

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2022

Patrik Červený



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

DETEKCE 3D OBJEKTŮ V MRAČNU BODŮ

3D OBJECT DETECTION FROM POINTCLOUD

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Patrik Červený

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Zemčík

BRNO 2022

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Patrik Červený

ID: 211139

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Detekce 3D objektů v mračnu bodů

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je navrhnout systém k detekci objektů na vlakovém nástupišti ve trojrozměrném mračnu bodů (např. z LiDARu nebo RADARu). Detekované objekty budou cestující a jiné objekty které mohou přesahovat / spadnout přes známou ochrannou zónu nástupní hrany nástupiště. Systém dále vyhodnocuje míru zásahu do ochranné zóny pro případný návazný bezpečnostní systém.

1. Proveďte rešerši používaných metod pro 3D detekci objektů.
2. Seznamte se s dostupnými nástroji pro zpracování mračen bodů.
3. Seznamte se se senzory pro pořízení 3D dat a data nasbírejte.
4. Navrhněte systém pro zpracování dat v popsané aplikaci.
5. Navrhovaný systém implementujte.
6. Systém otestujte s ohledem na kvalitu detekce a výpočetní náročnost.
7. Testování systému vyhodnoťte.

DOPORUČENÁ LITERATURA:

[1] - JAHNE, Bernd a Horst HAUSSECKER, 2000. Computer vision and applications: a guide for students and practitioners. San Diego: Academic Press. ISBN 01-237-9777-2.

Termín zadání: 7.2.2022

Termín odevzdání: 23.5.2022

Vedoucí práce: Ing. Tomáš Zemčík

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

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

Abstrakt

Bakalárska práca sa zaoberá detekciou 3D objektov v mračne bodov získaného pomocou LIDAR senzora. V práci sú popísané senzory, ktoré zbierajú dáta pre mračno bodov, metódy na ich spracovanie a konečná reprezentácia zozbieraných dát pomocou zvolenej aplikácie. Pomocou LIDAR senzora detegujeme 3D obraz mračna bodov na železničných staniach v Brne a za pomoci dostupných metód na spracovanie a zvolenej aplikácie sa snažíme popísať, čo nám senzor skrz dáta zobrazuje.

Kľúčové slová

Mračno bodov, LIDAR, RADAR, 3D detekcia

Abstract

The bachelor thesis deals with the detection of 3D objects in a point cloud acquired by a LIDAR sensor. The thesis describes the sensors that collect the data for the point cloud, the methods to process the data and the final representation of the collected data, using the chosen application. By using a LIDAR sensor we detect a 3D image of the point cloud at the railway stations in Brno and with the help of the available methods for processing and application we try to describe what the sensor shows us through the data.

Keywords

Pointcloud, LIDAR, RADAR, 3D detection

Bibliografická citace

ČERVENÝ, Patrik. Detekce 3D objektů v mračnu bodů [online]. Brno, 2022 [cit. 2022-05-20]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/142641>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Tomáš Zemčík.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	<i>Patrik Červený</i>
VUT ID studenta:	<i>211139</i>
Typ práce:	<i>Bakalářská práce</i>
Akademický rok:	<i>2021/22</i>
Téma závěrečné práce:	<i>Detekce 3D objektů v mračne bodů</i>

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 20. května 2022

podpis autora

Pod'akovanie

Chcel by som sa týmto poďakovať vedúcemu práce Ing. Tomášovi Zemčíkovi za jeho prístup ku konzultáciám, za ochotu a čas pri meraní dát, odbornú pomoc a užitočné rady pri písaní tejto práce.

V Brne dňa: 20. mája 2022

podpis autora

Obsah

ZOZNAM OBRÁZKOV	9
ZOZNAM TABULIEK	10
ÚVOD	11
1. MRAČNO BODOV A SENZORY NA JEHO TVORBU	12
1.1 ČO JE MRAČNO BODOV ALEBO „POINTCLOUD“ ?	12
1.2 SENZORY URČENÉ NA ZBER DÁT MRAČNA BODOV	14
1.2.1 RADAR senzor	15
1.2.2 LIDAR senzor.....	16
1.2.3 Hĺbková kamera.....	17
2. METÓDY SPRACOVANIA MRAČNA BODOV	18
2.1 FILTRÁCIA DÁT.....	18
2.1.1 Filtrovanie zmiešaných pixelov.....	19
2.1.2 Filtrovanie iného šumu	19
2.2 REGISTRÁCIA DÁT	20
2.2.1 Definícia deskriptora bodu	21
2.2.2 Definícia kľúčových bodov.....	21
2.2.3 Hrubá registrácia.....	21
2.3 SEGMENTÁCIA DÁT	22
2.3.1 Segmentácia na základe klastrovania	23
2.3.2 Segmentácia na základe hrany.....	23
2.3.3 Segmentácia na základe regiónu.....	24
2.3.4 Segmentácia na základe RANSAC algoritmu.....	24
3. NÁVRH SYSTÉMU SPRACOVANIA DÁT	26
3.1 PREDSTAVENIE APLIKÁCIE	26
3.2 POUŽÍVANÝ HARDWARE.....	27
3.3 KONFIGURÁCIA UMIESTNENIA SENZORA	30
3.3.1 Poloha senzora.....	31
3.4 SOFTVÉR	33
3.4.1 Knihnica PCL (Point Cloud Library).....	33
3.5 NASNÍMANÉ DÁTA.....	34
3.6 NAVRHOVANÝ ALGORITMUS	36
3.7 PROGRAM.....	37
3.7.1 Štart programu.....	38
3.7.2 Načítanie cloudov	38
3.7.3 Spájanie cloudov.....	39
3.7.4 PassThrough filter	39
3.7.5 VoxelGrid filter	40
3.7.6 Porovnanie cloudou Octree	41
3.7.7 Klastering.....	42
3.7.8 Načítavanie BoudingBoxov.....	42
3.7.9 Vytvorenie CropBoxu.....	43
3.7.10 Vizualizácia.....	44

3.8 TESTOVANIE SYSTÉMU	44
4. ZÁVER.....	46
LITERATÚRA.....	47
ZOZNAM PRÍLOH.....	49

ZOZNAM OBRÁZKOV

<i>1.1: Ukážka mračna bodov, obsahujúca približne 3 miliónov bodov</i>	12
<i>1.2: Graf rozsahu použitia jednotlivých skenerov [3]</i>	14
<i>1.3: Schéma radarového systému [5]</i>	15
<i>1.4: Princíp systému LIDAR [6]</i>	16
<i>1.5: Pasívny/Aktívny systém hĺbkovej kamery [19]</i>	17
<i>2.1: Koncept procesu spracovania údajov z mračna bodov [8]</i>	18
<i>3.1: Nástupná hrana na železničnej stanici označená žltou farbou</i>	26
<i>3.2: Meracie zariadenie MVG-MTDL</i>	27
<i>3.3: Senzor Livox Mid-70 [7]</i>	28
<i>3.4: Kamera Raspberry Pi HQ [20]</i>	29
<i>3.5: Nástupná oblasť na železničnej stanici</i>	30
<i>3.6: Umiestnenie senzora - poloha A</i>	31
<i>3.7: Umiestnenie senzora - poloha B</i>	32
<i>3.8: Umiestnenie senzora - poloha C</i>	33
<i>3.9: Porovnanie mračna bodov s obrázkom z kamery (Brno-Královo Pole)</i>	34
<i>3.10: Porovnanie mračna bodov s obrázkom z kamery (Brno-Hlavní nádraží)</i>	35
<i>3.11: Porovnanie mračna bodov s obrázkom z kamery (Brno-Židenice)</i>	35
<i>3.12: Návrh pre vývojový diagram algoritmu spracovania</i>	36
<i>3.13: Vývojový diagram programu ObjectDetection_PCL</i>	37
<i>3.14: Statická scéna mračna bodov (Brno-Královo Pole)</i>	38
<i>3.15: Statická scéna mračna bodov po filtrácii PassThrough (Brno-Královo Pole)</i>	40
<i>3.16: Statická scéna mračna bodov po filtrácii PassThrough a VoxelGrid (Brno-Královo Pole)</i>	41
<i>3.17: Porovnávanie Octree (Brno-Hlavní nádraží)</i>	41
<i>3.18: Zobrazenie BoundingBoxov</i>	43
<i>3.19: Zobrazenie CropBoxov</i>	43
<i>3.20: Zobrazenie osí vo vizualizácií</i>	44

ZOZNAM TABULIEK

<i>2.1: Súhrn algoritmov čistenia dát mračna bodov [8]</i>	18
<i>2.2: Súhrn metodológie segmentácie dát mračna bodov [8]</i>	22
<i>3.1: Špecifikácia meracieho zariadenia MVG-MTDL [15]</i>	27
<i>3.2: Špecifikácia senzora LIDAR [7]</i>	28
<i>3.3: Špecifikácia kamery Raspberry Pi HQ Camera [15]</i>	29

ÚVOD

Medzi ľuďmi je pojem mračno bodov neznámy. Avšak zoznámenie sa sním, Vám otvorí nové dvere pri vnímaní reality pomocou dnešných novodobých technológií. V ére technológií umelej inteligencie, autonómneho riadenia, robotiky alebo vnímania prostredia v reálnom čase, je veľmi dôležité sa zamerať, na bezpečnosť týchto technológií. Čo teda mračná bodov sú a k čomu nám slúžia? Z pohľadu detekcie a rozpoznávania objektov v doprave, dbáme na už spomínanú bezpečnosť a predikciu pred vzniknutím nebezpečenstva. K snímaniu scén reality nám slúžia rôzne snímače či senzory, ktoré nám získavajú dáta a skrz rôzne techniky spracovania, vieme s týmito dátami pracovať a spracovávať ich. Ide o dlhý proces, hlavne z hľadiska vývoja a zavedenia daných techník do reálneho sveta.

Význam práce spočíva v oboznámení sa hlbšie s problémami týkajúcich sa mračien bodov, tiež si predstavíme druhy snímačov, pomocou ktorých môžeme tieto body snímať. Popíšeme si oblasti využitia mračna bodov, ich spracovanie pomocou známej metodiky ale aj potenciálnu budúcnosť tohto odvetvia.

Cieľom bakalárskej práce je detekcia objektov v mračne bodov na železničnej stanici. Pri práci si zvolíme senzor, s ktorým budeme pracovať, popíšeme si dostupné metódy na spracovanie a tiež vo finálnom návrhu systém spracovania.

Bakalárska práca je rozdelená do štyroch hlavných častí. V prvej časti si povieme všeobecné informácie ohľadom mračna bodov, ich využitie v rôznych sektoroch, senzorov na zbieranie dát.

V druhej časti sú popísané vybrané metódy spracovania týchto bodov. Pôjde hlavne o metódy filtrácie, registrácie a segmentácie dát.

V tretej časti bude implementovaný systém spracovania zvolenej aplikácie, konfigurácia umiestnenia senzora, používaný softvér a samotný program.

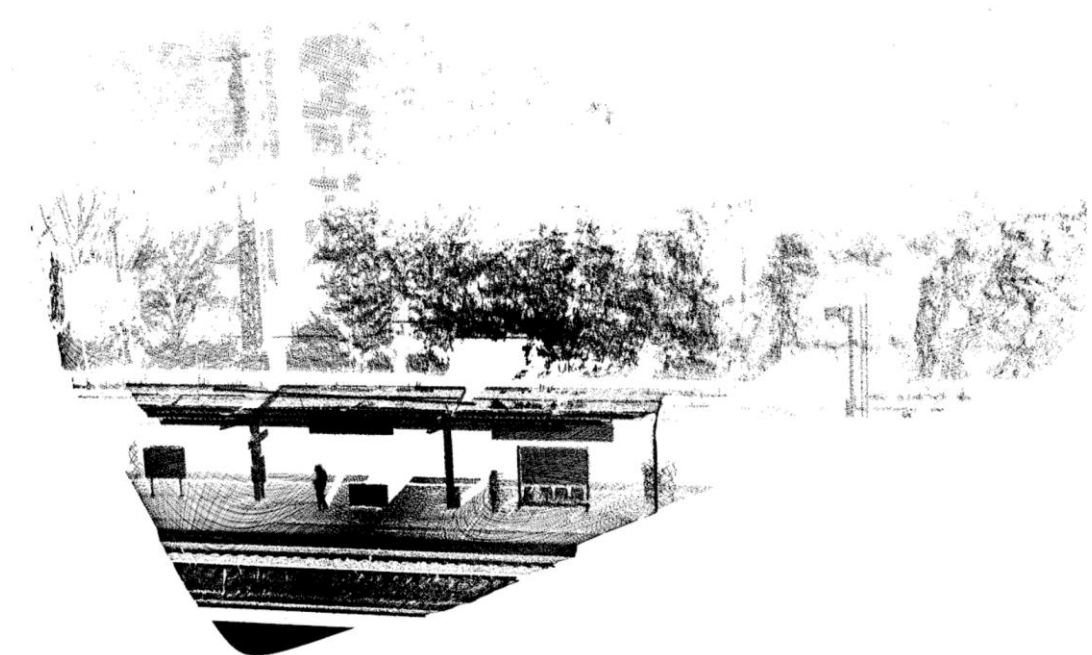
V poslednej časti, si túto prácu a jej dosiahnuté výsledky celé zhrnieme a vyhodnotíme. Tiež sa pozrieme na možnú budúcnosť tohto riešenia, ktoré by mohlo byť implementované v reálnom svete aj s vylepšeniami, ktoré som nemal k dispozícii.

1. MRAČNO BODOV A SENZORY NA JEHO TVORBU

Úvodná kapitola obsahuje všeobecné informácie o mračne bodov a senzoroch na jeho tvorbu. Zo všeobecného hľadiska nie je táto téma medzi ľuďmi veľmi známa. Dozvieme sa, čo si pod týmto pojmom môžeme predstaviť, na akú konkrétnu oblasť sa zameriava, pomocou akých nástrojov mračno generujeme a kde a ako sa využíva.

1.1 Čo je mračno bodov alebo „Pointcloud“ ?

Z technického hľadiska sú mračná bodov množiny údajov, ktoré predstavujú znázornený objekt alebo priestor. Je to dátová štruktúra, ktorá sa používa na reprezentáciu množiny, viacrozmerných bodov a väčšinou sa používa na reprezentáciu trojrozmerných údajov. V 3D reprezentácii týchto dát, body predstavujú geometrické súradnice x, y, z snímaného povrchu. Keď máme k dispozícii údaje o farbe, mračno sa zmení na 4D. Mračno bodov je možné získať z hardvérových senzorov, kamier alebo ich možno generovať synteticky z počítačového programu. Pri skenovaní aj tých najmenších objektov môžu vznikať milióny bodov. Hustota získaných bodov závisí od času snímania, počtu a rozlíšenia snímku [1].



Obr. 1.1: Ukážka mračna bodov, obsahujúca približne 3 miliónov bodov

Keď pomocou senzorov vytvoríme 3D model, existuje veľa variácií toho, ako sa tieto modely používajú. V prípadoch použitia toho, čo chceme pomocou svojho modelu dosiahnuť, môže ovplyvniť špecifiká mračna bodov, ktoré musíme vytvoriť [3].

Tu sú tri typické prípady použitia:

1. **Grafické modelovanie** (vizualizácia 3D plánov alebo preskúmanie priestoru)
2. **Technické merania** (vzdialené meranie scény, pozemné laserové skenery)
3. **Veľkoplošná kartografia** (drony, mobilné mapovanie...)

Pre grafické a technické prípady je možné tieto prípady použitia aplikovať rovnako dobre na objekty, ako na budovy alebo na scény. Okrem vytvorenia 3D modelu, môžu zhotovené skenovania pomôcť identifikovať špecifické, alebo geografické riziká a tým umožniť príslušnú prevenciu pred vzniknutím týchto rizík. V nebezpečnom prostredí to môže výrazne zlepšiť bezpečnosť pracovnej sily [3]. Meranie mračna bodov vieme využiť kdekoľvek, či už v otvorenom alebo uzavretom priestore.

Existuje niekoľko vzorových sektorov použitia, ktoré si tu predstavíme [3].

1. **Architektúra:** Vo veľkej miere sa používa pri renováciách – v súčasnosti sa čoraz viac nasadzuje ako referencia v reálnom svete pre nové budovy.
2. **Výstavby:** Aplikuje sa na celý projekt od prieskumu miesta až po návrh, výstavbu a údržbu - pre existujúce budovy alebo nové projekty.
3. **Baníctvo:** Podpora udržateľnosti, nákladovej efektívnosti a bezpečnosti – najmä v oblasti ťažobnej robotiky a automatizácie na mieste.
4. **Stavebné inžinierstvo:** Hľadanie aplikácií v mestskom plánovaní, monitorovaní stavu povrchov, povodňových rizík a analýze zosuvov pôdy.
5. **Doprava:** Skenovanie prostredia, pohybujúcich sa objektov ale tiež aj statických scén, napríklad pri aplikácií autonómneho riadenia vozidiel.

Kľúčovým faktorom pri získavaní údajov mračna bodov je prístup/viditeľnosť k naskenovaným povrchom. Vo väčšine prípadov sa body získavajú tak, že máme priamy záber na daný objekt alebo scénu, bez akéhokoľvek obmedzenia. Pri snímaní väčších objektov, treba pokryť všetky pozície pri meraní, aby bol objekt zachytený zo všetkých uhlov. Tým si skenovanie vyžaduje viac času [4].

Existuje mnoho formátov súborov pre 3D modelovanie. Každý snímač môže mať definovaný svoj formát súboru, do ktorého sa mračná ukladajú. Taktiež rôzne druhy softvérov má predefinované formáty, s ktorými pracuje. Preto je niekedy dôležitým faktorom vybrať správny snímač a tiež aj k nemu príslušný softvér [4].

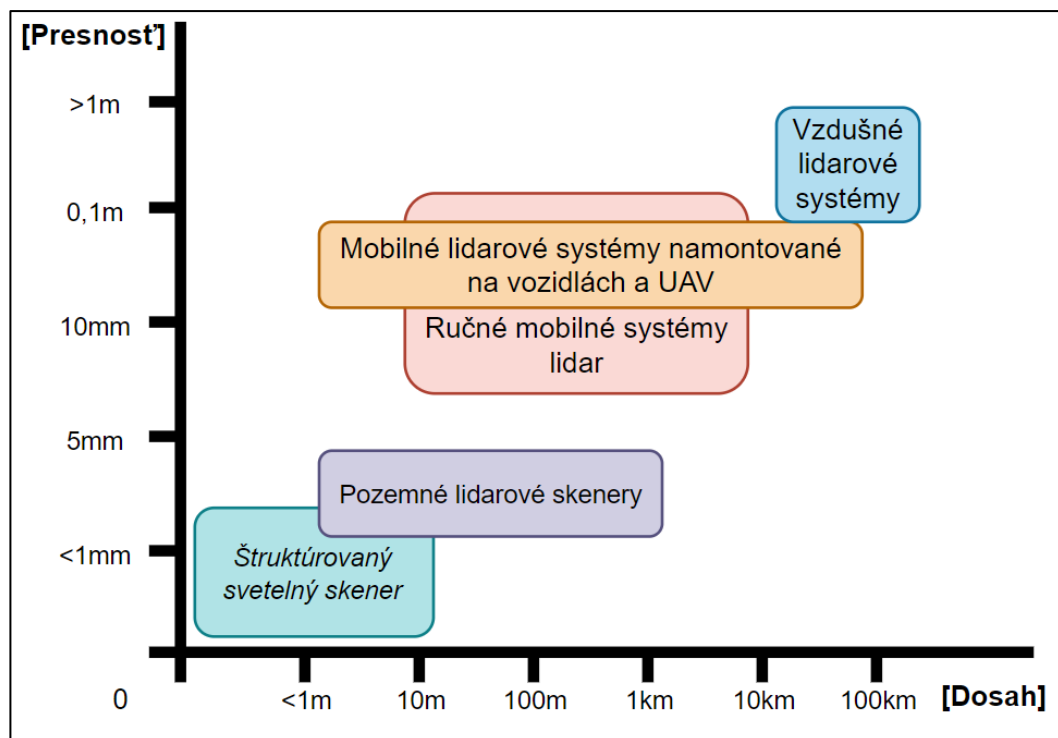
3D vnímanie si získava stále väčší význam nielen v robotike, ale aj v iných oblastiach. Za posledných 20 rokov sme prešli dlhú cestu vnímania okolia od jednoduchých snímačov dosahu, ako napríklad akustický zameriavač, radar alebo IR senzory poskytujúce určité množstvo informácií o prostredí, cez kamery až po laserové skenery [1].

Na tento hardvér na zber dát sa viac zameriame v ďalšej kapitole.

1.2 Senzory určené na zber dát mračna bodov

Pre získanie našich dát teda mračna bodov, je potreba určitých senzorov. Ukážeme si ich využitie v globálnom meradle a detailnejšie si ich popíšeme. Tieto senzory alebo typy snímání si môžeme rozdeliť do týchto kategórií.

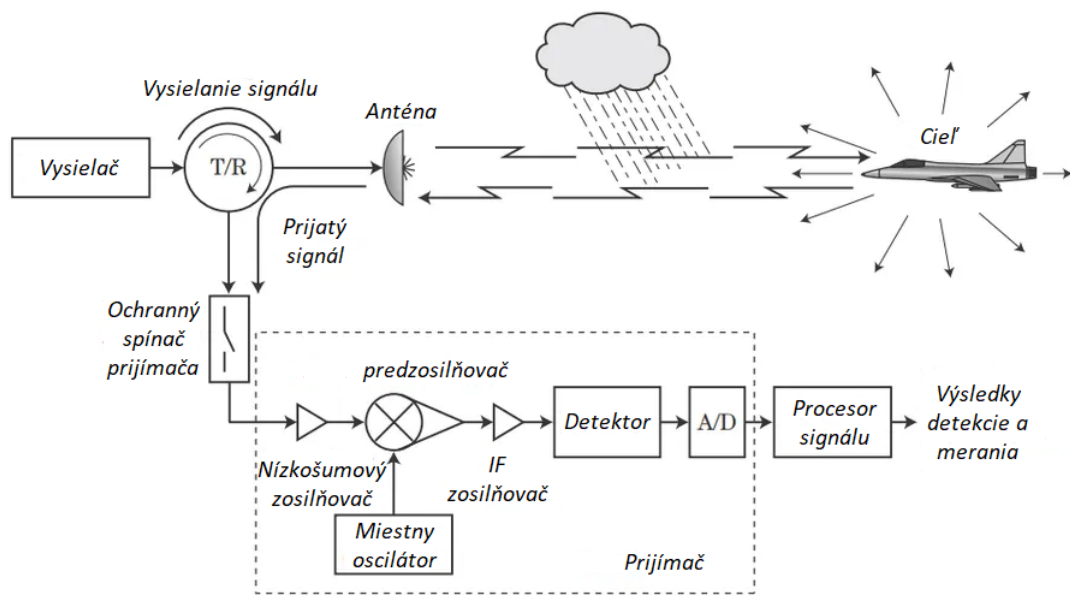
- 1. Mobilné systémy mapovania (MMS):** Tieto systémy vďaka tomu, že nie sú statické a menia svoju polohu vzhľadom na okolie, nám umožňujú rýchle a flexibilné zachytávanie reality. Preto sú ideálne na snímání veľkého množstva dát v kratšom časovom úseku. Do tejto kategórie napríklad patrí kamera, LIDAR senzor a systém určovania polohy (napr. SLAM, IMU, GNSS alebo ich kombinácia). MMS sa dnes využívajú napríklad pri autonómnom riadení a to namontované na automobile, prípadne na lietadlách čo umožňuje snímání rôznych scén prostredia [3].
- 2. Pozemné laserové skenery (TLS):** Sú vhodné na zachytávanie reality, ak potrebujete merania na technickej úrovni. Tieto skenery sú schopné poskytnúť pokrytie scény a vysokú presnosť detailov. Na **Obr. 1.2** máme uvedený graf, ktorý zobrazuje použitie a mierku dostupných typov skenerov a taktiež to, kde fungujú najlepšie a kde sa prekrývajú. Každý typ skenera má konkrétny prípad použitia [3].



Obr. 1.2: Graf rozsahu použitia jednotlivých skenerov [3]

1.2.1 RADAR senzor

Radar je skratka pre „Radio Detection and Ranging“, čo v preklade znamená rádiová detekcia a zameriavanie. Podľa toho vieme že princíp fungovania radaru bude založený na použití rádiových vln. Rádiové vlny majú oveľa dlhšiu vlnovú dĺžku v porovnaní so svetelnými vlnami, vďaka čomu sú radary schopné pokryť väčšie vzdialenosti. Frekvencia a typ použitých rádiových vln závisí od požiadaviek vášho meracieho zariadenia. Radarové systémy majú dve hlavné zložky: vysielač a prijímač. Na **Obr. 1.3** môžeme vidieť schému moderného systému Radar [5].



Obr. 1.3: Schéma radarového systému [5]

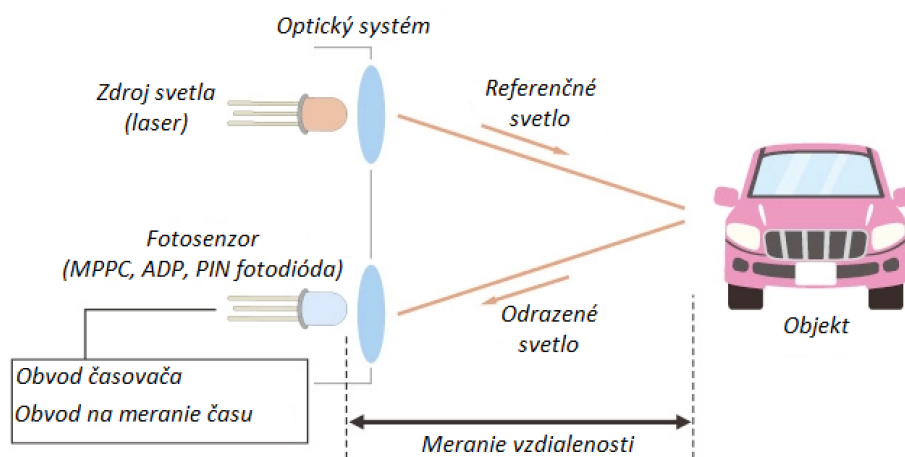
Rádiové vlny, ktoré generuje radarový systém môžu byť kontinuálne alebo pulzné (častejšia možnosť). Vysielač generuje elektromagnetickú vlnu a anténa vysiela vlnu cez médium. Potom signál prechádza médium až kým nenarazí na objekt od ktorého sa odrazí späť do radarového systému. Radarový systém prijme odrazenú vlnu cez prijímač. Radar spracováva odrazené rádiové vlny a určuje vzdialenosť vzhľadom k systému skrz časový interval medzi prenosom signálu a časom kedy je signál prijatý. V závislosti od aplikácie radarového senzora sa bude frekvencia prenášaného signálu líšiť, vplyv na to má napríklad dosah, vlnová dĺžka a veľkosť antény. Systémy s vyššími frekvenciami majú nižší výkon, vyšší útlm a jemnejšiu detekciu, vďaka čomu sú ideálne pre aplikácie s krátkym dosahom, ktoré vyžadujú vyššie rozlíšenie, ako napríklad v autonómnych vozidlách [5].

1.2.2 LIDAR senzor

LIDAR alebo „Light Detection And Ranging“ znamená detekciu a meranie svetla alebo vzdialenosti. Aj keď ide o pomerne novší druh senzora v porovnaní s radarom, tento senzor existuje od objavu laserov a to v šesťdesiatych rokoch minulého storočia. Existujú dva typy týchto senzorov a to vzdušné a pozemné.

Pozemný LIDAR využíva lasery s vlnovou dĺžkou okolo 700 - 1000 nm a používa sa v rôznych aplikáciách, ako sú napríklad štúdie zosuvov pôdy alebo detekcia okolitej reality v autonómnych vozidlách. Na druhej strane, vzdušný LIDAR používa lasery s vlnovou dĺžkou 1000 - 1600 nm a väčšinou sa používajú na zber údajov o teréne. Systémy LIDAR sú schopné merať prostredie s vysokou presnosťou a na základe výsledkov vytvárajú 3D obraz [5].

Laser vysielaný z LIDAR senzora môže byť diskretný alebo kontinuálny. Sensory, ktoré využívajú súvislé vlny, využívajú fázový rozdiel spätného signálu na určenie vzdialenosti a charakteristiky objektu. V prípade pulzných vln nás viac zaujíma amplitúda vysielaných a prijímaných signálov, aby sme vytvorili bod mraku, ktorý odráža detegovaný objekt. Systém LIDAR pozostáva hlavne zo zdroja svetla a prijímača alebo senzora. Zdroj vyžaruje svetlo alebo laser, ktorý sa odrazí od cieľa a späť smerom k systému LIDAR, kde snímač zachytí impulz. Na určenie presnej vzdialenosti objektu systém vypočítava časový interval medzi vyžarovaním lasera a jeho prijatím, pričom berie do úvahy rýchlosť svetla. Existujú laserové snímače, ktoré merajú presnú vzdialenosť na základe fázového posunu. Tento fázový posun medzi laserom odrazeným od cieľa a vyžarovaným laserom, je ale priamoúmerný času letu [5].



Obr. 1.4: Princíp systému LIDAR [6]

Samotný LIDAR dokáže zachytiť polohu, veľkosť a tvar objektu vzhľadom na systém. Tieto senzory sú však často spárované s GPS, IMU alebo kamerou, aby sa rozšírilo ich použitie [5].

1.2.3 Hĺbková kamera

Stereo kamera je typ kamery s dvoma alebo viacerými obrazovými snímačmi. To umožňuje kamere simulovať ľudské binokulárne videnie a teda jej dáva schopnosť vnímať hĺbku. Človek vníma hĺbku aj pomocou stereo disparity, ktorá sa vzťahuje na rozdiel v umiestnení obrazu videného ľavým a pravým okom. Mozog využíva disparitu na získavanie informácií o hĺbke dvojrozmerných obrazcov, ktoré sú známe ako stereopsia.

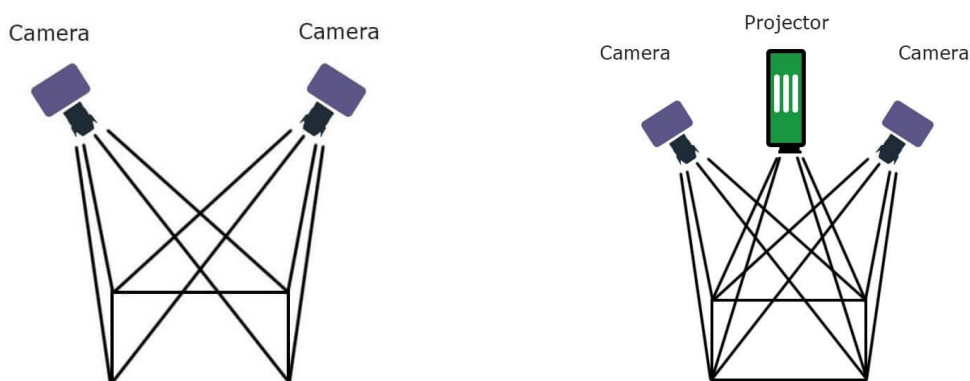
Práve hĺbkové kamery sa pokúšajú napodobniť techniku ľudského videnia na vnímanie hĺbky. Toho vnímanie sa vykonáva geometrickým prístupom nazývaním triangulácia.

Stereo-disparitu vo fotoaparáte možno nájsť pomocou dvoch 2D obrázkov nasnímaných z rôznych pozícií. Koreláciou medzi obrázkami možno použiť na vytvorenie hĺbkového obrazu. Na nájdenie korelácií však musia mať dva obrázky dostatočné detaily a textúru alebo nerovnomernosť.

Zachytenie tretej dimenzie sa dá uskutočniť mnohými rôznymi spôsobmi a každá z dostupných technológií strojového videnia má svoje výhody a nevýhody. Trojrozmerné zobrazovanie možno rozdeliť do dvoch hlavných kategórií: pasívne (hĺbka od zaostrenia, svetelné pole, stereo) a aktívne (čas letu, štruktúrované svetlo, interferometria).

Pasívny stereo systém závisí od dostupného svetla v prostredí a nevyužíva žiadne prídavné svetlo. Tento systém môžeme vidieť na ľavej strane obrázku 1.5.

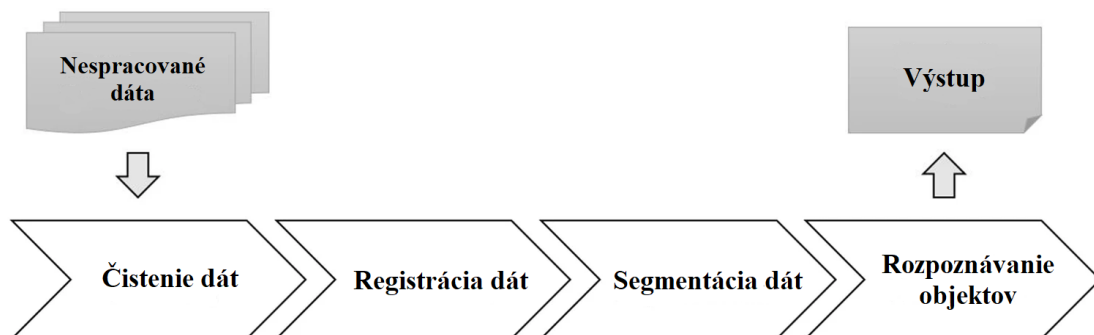
Aktívne stereo videnie využíva svetlo, ako je laser alebo štruktúrované svetlo, aby sa zjednodušil problém stereo zhody. Je užitočné v oblastiach, kde neexistuje dostatok prvkov na meraním objekte, ktoré by slúžili na nájdenie korešpondencií. Infračervený projektor alebo iný svetelný zdroj zaplaví scénu textúrou, čím preruší závislosť od externého svetelného zdroja. Tento systém môžeme vidieť na pravej strane obrázku 1.5 [19].



Obr. 1.5: Pasívny/Aktívny systém hĺbkovej kamery [19]

2. METÓDY SPRACOVANIA MRAČNA BODOV

Údaje mračna bodov zo snímacej techniky nám poskytujú surové body v priestore, v ktorých sa ťažko orientuje. Preto je potreba mračno bodov upraviť do stavu tak, aby obsahoval všetky informácie potrebné na analýzu. Tento proces sa volá spracovanie. Pre spracovanie mračna existuje enormné množstvo metód. V tejto kapitole si popíšeme podrobnejšie tie, ktoré by nám mohli byť užitočné. Na **Obr.2.1** môžeme vidieť koncept spracovania údajov, ktorý sa dnes používa.



Obr. 2.1: Koncept procesu spracovania údajov z mračna bodov [8]

2.1 Filtrácia dát

Tab. 2.1: Súhrn algoritmov čistenia dát mračna bodov [8]

Názov	Algoritmus
Odstránenie zmiešaných pixelov	Medián filter, Detekcia zmeny v odrazivosti a/alebo rozsahu, Filter normálového uhla, Filter dĺžky hrany, Algoritmus kužeľa vplyvu Optimálny prah založený na analýze pravdepodobnosti
Odstraňovanie šumu „sol’ a korenie“	Konverzia na obrázky, po ktorých nasledujú techniky filtrovania, vrátane stredného filtra, erózie morfológie a filtra veľkosti objektu
Odstraňovanie obklopujúceho šumu/zhluky	Filter intenzity
Vyhľadovanie šumu	Odhad hustoty jadra Odhad bodovej normály na základe robustných čiastkových klasifikácií Anizotropné geometrické priemerné zakrivenie toku
Odstránenie nezrovnalostí medzi skenmi	Prístup mapovania tieňov

Mračno bodov často obsahuje údaje o šume v dôsledku technických obmedzení snímačov. V literatúre sa uvádza množstvo algoritmov a techník na identifikáciu a odstraňovanie rôznych typov údajov o šume z údajov mračna bodov, ako je zhrnuté v *tab. 2.1* [8].

2.1.1 Filtrovanie zmiešaných pixelov

Existuje niekoľko druhov efektov pri skenovaní, ktoré môžu viesť k poškodeným alebo degradovaným dátam laserového skenovania vrátane zmiešaných pixelov, presahu alebo intenzity presluchu a posunu dosahu. Najzaujímavejšia téma v oblasti výskumu bola v poslednej dobe práve téma zmiešaných pixelov [8].

Zmiešaný pixel je typ chybných bodov laserového skenovania, ku ktorým dochádza, keď laserový lúč presne dopadne na okraj objektu a rozdelí sa na dve časti. Dve časti laserového lúča dopadajú na dva rôzne objekty a výsledkom sú dva rôzne laserové signály. Nakoniec laserový skener prijme dva reflektované signály a vytvorí zmiešaný pixel, ktorý nepredstavuje žiadny objekt a je zvyčajne izolovaný od normálnych bodov laserového skenovania [8].

Preto sa navrhlo použitie niektorých algoritmov na odstránenie zmiešaných pixelov ako napríklad použitie medián filtra, algoritmu zmeny v povrchovej odrazivosti a/alebo rozsahu s cieľom identifikovať zmiešané pixely. Tieto algoritmy boli v zásade založené na troch rôznych filtroch vrátane filtra s normálnym uhlom, filtra dĺžky hrany a algoritmu kužeľa vplyvu. Filter normálneho uhla, skúma uhol medzi normálou lokálneho povrchu a smerom pohľadu. Zmiešané pixely vždy prinášajú šikmý uhol. Filter dĺžky hrán, deteguje zmiešané pixely pohľadom na dĺžky hrán po triangulácii bodov laserového skenovania, pretože zmiešané pixely majú tendenciu viesť k dlhším dĺžkam hrán. Algoritmus kužeľa vplyvu deteguje zmiešané pixely, kde niekoľko susedných pixelov spadá do kužeľa vplyvu pozdĺž smeru pohľadu laserového skenera [8].

2.1.2 Filtrovanie iného šumu

Okrem algoritmov na filtrovanie zmiešaných pixelov sa používajú aj algoritmy na odstraňovanie iných typov údajov o šume. Na odstránenie škvŕn alebo známeho šumu „sol’ a korenie“ je možné previesť údaje z mračna bodov na obrázky v odtieňoch šedej a potom generovať binárne obrázky. Binárne obrazy sa potom filtrovali pomocou techník spracovania obrazu vrátane medián filtra, morfolologickej erózie a filtra veľkosti objektu. Po filtrovaní binárnych obrazov sa filtrované binárne obrazy použijú ako masky na originálne obrázky v odtieňoch šedej na odstránenie zašumených pixelov. Ďalší prípad šumu v údajoch je známy ako šum okolo snímača, ktorý nastáva, keď skutočná vzdialenosť od skenera k cieľovému objektu presahuje jednoznačnú vzdialenosť laserového skenera. Obvodový šum sa vyskytuje iba pri laserových skeneroch, ktoré

používajú metódu merania rozsahu fázového posunu. Keďže skener meria vzdialenosť na základe fázového posunu v rozsahu od 0 do 2π , nejednoznačná vzdialenosť by mala byť vzdialenosť, ktorá vedie k fázovému posunu 2π . Akonáhle skutočná vzdialenosť prekročí jednoznačnú vzdialenosť, bude meraná ako vzdialenosť, ktorá spadá do nejednoznačnej vzdialenosti a vytvorí rovnaký fázový posun [8].

Jedným z účinných prístupov k odstraňovaniu rušivého šumu je filter intenzity, ktorý skúma hodnoty odrazivosti bodov laserového skenovania. Hodnota odrazivosti snímacieho bodu meria intenzitu odrazeného laserového signálu, ktorý silne súvisí so skutočnou vzdialenosťou, pretože intenzita laserového lúča sa pri pohybe priestorom rozptyľuje a tlmí. Preto sa dá obklopujúci šum ľahko odfiltrovať na základe informácií o odrazivosti nastavením prahu intenzity. Sú algoritmy ktoré, odstraňujú odľahlé hodnoty bez ohľadu na typ šumu. Avšak tieto algoritmy predpokladajú že cieľový objekt bude hladkého povrchu a ľubovoľný bod, ktorý porušuje tento povrch sa považuje za šum [8].

2.2 Registrácia dát

Na úplné pokrytie komplexného priestoru sú zvyčajne potrebné údaje z mračna bodov nasnímané z viacerých pozícií, najmä pri laserovom skenovaní. Na získanie jedného a úplného mračna bodov z viacerých mračien bodov by sa všetky údaje mali skombinovať v rovnakom referenčnom systéme prostredníctvom procesu známeho ako registrácia údajov [8].

Registráciu údajov možno rozdeliť na jemnú a hrubú registráciu. Pre jemnú registráciu je najčastejšie používaným riešením algoritmus „Iteračný najbližší bod (ICP)“. Avšak aplikácia tohto algoritmu môže zlyhať, pretože prístupy založené na ICP sa zameriavajú na lokálnu optimalizáciu vysoko nekonvexnej objektívnej funkcie. Znamená to, že na úspešnú implementáciu je potrebné vopred zosúladiť dve mračná bodov. Preto sa v praxi používa menej presná metodika, hrubá registrácia, pomocou ktorej možno získať počiatočné riešenie na zarovnanie dvoch mračien. Jednou z široko používaných metód pre hrubú registráciu je prístup založený na kľúčových bodoch, v ktorom sa kľúčové body najskôr detegujú z mračna bodov na základe určitého deskriptora bodov a potom sa na základe dvoch súborov kľúčových bodov vypočíta zarovnanie medzi dvoma mračnami bodov [8].

Nasledujúce podkapitoly majú informatívny charakter o bežných procesoch hrubej registrácie. Deskriptory bodov sú najprv definované tak, aby popisovali každý bod a zdôrazňovali hlavné črty. Potom sa na základe deskriptorov pôvodné mračno bodov zredukuje na riedky súbor kľúčových bodov. Nakoniec sa vykoná hrubá registrácia, aby sa extrahovala transformačná matica medzi dvoma mračnami bodov na základe zistených kľúčových bodov na prekrývajúcej sa oblasti [8].

2.2.1 Definícia deskriptora bodu

Deskriptor bodu sa zvyčajne používa na identifikáciu, ku ktorému primitívu bod patrí. Často sa získava výpočtom vzťahu medzi bodmi v danom polomere. Môže byť definovaný geometrickými informáciami okolia každého bodu, kde je veľkosť okolia optimalizovaná pomocou určitého algoritmu. Ďalší z existujúcich algoritmov, dokáže vypočítať deskriptor pre každý bod povrchu pomocou lokálnej geometrie, ktorá je odolná voči zašumeným údajom. Deskriptor bodu s názvom Histogram bodových prvkov alebo (PFH)/ (FPPH) je pomocou histogramov geometrických vlastností okolitých susedstiev schopný, identifikovať, či bod patrí k rovinnému, lineárnemu alebo okrúhlemu povrchu. Ďalší zo série deskriptorov je napríklad lokálny 3D deskriptor s názvom SHOT. Zo susedných bodov bol definovaný nový súradnicový systém a kovariančná matica deskriptora rozdelila okolie sieťou izotropnej gule. Dôkladné experimentálne hodnotenie ukázalo, že SHOT prekonal najmodernejšie lokálne deskriptory [8].

2.2.2 Definícia kľúčových bodov

Po extrahovaní deskriptorov bodov sa kľúčové body detegujú z mračna bodov pred hrubou registráciou. Detekcia kľúčových bodov pozostáva z lokalizácie kľúčových častí objektu. Napríklad medzi kľúčové časti tváre patria končeky nosa, obočie atď. Tieto časti pomáhajú reprezentovať daný objekt detailnejšie. Pri rešerši detektorov môžeme nájsť ich rôzne typy, ktoré tu nebudem úplne popisovať kvôli ich komplexnosti. Napríklad 3D detektor kľúčových bodov Harris, ktorý bol inšpirovaný jeho 2D verziou [1]. Detektor Gaussovho rozdielu (DoG) atď. Pre tieto detektory sa vykonalo mnoho štúdií, kde boli porovnávané alebo experimentálne obohacované a konkrétne funkcie. Okrem detektorov kľúčových bodov DoG a Harris výskumníci v posledných rokoch navrhli niektoré nové kombinácie detektora a deskriptora kľúčových bodov. Zistilo sa, že novšie prístupy preukázali lepšiu registračnú výkonnosť, pokiaľ ide o účinnosť, presnosť a robustnosť [8].

2.2.3 Hrubá registrácia

Po zistení kľúčových bodov sa zaregistrujú dve mračná bodov nájdením matice transformácie zarovnania medzi nimi na základe pôvodných mračien bodov alebo kľúčových bodov. Bežne používaný registračný algoritmus je známy ako algoritmus 4-bodová kongruentná sada (4-PCS). Vypočítava hrubú registráciu tuhého telesa zo sady štyroch koplanárnych bodov a ich priestorových vzťahov. Hoci algoritmus 4-PCS predstavoval dôležitý pokrok pre hrubú registráciu 3D mračien bodov, stále existovala nevýhoda pri hľadaní koplanárnych bodov vo všetkých potenciálnych zdrojových bodoch. Pri tomto algoritme sa opäť vyskytli určité zlyhania, mal veľa verzií a bol časom vylepšovaný pomocou určitých funkcií. Nedávno Bueno a spol. [9] navrhol algoritmus založený na

zistených kľúčových bodoch vo farebnom mračne bodov pomocou 4-PCS. Navrhnutá metóda bola testovaná a preukázalo sa, že má lepšie výsledky pri hrubej registrácii ako predchádzajúce [8].

2.3 Segmentácia dát

Po zarovnaní všetkých mračien bodov do rovnakého referenčného systému sa zvyčajne vyžaduje segmentácia registrovaných údajov na zoskupenie bodov s podobnými charakteristikami, do homogénnych oblastí. Pri segmentácii údajov sú body, ktoré patria k rovnakému povrchu alebo oblasti, označené rovnakým štítkom a body, ktoré zdieľajú podobné prvky v súvislej oblasti, sú zoskupené, aby sa vytvoril segment. Súhrn metód segmentácie máme v *tab. 2.2*. Všetky metodológie sú rozdelené do šiestich kategórií, vrátane klastrovania, segmentácie založenej na okraji, na regióne, na grafe a modelovania. Nie každú kategóriu si tu predstavíme, iba tie čo nás budú najviac ovplyvňovať a ktoré pri najbližšom spracovaní najviac využijeme [8].

Tab. 2.2: Súhrn metodológií segmentácie dát mračna bodov [8]

Metodiky segmentácie	Výhody	Nevýhody
Na báze klastrovania	Jednoduché na pochopenie a implementáciu	Problém presnosti: citlivý na šum v údajoch a je ovplyvnení definíciou susedného bodu
Na základe hrán	Rýchla segmentácia	Problém presnosti: citlivý na šum a nerovnomernú hustotu mračien bodov
Na základe regiónu	Presnejšie na šum	Nadmerná alebo nedostatočná segmentácia a presnosť určovania hraníc
Na základe grafov	Lepšie na komplexných bodových oblakoch s nerovnomernou hustotou alebo šumom	Nedá sa spracovať v reálnom čase a na uľahčenie procesu je potrebné školenie alebo iný systém
Na základe prispôbenia modelu- Houghova transformácia	Rýchly a robustný s odľahlými hodnotami	Pomalšie a citlivejšie na parametre segmentácie daného modelu
RANSAC	Rýchly a robustný s odľahlými hodnotami, dokáže spracovať veľké množstvo údajov	Presnosť pri spracovaní rôznych zdrojov mračna bodov

2.3.1 Segmentácia na základe klastrovania

Segmentácia založená na klastrovaní je segmentácia dát mračna pomocou zhlukových algoritmov, založených na určitých vlastnostiach bodov. V segmentačných metódach založených na zhlukoch sa geometrické a rádiometrické charakteristiky zvyčajne používajú ako znaky na opis každého bodu. Vlastnosti môžu zahŕňať polohu bodu, lokálne odhadovanú normálu povrchu, zvyšky najlepšie lícujúcej plochy a odrazivosť skenovaných bodov [8]. Filin [10], definoval sedemrozmerný vektor prvkov pre každý bod v mračne bodov. Tieto vlastnosti zahŕňajú polohu, parametre určovania z roviny do okolia bodu a výškový rozdiel medzi bodom a jeho susedmi. Na identifikáciu priestoru prvkov sa použila metóda učenia bez dozoru a potom sa body zoskupili v priestore objektov pomocou merania priestorovej blízkosti. Filin a Pfeifer, [11] predstavili segmentačný algoritmus založený na novo-navrhovanom susedskom systéme nazývanom adaptívny sklon. Na definovanie susedstva medzi bodmi boli prijaté parametre ako hustota bodov, presnosť merania, horizontálne a vertikálne rozloženie bodov. Na segmentáciu sa s obľubou používa aj K-means, ktorý je schopný zoskupovať dáta mračna bodov minimalizovaním súčtu štvorcov vzdialeností medzi bodom a zodpovedajúcim ťažiskom klastra [12]. Segmentácia údajov mračna bodov na základe klastrovania sa dá ľahko pochopiť a implementovať. Keďže však vlastnosti každého bodu sú zvyčajne generované pomocou miestnych susedných bodov, metóda segmentácie založená na zhlukoch je citlivá na šum a je ovplyvnená definíciou susedstva [8].

2.3.2 Segmentácia na základe hrany

Hrany sa zvyčajne používajú na opis charakteristík tvaru rôznych predmetov. Algoritmy segmentácie založené na okrajoch zisťujú hranice regiónov v údajoch 3D mračna bodov, aby získali segmentované oblasti. Segmentačné algoritmy založené na hranách vyžadujú dokončenie dvoch hlavných úloh: identifikácia hraníc hrán a zoskupovanie bodov vo vnútri hraníc na formulovanie segmentov [8]. Všeobecným princípom detekcie hrán je lokalizovať body s rýchlou zmenou hustoty [13] lokálnych povrchov. Normála, gradienty, hlavné zakrivenia a derivácie vyššieho rádu sú bežne používané vlastnosti na identifikáciu hrán. V literatúre bolo navrhnutých množstvo metodológií založených na hranách na segmentáciu údajov mračna bodov. Napríklad algoritmus detekcie hrán výpočtom gradientu, prispôbením 3D čiar množine bodov a identifikáciou zmien v smere jednotkového normálneho vektora na povrchu. Algoritmus segmentácie mračna bodov pomocou odhadovanej povrchovej normály a lokálnej bodovej konektivity, ktorý bol schopný pracovať s neštruktúrovanými údajmi mračna bodov. Navrhovaný algoritmus bol robustný na detekciu rohov a hrán z údajov mračna bodov. Algoritmus využívajúci metodológiu zoskupovania skenovacích čiar. Na rozdiel od iných algoritmov segmentácie údajov tento algoritmus využíval segmenty kriviek ako segmentačné primitíva. Podľa veľkého počtu testov by bolo možné pomocou tohto algoritmu rýchlo segmentovať mračna bodov. Avšak metodologie založené na okrajoch umožňujú rýchlu

segmentáciu, majú problémy s presnosťou, pretože všetky navrhované metodológie sú citlivé na šum a nerovnomernú hustotu mračna bodov, čo sú bežne sa vyskytujúce problémy v skutočných údajoch mračna bodov [8].

2.3.3 Segmentácia na základe regiónu

Segmentačné algoritmy založené na regióne zvyčajne začínajú regiónom z jedného alebo niekoľkých počiatočných bodov a potom iteratívne rastú tak, aby daný región zahŕňal susedné body podľa určitých kritérií. Tieto algoritmy v podstate zahŕňajú dva kroky: identifikáciu a rast povrchu „semien“. Algoritmus pestovania regiónu bol prvýkrát predstavený Beslom a Jainom. [16] Ďalší z vyvinutých segmentačných algoritmov bol založený na raste regiónu a používal K-najbližších susedov na odhad normálového vektora pre každý bod. Ďalší z typov segmentácie je hrubá segmentácia. Identifikuje hlavné objekty na základe normálových vektorov čiže bodov, v rovnakej rovine. Podrobná segmentácia extrahuje ďalšie informácie pre prvky objektu. Zatiaľ čo väčšina metód segmentácie na základe regiónov potrebuje počiatočné body, existujú aj metódy, ktoré počiatočné body nepotrebujú. Táto kategória metód je známa ako metódy segmentácie nenasadených oblastí, v ktorých sú všetky body zoskupené do jednej oblasti predtým, ako ju proces rozdelenia začne rozdeľovať na menšie oblasti. Metódy segmentácie na základe regiónov sú viac presné v porovnaní s metódami založenými na hranách. Majú však problém s nadmernou alebo nedostatočnou segmentáciou a presným určením hraníc regiónu. Vo všeobecnosti platí, že pri použití globálnych informácií je metodika segmentácie založená na regióne odolnejšia voči šumu v porovnaní s metodikou založenou na okrajoch [8].

2.3.4 Segmentácia na základe RANSAC algoritmu

Algoritmus RANSAC Bol navrhnutý Fischlerom a Bollesom v roku 1981[21]. Používa sa na robustné nastavenie parametrického modelu, na údaje s odľahlými hodnotami alebo vysokým stupňom šumu. Generuje veľké množstvo hypotéz o primitívnych tvaroch pomocou náhodnej podmnožiny vzorových bodov, ktoré môžu jednoznačne určiť primitívne parametre. Hoci existujú obmedzenia, RANSAC funguje dobre, pokiaľ ide o výsledky segmentácie a výpočtový čas. Segmentačné metodológie založené na modelovaní obsahujú čisto matematický princíp, vďaka čomu sú rýchle a odolné voči odľahlým hodnotám. Okrem toho je RANSAC tiež schopný spracovať veľké množstvo údajov z mračna bodov v primerane krátkom časovom období. Čo sa týka obmedzení, RANSAC má problém s presnosťou pri spracovaní rôznych údajov z mračna bodov [8].

Základný algoritmus je zhrnutý takto [22] :

1. Náhodný výber minimálneho počtu bodov potrebný na určenie parametrov modelu.
2. Vyriešenie parametrov modelu.

3. Určenie, koľko bodov z množiny všetkých bodov sa hodí s preddefinovanou toleranciou ϵ .
4. Ak je zlomok počtu vložených hodnôt z celkového počtu bodov v súbore prekročí preddefinovaný prah τ , je potrebné prehodnotiť parametre modelu pomocou všetkých identifikovaných vložených hodnôt a následné ukončenie.
5. V opačnom prípade zopakujte kroky 1 až 4 (maximálne N-krát)

Počet iterácií N je zvolený dostatočne veľký, aby sa zabezpečila pravdepodobnosť p (zvyčajne nastavená na 0,99) a aby aspoň jedna zo sady náhodných vzoriek nezahŕňala odľahlú hodnotu [22].

Nech u predstavuje pravdepodobnosť, že ktorýkoľvek vybraný údajový bod je vnorenou hodnotou a $v = 1 - u$ pravdepodobnosť pozorovania odľahlej hodnoty.

Vyžaduje sa N iterácií minimálneho počtu bodov označených m , kde

$$1 - p = (1 - u^m)^N, \quad (2.1)$$

A teda s určitou úpravou ,

$$N = \frac{\log(1-p)}{\log(1-(1-v)^m)}, \quad (2.2)$$

3. NÁVRH SYSTÉMU SPRACOVANIA DÁT

V tejto kapitole si priblížime to o čom táto práca vlastne je, predstavíme si zvolenú aplikáciu a návrh riešenia tejto aplikácie. Taktiež si ukážeme, používaný hardvér, nazbierané dáta a ich spracovanie pomocou zvoleného softvéru. Nakoniec si tento systém otestujeme a vyhodnotíme výsledky.

3.1 Predstavenie aplikácie

Zo začiatku bolo najpodstatnejšie vybrať danú aplikáciu, ktorá by bola vhodná pre túto prácu. Keďže mračná bodov sa v poslednej dobe začali viac využívať v doprave tak som zvolil aplikáciu: „detekcia objektov na vlakovom nádraží s vyhodnotením ich polohy vzhľadom k hrane nástupišťa“. V tomto prípade môže ísť o bezpečnostný systém, ktorý zabezpečí, aby objekty, či už pôjde o ľudí alebo nie, nezasahovali svojím pohybom do zóny, kde končí nástupište a začína koľajová trať. Jeden z dôvodov použitia LIDAR senzora oproti kamere je taký, že nezanedáva priamo obraz ako kamera, ale iba body v priestore, teda mračno bodov a nikto nebude apelovať na to, že ho nasnímala nejaká kamera. Ďalším z dôvodov je aj viditeľnosť kamery v tme. Oproti LIDARU je to značná nevýhoda. Tiež pri tomto detegovaní pôjde aj o bezpečnosť, lebo ako už dobre vieme, na vlakovej stanici nemáme priamo zabezpečený prechod na trať a preto týmto snímačom môžeme doceliť to, že keď sa niekto priblíži k nebezpečnej oblasti pevnou zemou a traťou, kde sú koľaje, upozorní to napríklad niekoho z bezpečnosti alebo sa vyhlási poplach.



Obr. 3.1: Nástupná hrana na železničnej stanici označená žltou farbou

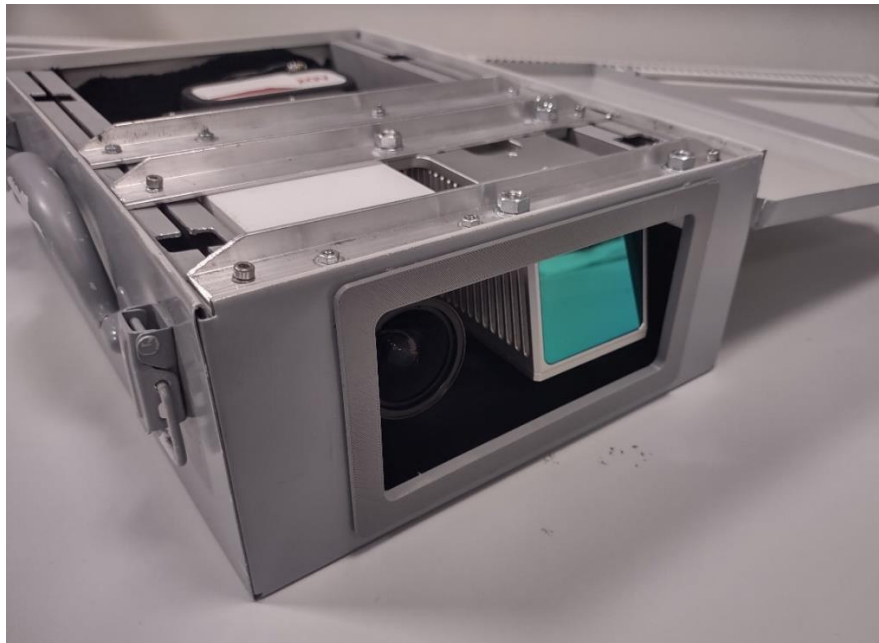
3.2 Používaný hardware

Zvolený senzor je od výrobcu Livox, typ MID-70. Tento senzor je základom meracieho zariadenia, v ktorej je umiestnený spolu s ďalšími komponentami. Zariadenie nesie názov MTDL (Multisensor Traffic Data Logger). Je napájaná batériou a ponúka diskretné a prenosné riešenie pre zbieranie dát v reálnom prostredí [15].

Tab. 3.1: Špecifikácia meracieho zariadenia MVG-MTDL [15]

Rozmery	335 x 230 x 105 [mm]
Hmotnosť	~2.5kg (záleží na konfigurácii)
Rozmer snímacieho okna	140 x 70 [mm]
Vývojová doska	Raspberry Pi 4B
Operačný systém	Linux
Vnútorňa pamäť	microSD 128GB (rozšíriteľná o USB)
Batéria	Li-on, 20V, 4Ah
Výdrž batérie	~3h (záleží na konfigurácii)
Nabíjanie batérie	Externá, rýchla výmena možná

Skladá sa z kovovej skrinky s vnútorným systémom uchytenia na koľajnicu, na ktorú je možné uchytiť všetko vybavenie. Snímacie okno, umožňuje výhľad pre akýkoľvek druh optického alebo RF snímača [15].



Obr. 3.2: Meracie zariadenie MVG-MTDL

Vnútorný napájací systém pozostáva z komerčne dostupného Li-ion akumulátora napájajúceho 20V zbernicu (až 4A špičkový prúd) a jednotlivých výkonových meničov pre elektroniku a snímače, v závislosti od konfigurácie. Nabíjanie batérie sa vykonáva mimo zariadenia a batérie je možné rýchlo vymeniť.

Vývojová doska následne načíta jednotlivé senzory cez Ethernet (1x), USB 3 (2x), USB 2 (2x) alebo USB C (1x) a dáta ukladá buď do vnútornej pamäte (128 GB v aktuálnej konfigurácii), alebo na externú pamäťovú jednotku (USB). SW výpočtovej jednotky umožňuje automatickú prevádzku s prednastavenou konfiguráciou po zapnutí a je možné ho pripojiť aj cez WiFi pre konfiguráciu za chodu.

V súčasnej podobe MVG-MTDL obsahuje snímač Livox MID-70 LiDAR a kameru Raspberry Pi HQ RGB, ďalšie špecifikácie nájdete v tabuľkách 3.2 a 3.3 nižšie [15].



Obr. 3.3: Senzor Livox Mid-70 [7]

Mid-70 je ďalším komponentom s aplikáciami prispôbenými na použitie pri autonómnom riadení pri nízkej rýchlosti alebo mobilnej robotike. Môže pracovať nepretržite až 8 000 hodín pri konštantnej teplote a normálne pracovať v rozsahu teplôt prostredia od $-20\text{ }^{\circ}\text{C}$ do $65\text{ }^{\circ}\text{C}$, čím spĺňajú vysoké požiadavky na dlhodobú neprerušovanú prevádzku komerčných robotov [7].

Tab. 3.2: Špecifikácia senzora LIDAR [7]

Zorné pole	$\pm 70.4^{\circ}$
Dosah	do 260m, použiteľných 130m
Vlnová dĺžka	905 nm
Presnosť rozsahu(1σ)	pri 20m ≤ 0.02 m
Uhlová presnosť	$< 0.1^{\circ}$
Bodová sadzba	100 000 bodov/s
Rozhranie	Ethernet

V porovnaní so súčasným trhom, ktorý primárne využíva drahé laserové žiariče alebo nevyzreté MEMS skenery, Mid-70 využíva lacné, vysokokvalitné polovodičové komponenty na generovanie a detekciu svetla. Celý optický systém, vrátane skenovacích jednotiek, využíva overené a ľahko dostupné optické komponenty, aké sa používajú v priemysle optických šošoviek [7].



Obr. 3.4: Kamera Raspberry Pi HQ [20]

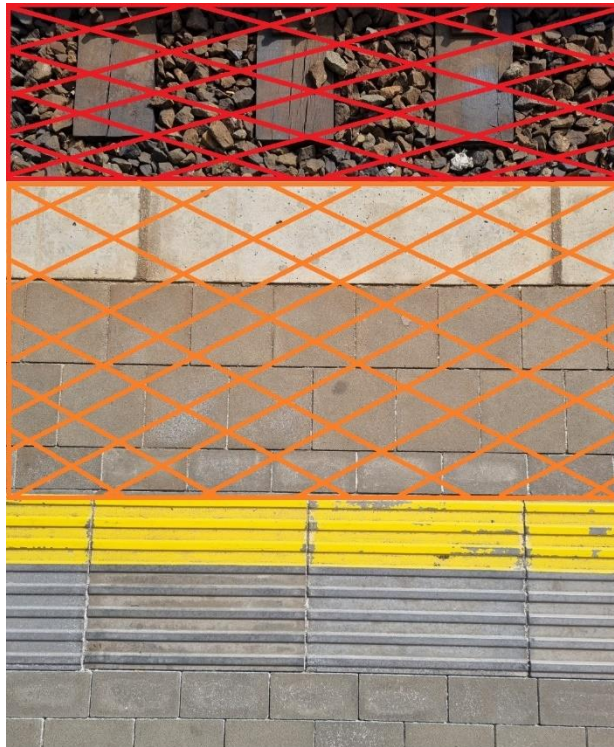
Tab. 3.3: Špecifikácia kamery Raspberry Pi HQ Camera [15]

Senzor	Sony IMX 477 family
Typ senzora	RGB
Veľkosť senzora	1/2.3 palca
Rozlíšenie	12,3MPx
Uchytenie šošovky	C/CS mount
Filter	IR cutoff
Rozhranie	50p, 0.5mm, XF3M FPC/FFC

3.3 Konfigurácia umiestnenia senzora

V tejto podkapitole si predstavíme dostupné prostredia, kde pravdepodobne prebiehať meranie, nastavenie senzora pri meraní a jeho najvhodnejšiu polohu. Z prieskumu dostupných vlakových nádraží v Brne, som zistil že nie na všetkých sú moderne označené nástupné hrany žltou farbou.

Z prvej snímky môžeme vidieť označenú „bezpečnú hranicu“ spomínanou žltou farbou. Tá oddeľuje priestor pred hranou nástupišťa (oranžová farba) a samotným priestorom kde sa nachádza železnica (červená farba). Žiaľ takéto označenie žltou farbou nenájdeme na každom nádraží a preto ľudia nevedia, kde sa povedomá hranica nachádza. Práve vyznačená oranžová farba a červená farba označuje „nebezpečný priestor“, pre ktorý by sme chceli docieľiť aby sa mu ľudia vyhýbali počas príchodu vlaku v závislosti od rizika, ktoré by mohlo nastať.



Obr. 3.5: Nástupná oblasť na železničnej stanici

Našou úlohou je detegovať objekty/ľudí na nástupišti pomocou senzora a priestor nástupnej oblasti definovať za nebezpečný. Takto budeme môcť predchádzať riziku. Preto je podstatné správne umiestniť a nakonfigurovať senzor, tak aby bola snímaná hrana viditeľná skrz senzor a aby sme dokázali dané objekty správne segmentovať. Predstavíme si možnosti umiestnenia senzora, jeho výhody, nevýhody a prípadné vylepšenia do budúcnosti.

3.3.1 Poloha senzora

Pre optimálnu polohu senzora, bolo nutné ísť sa zoznámiť s prostredím a spraviť malý prieskum dostupných vlakových nádraží v Brne. Určíme jednotlivé polohy senzora a zvolíme najvhodnejšiu a dostupnú polohu s ktorou budeme pracovať.

1. Poloha A:

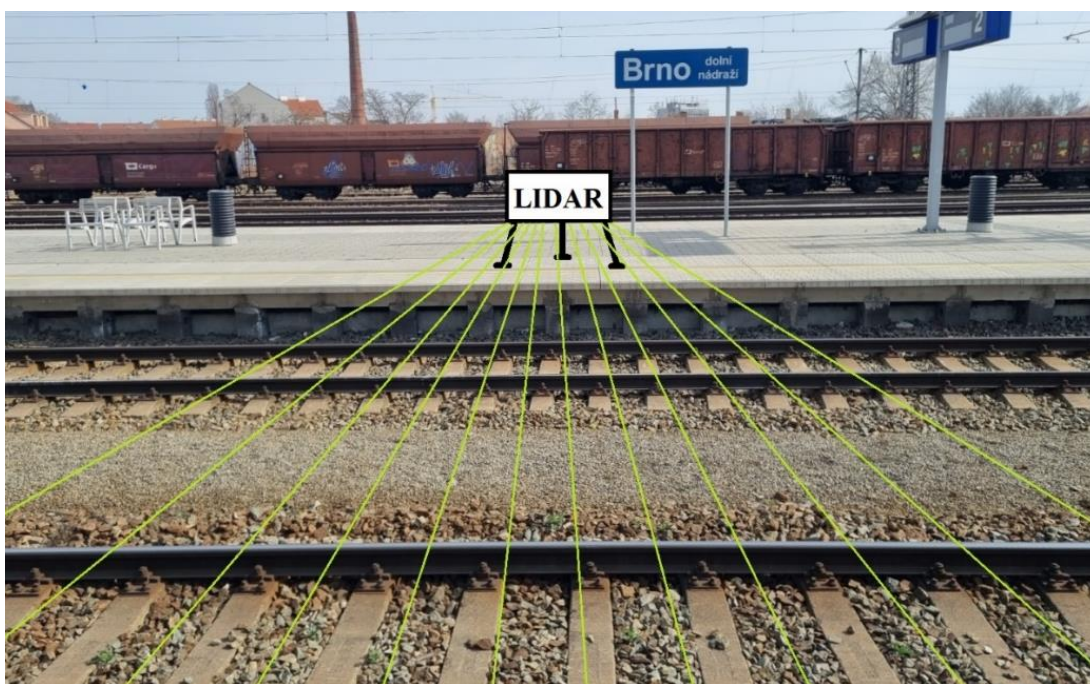
Senzor je umiestnený priamo nad hranou, medzi železnicou a nástupišťom a tak dokáže presne snímať objekty pod ním. Síce nemá veľký rozptyl zachytenia ale to by sa dalo riešiť tak, že by sa umiestnilo viac sensorov vedľa seba z určitou medzerou medzi nimi. Avšak nie všetky nádražia disponujú takýmto prístreškom a preto je nemožné aplikovať túto polohu na každé nádražie. Ďalším možným problémom by bolo umiestnenie senzora na tento prístrešok a taktiež by proces snímania mohol niektorým ľuďom prekážať, pretože by si mohli myslieť, že ide o kameru. Táto poloha senzora je podľa môjho uváženia najvhodnejšia pre meranie. Takúto polohu senzora sa mi nepodarilo otestovať, pretože som nemal priamy prístup k prístrešku a taktiež na podobné inštalovanie nejakého zariadenia na vlakovom nádraží je potrebné si vybaviť povolenie.



Obr. 3.6: Umiestnenie senzora - poloha A

2. Poloha B:

Senzor by bol umiestnený oproti nástupišťu, ktoré by snímal. Poloha B je mierne problematická, hlavne kvôli znemožneniu merania po príchode vlaku. Vlak by znemožnil snímať objekty za ním, teda snímanie by mohlo prebiehať len do jeho príchodu. Taktiež je táto poloha nevýhodná z dôvodu pohybu objektov pred snímačom. Môže sa stať že počas zberu dát sa nejaký objekt postaví priamo pred snímač a tým by stratilo meranie svoj význam. Táto poloha je preto vhodná na dočasné snímanie a iba dotedy, pokiaľ nepríde na opačné nástupište vlak. Teda ak by existovalo miesto, kde by sa ľudia nepohybovali a oproti tomuto miestu by sa nachádzalo nástupište mohli by sme o tomto meraní uvažovať.



Obr. 3.7: Umiestnenie senzora - poloha B

3. Poloha C:

Pri polohe C je senzor umiestnený približne na konci nástupišťu a zároveň s hranou nástupišťu, využijeme tu vlastnosť senzora a tou je dosah. Umožní nám to napríklad merať vzdialenosť prichádzajúceho vlaku a tiež pokryje väčšiu dĺžku nástupišťu. Sme obmedzení jedine dosahom snímača. Ďalším obmedzením je prekrývanie objektov inými objektami a tiež ako u polohy B možné prekrytie snímača v jeho blízkosti. Naopak výhodou tejto polohy by mohlo byť meranie vzdialenosti prichádzajúceho vlaku.



Obr. 3.8: Umiestnenie senzora - poloha C

Pre náš prípad merania budeme používať prevažne polohu B a C. Keďže poloha A, aj keď je najvhodnejšia, nám nebude dostupná, určite by bolo vhodné ju v budúcnosti pri rozširovaní tejto práce zvoliť.

3.4 Softvér

Vybrať softvérové riešenie daného problému nebolo náročné. Porovnal som dostupné nástroje pre prácu z mračnami a zistil, že pre jazyk C++, je dostupná a veľmi obľúbená knižnica. Ide o PCL knižnicu, ktorá má priamo implementované funkcie pre prácu s mračnami. Väčšina iných knižníc, nie je tak priamo zameraná na prácu s mračnami ako PCL. Jazyk C++ je vhodné použiť pre túto realizáciu, aj kvôli jeho využitiu pre 3D.

3.4.1 Knižnica PCL (Point Cloud Library)

Idete o knižnicu na spracovanie 2D alebo 3D obrazu mračna bodov v programovacom jazyku C++. Obsahuje množstvo algoritmov vrátane filtrácie, odhadu vlastností, rekonštrukcie povrchu, registrácie, prispôsobenia modelu či segmentácie. Tieto algoritmy môžeme použiť napríklad na filtráciu odľahlých a nepotrebných hodnôt zo zašumených dát, spojenie viacerých snímok 3D mračien bodov do jednej snímky, segmentáciu častí scény, extrahovanie kľúčových bodov a výpočet dekriptorov na rozpoznávanie objektov

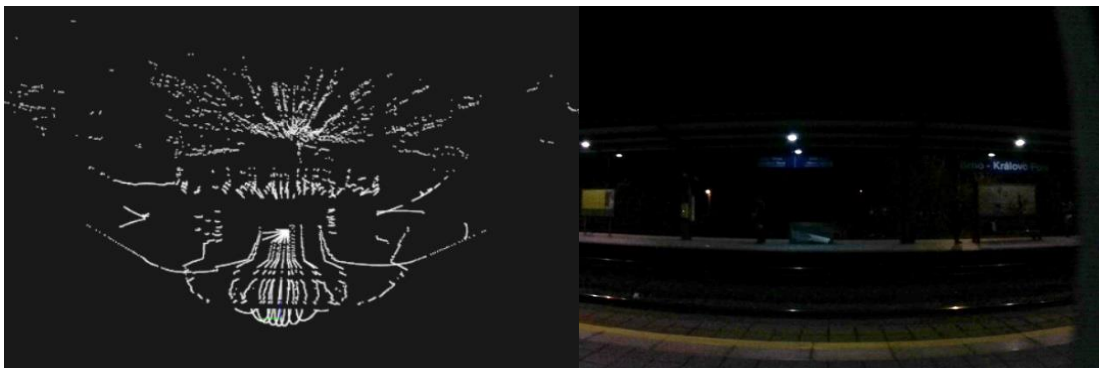
v priestore na základe ich geometrického vzhľadu a vytváranie povrchov z mračna bodov a ich vizualizáciu [1].

PCL softvér je multiplatformový a bol úspešne skompilovaný v systémoch Linux, MacOS, Windows a Android/iOS. Na zjednodušenie vývoja sa PCL rozdeľuje na série menších knižníc alebo kódov, ktoré je možné zostaviť samostatne. Táto modularita je dôležitá pre distribúciu PCL na platformách so zníženými výpočtovými alebo veľkostnými obmedzeniami [5].

Knižnica je open-source a je licencovaná podľa podmienok distribúcie softvéru Berkeley (BSD), preto je voľne použiteľná pre každého. Projekt PCL spája univerzity, výskumníkov, spoločnosti a jednotlivcov z celého sveta a rýchlo sa stáva referenciou pre každého, kto sa zaujíma o 3D spracovanie, počítačové videnie a robotické vnímanie skrz mračná bodov. Jadro PCL je štruktúrované do menších knižníc, ktoré ponúkajú algoritmy a nástroje pre špecifické oblasti 3D spracovania, ktoré možno kombinovať tak, aby efektívne riešili bežné problémy, ako je rozpoznávanie objektov, registrácia mračen bodov, segmentácia a rekonštrukcia povrchu. Inými slovami, nástroje a algoritmy, ktoré poskytuje PCL, Vám umožňujú lepšie sa sústrediť na vaše špecifické oblasti odbornosti [17].

3.5 Nasnímané dáta

Meraní som podstúpil niekoľko. Už v počiatku tejto práce a pri písaní rešerše, som bol zmerať prvé dáta na vlakovej stanici Brno-Královo Pole. Pri tomto meraní a týchto dátach, kde som použil spomínanú polohu senzora B, si môžeme všimnúť v porovnaní s kamerovým záznamom veľkú výhodu LIDAR senzora. Keďže dáta na **Obr. 3.9** boli snímané v neskorších hodinách, viditeľnosť snímku z kamery sú na veľmi nízkej úrovni. V nasledujúcich obrázkoch si ukážeme porovnanie kamerového snímku a snímku zo senzora. Aj keď na prvý pohľad asi nebude jasné o čo konkrétne ide, budem sa snažiť popísať to, čo na danom obrázku vidíme.



Obr. 3.9: Porovnanie mračna bodov s obrázkom z kamery (Brno-Královo Pole)

Na **Obr. 3.10** môžeme vidieť záber, ktorého pohľad na nástupište je mierne natočený, dôvodom bolo úmyselné umiestnenie senzora, pre širší záber nástupišťa. V tomto prípade je obrázok z kamery vo viditeľnejšej podobe ako pri predchádzajúcom.



Obr. 3.10: Porovnanie mračna bodov s obrázkom z kamery (Brno-Hlavní nádraží)

Na **Obr.3.11** je záber z nástupišťa Brno-Židenice pri polohe senzora C. Bohužiaľ pri tomto snímaní sa rozostřila kamera, čo bolo pravdepodobne spôsobené zlým manipulovaním s meracím zariadením.

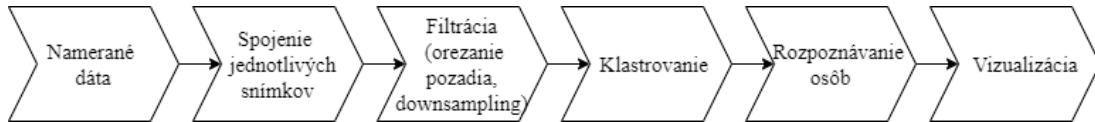


Obr. 3.11: Porovnanie mračna bodov s obrázkom z kamery (Brno-Židenice)

Záznam nasnímaných dát, ktoré som zriadil mal približne 2-3h. Tieto zobrazené dáta majú ukázkový charakter. Obrazová frekvencia senzoru bola nastavená na 10 snímok/s. Za túto jednu sekundu dokázal senzor nasnímať približne 100 000 bodov, to znamená že jeden zobrazený snímok má približne 10 000 bodov. Ako je vidieť na prvý pohľad, z týchto záberov moc vyčítať nedokážeme. Preto je potrebné aplikovať systém spracovania dát na dané snímky.

3.6 Navrhovaný algoritmus

Keďže ide o návrh, algoritmus bude podobný konceptu spracovania dát. V našom prípade nebudeme potrebovať registráciu, pretože nebudeme zriaďovať dáta z viacerých miest. Konceptia algoritmu je zobrazená na *Obr. 3.12*.



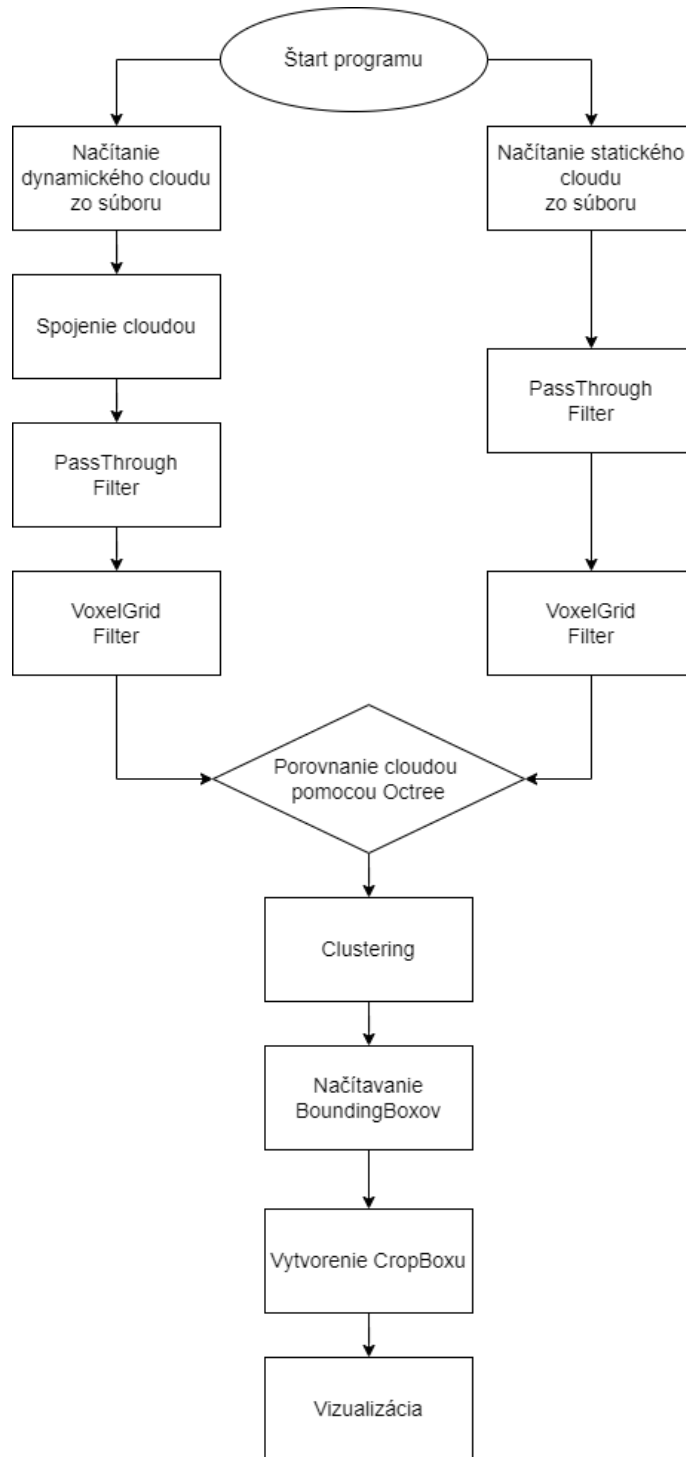
Obr. 3.12: Návrh pre vývojový diagram algoritmu spracovania

Prvý bod navrhovaného algoritmu sú namerané dáta. Účelom tohto bodu bude zriadenie snímku pomocou meraní. Pri meraní a nasledujúcom zobrazení jednotlivých snímkov, som usúdil, že z jednej snímky zo senzora toho veľa vyčítať nedokážem. Aplikovanie ďalšieho bodu zaistí väčší počet mračna bodov v snímanej scéne, taktiež ale môže dôjsť k väčšiemu zašumeniu, pohybujúcich sa objektov. Na základe toho, že tieto dáta sú neupravené a tiež obsahujú zbytočne veľa odľahlých bodov, ktorým nebudem venovať pozornosť, je ďalším bodom filtrácia. Senzor má použiteľný dosah približne 130m, mne postačia dáta na základe meraní približne do 50-60m. Na odstránenie vzdialených dát použijem filter na „orezanie pozadia“. Ďalším filtrom v návrhu je „downsampling“. Ide o ďalší filter, ako plyní z názvu na redukciu bodov v priestore. Po filtrácii potrebujem rozoznať, ktoré body patria k sebe a ktoré nie. Klastrovanie, by malo zabezpečiť a vytvoriť zhluky bodov na základe metód pre klastrovanie a tým bude rozoznávanie objektov, ktoré sa v scéne nachádzajú jednoduchšie. Na základe klastrovania, budem môcť povedať, ktoré body tvoria objekt človeka a zadať na jeho rozoznanie potrebné parametre ako napríklad šírku, výšku prípadne hĺbku. Posledným bodom, aby som si to dokázal všetko zobraziť je vizualizácia.

Návrh algoritmu mi má pomôcť držať sa jeho postupu a naprogramovať funkčný systém. Naprogramovaný funkčný systém spracovania a jeho vývojový diagram je popísaný v ďalšej podkapitole.

3.7 Program

Programovacie prostredie som zvolil Visual Studio 2019. Na *Obr. 3.13* je znázornený vývojový diagram programu. V nasledujúcich podkapitolách si rozoberieme každú jeho časť.



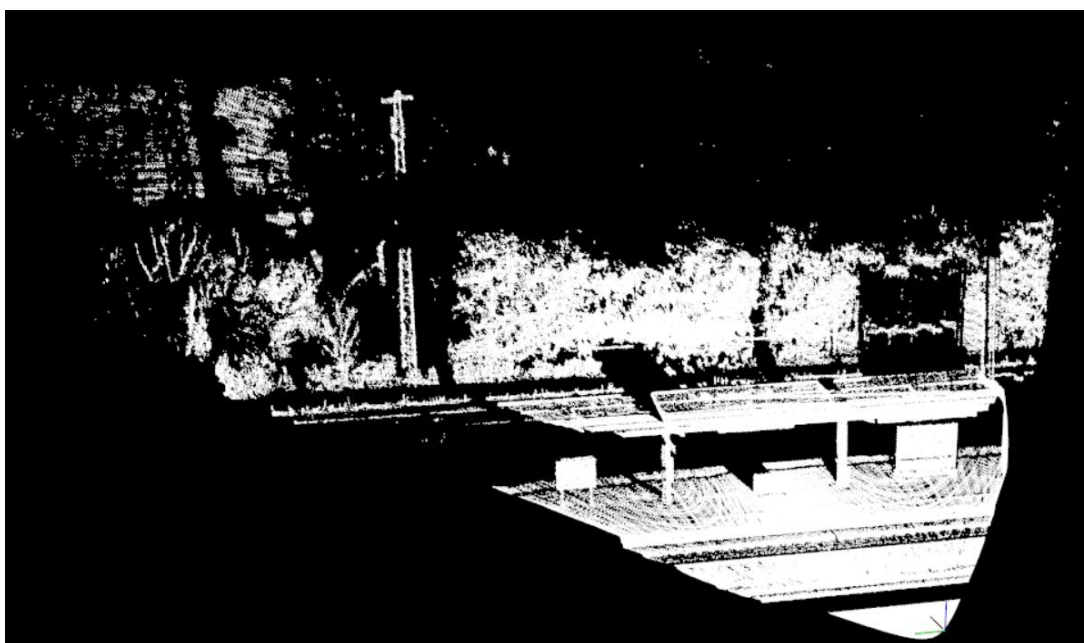
Obr. 3.13: Vývojový diagram programu ObjectDetection_PCL

3.7.1 Štart programu

Pri štarte programu je vhodné zmeniť konfiguráciu z Debug do Release pre rýchlejší chod programu. Na začiatku kompilácie sa nalinkujú všetky zahrnuté knižnice a vykoná sa build a následný debugging.

3.7.2 Načítanie cloudov

Ešte pred načítaním jednotlivých súborov som sa snažil vytvoriť statickú scénu. Tieto scény som vytváral spájaním jednotlivých cloudov do jedného, vo vedľajšom programe, v ktorých som nenašiel ani jeden pohyblivý objekt. Tento cloud, som potom označil za statický. Keď si porovnáme obrázky *Obr. 3.10*, na ktorom je iba jeden snímok v porovnaní s kamerovým záznamom a obrázkom *Obr. 3.14*, je vidieť značný rozdiel v detailoch.



Obr. 3.14: Statická scéna mračna bodov (Brno-Královo Pole)

Načítanie súborov sa delí na 2 časti. Máme vytvorenú statickú scénu, ktorá obsahuje veľké množstvo bodov. Načítanie prebieha v jednom kroku. Načítanie jednotlivých snímkom teda dynamickej scény, prebieha pomocou funkcie „getFileNames“. Táto funkcia má 3 vstupné parametre. Prvým parametrom je samotná cesta kde hľadáme súbory. V programe mám preddefinované cesty k nameraným dátam, potom stačí zvoliť správnu cestu k meraniu, ktoré chcem vizualizovať. Druhým parametrom je typ súboru. Ak by náhodou boli definované súbory s rovnakým názvom, ale z iným typom tak by to mohol byť problém. V našom prípade ide o .pcd súbory preto je v tomto parametri nastavené „pcd“. Posledný parameter funkcie je bool hodnota. Táto

hodnota nám hovorí to, či chceme do výsledku vyhľadávania uložiť celú cestu alebo iba názvy jednotlivých súborov.

Všetky namerané dáta sa pri snímaní ukladali na SD kartu a rozdelené boli do dvoch priečinkov. Prvý priečinkov je s názvom „image“. Prvý uložený súbor tomto priečinku niesol názov „frame000000.jpg“ a obsahoval dáta s kamery. Druhý priečinkov obsahoval dáta mračna bodov. Jeho názov bol „pcl“ a prvý súbor niesol názov „pcl000000.pcd“. Jednotlivé čísla sa po pridávaní snímkou, zvyšovali.

3.7.3 Spájanie cloudov

Ešte pri načítaní cloudou som si zvolil počet týchto cloudou na základe môjho testovania. Keďže jeden snímok obsahuje 10 000 bodov, prišlo mi najvhodnejšie že spojenie šiestich snímok bude ideálne riešenie. Toto spájanie je pomerne jednoduché a funguje cez operátor „+“. Táto funkcia, ktorá spája snímky je zahrnutá v PCL knižnici. Platí tu však obmedzenie, že typ a počet polí v dvoch súboroch údajov sa musia zhodovať.

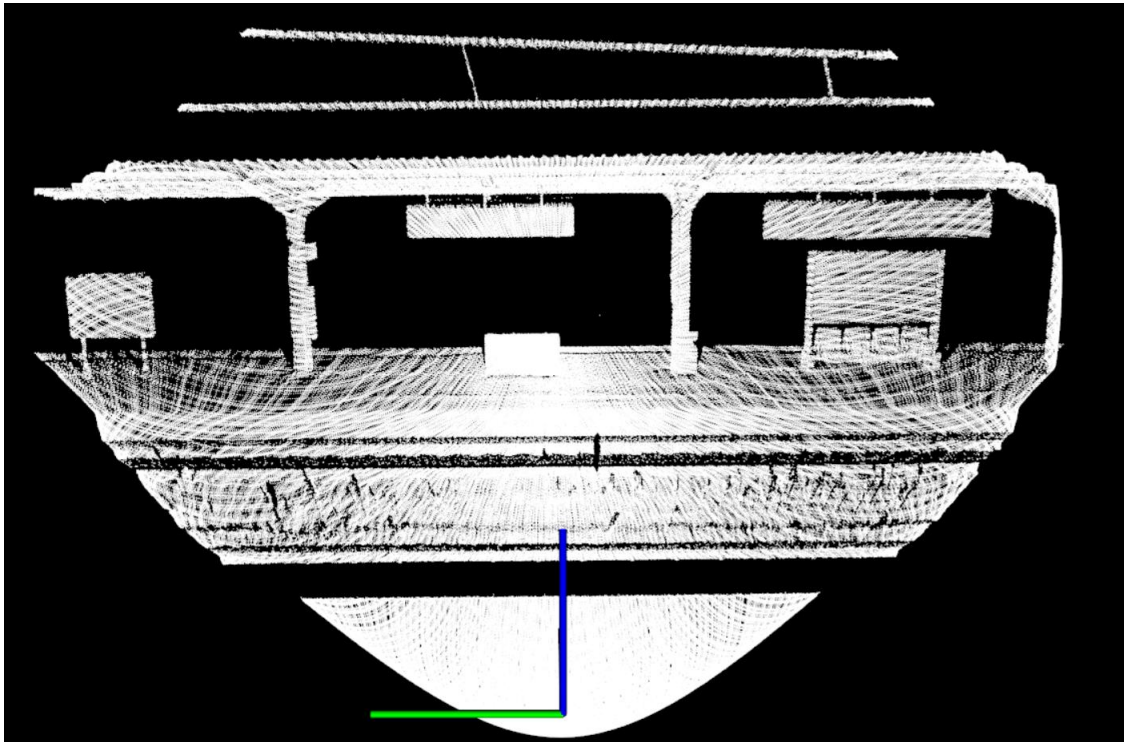
Spájané snímky sa ukladajú do akumulátora a celé to prebieha vo „for cykle“. Vždy sa načíta šesť snímok a všetky sa sčítajú do akumulátora. Aby sa vytvorila dynamická scéna sa po načítaní, prvý snímok odstráni a pričíta sa ďalší v poradí. Týmto máme ošetrené to že počet bodov v dynamickej scéne sa stále mení a vždy sa mení len o jeden snímok, čiže 10 000 bodov.

3.7.4 PassThrough filter

Ďalším krokom programu je filtrácia pozdĺž zadanej osy. Ide v podstate o odrezanie odľahlých hodnôt, ktoré nebudeme potrebovať.

```
*****
// Create the filtering object for KRALOVO POLE
pcl::PassThrough<pcl::PointXYZRGB> pass;
pass.setInputCloud(cloud);
pass.setFilterFieldName("x");
pass.setFilterLimits(0.0, 20.0);
//pass.setFilterLimitsNegative (true);
pcl::PointCloud<pcl::PointXYZRGB>::Ptr
filtered(new pcl::PointCloud<pcl::PointXYZRGB>);
pass.filter(*filtered);
*****
```

V priloženom kóde vytvárame PassThrough objekt z názvom pass, nastavujeme vstupný cloud, filtráciu podľa osy x, a limity odľahlých hodnôt, ktoré boli taktiež nájdené testovaním. V programe existujú 2 varianty nastavení tohto filtru. Prvý variant slúži pre druhé meranie na nádraží Kráľovo Pole a druhý pre tretie meranie na Hlavnom nádraží. Na **Obr. 3.15** je znázornená statická scéna po tomto filtrovaní.



Obr. 3.15: Statická scéna mračna bodov po filtrácii PassThrough (Brno-Královo Pole)

3.7.5 VoxelGrid filter

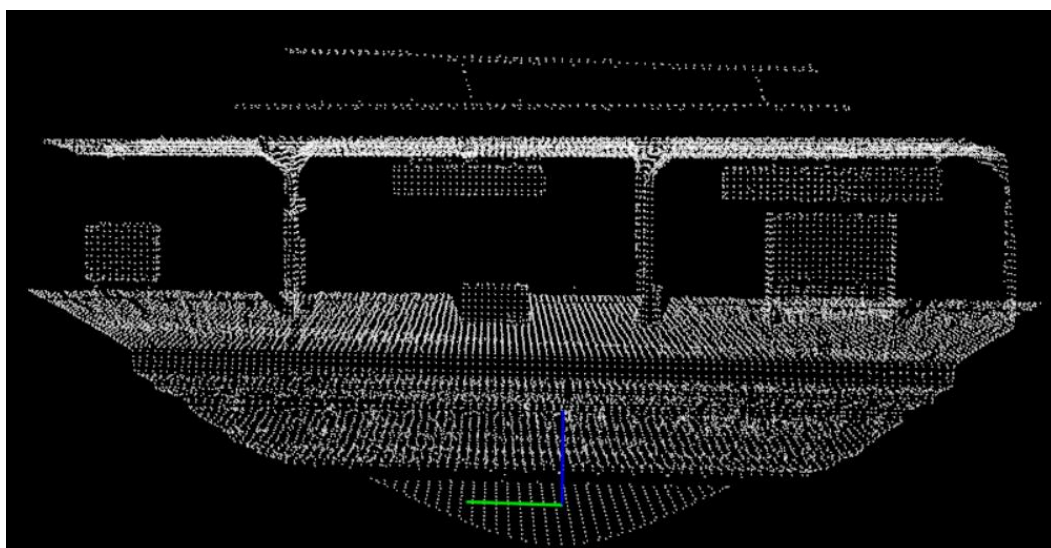
VoxelGrid slúži na redukciu bodov, alebo množiny údajov mračna pomocou voxelovej mriežky. Filter vytvára 3D mriežku (3D box v priestore) a potom v každom takomto boxe sú všetky body aproximované s ich ťažiskom. Existuje aj aproximácia so stredom voxelu, ale tá nám nepredstavuje tak presne podkladový povrch aj keď je rýchlejšia [1].

```

*****
pcl::VoxelGrid<pcl::PointXYZRGB> vg;
pcl::PointCloud<pcl::PointXYZRGB>::Ptr
cloud_filtered(new pcl::PointCloud<pcl::PointXYZRGB>);
vg.setInputCloud(cloud);
vg.setLeafSize(0.15f, 0.15f, 0.15f);
vg.filter(*cloud_filtered);
*****

```

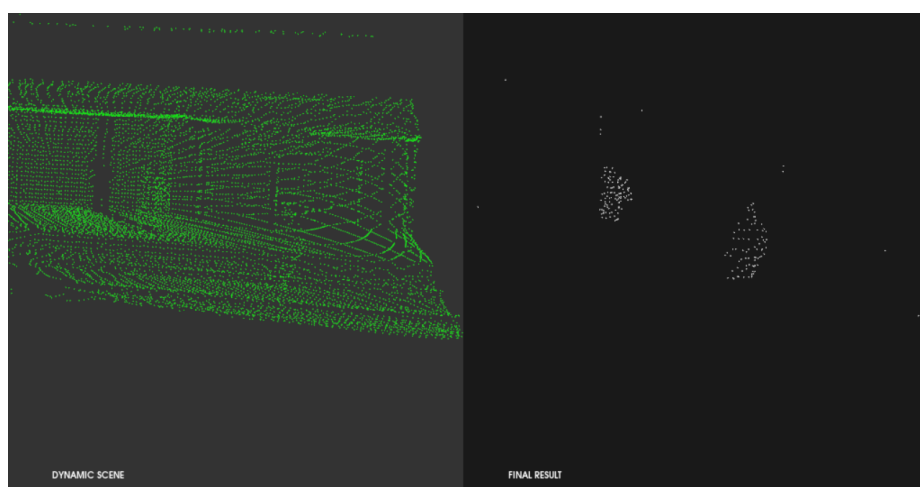
Na nasledujúcom obrázku *Obr. 3.16* vidíme značnú redukciu a usporiadanie bodov. Táto filtrácia je súčasne aplikovaná aj na dynamickú scénu, ale pre lepšiu vizualizáciu toho čo filtre robia som zvolil pridávanie obrázkov zo statickej.



Obr. 3.16: Statická scéna mračna bodov po filtrácii PassThrough a VoxelGrid (Brno-Královo Pole)

3.7.6 Porovnanie cloudou Octree

Po filtrácii prichádza na rad porovnanie. Octree je stromová štruktúra na správu riedkych 3D dát, Používa sa na detekciu zmien medzi viacerými neusporiadanými mračnami bodov. Spätným porovnaním štruktúry, možno identifikovať priestorové zmeny reprezentované rozdielmi pomocou vyhľadávania susedov v rámci voxelov. Rozdiely v priestore teda možno zistiť v reálnom čase pomocou technológie „double buffer“ octree [1]. Najprv načítame statickú scénu a vygenerujeme octree štruktúru. Potom sa načíta dynamická scéna a tiež sa vytvorí štruktúra a tieto štruktúry sa porovnajú. Niektoré body dynamickej scény, hlavne pohybujúce objekty sa nám zobrazujú samostatne na pravej strane obrázku *Obr.3.17*.



Obr. 3.17: Porovnanie Octree (Brno-Hlavní nádraží)

3.7.7 Klastering

Po porovnaní pomocou octree, nám ostávajú niektoré body, samostatne v priestore. Tie body, ktoré majú v ich definovanom okolí viacnásobný počet iných bodov, môžeme klastrovať. Toleranciu okolia bodov v priestore som určil na základe testovania. Nižšie hodnoty tolerancie spôsobili napríklad to, že dva objekty, ktoré boli blízko seba boli brané ako jeden alebo dokonca jeden objekt niekedy ani nezklastrovalo. Ďalej sa nastavila minimálna a maximálna veľkosť klastru.

```
*****
pcl::search::KdTree<pcl::PointXYZRGB>::Ptr
tree(new pcl::search::KdTree<pcl::PointXYZRGB>);
tree->setInputCloud(cloud);

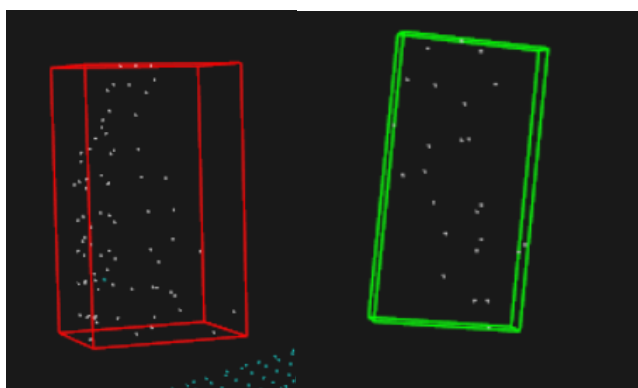
pcl::EuclideanClusterExtraction<pcl::PointXYZRGB> ec;
ec.setClusterTolerance(0.35); // 35cm ideal for person clustering
ec.setMinClusterSize(10);
ec.setMaxClusterSize(1000);
ec.setSearchMethod(tree);
ec.setInputCloud(cloud);
*****
```

3.7.8 Načítavanie BoudingBoxov

BoundingBox je najmenší ohraničujúci rámček pre určitú množinu bodov v N rozmeroch, v ktorom ležia všetky body [24]. V našom prípade budeme tieto boxy, používať na ohraničenie klastrovaných objektov.

```
*****
if (cluster->empty())
return;
//Get cluster ranges
pcl::PointXYZ min, max;
pcl::getMinMax3D(*cluster, min, max);
///// Add bounding cube
viewer->addCube(min.x, max.x, min.y, max.y, min.z, max.z, r, g, b, id,
v3);
*****
```

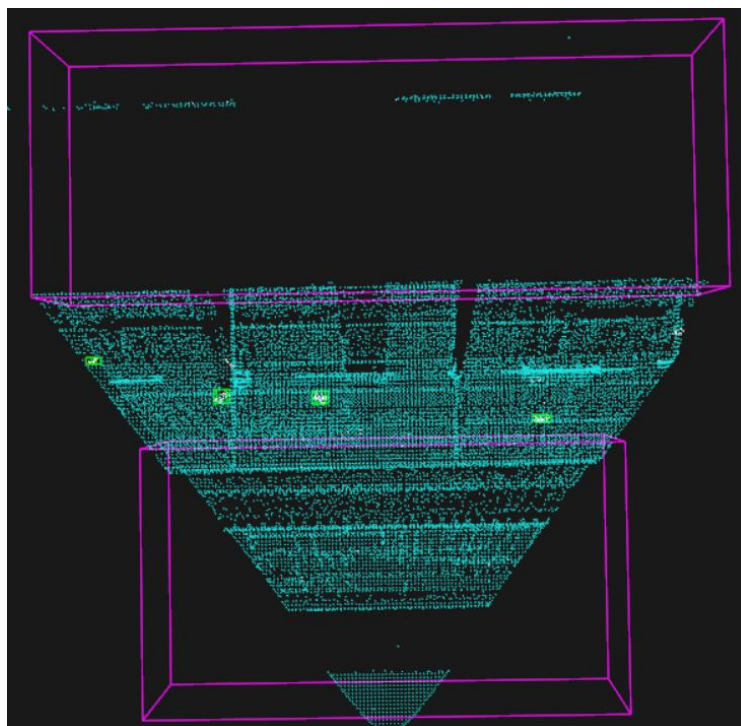
Ako prvé vezmeme klastrovaný objekt, pomocou preddefinovanej funkcie v PCL knižnici getMinMax3D, získame minimálne a maximálne hodnoty x,y,z súradníc tohto klastru a vytvoríme pomocou týchto bodov kocku. Tým sa nám daný objekt ohraničí a vytvorí okolo neho BoundingBox. Pri vniknutí do nebezpečných oblastí, sa tento BoundingBox zobrazí v oranžovej alebo červenej farbe, mimo toho je zelený.



Obr. 3.18: Zobrazenie BoundingBoxov

3.7.9 Vytvorenie CropBoxu

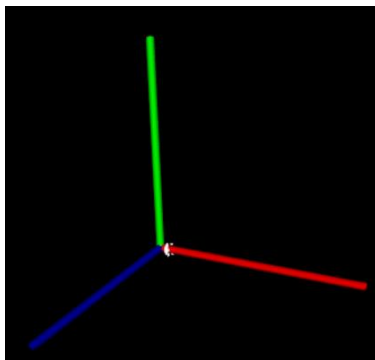
CropBox je označenie pre nebezpečnú oblasť. Týchto oblastí je možné definovať viac. Ide v podstate taktiež o geometrický útvar podobný kváдру, s tým že môžeme meniť hodnoty tohto boxu. Cropbox sa načítava z textového súboru priloženého na miestach kde sa nachádzajú aj statické scény. Má tieto parametre X, Y, Z súradnice, šírku výšku a hĺbku a rotáciu podľa osy z, ktorá sa prepočítava cez Eulerove uhly pomocou funkcie „euler2Quaterion“ v programe.



Obr. 3.19: Zobrazenie CropBoxov

3.7.10 Vizualizácia

Samotná vizualizácia, ktorá je tiež preddefinovaná v knižnici sa používa veľmi spoľahlivo. Súradnicový systém je označený podľa farieb RGB, ktoré sú priamoúmerné osiam. Os x je v červenej farbe, os y v zelenej a os z v modrej. Vizualizácia nám umožňuje meniť počet zobrazovacích okien, tiež je tu možnosť počiatočného nastavenia pohľadu, mračno bodov môžeme zobrazit' farebne a veľa iných funkcií.



Obr. 3.20: Zobrazenie osí vo vizualizácii

3.8 Testovanie systému

Pri vývoji každého systému, je potrebné na konci tento systém otestovať. Systém spracovania dát a celkovej vizualizácie, bol testovaný na nameraných dátach, ktoré boli nasnímané na nádražiach v Brne. Konkrétne išlo o nádražia Brno-Královo Pole, Brno-Židenice a Brno-Hlavní nádraží. Merania neboli vždy za ideálnych podmienok, napríklad kvôli počasiu, počtu osôb na nástupištiach, alebo správnom umiestnení meracieho zariadenia. Meracie zariadenie by malo byť staticky umiestnené na jednom mieste. Pri jeho pohybe by systém neplnil dobre svoj algoritmus. Tieto podmienky nám znižovali kvalitu nameraných snímok, či už z kamerového snímača alebo z LIDARu. Výpočtovú náročnosť programu, som overil pri jeho spúšťaní. Pri väčších vstupných dátach mračna bodov, hlavne statických scén, trvala kompilácia, celkové načítanie a spracovanie dlhší čas. Pri filtrácii išlo o správne nastavenie tolerancií. V prípade merania na nádraží Brno-Královo Pole, bola filtrácia odľahlých hodnôt nastavená podľa osy x a má hodnotu 20 metrov, pretože snímaná oblasť ďalej boli nepotrebné dáta. Naopak pri meraní na Hlavnom nádraží bola táto hodnota, tiež podľa osy x nastavená na 60 metrov. Pri nesprávnom nastavení oboch filtrov, nastávali scenáre ako napríklad nedokonalé ofiltrovanie odľahlých hodnôt, alebo príliš veľký počet bodov. Porovnávanie bolo závislé na predchádzajúcich nastaveniach. Pri tomto porovnávaní dynamickej a statickej scény, vznikali body v priestore, ktoré boli pre nás zbytočné. Pri následnom klastrovaní, ktorý tiež závisel od tolerancií okolitých bodov sme sa snažili detegovať prevažne osoby.

Týmto testovaním sme dospeli k okoliu bodov ideálnych pre klastrovanie osôb, približne 35 cm. Pri tomto okolí sme dokázali detegovať osobu aj v pohybe. Pri rýchlejšom pohybe osôb nám klastering v niektorých prípadoch, nedokázal správne určiť daný zhluk bodov. Problémy tiež nastávali pri prekryvaní objektov inými objektami. Tieto problémy by sa dali vyriešiť, napríklad spojením viacerých snímkou, avšak mohlo by dôjsť k vyššiemu zašumeniu dát. Vykresľovanie boundingboxov záviselo taktiež na predchádzajúcom kroku. Keďže išlo o hľadanie maximálnych a minimálnych hodnôt jednotlivých zhlukov podľa osí, vždy sa ohraničil tento box podľa týchto bodov. Niekedy to zlyhávalo pretože, zhluk nebol dostatočný na to, aby sa dokázal prispôbiť do tohto boxu. Tento box potom označoval neidentifikovateľný objekt a nevedeli sme určiť či ide o osobu alebo napríklad príručnú batožinu. Na druhej strane pri pomalom pohybe, prípadne žiadnom pohybe sa tento box dokázal vykresliť veľmi slušne. Následné vniknutie do nebezpečnej zóny cropboxu, zmenilo farbu boundingboxu na oranžovú, prípadne červenú.

4. ZÁVER

Ak by som to mal zhrnúť, cieľom bakalárskej práce, bolo navrhnuť systém k detekcii objektov v mračne bodov na vlakovom nástupišti. Tieto body boli reprezentované v súradnicovom systéme x, y, z .

Všeobecný opis mračna bodov a senzory na jeho tvorbu máme opísané hneď na začiatku. Taktiež sa dozvedáme využitie mračna bodov a oblasti, kde je populárne no my sa prevažne zaujíname o oblasť vlakovej dopravy. Spolu sním sa potrebujeme oboznámiť aj s nástrojmi na jeho tvorbu. Práve toto bolo našou úlohou. Zvoliť správny senzor na jeho zber. Pri voľbe senzora, sme vybrali LIDAR, kvôli jeho výhodám hlavne v rozlíšení oproti RADAR-u a taktiež, kvôli jeho dostupnosti. V porovnaní napríklad s kamerou majú tieto senzory značnú výhodu keďže nezachytávajú priamo obraz, ale iba body v priestore. Tiež ich podstatnou výhodou je viditeľnosť obrazu v noci.

V rámci rešerše bolo treba oboznámiť sa s dostupnými metódami na spracovanie. V tejto rešerši v kapitole 1.2, sa venujeme metódam filtrácie rôznych zašumených údajov, vzniknutými nedokonalosťou snímačov, prípadne podmienkami daného merania. Ďalšou skupinou metód sú metódy na registráciu. Tieto metódy nie sú aplikované v práci, lebo ide o merania z viacerých miest a ich vzájomné prepojenie. Posledná skupina metód patrí metódam segmentácie. Ide o metódy priradovania zozbieraných dát do segmentov a následne ich rozlíšenie od iných segmentov.

Ďalšou úlohou bolo dáta nazbierať. Meraní sme podstúpili niekoľko a na rôznych nádražiach v Brne. Išlo o merania na nádraží Brno-Královo Pole, Brno-Židenice a Brno-Hlavní nádraží. Záznamy nameraných dát, ktoré sme podstúpili majú približne 2-3h. Systém spracovania dát má niekoľko úrovní. Prvá úroveň pozostávala zo samotného návrhu tohto spracovania. Pre návrh je vytvorený algoritmus v kapitole 3.6. Ďalším krokom bol samotný návrh programu. Vývojové prostredie sme zvolili Visual Studio 2019 a programovací jazyk C++. Z dostupných knižníc na prácu s mračnami sme použili PCL knižnicu. V tejto knižnici sú integrované funkcie, užitočné pre prácu s mračnami bodov. Návrh programu a jeho jednotlivé časti sú dostupné v kapitole 3.7. Tento systém sme následne implementovali na jednotlivé merania a otestovali. Systém rozpoznáva objekty, prevažne ľudí, ktorí sa nachádzajú na vlakovom nástupišti a deteguje ich pohyb. Služi nám ako bezpečnostný systém, ktorý vyhodnocuje to, či sa dané objekty približujú k nebezpečnej oblasti medzi koľajami a nástupnou hranou. Keď sa priblížia alebo prekročia pomyselnú hranicu, systém to vyhodnotí ako nebezpečný stav. Vyhodnotenie tohto systému a jeho funkčnosti by som označil za úspešný, aj keď nie dokonalý. Určite by sa do budúcnosti dali spraviť určité vylepšenia, ako napríklad rôzne polohy senzorov, vytvorenie bezpečných hraníc pred príchodom vlaku a podobne. Na tieto vylepšenia, som bohužiaľ nemal potrebné prostriedky a čas.

LITERATÚRA

- [1] Rusu, Radu BOGDAN a Steve COUSINS. 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation (ICRA)*. May 9-13. Shanghai, China, **2011**. Dostupné také z: <https://pointclouds.org/>
- [2] LUNDQVIST, Annika a Jonas KARLSSON. Deploying interactive machine learning. : *ARTICLES / TECHNOLOGY* [online]. **2020** [cit. 2022-01-03]. Dostupné z: <https://www.annotell.com/articles/deploying-interactive-machine-learning/>
- [3] CROPP MRICS, Charlie. *What are point clouds? Understanding how reality capture is stored and manipulated* [online]. 2021, 2021 [cit. 2022-01-03]. Dostupné z: <https://info.vercator.com/blog/point-cloud>
- [4] THOMSON, Charles. *What are point clouds? 5 easy facts that explain point clouds* [online]. 2019 [cit. 2022-01-03]. Dostupné z: <https://info.vercator.com/blog/what-are-point-clouds-5-easy-facts-that-explain-point-clouds>
- [5] HANI, Dhiya. LiDAR Vs RADAR: Detection, Tracking, And Imaging. : *Sensors - Lidar - Autonomous Vehicles* [online]. 2021, 19.4.2021 [cit. 2022-01-03]. Dostupné z: <https://www.wevolver.com/article/lidar-vs-radar-detection-tracking-and-imaging/>
- [6] LiDAR. *Hamamatsu* [online]. [cit. 2022-01-03]. Dostupné z: <https://www.hamamatsu.com/eu/en/applications/lidar/index.html>
- [7] MID-70: Livox. *LivoxTech* [online]. [cit. 2022-01-03]. Dostupné z: <https://www.livoxtech.com/mid-70>
- [8] WANG, Qian, Yi TAN a Zhongya MEI. *Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications*. 2019, , 1-21. Dostupné také z: <https://link.springer.com/article/10.1007%2Fs11831-019-09320-4>
- [9] BUENO, M., H. GONZÁLEZ-JORGE, J. MARTÍNEZ-SÁNCHEZ a H. LORENZO. *Automatic point cloud coarse registration using geometric keypoint descriptors for indoor scenes: Automation in Construction*. **2017**, 134-148. ISSN ISSN 0926-5805. Dostupné také z: <https://doi.org/10.1016/j.autcon.2017.06.016>
- [10] FILIN, Sagi. *Surface clustering from airborne laser scanning data: In International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, 3A*. **2002**, 119/124.
- [11] FILIN, Sagi a Norbert PFEIFER. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS Journal of Photogrammetry and Remote Sensing*. Elsevier, 2006, s. 71-80.
- [12] MACQUEEN J (1967) Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol 1, Oakland, CA, USA, 281–297
- [13] NGUYEN A, LE B (2013) 3D point cloud segmentation: A survey. 2013 6th IEEE conference on robotics, automation and mechatronics (RAM). IEEE

- [14] JÄHNE, Bernd a Horst HAUSSECKER, 2000. Computer vision and applications: a guide for students and practitioners. San Diego: Academic Press. ISBN 01-237-9777-2.
- [15] MACHINE Vision Group, UAMT VUTBR. *Multisensor Traffic Data Logger: MVG-MTDL*. Brno, 2021. Dostupné také z: http://vision.uamt.feec.vutbr.cz/data/MVG_2021_MTDL_tech_spec.pdf
- [16] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 167-192, Marec 1988, doi: 10.1109/34.3881.
- [17] A. Aldoma et al., "Tutorial: Point Cloud Library: Three-Dimensional Object Recognition and 6 DOF Pose Estimation," in *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80-91, Sept. 2012, doi: 10.1109/MRA.2012.2206675.
- [18] Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in construction*, 19(7), 829-843. <https://doi.org/10.1016/j.autcon.2010.06.007>
- [19] VISION, team. What is a stereo vision camera?. E-consystems [online]. 2018, 10.10.2018 [cit. 2022-05-18]. Dostupné z: <https://www.e-consystems.com/blog/camera/technology/what-is-a-stereo-vision-camera-2/>
- [20] Raspberry Pi High Quality Camera. Raspberry Pi [online]. 2022 [cit. 2022-05-18]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/>
- [21] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [22] Derpanis, Konstantinos G.. "Overview of the RANSAC Algorithm." (2005) [cit. 2022-05-18]. Dostupné z: <https://www.semanticscholar.org/paper/Overview-of-the-RANSAC-Algorithm-Derpanis/e69e7b09a62727a05a4547249f15d99aeca9172f>
- [23] ČERVENÝ, Patrik. Detekce 3D objektů v mračnu bodů [online]. Brno, 2022 [cit. 2022-05-19]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/138417>. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Tomáš Zemčík.
- [24] Minimum bounding box. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2022-05-19]. Dostupné z: https://en.wikipedia.org/wiki/Minimum_bounding_box

ZOZNAM PRÍLOH

PRÍLOHA A – ZDROJOVÝ KÓD JE ULOŽENÝ NA PRILOŽENEJ SD KARTE

PRÍLOHA B – NAMERANÉ DÁTA SÚ ULOŽENÉ NA PRILOŽENEJ SD KARTE

PRÍLOHA C – README SÚBOR JE ULOŽENÝ NA PRILOŽENEJ SD KARTE