



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**AUTOMATICKÉ URČENÍ SOUŘADNIC ZAŘÍZENÍ
POZIČNÍHO SYSTÉMU Z PROSTOROVÝCH DAT**

AUTOMATIC DETERMINATION OF POSITIONING SYSTEM DEVICES COORDINATES FROM

POINT CLOUD DATA

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL BUCHER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK,

BRNO 2023

Zadání diplomové práce



144908

Ústav: Ústav počítačových systémů (UPSY)
Student: **Bucher Michal, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Vývoj aplikací
Název: **Automatické určení souřadnic zařízení pozičního systému z prostorových dat**
Kategorie: Počítačové vidění
Akademický rok: 2022/23

Zadání:

1. Seznamte se s principy bezdrátové komunikace dle standardu UWB a technickými aspekty lokalizační platformy Sewio pro monitorování pozice objektů ve vnitřních prostorech.
2. Prostudujte základní principy mapování prostoru pomocí LiDAR technologie, panoramatických kamer a obdobných zařízení. Dále se zabývejte strukturou výstupních dat.
3. Zabývejte se uživatelským API rozhraním lokalizační platformy Sewio. Prostudujte službu Matterport nebo její ekvivalent včetně dostupných prostředků pro podporu vývoje.
4. S využitím vhodného přístrojového vybavení proveďte zmapování prostorů s instalovanými prvky lokalizačního systému Sewio a získajte výstupní data prostoru.
5. Navrhněte algoritmus pro detekci specifikovaných prvků lokalizačního systému Sewio Point nad daty získanými v bodě 4.
6. Implementujte navržené řešení z bodu 5. Následně pečlivě otestujte jeho funkčnost s daty získanými v bodě 4.
7. Zhodnoťte dosažené výsledky a diskutujte možnosti pokračování projektu.

Literatura:

- Ian Oppermann, UWB: Theory and Applications Hardcover, Wiley-Blackwell, 2004, ISBN-13: 978-0470869178
- Steven W. Smith, The Scientist & Engineer's Guide to Digital Signal Processing, California Technical Pub, 1997, ISBN 0966017633

Při obhajobě semestrální části projektu je požadováno:
Splnění bodů 1 až 4 zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Šimek Václav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1.11.2022
Termín pro odevzdání: 17.5.2023
Datum schválení: 31.10.2022

Abstrakt

Diplomová práce se zabývá detekcí kotev lokalizačního systému v prostoru a určením jejich pozice. Pro zajištění správného fungování RTLS systému firmy Sewio Networks s. r. o. je nutné znát umístění kotev v prostorech instalace. Kotvy zachytávají informace o pozici tagů v prostoru. V textu práce jsou popsány technologie využívané pro detekci objektu a pro záznam 3D dat prostoru. Velká část textu je věnována vyhodnocení již existujících řešení. Na základě vyhodnocení byl vybrán způsob, který byl součástí této práce implementován. Výsledná data získaná za pomoci implementace jsou porovnaná s již změřenými pozicemi kotev ze systému Sensmap Visualization firmy Sewio Networks s. r. o..

Abstract

Thesis describes the problem and solution of detecting anchors of localization systems and evaluating anchor position in 3D space. Position system from Sewio Networks s. r. o. requires position in 3D space for individual anchors. These anchors then gather information about the position of individual tags in their range. The work describes some commonly used technologies for object detection and 3D data reconstruction. Compares the existing solutions in context of this work. Based on the results chooses one approach and implements it. Final gathered data with implementation are compared with real existing measurements used in the RTLS Sewio Networks s. r. o. system.

Klíčová slova

LIDAR, RGBd, Kamera, UWB, Point cloud, Počítačové vidění, Detekce objektů, Localization

Keywords

LIDAR, RGBd, Camera, UWB, Point cloud, Computer vision, Object detection, Localization

Citace

BUCHER, Michal. *Automatické určení souřadnic zařízení pozičního systému z prostorových dat*. Brno, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek,

Automatické určení souřadnic zařízení pozičního systému z prostorových dat

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Václava Šimeka. Další informace mi poskytla firma Sewio Networks, s. r. o.. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Michal Bucher
15. května 2023

Poděkování

Děkuji všem, kteří mi poskytli možnost se podílet a nahlédnout do problematiky pozičních systémů. Hlavně tedy firmě Sewio Networks s. r. o. za poskytnutí zajímavého zadání a velmi příjemné interakce v průběhu práce. Také bych chtěl poděkovat své rodině a kamarádům za motivace v průběhu práce.

Obsah

1	Úvod	5
2	Rozbor tématu	7
2.1	UWB	7
2.2	RTLS lokalizační systém	8
2.3	UWB v RTLS	8
2.3.1	Metody ToF a TDoA	8
2.4	Sewio Networks s. r. o. řešení	9
2.4.1	Instalace zařízení	9
2.4.2	Software	11
2.5	Point cloud	11
2.5.1	Způsob uložení dat	12
2.6	Lidar	13
2.6.1	Princip	13
2.7	RGB-d kamery	15
2.7.1	Stereoskopické RGB-d kamery	15
2.7.2	RGB-d kamery s senzorem	16
2.7.3	SLAM	16
2.8	Radar	17
2.9	QR kod	17
2.9.1	Princip ukládání dat	18
2.9.2	Lokalizace a čtení	18
3	Zhodnocení možností	19
3.1	Mobilní telefon	19
3.2	Matterport	19
3.3	Mobilní telefon s lidarem	22
3.3.1	3D scanner app	22
3.3.2	Polycam	22
3.4	ARkit	22
3.5	Metal API	23
3.6	RGB-d kamery	23
3.7	Knihovna OpenCV	24
3.8	Shrnutí	24
4	Návrh řešení	25
4.1	Obecný návrh	25
4.2	Mapování prostoru	26

4.2.1	Konverze 2D bodu do 3D prostoru	26
4.3	Detekce objektu	29
5	Implementace	31
5.1	Zprostředkování ARKit	31
5.2	Zpracování obrazu	31
5.2.1	Detekce QR kódů	32
5.2.2	Vytvoření point cloud	33
5.2.3	Komunikace mezi CPU a GPU	34
5.2.4	Zpracování dat	35
5.2.5	Obarvení bodů	36
5.2.6	Vykreslení dat na displej	36
5.3	Filtr dat	36
5.4	Export dat	37
6	Experimenty a testování	38
6.1	Chování detekce	38
6.1.1	Velikost QR kódu	38
6.2	Kvalita výsledných dat	39
6.2.1	Mapování prostor	39
6.2.2	Mapování větších prostor	41
6.3	Intemac	43
6.3.1	Způsob připevnění kódů	44
6.3.2	Způsob zvolení referenčního bodu	45
6.3.3	Prostory „H“	45
6.3.4	Skupina „2-M“	48
6.4	Kombinace obou skupin	50
6.5	Zhodnocení provedení	54
6.6	Následující práce	54
7	Závěr	56
	Literatura	58

Seznam obrázků

2.1	Schéma metody TDoA a ToF	8
2.2	Detekční senzory firmy Sewio [30]	9
2.3	Dodržení čtvercové formace senzorů [29]	10
2.4	Pravidla střídavé instalace [29]	10
2.5	Instalace v Vista DirectFive	11
2.6	Ideální nastavení Vista DirectFive	11
2.7	Vykreslená point cloud data	12
2.8	Princip diskrétního a kontinuálního lidar senzoru	14
2.9	Samplování 3D prostoru pomocí lidar zařízení	14
2.10	Vliv vlastností materiálů na získávání dat	15
2.11	Princip výpočtu hloubky za pomoci stereoskopie [12]	16
2.12	Neúplnost objektu při získávání dat lidar senzorem.	17
2.13	QR kód s uloženým textem „DP Michal Bucher“	18
2.14	Bar kód s uloženým textem „DP Michal Bucher“	18
3.1	Automaticky vytvořený 2D plán v aplikaci	20
3.2	Část chybně vygenerovaného modelu dveří tělocvičny	21
3.3	Část 3D modelu, kde došlo k výrazné deformaci prostředí.	21
3.4	Vzájemná komunikace CPU a GPU přes paměť. Zdroj [20]	23
4.1	Grafické znázornění algoritmu.	25
4.2	Projekce normalizovaných 2D bodu do lokálního 3D prostoru kamery. [25]	26
4.3	Konverze lokálního bodu kamery do 3D virtuálního světa	28
4.4	Překryv fotky s daty jistoty ze zařízení iPhone 12 Pro.	28
4.5	Kvalita dat v kontextu záznamu objektu s nízkou úrovní jistoty na hranách	29
5.1	Problém s ohraničením za pomoci nejmenšího obdélníku při detekci.	32
5.2	Vytvořená mřížka bodů na obrazovce	33
5.3	Barevně vykreslená data „confidenceMap“.	34
5.4	Barevně vykreslená data „depthMap“.	34
5.5	Způsob ukládání bodů detekce.	35
5.6	Způsob eliminace 3D bodů za pomoci box grafů.	37
6.1	Vliv snímkovací frekvence na kvalitu dat	39
6.2	Špatný rozptyl bodů QR kódu.	40
6.3	Dobry rozptyl bodů QR kódu	40
6.4	Referenční obrázek velikosti záznamu.	41
6.5	Velikost naakumulované chyby.	41
6.6	Záznam za pomoci stabilizace zařízení	42

6.7	Záznam za pomoci stabilizace zařízení.	43
6.8	Plán podlaží firmy Intemac v Kuřimi s instalací RTLS firmy Sewio Networks s. r. o.	43
6.9	Startovní pozice QR kódu pro „H“	45
6.10	Startovní pozice QR kódu pro „2-M“	45
6.11	Naznačení pohybu pro záznamy skupiny „H“.	46
6.12	Naznačení pohybu pro záznamy skupiny „2-M“.	48
6.13	Import dat z tabulky 6.9 do Sensmap Visualization systému	49
6.14	Point cloud záznamu s detekovanými kotvami	52
6.15	Import dat z tabulky 6.13 do Sensmap Visualization systému	53

Kapitola 1

Úvod

Lokalizace v prostoru je v dnešní době poměrně vyvíjená oblast technologií. V dnešní době existuje hned několik alternativ řešení lokalizace. Nejznámější technologií je GPS, která se zaměřuje převážně na lokalizaci ve venkovním prostředí. U vnitřních prostor se setkáváme s větším počtem nejrůznějších technologií. Znamější WIFI, nebo bluetooth technologie, které se využívají převážně hlavně v domácnostech. V industriálním prostředí je nejvíce prominentní alternativou UWB technologie, která přináší hned několik benefitů. Lepší dosah a poměrně široké pásmo vysílání. Díky těmto technologiím lze optimalizovat například pohyb osob nebo vozidel v prostorách. Kromě zvýšení efektivity na pracovišti se také zvyšuje bezpečnost.

Důležitým krokem pro správné fungování UWB lokalizačních systémů je instalace a nastavení systému, díky kterému se spojují detekční senzory do skupin. Práce se zaměřuje na instalační proces firmy Sewio Networks s. r. o.. Po připevnění detekčních senzorů se vynášejí jejich pozice do 2D plánu místnosti na daném podlaží. Pozice jsou získány z komplikovaného měření, které je zdlouhavé a v některých případech složité. kvůli tomu vznikly požadavky na průzkum alternativních možností, jak by bylo možné měření pozic senzorů ulehčit a umožnit tak zákazníkovi možnost vlastního měření.

Část práce se zabývá možnostmi již existujících řešení záznamu a zpracování 3D prostoru, které jsou v dnešní době dostupné. Účelem této práce nebylo využití profesionálních zařízení, ale zaměřit se na dostupnější možnosti. V rámci práce se objevují převážně dvě zařízení. Mobilní telefon nebo kamery, které díky speciálním sensorům umožňují zaznamenávat a ukládat informace o pohybu v prostoru a zároveň mapovat jednotlivé objekty do virtuálního 3D prostoru. Účel práce je zaměřený na technologie speciálních kamer Intel RealSense, který využívá RGB-d modelu záznamu a nebo Apple mobilních telefonů, které mají lidar sensor.

První kapitola pojednává o fungování jednotlivých technologií a ukazuje, jakým způsobem poskytují svá data. Součástí této kapitoly je také seznámení s jednotlivými pojmy a teoriemi, které jsou v práci dále využívány. Navazující část práce demonstruje problémy již existujících řešení pro zjištění pozic detekčních senzorů. Podstatná část obsahu kapitoly o srovnání technologií se věnuje aplikacím, které umožňují využít klasických fotoaparátů, nebo fotoaparátu v mobilním telefonu k získání 3D modelu na základě fotogrammetrické reprodukce v kombinaci s umělou inteligencí. Zbývá část pojednává o aplikaci, která byla základem myšlenky využít chytrého mobilního telefonu pro určení pozic detekčních senzorů.

Na základě zhodnocení již existujících řešení byl vytvořený návrh algoritmu. Kapitola popisuje obecný návrh, který lze aplikovat na obě zařízení. Uvádí základní myšlenky procesu zpracování vstupních dat a rozdělení algoritmu na části.

Po zhodnocení možností společně se zpětnou vazbou firmy bylo vybráno referenční zařízení pro implementaci. Tato kapitola popisuje, jak jsou jednotlivé části navrženého algoritmu implementované.

Poslední kapitola práce se zabývá testováním a vyhodnocením výsledků. Část kapitoly se soustředí na vypořádané chování zařízení při záznamu. V navazující části popisuje demonstraci řešení v prostorách firmy Intemac Kuřim, kde má firma Sewio Networks s. r. o. funkční instalaci jejich lokalizačního systému.

Kapitola 2

Rozbor tématu

V následujících několika sekcích jsou postupně popsány technologie, na které se navazující části práce odkazují a zmiňují je. První sekce jsou obecně zaměřené na princip a fungování UWB technologií, na které navazuje popis řešení firmy Sewio Networks s. r. o.. Následujících několik sekcí se pak zaměřuje na způsob reprezentace a způsob získání 3D dat. Kapitola také zahrnuje detailnější popis, jak fungují senzory, které zařízení k záznamu dat využívají.

2.1 UWB

UWB nebo-li Ultra-Wideband je bezdrátová technologie pro přenos dat krátké vzdálenosti, která využívá velkou část rádiového spektra pro širokopásmovou komunikaci. Výhodou je nízká spotřeba energie potřebná k vysílání signálů. Využívá krátké pulzy frekvencí od 3,1 až 10,5 GHz. Při využití celého 7,5 GHz pásma je maximální spotřeba zhruba 0.5 mW. Tyto hodnoty jsou tak podstatně nižší na rozdíl od 2.45 mW u industriálního standardu jako je IEEE 802.11 a/b/g. Rychlost přenosu dat ovlivňuje vzdálenost vysílače a přijímače. Na vzdálenosti jeden až deset metrů umožňuje UWB přenášet data rychlostí až stovky Mbps. Ačkoliv dosah může být až 50 metrů, podstatně zde klesá přenosová rychlost v porovnání s klasickými WLAN systémy. UWB k přenosu dat využívá velmi krátké pulzy v rámci pikosekund, které zároveň zahrnují všechny frekvence. Díky tomuto chování se UWB signál popisuje jako bezpečný. Při detekování signálů se na první pohled jeví jako náhodné rušení. UWB technologie se převážně využívá pro lokalizační a sledovací systémy nebo ke čtení dat ze senzorů. V dnešní době se zároveň využívá pro rychlý přenos dat v mobilních telefonech na krátkou vzdálenost. Technologie je podobná Bluetooth, jedná se ale o rychlejší, bezpečnější a energicky úspornější alternativu. V industriálním prostředí UWB technologie umožňuje využití velmi rozsáhlých instalací bez narušení stability a přenosů ostatní infrastruktury. [26]

Hlavní výhody UWB systémů

- Levná a jednoduchá technologie.
- Signály se při pokusech o detekci chovají jako náhodný šum.
- Obtížné rušení signálů.
- Rychlá odezva umožňující poziční systémy.

2.2 RTLS lokalizační systém

Lokalizace za pomoci UWB komunikace spadá do skupiny systémů RTLS, nebo-li „Real-time location system“. Systémy díky rychlé komunikaci poskytují data pozice sledovaných objektů v reálném čase.

2.3 UWB v RTLS

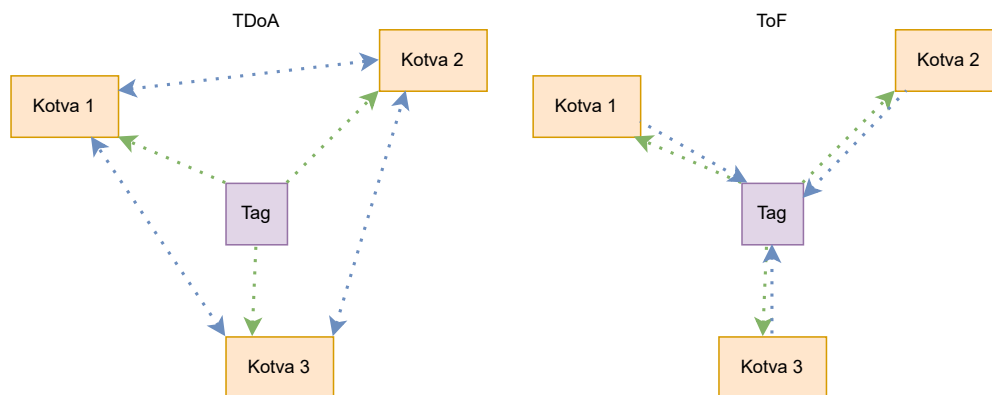
Předchozí sekce popisuje co UWB znamená a jaké je jeho obecné využití a princip. RTLS Poziční systémy využívají převážně UWB komunikace. Tomu napomáhají vhodné vlastnosti UWB díky možnosti koexistence UWB s ostatními systémy a rychlosti přenosu dat na vzdálenosti do 50 metrů. S lokalizací založené na UWB komunikacích se lze setkat na místech, kde jsou součástí instalace i další systémy citlivé na rušení externími vlivy. Taková prostředí jsou například nemocnice, výrobní haly a sklady. K fungování pozičního systému je zapotřebí dvou typů zařízení „tagu“ a „detekčního senzoru“, nebo-li kotvy. V principu tag funguje jako vysílač signálu a je připevněn na sledovaný objekt. Kotva slouží jako přijímač a je upevněná na pevné pozici. Její úkolem je přijmout UWB zprávu a přeposlat ji serveru, který vypočte pozici tagu v prostoru. [19]

Rychlý přenos velkého množství dat na kratší vzdálenosti umožňuje lokalizace sledovaný objektu v reálném čase, kde s velmi nízkou odezvou je systém schopný určit pozici. K tomu systémy využívají dvou rozdílných metod. [19]

2.3.1 Metody ToF a TDoA

Jedná se o metody určení polohy v RTLS systémech. Metoda „Time of Flight“ (ToF) vypočítává polohu zařízení podle času, který uběhl od vyslání signálu až po jeho zachycení. Druhá metoda se nazývá „Time difference of arrival“ (TDoA) a využívá rozdílného času doletu s využitím většího množství synchronizovaných přijímacích kotev. Poloha je určena v momentě vyslání signálu a zachycení signálu minimálně třech kotvách. [33]

Metoda TDoA klade vyšší nárok na hardware z důvodu neustálé synchronizace kotev mezi sebou vzhledem k pasivnímu chování tagu. Tag v TDoA metodě pouze vysílá signál. TDoA oproti ToF metodě umožňuje sledovat podstatně vyšší počet zařízení. U metody ToA musí být sledovaná trasa tam i zpátky. U ToF metody musí být tag schopný i přijímat signály kotev. Důsledkem toho je větší spotřeba baterie v tagu. [33]



Obrázek 2.1: Schéma metody TDoA a ToF

2.4 Sewio Networks s. r. o. řešení

Sewio Networks s. r. o. se zabývá problematikou sledováním zařízení a personálu v budově. Zaměřuje se na optimalizaci a zefektivnění pohybu po skladu. Řešení firmy představuje RTLS systém, který využívá UWB komunikace. Poskytuje software a hardware, který slouží k sledování jednotlivých osob, předmětů a vozidel po vnitřních prostorách. Zároveň slouží jako bezpečnostní systém. [32]

K detekci využívá v základu dvou typů zařízení kotev a tagů. Zařízení jsou určena do industriálního prostředí s ochranou IP65 proti prachu a vlhkosti. Systém je postaven na RTLS metodě ToA. Přesnost detekce pozice sledovaného objektu je na 30 centimetrů. [32]



Obrázek 2.2: Detekční senzory firmy Sewio [30]

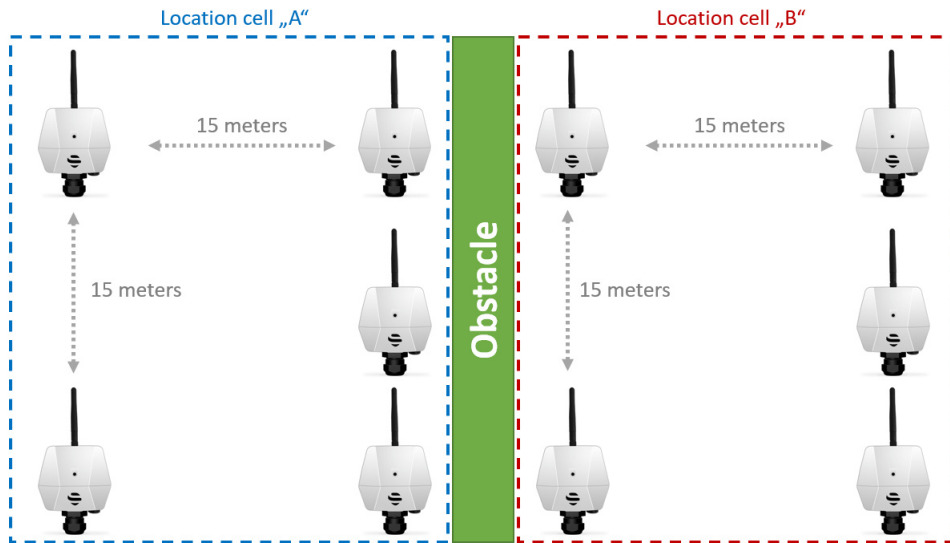
Poskytovaný software je rozdělený do pěti skupin na správu, plánování, sledování, přehrávání a analýzu. Část, která nejvíce souvisí s touto prací, je plánování a následné sledování. Vstupní data těchto systémů jsou 3D, nebo 2D plány vnitřních prostor, ve kterých je možné zasadit zařízení a kontrolovat jejich základní chování, například pokrytí signálem. [32]

2.4.1 Instalace zařízení

Firma Sewio Networks s. r. o. poskytuje dvě alternativy kotev. Každý typ má vlastní pravidla, které je potřebné pro správné chování systému dodržet. Kotvy na rozdíl od tagů vyžadují neustálý přístup k energii. Pozice zařízení se po zavedení do systému nemění. [29]

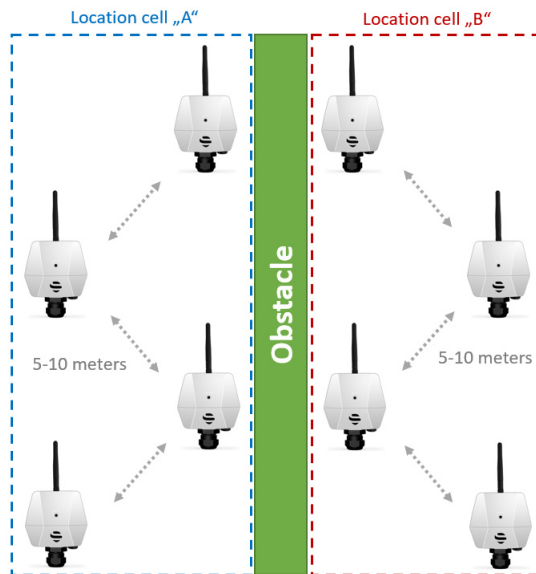
Vista Omni

Pro kotvy s všesměrovou anténou je důležité, aby anténa směřovala přímo vzhůru, nebo v ojedinělých případech lze směřovat anténu směrem dolů. To z důvodu, že i samotné tělo kotvy slouží jako anténa. Jakékoliv natočení, či ohyb antény v jakémkoliv směru, má negativní vliv na zachycená data. Zároveň, aby kotva fungovala spolehlivě, je důležité nepřesáhnout výšku instalace šesti metrů od podlahy. Pro lepší funkčnost systému se doporučuje instalovat zařízení do jedné výšky s minimálním rozdílem. Rozdíl výšky ale nesmí přesáhnout 100 centimetrů. Kotva musí být vždy připevněná na nosič, který se k zařízení dodává. Nesmí být přímo připevněná ke zdi, nebo stropu. [29]



Obrázek 2.3: Dodržení čtvercové formace senzorů [29]

Při vytváření skupin zařízení musí být z hlavního senzoru vždy přímý dohled na ostatní senzory v jedné skupině. Nesmí být narušena zdi. Vzájemná pozice zařízení zásadně ovlivňuje chování a přesnost systému. Doporučené je dodržovat čtvercové formace zařízení, které demonstruje obrázek 2.3. [29]

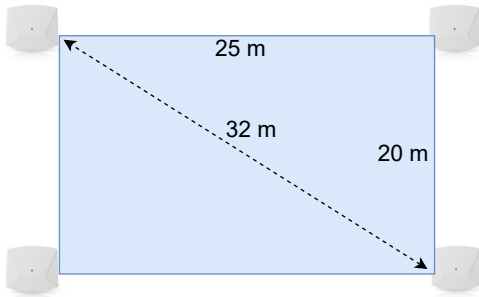


Obrázek 2.4: Pravidla střídavé instalace [29]

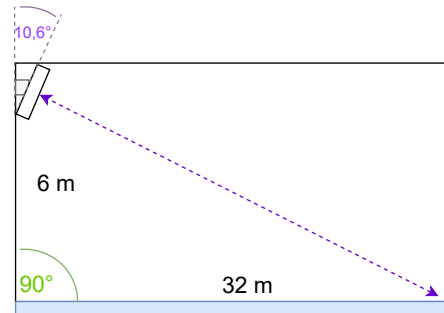
Alternativně pro delší a užší prostory je možné využívat instalaci, kterou naznačuje obrázek 2.4. Zařízení se připevní na přeskáčku na pravou a levou stranu průchodu. Vznikají tak trojúhelníky z třech nejbližších kotev. U průchodů lze využít zařízení instalované střídavě na pravé a levé straně průchodu znázorňuje obrázek 2.4. [31]

Vista DirectFive

Kotva Vista DirectFive na rozdíl od Vista Omni je zaměřená na větší otevřené prostory bez překážek. Jak naznačuje obrázek 2.5, zařízení je vhodné umístit na vyšší místo do rohu místnosti. Díky směrové anténě je důležité správně natočit zařízení tak, aby anténa směřovala směrem dolů před zařízení umístěné na druhé straně místnosti. Způsob náklonu znázorňuje obrázek 2.6.



Obrázek 2.5: Instalace v Vista DirectFive



Obrázek 2.6: Ideální nastavení Vista DirectFive

Zařízení, stejně jako u Vista Omni, nesmí být přímo připevněné na zdi. Držák umožňuje natáčení a náklon o 30 stupňů. V praxi se často využívají obě zařízení za účelem robustního pokrytí a zajištění spolehlivosti.

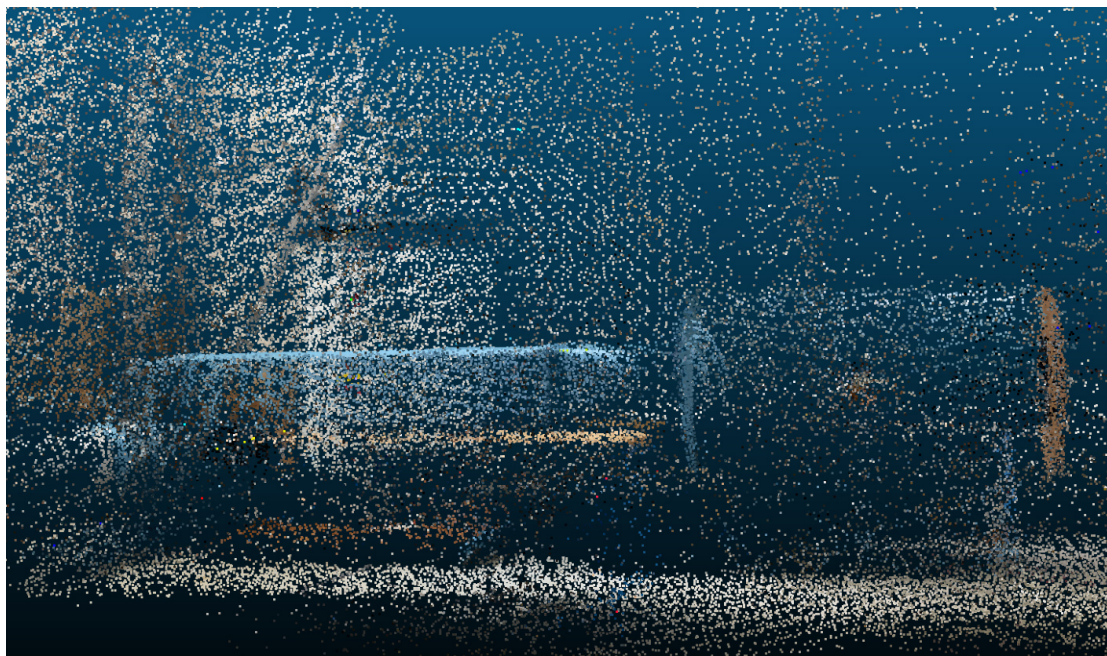
2.4.2 Software

Nedílnou součástí RTLS je software. Firma Sewio Networks s. r. o. dodává balíček RTLS Studio, které zahrnuje několik aplikací. RTLS Manager poskytuje uživatelsky přívětivé prostředí pro konfiguraci systému. Nachází se zde nastavení od základních po pokročilé možnosti nastavení kotev a tagů. Aplikace Sensmap visualization se zaměřuje na instalace ve vnitřních prostorách budov. Vizualizuje získaná data kotev v reálném čase na 2D mapě plánu budovy. Zároveň poskytuje možnost ověření kvality nastavení. Sage Analytics slouží pro vizualizaci zaznamenaných dat za pomoci grafů a tabulek.

2.5 Point cloud

Point cloud jsou seskupená data bodů 3D prostoru. Jednotlivé body reprezentují povrch objektu ve 3D. Každý bod v setu dat má tři unikátní souřadnice v kartézské soustavě souřadnic (x, y, z) . Volitelně uvádí další informace, jako například barvu ve formátu RGB. Typicky jeden data set point cloud obsahuje velké množství bodů (v řádech miliónů). Body nejsou při ukládání organizované a při zpracování se s tímto musí počítat. Standardně se hodnoty na osách ukládají v metrech (pro stroj velmi jednoduchý přístup k ukládání 3D dat). Body vznikají za pomoci 3D skenerů a fotogrammetrie. Body popisují skenované objekty po povrchu. Vlastnosti point cloud způsobu reprezentace 3D dat zjednodušuje práci s objekty při modelování, vykreslení, nebo metrologii. Formát byl popularizovaný při rozvoji počítačového vidění. O způsobu, detekci a analýze dat uložených ve formátu point cloud se zabývá práce [24]. [17]

Na rozdíl od standardních obrázků získaných kamerou, kde se opticky zaznamenávají světelné paprsky, se k získání 3D point cloud dat využívá lidar senzorů. Point cloud data tak narozdíl od standardních 2D obrázků přenáší informace o struktuře objektu. Zároveň díky využití rozdílné technologie při záznamu je nezávislý na kvalitě světla při pořizování záznamu. [17]



Obrázek 2.7: Vykreslená point cloud data

2.5.1 Způsob uložení dat

Základní rozdíl mezi formáty ukládání point cloudových dat je způsob kódování, které může být v ASCII, nebo v binární podobě. Standardní ASCII reprezentace využívá sedmibitové binární číslo pro každý znak pro případ point cloudových dat, kde se hodnoty pohybují v číselných hodnotách x , y a z . Každý řádek tak v souboru nese informaci o jednom bodě, pro který jsou postupně zapsané hodnoty souřadnic x , y a z oddělené mezerou. Tento přístup není vhodný z pohledu výsledné velikosti souboru. Typy souborů zakódovaných tímto způsobem jsou XYZ, OBJ, PTX a ASC.

Binární způsob kódování dat kóduje hodnoty souřadnic bodu podle určitých pravidel. Pro uživatele je forma přímo bez dekódování nečitelná. Výhodou jsou podstatně menší velikosti generovaných souborů a rychlost zpracování. Typy souborů kódované tímto způsobem jsou FLS, PCD, LAS.

Speciálním případem jsou formáty, které podporují oba způsoby kódování, jako jsou například PLY, FBX a E57. Zde se nejčastěji využívá PLY a E57. Formáty lze mezi sebou převádět, avšak mezi některými může docházet ke ztrátě dat.

XYZ Jedná se o volný formát, který slouží k ukládání kartézských souřadnic 3D prostoru. Každý bod je uložený na jeden řádek souboru, kde jsou uvedené souřadnice bodu (x, y, z) a v některých případech je uvedena i barva bodu.

OBJ Tento formát, zpočátku vyvíjený firmou Wavefront technologies a později široce adaptovaný v 3D grafických aplikacích, využívá jednoduchých datových formátů, které reprezentují 3D geometrii, normály, barvu nebo textury. Nejčastěji využívá ASCII kódování, avšak existují méně využívané binární alternativy.

PLY Tento Formát reprezentující list polygonů, které reprezentují objekty, vznikl na základě OBJ formátu a snaží se o přidání nových možností ukládání více fyzických elementů, jako jsou například barva, průhlednost, povrchových normál a důvěrnost bodů. Existují alternativy pro ASCII i binární kódování.

E57 E57 vznikl za účelem spojit a unifikovat způsob ukládání a předávání 3D reprezentace mezi různými programy. Data ukládá formou normálových vektorů a uvádí informace o jejich barvě. Navíc ukládá obrázky a metadata, které vznikají při laserovém skenování a 3D focení. Dnes je to jeden z nejrozšířenějších a nejpodporovanějších formátů, který využívá obě alternativy kódování, v ASCII i v binární podobě.

2.6 Lidar

Lidar, neboli „Light detection and ranging“ technologie, funguje podobně jako klasický radar, ve kterém se místo rádiových vln využívá elektromagnetických vln o rozsahu optické, nebo infračervené vlnové délce. Lidar je aktivním senzorem, který po vyslání signálu čeká a přijímá odraz vln od okolí. Díky využití kratších vln není lidar senzor na rozdíl od radaru náchylný na mlhu nebo mraky. Lidar obdobně jako infračervené zařízení umožňují snímat ve tmě. [23]

Lidar lze využívat ke snímání 2D i 3D obrazu, kde se pro každý pixel vypočítá vzdálenost. Senzor je možné využít i pro získání dat o snímaném objektu. Umožňuje také získat data o úrovni reflektivity materiálů, nebo v případě vysílání vln o různé barevné délce lze snímat i barvu povrchu materiálu. Senzor umožňuje snímání rychlosti díky zaznamenaným změnám ve frekvenci způsobené pohybem zařízení. [23]

Lidar senzory mají rozsáhlé využití v armádě hlavně díky nízké míře osvětlení, které senzor k záznamu potřebuje, ale i v civilním prostředí. Přední výhodou je vhodné využití senzoru pro indentifikaci objektů a orientace stroje v prostoru. Pro člověka jsou získaná data velmi podobná tomu, co vidí. Lidské oko je schopné vnímat podobné frekvence lidar senzoru na rozdíl například od radaru. [23]

Využití technologie se v dnešní době rozrůstá do nejrůznějších odvětví. Nejčastěji se využívá lidar pro vytváření map ve vysokém rozlišení v aplikaci geografie a mapování prostředí nebo využití pro orientaci automobilů v prostoru. Informace o využití lidar senzorů v chytrých automobilech shrnuje článek [10]. Také Kniha [9] popisuje využití senzoru pro letecké mapování terénu.

2.6.1 Princip

Existují dva způsoby zpracování signálu. Diskrétní lidar senzor měří čas letu cesty krátkého světelného pulzu od zařízení k odrazu od objektu a zpět. Kontinuální lidar senzor měří čas rozdílu maxima vln. Tento přístup je hlavně prominentní v automobilním průmyslu, kde je nutné neustále přijímat data o okolí. [11]

Pro výpočet vzdáleností platí vztahy 2.1, kde c je rychlost světla, R je vzdálenost mezi detektorem a objektem, který detekujeme a t je čas od vyslání signálu po příjem. Maximální

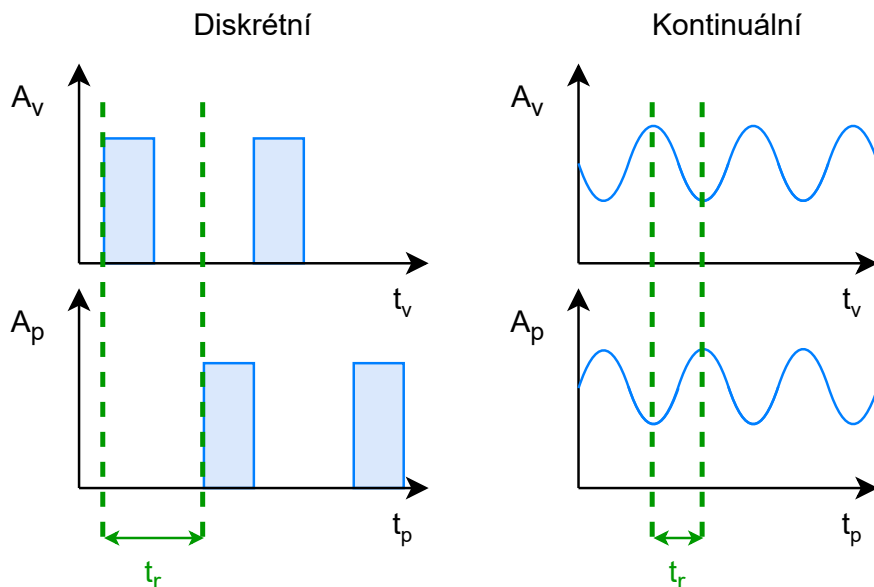
detekovaná vzdálenost je specifikovaná časem t_{rmax} . [11]

$$\text{Vzdálenost: } R = \frac{c * t_r}{2} \quad (2.1)$$

$$\text{Rozložení vzdáleností: } \Delta R = \frac{c * \Delta t_r}{2} \quad (2.2)$$

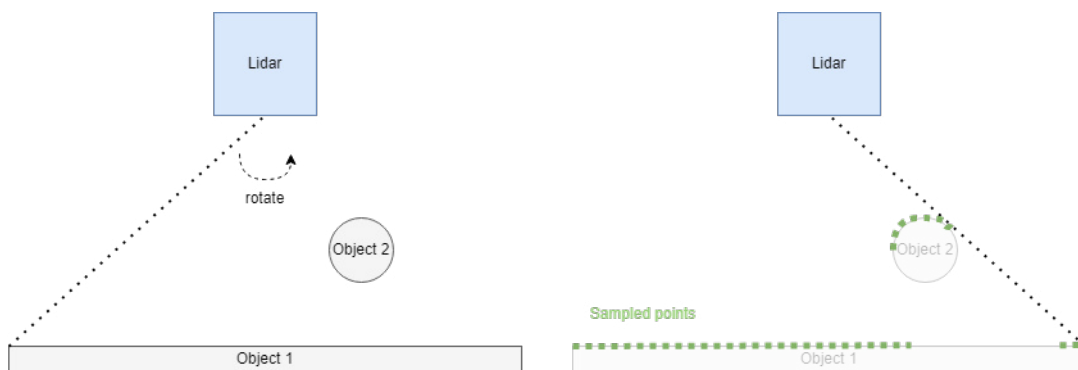
$$\text{Maximální vzdálenost: } R_{max} = \frac{c * t_{rmax}}{2} \quad (2.3)$$

Princip rozdílu získávání doby t_r mezi vysílačem a přijímačem naznačuje graf 2.8. Znamenávané hodnoty amplitudy na vysílači představuje A_v a přijímači A_p . [11]



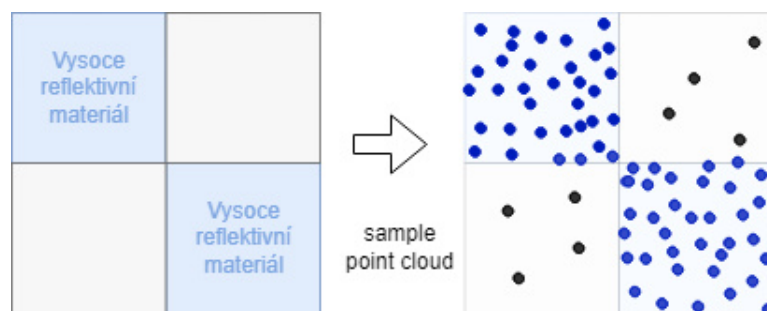
Obrázek 2.8: Princip diskrétního a kontinuálního lidar senzoru

Samotné vzorkování 3D prostoru za pomoci lidarů znázorňuje obrázek 2.9, kde získáváme point cloudové data vzorkováním 3D prostoru. Proces otáčení lidarovým zařízením může být automatický, nebo manuální. Při vzorkování částečně průsvitných materiálů dochází k detekci více odrazů na přijímači pro jeden vyslaný signál s různou amplitudou [13]. [11]



Obrázek 2.9: Samplování 3D prostoru pomocí lidar zařízení

Obrázek 2.7 demonstruje vliv reflexivity materiálu na záznam pořízený lidar senzorem. Způsob odrazu světla od materiálu určuje, jak lidar sensor zaznamená povrch. Průsvitné materiály, jako například sklo, lidarem není možné detekovat přímo. Podobně se chovají materiály, které pohlcují velké množství světla a redukuje množství odraženého světla. Vlastností odrazu se využívá při hledání přesného bodu v datech. K tomu se využívá štítku (viz obrázek 2.10), který naznačuje obrázek, kde jsou dva materiály s rozdílnou reflexivitou. Při záznamu dojde k vytvoření dvou různých skupin bodů, mezi kterými se dá strojem určit střed. Lokalizace za pomoci geolokačních štítků popisuje [28]. [11]



Obrázek 2.10: Vliv vlastností materiálů na získávání dat

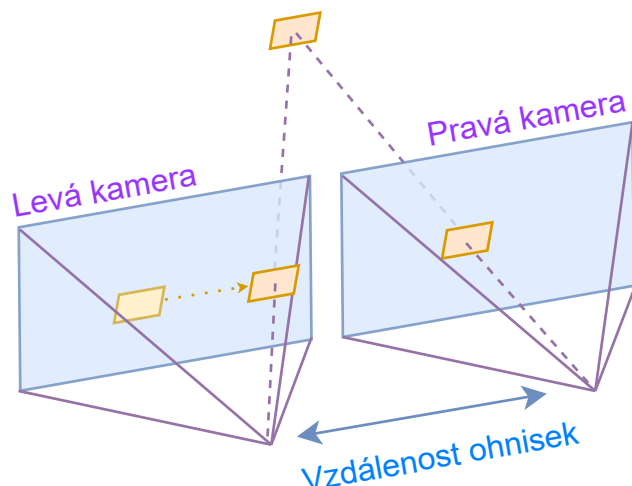
2.7 RGB-d kamery

RGB-d kamera je typ kamery, který poskytuje standardní RGB data s daty hloubky obrazu v reálném čase. RGB data vznikají ze záznamu vlnových délek červeného (R), zeleného (G) a modrého (B) světla. K záznamu dat (d) hloubky se využívají „Time of flight“ senzory, nebo metody stereoskopie. Každý pixel RGB obrázku se mapuje se souvisejícím pixellem hloubkové mapy do jednoho obrázku. Mapování RGB pixelů probíhá v rámci procesoru kamery. Zdroj [3]. Často se spojuje tento typ kamer s VR (Virtuální realita), díky jednoduchému spojení reálného obrazu s strojovou projekcí. Článek zabývající se rekonstrukcí 3D prostoru za pomoci spotřebitelských RGB-d kamer [16]. Článek zabývající se detekcí a klasifikací objektů [2].

2.7.1 Stereoskopické RGB-d kamery

Velmi specifický typ kamer, u kterých se o získávání dat hloubky stará systém založený na stereoskopii, inspirovaný lidským binokulárním viděním. Dvě kamery tak souběžně tvoří záznamy, které jsou následně zpracovány speciálním procesorem kamery, jenž mapuje bloky pixelů levé kamery na informace pravé kamery.

Lz je považovat za alternativní zařízení mobilnímu telefonu. Kamery Intel RealSense řady D, nebo například Microsoft kinect využívají této technologie. Existuje několik výrobců, které se zabývají výzkumem a rozvojem této technologie. Již dříve zmíněný Intel RealSense [14] je řada kamer s rozšířením vnímání prostoru. Další alternativou je firma PMD, která vyvíjí řadu RGB-d kamer pro záznam a orientaci v 3D prostoru.



Obrázek 2.11: Princip výpočtu hloubky za pomoci stereoskopie [12]

Obrázek 2.11, znázorňuje princip výpočtu hloubkové mapy. Na základě rozdílu pozice mapovaného pixelu vzniká trojúhelník, ze kterého lze vypočítat vzdálenost pixelu. [12]

2.7.2 RGB-d kamery s senzorem

Narozdíl od stereoskopických kamer je zde eliminovaný výpočet vzdáleností. Stereoskopický systém je u něj zastoupený senzorem umožňující záznam 3D dat. Nejčastěji se využívají lidar senzory 2.6.

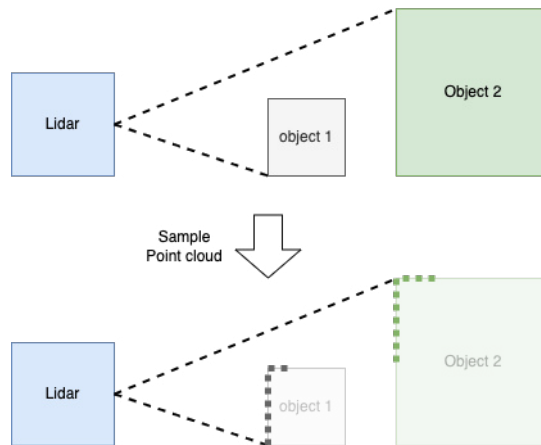
Intel představil pouze jednu kameru s lidar senzorem L515. Alternativní produkt s „time of flight“ senzorem je Flexx2 od firmy PMD Technologies [27]. Hlavním rozdílem těchto produktů je způsob využití. Flexx2 oproti Intel umožňuje záznam okolního prostředí díky vestavěné podpoře SLAM 2.7.3. [1]

2.7.3 SLAM

SLAM, neboli „Simultaneous Localization and Mapping“, je proces způsobu výpočtu pozice kamery a současně záznamu v 3D prostoru. Proces využívá pouze vizuálních dat. Uplatňuje se dnes především v automobilním průmyslu při vývoji autonomních automobilů. [21]

Typicky využívá několika bodů, které jsou pro algoritmus nejzajímavější. Tyto body jsou algoritmem spojeny v trojúhelníky. Pohybem v prostoru se sleduje změna ve vytvořených trojúhelnících, z něj je následně rekonstruován 3D model. [8]

Alternativní Lidar-SLAM využívá senzoru lidar. Výhodou je přesnější a jasnější kontext pozice v 3D prostoru díky kvalitě získávaných dat. Hlavní nevýhodou tohoto přístupu je mapování průsvitných materiálů. Lidar-SLAM je odkázán na data pouze z lidar senzoru. V případech, kdy je část objektů před senzorem blokována jiným objektem, dochází k různě nedokončeným tvarům. Obrázek 2.12 naznačuje, jak pro nás je důležité získat data o objektu 2. Objekt 1 brání senzoru ve výhledu, protože je na stejné úrovni. Tím dojde ke změně tvaru a ztrátě dat při konverzi do point cloud. [21]



Obrázek 2.12: Neúplnost objektu při získávání dat lidar senzorem.

Tento problém značně ztěžuje detekci pomocí umělé inteligence, kde tyto artefakty mohou mít nejrůznější podobu a není tak snadné pouze z dat lidarů zjistit, o jaký objekt se jedná. [21]

2.8 Radar

„Radio detection and ranging“, neboli radar, je radiolokační systém, využívající rádiové vlny na odhad vzdálenosti, úhlu a rychlosti sledovaného objektu. Dnes se radar využívá především k lokalizaci letadel, lodí, vozidel, nebo mapování počasí či terénu. Systém je tvořen vysílačem, který vysílá přes anténu rádiové vlny a přijímačem, který předává informace procesoru, jenž zpracovává požadované parametry objektu. Rádiové vlny jsou pulzované a následným odrazem od objektů se vrací k přijímači udávající informace o rychlosti a lokaci objektu.

2.9 QR kod

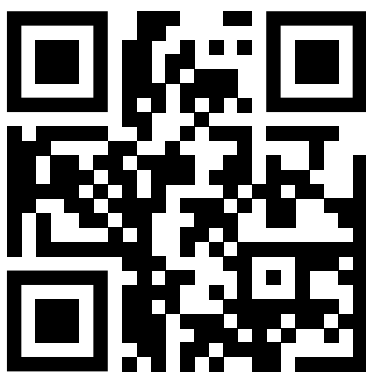
„Quick response code“, neboli QR kód, je alternativou, snadno rozpoznatelnou strojem. Často tak QR kódy ukládají data, která slouží k sledování, identifikaci, nebo lokalizaci. Existují čtyři standardní módy kódování lišící se počtem maximálně uložených dat. Existují však další méně populární alternativy kódování. [15]

Kódování	Maximální počet	Bitů na znak	Přehled dostupných znaků
Číselné	7,089	$3\frac{1}{3}$	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Alfanumerické	4,296	$5\frac{1}{2}$	0–9, A–Z, mezera, \$, %, *, +, -, ., /, :
Binární	2,953	8	ISO/IEC 8859-1
Kanji	1,817	13	Shift JIS X 0208

Tabulka 2.1: Přehled velikosti kódovaných datových typů QR kódu pro „40-L“. [15]

2.9.1 Princip ukládání dat

Na rozdíl od čárových kódů QR kódy jsou čitelné pod různými úhly díky způsobu ukládání dat do 2D prostoru. Data jsou reprezentována černobílou barvou sektorů, které se nachází uvnitř čtverce. V blízkosti tří rohových čtverců jsou umístěná data o typu formátu, nastavení korekce chyb a jeho úroveň. Následně se dělí do sektorů po bytech, kde tvar sektorů a typ ukládání je specifikovaný v informaci o formátu. [15]



Obrázek 2.13: QR kód s uloženým textem „DP Michal Bucher“



Obrázek 2.14: Bar kód s uloženým textem „DP Michal Bucher“

Obrázky 2.13 a 2.14 ukazují rozdíl ukládání dat do QR, nebo čárového kódu. Obě varianty kódují stejný alfanumerický řetězec „DP Michal Bucher“.

2.9.2 Lokalizace a čtení

K detekci slouží tři čtverce umístěné na rozích QR kódu, jenž umožňuje určit pozici a úhel natočení a pohledu pro správné čtení dat. Menší černé čtverce jsou konvertovány do binární podoby a zkontrolovány validací chyb. Algoritmus korekce chyb má čtyři úrovně L, M, Q, H. L je nejnižší úroveň korekce, která dovoluje opravit zhruba 7% ztracených bytů. Nejvyšší H tak umožňuje opravit až 30% chybějících bytů. [15]

Kapitola 3

Zhodnocení možností

Část teorie uvádí možnosti, které umožňují 3D mapování prostoru využít. Tato kapitola shrnuje získané vědomosti o typech možností mapování a detekce objektu. Blíže popisuje existující řešení, která by byla aplikovatelná pro zaměření nalezení objektů v prostoru. Zaměřuje se na porovnání výstupních dat a vyhodnocení vhodnosti pro kontext práce. Bere se zde i ohled na dostupnost větší skupině uživatelů.

3.1 Mobilní telefon

Jedním ze zařízení, které je po ruce skoro každému člověku, je mobilní telefon. Z tohoto důvodu se tato práce snaží hledat řešení právě ve využití aplikací a dalších možností různých mobilních zařízení. Některé mobilní telefony jsou osazené i podpůrnými senzory pro orientaci v prostoru. Samozřejmě ideální by bylo vystačit se ke zmapování prostředí pouze s kamerou telefonu. Na základě této myšlenky vzniklo samotné zadání této práce, jež vycházelo z možností aplikace Matterport [22].

3.2 Matterport

Matterport je balíček služeb, který zprostředkovává záznam a vytváření 3D modelů vnitřních prostorů na základě fotografií nebo lidar dat. Umožňuje tak jednoduché získání dat například mobilním zařízením a následnou rekonstrukcí 3D prostoru ze získaného data setu fotografií za pomoci natrénované umělé inteligence, která vyhodnocuje předměty a pozici jednotlivých panoramaticky nasnímaných fotografií. Celé zpracování a vyhodnocení modelů probíhá v rámci cloudového serveru.

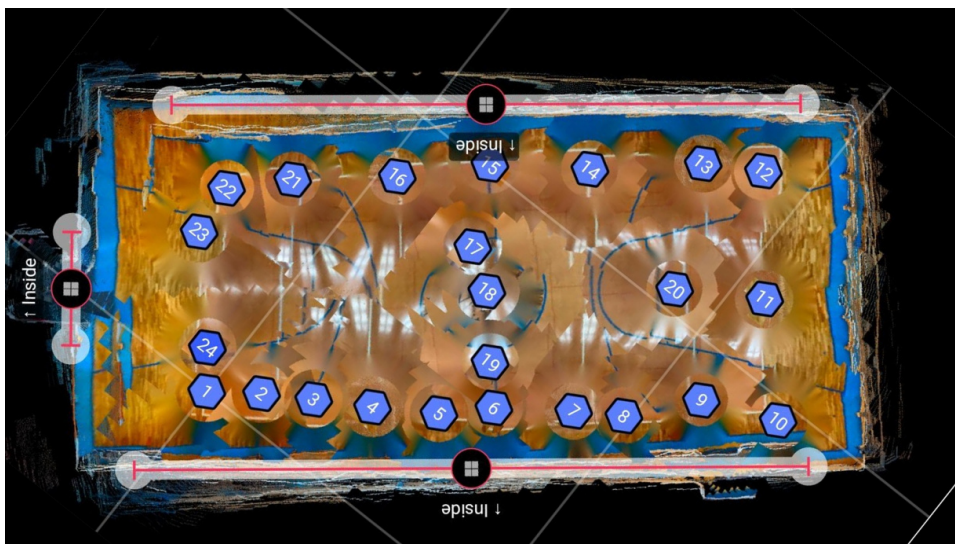
Zároveň Matterport poskytuje SDK, umožňující editaci vygenerovaného 3D modelu. Dovoluje vkládání objektů a základní skriptování pro simulaci prezentace prostředí, jako například změnu světla prostorů. Software běží v rámci javascriptového frameworku v prohlížeči. [22]

Součástí balíčku je aplikace „Matterport Capture“, která je volně dostupná pro mobilní telefony s operačním systémem Android a IOS. Umožňuje skenovat 3D prostor s využitím standardní kamery. Aplikace intuitivně provádí uživatele při záznamu fotografií na místě v prostoru. Proces na zvolených místech uživatel několikrát opakuje v průběhu mapování souvislého záznamu podle velikosti prostoru.

Seznamování se s funkcionalitou aplikace probíhalo na dvou velmi rozdílných místech. První byla domácí pracovna, která je zhruba 4 metry široká a 5 metrů dlouhá. Druhým

prostorem byla tělocvična lokální školy, do které mi bylo pro účely této práce umožněn přístup. Jedná se o prostor 30 metrů široký a 50 metrů dlouhý. Na rozdíl od první místnosti zde chybí detailní dekorace a objekty uvnitř místnosti. Účelem bylo vyzkoušet aplikaci ve větším prostoru. Během experimentu zde však bylo zjištěno hned několik problémů.

Záznam prostorů byl vždy testován na třech různých přístrojích Razer phone, Samsung Galaxy Tab 7 a iPhone 12 Pro. Zde je důležité dodat, že na kvalitu a výsledný 3D model nemělo zvolené zařízení téměř žádný vliv. Velký rozdíl byl ale v kvalitě jednotlivých 2D panoramat, která vznikají při fotografování v jednom bodě. Panoramata nejsou příliš zajímavá a u větších prostorů mohou být získaná data velmi rozsáhlá (řádově v GB). Při generování 3D modelu zmiňované tělocvičny, která se skládá zhruba z 30 individuálních záznamů, je téměř 4 GB v horší kvalitě fotografií a až 10 GB v kvalitnější. Aplikace v průběhu skenování po zhruba 10 individuálních záznamech začala na systému Android neustále upadat do chybového stavu a bylo tak nutné několikrát opakovat záznam v jednom místě. Doba snímání fotek na vytvoření jednoho panoramatu se pohybovala kolem 1 minuty. Po dokončení bylo panorama podlaží zasazené do 2D virtuálního plánu získaných dat. Obrázek 3.1 demonstruje vytvořený plán podlaží v aplikaci. Bylo nutné neustále aplikaci napomáhat a data manuálně upravovat.

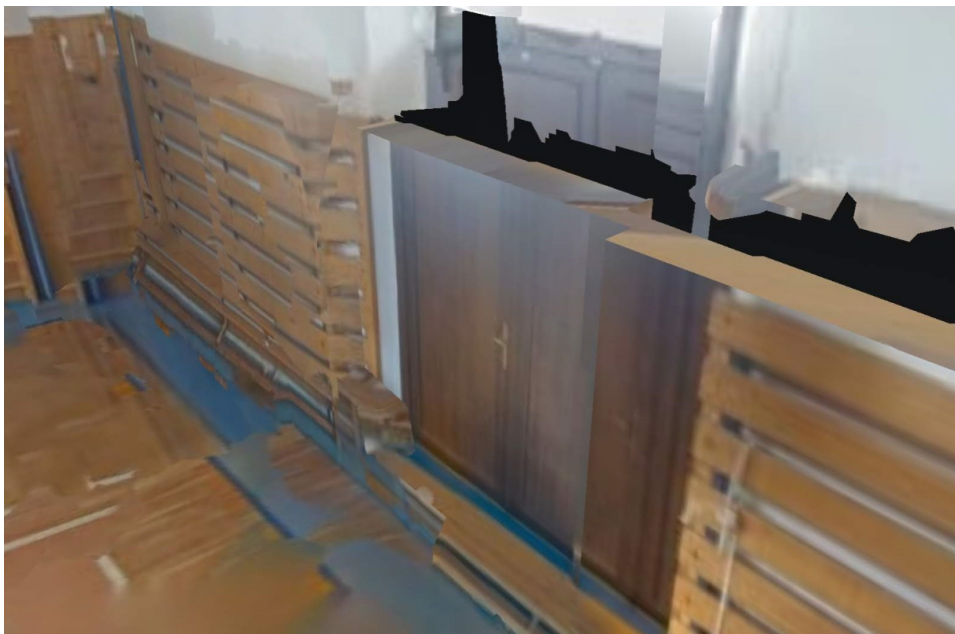


Obrázek 3.1: Automaticky vytvořený 2D plán v aplikaci

Další problém se objevil po dokončení skenování prostoru, který představuje dobu zpracování a limitace na cloud serveru. Pro vytvoření 3D modelu musí uživatel odeslat data o záznamu prostoru na cloudový server, který data zpracovává. Doba zpracování závisí na počtu a velikosti dat. V průběhu testování se doba pohybovala zhruba mezi 2 hodinami u pracovních a až 10 hodinami u prostoru tělocvičny. Zároveň neexistuje možnost ovlivnit kvalitu generovaného modelu, pro který mobilní aplikace umožňuje pouze několik základních úprav jako už zmiňovaná korekce snímaných pozic panoramat mezi sebou. I zde při velkém počtu míst aplikace padala do chybného stavu při úpravě nebo přepočtu pozic. Pro malé prostory byl tento problém zanedbatelný.

Významným omezením je pak mít možnost využít pouze jeden aktivní záznam pro zpracování na cloudovém serveru. Zároveň neumožňuje nahrávat více než jeden záznam ke zpracování. Snímání většího počtu prostorů, nebo opakování záznamu tedy není možné v ro-

zumné době provést, kvůli dlouhému zpracování na straně serveru. Po dokončení zpracování záznamu lze data archivovat a následně nahrát další.



Obrázek 3.2: Část chybně vygenerovaného modelu dveří tělocvičny

Společnost Matterport udává, že umožňuje export do mesh 3D modelů (chybné části modelu demonstrují obrázky 3.2 a 3.3) a cloud pointu. Nejsou zde ale na první pohled příliš jasné podmínky, za kterých lze exportovat data. Tlačítko v aplikaci je zobrazené, ale není přístupné. Vyexportovat model v cloud pointovém formátu lze, pouze pokud byl záznam pořízen speciálním hardwarem poskytovaným přímo od společnosti Matterport nebo za využití Matterport Capture na Apple zařízením s lidarem. Pro odblokování exportu také musí mít uživatel aktivní předplatné.



Obrázek 3.3: Část 3D modelu, kde došlo k výrazné deformaci prostředí.

Jedinou výhodou aplikace je podpora téměř jakéhokoliv mobilního telefonu s kamerou. Oproti ostatním způsobům uvedených v této kapitole poskytuje data, která nejsou pro tuto práci vhodná. Stejně problémy se objevují i u ostatních aplikací založených na stejném principu.

3.3 Mobilní telefon s lidarem

Alternativou je mobilní telefon s lidar senzorem. Práce se primárně zaměřila na využití mobilních telefonů společnosti Apple. Ta dodává tento senzor do řady tabletů a telefonních mobilů typu Pro. Od řady 11 lze tak využívat tato zařízení k záznamu 3D prostoru.

Pro zařízení Apple typu Pro existuje poměrně rozsáhlé množství aplikací pro vytváření point cloud, nebo mesh 3D modelů. Na základě hodnocení byly vybrány dvě aplikace Polycam, 3D scanner app. Záznam dat byl velmi podobný, avšak hlavním rozdílem je způsob zpracování 3D prostoru a formát, ve kterém aplikace data vizualizuje a exportuje. Zde je nejlepším kandidátem aplikace 3D scanner app, která podporuje formáty jako PLY a E57. Lze tak jednoduše zobrazit point cloud data dalšími aplikacemi.

Při testování ale nebylo jasné, jakým způsobem menší kotvy s generickým tvarem a bílou barvou přesně detekovat. Záznam z větší dálky kvůli nepřesnosti a chybným bodům zabraňuje možnosti detekovat kotvy pouze na základě point cloud dat a určit, zda se opravdu jedná o zařízení. Při detekci by bylo vhodné využít kvalitní kamery, které jsou mobilnímu telefonu také přístupné.

3.3.1 3D scanner app

Jedná se o aplikaci pro iOS mobilní zařízení. Podporuje základní funkce záznamu objektů a prostředí do 3D modelů. Hlavní zaměření aplikace je skenování jednotlivých objektů a následný export do programů pro 3D vizualizaci nebo rozšířenou realitu. Hlavním nedostatkem aplikace pro účel detekce je omezení parametrizace záznamu. Množství bodů, které aplikace zaznamenává, ani jejich četnost není možné změnit. Nebylo tak možné experimentovat s hustotou zaznamenaných bodů.

3.3.2 Polycam

Polycam je placenou aplikací pro iOS a Android mobilní zařízení. Obdobně jako služba Matterport aplikace využívá především rekonstrukce 3D prostředí na základě fotogrametrie. Pro mobilní zařízení s lidar senzorem umožňuje vytvořit záznam lidar senzorem. Hlavním rozdílem je použitý formát výstupních dat s hlavním zaměřením 3D mesh modelu. Mesh reprezentace nebyla vhodným formátem, který by bylo jednoduché jakkoliv strojově detekovat a rozpoznávat. Zároveň omezená možnost parametrizace záznamu zabránila více experimentovat s různými počty zaznamenaných bodů stejně jako u předchozí aplikace.

3.4 ARKit

ARKit je řešení knihovny pro rozšířenou realitu neboli „Augmented Reality (AR)“ od společnosti Apple. Balíček obsahuje základní funkcionality pro podporu spojení záznamů senzorů mobilního zařízení s počítačem generovanými 3D objekty. Knihovna využívá kamer a senzorů mobilního zařízení. Knihovna implementuje hned několik funkcionalit pro rozpoznávání objektů, sledování pohybu člověka, nebo vkládání virtuálních objektů do prostoru.

Modelová řada „Pro“ mobilů a tabletů od generace 11 mají na přední i zadní části umístěný lidar senzor 2.6, který umožňuje získávání přesnějších dat o prostoru kolem. [4]

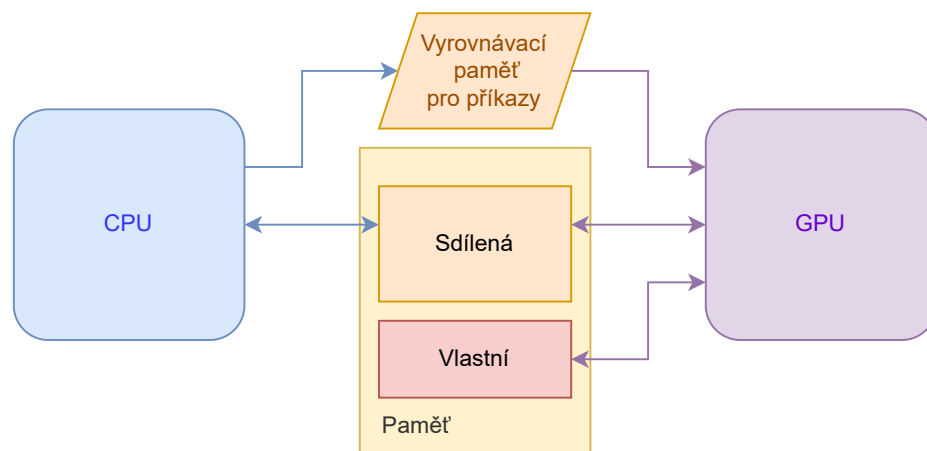
3.5 Metal API

Metal API zprostředkovává přímý přístup ke grafické kartě iOS zařízení. Umožňuje vyobrazení komplexních scén a výpočet náročných matematických rovnic se zaměřením na výkon. Metal se využívá pro vývoj 3D her, zpracování a úpravu videa a vědeckých výzkumech, kde je snahou analyzovat rozsáhlé datové sady například pro strojové učení. Metal API je využíván jazyky Swift, Objective-C a C++, které přes přístupné API komunikují s „Metal device“ (Grafický procesor zařízení). [6]

API poskytuje přístup k tzv. „Metal Performance Shaders“, které jsou kolekcí optimalizovaných grafických „shaderů“. Optimalizace jsou specifické pro grafický procesor zařízení. Optimalizace grafických shaderů se zaměřuje na: [7]

1. Extrakci statistických dat z obrázků.
2. Implementaci neuronových sítí pro strojové učení.
3. Výpočet rovnic s využitím maticových operací.
4. Akcelerace „ray tracing“ vykreslování.

Bližší informace o komunikaci mezi CPU a GPU jednotkami lze dohledat v dokumentaci [6] a online článku [18].



Obrázek 3.4: Vzájemná komunikace CPU a GPU přes paměť. Zdroj [20]

3.6 RGB-d kamery

Ačkoliv zařízení poskytují velmi dobrý základ pro snímání obrazu a detekci objektů, u většiny zařízení není zpětná vazba o pohybu v 3D prostoru, které mobilní telefony poskytují ze senzorů, jako jsou akcelerometr, gyroskop. Pro účely této práce tak připadá v úvahu možné využití Flexx2, protože není třeba přidávat žádných externích senzorů vzhledem k využití SLAM algoritmu, jenž je součástí dodávaného vývojářského balíčku.

Alternativním kandidátem byla kamera Intel RealSense L515. Ostatní zařízení uvádí v listu specifikací dosah zhruba 4 až 5 metrů s chybou 2%. U kamery L515 je uveden dosah až 9 metrů a chybou v milimetrech. Intel ukončil podporu a vývoj pro tento typ kamer.

Knihovna Intel RealSense SDK tvoří vrstvu mezi kamerou a počítačem. Podporuje množství dnes populárních jazyků, jako například C++, Python nebo C#. Knihovna obdobným způsobem jako ARKit zaobaluje základní funkcionalitu a prvky kamery a umožňuje jednoduchý přístup k datům. Na rozdíl od knihovny Apple poskytuje bližší přístup přímo k hardwarové implementaci a umožňuje přímou práci s prostředky na kameře. Jednotlivé senzory kamery lze využívat nezávisle bez vzájemného narušování.

Samotný hardware na základě specifikací a dostupných informací podával lepší výsledky než mobilní telefony Apple. Problém byl s dostupností a s velmi účelným zaměřením kamer. Firma Sewio Networks s.r.o. tak preferovala variantu mobilního telefonu.

3.7 Knihovna OpenCV

Open source vyvíjená knihovna OpenCV poskytuje vývojářům několik stovek algoritmů se zaměřením na strojové vidění. Základ knihovny je postavený na jazyce C++. Knihovna poskytuje definice základních datových struktur při práci s vícerozměrnými maticemi, které se využívají při zpracování obrazu. Za tím účelem poskytuje hned několik lineárních a nelineárních filtrů, geometrických transformací obrazu, vytváření histogramů nebo coverze mezi barevnými formáty. Využití knihovny není omezené pouze na statické obrazy, ale implementuje i funkcionality pro zpracování videa, která jsou například zaměřená na sledování pohybu objektů, či extrakci objektu z videa nebo pro odhad pohybu objektu. Velká část knihovny se věnuje detekci a klasifikaci objektů.

3.8 Shrnutí

Většina z popsaných řešení, která jsou v této kapitole uvedena by fungovala a objekt detekovala. Rozhodujícím faktorem finálního výběru proto byla dostupnost zařízení. Systém má sloužit zákazníkům a firmě Sewio pro usnadnění a urychlení instalace lokalizačního systému. Selekci došlo k výběru mobilního zařízení Apple jako referenčního přístroje pro implementaci. Není tak zapotřebí žádný speciální hardware. Firma Apple nabízí lidar senzor už v několika řadách tabletů a mobilních zařízeních.

Hlavním důvodem pro vytvoření nové aplikace místo využití už stávajících aplikací 3D scanner app a Polycam jsou velmi limitované možnosti nastavení. Programy také nepodporují detekci. Znamená to další krok pro uživatele, který by musel být prováděný na externím zařízení a dalším programem pro nalezení kotev.

Vzhledem k účelu práce bylo vhodné vytvořit program, který bude více specifický k charakteru zadání a umožní záznam a detekci zároveň, a tedy eliminuje přenášení dat a další zpracování. Uživatel tak dostane ihned zpětnou vazbu o pozicích zařízení.

Dalším faktorem je také rozsáhlá podpora knihoven vhodných pro vytváření aplikací s vizualizací a práci s objekty v 3D virtuálním světě. Knihovny jako Vision a ARKit jsou stále vyvíjené a podporované. Není tak třeba psát vše od úplného začátku.

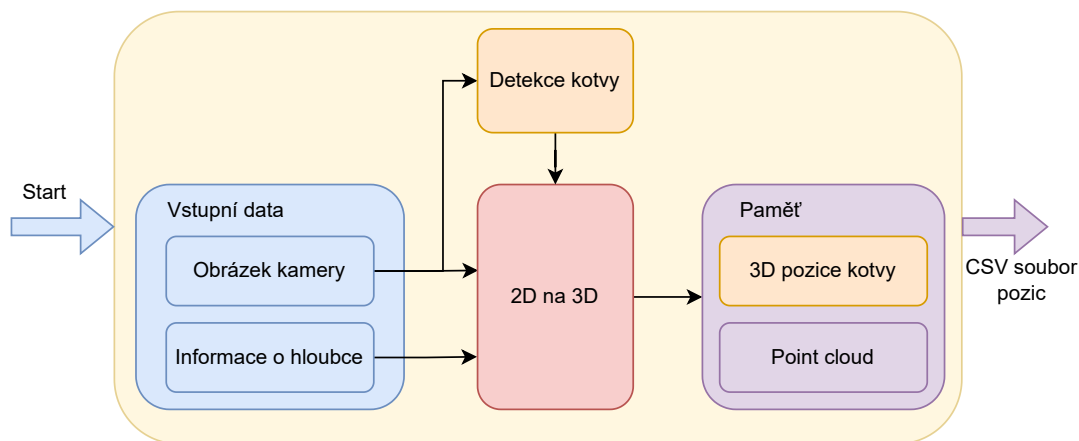
Kapitola 4

Návrh řešení

Na základě výsledků a srovnání v části 3 byl vytvořen návrh algoritmu, který lze aplikovat na mobilní zařízení s podporou 3D senzorů a RGB-d kamery. Obecná část návrhu popisuje myšlenku a propojení souběžně komunikujících částí. Část mapování prostoru popisuje způsob transformace a spojení 2D dat z kamery a hloubkové mapy na 3D body. Poslední část popisuje způsoby přístupu při detekování pozic jednotlivých kotev.

4.1 Obecný návrh

Obrázek 4.1 naznačuje rozdělení a průběh, kde aplikace po spuštění umožňuje zahájit kontinuální záznam, který paralelně získává data a vyhodnocuje pozice zařízení. Vstupními daty jsou získané snímky zachycené kamerou a bitová mapa informací o vzdálenosti nástroje od objektu. Výstupem je point cloud formát všech zaznamenaných bodů ve formátu .xyz 2.5.1 včetně barev a .csv soubor pozic kotev ve vhodném formátu pro import do Sensmap Visualization aplikace. Navržený algoritmus lze využít na mobilních telefonech s lidar senzorem a RGB-d kamer.



Obrázek 4.1: Grafické znázornění algoritmu.

Po spuštění aplikace se čeká na nová data. Při získání nových snímků z kamery a bitové mapy vzdálenosti zařízení je notifikovaná část, která odešle data do detekční části a do části, která transformuje 2D body displeje do 3D pozic virtuálního světa. Detekci a nalezení pozice kotev zajišťuje strojové vidění. Při detekování objektu tak vypočítá pozice na displeji.

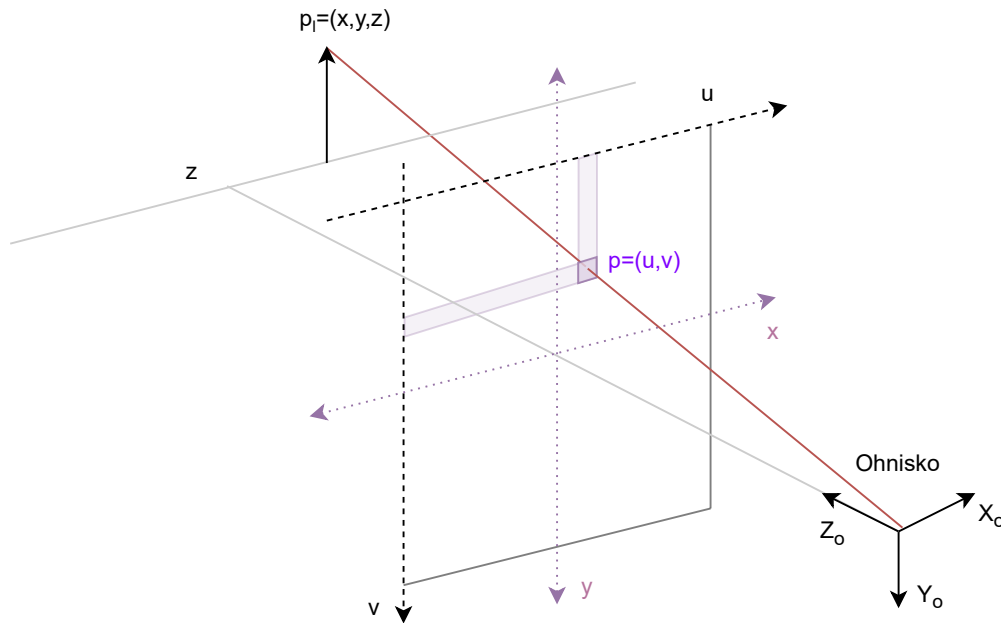
Zároveň se vytvoří statická mřížka bodů na obrazovce. Pozice bodů mřížky a získané pozice kotvy na displeji se za pomoci vzorkování vzdálenostní mapy transformuje do 3D prostoru. Body se ukládají na dvě místa podle typu pro jednodušší a rychlejší exportování dat.

4.2 Mapování prostoru

K mapování prostoru je zapotřebí informací o tom, kde se zařízení nachází a jaké jsou data poskytované kamerou a senzory. Data prostoru z RGB-d kamer a mobilních telefonů se reprezentují 2D mapou, kde u každého pixelu se nachází kódovaná hodnota vzdálenosti. Algoritmus je inspirován standardním přístupem pro práci s 3D reprezentací scén v počítačové grafice. Využívá se zde transformací bodů na základě kontextu, ke kterému se bod vztahuje. Lokálním kontextem je popsán vztah bodu ke kameře. Přenáší se informace o pozici bodu na displeji a transformace natočení displeje. Druhý globální kontext bodu přenáší data o tom, jak je oproti počáteční pozici zařízení natočené a posunuté do prostoru.

4.2.1 Konverze 2D bodu do 3D prostoru

Známe-li pozici bodu na displeji v pixelech, tedy pozici $p = (u, v)$, pro hodnoty platí $u, v \in \{0, 1, 2, \dots\}$ a hodnotu z , kde $z \in \mathbb{R}$. Hodnota z představuje vzdálenost zařízení od objektu na pozici pixelu displeje bodu p . Spojením vznikne vektor $p_l = (x, y, z)$, který je nutné transformovat do bodu p_w . Bod p_w představuje pozici v právě běžícím záznamu vůči pevnému bodu začátku. Ten vzniká při zahájení záznamu. Transformace bodu probíhá p_w vynásobením matice Z_w . Vztah nezahrnuje žádnou formu optického zkreslení. [25]



Obrázek 4.2: Projekce normalizovaných 2D bodu do lokálního 3D prostoru kamery. [25]

Princip výpočtu pozice lokálního bodu p_l z pozice p na pixelu (u, v) demonstruje obrázek 4.2. Bod p na obrázku naznačuje pozici pixelu na souřadnicích (u, v) . p_l se vztahuje pouze k

aktuálnímu bodu ohniska. Převod bodu p_l na p_w za pomoci matice Z_w demonstruje obrázek 4.3.

$$K_{ii} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$p_l = K_{ii} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} z = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

K_{ii} je inverzní matice informací o lokálním prostředí kamery zařízení. Bod p_w vznikne vynásobením bodu p_l s inverzní projekční maticí Z_w , která je produktem násobení matic rotace zařízení Z_r , transformace, neboli pohybu zařízení Z_t a jednotkové matice I . Hodnoty rotace a transformace pochází z pohybových senzorů zařízení.

$$Z_w = Z_r * Z_t * I = \begin{bmatrix} r_x & u_x & f_x & 0 \\ r_y & u_y & f_y & 0 \\ r_z & u_z & f_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Hodnoty r, u, f reprezentují rotaci bodu na příslušné ose x, y, z . t je hodnota transformace posunu bodů na osách x, y, z .

$$p_w = \begin{bmatrix} x_{pl} \\ y_{pl} \\ z_{pl} \\ 1 \end{bmatrix} Z_w = \begin{bmatrix} X' \\ Y' \\ Z' \\ w \end{bmatrix}$$

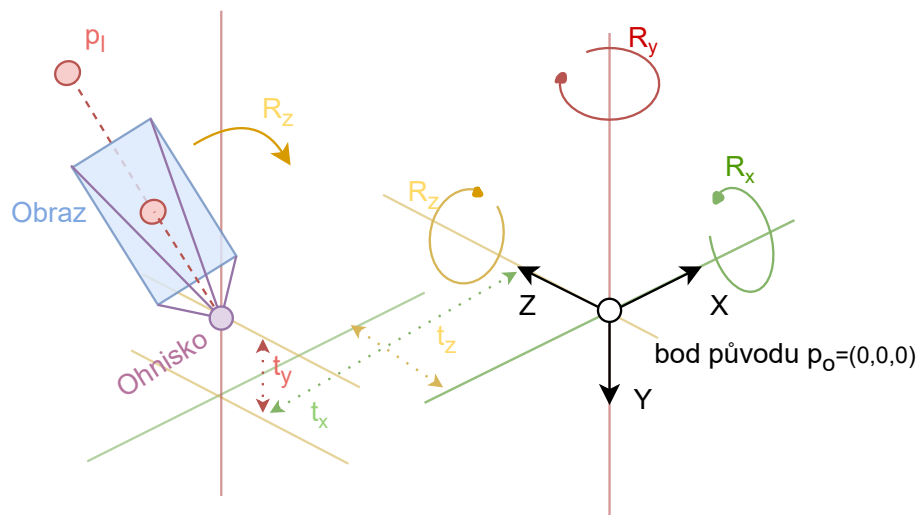
Poslední krok normalizace 3D souřadnic bodu získáme pomocí dělení hodnotou umístěné na pozici w ve výsledném vektoru p_w .

$$p_w = \begin{bmatrix} X'/w \\ Y'/w \\ Z'/w \\ w/w \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Výsledný vektor p_w představuje body v 3D prostoru, se kterými lze dále pracovat v algoritmu.

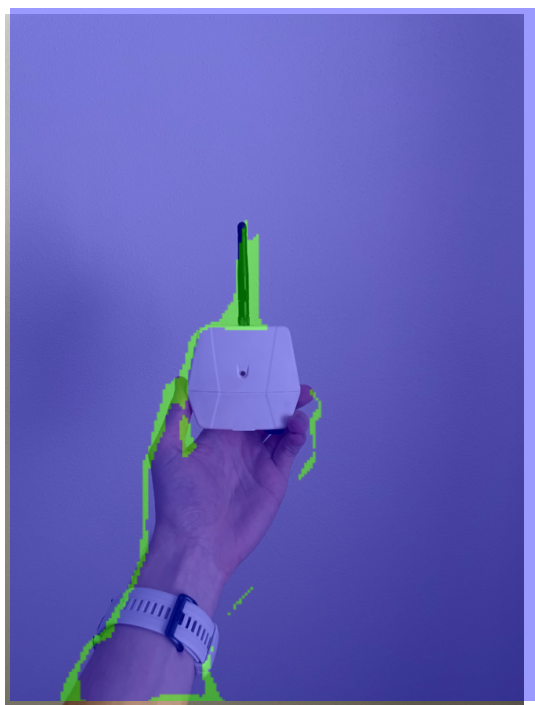
Nové body p_w vznikají na základě podmínek, za kterých se kontroluje jestli bylo zařízení natočené o dva stupně v jakémkoliv směru, nebo zařízení zaznamenalo pohyb 2 centimetry. Toto opatření zabraňuje zahlcování paměti body zaznamenaných na stejném místě.

Obrázek 4.3 demonstruje závislost lokálního bodu p_l získaného transformací. Pro určení správné pozice bodu je důležité porovnat pozici a úhel náklonu bodu ohniska. Od počátečního bodu se k pozici přičte kompenzace pozice na základě detekovaných hodnot pohybových senzorů a rotace zařízení.



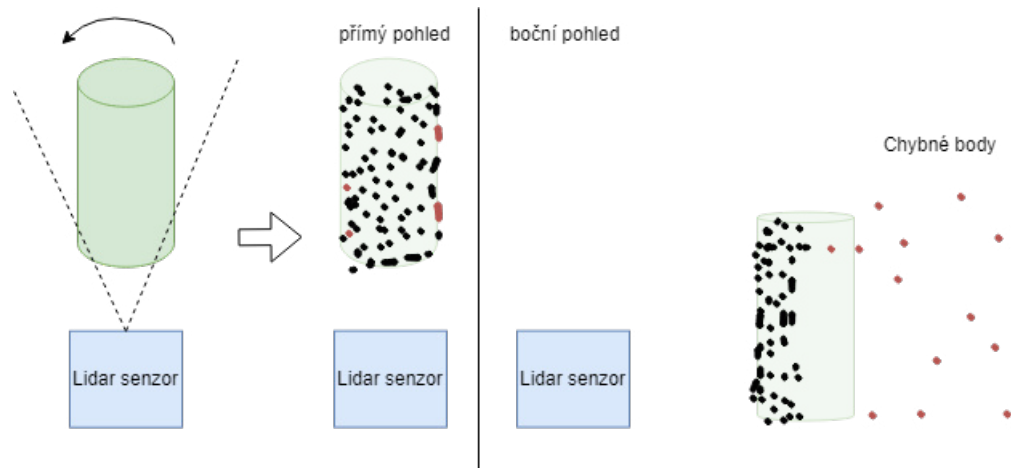
Obrázek 4.3: Konverze lokálního bodu kamery do 3D virtuálního světa

Obecně k funkci algoritmu v okamžiku transformace z 2D do 3D bodu stačí pouze pozice pixelu a jeho vzdálenost. Některá zařízení přikládají k mapě kódovaných vzdáleností také informace o jistotě, se kterou lidar sensor vzdálenosti detekoval. Přidává k jednotlivým bodům informaci o kvalitě kódované vzdálenosti. Vrstva mapy jistoty je znázorněná na obrázku 4.4. Ve spojení s obrázkem 4.5 lze lépe znázornit vliv informací o jistotě s informací detekované vzdálenosti.



Obrázek 4.4: Překryv fotky s daty jistoty ze zařízení iPhone 12 Pro.

Pro zlepšení kvality výstupních bodů algoritmus při transformaci bodu zároveň přidává informaci, na základě které lze body filtrovat. Jistota se u většiny knihoven implementuje jako celočíselné pozitivní číslo, které nabírá rozsahu podle hardwarové limitace senzoru. To naznačuje obrázek 4.5, kde body na hranách se detekují několikrát a je z nich následně počítaný průměr pro finální bod. V prostoru tak náhodně vznikají body, které mají vlastnosti šumu.



Obrázek 4.5: Kvalita dat v kontextu záznamu objektu s nízkou úrovní jistoty na hranách

4.3 Detekce objektu

Paralelně k záznamu a informacích o prostoru je třeba detekovat kotvy. Pro účel této práce bylo třeba najít efektivní cestu detekce a klasifikace kotev. Zvolený způsob musí splňovat dvě charakteristiky. V první řadě je důležité, aby při detekci nedocházelo k detekci na jiných místech. Zároveň musí být algoritmus schopný určit, o který typ kotvy se jedná. Vzhledem k typu prostředí, ve kterém se zařízení detekuje, bylo důležité zavést pravidlo, jak odlišit kotvy firmy Sewio Networks s. r. o. od ostatních podobným způsobem instalovaných zařízení. Vzhledem k jednoduchému vzhledu a relativně malé velikosti se u Vista Omni kotvy včetně antény jedná o 95 milimetrů na šířku a 200 milimetrů na výšku. Vista DirectFive má oproti Vista Omni výraznější a unikátnější tvar se šířkou 212 milimetrů a výškou 190 milimetrů 2.2.

Zařízení se instalují do průměrné výšky 3 metrů. Detekce by měla umožnit detekovat kotvy na dostatečnou vzdálenost pro případy, kdy není možné se dostat přímo pod kotvu. S firmou Sewio Networks s. r. o. tak bylo dohodnuto, že ideální vzdálenost detekce bude průměrně 5 metrů. Pro kotvu Vista Omni s šířkou o velikosti 95 milimetrů by byla detekce na vzdálenosti větší než 2 metry problematická. Změna velikosti zařízení, nebo vzhledu není možná.

Pro účely práce tak byl zvolen alternativní způsob detekce za pomoci QR kódu 2.9. Důvodem pro jeho zvolení je jednoduchost a rychlost detekce za pomoci strojového vidění. Zároveň díky možnosti přenosu a kódování dat lze QR kód využít pro označení kotev a uvedení základních parametrů. Problematiku detekce a dekódování QR kódů poskytuje hned několik volně dostupných knihoven. Vision pro Apple a OpenCV. OpenCV má rozsáhlou podporu téměř ve všech dnes populárních programovacích jazycích. Existují avšak i další

specifické knihovny pro jazyk Python například RoboFlow. Díky QR kódům lze jednoduše na základě zakódovaných informací určit, o kterou kotvu se jedná.

V době, kdy je na záznamu z kamery detekován QR kód, je vyhodnocen jeho obsah a posoudí se, zda je ve správném formátu. Zároveň algoritmu poskytuje informaci o pozici detekovaného QR kódu. Informace se reprezentuje nejmenším možným obdélníkem, který opisuje detekovaný QR kód. V prostoru detekované pozice se za pomoci mřížky vytvoří body, které se transformují do 3D. Stejným způsobem uvedeným zde 4.2.1. Takto vytvořené pozice bodů v prostoru se následně uloží do vlastní části paměti. Účelem je zjednodušit práci a filtrování v datech. Point cloud jsou neřazená data a filtrování, nebo jakákoliv práce nad velkým počtem bodů a je výpočetně náročná.

Data QR kódu

Pro ukládání dat byl využit standardní formát JSON. Kód níže demonstruje způsob a typ jednotlivých parametrů ukládaného objektu do textové podoby.

```
{
  m: string,
  a: string,
  o: {
    x: float,
    z: float,
    y: float
  }
}
```

Každé detekované zařízení má vlastní originální MAC adresu (m „mac“), pod kterou se ukládají informace o detekované pozici. V případě, že je MAC adresa duplikátní, je uložen pouze nejnovější záznam s tímto klíčem. Alias (a „alias“) slouží jako jednoznačné pojmenování kotvy a zároveň jednodušeji čitelný identifikátor pro uživatele. Poslední informace uvádí vzdálenost QR kódu od zařízení (o „device offset“). Informace je důležitá v případech, kdy není kotva umístěná na viditelném místě. V základu jsou hodnoty na osách (x, y, z) nastavené na 0. Import do systému Sensmap Visualization vyžaduje sloupec MAC adresy s unikátními záznamy a alias s neprázdnou hodnotou. Pro vytvoření záznamu musí být v QR kódu uvedená hodnota MAC adresy. Alias a definice pozice jsou volitelné. Vzdálenost od podlaží prostor se vypočítá pomocí speciálního QR kódu s definicí řetězce pod MAC adresou s hodnotou „start“.

Kapitola 5

Implementace

Tato kapitola detailně popisuje finální implementace algoritmu, který byl navržený v předchozí kapitole. Uvádí, jak se jednotlivé části řeší na vybraném referenčním zařízení Apple iPhone 12 pro. Uvádí i odlišnosti od původního návrhu, které bylo nutné vyřešit při jednotlivém testování částí. Vzhledem k vybranému typu zařízení bylo vhodné využít jazyk Swift a Metal.

5.1 Zprostředkování ARKit

Jádro celé implementace tvoří třída `ARReceiver`, která získává data o pohybu zařízení, poskytuje data z kamer a analýzu obrazu `ARFrame` z `ARSession`. S touto třídou je spojený protokol `ARSessionDelegate`, jehož úkolem je upozornit program na nově získaná data. Protokol předává objekt `ARFrame`, který obsahuje všechna potřebná data k vyhodnocení a zpracování obrazu. Třída `Renderer` dědí od `ARDataReceiver` a implementuje funkci `onNewARData`, v níž se nastaví aktuálně získaný `ARFrame`.

```
class Renderer : ARDataReceiver {
    private var currentFrame : ARFrame?

    func onNewARData(arFrame: ARFrame) {
        currentFrame = arFrame
    }

    ...
}
```

Předané informace o nových datech do třídy `Renderer` se zároveň zpracují a vykreslí na displej za pomoci protokolu `MTKViewDelegate`. Z protokolu se implementuje funkce `Draw`, která zastřešuje a řeší většinu funkcionality. Funkce předává informace o texturách a detekci grafickému procesoru přes `MTLCommandBuffer`. Výsledek se vykreslí do hlavního prvku uživatelského rozhraní. Komunikaci demonstruje obrázek 3.4.

5.2 Zpracování obrazu

Zpracování `currentFrame` proměnné probíhá ve dvou paralelních krocích. Detekce QR kódů a transformace 2D bodů displeje do 3D virtuálního prostoru.

5.2.1 Detekce QR kódů

Společnost Apple vytvořila vlastní knihovnu pro strojové vidění **Vision**. Knihovna poskytuje funkcionalitu a podporu pro detekci objektů na fotce, nebo záznamu videa. Pro implementaci je především zajímavá třída `VNSequenceRequestHandler`, která umožňuje pro každý obraz v sekvenčním záznamu detekovat požadované objekty z definice. Implementace využívá požadavek `VNDetectBarcodesRequest`, který definuje typ hledaného kódu v obrazu. Inicializace objektu zprostředkování a typ požadavků je na začátku konstruktoru třídy `Renderer`. Symbolika požadavku je nastavená pouze na `.qr`.

```
private func updateCapturedImageTextures(frame: ARFrame) -> Bool {
    let pixelBuffer = frame.capturedImage
    guard CVPixelBufferGetPlaneCount(pixelBuffer) >= 2 else {
        return false
    }

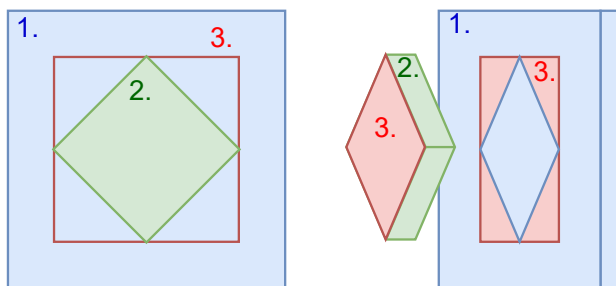
    try? self.sequenceHandler.perform([barcodeRequest], on: pixelBuffer)
    barcodes = barcodeRequest.results;

    ...

    return true
}
```

V případě, že se v průběhu aplikace vyskytnou nová data `ARFrame`, se pro detekci využije `capturedImage`, ze které se přetypováním vytvoří `pixelBuffer`. Nad instancí třídy `VNSequenceRequestHandler` se zavolá metoda `perform()`. Funkci jsou předané dva parametry: požadavek detekce QR kódu se symbolikou `.qr` a `pixelBuffer`, který reprezentuje aktuální obraz snímaný kamerou.

Výsledek z požadavku je uložený pod vlastností `results` zpracovaného požadavku. Data reprezentuje list, který obsahuje všechny důležité informace o detekovaném záznamu. Pro algoritmus je důležitá oblast detekce ve formátu čtyř desetinných čísel v intervalu $(0;1)$ pro pozici (x, y) a šířky, délky obdélníku na displeji. Data se převádí na celočíselnou hodnotu pozice a velikosti v pixelech. V `results` je u každého záznamu zároveň i dekodovaná informace o zakódovaných datech.



Obrázek 5.1: Problém s ohraničením za pomoci nejmenšího obdélníku při detekci.

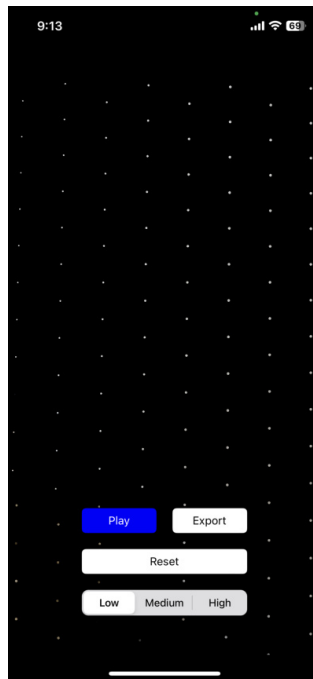
V předchozí části byl detekovaný záznam s informacemi o jeho pozici v 2D prostoru na obrazovce. Reprezentace je standardní ohraničující obdélník se souřadnicemi x a y a

velikostí definovanou výškou a šířkou. Očekávaná pozice je uvnitř tohoto obdélníku. Obrázek 5.1 naznačuje problém, kdy ohraničení obdélníkem nevyužívá informaci o náklonu zařízení. Modrá představuje vzdálenější objekt, zelená detekovaný objekt a červená hraniční obdélník detekce. Červené místo s číslem tři na pravé straně obrázku představuje chybu, která vzniká při propisování bodů, v případě, že se konvertují všechny body spadající do detekované oblasti. Implementace knihovny Vision vždy tvoří nejmenší možný opisný obdélník QR kódu, ve kterém jsou hrany zobrazené paralelně k hranám displeje mobilního telefonu. Při rotaci tak často dochází k záznamu mnoha bodů mimo bod zájmu.

U finální implementace se tak na místo tvoření bodů za pomoci mřížky vypočítá pouze středový bod obdélníku, pro který je velmi vysoká šance, že se nachází na správně pozici. Přístup částečně eliminuje problém vzniku bodů, které nereprezentují správnou pozici zařízení.

5.2.2 Vytvoření point cloud

Kromě bodů pozice kotvy se mapuje i několik zvolených bodů obrazovky pro lepší orientaci uživatele v point cloudu a pro záznam. Obrázek 5.2 ukazuje vytvořenou mřížku, která je následně namapovaná do 3D prostoru. Hustotu záznamu bodů lze upravit.

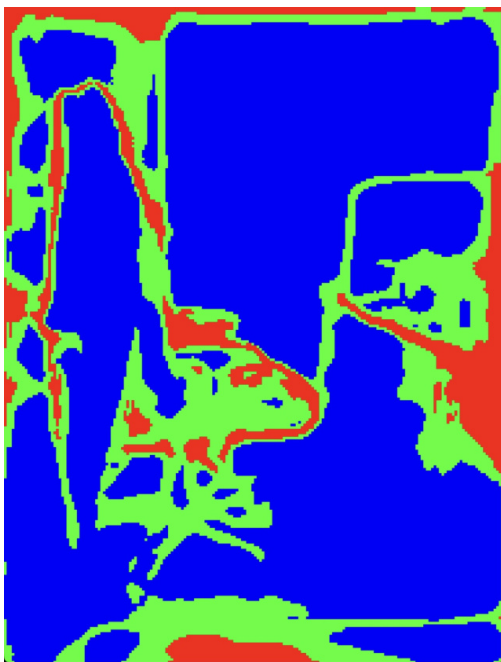


Obrázek 5.2: Vytvořená mřížka bodů na obrazovce

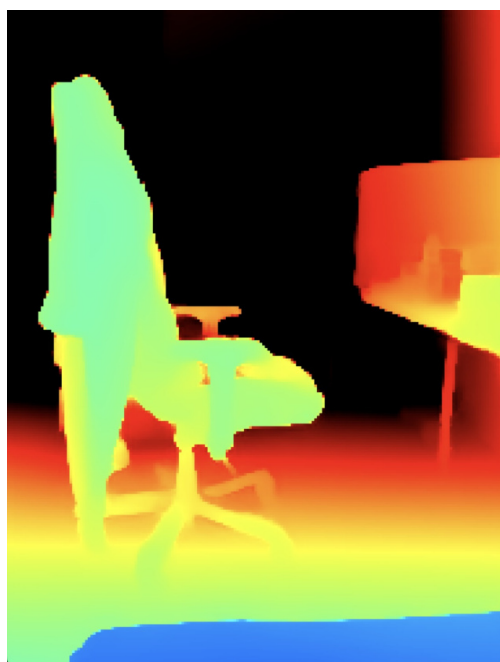
K projekci 2D bodů do 3D se využívá třech typů informací, které poskytuje `ARFrame`. Data o vzdálenosti mobilního telefonu od objektu poskytuje `sceneDepth`, jehož součástí jsou dvě textury. Textura `depthMap` 5.4 reprezentuje zakódované hodnoty vzdálenosti, pro každý pixel ve 32 bitovém plovoucí čárce. Druhá textura `confidenceMap` 5.3 poskytuje informace o kvalitě uvedených vzdáleností v `depthMap`. V sekci 2.6 je uvedené, jaký má vliv povrch materiálu na kvalitu a jistotu čtení s lidar senzory. Hodnoty mapy jistoty jsou uváděné v osmibitovém celém čísle. Používá se pouze $c = \{0, 1, 2\}$. Nejnížší hodnota jistoty je reprezentována 0 a nejvyšší 2.

Aby bylo možné získat hodnoty z příkládaných textur při zpracovávání, je nutné za-
definovat `Sampler`, který na základě pozice v pixelech 2D bodu a definice vlastnosti filtru
dekóduje hodnotu, se kterou dál umožňuje pracovat. Formát výstupní hodnoty `Sampler`
záleží na typu, kterým textura data zakódovává.

Obrázky 5.4 a 5.3 demonstrují barevně reprezentovaná data získaná pomocí `Sampler` z
textur `depthMap` a `confidenceMap`. Vzdálenost je tu reprezentovaná normalizovanou hodno-
tou přechodu v RGB gradientu. Přejít je od modré do červené. Černá barva reprezentuje
hodnoty mimo rozsah podporovaného zařízení. Nejvyšší hodnotu jistoty reprezentuje modrá
(2), zelená (1) a červená (0), nejnižší jistotu. Nižší hodnoty se nachází na okrajích objektů
a na povrchu materiálů s příliš vysokou nebo naopak s příliš nízkou reflexivitou.



Obrázek 5.3: Barevně vykreslená data „confidenceMap“.



Obrázek 5.4: Barevně vykreslená data „depthMap“.

5.2.3 Komunikace mezi CPU a GPU

Způsob přenosu dat mezi procesorem a grafickým procesorem popisují deskriptory. Deskrip-
tor slouží jako předpis způsobu načítání dat a zpracování na grafické kartě. Do deskriptorů
je zapotřebí přidat shader funkce, které jsou definované v souboru `shaders.metal`. Soubor
obsahuje vertex shader funkce pro transformaci 2D bodů do 3D. Dále obsahuje vertex a
fragment shader funkce pro vykreslení záznamu na displej. Pro každou funkci se vstupní
parametry specifikují celým kladným číslem, které reprezentuje index pozice dat ve vyrovná-
vací paměti. Pro každý typ textury existuje záznam ve výčtu hodnot `TextureIndices`.
Oddělené jsou indexy dat vyrovnávací paměti pro seznam bodů `BufferIndices`.

Deskriptory jsou inicializované v konstruktoru `Renderer` třídy. Implementace využívá
pouze třídy `MTLRenderPipelineState`, do kterých je zakódovaný příslušný deskriptor.
Deskriptor tak využívá optimalizace pro vykreslování a zpracování dat pro zobrazení na
displej.

Prvním krokem kódování dat při přenosu je specifikace deskriptoru se správnou vertex a fragment funkcí. Za ním následuje krok, při kterém se na jednotlivě definované indexy v paměti nahrávají data. Posledním krokem je specifikace, jak budou data zpracována. V rámci implementace se vždy jedná o typ primitiva `.point`. Po vytvoření instrukční sady v kodéru se přeloží vytvořený `MTLCommandBuffer` do `MTLCommandQueue`, ze které si instrukce přebírá grafický procesor.

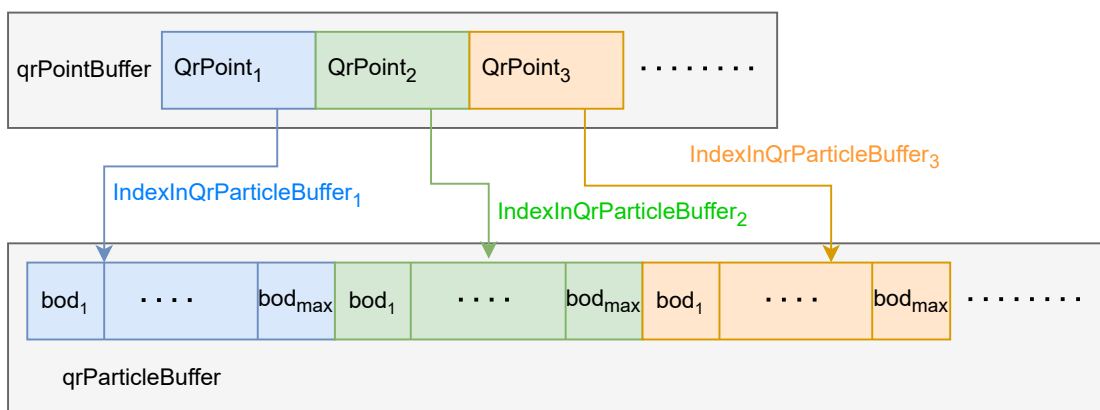
5.2.4 Zpracování dat

Zpracování dat probíhá v několika částech. První a nejdůležitější částí pro účely této práce je transformace 2D bodů displeje na 3D body virtuálního světa. K transformaci dochází v shader funkci `unprojectVertex`. Výstupní body jsou ukládány do paměti `particleBuffer`, ke které má přístup i CPU. Důležitou částí pro výpočet tvoří informace o pozici a stavu mobilního zařízení. Ty reprezentuje struktura `uniforms`, jejíž součástí jsou předané matice popsané v části 4.2.1 a velikost rozlišení obrazu. Vstupní pozice 2D bodů jsou předané přes parametr `gridPoints`.

```
static simd_float4 worldPoint(simd_float2 cameraPoint,
                             float depth,
                             matrix_float3x3 cameraIntrinsicsInversed,
                             matrix_float4x4 localToWorld) {
    const auto localPoint = cameraIntrinsicsInversed *
        simd_float3(cameraPoint, 1) * depth;
    const auto worldPoint = localToWorld * simd_float4(localPoint, 1);

    return worldPoint / worldPoint.w;
}
```

Výše uvedený kód představuje část implementace popsaného algoritmu konverze 2D bodů do 3D v části 4.2.1. Parametr `cameraIntrinsicsInversed` implementuje matici K_{ii} . Parametr `localToWorld` implementuje matici Z_w . Výpočty využívají vektorové operace s typem `simd_float`.



Obrázek 5.5: Způsob ukládání bodů detekce.

Obdobně implementovaná je funkce `qrUnprojectVertex`, která slouží specificky pro transformaci detekovaných bodů kotev. Na vstupu jsou připravena nově detekovaná data

ve formátu `QrPoint`, ve kterém je pozice na displeji uvedena v pixelech, barva a pozice, na kterou se nově vytvořený bod uloží `IndexInQrParticleBuffer`. Hodnota pozice bodů v `qrParticleBuffer` je vypočtená z odsazení na základě indexu a počtu detekovaných bodů pro QR kód. Interakce naznačuje obrázek 5.5. K výpočtu pozice indexů nového bodu a barvy dochází na CPU. Předchází se podmínkám v shader funkcích, které výrazně zpomalují rychlost zpracování. Červeně se označují detekované body zařízení a zelená barva naznačuje startovní bod, od kterého se počítají vzdálenosti.

5.2.5 Obarvení bodů

Kamera mobilního telefonu poskytuje dvě textury, které obsahují informaci o barvě. Využívá kódování Y a CbCr, proto je zapotřebí transformovat tyto informace do RGB formátu, který lze následně vykreslit na displej. Výpočet RGB hodnot z YCbCr formátu znázorňuje vzorec. Hodnoty 1 jsou reprezentace alfa kanálu průhlednosti.

$$\begin{bmatrix} R \\ G \\ B \\ 1,0000 \end{bmatrix} = \begin{bmatrix} +1,0000 & +1,0000 & +1,0000 & +0,0000 \\ +0,0000 & -0,3441 & +1,7720 & +0,0000 \\ +1,4020 & -0,7141 & +0,0000 & +0,0000 \\ -0,7010 & +0,5291 & -0,8860 & +1,0000 \end{bmatrix} * \begin{bmatrix} Y \\ C_b \\ C_r \\ 1,0000 \end{bmatrix}$$

Matice pro převod je uvedená v dokumentaci [5].

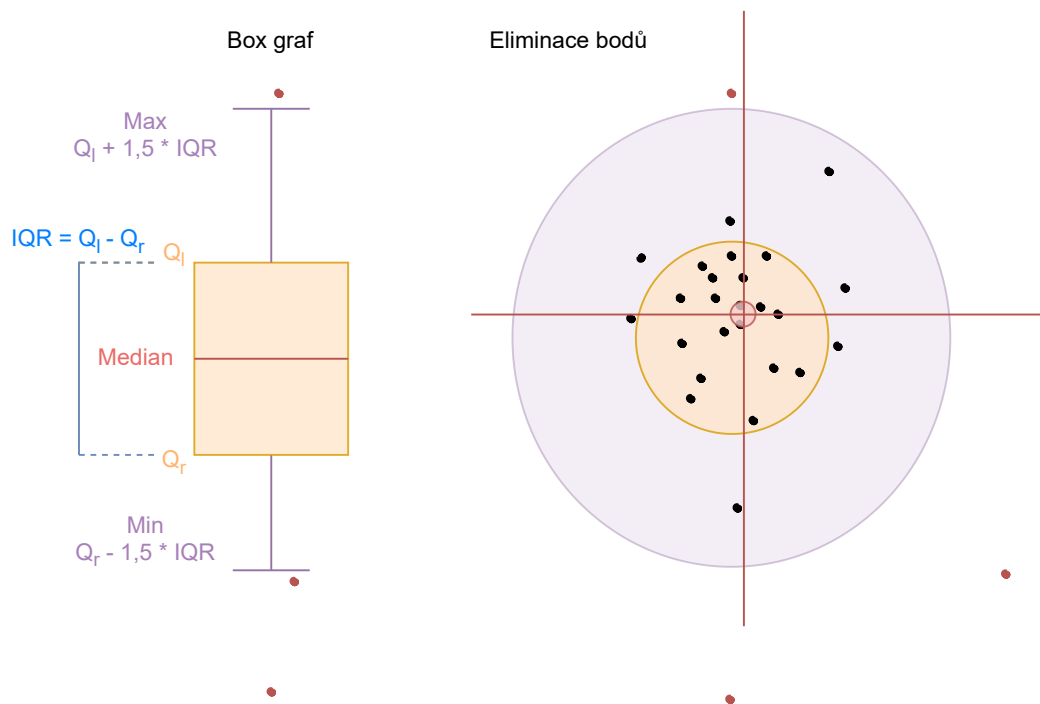
5.2.6 Vykreslení dat na displej

Ke zlepšení orientace a zpětné vazbě uživateli o zaznamenaných informacích napomáhá vizualizace dat v reálném čase na displej. Vykreslení probíhá za pomoci dvou shader funkcí. Vertex funkce pracuje s parametrem transformovaných 3D bodů `particleBuffer`. Zpětně se transformuje 3D bod na 2D pozici displeje. Vykreslování bodů je omezené spodní hranicí 0,3 metrů od zařízení a 8 metrů pro nejvzdálenější bod. Transformovaný bod s 2D souřadnicemi a barvou předává fragment funkci. Fragment funkce účelně eliminuje průhledné body a body, které jsou na hraně pixelu, za pomoci funkce `discard_fragment()`. Pro ostatní body se navrácí barva příslušného bodu na pozici displeje.

5.3 Filtr dat

Při testování implementace části detekování bodů algoritmu nastával problém, že při detekci pod větším úhlem byly zaznamenané body zasazené na špatnou pozici, protože střed detekovaného místa byl pod náklonem mimo pozici, nebo na kraji detekované kotvy. Zároveň pro případy, kdy došlo k detekci na okraji rozsahu lidar senzoru. Jak lze ale vidět u obrázku 5.3, data na okrajích displeje a objektu nemají vysoké hodnoty. Původně se z principu ukládala pouze poslední detekovaná hodnota, jenž reprezentovala pozici kotvy. Bylo tak nutné vytvořit alternativní způsob pro eliminaci hodnot na základě pravidel.

Pro účely této práce bylo vhodné využít techniku aplikovanou při vytváření krabicových grafů. Principem je seřadit data od nejmenšího po největší a vzít dvě střední hodnoty a vypočítat medián. Následně obdobně vypočítat hodnoty Q_r a Q_l , které reprezentují medián dat na pravé a levé straně od středového mediánu. Na základě hodnot Q_r a Q_l následně vybrat pouze nejlepší kandidáty v intervalu od Q_r do Q_l a vypočítat průměr.



Obrázek 5.6: Způsob eliminace 3D bodů za pomoci box grafů.

Obrázek 5.6 demonstruje využití obdobné logiky pro eliminaci vyčnívajících bodů od ostatních. Vstupní hodnotou do filtru je vzdálenost bodu od počátečního bodu záznamu.

5.4 Export dat

Export je možné provádět v jakoukoliv dobu průběhu. V základu se exportují pouze nalezené body zařízení, hlavně z důvodu velikosti dat při delším záznamu. Lze ale zapnout export bodů prostředí, při kterém dojde k exportu ve formátu xyz 2.5.1 s RGB barvami.

Při exportu dochází k nalezení startovního bodu uloženého pod klíčem „start“ a k zjištění jeho pozice. Pro každé další uložené zařízení se od startovních pozic vypočítá vzdálenost na každé ose.

$$p_e = \begin{pmatrix} (x_a + x_{oa}) - x_s \\ (y_a + y_{oa}) - y_s \\ (z_a + z_{oa}) - z_s \end{pmatrix}$$

p_e je výsledná pozice exportovaného bodu. $s = (x_s, y_s, z_s)$, jsou souřadnice startovního bodu a $a = (x_a, y_a, z_a)$, jsou souřadnice pozice zařízení. Hodnota $o_a = (x_{oa}, y_{oa}, z_{oa})$ je v definovaná odchylka QR kódu od zařízení. Využívá se u zařízení, ke kterým není přímý přístup, anebo jsou blokována a přímý záznam tak není možný. U chybějících je automaticky hodnota nastavena na $o_a = (0, 0, 0)$.

Kapitola 6

Experimenty a testování

V této kapitole se práce věnuje vyhodnocení algoritmu v různých prostředích. Na začátku kapitoly se text věnuje hlavně popisu chování detekce. Navazuje na ní část o zhodnocení vlivu kvality dat na základě nastavení zařízení. Hlavní částí je popis výsledků získaných dat při experimentování v prostorách instalace RTLS systému Sewio Network s. r. o. ve firmě Intemac v Kuřimi.

6.1 Chování detekce

Velmi důležitou částí pro správný záznam dat je správně detekované místo pozice kódu na displeji. V průběhu implementace se při testování povedlo odhalit několik základních problémů a následně algoritmus upravit. Stále je zde několik problematických částí.

6.1.1 Velikost QR kódu

Jeden z problémů, které algoritmus neřeší, jsou menší velikosti QR kódů při záznamu. Vzhledem k návrhu je tak důležité dodržet dostatečnou velikost QR kódu, který je připevněný k zařízení. Tabulka 6.1 představuje referenční data. Vzdálenost reprezentuje první zaznamenanou hodnotu detekce při přímém pohledu na QR kód.

Velikost QR kódu [cm]	Vzdálenost [m]
8	1,27
10	1,72
15	2,68
20	5,19

Tabulka 6.1: Vliv velikosti QR kódu na detekovatelnou vzdálenost.

K tomu, aby bylo možné zaznamenat data na větší vzdálenosti a zároveň nedocházelo k negativním vlivům na získaná data, je vhodné využívat QR kódů alespoň o velikosti 15 centimetrů na výšku i na šířku. Ideální velikostí je 20 centimetrů a větší, při které se vzdálenost detekování pohybuje na limitu lidar senzoru mobilního telefonu. Předpokládá se, že prostory jsou dostatečně osvětlené. Při sníženém kontrastu (na špatně osvětlených místech) efektivita detekování oproti výchozím hodnotám klesá, takže se v tmavé místnosti vzdálenosti rozpoznání zkrátí až na polovinu. Při testování byl vždy použitý standardní

kancelářský papír A4 formátu, na který byl kód vytištěn. Papír zároveň vytvořil plochu, na které lidar senzor mobilního telefonu přesněji detekoval vzdálenosti. Díky okrajům papíru tak bylo dosaženo lepších výsledků.

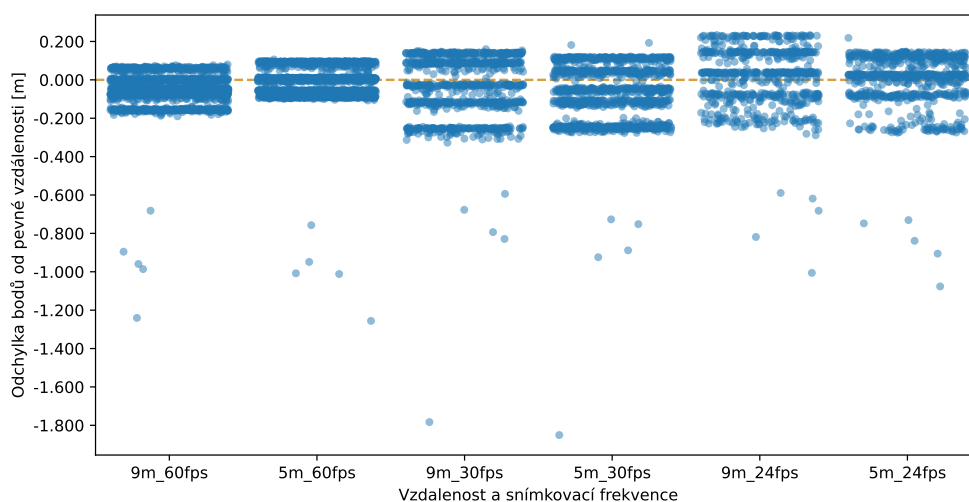
U větších QR kódů je zároveň možné rozpoznat z většího úhlu na větší vzdálenosti. Pro zařízení umístěné zhruba 6 metrů nad zemí je vhodné využít o několik centimetrů větší rozměry QR kódu. Při testování neměl úhel pohledu při vychýlení až o 45 stupňů od kolmého pohledu vliv na vzdálenost možné detekce. U větších úhlů už dochází k redukovaní vzdálenosti, na kterou jsou schopné QR kód rozpoznat.

6.2 Kvalita výsledných dat

Kvalita výstupního záznamu hodně závisí na prostředí a způsobu záznamu. V této sekci jsou popsány limitace prostředí na kvalitu záznamu. Největší problém při záznamu způsobuje pohyb člověka v prostoru. Zařízení s pomocí pohybových senzorů se snaží mapovat pohyb do virtuálního světa, ve kterém se data ukládají. Většina chyb pochází právě z nepřesnosti přenosu mezi reálným pohybem a pohybem ve virtuálním světě.

6.2.1 Mapování prostor

Vliv pohybu uživatele při záznamu má zásadní dopad na kvalitu dat. Směr, který uživatel zvolí při mapování sektorů, je důležitý. Částečně tento problém lze eliminovat využitím stabilizace telefonu. Chyba, která při záznamu rozsáhlých prostor vzniká, je částečně náhodná a nedá se tak eliminovat bez pomoci externích kalibrací. Pohyby, jako třes ruky nebo houpaní se zařízením při chůzi, zavádí při každém vzorkování chybu k nově vytvořeným bodům v point cloud datech. Data zaznamenaná v kratším rozestupu nejsou zásadně ovlivněná na rozdíl od porovnání prvního detekovaného bodu s posledním, kde na základě délky a způsobu pohybu může být zásadní chyba v rozdílu s reálnou pozicí.



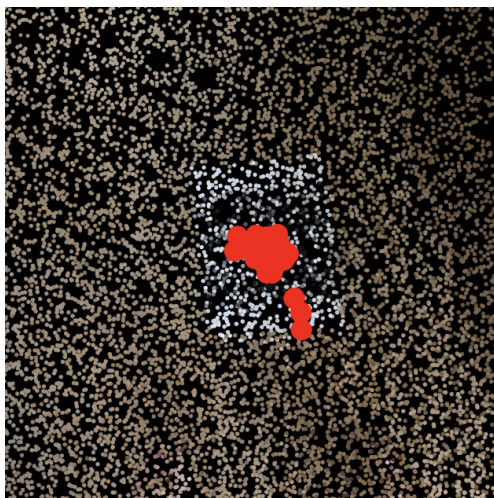
Obrázek 6.1: Vliv snímkovací frekvence na kvalitu dat

V průběhu testování byly využité různé rychlosti snímkování za účelem zjistit, zda je možné pomalejším záznamem částečně tento problém eliminovat. V průběhu testů byly testované hodnoty 24, 30 a 60 snímků za sekundu.

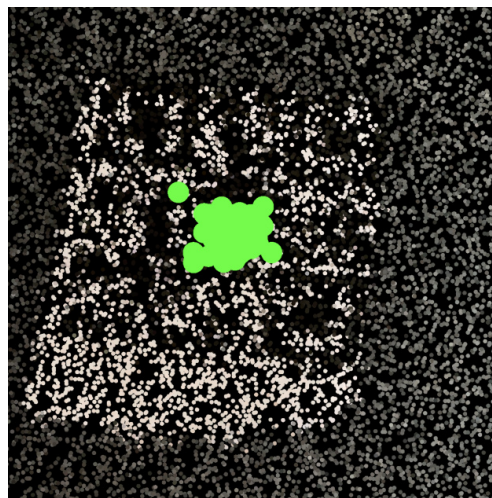
Graf 6.1 představuje souhrn dat, které jsou získány skenováním dvou QR kódů, které jsou od sebe vzdálené 5 a 9 metrů. Při záznamu bylo snahou zreplikovat standardní pohyb uživatele při mapování prostředí pohledu do prostoru. Zaznamenané body tak vznikaly při chůzi uživatele. Hodnota na ose y tak reprezentovala rozptyl bodů pro jednotlivé snímkovací frekvence. Z dat lze zhodnotit, že 60 snímků za sekundu je nejstabilnější při pořizování dat. Při nižších frekvencích je zařízení náchylnější na malé pohyby. Vznikají tak prodlevy od získání pozice dat zařízení a snímkem z kamery. Body, které velmi patrně nezapadají do žádné skupiny, jsou velmi často efektem rychlého pohybu mobilním telefonem při pořizování dat. Za účelem eliminování takto vzniklých bodů se využívá implementovaný filtr (5.3). Pomalejším pohybem je možné omezit vliv špatně zaznamenaných bodů.

Filtr vyžaduje pro správnou funkci minimálně 10 získaných bodů, maximální počet bodů pro jeden QR kód je 500. Uživatel má určitým způsobem kontrolu nad kvalitou dat. Nejlepším způsobem záznamu je pozastavení a zaměření kamery mobilního telefonu v přímém úhlu s viditelností na kód. Detekčnímu algoritmu se v kontextu více snímků podaří správně zaznamenat existenci kódu na kameře.

Další vliv na výslednou vzdálenost má správně detekovaný startovní bod, který slouží k vyhodnocení výškové pozice od podlahy místnosti. Na rozdíly mezi skupinami bodů v grafu 6.1 má zásadní vliv kvalita bodů zaznamenaných pro startovní pozici.



Obrázek 6.2: Špatný rozptyl bodů QR kódu.

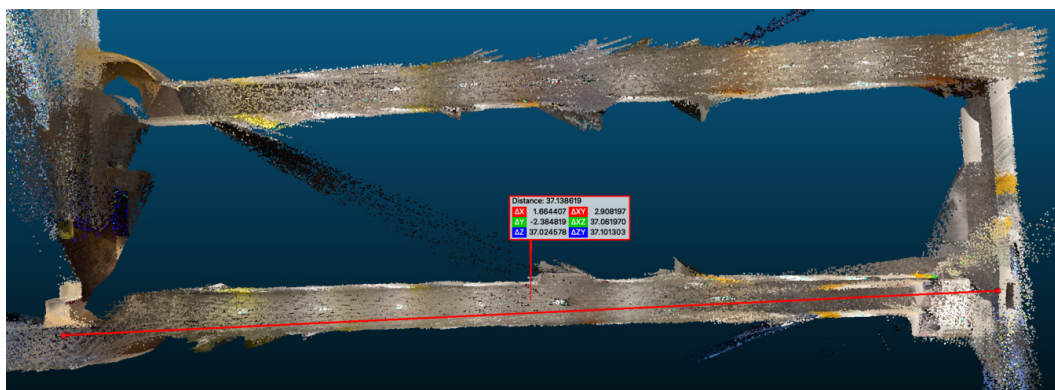


Obrázek 6.3: Dobrý rozptyl bodů QR kódu

Obrázek 6.2 znázorňuje jeden z problémů, který vzniká při záznamu probíhajícího při chůzi uživatele. Body, které jsou viditelně oddělené od zbytku, jsou prvními získanými body. Ty jsou detekované z větší vzdálenosti a kvůli tomu vzniká chyba od reálné pozice nepřesností lidar senzoru. Obrázek 6.3 naopak ukazuje případ, jak zhruba vypadá standardní rozptyl při zachycení dat. Obecně rozptyl bodů vzniká při mírném náklonu, nebo detekci z většího úhlu pohledu. Vzhledem k rozdílu vzdáleností krajních bodů mezi nimi vzniká odchylka až 5 centimetrů.

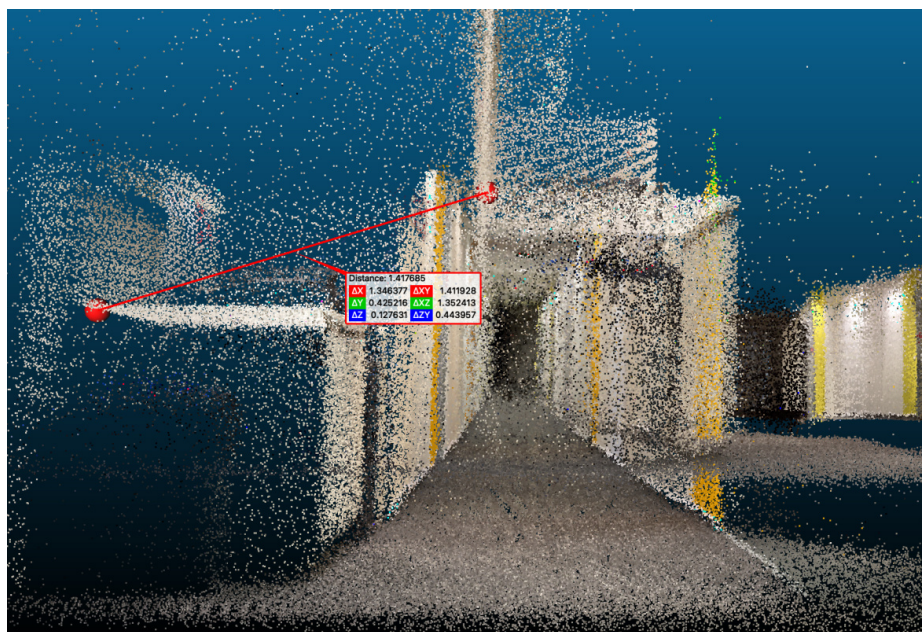
6.2.2 Mapování větších prostor

Při záznamu se na vzdálenostech do 10 metrů příliš neprojevuje akumulovaná chyba pohybových senzorů zařízení. Nejpatrnější efekt nepřesného mapování se projeví při záznamu větších prostorů, v nichž uživatel obchází dokola po obvodu místnosti. Při průchodu se postupně eliminuje chyba a při uzavření okruhu jsou body ve stejném místě zaznamenány s výrazným driftem. Vnitřní implementace ARSession knihovny ARKit zapříčiní, že při návratu do startovního místa se tato chyba za 1 sekundu zresetuje.



Obrázek 6.4: Referenční obrázek velikosti záznamu.

Během testování využívala pouze data zaznamenaná z prostředí. Prostřednictvím většího počtu bodů se chyba vizuálně prokáže odchylkou bodů po projití dokola na stejném místě. K testování byl využit průchozí koridor dlouhý 39 metrů, který je spojován uličkami o délce 10 metrů. Při záznamu dat byl mobilní telefon vždy natočený ve směru chůze. Celkový záznam pokrývá průchod prostorem o délce 98 metrů. Cílem bylo získat představu o tom, jak na sebe data ve stejném bodě navazují. Tvar koridoru je naznačuje 6.4.

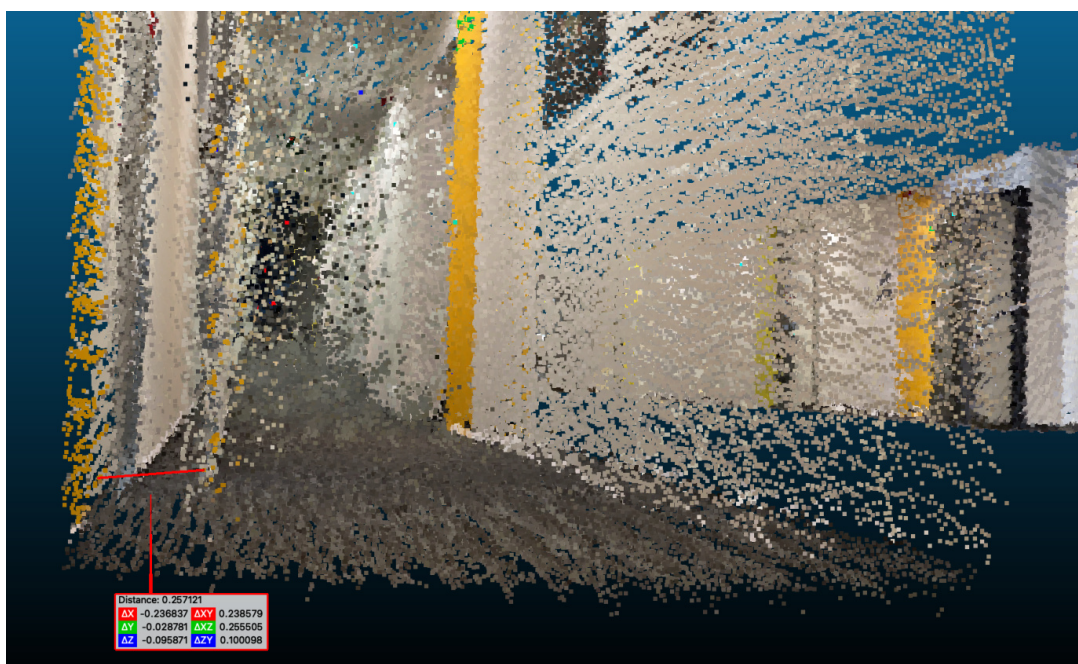


Obrázek 6.5: Velikost naakumulované chyby.

Jak demonstruje obrázek 6.5, dochází zde k deformaci v průběhu záznamu o zhruba 141 centimetrů. Záznam byl pořízen při normální chůzi přibližně 4 kilometry za sekundu a mobil nebyl podporovaný žádnou stabilizací. V příložených datech práce jsou další pořízené záznamy, kde chyba dvou bodů reprezentující stejné místo se pohybuje od 20 centimetrů do zmiňovaných 141 centimetrů.

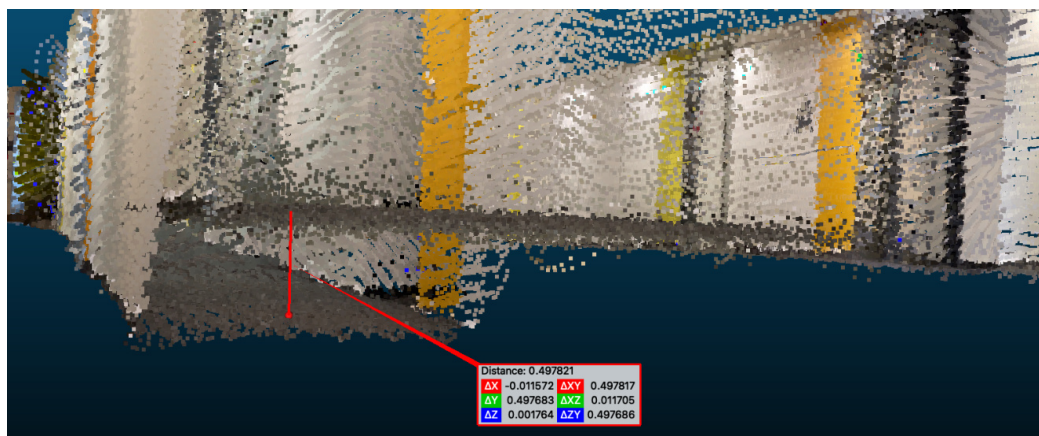
Při každém průchodu byla vždy vychýlená nejvíce osa x (horizontální osa). Chyba se pokaždé vytvořila v protisměru záznamu. Prostor je tak při zpáteční cestě halou na druhou stranu vždy vychýlený vlivem otáčení telefonu. Pomalejší a plynulejší pohyb při otáčení umožňuje získat lepší výsledná data. Chyba při standardní chůzi zapříčiní odchylku bodů na ose y (vertikální osa).

Tento problém redukuje využití nějaké formy stabilizace telefonu. Pro testování byl použit tříosový stabilizátor od firmy Hohem „iSteady Mobile +“. Obrázek 6.6 zobrazuje data zaznamenaná ve stejném prostředí za pomoci tříosové stabilizace, jež napomáhá eliminovat vliv třesu ruky při chůzi. Částečně redukuje i chyby při otáčení v prostoru. Zde záleží na nastavení stabilizace. Pro zařízení je tak pohyb plynulejší. Ve stejném prostředí se při stejných podmínkách chyba podařila zredukovat na zhruba 25 centimetrů. Nejčastější chyba se projevila hlavně na ose y (vertikální osa), kde je stabilizace omezená. Výrazně stabilizátor napomáhá v omezení deformace virtuálního záznamu a při opakování záznamů lze získat podobnější výsledky.



Obrázek 6.6: Záznam za pomoci stabilizace zařízení

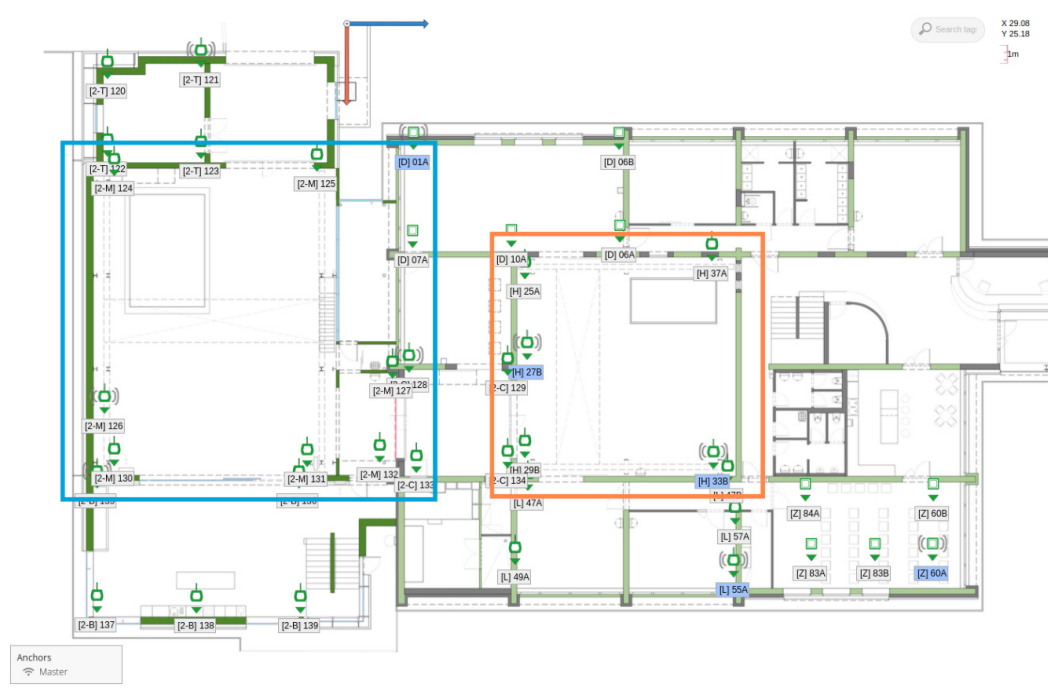
Výslednou chybu s využitím stabilizace naznačují obrázky 6.6 a 6.7. Oproti obrázku 6.5 se podstatně snížila chyba, která při záznamu vznikla.



Obrázek 6.7: Záznam za pomoci stabilizace zařízení.

6.3 Intemac

V prostředí firmy Intemac v Kuřimi, pro kterou firma Sewio Networks s. r. o. poskytuje svůj poziční systém, probíhalo testování programu. Prostory, ve kterých došlo k testování, jsou označené na obrázku 6.8 modrým a oranžovým obdélníkem. Obrázek byl pořízen v systému Sensmap Visualizaion pro referenci pozic kotev při testování. Testování probíhalo nad instalací skupin zařízení označené „H“ a „2-M“. Sektory jsou na plánu označené barevnými obdélníky. Zařízení s označením „H“ se nachází v oranžové části a „2-M“ v modré.



Obrázek 6.8: Plán podlaží firmy Intemac v Kuřimi s instalací RTLS firmy Sewio Networks s. r. o.

Skupina „H“ se nachází v prostorách o velikosti zhruba 15 metrů na šířku, 20 metrů na délku a 10 metrů na výšku. Ve skupině je 5 kotev, kde 4 z nich jsou instalované do rohů místnosti s rozdílem kotvy „27B“. Ta je připevněná uprostřed mezi kotvy „25A“ a „29B“. Kotva „27B“ slouží jako hlavní kotva pro celou skupinu.

Obdobně pak skupina „2-M“ je instalovaná do místnosti o zhruba 10 metrů větší. Ve skupině je celkem 7 kotev. Způsob instalace je podobný jako u předchozí skupiny. 5 zařízení je umístěných do rohů místnosti a zařízení „126“ a „131“ jsou umístěné na místo s dohledem na další skupinu zařízení.

Referenční data instalace poskytnutá firmou Sewio Networks s. r. o. uvádí instalovanou výšku 3 metry od podlahy. Pro souřadnicová data je referenčním bodem modročervené naznačení osy x a y na horní straně obrázku 6.8.

MAC adresa	Alias	x [m]	y [m]	z [m]
0xE8EB1B3BBF29	[H] 25A	9,53004	13,71259	3
0xE8EB1B3C5D86	[H] 37A	19,51574	12,73528	3
0xE8EB1B3C4C28	[H] 27B	9,65397	18,01489	3
0xE8EB1B3BAD4A	[H] 29B	9,53711	23,25298	3
0xE8EB1B3BB8E3	[H] 33B	19,59633	23,87351	3
0x049162C98AA9	[2-M] 131	-2,14278	23,72665	2,84
0x049162C9487E	[2-M] 127	2,42846	19,0271	2,76
0x049162C999F9	[2-M] 130	-12,45619	23,71249	2,84
0x049162C98AD7	[2-M] 125	-1,59996	7,94208	2,86
0x049162C90C4F	[2-M] 124	-12,44192	8,18514	2,85
0x049162C90ABE	[2-M] 126	-12,95618	20,8837	2,84
0x049162C98292	[2-M] 132	1,74281	23,49804	2,83

Tabulka 6.2: Referenční hodnoty souřadnicových pozic (x, y, z) kotev od vyznačeného bodu.

Tabulka 6.2 uvádí referenční hodnoty pozic zařízení, které jsou v systému uvedené. Hodnoty pozice jsou uvedené vůči pevnému bodu v plánu v horní části obrázku 6.8. Souřadnice bodů dat bylo pro přehlednější porovnávání vhodné přepočítat. K tomu byla vždy zvolena jedna z kotev ve skupině, od které jsou odečtené vzdálenosti ostatních. Pro testy skupiny „H“ bylo zvolené zařízení „29B“, pod kterým je umístěná výchozí pozice skupiny. U skupiny „2-M“ je stejným způsobem zvolené zařízení „127“.

6.3.1 Způsob připevnění kódů

Prvním krokem při testování bylo připevnění QR kódů ke kotvám. Všechny kotvy v sektoru „H“ a „2-M“ jsou typu Vista Omni. Anténa zařízení tak funguje jako středový bod. Anténa zároveň slouží jako závěsné místo, kde se za vystřiženou díru nad vytištěným QR kódem papír pověsí. Střed QR kódu pro velikost 20 centimetrů pasuje na střed krabičky zařízení. Papír by měl být dostatečně ohnutý tak, aby neodstával od zařízení. Problematické byly pouze kotvy, které byly instalované anténou směrem dolů. U těchto zařízení instalace vyžadovala přidržení lepicí páskou. Obdobného způsobu připevnění je možné využít i pro Vista DirectFive.

6.3.2 Způsob zvolení referenčního bodu

K tomu, aby algoritmus správně fungoval, je nutné někde do prostoru umístit QR kód reprezentující startovní pozici. V případě prostorů skupiny „H“ byla startovní pozice umísťována do rohu pod kotvu „29B“. U skupiny „2-M“ byla startovní pozice zvolena pod zařízením „127“.

Referenční hodnoty z tabulky 6.2 počítají s případem ideálně rovné podlahy, proto je u osy z pro sektor „H“ vždy uvedená hodnota 3. Reálné vzdálenosti od podlahy při přeměření se pohybují mezi 2,9 až 3 metry. Umístění počátečního bodu v prostorech „H“ ukazuje obrázek 6.9. Stejně tak pro část „2-M“ obrázek 6.10



Obrázek 6.9: Startovní pozice QR kódu pro „H“



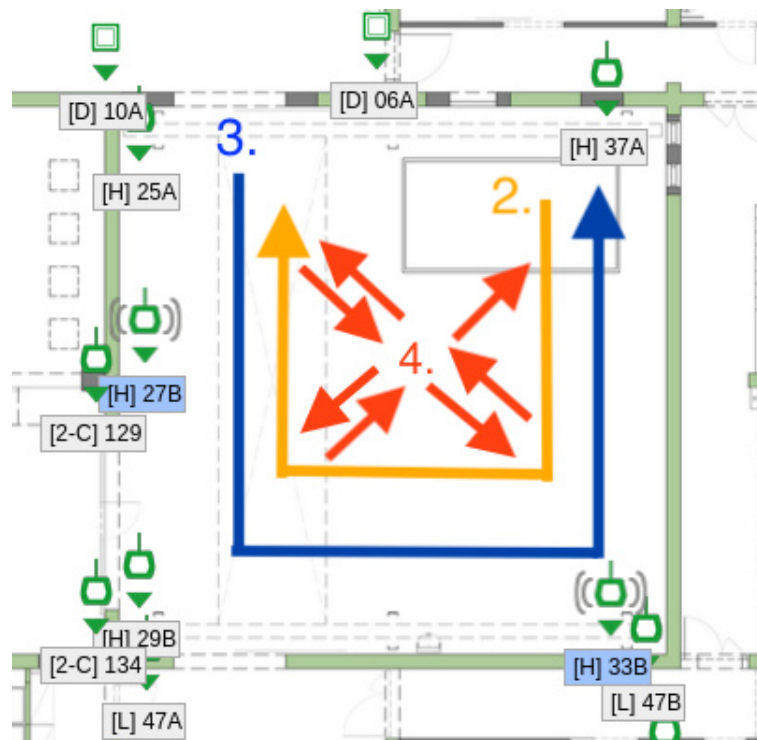
Obrázek 6.10: Startovní pozice QR kódu pro „2-M“

6.3.3 Prostory „H“

Šipky na obrázku 6.11 naznačují přibližný pohyb v místnosti. Čísla určují, která z tabulek reprezentuje data jakého záznamu. První tabulka obsahuje přepočtené hodnoty pozic z referenční tabulky 6.2.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[H] 25A	-0,007	-9,540	0	9,540
[H] 37A	9,978	-10,517	0	14,498
[H] 27B	0,116	-5,238	0	5,239
[H] 29B	0	0	0	0
[H] 33B	10,059	0,620	0	10,078

Tabulka 6.3: Přepočítané referenční hodnoty s odchylkou kotev na osách (x, y, z) od kotvy „29B“.



Obrázek 6.11: Naznačení pohybu pro záznamy skupiny „H“.

Tabulka hodnot 6.3 představuje rozdíl vzdáleností od zařízení „29B“. Referenční hodnoty dle měření z místa nerepresentují reálné vzdálenosti. Předpokládá se, že zdi a podlaha jsou ideálně rovné. Chyba nastává při ručním měření a chybný je i předpoklad, že jsou zařízení ideálně v rovině. Hodnoty referenční tabulky tak nerepresentují reálné hodnoty a nově získané hodnoty z proběhlých záznamů jsou přesnější.

Tabulka 6.4 představuje získané hodnoty pozic zařízení a vzdáleností od zařízení „29B“. Při záznamu se vždy zaznamená počáteční pozice mobilního telefonu. Zavádí se zde mezi záznamy různě natočené point cloud dat. Pro přehlednost porovnání jsou původní získané hodnoty natočené o úhel $18,401^\circ$ na rovině xy kolem bodu $p_o = (0,0)$. Při exportování dat do pozic Sewio systému je důležité vyměnit osy následujícím způsobem. V souřadnicích mobilního zařízení se využívá osa x a y pro pozici na displeji. Osa z představuje hloubku záznamu. Pro pozice v systému Sewio tak osa z bodu představuje osu x v systému Sewio. Osa x bodu představuje y v systému Sewio. Osa y bodu představuje osu z v systému Sewio.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[H] 25A	0,261	-9,697	-0,094	9,701
[H] 37A	10,202	-10,751	0,125	14,821
[H] 27B	0,278	-5,195	-0,096	5,204
[H] 29B	0	0	0	0
[H] 33B	10,230	0,621	-0,078	10,249

Tabulka 6.4: Přepočítané hodnoty pozic ze záznamu označeném číslem 2 s odchylkou kotev na osách (x, y, z) od kotvy „29B“.

Největší rozdíl v hodnotách je na první pohled vidět u výškového rozdílu kotev. Největší rozdíl vzdálenosti je u kotvy „37A“. Kotva je umístěná nejdál od zařízení, které slouží jako střed. Reálná vzdálenost měřená laserovým metrem je 14,63 metrů. Pro správné fungování systému by neměla chyba pozice kotvy přesahovat 50 centimetrů. Hodnoty tabulky se průměrně pohybují v rámci tolerované chyby.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[H] 25A	-0,007	-9,955	-0,001	9,955
[H] 37A	9,72	-10,983	0,052	14,666
[H] 27B	-0,138	-5,545	0,059	5,547
[H] 29B	0	0	0	0
[H] 33B	10,064	0,291	-0,108	10,069

Tabulka 6.5: Přepočítané hodnoty pozic ze záznamu označeném číslem 3 s odchylkou kotev na osách (x, y, z) od kotvy „29B“.

Obdobně transformovaná jsou data tabulky 6.5. Korekce úhlu o $-68,552^\circ$ v rovině xy . Na první pohled se hodnoty kotev „25A“ a „27B“ od předešlého záznamu liší o zhruba 20 centimetrů. Důvodem chyby je příliš krátká doba záznamu bodů pozice zařízení, při kterém došlo k zprůměrování hodnot umístěných mimo kotvu. Směr pohybu při záznamu naznačuje číslo 3 na obrázku 6.11. Hodnoty u záznamu se pohybují na hraně tolerované hodnoty odchylky bodů, ale nepřesahují. Při reálném využití je možné zabránit takovým případům, při kterých se zachycení pozice kotvy uživatel pozastaví na sekundu. Tím dojde k záznamu dostatečného množství bodů. Rychlost a plynulost pohybu významně ovlivní funkcionalitu detekování QR kódu ze záznamu, díky implementaci sekvenčního zpracování požadavků se využívá kontextu několika po sobě jdoucích snímků. Návaznost mezi nimi tak ulehčuje a urychluje detekci.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[H] 25A	-0,282	-9,733	-0,079	9,738
[H] 37A	8,421	-11,933	0,102	14,605
[H] 29B	0	0	0	0
[H] 33B	10,07	-0,831	-0,058	10,104

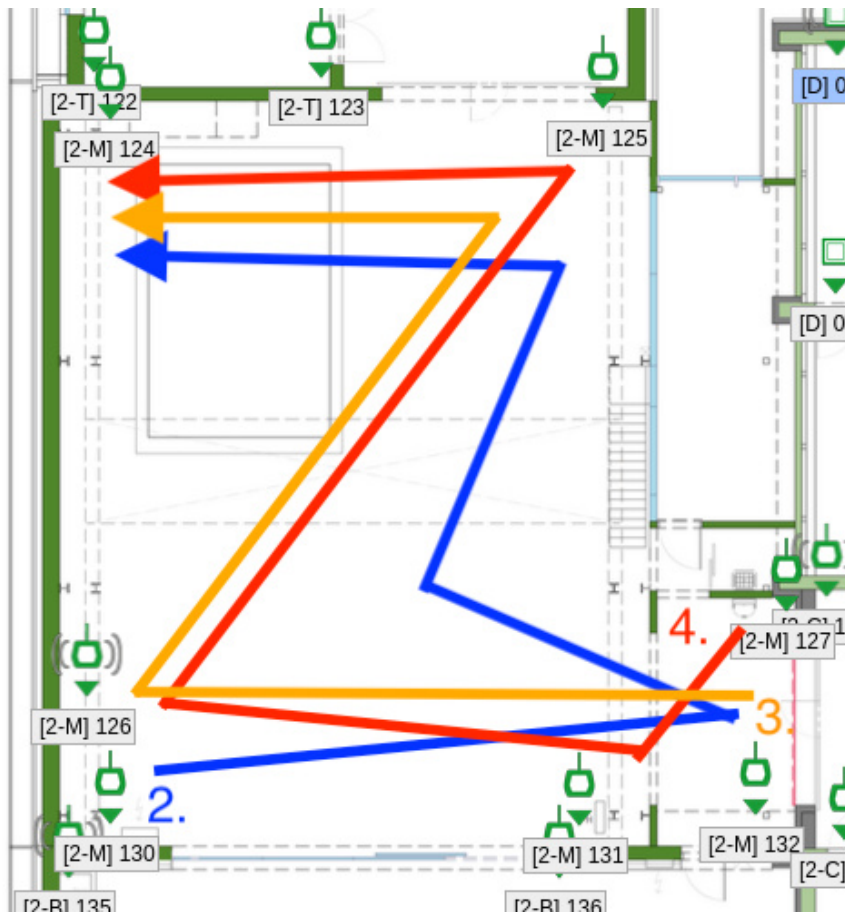
Tabulka 6.6: Přepočítané hodnoty pozic ze záznamu označeném číslem 4 s odchylkou kotev na osách (x, y, z) od kotvy „29B“.

Nejstabilnější výsledky tak poskytuje způsob, který je naznačen číslem 4. Při přechodu přes středový bod místnosti s krátkým pozastavením se vždy zresetuje chyba, které zařízení nasčítalo. V tabulce není uvedena kotva „27B“, protože se při záznamu neobjevila ani jednou na kameře.

Rozdíly mezi záznamy jsou způsobené problémy, které jsou popsány v předchozí části textu. Mezi záznamy se hodnoty pohybují v rozmezí $(-10; 10)$. V průběhu záznamů nebyla využita žádná stabilizace. Za účelem získání dat je vhodné provést více individuálních záznamů nad určitou částí a vytvořit několik alternativních datasetů. Následně vyhodnotit rozdíly dat a naimportovat do systému. V průměru doba nutná pro vytvoření záznamu nepřesahovala 2 minuty. Uživatel má možnost rychle opakovat záznam vícekrát a získat větší množství dat.

6.3.4 Skupina „2-M“

Obdobně jako u skupiny „H“, jsou data zpracována i v této části. Kotva „132“ byla vybrána jako středový bod. Úhel náklonu dat byl vypočtený podle kotvy „126“. Tabulka 6.7 obsahuje referenční data, která pochází z tabulky v první sekci 6.2.



Obrázek 6.12: Naznačení pohybu pro záznamy skupiny „2-M“.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[2-M] 126	-14,699	-2,614	0,01	14,93
[2-M] 124	-14,185	-15,313	0,02	20,873
[2-M] 127	0,686	-4,471	-0,07	4,524
[2-M] 132	0	0	0	0
[2-M] 131	-3,886	0,229	0,01	3,892
[2-M] 125	-3,343	-15,556	0,03	15,911
[2-M] 130	-14,199	0,214	0,01	14,201

Tabulka 6.7: Přepočítané referenční hodnoty s odchylkou kotev na osách (x, y, z) od kotvy „132“.

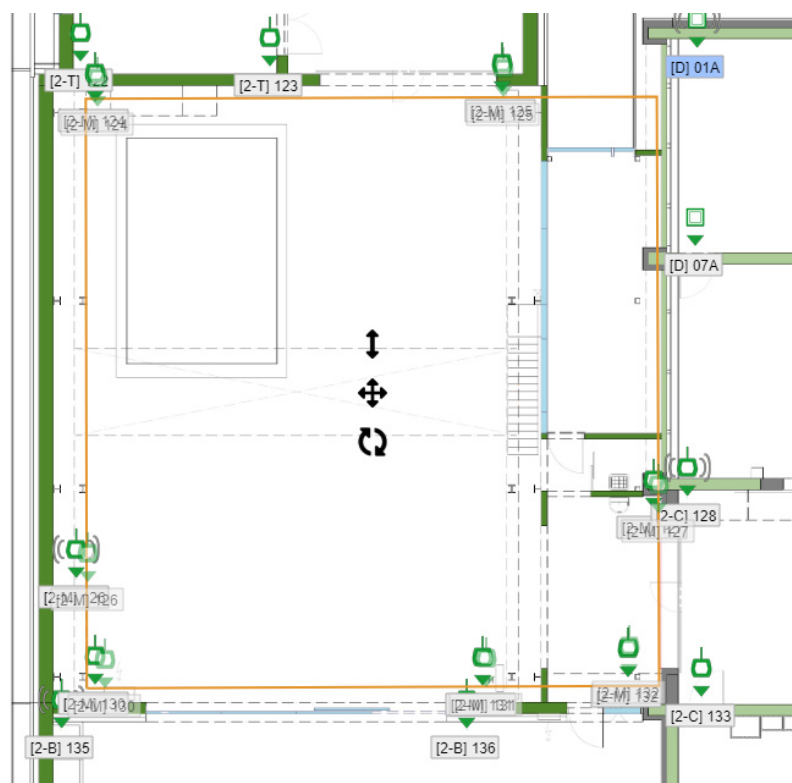
Oproti předchozí skupině jsou poskytnuté referenční hodnoty pozic kotev přesnější. Uváděná data se liší nejvíce o 2,7 centimetrů. Tabulka hodnot 6.8 je na rovině xy rotována o

162, 14°. Hodnoty vzdáleností se od referenčních hodnot liší u každé kotvy o přibližně 15cm. Největší odchylka se projevila u zařízení „130“, kde je oproti referenčním hodnotám vzdálenost o 27,5 centimetrů kratší. Chyba je způsobená malým počtem zaznamenaných bodů pro kotvu a nepřesně zaznamenané body tak mají výrazný vliv na finální hodnotu.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[2-M] 126	-14,428	-2,566	-0,018	14,655
[2-M] 124	-14,053	-15,242	0,047	20,732
[2-M] 127	0,812	-4,275	-0,066	4,352
[2-M] 132	0	0	0	0
[2-M] 131	-3,803	0,278	0,037	3,813
[2-M] 125	-3,243	-15,425	0,019	15,763
[2-M] 130	-13,974	0,243	-0,011	13,976

Tabulka 6.8: Přepočítané hodnoty pozic ze záznamu označeném číslem 2 s odchylkou kotev na osách (x, y, z) od kotvy „132“.

Při opakování záznamu z místa 6.9 na druhé straně místnosti se chyba přenesla na kotvy umístěné na opačné straně místnosti. Zaznamenané pozice se od referenčních hodnot výrazně liší pro kotvu „125“. Záznam proběhl pouze velmi krátce při pohybu z natočeného úhlu kvůli omezenému přístupu ke kotvě. Většina bodů tak nebyla zaznamenaná na pozici zařízení, ale na zdi za zařízením. To způsobilo rozdíl na Δx přibližně 25 centimetrů od ostatních zaznamenaných hodnot.



Obrázek 6.13: Import dat z tabulky 6.9 do Sensmap Visualization systému

Obrázek 6.13 naznačuje importované hodnoty (průhledně naznačené ikonou kotvy) získané při záznamu číslo 2 s uloženou pozicí kotev v systému 6.2. Pozice v systému jsou téměř identické až na kotvy „130“ a „126“, které jsou v průměru o 25 centimetrů posunuté do prostoru.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[2-M] 126	-14,557	-2,589	0,105	14,786
[2-M] 124	-13,604	-15,244	0,135	20,432
[2-M] 127	0,701	-4,372	-0,105	4,429
[2-M] 132	0	0	0	0
[2-M] 131	-3,768	0,299	0,058	3,78
[2-M] 125	-2,8	-15,213	0,127	15,469
[2-M] 130	-14,186	0,338	0,121	14,19

Tabulka 6.9: Přepočítané hodnoty pozic ze záznamu označeném číslem 3 s odchylkou kotev na osách (x, y, z) od kotvy „132“.

Obdobně jako u předchozí tabulky se chyba u zařízení „126“ projevila vlivem krátké doby detekce z velké vzdálenosti. Pozice zařízení je tak nesprávným způsobem umístěná do prostředí.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[2-M] 126	-14,875	-2,645	0,128	15,109
[2-M] 124	-13,638	-15,438	0,16	20,6
[2-M] 127	0,803	-4,323	-0,122	4,399
[2-M] 132	0	0	0	0
[2-M] 131	-3,881	0,165	0	3,885
[2-M] 125	-3,036	-15,525	0,11	15,82
[2-M] 130	-14,118	-0,58	-0,039	14,13

Tabulka 6.10: Přepočítané hodnoty pozic ze záznamu označeném číslem 4 s odchylkou kotev na osách (x, y, z) od kotvy „132“.

6.4 Kombinace obou skupin

Finální testování probíhalo nad sektory „H“ a „2-M“ zároveň a po vyhodnocení dat předchozích skupin. Hlavním problémem byl chybný záznam dat způsobený krátkou dobou zachycení pozice jedné kotvy. Při testování pro obě sekce tak u každé kotvy došlo k pozastavení, aby se vytvořil dostatečně velký počet dat. Cílem bylo vyhodnotit, zda má delší záznam vliv na jednotlivě oddělené skupiny. Samozřejmě na základě předchozího testování lze očekávat, že zde bude větší chyba mezi nejvzdálenějšími kotvami. První tabulka stejně jako v předchozích sekcích slouží k porovnání přepočítaných dat z tabulky 6.2. Pro jednodušší porovnání má každá skupina vybrané stejné zařízení jako u předchozích testů.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[2-M] 126	-14,699	-2,614	0,01	14,93
[2-M] 124	-14,185	-15,313	0,02	20,873
[2-M] 127	0,686	-4,471	-0,07	4,524
[2-M] 132	0	0	0	0
[2-M] 131	-3,886	0,229	0,01	3,892
[2-M] 125	-3,343	-15,556	0,03	15,911
[2-M] 130	-14,199	0,214	0,01	14,201
[H] 29B	0	0	0	0
[H] 33B	10,059	0,621	0	10,078
[H] 25A	-0,007	-9,54	0	9,54
[H] 27B	0,117	-5,238	0	5,239
[H] 37A	9,979	-10,518	0	14,498

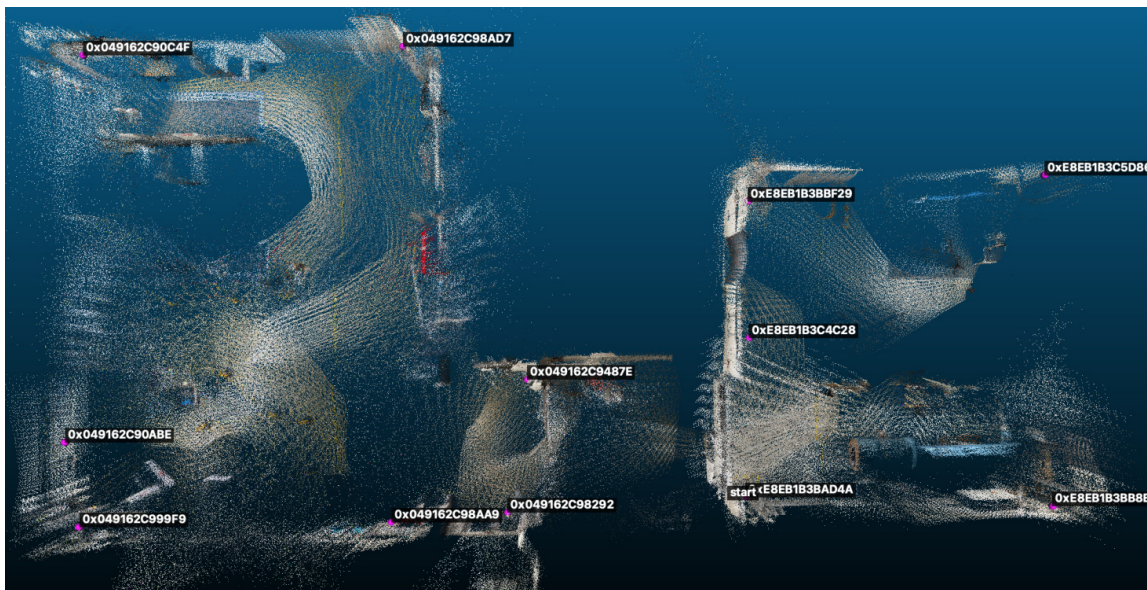
Tabulka 6.11: Přepočítané hodnoty referenčních pozic s odchylkou kotev na osách (x, y, z) od kotvy „132“ pro skupinu „2-M“ a „29B“ pro „H“.

První záznam probíhal od kotvy „[2-m] 124“, která je umístěná v levém horním rohu obrázku 6.8. Postupně se projde kolem ostatních kotev do prostorů skupiny „H“, kde poslední zachycená kotva je „[H] 37A“ v pravém horním rohu místnosti. Data tabulky 6.12 jsou stejně jako u předchozích testů rotovaná o $65,683^\circ$ na rovině xy , pro jednodušší čtení dat. Chyba od referenčních dat nepřesahuje více 21 centimetrů. Největší rozdíl je zaznamenaný u kotvy „[2-M] 125“. U ostatních kotev se v rámci skupiny pozice neliší o více než 10 centimetrů.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[2-M] 126	-14,624	-2,601	-0,008	14,854
[2-M] 124	-14,407	-15,106	-0,019	20,875
[2-M] 127	0,81	-4,335	-0,046	4,41
[2-M] 132	0	0	0	0
[2-M] 131	-3,836	0,229	0,011	3,843
[2-M] 125	-3,641	-15,704	0,023	16,121
[2-M] 130	-14,122	0,218	0	14,124
[H] 29B	0	0	0	0
[H] 33B	10,142	0,45	-0,118	10,153
[H] 25A	0,237	-9,62	0,02	9,623
[H] 27B	0,145	-5,268	0,005	5,27
[H] 37A	9,835	-10,745	0,04	14,566

Tabulka 6.12: Přepočítané hodnoty pozic ze záznamu s odchylkou kotev na osách (x, y, z) od kotvy „132“ pro skupinu „2-M“ a „29B“ pro „H“.

Obrázek 6.14 ukazuje výsledný zaznamenaný point cloud z tabulky 6.13. Fialové body s označením mac adresy jsou zachycené pozice kotev.



Obrázek 6.14: Point cloud záznamu s detekovanými kotvami

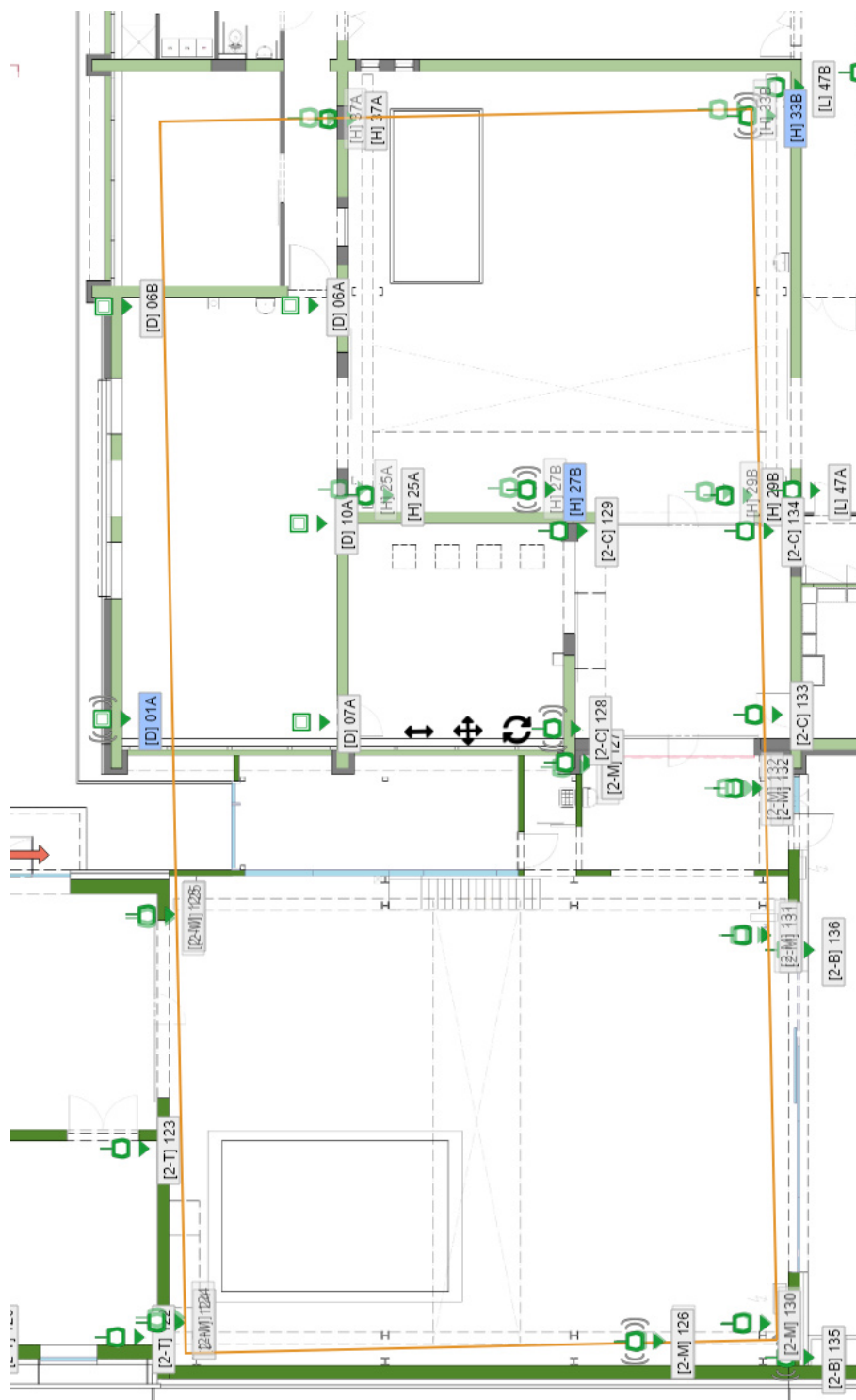
Druhý záznam se více přibližuje k referenčním hodnotám pozic kotev. Hodnoty pozic kotev jsou z původního záznamu rotované o 99,930°. Rozdíl u žádné hodnoty nepřesahuje 15 centimetrů. Zásadní rozdíl pozic kotev se neprojevil ani v rámci druhé skupiny „2-M“ záznamu. První zaznamenaná je kotva „[H] 37A“. Poslední zachycená je kotva „[2-M] 124“. Oproti předchozímu záznamu se hodnoty liší hlavně v umístění výšce.

Alias	Δx [m]	Δy [m]	Δz [m]	Vzdálenost [m]
[2-M] 126	-14,615	-2,599	0,087	14,844
[2-M] 124	-13,774	-15,38	0,128	20,647
[2-M] 127	0,73	-4,396	-0,025	4,457
[2-M] 132	0	0	0	0
[2-M] 131	-3,858	0,245	0,047	3,866
[2-M] 125	-3,179	-15,48	0,118	15,804
[2-M] 130	-14,185	0,207	0,103	14,187
[H] 29B	0	0	0	0
[H] 33B	10,168	0,498	-0,078	10,18
[H] 25A	0,262	-9,741	-0,034	9,744
[H] 27B	0,218	-5,237	-0,068	5,242
[H] 37A	10,12	-10,442	-0,118	14,542

Tabulka 6.13: Přepočítané hodnoty pozic ze záznamu s odchylkou kotev na osách (x, y, z) od kotvy „132“ pro skupinu „2-M“ a „29B“ pro „H“.

Obrázek demonstruje 6.15 výrazně viditelnou odchylku pozic kotev skupiny „H“ (importovaná data jsou průhledně naznačená). Importovaná data jsou otáčena z pozice kotvy „130“ ve skupině „2-M“ v levém dolním rohu obrázku. Vhodnějším postupem pro nahrání pozic by bylo rozdělení dat na 2 menší skupiny a importovat stejným způsobem, jak znázorňuje obrázek 6.13. Aplikace Sensmap Visualization umožňuje volně pohybovat a natáčet

skupinou pozic kotev. Počáteční pozice mobilního telefonu tedy nezpůsobuje žádné problémy.



Obrázek 6.15: Import dat z tabulky 6.13 do Sensmap Visualization systému

6.5 Zhodnocení provedení

Z testů vyplynulo, že hlavním problémem je nedeterministická chyba pohybových senzorů, která značně ovlivňuje rozdíl zachycených pozic mezi záznamy. Na kvalitu a konzistenci záznamů tak má výrazný vliv, jak se uživatel v prostoru pohybuje. Stabilizací lze podstatně snížit rozdíl pozic mezi záznamy. Vzhledem k dosahu a způsobu instalace by vzdálenost mezi kotvami neměla přesahovat 25 metrů. Lze pořídit záznam, který je pro systém RTLS Studio od firmy Sewio Networks s. r. o. dostačující. Vzdálenosti mezi kotvami při instalaci v prostorech nepřesahují 15 metrů. Výsledné záznamy zachycené algoritmem nepřesahují tolerovanou odchylku od reálné pozice.

Hlavní výhodou pořizování dat s implementací této práce je rychlost a pohodlnost pořízení dat. V průběhu testování průměrný záznam trval zhruba 2 až 3 minuty. Není tak problém si zpětně ověřit a získat více dat, popřípadě upravit nastavení algoritmu pro speciální případy. Pro zlepšení přesnosti při pořizování záznamu je vhodné vyčkat několik sekund s přímým pohledem mobilního telefonu na pozici kotvy. Tím selepší zachycené pozice kotvy a dojde k eliminaci chybně získaných bodů detekováním pod úhlem, nebo při rychlejšímu pohybu, protože lidar mobilního telefonu v kombinaci s místem detekovaného zařízení může být mimo povrch kotvy.

Při pořizování dat posledních dvou záznamů bylo využito získaných znalostí z předchozích testů a došlo k redukci vlivu chybně zaznamenaných bodů. Lidar senzor podle specifikace umožňuje zachytit hodnoty v rozmezí 1 až 4 metrů, při testování však docházelo dokonce k záznamu bodů až na vzdálenost 6 metrů.

Pro záznam na větší vzdálenost je důležité být v dostatečně osvětleném prostoru a QR kód by měl mít rozměry zhruba 20 centimetrů na výšku i na šířku. Při menších velikostech se snižuje vzdálenost, na kterou je možné pozice detekovat.

6.6 Následující práce

Jednoduchost využití aplikace podstatně předběhla původní myšlenku, ze které zadání vychází. Hlavní výhodou využití je rychlá zpětná vazba díky výpočtu vzdáleností v reálném čase, která umožňuje v průběhu instalace kotev kontrolovat pozici zařízení. Podpora exportu formátu přímo pro import do systému eliminuje lidskou chybu nepřesného zadání informace do tabulek, které se následně v systému zobrazí.

Vzhledem k dostupnosti kontextu prostředí, v jakém se detekované kotvy nacházejí, by bylo možné vytvářet skupiny kotev přímo při záznamu na základě zakódovaných informací v příslušném QR kódu. Z point cloud dat lze na základě dosahu a priorit vytvořit kontrolu přímého dohledu mezi kotvami. Zároveň lze rozšířit vykreslování na obrazovce tak, že by informace o kotvě (například dosah) byla vizualizována na displeji mobilního telefonu. Uživatel by tak dostal zpětnou vazbu o tom, zda kotvy mají přímý dohled mezi sebou. Vizualizace by také umožnila detekovat místa s horším pokrytím a simulovat odrazy signálů kotev v prostoru od objektů.

Další možností by bylo vylepšení záznamu prostředí za účelem zpřesnění dat. Ačkoliv algoritmus funguje lépe, než se očekávalo, je zde velký prostor pro zlepšení kvality dat záznamu obsahující delší a větší oblasti. V popsané implementaci této práce lze momentálně zahlédnout velké odchylky v datech mezi více záznamy. Bylo snahou využít vzdálenostních dat záznamu pro kontrolu pohybu zařízení oproti předchozímu snímku. Při testování vznikly chyby pohybových senzorů, kdy byl detekován pohyb, i když se s mobilním telefonem nepo-

hybovalo. Využití částečné integrace V-SLAM algoritmu [21] by umožnilo redukovat rozdíly mezi záznamy.

Kapitola 7

Závěr

Při práci jsem se musel seznámit s několika různými aplikacemi a způsoby mapování prostoru. Na základě získaným vědomostí jsem vyhodnotil jednotlivé vyzkoušené způsoby. Problematika strojového vidění je poměrně rozsáhlou a velmi rozvětvenou oblastí, se kterou se během každodenního života setkáváme, například při rozpoznávání lidí na fotkách nebo detekce jídla za pomoci umělé inteligence ve spojení s obrazovými daty. Pro účely práce bylo testováno několik alternativních způsobů detekce kotev. Úspěšnost byla nízká, zejména kvůli generickému vzhledu zařízení a malým rozměrům. Potřeba detekovat zařízení z několika metrů se jevila problematická. Na základě experimentů jsem se rozhodl vydat se alternativní cestou detekování QR kódů pro okamžité a jednodušší rozpoznání pozice kotvy a zároveň jejich snadný přenos informací. Nečekanou částí práce bylo seznámení se s programováním na grafické kartě, které urychluje zpracování zachycených dat a poskytuje vizualizace v reálném čase. Značná část práce objasňuje fungování komunikace a předávání dat o instrukcích mezi procesory mobilního zařízení.

Velký vliv na kvalitu výsledných dat získaných za pomoci aplikace Matterport, či obdobných alternativ, má umělá inteligence, která zodpovídá za generování 3D modelu. V nacházející době lze tedy očekávat velmi rozsáhlé pokroky. Momentálně je stále pro účely této práce kvalita získaných dat nedostačující, a proto se práce vydala jinou cestou. Záměr se přesunul k zařízením s podporou senzorů, která zprostředkovávají data o prostředí formou čitelnou pro stroje. Na základě znalostí získaných z dokumentace a specifikací dostupných zařízení došlo k návrhu algoritmu. Algoritmus byl navržený pro typ zařízení, na kterém jsme se s firmou Sewio Networks s. r. o. shodli, tedy na výběru mobilních zařízení řady Pro od firmy Apple. Hlavním důvodem této volby byla podpora a dostupnost zařízení. V rámci implementace vytvořeného algoritmu se podařilo původní návrh vylepšit. Testování probíhalo hlavně při vývoji. Algoritmus byl zaměřen na jednoduché využití bez nutnosti speciálních požadavků na změnu vzhledu kotvy. Zpětná vazba přímo na obrazovce telefonního zařízení sloužila jako reference kvality získaných dat v reálném čase.

Testováním bylo ověřeno, že použití implementace algoritmu při instalaci je určité možné. Zachycená data se pohybují s rozdílem v řádech centimetrů. Hlavní výhodou je zde jednoduchost, se kterou implementace umožňuje získat data. Velký potenciál skrývá implementace v oblasti dalšího rozšíření, existují již vize, jak algoritmus vylepšit a rozšířit jej o další funkcionality. Uživateli tak bude možné zprostředkovat více dat se záměrem jednodušší a lepší instalace prvků RTLS systému.

Velký potenciál má implementace i pro další rozšíření, kde už je několik nápadů, jak lze existující algoritmus vylepšit a rozšířit o další funkcionality. Uživateli tak zprostředkovat více dat s účelem jednodušší a lepšího procesu instalace.

Velmi oceňuji spolupráci s firmou Sewio Networks s. r. o.. Díky zpětné vazbě, která byla poskytována externím vedoucím, byl udržen rámec práce ve vytčených mantinelech a díky tomu bylo možné implementaci dokončit s předstihem a provést hned několik testů v prostředí firmy Intemac.

Literatura

- [1] ALENYÀ, G., FOIX, S. a TORRAS, C. Using ToF and RGBD Cameras for 3D Robot Perception and Manipulation in Human Environments. *Intell. Serv. Robot.* Berlin, Heidelberg: Springer-Verlag. oct 2014, sv. 7, č. 4, s. 211–220. DOI: 10.1007/s11370-014-0159-5. ISSN 1861-2776. Dostupné z: <https://doi.org/10.1007/s11370-014-0159-5>.
- [2] ALLEN, P. G. *RGB-D Object Recognition and Detection*. 2020 [cit. 2022-07-03]. Dostupné z: <https://www.cs.washington.edu/research-projects/robotics/rgb-d-object-recognition-and-detection/>.
- [3] ANDERS GRUNNET JEPSEN, J. W. *Stereo depth cameras for mobile phone*. 2022 [cit. 2023-02-10]. Dostupné z: <https://dev.intelrealsense.com/docs/stereo-depth-cameras-for-phones>.
- [4] APPLE INC.. *ARKit*. 2023 [cit. 2022-11-12]. Dostupné z: <https://developer.apple.com/documentation/arkit>.
- [5] APPLE INC.. *CaptureImage - Apple developer documentation*. 2023 [cit. 2023-04-22]. Dostupné z: <https://developer.apple.com/documentation/arkit/arframe/2867984-capturedimage>.
- [6] APPLE INC.. *Metal*. 2023 [cit. 2022-11-12]. Dostupné z: <https://developer.apple.com/documentation/metal/>.
- [7] APPLE INC.. *Metal Performance Shaders*. 2023 [cit. 2023-04-22]. Dostupné z: <https://developer.apple.com/documentation/metalperformanceshaders/>.
- [8] BRUNETTO, N., FIORAIO, N. a DI STEFANO, L. Interactive RGB-D SLAM on Mobile Devices. Cham: Springer International Publishing. 2015, s. 339–351.
- [9] CHEUNG, W. LiDAR remote sensing and applications , by Pinliang Dong and Qi Chen, Boca Raton, CRC Press, 2018, 200 pp., US 94.95 (paperback), ISBN 9781138747241. *Annals of GIS*. First. Květen 2018, sv. 24, č. 1, s. 1–2. DOI: 10.1080/19475683.2018.1471522.
- [10] DIGITALTRENDS. *A self-driving car in every driveway? Solid-state lidar is the key*. 2023 [cit. 2023-03-05]. Dostupné z: <https://www.digitaltrends.com/cars/solid-state-lidar-for-self-driving-cars/>.
- [11] DONG, P. a CHEN, Q. *LiDAR Remote Sensing and Applications*. CRC Press, 2017. Remote Sensing Applications Series. ISBN 9781351233330. Dostupné z: <https://books.google.cz/books?id=utZCDwAAQBAJ>.

- [12] DORODNICOV, S. *The basics of stereo depth vision*. 2018 [cit. 2023-02-10]. Dostupné z: <https://www.intelrealsense.com/stereo-depth-vision-basics/>.
- [13] GISGEOGRAPHY. *A Complete Guide to LiDAR: Light Detection and Ranging*. 2022 [cit. 2022-10-02]. Dostupné z: <https://gisgeography.com/lidar-light-detection-and-ranging/>.
- [14] INTEL CORPORATION. *Intel® RealSense™ Documentation*. 2023 [cit. 2023-01-20]. Dostupné z: <https://dev.intelrealsense.com/docs>.
- [15] *QR Code bar code symbology specification*. Standard ISO 18004:2015. Geneva, CH: International Organization for Standardization, 2015.
- [16] LI, J., GAO, W., WU, Y., LIU, Y. a SHEN, Y. High-quality indoor scene 3D reconstruction with RGB-D cameras: A brief review. *Computational Visual Media*. Březen 2022, sv. 8, s. 1–25. DOI: 10.1007/s41095-021-0250-8.
- [17] LIU, S., ZHANG, M., KADAM, P. a KUO, C. *3D Point Cloud Analysis: Traditional, Deep Learning, and Explainable Machine Learning Methods*. Springer International Publishing, 2021. ISBN 9783030891800. Dostupné z: <https://books.google.cz/books?id=nEpUEAAAQBAJ>.
- [18] LUDOVICO CELLENTANI. *An introduction to Metal API*. 2018 [cit. 2023-04-22]. Dostupné z: <https://medium.com/@lcellentani/an-introduction-to-metal-api-f69ded594fee>.
- [19] MALIK, A. *RTLS For Dummies*. Wiley, 2009. ISBN 9780470398685. Dostupné z: <https://books.google.cz/books?id=e8X2jwEACAAJ>.
- [20] MARIUS HORGA. *Working with memory in Metal*. 2017 [cit. 2023-02-14]. Dostupné z: <https://metalkit.org/2017/04/30/working-with-memory-in-metal/>.
- [21] MATHWORKS. *What Is SLAM? 3 things you need to know*. 2023 [cit. 2023-02-10]. Dostupné z: <https://www.mathworks.com/discovery/slam.html>.
- [22] MATTERPORT. *Get Started With Matterport*. 2022 [cit. 2022-08-07]. Dostupné z: https://support.matterport.com/s/?language=en_US.
- [23] MCMANAMON, P. *LiDAR Technologies and Systems*. SPIE Press, 2019. Press Monographs. ISBN 9781510625396. Dostupné z: <https://books.google.cz/books?id=cIIIwAEACAAJ>.
- [24] MESIKA, A., BEN SHABAT, Y. a TAL, A. CloudWalker: Random walks for 3D point cloud shape analysis. *Computers & graphics*. Ithaca: Elsevier Ltd. 2022, sv. 106, s. 110–118. ISSN 0097-8493.
- [25] OPENCV. *OpenCV 2.4.13.7 documentation - Camera Calibration and 3D Reconstruction*. 2019 [cit. 2022-11-12]. Dostupné z: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.
- [26] OPPERMANN, I., HÄMÄLÄINEN, M. a IINATTI, J. *UWB: Theory and Applications*. First. Wiley, 2005. ISBN 9780470869185. Dostupné z: <https://books.google.cz/books?id=KiaquNK9r4oC>.

- [27] PMD TECHNOLOGIES. *Tutorial - Point Cloud Library*. 2022 [cit. 2023-02-10].
Dostupné z: <https://3d.pmdtec.com/en/tutorial-point-cloud-library/>.
- [28] ROCKROBOTIC. *Ground Control Points (GCPs) Best Practices*. 2023 [cit. 2023-03-03].
Dostupné z: <https://learn.rockrobotic.com/gcp-best-practices>.
- [29] SEWIO NETWORKS, S.R.O.. *Anchor Omni Deployment Rules*. 2022 [cit. 2023-01-20].
Dostupné z:
<https://docs.sewio.net/docs/anchor-omni-deployment-rules-43713521.html>.
- [30] SEWIO NETWORKS, S.R.O.. *Anchor Overview*. 2022 [cit. 2023-01-20]. Dostupné z:
<https://docs.sewio.net/docs/anchor-overview-30147655.html>.
- [31] SEWIO NETWORKS, S.R.O.. *LOS vs NLOS*. 2022 [cit. 2023-01-20]. Dostupné z:
<https://docs.sewio.net/docs/los-vs-nlos-25593229.html>.
- [32] SEWIO NETWORKS, S.R.O.. *Sewio Public Documentation - Sewio Documentation*.
2022 [cit. 2022-10-02]. Dostupné z: <https://docs.sewio.net/docs>.
- [33] TSCHIRSCHNITZ, M. von, WAGNER, M., PAHL, M. a CARLE, G. A Generalized TDoA/ToA Model for ToF Positioning. *CoRR*. 2019, abs/1909.09405. Dostupné z:
<http://arxiv.org/abs/1909.09405>.