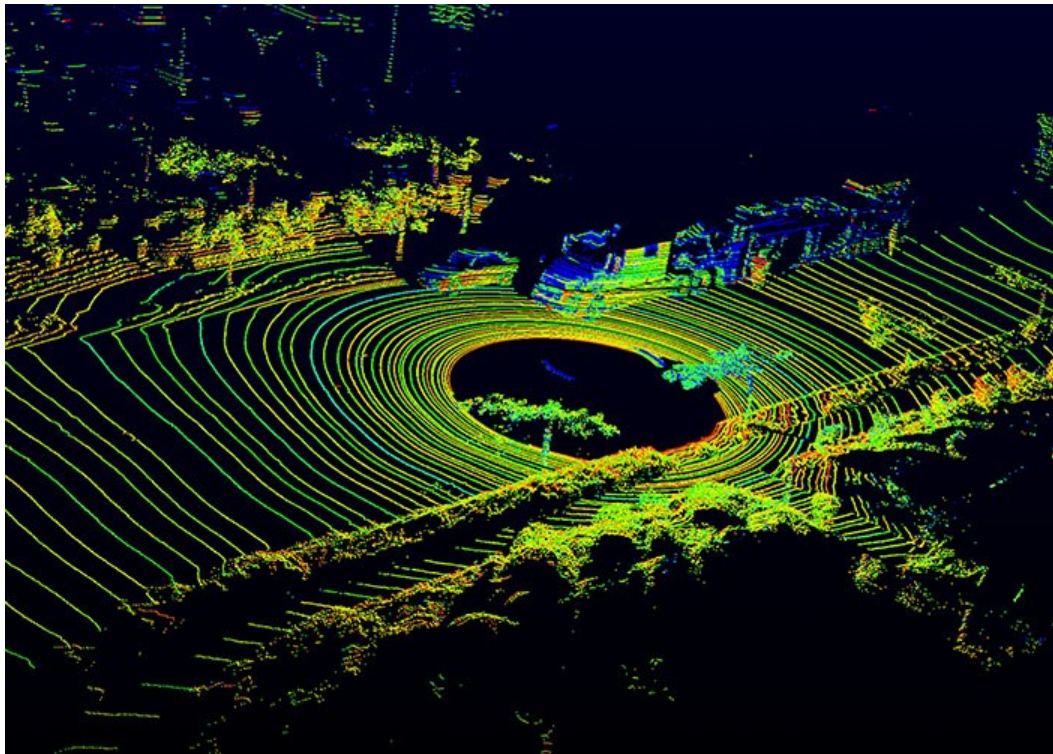
Laserové skenování kombinuje řízené usměrňování laserového paprsku s laserovým dálkoměrem, měříme odraz nebo rozptyl laseru a triangulací vypočítáváme vzdálenost na povrchu objektu, viz. Obrázek 5. Pokud by nebyl zajištěn pohyb při skenování, tak zpracujeme pouze jeden řádek 3D modelu, proto je nutné zajistit buďto pohyb kamery a projektoru nebo snímaného objektu. Tato metoda umožňuje rychlé zachycení tvaru povrchu objektů. Tato metoda je velmi přesná, rychlá a funguje i za špatných světelných podmínek. Nevýhodou je vyšší cena a jako výše zmíněné principy, strádá při snímání lesklých a průhledných objektů [1,11].



Obrázek 5. Diagram principu Laserového skenování [16]

2.1.4 LiDAR skenování

LiDAR (Light Detection and Ranging) skenování funguje na Time of Flight principu, tedy LiDAR skener oproti ToF skeneru vysílá pulzy více laserových paprsků, které se odráží od objektů a vrací zpátky ke skeneru. Samotný skener se otáčí, aby byla pokryta co největší oblast snímání nebo abychom dosáhli plného 360° záběru okolí. Následně vypočítáváme přesnou vzdálenost objektů na základě známé rychlosti světla, z těchto dat nakonec vytvoříme 3D model prostředí, viz. Obrázek 6 [1,18,20].



Obrázek 6. Ukázka výstupu LiDAR skeneru [18]

Tuto technologii například využívá společnost Google při vytváření 3D modelů map nazvaných Google Street View. Na dopravní prostředek, nejčastěji automobil, viz. Obrázek 7, se připevní LiDAR skener a poté tento dopravní prostředek projíždí infrastrukturou měst, hlavních silnic mimo města a ze získaných dat vytvoří 3D pohled na okolí kolem dopravního prostředku, který si na webových stránkách Google Map můžeme zobrazit. [18,20]



Obrázek 7. Ukázka Google Street View automobilu s LiDAR skenerem na střeše [19]

2.2 Pasivní kamery

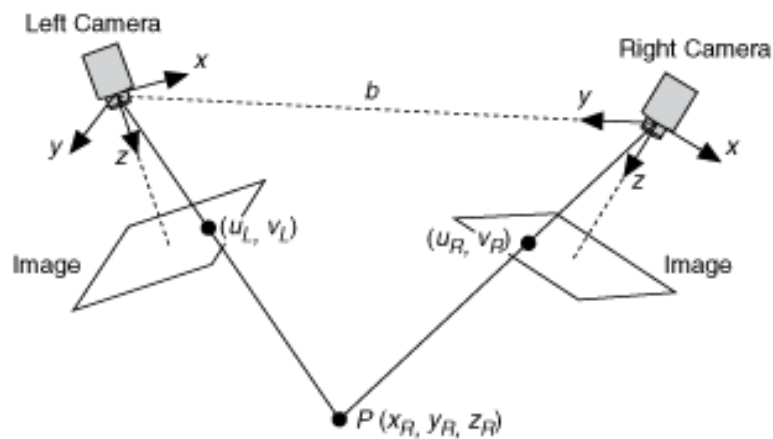
Pasivní kamery využívají stereoskopické vidění, tedy dvě kamery umístěné ve vhodné vzdálenosti pro zachycení obrazu z různých úhlů. Tyto obrazy se poté porovnají pro získání hloubkové mapy. Jsou levnější než aktivní kamery, ale oproti aktivním kamerám využívají pouze okolní světlo pro snímání hloubky, tudíž mohou být více náchylné na nedostatek světla [13].

2.2.1 Stereo vidění

Stereo vidění je klíčové pro vytváření přesných 3D map prostředí, což umožňuje robotům a autonomním vozidlům navigovat a interagovat se svým okolím. Je také zásadní pro pokročilé aplikace ve virtuální a rozšířené realitě, kde poskytuje uživatelům realistické trojrozměrné zážitky. Kromě toho se stereo vidění využívá v průmyslové inspekci a zdravotnictví, například pro navigaci chirurgických robotů [1,11].

Paralaxa je změna pozice objektu v zorném poli pozorovatele, která vzniká při pohledu z různých úhlů.

Princip stereo vidění pro optickou lokalizaci objektů ve 3D prostoru, viz. Obrázek 8, spočívá ve využití dvou kamer umístěných na různých místech, které simultánně snímají stejnou scénu z různých úhlů. Na základě paralaxního posunu mezi obrazy zachycenými oběma kamerami je možné vypočítat hloubku a pozici objektů ve scéně. Paralaxa stereo kamer je nepřímo úměrná vzdálenosti objektu od kamer. Při zvyšující se vzdálenosti objektu od kamer se hodnota paralaxy snižuje. Stereo vidění vyžaduje robustní algoritmy pro párování obrazů a výpočet hloubky. Tento proces může být realizován pomocí různých technik, jako jsou korelační metody, optimalizační algoritmy nebo metody založené na strojovém učení [1,11].



Obrázek 8. Diagram principu Stereo vidění [16]

2.2.2 Deep Learning

Dalším pasivním přístupem pro rekonstrukci 3D obrazu je hluboké učení, kdy se používá obraz z jedné kamery a jsme schopni rekonstruovat 3D obraz díky naučené neuronové síti na velkém množství učících snímků, ve kterých byla známá vzdálenost.

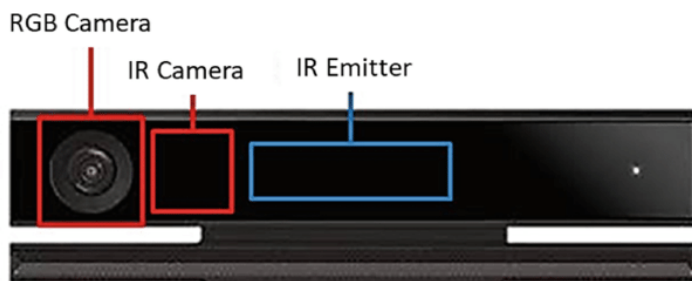
2.3 3D Kamery na trhu

Aktuální trh nabízí velké množství 3D kamer, které si lze pořídit. Nejčastěji se jedná o aktivní 3D kamery, které pracují na principu Time of Flight nebo pasivní 3D kamery na principu stereo vidění. V této kapitole si některé z kamer představíme.

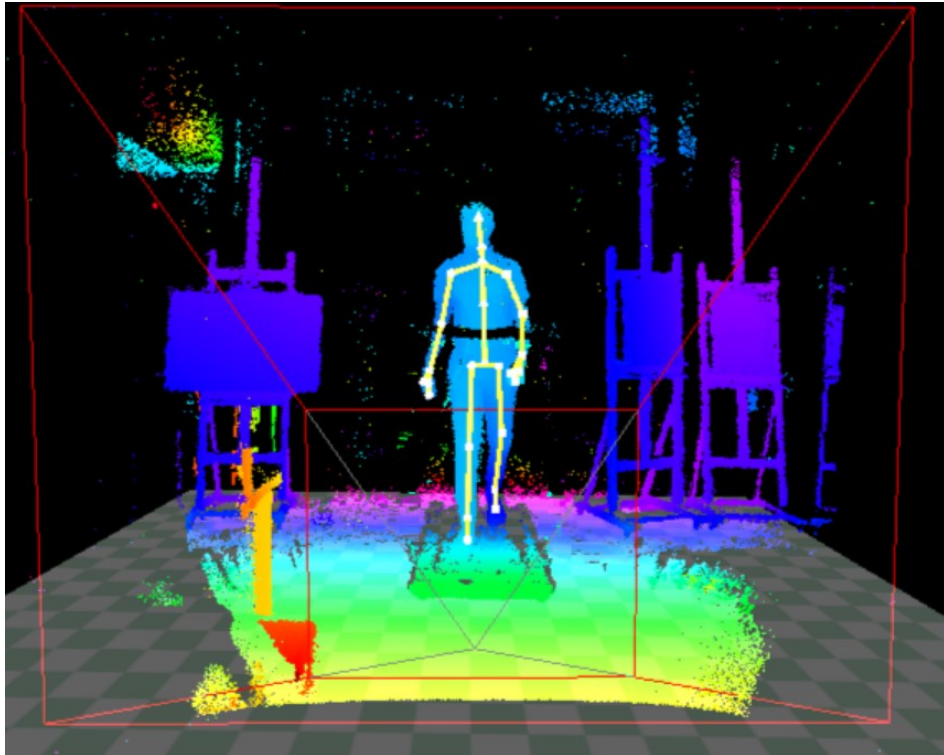
2.3.1 Microsoft Kinect v2

Původní verze této oblíbené kamery pracovala na technologii strukturovaného světla, novější verze Microsoft Kinect V2 však už používá technologii Time of Flight. Barevná kamera má rozlišení 1920 x 1080 pixelů při 30 snímcích za sekundu. Hloubková kamera má rozlišení 512 x 424 pixelů při 30 snímcích za sekundu. Připojení ke kameře je pomocí rozhraní USB-B 3.0 [24,25].

Kamera byla vyvinuta pro herní konzole Xbox One pro ovládání postavy pomocí snímaného pohybu těla. Díky sledování pohybu až 25 kloubů lidského těla, viz. Obrázek 10, lze kameru také použít při rehabilitaci pacientů ve zdravotnictví. Mimo rozpoznávání obličeje a gest se také používá pro 3D skenování objektů [24,25].



Obrázek 9. Microsoft Kinect V2 kamera [21]



Obrázek 10. Ukázka výstupu hloubkové mapy se skeletem osoby z kamery Microsoft Kinect V2 [28]

2.3.2 MESA SwissRanger SR4000

MESA SwissRanger SR4000, viz. Obrázek 11, je 3D kamera založená na technologii Time of Flight. Rozlišení kamery je 176 x 144 pixelů s až 50 snímků za sekundu. Komunikace s kamerou je pomocí USB nebo Ethernetu. Kamera se využívá ve skladech pro navigaci robotů a vyhýbání se objektům. Další aplikace kamery jsou například v oblasti medicíny při rehabilitacích nebo v zemědělství u dojíčích robotů [23,26].

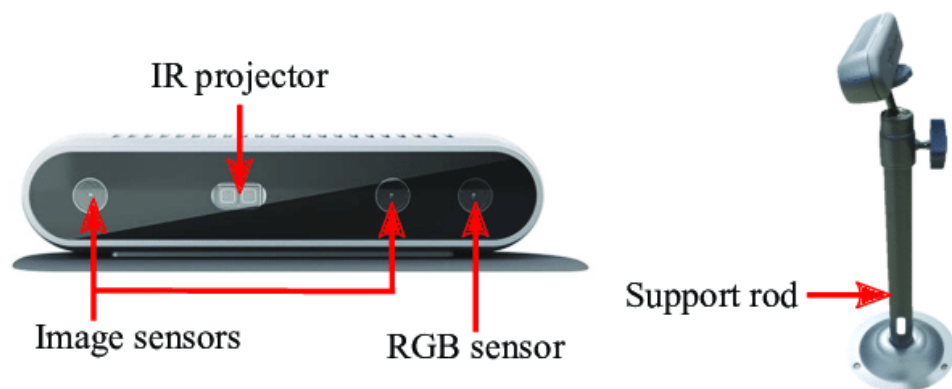


Obrázek 11. MESA SwissRanger SR4000 [23]

2.3.3 Intel RealSense D415

Intel RealSense D415 je 3D kamera pracující na technologii stereo vidění. Barevná kamera má rozlišení až 1920 x 1080 pixelů, hloubková kamera má rozlišení až 1280 x 720 pixelů při maximálně 90 snímcích za sekundu pro odpovídající rozlišení. Kamera se používá pro 3D skenování a vytváření 3D modelů a 3D map. Kamera se také používá pro rozpoznávání a sledování objektů [27].

Tato kamera bude použita v praktické části této diplomové práce a bude tedy více rozebrána v následujících kapitolách.



Obrázek 12. Intel RealSense D415 [22]

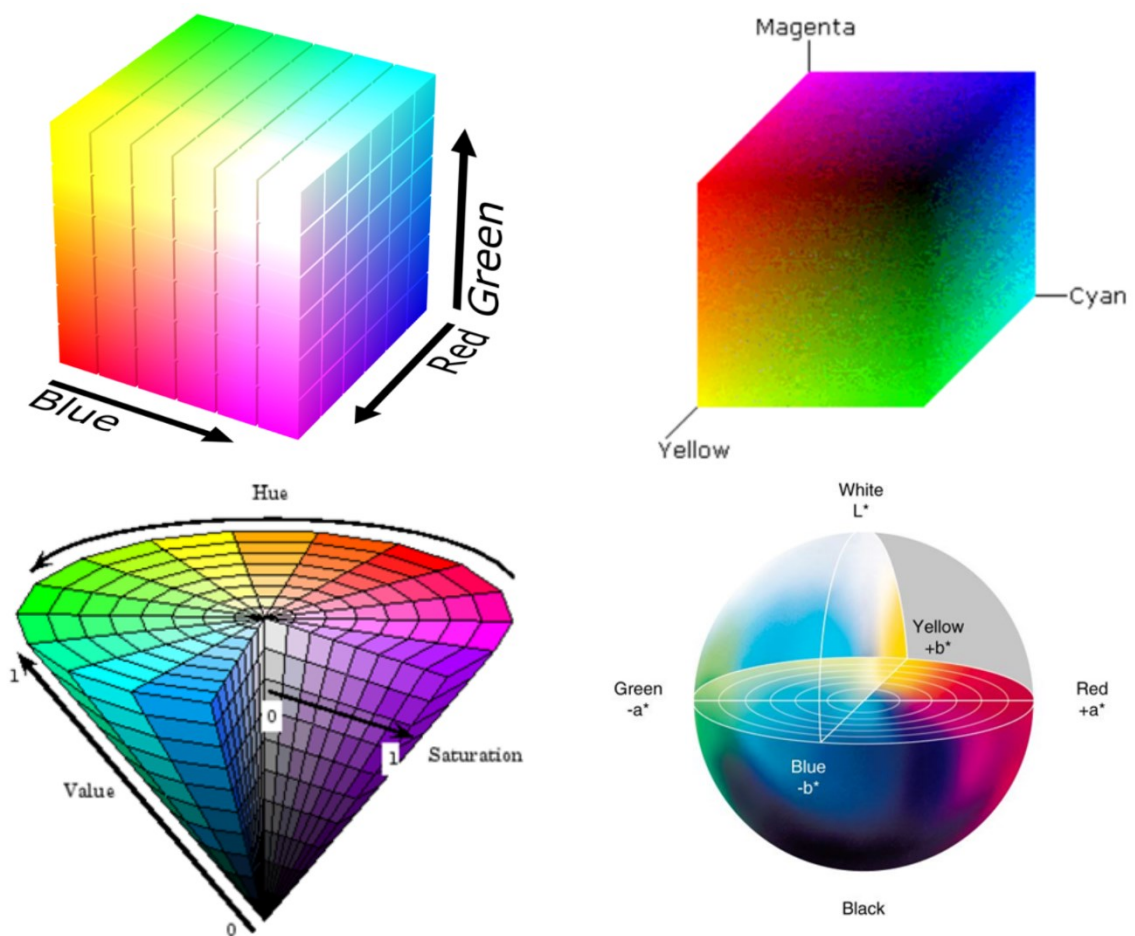
3 ZPRACOVÁNÍ OBRAZU A 3D DAT

3.1 Zpracování obrazu

Tato část stručně popisuje barevné prostory a jednotlivých částí zpracování obrazu. Této oblasti se hlouběji věnuji ve své bakalářské práci.

3.1.1 Barevné prostory

Barevný prostor je matematický model, který popisuje barvy na základě podílů jednotlivých složek barev. Nejčastěji se setkáváme s barevnými prostory RGB, CMYK, HSV a LAB.



Obrázek 13. Porovnání barevných prostorů, zleva doprava z vrchu dolů: RGB, CMYK, HSV, LAB [6,7,8,9]

3.1.2 Preprocessing

V analýze obrazu se preprocessing zaměřuje na úpravu obrazových dat před jejich dalším zpracováním. Účelem této fáze je například potlačení šumu nebo zlepšení kontrastu. Mezi používané metody patří diskretní dvourozměrná konvoluce s Gaussovým jádrem nebo metoda s filtrem Mean-shift [10].



Obrázek 14. Ukázka preprocessingu – porovnání obrázku před a po použití filtru mean-shift [10]

3.1.3 Segmentace obrazu

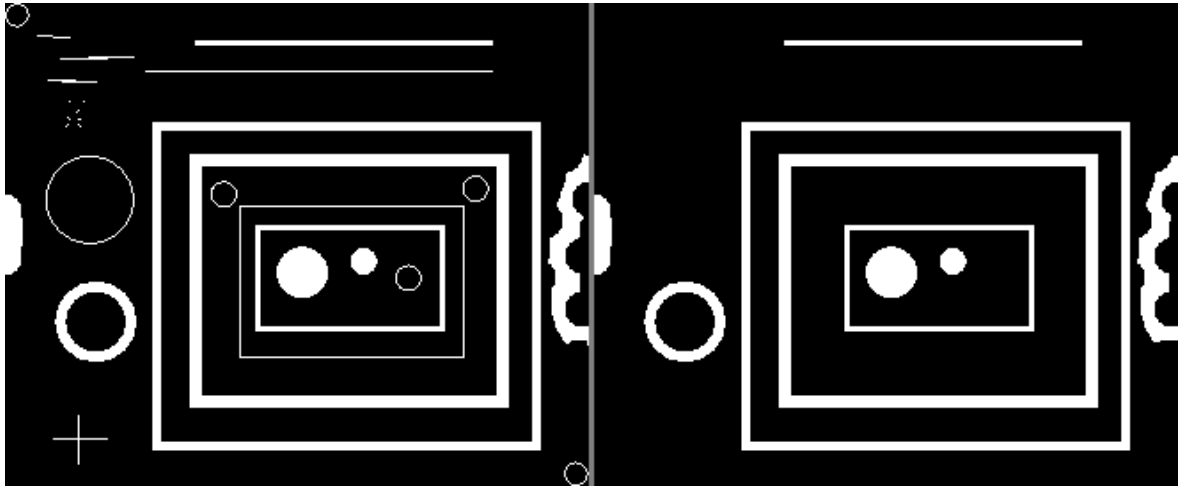
Segmentace obrazu zahrnuje několik postupů, které umožňují rozčlenit obraz do různých segmentů na základě společných charakteristik. Tento proces obvykle zahrnuje oddělení objektů od pozadí. Mezi metody segmentace patří například Prahování a metoda Rostoucích regionů [10].



Obrázek 15. Ukázka segmentace obrazu – porovnání originálního obrázku s obrázkem, na který je aplikována metoda prahování [10]

3.1.4 Postprocessing

Tato disciplína se zaměřuje na další úpravy dat po absolvování kompletního zpracování obrazu. To může zahrnovat řešení problémů, jako je nesprávné přiřazení pixelů během segmentace obrazu, spojování menších oblastí do větších s podobnými vlastnostmi, nebo odstranění oblastí, které se svými charakteristikami nehodí do celku. Součástí jsou morfologické operace jako dilatace a eroze a jejich kombinace [10].



Obrázek 16. Ukázka postprocessingu – porovnání obrázku před a po použití metody morfologického otevření [10]

3.2 Feature point detekce

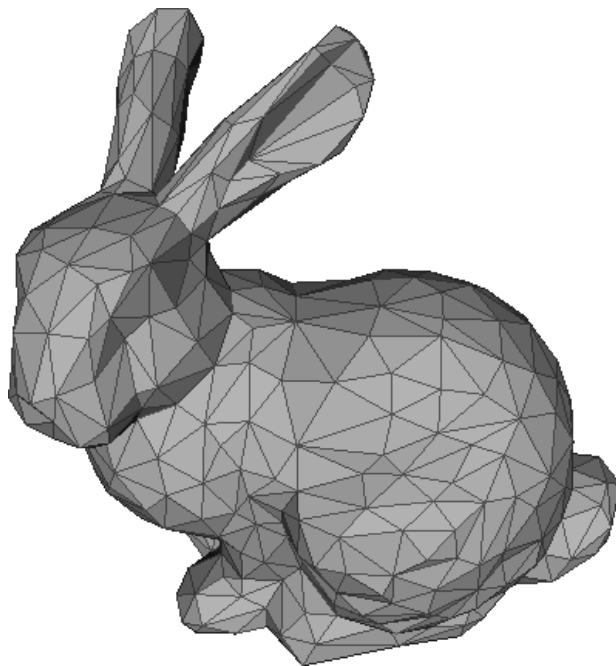
Detekce bodů příznaků, tedy feature point detection, je technika, která hledá důležité body v obraze, jako jsou například rohy nebo hrany. Tyto body se vybírají tak, aby byly invari-
antní vůči změnám, jako změna intenzity, měřítko, rotace, atd. Tyto body se používají pro
další zpracování v oblasti 3D zpracování obrazu. Mezi nejoblíbenější algoritmy pro detekci
bodů příznaků patří například Harrisův detektor rohů, ANMS, SURF nebo SIFT detektor
[11,13].

3.3 Vykreslování

Vykreslování v oblasti 3D obrazu je proces zobrazení trojrozměrné scény do dvojrozměr-
ného obrazu. Existuje několik metod, které se vykreslováním zabývají. Několik vybraných
bude níže popsáno.

3.3.1 Rasterizace

Metoda Rasterizace je jedna z nejpoužívanějších metod vykreslování 3D objektů v počíta-
čové grafice. Metoda funguje na převodu geometrických objektů do rastrového, dvojrozměr-
ného formátu spojených trojúhelníků, kde s rostoucím množstvím trojúhelníků, roste přes-
nost převedeného objektu [31,32].



Obrázek 17. Ukázka vykreslení objektu pomocí
metody Rasterizace [32]

3.3.2 Ray tracing

Ray tracing je metoda vykreslování 3D objektů, která simuluje chování světelných paprsků na povrchu objektů a celkový dopad na prostředí scény. Metoda pracuje například s odrazy světla, lomem světla a absorpcí. Tato metoda vytváří realistické a fyzikálně velmi přesné obrazy a stává se stále oblíbenější v oblasti 3D grafiky [30].

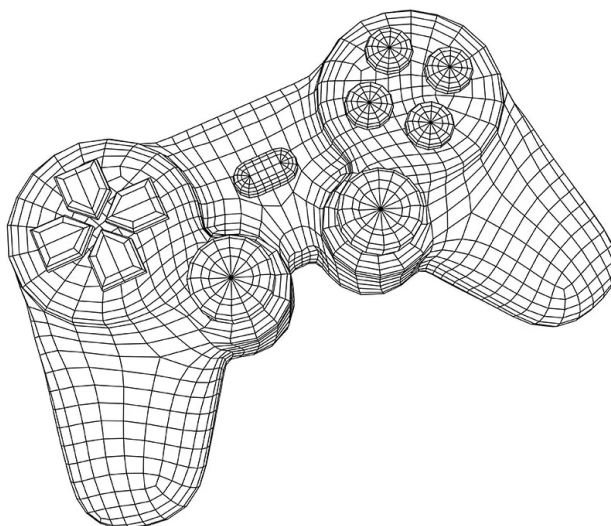
Příkladem může být Obrázek 18 níže. Světelný paprsek dopadl na zrcadlo, ve kterém se odrazil na sklenici a ve sklenici se láme na kostku ledu [30].



Obrázek 18. Ukázka vykreslení objektů s realisticky vykreslenými prvky světelného paprsku s pomocí metody Ray tracing [30]

3.3.3 Wireframe rendering

Wireframe rendering je metoda, která vykresluje pouze hrany objektů a jejich vrcholy, vzniká tedy jednoduchá kostra 3D objektu. Tato metoda se často používá jako první krok pro odhad, díky své rychlosti a efektivitě [13,29].



Obrázek 19. Ukázka vykreslení 3D objektu pomocí Wireframe rendering metody [29]

3.4 Dynamické 3D rekonstrukční metody

Dynamické 3D rekonstrukční metody slouží k rekonstrukci 3D modelu objektu nebo scény v reálném čase na základě dat získaných z 3D kamer. Zahrnují efektivní zpracování dynamických faktorů jako je například pohyb objektů a změna osvětlení. Některé z metod budou představeny níže [11].

3.4.1 Dynamická 3D rekonstrukce založená na RGB-D kameře

U metody dynamické 3D rekonstrukce založené na RGB-D kameře je vstupem informace o barvě z RGB kamery a informace o hloubce z hloubkové kamery pro každý bod ve scéně. Pro zpracování dat z těchto kamer se využívají pokročilé algoritmy, které splňují požadavky na běh v reálném čase a přesnost rekonstrukce [11,33].

3.4.2 3D Gaussian Splatting (3DGS)

Novější metoda 3DGS k reprezentaci 3D scény využívá 3D Gaussovy funkce. V okolí každého bodu jsou vytvořeny tisíce Gausiánů, které se optimalizují, aby odpovídali co nejlépe vstupním datům, následně je vykreslován model kumulací jednotlivých Gaussových rozdělání. Díky schopnosti této metody fungovat s vysokou úrovní detailů, při zachování nízkých nároků na výpočetní výkon je 3DGS významným přínosem v této oblasti [34].

3.5 Metody 3D rekonstrukce založené na strojovém učení

Metody 3D rekonstrukce založené na strojovém učení jsou pokročilé techniky pro vytváření trojrozměrných modelů z obrazových dat. V následující kapitole si některé z metod představíme.

3.5.1 Statistické učící metody

Tyto metody se v 3D rekonstrukci často používají k predikci a modelování vztahů mezi obrazovými daty a trojrozměrnou scénou ze vstupních dat, například obrázků nebo videí. Z velkého množství dat dokáží předpovídat 3D tvar objektů pomocí tréninkových modelů [11,35].

3.5.2 Deep learning metody

Metody založené na hlubokém učení, tedy deep learning, využívají hluboké neuronové sítě k získání funkcí a modelování vztahů mezi obrazovými daty a trojrozměrným prostorem.

Z obrazu jedné kamery jsme tedy schopni rekonstruovat 3D obraz díky učící sadě, kterou prošly miliardy snímků se známou informací o vzdálenostech objektů [13].

3.5.2.1 *Neural Radiance Fields (NeRF)*

Metoda NeRF, založená na deep learningu, modeluje pomocí hloubkového učení scénu z jednoho nebo více obrazů, jako spojitou 5D funkci, která mapuje 3D souřadnice polohy a 2D směr pohledu na barvu a míru absorpce světla (volume density) [11,13,36].

3.6 Hloubkové mapy (Depth maps)

Hloubkové mapy reprezentují trojrozměrnou scénu ve dvojrozměrném obraze a pro každý pixel ukládají vzdálenost od 3D kamery. Intenzita pixelu je nepřímo úměrná vzdálenosti, tedy čím blíže se objekt nachází, tím světlejší pixel je. Hloubkové mapy se využívají v mnoha aplikacích rekonstrukce 3D modelů. Hloubkové mapy jsou přímo ovlivněny světelnými podmínkami, odrazy, odlesky, šumem, atd. [38].



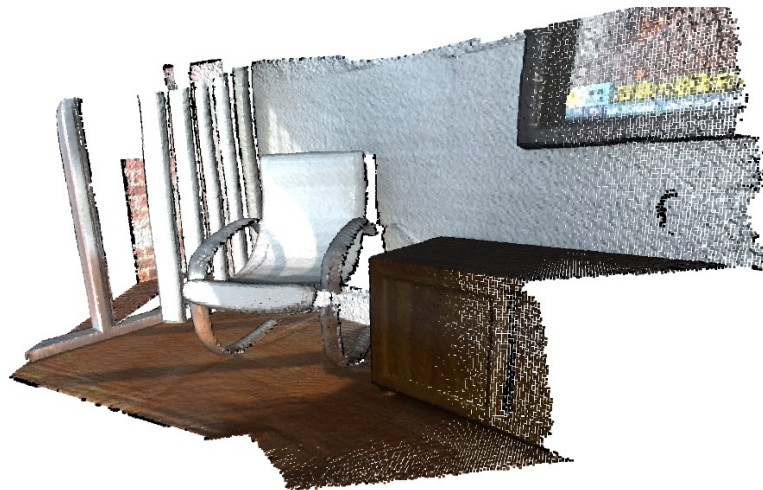
Obrázek 20. Ukázka hloubkové mapy [32]

3.7 Mračna bodů – Point cloud

Mračno bodů je datová struktura neuspořádaných bodů, které jsou reprezentovány nejenom souřadnicemi X, Y, Z, ale také dalšími vlastnostmi jako barva, intenzita světla nebo normála povrchu, viz. Obrázek 21. Tyto data jsou získávány právě s pomocí 3D kamer, které byly popsány dříve v práci [39].

Mračna bodů se využívají v mnoha oblastech jako je skenování objektů, osob, interiérů budov nebo v oblasti autonomní navigace a robotiky.

Mračna bodů obsahují velké množství informací a jsou tedy náročné na výpočetní výkon. Musíme tedy určit správný poměr mezi hustotou bodů a přesností, kterou potřebujeme. Mračna bodů se ukládají v různých datových formátech, jeden z těchto formátů může být například PCD, který si představíme níže.



Obrázek 21. Ukázka mračna bodů [39]

3.7.1 Datový formát PCD

PCD, tedy Point Cloud Data je datový formát pro ukládání mračen bodů. Soubor se skládá z hlavičky, která obsahuje metainformace o datovém souboru, jako jsou typy atributů, velikost nebo rozlišení. Dále soubor obsahuje samotné body popsány souřadnicemi X, Y, Z a dalšími atributy. PCD může být uloženo v různých formátech, jako binární, který je rychlejší pro načítání anebo ASCII formát, který je čitelnější a lépe editovatelný pro uživatele [40].

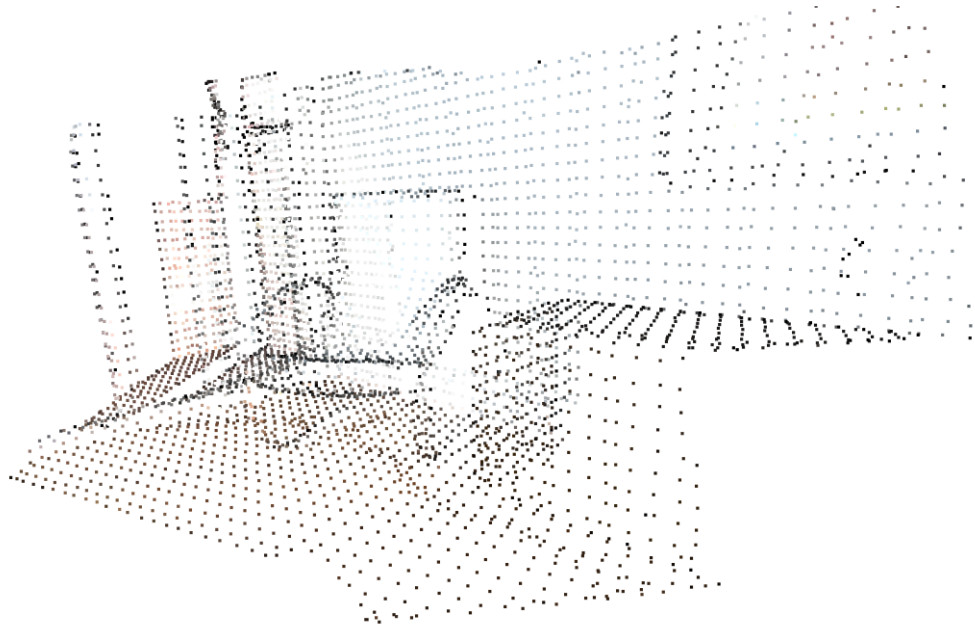
```
# .PCD v.7 - Point Cloud Data file format
VERSION .7
FIELDS x y z rgb
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 213
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 213
DATA ascii
0.93773 0.33763 0 4.2108e+06
0.90805 0.35641 0 4.2108e+06
0.81915 0.32 0 4.2108e+06
0.97192 0.278 0 4.2108e+06
0.944 0.29474 0 4.2108e+06
```

Obrázek 22. Ukázka PCD souboru [40]

3.8 Voxel Sampling

Voxel Sampling, tedy vzorkování voxelů, je metoda vzorkování pro identifikaci a zpracování mračen bodů. Metoda spočívá ve zjednodušení mračen bodů převedením bodů do takzvané mřížky voxelů. Každý voxel reprezentuje zprůměrované vlastnosti bodů v dané oblasti [39].

Metoda zajišťuje zredukování dat a snižuje tedy výpočetní náročnost zpracování velkého množství dat. Příkladem může být Obrázek 23, kde se aplikoval Voxel Sampling na originální mračen bodů z Obrázek 21 [39].



Obrázek 23. Ukázka vzorkování mračen bodů pomocí Voxel Samplingu [39]

3.9 Temporal filtr

Temporal filtr, v češtině Časový filtr, zlepšuje konzistenci hloubkových dat manipulací pixelů, na základě předchozích dat. Pokud filtr nalezne pixel, kterému chybí data, má možnost data upravit podle předchozích hodnot. Filtr se může také využít k vyhlazení šumu, který může v datech vzniknout [41].

3.10 Spatial filtr

Spatial filtr, v češtině Prostorový filtr, se zabývá upravením nebo odstraněním určitých bodů z mračna bodů, na základě jejich souřadnic nebo jiných prostorových vlastností, se zachováním hran. Pomocí tohoto filtru můžeme odstraňovat šum vyhlazováním nebo zostřit obraz zvyšováním intenzity [41].

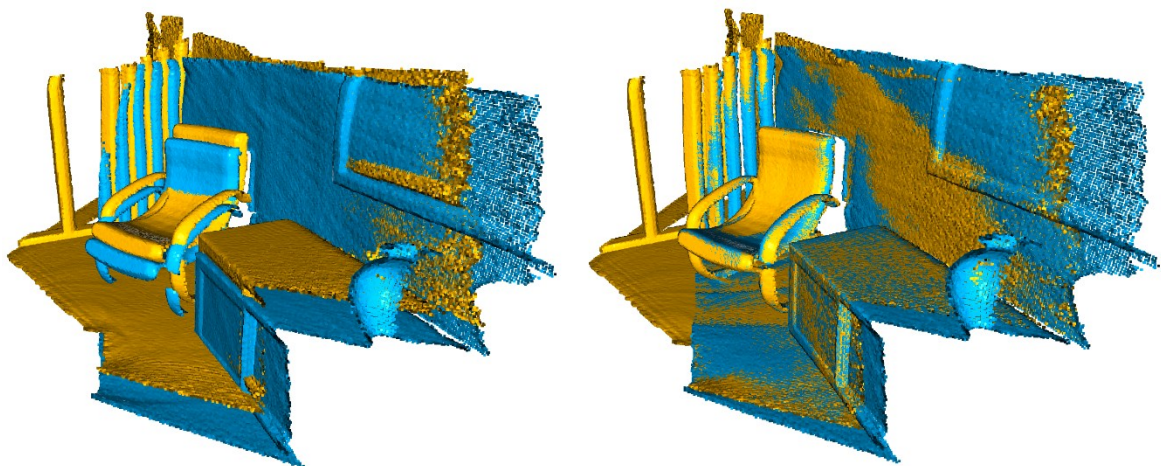
3.11 Odstranění outlierů

Při zpracování mračna bodů je jedním z kroků odstranění outlierů, tedy odlehlých bodů. Outliery mohou být způsobeny šumem, chybami senzoru kamery nebo neobvyklými změnami v prostředí. Pro odstranění těchto nežádoucích bodů, které mohou negativně ovlivnit zpracování obrazu, se používají speciální metody a filtry, jako například statistická metoda průměrování, prostorové metody klastrování K-means nebo DBSCAN, filtrování na základě modelu s využitím metody RANSAC a další různé kombinace těchto metod a filtrů [40].

3.12 ICP – Iterative Closest Point

ICP, Iterative Closest Point, je populární a velmi robustní technika v oblasti zpracování 3D obrazu, která se využívá pro zarovnání dvou a více mračen bodů, tedy k nalezení optimální transformace (rotace a translace), s minimalizací vzdálenosti mezi těmito mračny bodů.

Algoritmus odhadne počáteční transformaci mezi dvěma mračny bodů, následně pro každý bod mračna se nalezne nejbližší bod z druhého mračna. Na základě nejbližších bodů se vypočítá nová transformace a iterativně se tyto dva kroky opakují do dosažení určené minimální chyby nebo do dosažení určeného počtu iterací [39,40].



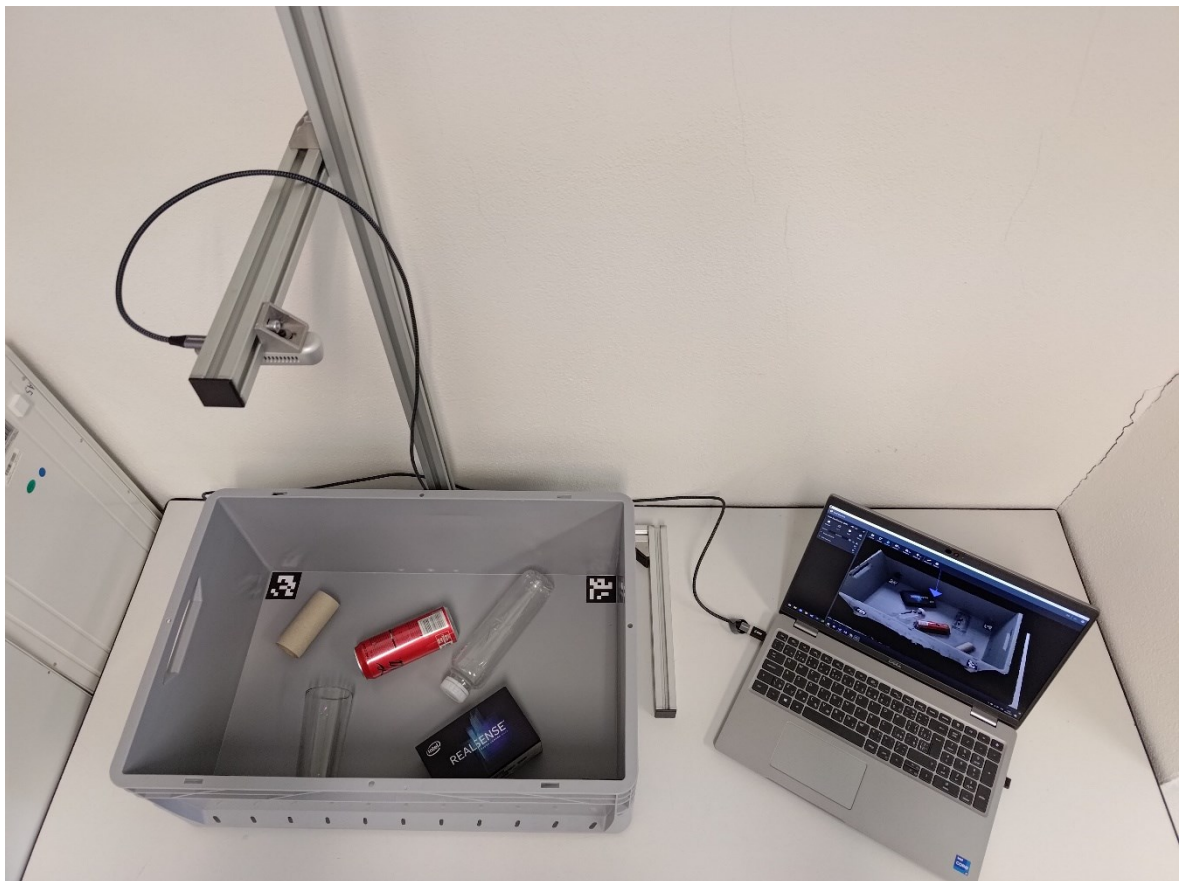
Obrázek 24. Ukázka zarovnání žlutého a modrého mračna bodů s využitím ICP [39]

II. PRAKTICKÁ ČÁST

4 POPIS ZAŘÍZENÍ

Byla vytvořena konstrukce z hliníkových profilů, na které byla umístěna 3D kamera vodorovně ke snímané ploše (viz. Obrázek 25). Do oblasti snímané plochy byla vložena plastová krabice bez víka, která reprezentuje pracovní prostor, ve kterém se budou nacházet hledané objekty. Kamera je připojena USB kabelem k notebooku, na kterém běží program.

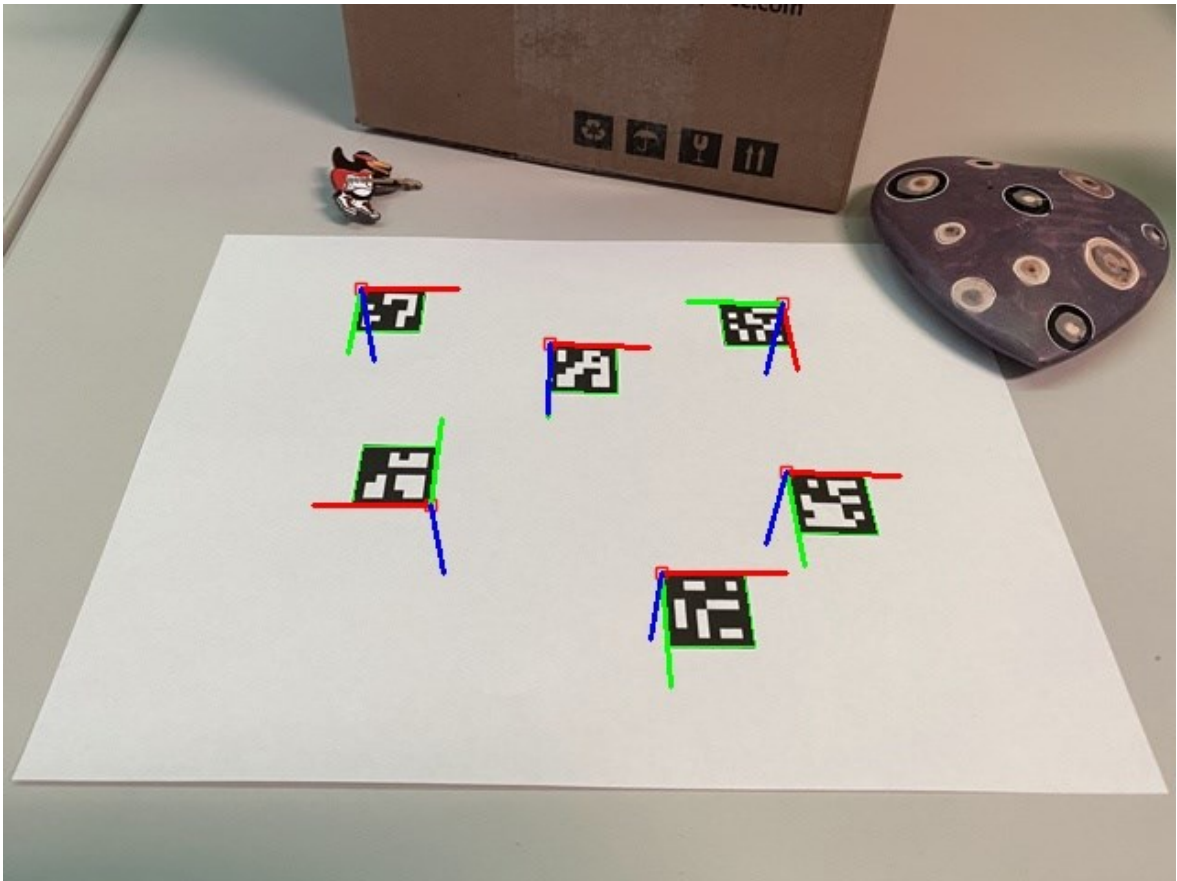
Spodní strana kamery se nachází ve výšce 69,3cm ode dna krabice, kterou snímá pod sebou. Vnitřní rozměry krabice jsou 37cm x 57cm. V rozích snímané plochy se nachází arUco značky, které využíváme k výpočtu transformační matice pro body mezi souřadným systémem kamery a souřadným systémem právě jedné z arUco značek, který jsme si zvolili jako počátek.



Obrázek 25. Ukázka fyzické konstrukce soustavy

4.1 ArUco značky

ArUco, neboli Augmented Reality University of Cordoba, jsou jedinečné, čtvercové, černo-bílé značky, které se používají v oblasti počítačového vidění k odhadu polohy a orientace, detekci objektů v prostoru, kalibraci kamer atd. Tyto značky jsou součástí knihovny OpenCV. Každá značka má svůj jedinečný identifikátor (ID) [42].



Obrázek 26. Ukázka arUco značek s osami souřadného systému [42]

V našem případě se arUco značky používají k výpočtu transformační matice ze souřadného systému 3D kamery do souřadného systému s počátkem právě v jedné z arUco značek ve scéně.

5 INTEL REALSENSE D415

V úvodní kapitole 3D kamer byla krátce popsána naše kamera používaná v praktické části této práce (viz. Obrázek 12). Nyní si kameru představíme více.

Kamera pomocí levého a pravého snímku ze dvou snímačů, vypočítá hloubku obrazu a dále s infračervenou kamerou snímá promítaný infračervený vzor, který zlepšuje přesnost měření ve scéně s nízkou hloubkou. Kamera je schopna pomocí RGB snímače zaznamenávat také barevné snímky obrazu. Je tedy možné využívat informaci o barvě daných bodů z načteného mračna [27].

Díky své kompaktnosti, s rozměry 99 mm x 20 mm x 23 mm a váhou 72 gramů, lze kameru využít ve velkém množství aplikací [27].

Možné rozlišení hloubkové kamery, které lze použít, je ovlivněno minimální vzdáleností kamery od objektů podle následující Tabulka 1. Maximální množství snímků za sekundu u hloubkové kamery, při použití HD rozlišení 1280x720px je 30fps, u nižšího rozlišení lze dosáhnout 90fps [27].

Tabulka 1. Možné rozlišení hloubkové kamery podle minimální vzdálenosti od snímaných objektů [27]

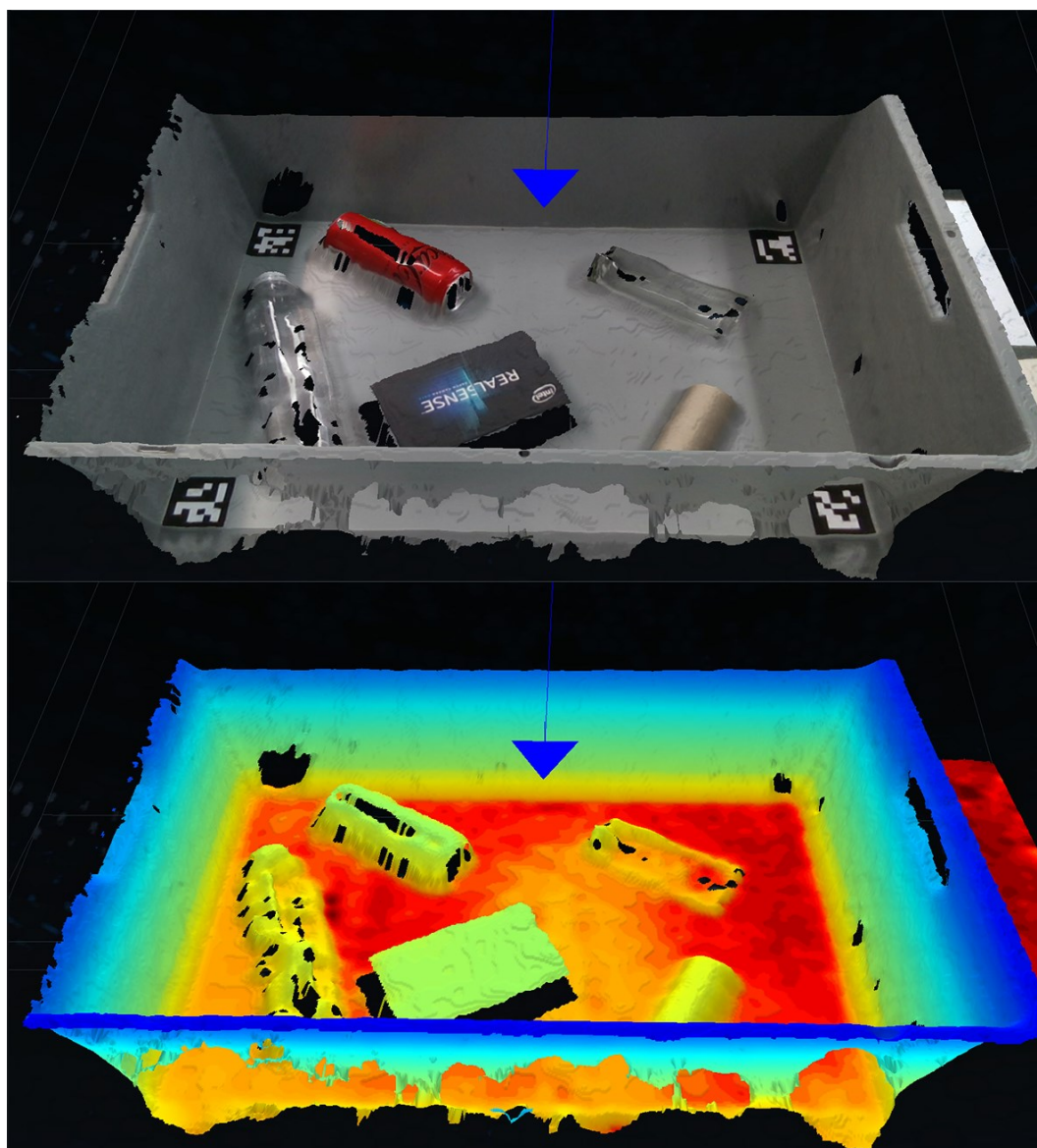
Rozlišení [px]	Minimální vzdálenost [mm]
1280x720	450
848x480	310
640x480	310
640x360	240
480x270	180
424x240	160

Pro připojení kamery k počítači a jiným zařízením, se využívá USB-C, tento konektor slouží také k napájení kamery. Pro zajištění maximální rychlosti přenosu 5Gbit/s, je nutné používat kabel podporující standard USB 3.1 Gen 1 [27].

Intel dále ke svým produktům zdarma poskytuje interaktivní nástroj Intel RealSense Viewer, který slouží k vizualizaci získaných dat z RealSense kamer. Software je součástí RealSense SDK a je podporován na operačních systémech Windows, Linux a macOS.

Prostředí poskytuje možnost nejen zobrazovat aktuální data, ale přehrávat data ze souborů, provádět kalibraci a konfiguraci kamer nebo provádět měření vzdálenosti ploch objektů.

V prostředí si lze zobrazovat data z hloubkové kamery, je možné si spustit RGB kameru a zobrazovat na hloubkových datech také barvu. U jednotlivých kamer si lze měnit různé parametry jako rozlišení, počet snímků za vteřinu, expozice, povolení různých post processing filtrů, atd.



Obrázek 27. Ukázka dat z Intel RealSense Viewer prostředí, nahoře zobrazení hloubkových dat s barvou z RGB kamery, dole pouze hloubková data

6 KNIHOVNA SDK REALSENSE – LIBREALSENSE2

C++ knihovna SDK RealSense nebo její python verze Librealsense2, kterou používáme, je nedílnou součástí této práce. Některé prvky knihovny jsou popsány v průběhu celé práce, zde si knihovnu představíme blíže.

Knihovna Librealsense2 je oficiální wrapper pro práci s Intel RealSense kamerami v pythonu. Mezi podporované modely patří například D415, D435, D435i, L515 a další. Knihovna je podporována na hlavních operačních systémech jako Windows, Linux a macOS. Mezi hlavní funkce této knihovny patří přístup k datovým proudům (streamům), jako je hloubka, barva, infračervené záření, zpracování hloubkových map nebo vizualizace dat [43].

Níže budou představeny nejdůležitější funkce, které knihovna nabízí a které se běžně používají.

- Pipeline management s využitím funkcí „rs.pipeline“ a „rs.config“, umožňuje vytváření, konfiguraci a správu datových toků ze senzorů, to může znamenat inicializaci pipeline, nastavení parametrů streamů jako rozlišení nebo framerate, spouštění nebo zastavování pipeline.
- Pomocí metody „enable_stream“ můžeme povolit konkrétní datový proud, využitím „get_data“ získáváme data z jednotlivých snímků.
- Knihovna nabízí také post-processing filtry jako „rs.spatial_filter“ a „rs.temporal_filter“, které redukuje šum a nerovnosti.
- Funkce „rs.align“, pomocí které dokážeme zarovnat různé typy snímků, například z hloubkové a barevné kamery.
- Pro kalibraci a konfiguraci senzorů podle specifických požadavků se používá funkce „rs.calibration“.
- Pokud budeme potřebovat ukládat nebo načítat data ze souborů, můžeme využít funkce „rs.record_device“ a „rs.playback_device“.

7 DALŠÍ POUŽITÉ KNIHOVNY

V této kapitole budou představeny další důležité python knihovny, které byly použity v programu diplomové práce mimo výše zmíněné knihovny librealSense2. Použité funkce z jednotlivých knihoven budou blíže popsány v kapitole s kroky programu.

7.1 OpenCV

OpenCV je velmi oblíbená open-source knihovna pro zpracování obrazu a počítačového vidění, vyvinuta společností Intel. Knihovna samotná je napsána v jazyce C++, ale má rozhraní pro mnoho dalších jazyků, včetně námi využívaného pythonu. Knihovna nabízí funkce pro zpracování obrazu (od načtení, filtrování až po různé detekce bodů příznaků), zpracování videostreamů, sledování objektů, algoritmy strojového učení, atd. [44].

7.2 Open3D

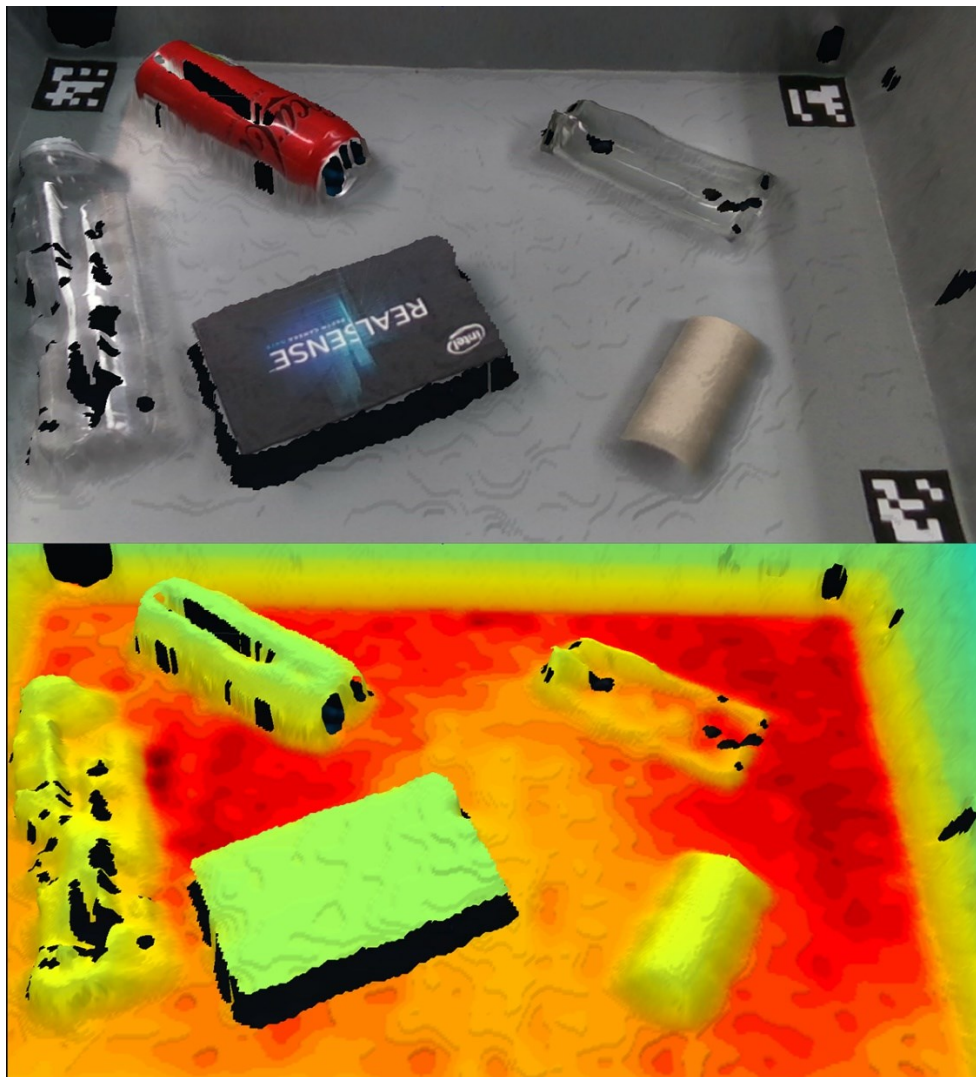
Open3D je další open-source knihovna pro práci s 3D daty, která poskytuje nástroje pro zpracování, vizualizaci a analýzu. Knihovna obsahuje funkce pro načítání 3D dat v různých formátech, nástroje na filtrování a segmentaci mračen bodů, umožňuje provádět rekonstrukce 3D modelů nebo provádět zarovnávání mračen bodů [39].

7.3 CustomTkinter

CustomTkinter je knihovna, která rozšiřuje možnosti standardní knihovny Tkinter pro tvorbu grafických uživatelských rozhraní. Tato knihovna přidává možnost jednoduše vytvářet moderní rozhraní s použitím přizpůsobených komponent. Knihovna poskytuje širokou škálu rozšíření původních widgetů, jako jsou například tlačítka se zaoblenými rohy, panely s rámečky, rozbalovací nabídky, atd. [45].

8 VÝBĚR PŘEDMĚTŮ

Před vytvořením samotného programu diplomové práce byly provedeny testy v prostředí Intel RealSense Vieweru s různými předměty, přesněji různými typy materiálů. Bylo zjištěno, že kamera má problém s určením hloubky sklenice, plastové láhve i lesklé plechovky viz. Obrázek 28, kdy se u těchto materiálu vyskytovala deformace a ztratila se velká část hloubkových dat. Pro správnou funkčnost programu a jeho odladění byla tedy vybrána papírová krabice a rulička reprezentující krychli s válcem.



Obrázek 28. Porovnání chování hloubkové kamery na jednotlivé materiály v okně Intel RealSense Viewer

9 VYGENEROVÁNÍ POINT CLOUDŮ HLEDANÝCH OBJEKTŮ VE FORMÁTU PCD

Abychom mohli provést lokalizaci předdefinovaného objektu ve scéně, musíme funkci ICP poskytnout mimo mračno bodů aktuální scény, také mračno bodů hledaného objektu.

Proto je nutné, abychom si předpřipravili jednotlivé mračna reprezentující naše hledané předměty a tyto mračna uložili jako „pcd“ soubor na disk.

V Tabulka 2 a Tabulka 3, nalezneme jednotlivé parametry, které se zadávaly do programů pro vygenerování mračen bodů.

Tabulka 2. Parametry pro generování point cloudů objektů válce

Název objektu	Poloměr [m]	Výška [m]
Plechovka	0,029	0,125
Sklenice	0,036	0,173
Rulička	0,02	0,095
PET láhev	0,0285	0,18

Tabulka 3. Parametry pro generování point cloudů objektů kvádrů

Název objektu	Výška [m]	Šířka [m]	Hloubka [m]
Krabice	0,052	0,143	0,091

9.1 Funkce válce

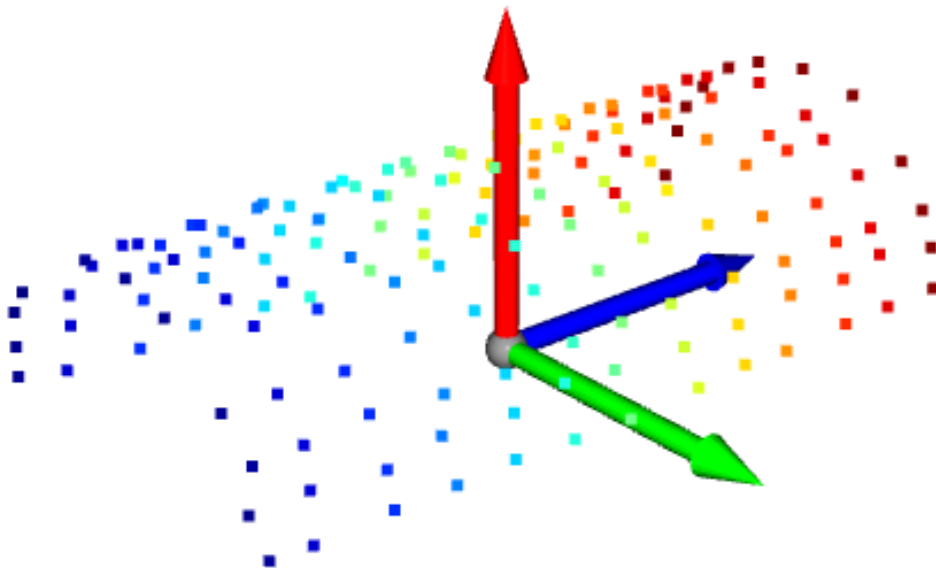
Ve for smyčce v každém cyklu vygenerujeme úhel théta mezi 0 až 180 stupni, hodnoty v ose z jsou voleny v rozsahu poloviny výšky objektu, oběma směry. Poté s pomocí goniometrických funkcí, parametry poloměru a úhlem théta vygenerujeme body v osách x a y.

$$x = \text{poloměr} \cdot \cos \theta \quad (1)$$

$$y = \text{poloměr} \cdot \sin \theta \quad (2)$$

$$z = t \quad (3)$$

Kde úhel $\theta = \langle 0; \pi \rangle$ a $t = \langle -\frac{\text{výška}}{2}; \frac{\text{výška}}{2} \rangle$



Obrázek 29. Ukázka vygenerovaného mračka bodů poloviny válce

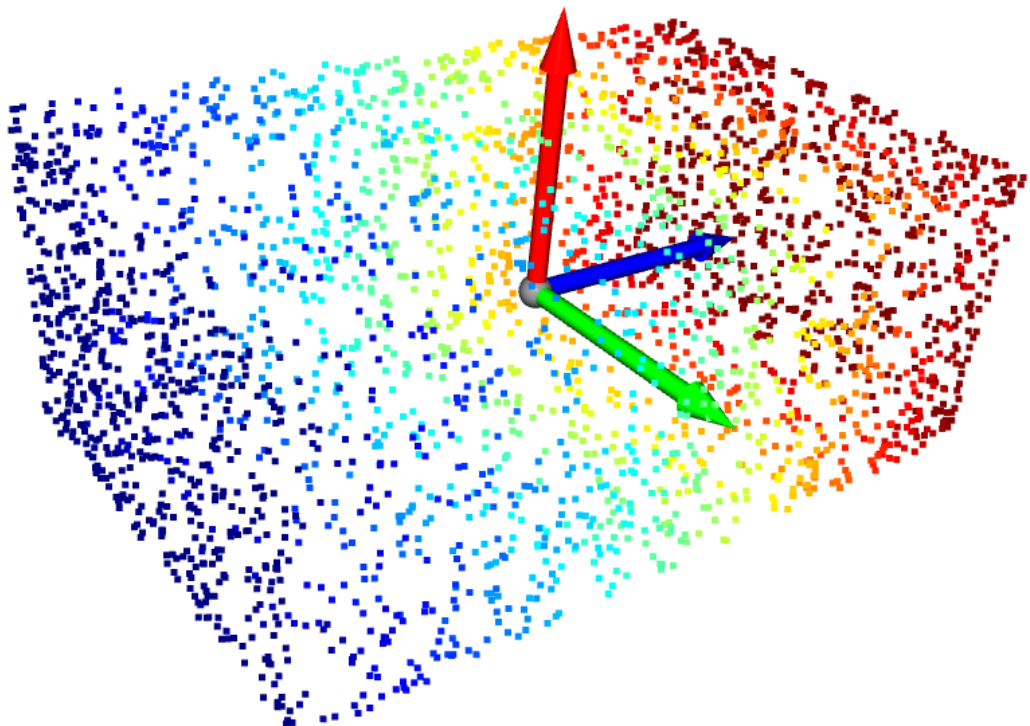
9.2 Funkce kvádru

Funkce kvádru ve for smyčce vygeneruje v daném rozsahu souřadnice x a y daného bodu, souřadnici v ose z nastaví na 0 a vytvoří jednu stranu kvádru. Poté body zkopíruje a souřadnici v ose z nastaví na hodnotu předdefinované hloubky, tímto vytvoříme druhou, vodorovnou stranu kvádru. Stejným způsobem vytvoříme zbylé 4 strany kvádru. Nakonec provedeme posunutí bodů ve všech osách o polovinu dané délky abychom docílili toho, že je těžiště mračna v počátku souřadného systému.

$$x = \langle 0; \text{šířka} \rangle \quad (4)$$

$$y = \langle 0; \text{výška} \rangle \quad (5)$$

$$z = \langle 0; \text{hloubka} \rangle \quad (6)$$



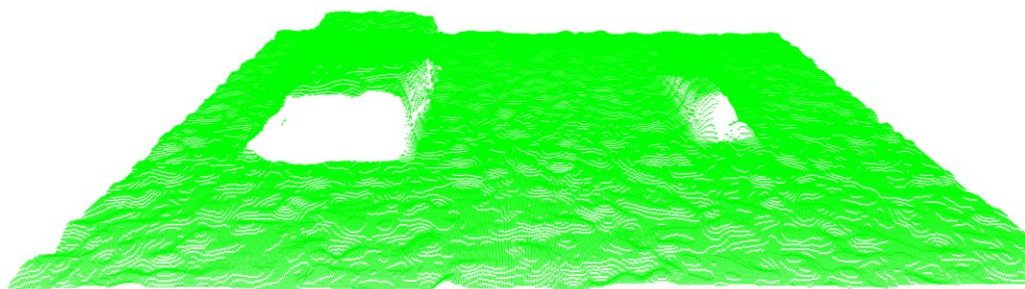
Obrázek 30. Ukázka vygenerovaného mračna bodů kvádru

10 KROKY PROGRAMU

V této kapitole si představíme nejdůležitější části programu, zaměřené na zpracování mračna bodů až do fáze nalezení transformační matice.

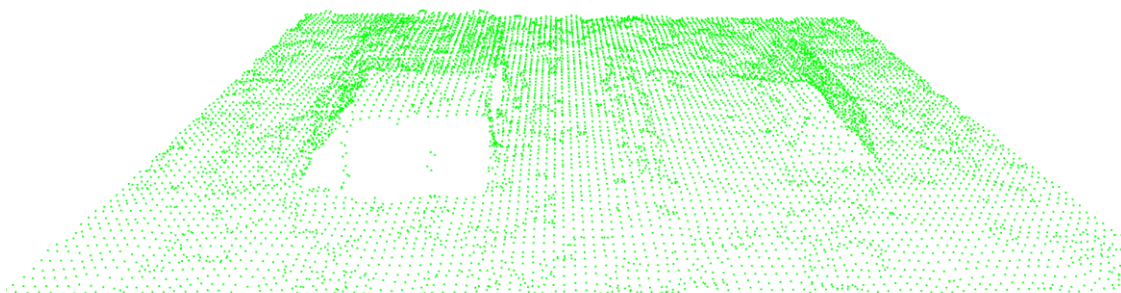
Nejprve se načtou hloubkové a barevné snímky a zarovnají se s pomocí `librealsense2` funkce „align“, která přepočítá pixelové souřadnice jednoho snímku do druhého a naopak. Poté se do proměnné „depth_intrinsics“ uloží parametry video streamu jako šířka, výška, souřadnice bodu `ppx` a `ppy`, ohnisková vzdálenost a koeficient zkreslení. Tyto parametry se dále používají v průběhu programu.

V další části se vytvoří objekt bodového mračna. Jsou načteny jednotlivé body z hloubkového obrazu, odstraní se všechny body mimo určitou vzdálenost od kamery, které nejsou v naší aplikaci použity.



Obrázek 31. Načtené mračno bodů aktuální scény

Následně je aplikována `open3D` „`voxel_down_sample`“ funkce, která zredukuje počet bodů v mračnu a tímto dokážeme snížit výpočetní výkon potřebný u následujících funkcí. Parametr „`voxel_size`“ tedy určuje, jak velká je oblast dané mřížky voxelu, tedy čím větší parametr, tím více bodů se sloučí do jednoho voxel, což vede k větší redukci bodů v mračnu.



Obrázek 32. Aplikování funkce voxel down sampling na načtené mračno

Poté je aplikována další open3D funkce „segment_plane“, pomocí které se z mračna odstraní body reprezentující podlahu, které nejsou pro naši aplikaci potřebné a toto odstranění zároveň umožní s mračnem lépe pracovat.



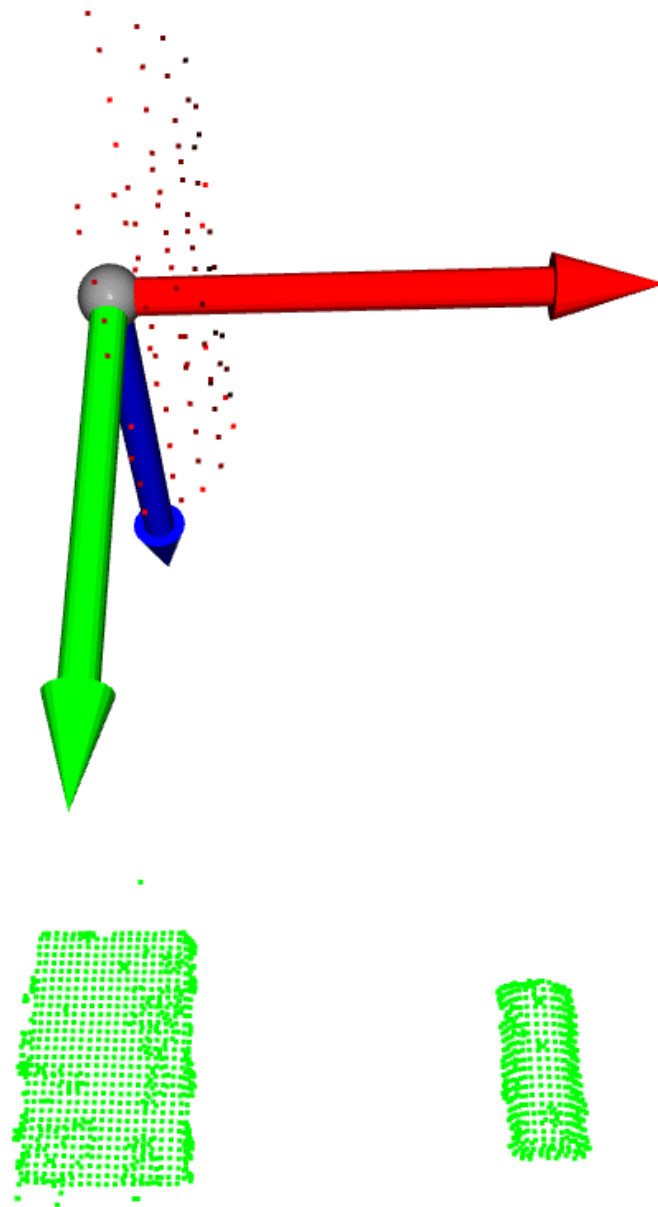
Obrázek 33. Aplikování funkce odstraňující podlahu „segment plane“ na načtené mračno

Jako poslední krok je provedena DBSCAN clustering funkce „cluster_dbscan“ a body v mračnu se rozdělí na jednotlivé shluky, ke kterým lze samostatně přistupovat.



Obrázek 34. Aplikování funkce odstraňující podlahu „segment plane“ na načtené mračno

Nyní jsou připraveny jednotlivé shluky mračen a může se začít provádět ve for smyčce hledání transformační matice mezi předdefinovaným mračnem a jednotlivými mračny ve shlucích. Na konci každého cyklu se provede kontrola parametru přesnosti transformace „fitness“ a pokud bude mít aktuální shluk lepší výsledek, tak bude aktualizována hodnota proměnné s novým výsledkem.



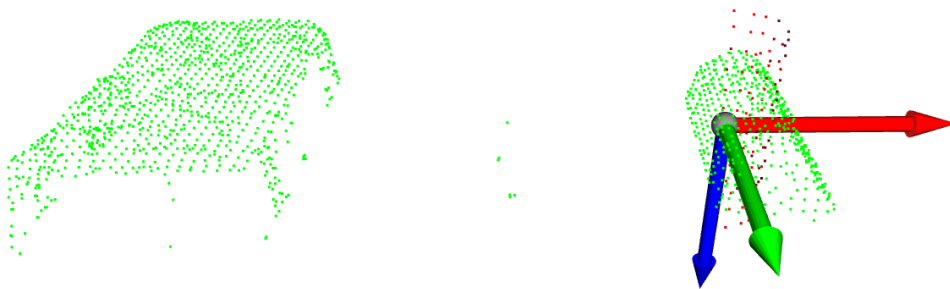
Obrázek 35. Zeleně označená mračna jednotlivých shluků ve scéně, červeně označené mračno načtené ze souboru, aktuálně v počátku souřadného systému kamery.

Hledání je rozděleno na dvě funkce, FGR – Fast Global Registration a ICP – Iterative Closest Point.

Před samotnou for smyčkou je volána vlastní funkce „compute_fpfh“, ve které jsou pro FGR vypočítány „Fast Point Feature Histograms“ rysy, které popisují informace o orientacích a vzdálenostech mezi body ve 3D prostoru a umožňují efektivní a přesnou registraci, tedy zarovnávání dvou nebo více mračen bodů. Rysy jsou pro bodové mračno vypočítány pomocí KD-stromu s poloměrem dvojnásobku voxelu. Parametr „max_nn“ nám určuje maximální počet sousedů použitých pro výpočet.

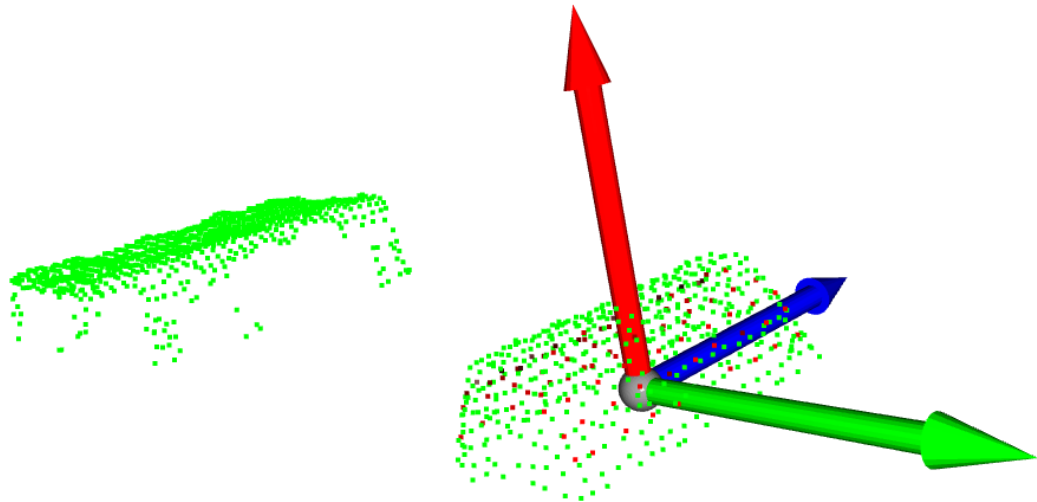
S vypočítanými rysy tedy vstupujeme do for smyčky, kde je nejprve aplikován Fast Global Registration algoritmus, který se používá k přibližnému zarovnání dvou mračen bodů, na základě jejich rysů, výsledný odhad transformace je poté použit v robustnější metodě ICP, kde jsou mračna přesněji zarovnána.

Ve for smyčce je tedy načten jeden ze shluků reprezentující jeden z předmětů ve scéně a je na něj aplikován Fast Global Registration „registration_fgr_based_on_feature_matching“ algoritmus se vstupními parametry: mračno ze scény a ze souboru, rysy mračna ze scény a ze souboru a parametr omezující vyhledávání korespondencí do vzdálenosti „distance_threshold“.



Obrázek 36. Aplikování funkce FGR, která červené mračno ze souboru přibližně proloží do daného shluku zeleného mračna bodů

Nyní se aplikuje algoritmus ICP „registration_icp“, se vstupními parametry: mračno ze scény, mračno ze souboru, tolerance vzdálenosti, při které jsou body považovány za odpovídající, odhad transformace z FGR, nastavení metody odhadu, v tomto případě minimalizace euklidovské vzdálenosti mezi body a jako poslední maximální počet iterací algoritmu.



Obrázek 37. Aplikování funkce ICP, která červené mračno ze souboru správně proloží do daného shluku zeleného mračna bodů

Po vypočítání ICP je porovnán fitness parametr aktuálního a předchozího hluku, pokud je nalezen ve scéně předmět s lepším výsledkem, hodnotu v proměnné se aktualizuje.

Do proměnné ${}^M T_C$ je uložena vypočítaná transformační matice, provedením inverze a vynásobením vektorem A, získáme vektor B, obsahující souřadnice bodu těžiště nalezeného předmětu v souřadném systému hloubkové kamery „C“.

$${}^M T_C = \begin{bmatrix} {}^M R_C & {}^M t_C \\ 0 & 1 \end{bmatrix} \quad (7)$$

$${}^M T_C^{-1} = {}^C T_M = \begin{bmatrix} {}^M R_C^T & -{}^M R_C^T \cdot t \\ 0 & 1 \end{bmatrix} \quad (8)$$

$$A = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (9)$$

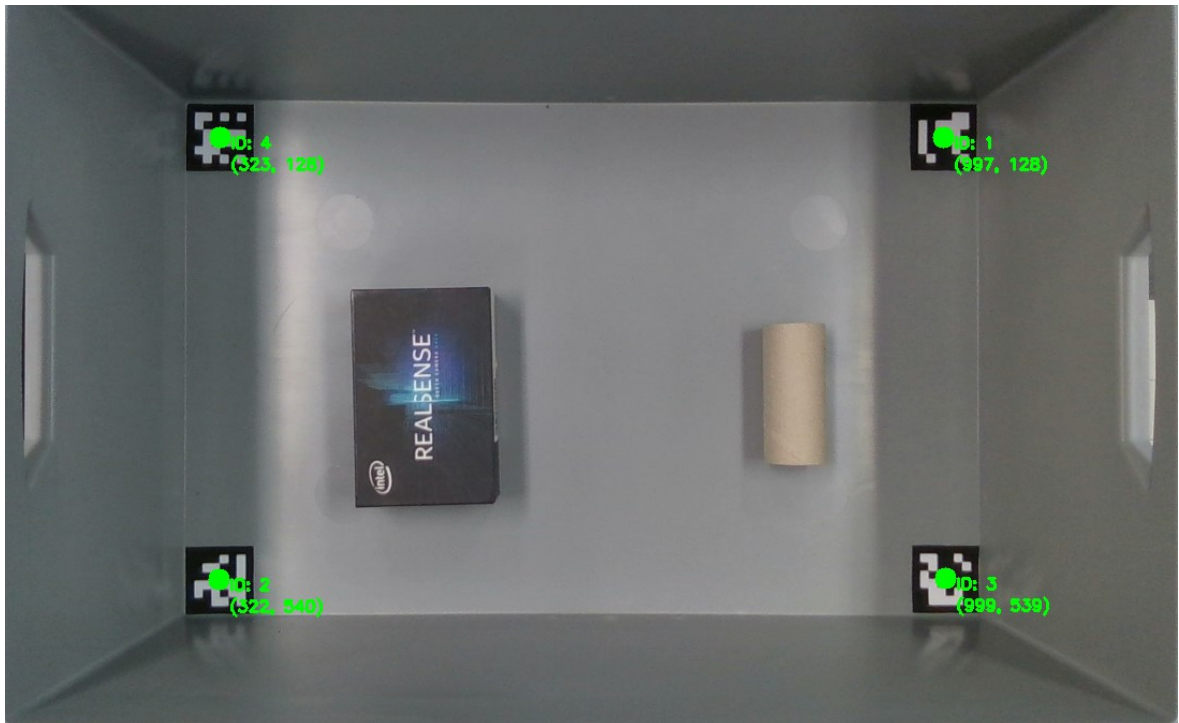
$$B = {}^M T_C^{-1} \cdot A = \begin{bmatrix} {}^M R_C^T & -{}^M R_C^T \cdot t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (10)$$

Kde, ${}^M T_C$ je transformační matice ze souřadného systému načteného mračna ze souboru (modelu) „M“ do souřadného systému kamery „C“. V této transformační matici se nachází ${}^M R_C$ rotační matice a ${}^M t_C$ je translační vektor. Dále se vypočítá inverzní ${}^M T_C^{-1}$, jinak popsána jako ${}^C T_M$. Vektor B obsahuje souřadnice bodu těžiště v souřadném systému kamery „C“.

Mezi další cíle programu patří tyto souřadnice přepočítat do souřadného systému, který bude lépe definovaný a bude možné ho použít pro navazující práce. Poté je cílem všechny důležité informace zakreslit do 2D obrazu RGB kamery, který se bude zobrazovat v okně aplikace.

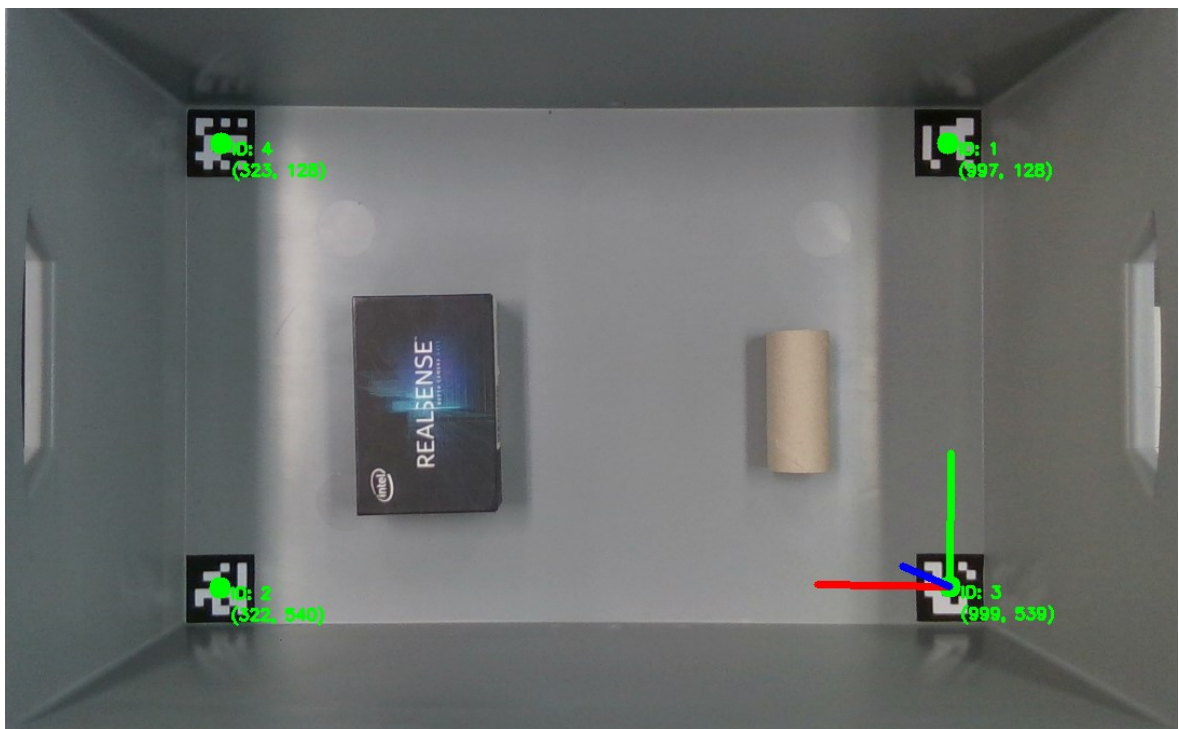
Do krabice ve scéně byly uloženy 4 arUco značky. Jedna z těchto značek, přesněji na indexu 3, bude využita jako počátek nového souřadného systému, do kterého budou souřadnice těžiště nalezeného objektu přepočítány.

Vstupní data z barevné kamery se převedou do numpy pole a toto pole je zadáno jako parametr naší funkce „detect_aruco“. Ve funkci je inicializován detektor arUco značek s použitím nadefinovaného slovníku arUco značek, v našem případě velikost 6x6 čtverců s 250 různými značkami a výchozími detekčními parametry. „detector.detectMarkers“ nalezne tyto arUco značky v obraze a vrátí rohové body a identifikační čísla nalezených značek. Nakonec je vypočítán střed každé arUco značky, uložený do seznamu „markers“ společně s ID.



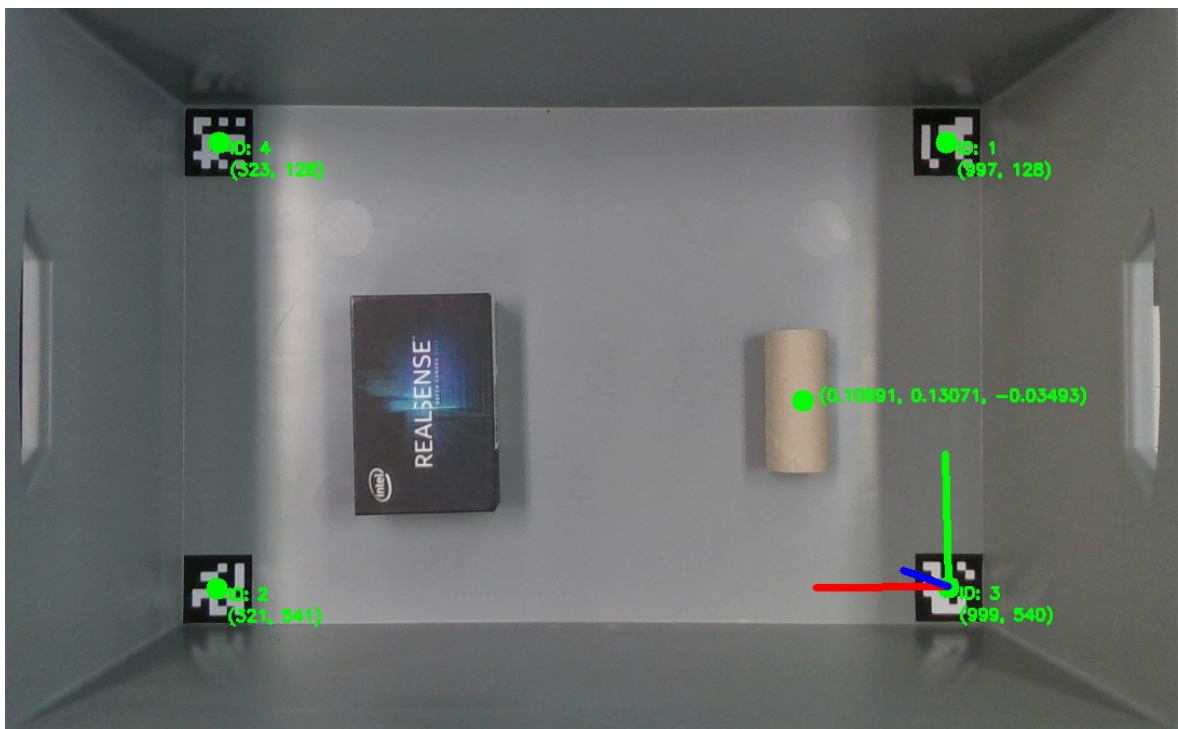
Obrázek 38. Vykreslení jednotlivých arUco značek

Pro vykreslení os souřadného systému v naší zvolené arUco značce na indexu 3 je použita vlastní funkce „draw_axes“, se vstupními parametry: barevný obraz, nalezené rohy a ID arUco značek a matice kamery obsahující právě uložené parametry kamery, které byly popsány na začátku programu. Funkce pro značku na ID 3 vypočítá odhad rotace a translace, nadefinuje si body os v 3D prostoru, pomocí funkce „cv2.projectPoints“ je provedena projekce bodů z 3D prostoru do 2D prostoru, body jsou upraveny na celočíselné hodnoty a následně se s pomocí „cv2.line“ vykreslí jednotlivé osy do obrazu RGB kamery.



Obrázek 39. Vykreslení os souřadného systému arUco značky ID 3

Nyní byly přepočítány souřadnice objektu ze souřadného systému kamery do souřadného systému arUco značky. Soustava se nastavila tak, aby osy x souřadného systému kamery a delší strany naší krabice byly k sobě rovnoběžné. Pomocí vlastní funkce „get_3d_point“, s využitím funkce „rs2_deproject_pixel_to_point“, si vypočítáme pozici středu arUco značky, reprezentované pixelem do 3D souřadnic souřadného systému kamery, následně si pomocí jednoduché transformace, kde se pouze odečtou body mezi sebou, vypočítáme souřadnice těžiště objektu v souřadném systému arUco značky. Nakonec pozici těžiště vykreslíme do obrazu jako zelený bod a vypíšeme vzdálenosti v metrech v jednotlivých osách souřadného systému arUco značky.



Obrázek 40. Vykreslení těžiště nalezeného objektu a vypísání souřadnic v souřadném systému arUco značky ID 3

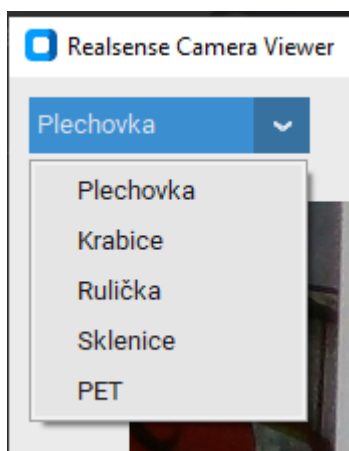
11 POPIS APLIKAČNÍHO PROSTŘEDÍ

Prostředí bylo vytvořeno s využitím knihovny CustomTkinter, které nabízí modernější vzhled oproti klasické knihovně Tkinter. Okno samotné aplikace obsahuje v hlavní části aktuální obraz z kamery, ve kterém jsou vykreslené středy nalezených arUco značek s vypsanou pozicí v pixelech. V arUco značce na indexu 3 byl vytvořen nový souřadný systém reprezentovaný vykreslenými osami. Právě do tohoto souřadného systému je přepočítávána pozice nalezeného objektu, který je označen zeleným bodem ve svém těžišti, spolu se vzdálenostmi v metrech v jednotlivých osách arUco značky ID 3.



Obrázek 41. Náhled na celé okno aplikace vytvořené pomocí knihovny CustomTkinter

V levém horním rohu viz. Obrázek 42, se nachází rozevírací seznam, ve kterém si lze zvolit předdefinované objekty, tedy uložená mračna bodů, které lze hledat ve snímané scéně.



Obrázek 42. Detail na rozevírací seznam aplikace

Ve spodní části okna aplikace nalezneme vypsané informace o pozici těžiště nalezeného objektu ve snímané scéně, v souřadných systémech kamery a arUco značky ID 3 a úhel natočení objektu kolem osy z.

souřadnice těžiště v souřadném systému kamery	X: 0.17735 m, Y: -0.04194 m, Z: 0.68134 m
souřadnice těžiště v souřadném systému arUco značky ID3	X: 0.09252 m, Y: 0.18156 m, Z: -0.01466 m
	Z-rotation: 36.58°

Obrázek 43. Náhled na vypsané informace pozice těžiště objektu ve spodní části okna aplikace

12 TESTOVÁNÍ A VALIDACE

V této kapitole bude provedeno vyhodnocování přesnosti lokalizace a vliv externích parametrů na přesnost lokalizace.

12.1 Výpočetní zařízení

Program běží na notebooku s procesem nastaveným na nejvyšší prioritu. Níže jsou vypsány jednotlivé parametry zařízení. Průměrná doba zpracování jednoho snímku v našem programu byla 0,67 sekundy.

Název: Dell Precision 3560

Operační systém: Windows 10 Pro 64-bit (build 19045)

BIOS: 1.31.0

Procesor: 11th Gen Intel® Core™ i7-1165G7 @ 2.80GHz (8 CPUs), ~1.7GHz

RAM: 16384 MB

Grafická karta: NVIDIA T500

VRAM: 1920 MB

Připojení kamery k notebooku proběhlo pomocí: PremiumCord kabel USB-C – USB 3.0 A (USB 3.2 generation 1, 3A, Gbit/s), 3m

12.2 Vyhodnocení přesnosti lokalizace

Pro vyhodnocení přesnosti lokalizace byl vytvořen test, kdy se do pracovního prostoru vkládaly na předurčené pozice předměty papírové krabice a ruličky a porovnávala se pozice těžiště mezi reálnou pozicí a vypočítanou naším programem.

Tabulky obsahují 10 měření, kdy jsou poskytnuty reálné souřadnice těžiště objektu „r“, souřadnice z programu „p“ a jako poslední sloupec je uvedena velikost chyby vypočítaná jako Euklidovská vzdálenost $d_s(r, p)$ mezi souřadnicemi daných bodů.

$$d_s(r, p) = \sqrt{(p_0 - r_0)^2 + (p_1 - r_1)^2 + (p_2 - r_2)^2} \quad (11)$$

Jako další parametr se vyhodnocovala odchylka v úhlu natočení kolem osy z. Proběhlo 10 měření pro každý předmět, 5 měření pro úhel natočení 0° a 5 měření pro úhel natočení objektu 90° . Tabulky obdobně obsahují chybu měření úhlu $d_u(r, p)$.

$$d_u(r, p) = \sqrt{(p - r)^2} \quad (12)$$

12.2.1 Pozice papírové ruličky

Tabulka 4. Porovnání reálných a vypočítaných souřadnic těžiště papírové ruličky v souřadném systému arUco značky ID 3

Č. měření	Souřadnice papírové ruličky						$d_s(r, p)$ [m]
	Reálné			Program			
	x_r [m]	y_r [m]	z_r [m]	x_p [m]	y_p [m]	z_p [m]	
1	0,041	0,141	-0,020	0,034	0,117	-0,011	0,027
2	0,101	0,160	-0,020	0,099	0,162	-0,015	0,006
3	0,074	0,220	-0,020	0,072	0,225	-0,012	0,010
4	0,178	0,284	-0,020	0,174	0,274	-0,014	0,012
5	0,173	0,024	-0,020	0,169	0,022	-0,017	0,005
6	0,242	0,137	-0,020	0,239	0,142	-0,015	0,008
7	0,276	0,316	-0,020	0,274	0,300	-0,016	0,017
8	0,355	0,214	-0,020	0,355	0,215	-0,014	0,006
9	0,386	0,051	-0,020	0,388	0,047	-0,014	0,007
10	0,475	0,152	-0,020	0,464	0,154	-0,015	0,012

Zprůměrovaná odchylka pozice pro papírovou ruličku $d_s(r, p)$ mezi 10 měřeními je

$$\overline{d_s(r, p)} = 0,011m.$$

12.2.2 Pozice papírové krabice

Tabulka 5. Porovnání reálných a vypočítaných souřadnic těžiště papírové krabice v souřadném systému arUco značky ID 3

Č. měření	Souřadnice papírové ruličky						$d_s(r,p)$ [m]
	Reálné			Program			
	x_r [m]	y_r [m]	z_r [m]	x_p [m]	y_p [m]	z_p [m]	
1	0,081	0,124	-0,026	0,082	0,122	-0,022	0,005
2	0,122	0,168	-0,026	0,124	0,168	-0,023	0,004
3	0,187	0,262	-0,026	0,187	0,264	-0,023	0,004
4	0,186	0,070	-0,026	0,186	0,060	-0,024	0,010
5	0,247	0,158	-0,026	0,252	0,160	-0,027	0,005
6	0,281	0,240	-0,026	0,284	0,243	-0,025	0,004
7	0,362	0,193	-0,026	0,365	0,196	-0,024	0,005
8	0,463	0,151	-0,026	0,466	0,150	-0,024	0,004
9	0,188	0,214	-0,026	0,187	0,212	-0,023	0,004
10	0,259	0,160	-0,026	0,261	0,159	-0,023	0,004

Zprůměrovaná odchylka pozice pro papírovou krabici $d_s(r, p)$ mezi 10 měřeními je

$$\overline{d_s(r, p)} = 0,005m.$$

12.2.3 Úhel papírové ruličky

Tabulka 6. Porovnání reálných a vypočítaných úhlů natočení těžiště papírové ruličky kolem osy z v souřadném systému arUco značky ID 3

Úhel papírové ruličky			
	Reálný	Program	
Č. měření	θ_r [°]	θ_p [°]	$d(r,p)$ [°]
1	0	1,46	1,46
2	0	1,54	1,54
3	0	1,78	1,78
4	0	0,3	0,3
5	0	1,03	1,03
6	90	90,16	0,16
7	90	89,55	0,45
8	90	90,92	0,92
9	90	89,95	0,05
10	90	90,24	0,24

Zprůměrovaná odchylka úhlu pro papírovou ruličku $d_u(r, p)$ mezi 10 měřeními je

$$\overline{d_u(r, p)} = 0,793^\circ.$$

12.2.4 Úhel papírové krabice

Tabulka 7. Porovnání reálných a vypočítaných úhlů natočení těžiště papírové krabice kolem osy z v souřadném systému arUco značky ID 3

Úhel papírové krabice			
	Reálný	Program	
Č. měření	θ [°]	θ [°]	d(r,p) [°]
1	0	1,8	1,8
2	0	2,37	2,37
3	0	2,39	2,39
4	0	1,9	1,9
5	0	0,72	0,72
6	90	88,58	1,42
7	90	91,5	1,5
8	90	89,9	0,1
9	90	89,93	0,07
10	90	90,84	0,84

Zprůměrovaná odchylka úhlu pro papírovou krabici $d_u(r, p)$ mezi 10 měřeními je

$$\overline{d_u(r, p)} = 1,311^\circ.$$

Z průměrných hodnot během měření můžeme tedy říci, že se lokalizace objektů a vypočet jejich pozice těžiště pohybuje s chybou polohy mezi 0,5 až 1,1 milimetrem a chybou úhlu natočení mezi 0,793 a 1,311 stupni.

12.3 Vyhodnocení vlivu externích parametrů na přesnost lokalizace

Jak bylo popsáno dříve, viz. Obrázek 28, proběhly testy v aplikačním okně Intel RealSense Viewer, kdy byly do zorného pole kamery vloženy předměty různého materiálu.

Matné předměty s texturou (papírová krabice a papírová rulička) byly dobře detekovány a bylo možné určit hloubku jejich bodů.

Ve chvíli, kdy se do scény vložil předmět z čirého skla nebo plastu, mračno bodů se zdeformovalo takovým způsobem, že není možné určit polohu těchto objektů. Body mračna se buďto vůbec nezobrazí nebo se zobrazí mimo objekt.

Při vložení předmětu do scény s lesklou odrazivou povrchovou úpravou, jako byla v našem případě plechovka, proběhla částečná deformace mračna bodů, ale zároveň pořád dostatečně velká na to, aby nebylo možné zaručit robustní lokalizaci předmětu.

Mimo vliv parametrů materiálu stojí za zmínku světelné podmínky. Pokud by nastala situace s nedostatečným osvětlením pracovní plochy, může se stát, že RGB kamera nedokáže ve scéně najít arUco značky a nebylo by tedy možné přepočítat souřadnice do systému arUco značky. Pozice těžiště by ale bylo pořád možné určit v souřadném systému hloubkové kamery.

ZÁVĚR

V této práci byla nejprve představena „Bin Picking“ technologie výběru náhodně rozmístěných objektů, následně byly popsány technologie a rozdělení 3D kamer a několik populárních 3D kamer bylo blíže představeno. Poté byly popsány technologie a přístupy zpracování obrazu, používané v kontextu 3D lokalizace objektů.

V praktické části bylo vytvořeno testovací pracoviště s využitím 3D kamery Intel RealSense D415, které bylo popsáno společně s použitými knihovnami v python programu. Toto pracoviště bylo použito k implementaci aplikace pro lokalizaci předdefinovaných objektů ve 3D scéně. Proběhne výběr vhodných předmětů, je popsáno, jakým způsobem byly vygenerována jednotlivá mračna bodů našich předmětů a postupně je vysvětlen kód programu zpracovávající obraz scény. Získáme transformační matici pro přepočítání ze souřadného systému namodelovaného objektu do souřadného systému hloubkové kamery. Dále si přepočítáme souřadnice těžiště nalezeného objektu do souřadného systému vytvořeném v arUco značce ID3 a veškeré důležité informace vepíšeme a vykreslíme do okenní aplikace vytvořené pomocí knihovny CustomTkinter.

V poslední kapitole diplomové práce probíhá testování a validace výsledků a je diskutována přesnost lokalizace a vliv externích parametrů na tuto přesnost. Matné materiály s texturou fungovaly při lokalizaci spolehlivě, u ostatních materiálů, jako čiré sklo s PET lahví nebo lesklá plechovka, nebyla kvalita mračna dostatečně robustní pro zaručení správné lokalizace objektů. U měření bylo zjištěno, že se určení polohy těžiště v souřadném systému arUco značky ID3 pohybuje s chybou mezi 0,5mm a 1,1mm. U chyby úhlu v rozsahu $0,793^\circ$ až $1,311^\circ$.

Diplomová práce je vypracována tak, aby bylo možné ji použít pro budoucí bakalářské a diplomové práce, které mohou využít lokalizované objekty pro další aplikace.

Jako další kroky rozšíření této práce by mohlo být propojení s robotickým zařízením, které by bylo navigováno s pomocí aplikace a předmětem zkoumání by mohly být minimální požadavky na přesnost pro navigaci robota. Další rozšíření práce by se mohlo věnovat optimalizaci algoritmů zpracování snímků například s využitím neuronových sítí.

SEZNAM POUŽITÉ LITERATURY

- [1] Optical 3D Acquisition Methods: A Comprehensive Guide [Part 2] [online]. [cit.:2024-11-4]. Dostupné z: <https://www.ml6.eu/blogpost/optical-3d-acquisition-methods-a-comprehensive-guide-part-2>
- [2] BOGUE, R. *Bin picking: a review of recent developments*. Industrial Robot, Vol. 50 No. 6, [2023]
- [3] Bin-Picking Robots Tackle Supply Chain Chaos [online]. [cit.:2024-11-4]. Dostupné z: <https://www.automate.org/blogs/bin-picking-robots-tackle-supply-chain-chaos>
- [4] Unmatched Detection Capability and Ease of Use for Robot Bin Picking [online]. [cit.:2024-11-4]. Dostupné z: https://www.keyence.com/landing/vision/pr_3d_robot_vision.jsp
- [5] Bin Picking Up To 60 Parts Per Minute [online]. [cit.:2024-11-4]. Dostupné z: <https://metrologicallyspeaking.com/bin-picking-up-to-60-parts-per-minute/>
- [6] RGB Color Model [online]. [cit.:2024-11-4]. Dostupné z: https://favpng.com/png_view/space-rgb-color-model-rgb-color-space-png/c2YBj7e1
- [7] CMY Color Space [online]. [cit.:2024-11-4]. Dostupné z: <https://docs.microsoft.com/en-us/windows/win32/wcs/cmy-and-cmyk-color-spaces>
- [8] NIRAPURE, D., UDAYPAL R. *Fast Retrieval of Images Using Filtered HSV Color Level Detection*. International Journal of Emerging Technology and Advanced Engineering, Vol. 3, No. 3, PP. 414-419, 2013.
- [9] CIE-LAB COLOR SPACE [online]. [cit.:2024-11-4]. Dostupné z: <http://she-riffblathur.blogspot.com/2013/07/cie-lab-color-space.html>
- [10] LÁNÍK, P. *Analýza vhodných metod zpracování obrazu pro řízení pohybu kolového robota*. Zlín, 2021. Bakalářská práce. Univerzita Tomáše Bati, Fakulta informačních technologií, Ústav automatizace a řídicích technologií. Vedoucí práce Ing. Jiří Zátapek.
- [11] ZHOU, L., WU, G., ZUO, Y., CHEN, X., HU, H. *A Comprehensive Review of Vision-Based 3D Reconstruction Methods*. Sensors. 2024
- [12] WILLERT, CH. *Particle Image Velocimetry A Practical Guide*. Springer. 2012

- [13] SZELISKI, R. *Computer vision: algorithms and applications*. Texts in computer science (Springer). London: Springer, 2011. ISBN 978-184-8829-343.
- [14] Structured Light 3D Scanning: What Is It and How Does It Work? [online]. [cit.:2024-11-4]. Dostupné z: <https://3dinsider.com/structured-light-3d-scanning/>
- [15] The schematic diagram of a time of flight TOF sensor [online]. [cit.:2024-11-4]. https://www.researchgate.net/figure/The-schematic-diagram-of-a-time-of-flight-TOF-sensor_fig4_335317560
- [16] Quality Basics: How Does 3D Laser Scanning Work? [online]. [cit.:2024-11-4]. <https://www.engineering.com/story/quality-basics-how-does-3d-laser-scanning-work>
- [17] Parts of a Stereo Vision System [online]. [cit.:2024-11-4]. https://www.ni.com/docs/en-US/bundle/ni-vision-concepts-help/page/stereo_parts_of_a_stereo_vision_system.html
- [18] Google's Street View cars are now giant, mobile 3D scanners [online]. [cit.:2024-15-4]. <https://arstechnica.com/gadgets/2017/09/googles-street-view-cars-are-now-giant-mobile-3d-scanners/>
- [19] Google Launches Street View in 10 Indian Cities [online]. [cit.:2024-15-4]. <https://www.geospatialworld.net/prime/technology-and-innovation/google-launches-street-view-in-10-indian-cities/>
- [20] What Is Lidar and How Does It Work? [online]. [cit.:2024-15-4]. <https://www.faro.com/en/Resource-Library/Article/What-is-Lidar>
- [21] Experimental Measurement of Underactuated Robotic Finger Configurations via RGB-D Sensor [online]. [cit.:2024-16-4]. https://www.researchgate.net/figure/The-Kinect-V2-sensor-front-with-cameras-and-emitter-positions_fig1_326398823
- [22] LIU, S., GAO, D., WANG, P., GUO, X., XU, J., LIU, D-X. A Depth-Based Weighted Point Cloud Registration for Indoor Scene. *Sensors*. 2018 [online]. [cit.:2024-16-4]. <https://www.mdpi.com/1424-8220/18/11/3608>
- [23] SR4000/SR4500 User Manual [online]. [cit.:2024-16-4]. https://acroname.com/sites/default/files/assets/sr4000_sr4500_manual.pdf

- [24] JIAO, J., YUAN, L., TANG, W., DENG, Z., WU, Q. A Post-Rectification Approach of Depth Images of Kinect v2 for 3D Reconstruction of Indoor Scenes. *ISPRS International Journal of Geo-Information*. 2017 [online]. [cit.:2024-16-4]. <https://www.mdpi.com/2220-9964/6/11/349>
- [25] Kinect for Windows [online]. [cit.:2024-16-4]. <https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows>
- [26] MESA IMAGING 3D Time-of-Flight Cameras [online]. [cit.:2024-16-4]. https://www.swissphotonics.net/libraries.files/MESA_SLN_2010.pdf
- [27] Intel RealSense Product Family D400 Series Datasheet [online]. [cit.:2024-16-4]. <https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf>
- [28] GUFFANTI, D., G., BRUNETE, A., HERNANDO, M. Kinematic Gait data using a Microsoft Kinect V2 sensor during gait sequences over a treadmill [online]. [cit.:2024-17-4]. <https://iee-dataport.org/open-access/kinematic-gait-data-using-microsoft-kinect-v2-sensor-during-gait-sequences-over>
- [29] What is wireframe modeling and why is it so crucial to design? [online]. [cit.:2024-18-4]. <https://www.3ds.com/store/cad/wireframe-modeling>
- [30] How Ray Tracing Works – Technology Explained [online]. [cit.:2024-18-4]. <https://www.theoverclocker.com/how-ray-tracing-works-technology-explained/>
- [31] Rasterization [online]. [cit.:2024-18-4]. <https://www.scratchapixel.com/lessons/3d-basic-rendering/rasterization-practical-implementation/rasterization-stage.html>
- [32] ASHBERY, S. Rasterization [online]. [cit.:2024-18-4]. <https://si-ashbery.medium.com/rasterization-13287cee9ab4>
- [33] HENRY, P., KRAININ, M., HERBST, E., REN, X., FOX, D. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments [online]. [cit.:2024-18-4]. <https://rse-lab.cs.washington.edu/postscripts/3d-mapping-iser-10-final.pdf>
- [34] 3D Gaussian Splatting for Real-Time Radiance Field Rendering [online]. [cit.:2024-18-4]. <https://ar5iv.labs.arxiv.org/html/2308.04079>

- [35] SARMAH, M., NEELIMA, A. & Singh, H.R. Survey of methods and principles in three-dimensional reconstruction from two-dimensional medical images. *Vis. Comput. Ind. Biomed. Art* 6, 15 [online]. [cit.:2024-18-4]. <https://doi.org/10.1186/s42492-023-00142-7>
- [36] Neural Radiance Field (NeRF): A Gentle Introduction [online]. [cit.:2024-2-5]. <https://datagen.tech/guides/synthetic-data/neural-radiance-field-nerf/>
- [37] LIU, S., LAI, P., TIAN, D., CHEN, C., W. "New Depth Coding Techniques With Utilization of Corresponding Video," in *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 551-561, June 2011 [online]. [cit.:2024-18-4]. https://www.researchgate.net/publication/224227340_New_Depth_Coding_Techniques_With_Utilization_of_Corresponding_Video
- [38] Depth Maps: How Software Encodes 3D Space [online]. [cit.:2024-18-4]. <https://lookingglassfactory.com/blog/depth-map>
- [39] Open3D Index of /docs/latest/tutorial [online]. [cit.:2024-28-4]. <https://www.open3d.org/docs/latest/tutorial/>
- [40] Point Cloud Library [online]. [cit.:2024-28-4]. <https://pointclouds.org/>
- [41] Intel RealSense Post-processing filters [online]. [cit.:2024-28-4]. <https://dev.intelrealsense.com/docs/post-processing-filters>
- [42] Detection of ArUco Markers [online]. [cit.:2024-19-5]. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html
- [43] Intel RealSense SDK 2.0 [online]. [cit.:2024-19-5]. <https://github.com/IntelRealSense/librealsense>
- [44] OpenCV [online]. [cit.:2024-19-5]. <https://opencv.org/>
- [45] CustomTkinter [online]. [cit.:2024-19-5]. <https://github.com/TomSchimansky/CustomTkinter>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SDK	Software Development Kit
ICP	Iterative Closest Point
2D	2 Dimensional
3D	3 Dimensional
5D	5 Dimensional
LiDAR	Light Detection And Ranging
TOF	Time Of Flight
RGB	Red Green Blue
USB	Universal Serial Bus
IR	Infrared Radiation
CMYK	Cyan Magenta Yellow Key
HSV	Hue Saturation Value
LAB	Lightness A B
ANMS	Adaptive Non-Maximal Suppresion
SURF	Speeded-Up Robust Features
SIFT	Scale-Invariant Feature Transform
RGB-D	Red Green Blue-Depth
3DGS	3 Dimensional Gaussian Splatting
NeRF	Neural Radiance Fields
PCD	Point Cloud Data
ASCII	American Standard Code for Information Interchange
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
RANSAC	RANdom Sample Consensus
arUco	Augmented Reality University of Cordoba

ID	Identification
OpenCV	Open Source Computer Vision Library
mm	Milimetr
m	Metr
px	Pixel
HD	High Definition
fps	Frames Per Second
PET	Polyethylentereftalát
FGR	Fast Global Registration
BIOS	Basic Input Output System
RAM	Random Access Memory
VRAM	Video Random Access Memory

SEZNAM OBRÁZKŮ

Obrázek 1. Ukázka Bin Picking aplikace [4].....	10
Obrázek 2. Rozdělení 3D kamer podle způsobu snímání obrazu a jednotlivých technologií [1]	12
Obrázek 3. Princip triangulace strukturovaného světla [11].....	13
Obrázek 4. Diagram principu ToF [15]	14
Obrázek 5. Diagram principu Laserového skenování [16]	15
Obrázek 6. Ukázka výstupu LiDAR skeneru [18].....	16
Obrázek 7. Ukázka Google Street View automobilu s LiDAR skenerem na střeše [19]	17
Obrázek 8. Diagram principu Stereo vidění [16].....	18
Obrázek 9. Microsoft Kinect V2 kamera [21]	19
Obrázek 10. Ukázka výstupu hloubkové mapy se skeletem osoby z kamery Microsoft Kinect V2 [28].....	20
Obrázek 11. MESA SwissRanger SR4000 [23]	20
Obrázek 12. Intel RealSense D415 [22]	21
Obrázek 13. Porovnání barevných prostorů, zleva doprava z vrchu dolů: RGB, CMYK, HSV, LAB [6,7,8,9]	22
Obrázek 14. Ukázka preprocessingu – porovnání obrázku před a po použití filtru mean-shift [10]	23
Obrázek 15. Ukázka segmentace obrazu – porovnání originálního obrázku s obrázkem, na který je aplikována metoda prahování [10].....	23
Obrázek 16. Ukázka postprocessingu – porovnání obrázku před a po použití metody morfologického otevření [10].....	24
Obrázek 17. Ukázka vykreslení objektu pomocí metody Rasterizace [32].....	25
Obrázek 18. Ukázka vykreslení objektů s realisticky vykreslenými prvky světelného paprsku s pomocí metody Ray tracing [30].....	26
Obrázek 19. Ukázka vykreslení 3D objektu pomocí Wireframe rendering metody [29]	27
Obrázek 20. Ukázka hloubkové mapy [32]	29
Obrázek 21. Ukázka mračna bodů [39]	30
Obrázek 22. Ukázka PCD souboru [40]	30
Obrázek 23. Ukázka vzorkování mračna bodů pomocí Voxel Samplingu [39]	31

Obrázek 24. Ukázka zarovnání žlutého a modrého mračna bodů s využitím ICP [39]	32
Obrázek 25. Ukázka fyzické konstrukce soustavy	34
Obrázek 26. Ukázka arUco značek s osami souřadného systému [42]	35
Obrázek 27. Ukázka dat z Intel RealSense Viewer prostředí, nahoře zobrazení hloubkových dat s barvou z RGB kamery, dole pouze hloubková data.....	37
Obrázek 28. Porovnání chování hloubkové kamery na jednotlivé materiály v okně Intel RealSense Viewer.....	40
Obrázek 29. Ukázka vygenerovaného mračna bodů poloviny válce.....	42
Obrázek 30. Ukázka vygenerovaného mračna bodů kvádru	43
Obrázek 31. Načtené mračno bodů aktuální scény	44
Obrázek 32. Aplikování funkce voxel down sampling na načtené mračno.....	44
Obrázek 33. Aplikování funkce odstraňující podlahu „segment plane“ na načtené mračno	45
Obrázek 34. Aplikování funkce odstraňující podlahu „segment plane“ na načtené mračno	45
Obrázek 35. Zeleně označená mračna jednotlivých shluků ve scéně, červeně označené mračno načtené ze souboru, aktuálně v počátku souřadného systému kamery.	46
Obrázek 36. Aplikování funkce FGR, která červené mračno ze souboru přibližně proloží do daného shluku zeleného mračna bodů.....	47
Obrázek 37. Aplikování funkce ICP, která červené mračno ze souboru správně proloží do daného shluku zeleného mračna bodů.....	48
Obrázek 38. Vykreslení jednotlivých arUco značek	50
Obrázek 39. Vykreslení os souřadného systému arUco značky ID 3	51
Obrázek 40. Vykreslení těžiště nalezeného objektu a vypsání souřadnic v souřadném systému arUco značky ID 3.....	52
Obrázek 41. Náhled na celé okno aplikace vytvořené pomocí knihovny CustomTkinter	53
Obrázek 42. Detail na rozevírací seznam aplikace	54
Obrázek 43. Náhled na vypsání informace pozice těžiště objektu ve spodní části okna aplikace.....	54

SEZNAM TABULEK

Tabulka 1. Možné rozlišení hloubkové kamery podle minimální vzdálenosti od snímaných objektů [27]	36
Tabulka 2. Parametry pro generování point cloudů objektů válce	41
Tabulka 3. Parametry pro generování point cloudů objektů kvádru.....	41
Tabulka 4. Porovnání reálných a vypočítaných souřadnic těžiště papírové ruličky v souřadném systému arUco značky ID 3.....	56
Tabulka 5. Porovnání reálných a vypočítaných souřadnic těžiště papírové krabice v souřadném systému arUco značky ID 3.....	57
Tabulka 6. Porovnání reálných a vypočítaných úhlů natočení těžiště papírové ruličky kolem osy z v souřadném systému arUco značky ID 3	58
Tabulka 7. Porovnání reálných a vypočítaných úhlů natočení těžiště papírové krabice kolem osy z v souřadném systému arUco značky ID 3	59

SEZNAM PŘÍLOH

P I Obsah CD

PŘÍLOHA P I: OBSAH CD

/text – text práce

/code – zdrojové kódy, PCD soubory s vygenerovanými mračny objektů