



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**POŘÍZENÍ 3D MODELU ULICE POMOCÍ LASEROVÉHO
SKENOVÁNÍ**

BUILDING 3D MODEL OF STREETS USING LASER SCANNING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL JAKABOVIČ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VEĽAS MARTIN

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Jakabovič Michal**

Obor: Informační technologie

Téma: **Pořízení 3D modelu ulice pomocí laserového skenování
Building 3D Model of Streets Using Laser Scanning**

Kategorie: Počítačová grafika

Pokyny:

1. Zorientujte se v problematice počítačového vidění v robotice a v základních technikách rekonstrukce 3D scény pomocí laserového skenování a senzoru Velodyne LiDAR.
2. Seznamte se s platformou ROS (Robot Operation System) a jejími možnostmi pro dynamickou tvorbu 3D mapy okolí na úrovni mračna bodů.
3. Vyberte vhodné metody a nástroje a navrhnete modul do systému ROS schopný postupně vytvářet 3D model okolního prostředí s využitím senzoru Velodyne LiDAR.
4. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
5. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
6. Vytvořte stručný plakát prezentující vaši práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Veřas Martin, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Súčasný rýchlo rozvíjajúci sa trend nových technológií metód zberu údajov zo skutočného priestoru prostredníctvom mobilných laserových snímacích zariadení sa čoraz častejšie uplatňuje v získavaní virtuálnych 3D modelov objektov, miest, ulíc a krajín. Cieľom mojej práce bolo zhotovenie nástroja tvorby trojrozmerných máp v obytných častiach miest. Výhodou tohto nástroja by mala byť schopnosť detekovať, ignorovať a zvýrazniť nasnímané obyvateľstvo a iné pohyblivé objekty. Program je zhotovený v systéme Robot Operatin System, ROS za pomoci nástroja octomap.

Abstract

With the current advancements in environmental data collection, laser scanning devices are becoming more and more relevant as a means of creating virtual models of objects, streets, cities, and landscapes. The purpose of my work is to create a device for constructing 3D maps of inhabited urban and suburban areas. The tool will be able to detect, ignore, and highlight scanned people and other moving objects. The program will be built in the Robot Operating System (ROS) framework, utilizing the octomap tool.

Klíčové slová

3D model, LiDAR, robotika, oktálový strom, oktomapa, ROS, detekcia objektov

Keywords

3D model, LiDAR, robotics, octree, octomap, ROS, object detection

Citácia

JAKABOVIČ, Michal. *Pořízení 3D modelu ulice pomocí laserového skenování*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Velas

Pořizení 3D modelu ulice pomocí laserového skenování

Prehlásenie

Prehlasujem, že som bakalársku prácu s názvom "Pořizení 3D modelu ulice pomocí laserového skenování" vypracoval samostatne pod vedením pána Ing. Martina Veľasa. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Michal Jakabovič
18. mája 2017

Podakovanie

Chcel by som poďakovať svojmu školiteľovi pánovi Ing. Martinovi Veľasovi za odbornú pomoc a usmernenie pri písaní mojej práce a za cenné rady.

Obsah

1 Úvod	3
2 Modelovanie 3D priestoru	4
2.1 Spôsob získavania dát	7
2.1.1 Optické skenovanie	8
2.1.2 Laserové skenovanie	9
2.1.3 Porovnanie	11
2.2 Octree	11
2.3 Detekcia chýb a pohyblivých objektov laserovým skenovaním	12
2.3.1 Pravdepodobnostný model laserového diaľkomera	12
3 Návrh a Implementácia	16
3.1 Ovládanie	16
3.2 Implementácia	16
3.3 Robot Operating System	17
3.3.1 Štruktúra ROSu	18
3.3.2 Dôležité moduly	19
3.4 Octomap	20
3.4.1 Pravdepodobnosť obsadenia	21
3.5 Implementačné riešenie	21
4 Zhodnotenie	24
4.1 Nástroj ReCap	24
4.2 Testovanie	25
5 Záver	27
Literatúra	28
A Obsah príloženého pamätového média	30
B Dodatočné snímky programu	31
C Plakát	32

Zoznam obrázkov

2.1	Polygonálny model	5
2.2	Drôtový model	5
2.3	Model vytvorený konštruktívnou geometriou	6
2.4	Boundary Representation model	6
2.5	Dekompozičný model z voxelov	6
2.6	Model vytvorený šablónou, krivkami	7
2.7	Zobrazenie metód 3d skenovania[14]	8
2.8	Optický skener	9
2.9	Laserový skener na krátke vzdialenosti	10
2.10	Auto spoločnosti Google s LiDAR senzorom	10
2.11	Štruktúra októlového stromu	12
2.12	Parametre laserového skenera	13
2.13	Detekcia falošného merania	14
2.14	Vyhodnocovanie súvisiacich celkov	15
3.1	Senzory Velodyne	17
3.2	Ukážka funkčnosti správ v systéme ROS	19
3.3	Strom transformácií na robotovi PR2	20
3.4	Postupné znižovanie rozlíšenia októlového stromu	21
4.1	Ukážka tvorby voxelového modelu časť 1.	25
4.2	Ukážka tvorby voxelového modelu časť 2.	26
B.1	Ukážka tvorby voxelového modelu časť 3.	31
B.2	Ukážka tvorby voxelového modelu časť 4.	31

Kapitola 1

Úvod

V súčasnosti množstvo miest a obcí na svoju prezentáciu čoraz častejšie požaduje vytvorenie virtuálneho 3D modelu. 3D model mesta predstavuje digitálny model s využitím trojrozmernej reprezentácie objektov. Rozvoj modelovania úzko súvisí s rozvojom rôznych vedných odborov a taktiež modernejších technológií.

Zber dát pre tvorbu 3D modelov je realizovaný rôznymi typmi skenovacích technológií, či už kontaktných alebo bezkontaktných. V popredí sú však bezkontaktné snímacie metódy, ktoré sa vyznačujú svojou rýchlosťou. Vo všeobecnosti sa konštatuje, že 3D skenovacie technológie sú využiteľnejšie pre svoju jednoduchosť, rýchlosť a presnosť prenesenia reálneho objektu do virtuálneho obrazu.

V práci som sa zameril na priblíženie laserového skenovania objektov, ktorého základom je využitie laserového lúča na zistenie polohy bodov v trojrozmernom priestore. Spracovaním nameraných údajov získaných z trojdimenzionálneho priestorového mapovania lokality môžeme vytvoriť 3D model mesta, obce, ulice, krajiny a pod..

Cieľom práce je zhotoviť program schopný vytvárať 3D modely vonkajších priestorov získaných pozemným laserovým skenovaním použitím Velodyne LiDAR (z angl. Light Detection and Ranging), detekovanie a ignorovanie pohyblivých objektov ako chodcov a hýbajúcich sa áut a následne zostrojiť prehliadateľný 3D model. Pri spracovaní práce vychádzame predovšetkým z odborných článkov, virtuálnej knižnice a internetových zdrojov zameraných na problematiku 3D modelovania, skenovania a práce LiDARu

Laserové skenovanie asi najčastejšie slúži na zachytávanie reálnych objektov do ich digitálnej trojrozmernej podoby. Je to taktiež rýchly a efektívny spôsob, ako zmapovať väčšie priestory, či už v budovách alebo zmapovať veľké plochy v otvorenom priestranstve.

Kapitola 2

Modelovanie 3D priestoru

3D modelovanie je proces vytvárania matematického trojrozmerného objektu za pomoci špecializovaného softvéru. Výhodou 3D modelovania je zjednodušenie vizualizácie objektu a výsledného návrhu riešenia aj pre laika. Má vyšší stupeň dôveryhodnosti a presnosti ako iné spôsoby preentácie. Existuje niekoľko typov trojdimenzionálnych modelov:

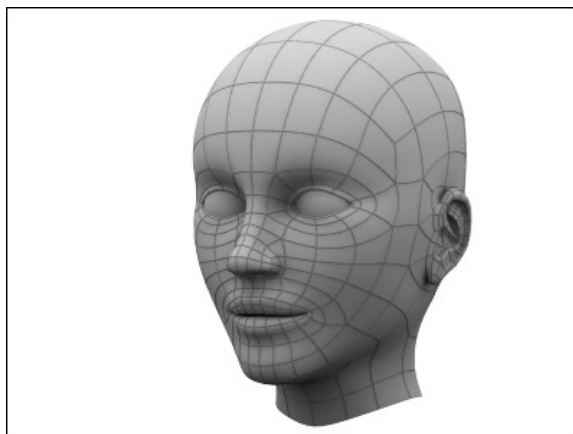
- Analytická reprezentácia
- Polygonálny model
- Drôtový model
- Plošný model
- Objemový model
- Voxelový model
- Model pomocou kriviek

Analytická reprezentácia

Model skladajúci sa z bodov, kriviek a spline plôch. Objekty bývajú reálne a pamäťovo nenáročné. Nepoužíva sa na zobrazovanie scén v reálnom čase nakoľko je to veľmi výpočtovo náročné. Nevýhodou je neustála nutnosť kontrolovať uzavretosť modelovaného objektu.

Polygonálny model

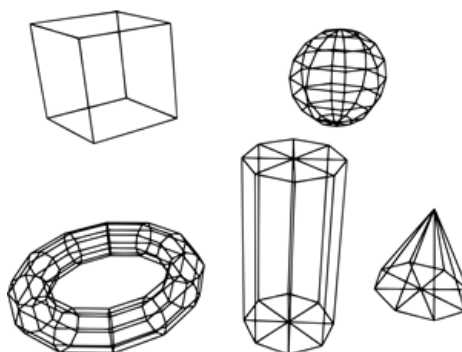
Jeho základom je vertex, bod v trojrozmernom priestore. Spájaním vertexov získavame úsečky, ktoré tvoria útvary zvané polygony. Tento model je v súčasnosti najpoužívanejší, z dôvodu prirodzenej podpory hardvérom. Je najrýchlejšie zobraziteľný. Pri vizualizácií ostatných modelov prebehne ich transformácia na polygonálny model.



Obr. 2.1: Polygonálny model

Drôtový model

Tento model sa skladá iba z bodov a kriviek spájajúcich body. Nevýhodou používania tohto typu je, že si nenesie údaje o stenách, medzi jednotlivými krivkami a vnútornom obsahu. Využíva sa hlavne pri simulovaní pohybu nástrojov a rôzne rýchle a výpočtovo nenáročné zobrazenie objektov.



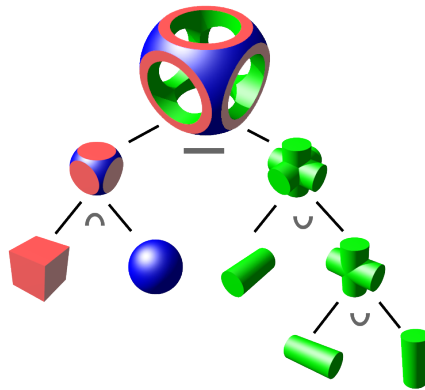
Obr. 2.2: Drôtový model

Plošný model Objekty sú tvorené bodmi, hranami a stenami, ktoré definujú plochu. Jeho využitie je hlavne v leteckom, lodiarском a automobilovom priemysle. Narozdiel od drôtového modelu sa tu uchováajú aj informácie o ploche objektu, no stále chýbajú informácie o objeme.

Objemový model

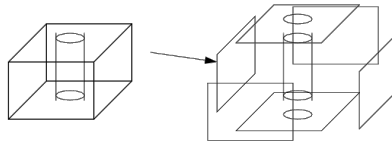
Je zostavený iba z 3D primitív, napríklad s pomocou hraníc alebo geometrických telies (Konštruktívna geometria).

- Konštruktívna geometria (CSG) Spočíva vo využívaní booleovských operácií a transformáciach jednoduchých geometrických objektov. Objekty sú v pamäti reprezentované pomocou stromu



Obr. 2.3: Model vytvorený konštruktívnou geometriou

- B-rep Objekt je popísaný prostredníctvom svojho povrchu, hranicami. Takéto geometrické objekty sú prepojené v špeciálnej dátovej štruktúre podľa vzájomného geometrického vzťahu.

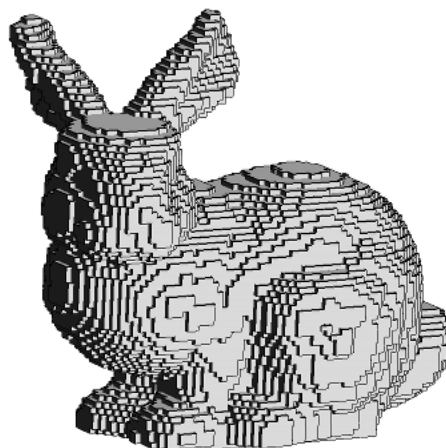


Obr. 2.4: Boundary Representation model

- Hybridný model Kombinácia *B-Rep* a *CSG*.

Voxelový model

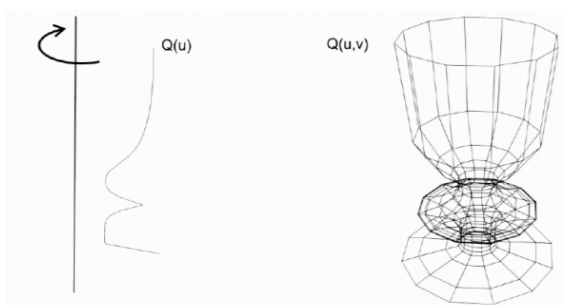
Tiež sa nazýva **dekompozičný**. Jedná sa o diskretný typ modelu. Narozdiel od vyššie spomenutých sa skladá z malých kociek, z *voxelov* 3D pixelov. Vizualizácia voxelových modelov je zväčša náročná na pamäť a procesorový čas.



Obr. 2.5: Dekompozičný model z voxelov

Model pomocou kriviek

Vzniká transformovaním, rotovaním a posúvaním 2D alebo 3D kriviek.[13]



Obr. 2.6: Model vytvorený šablónou, krivkami

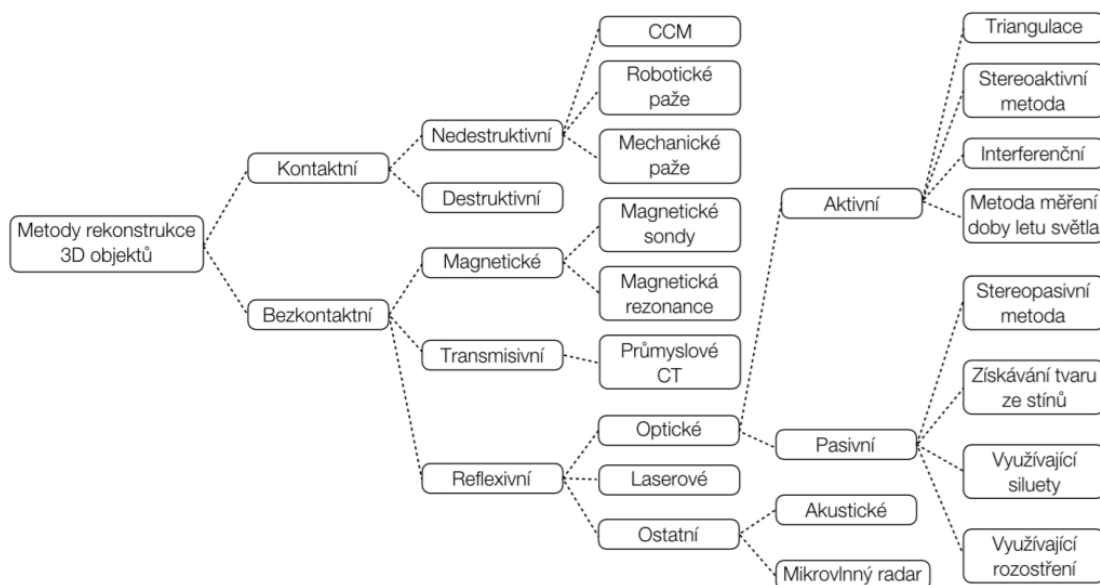
Pokrok vo vytváraní 3D máp vzbudzuje čoraz väčší záujem o ich využitie v rôznych odvetviach. Existuje viacero spôsobov získavania dát o trojrozmerných priestoroch, z ktorých najefektívnejšie je zber dát laserovým alebo optickým skenerom. Uvedené metódy sú založené na diaľkovom získavaní údajov na základe merania vzdialenosti medzi cieľovým objektom a senzorom najčastejšie umiestneným na pohybujúcich sa vozidlách.

Práca je zameraná na počítačové videnie senzorov robotov, a na následne vytvorenie 3D máp z nazbieraných dát.

2.1 Spôsob získavania dát

Základom pre tvorbu priestorového modelu a následne vyhotovenie jeho 3D varianty je nevyhnutné získanie informácií o polohe bodov v trojdimenzionálnom priestore, z ktorých sa vyskladajú plochy a následne celý objekt. V dnešnej dobe existuje mnoho spôsobov získavania dát pre vytváranie 3D modelov. Zariadenia pre zachytávanie tvarov a textúr objektov a jeho následného spracovania sa nazývajú 3D skenery. Podľa literatúry existuje množstvo druhov delenia 3D skenerov. V práci som sa zamerlal na bezkontaktné, reflexívne riešenie.

Medzi metódy použiteľné pre modelovanie ulíc a máp patrí laserové a optické skenovanie.



Obr. 2.7: Zobrazenie metód 3d skenovania[14]

Výhody laserového skenovania:

- automatický a systematický zber bodov
- vysoká hustota bodov a rýchlosť namerania
- určenie súradníc

Najčastejšie použitie 3D skenerov v praxi:

- statické
- mobilné, umiestnené na statívy určené na krátke a stredné vzdialenosti
- letecké skenery pre topografické aplikácie

[15]

2.1.1 Optické skenovanie

Snímanie skenovaného objektu prebieha z niekoľkých uhlov pomocou optického zariadenia. Pri každej zmene natočenia cieľového objektu, ktoré sa vykoná buď ručne alebo za pomoci polohovacieho zariadenia dôjde k získaniu fotografických snímok z mnohých rôznych uhlov pohľadu. Tieto dáta sa následne pomocou algoritmov spracovávajú na digitalizovaný model metódou aproximácie. Väčšia časť skenerov vytvára 3D modely využitím polygonálnych modelov, spline kriviek alebo iných štandardných geometrických entít.

Pred samotným skenovaním sa doporučuje na teleso umiestniť niekoľko orientačných bodov, značiek pre presnejšie spájanie obrázkov do trojrozmerných objektov. Orientačné body, nazývané tiež ako referenčné body sa umiestňujú priamo na povrch snímaného objektu. Môže ich byť až niekoľko tisíc. Nevyhnutnou súčasťou pri vytváraní snímok bodov je kalibračná tyč, ktorá musí byť umiestnená vedľa snímaného objektu.



Obr. 2.8: Optický skener

2.1.2 Laserové skenovanie

Laserové skenovanie je jednou z najmladších metód snímania dát v problematike počítačového videnia. Spočíva vo vyslaní laserového lúča kolmo na snímaný objekt, odrazení sa laserového lúča od objektu a následného vrátenia sa na senzor snímacieho zariadenia, kde dôjde k jeho vyhodnoteniu. Vyhodnotením dĺžky presunu vyžarovaného laserového paprsku od jeho vyslania až po jeho vrátenie je možné zistiť informáciu o rozmere objektu v smere letu paprsku. Informácia o zakryvení telesa vyplýva z uhlu pod akým sa paprsok vracia späť do zariadenia. Spojením oboch základných informácií o dĺžke a zakryvení lúča skener získa presnú polohu bodov v priestore, ktorú následne uchováva v pamäti počítača. V bežnej praxi sa využívajú skenery určené na menšiu vzdialenosť na vytváranie trojrozmernej vizualizácie slúžiace na prezentačné účely ktorých cieľom je uľahčenie orientácie a predstavivosti modelu vo virtuálnom priestore. Skener zosníma informácie o povrchu celého telesa jeho obkružným, alebo otočením meraného objektu po vlastnej osi. Kvalita nameraného povrchu je daná hustotou pokrytia povrchu meraného objektu laserovými lúčmi vyslanými zo snímacieho zariadenia. Vo všeobecnosti sa takýmto spôsobom vytvárajú digitálne reprezentácie rôznych predmetov, častí ľudského tela ako napríklad tváre. Touto metódou je možné získať milióny takýchto bodov za sekundu. Súčasťou týchto laserových skenerov bývajú aj kamery, vďaka ktorým je na vytvorený model možné naniest informáciu o farbe objektu.



Obr. 2.9: Laserový skener na krátke vzdialenosti

Záujem o digitalizáciu väčších plôch ako miest a krajín vznikol až vývojom novších technológií 3D laserového skenovania LiDAR (skratka z angl. *Light Detection and Ranging*), ktoré má za cieľ využívanie laserového skenovania určené na väčšie vzdialenosti. Skenery typu LiDAR sa delia do dvoch kategórií, a to **pozemné**, ktoré majú efektívny rozsah skenovania od 10 metrov až do 200 metrov a **vzdušné** s rozsahom až do dvoch kilometrov. Pozemné skenery sa ukladajú na rôzne robotické jednotky, ale aj mobilné zariadenia ako napríklad motorové vozidlá [9], ktoré môžeme vidieť i na autách spoločnosti Google.



Obr. 2.10: Auto spoločnosti Google s LiDAR senzorm

2.1.3 Porovnanie

Podľa dostupnej literatúry je zrejmé, že porovnaním výsledkov údajov o snímanom povrchu získaných laserovým a optickým skenovaním vyplývajú závery, ktoré vyjadrujú, že presnosť merania pri vzdialenosti do 20 metrov je oboma skenovacími technológiami na rovnakej úrovni. Na presnosť merania priamo vplyva členitosť povrchu a algoritmus modelovania povrchu z nameraných mračien bodov. Pre vzdialenosti menšie ako 5 metrov je možné optickým skenovaním dosahovať vyššiu presnosť a to až submilimetrovú. Zväčšovaním vzdialenosti skenera od meraného objektu presnosť úmerne klesá. Naopak laserové skenery dokážu zachovať svoju presnosť merania na vzdialenosť od niekoľkých metrov až po 100 metrov s presnosťou merania 3 až 5 milimetrov.

Hlavnou výhodou optického skenovania je vytváranie záznamu s vysokým rozlíšením, dôležitým pri vytváraní textúr a správnu interpretáciu výsledkov. Najvhodnejšou metódou pre získanie kvalitnej digitálnej reprezentácie je použitie oboch technológií súčasne.[3]

2.2 Octree

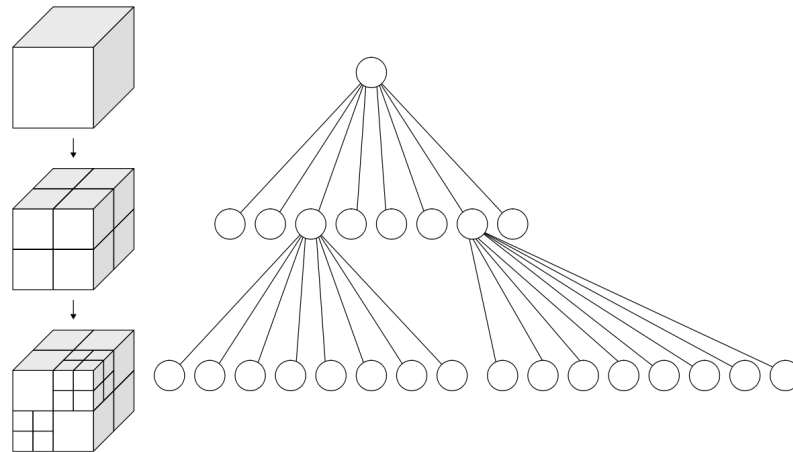
Existuje mnoho spôsobov ako reprezentovať 3D modely. V robotike sa na snímanie využívajú laserové snímky. Ich výstupom je mračno bodov, ktoré sa dá jednoducho zaznačiť v modele skladajúcom sa z voxelov.

Octree, teda *oktálový strom*, je strom, v ktorom každý prvok má rovnaký tvar. Vytvára hierarchickú stromovú štruktúru postupným delením priestoru tromi rovinami kolmými na súradnicové osi, čo v zobrazení predstavuje oddelené, stále sa zmenšujúce kocky. Pokiaľ takáto kocka dokáže úplne a presne popísať daný bod v priestore, nazýva sa koncovým uzlom alebo listom. V prípade, že uzol nedokáže však úplne popísať priestor vytvorí si 8 nových dcérskych uzlov, ktoré reprezentujú svoj unikátny priestor a sú súčasťou pôvodného uzlu. Každý uzol môže mať 0 až 8 listov. Každý uzol je zadaný svojím stredom a svojou dimenziou.

Vďaka takejto štruktúre sa dá presne popísať aj tá najmenšia časť modelu, kocka. Na spracovanie celého modelu teda vždy stačí jedna sada algoritmov. Z dôvodu používania stromovej štruktúry sa v pamäti nachádzajú všetky bunky zoradené, čo v prípade hľadania šetrí procesorový čas.

Z hierarchickej štruktúry stromu Octree vyplýva, že koreň stromu reprezentuje celý objekt na najvyššej úrovni. Toto platí pre všetky úrovne stromu a tomu sa vedia algoritmy prispôbiť a urýchliť tak svoj beh. Vedia, ktoré bloky dať je možné preskočiť, a ktoré zase nie.

Výrazne sa urýchľujú niektoré výpočty ako napríklad výpočet váhy nameraného objektu, buď aproximáciou cez objem najvyššieho koreňového uzlu, alebo prípadným rozdelením výpočtu, presunutím sa na nižšiu vrstvu a opakovaním.[10]



Obr. 2.11: Štruktúra oktálového stromu

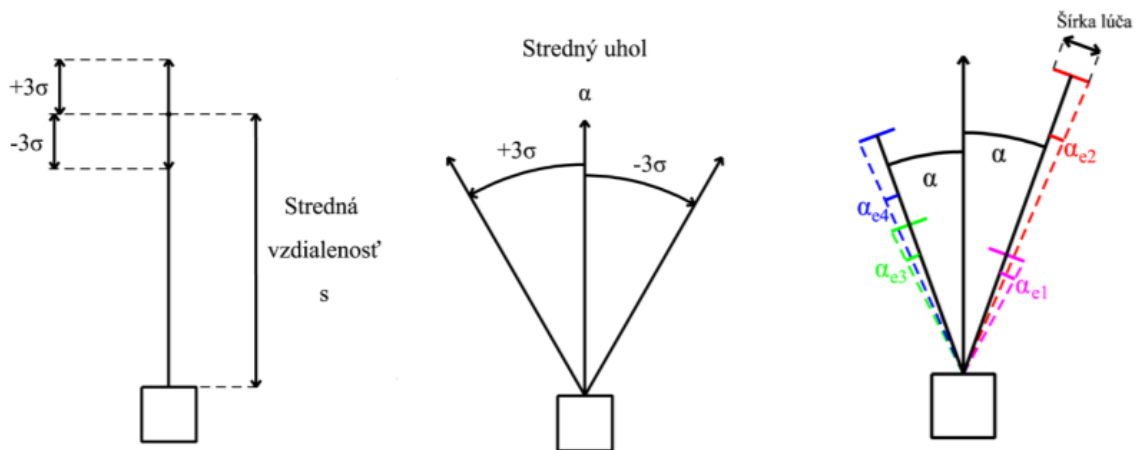
2.3 Detekcia chýb a pohyblivých objektov laserovým skenovaním

Každé meracie laserové zariadenie je zatažené chybami v meraní. Pre spracovanie údajov získaných z laserového skenera je nevyhnutné uvedenie si závažnosť a rozsah chyby. Pri určovaní odchýlok sa najčastejšie vychádza z údajov poskytnutých výrobcom zariadenia, čo však v konkrétnych podmienkach merania nie je možné dodržať.

2.3.1 Pravdepodobnostný model laserového diaľkomera

Pravdepodobnostný model laserového diaľkomera sa zameriava na 3 základné parametre laserového skenera:

- opakovateľnosť merania (stabilita získanej vzdialenosti pri opakovaní merania bez zmeny podmienok),
- opakovateľnosť merania vzhľadom na otočenie laserového lúča v priestore,
- vplyv šírky lúča na meranie (fiktívna veľkosť, ktorá ovplyvňuje meranie).



Obr. 2.12: Parametre laserového skenera

Určenie opakovateľnosti merania vzhľadom na meranú vzdialenosť

Z dostupnej literatúry bol ako najvhodnejší pravdepodobnostný model vybraný model normálneho rozdelenia opísaný gaussovou funkciou pravdepodobnosti.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

μ je stredná hodnota meranej hodnoty a σ je smerodajná odchýlka vypočítaná podľa vzťahu

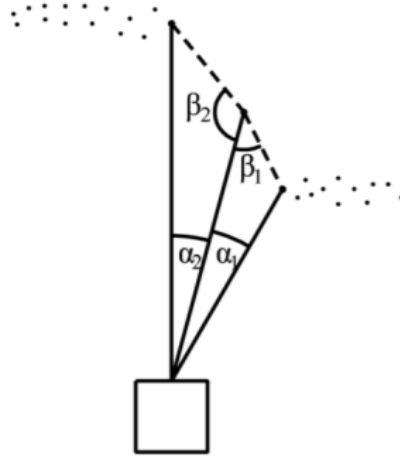
$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Počas realizácie meraní objektov v priestore vznikajú i takzvané falošné merania. Sú to merania, ktorých odmeraná vzdialenosť nekorešponduje so skutočnou vzdialenosťou k žiadanému objektu v priestore. Všeobecne sa táto vzdialenosť nachádza medzi vzdialenosťou objektu a pozadia. Opakovanými meraniami sa dosahuje väčšieho rozptylu, hodnoty vzdialenosti, ktorú nie je možné priradiť k žiadnemu objektu (falošné meranie). Falošné meranie sa vyskytuje na okrajoch reálnych objektov.

Falošné merania je možné efektívne odstrániť za použitia mriežky obsadenia a Bayesovho pravidla alebo aplikáciou rôznych filtrov.

Filter falošných meraní

Tieto merania sú pre navigáciu robotov a mapovanie prostredia nevyhovujúce nakoľko môžu spôsobiť nežiadúce zatvorenie voľných priechodov v mape.



Obr. 2.13: Detekcia falošného merania

Pre určenie falošného merania bol zvolený uhol, ktorý zvierá meranie so susednými bodmi. Tieto uhly je možné určiť podľa rovníc:

$$\beta_1 = \arccos \left(\frac{l_i - l_{i-1} \cos(\alpha_i)}{\sqrt{l_i^2 + l_{i-1}^2 - 2l_i l_{i-1} \cos(\alpha_i)}} \right)$$

$$\beta_2 = \arccos \left(\frac{l_i - l_{i+1} \cos(\alpha_{i+1})}{\sqrt{l_i^2 + l_{i+1}^2 - 2l_i l_{i+1} \cos(\alpha_{i+1})}} \right)$$

Pre uhol β_1 a β_2 je určený interval od 0 do 180 stupňov. Uhol α_i predstavuje hodnotu $l_i^\alpha - l_{i-1}^\alpha$. Funkcia $F = (l_i)$ má tvar

$$F(l_i) = (\beta_1 \leq \beta_{min} \wedge \beta_2 \geq \beta_{MAX}) \vee (\beta_1 \geq \beta_{MAX} \wedge \beta_2 \leq \beta_{min})$$

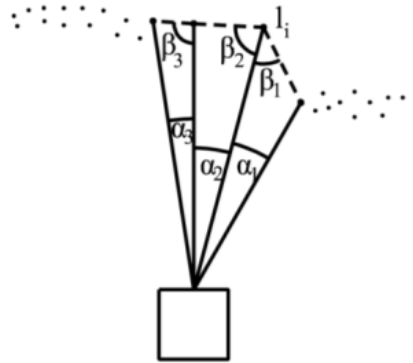
Pričom β_{min} a β_{MAX} sú spodná a horná hranica, pri ktorej sa považujú 2 po sebe idúce údaje za predpokladané falošné meranie. V prípade výskytu falošného merania sa vo výpočte podľa rovníc uvedených v texte táto skutočnosť zohľadní zmenou uhla α_i . Mobilný robot sníma počas svojho pohybu priestorom teleso z rôznych uhlov. Údaje sa zaznamenávajú do mapy. Ak bol za behu programu vďaka filtru odstránený objekt z merania bude zosnímaný v inom čase a z iného uhla vďaka čomu sa v mape objaví.

Filter šumu

Šum spôsobuje pri meraní laserovým skenerom nesprávne určené vzdialenosti od objektu v priestore. Pre mobilné zariadenie často nie je možné opakovať trajektóriu pohybu v priestore. Ako jednu z možností odstránenia šumu je možné použiť výpočet váženého plávajúceho priemeru, pri ktorom sa výsledná hodnota určí ako priemer hodnôt v okolí merania.

Za účelom zabránenia porušenia ostrých prechodov medzi dvoma objektami je nutné vážený plávajúci priemer upraviť. Identifikácia podmienky, na základe ktorej sa určí či došlo v meraní k ukončeniu súvislej časti sa určí podľa vzorca

$$F_d(l_i) = (\beta_1 \leq \beta_{min} \wedge \beta_2 \leq \beta_{MAX} \wedge \beta_3 \leq \beta_{MAX}) \vee (\beta_1 \geq \beta_{MAX} \wedge \beta_2 \geq \beta_{min} \wedge \beta_3 \geq \beta_{min})$$



Obr. 2.14: Vyhodnocovanie súvisiacich celkov

Segmentácia

Úlohou pri spracovaní merania z laserového skenera je meranie rozdeliť na časti súvisiace s konkrétnymi vlastnosťami prostredia. Pri úplnej segmentácii jednému objektu zodpovedá práve jeden segment, pri čiastočnej môže jednému objektu prináležať viac, alebo jednému segmentu viac objektov.

Najjednoduchším algoritmom segmentácie je prahovanie, ktoré sa často využíva pri spracovaní obrazu. Jeho úlohou je oddeliť popredie od pozadia. Pri laserovom skenovaní zohráva úlohu oddelovania jednotlivých objektov od seba v závislosti na vzdialenosti. Algoritmus spočíva v pridelovaní označenia jednotlivým bodom segmentu. V prípade prekročenia vzdialenosti bodu od zadáneho prahu je segment ukončený a vytvorený nový. Nevýhodou algoritmu prahovania je, že pri väčších objektoch dochádza k tvorbe viacerých menších segmentov.[4]

Detekcia pohyblivej prekážky

Východiskom detekcie je predspracovanie a segmentácia objektov. Algoritmus pre určenie pohyblivej prekážky pozostáva z nasledovných krokov:

- Určenie polohy mobilného robota
- Hľadanie zhodných objektov
- Vyhodnotenie dynamických objektov

Na určenie polohy mobilného robota sa využíva odometria, pri ktorej sa pomocou snímačov určí relatívny posun robota medzi jednotlivými krokmi. Zhodné objekty sa hľadajú medzi prebiehajúcimi meraniami a množinou objektov charakterizujúcich celé prostredie. Pre vyhodnotenie pohybu objektu sa používajú význačné body, ktoré charakterizujú daný objekt a jeho posun. [1]

Kapitola 3

Návrh a Implementácia

Laserové skenovanie umožňuje detailne a presne zachytávať priestorové objekty prostredníctvom mračien bodov. Mračno bodov predstavuje veľký súbor bodov, pričom každý bod je charakterizovaný svojou polohou v priestore prostredníctvom troch súradníc X, Y, Z. Tieto mračná bodov sú premietané do trojrozmernej mriežkovej siete, podľa ktorej sa konvertujú na voxely. Voxely majú predom stanovenú veľkosť. V políčkach mriežky, v ktorých sa nachádza aspoň jeden bod sa vytvorí voxel. Voxely sa následne ukladajú do štruktúry oktálového stromu. Zo stromu je potrebné odstrániť chyby vytvorené pri meraní a pohyblivé objekty ako ľudí a motorové vozidlá. Chyby sú odstránené prostredníctvom špeciálneho algoritmu. Pohyblivé objekty sú detekované segmentáciou každej snímky a následne špeciálne označené, užívateľ si môže zobrazit buď voxelovú reprezentáciu skutočného prostredia bez pohybujúcich sa objektov, s pohybujúcimi sa objektami alebo iba zviditeľniť všetky tieto pohyblivé objekty nastavením správneho príznaku. Takto zostrojený nástroj je vhodný pre použitie na mapovanie trojrozmerných priestorov, bezpečnú autonómnú navigáciu robota alebo iného zariadenia, ku ktorému je LiDAR senzor pripevnený.

3.1 Ovládanie

Aplikácia je ovládaná parametrami nastaviteľnými pri jej spustení. Medzi parametre patria prepínače módu zobrazenia dát, nastavenie vstupného toku dát. Výsledné uloženie vytvorenej trojrozmernej voxelovej reprezentácie reálnej krajiny bude možné do štandardného súboru zvoleného správnym prepínačom. Predchádzajúco vytvorené mapy je možné na novo otvoriť a pokračovať v ich tvorbe. Odporúča sa užívateľom použiť rovnakú začiatočnú pozíciu.

3.2 Implementácia

Cieľom mojej práce bolo vytvorenie aplikácie schopnej zostaviť trojrozmerný model prostredia pomocou laserového skenovania senzorom Velodyne LiDAR¹. Senzor HDL-32E LiDAR je malých rozmerov, prenosný obsahujúci 32 laserov pokrývajúcich zorné pole o veľkosti 40 stupňov zvislo. Ideálna vzdialenosť medzi senzorom a sledovaným objektom je 80 až 100 metrov. Za sekundu dokáže nasnímať až 700 tisíc bodov. Obsahuje veľké množstvo údajov s vysokou úrovňou rozpoznávania objektu v reálnom čase. Senzor je riešením výzvy preko-

¹<http://velodynelidar.com/>

návania najväčších problémov samostatnej navigácie, mapovaniu a iných aplikácií využitia v reálnom svete.



Obr. 3.1: Senzory Velodyne

Senzor HDL-32E môže byť ovládaný pomocou frameworku *Robot Operating System*. Výsledkom snímania údajov zo senzora sú mračná bodov, ktoré sú následne konvertované na voxely, kocky. Tieto údaje sa ukladajú do oktálového stromu, ktorý spravuje framework *octomap*. Výsledná oktomapa, voxelový model je v spätnej relácii prečítaný a vizualizovaný prostredníctvom ROSu cez nástroj Rviz.

Nástroj je tvorený v jazyku C++ za použitia knižníc Robot Operating System a Octomap, ktoré riešia hlavnú problematiku témy.

Ďalej sa v práci zaoberám popisom a vysvetlením ROSu a octomapy.

3.3 Robot Operating System

Robotika je veľmi široký obor. Veľká časť výskumníkov si pre ich potreby vytvorila mnoho *frameworkov* určených na prototypovanie softvérov pre ich experimenty. Toto malo za následok vytvorenie mnoho softvérových systémov, ale všetky z nich boli zamerané na špecifickú službu. Robot Operating System aj napriek svojmu menu nie je žiaden operačný systém. Je to sada *frameworkov*, ktoré boli pôvodne zamerané na správu robotov na *Stanford University*², ale výsledná architektúra bola použiteľná na omnoho viac systémoch.

Tvorba platformy ROS sa dá sumarizovať do niekoľkých bodov:

- peer-to-peer
- založená na nástrojoch
- viacjazyčná
- open-source

Systém zostrojený na platforme ROS sa skladá z mnohých procesov, ktoré sa nemusia nachádzať na tom istom stroji. V prípade, že sa nachádzajú na rozdielnych zariadeniach komunikujú medzi sebou za behu programu princípom *rovný s rovným* (z angl. *peer-to-peer*). Rozhranie nie je špecifikované, takže je možné komunikovať ako cez *ethernetový* kábel tak aj bezdrôtovo.

ROS umožňuje prácu vo viacerých jazykoch, pre ktoré boli vytvorené základné knižnice na komunikáciu podľa protokolu **XML-RPC**. Tento protokol je vytvorený spojením

²<http://stair.stanford.edu>

Remote Procedure Calls (RPC) a *XML*, na zakódovanie volaní funkcií a využíva HTTP na transport. Jeho cieľom je zabezpečovanie komunikácie medzi počítačmi a jednoduchého zdieľania dát[8]. Vďaka tomuto sa v systéme ROS dá komunikovať iba za použitia jednoduchých správ, ktoré sú nezávislé na programovacom jazyku. Medzi základné jazyky patrí C++, Python, Octave a Lisp.

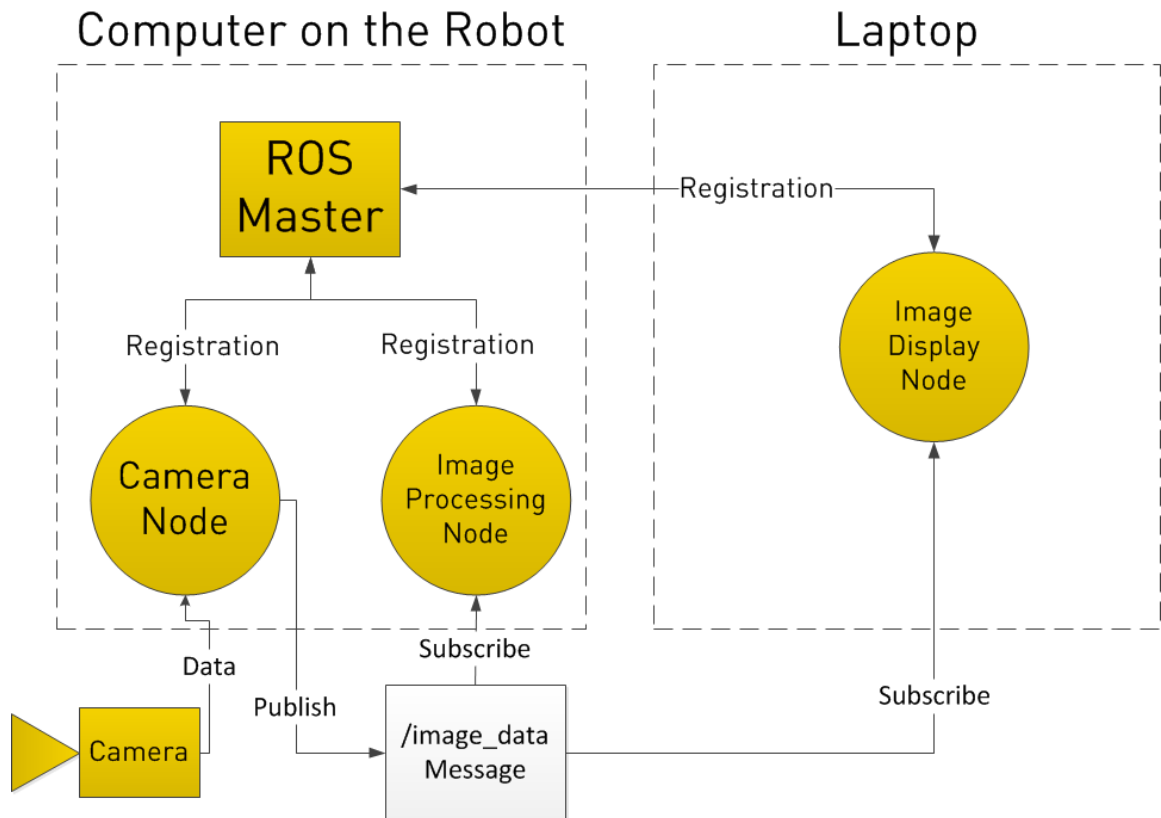
Radšej než, aby mal ROS jedno veľké jadro, vývojári sa rozhodli pre mnoho menších programov, ktoré budú vykonávať malý počet vecí. Tieto nástroje sú napríklad nástroje na meranie šírky potrebného internetového pásma, grafické zobrazenie správ medzi procesmi, automatická generácia dokumentácie atď.. Cieľom bolo takýmto spôsobom zvýšiť stabilitu a efektivitu prostredia.

3.3.1 Štruktúra ROSu

Základy implementácie v systéme ROS sú uzly, správy, témy a služby. **Uzly**(z angl. *Nodes*) sa nazývajú samotné procesy, ktoré vykonávajú všetku výpočetnú prácu. Keďže je ROS modulárny, dajú sa tieto uzly tiež nazývať ako "softvérové moduly".

Správy(z angl. *Messages*) sú spôsob komunikácie medzi jednotlivými uzlami. Správa musí byť jednoznačne definovaná silne typovanou štruktúrou. Tá pozostáva z jednoduchých dátových typov ako napríklad *integer*, *boolean* a tak ďalej, poliami alebo ďalšími správami. V pamäti sa pre rýchlu orientáciu v dátach uchováva viacerov správ rovnakého typu. Dajú sa rozlíšiť ich časovou známku, ktorú si nesú.

Uzly posielajú všetky správy na takzvané **témy**(z angl. *topic*), ktoré sú značené ich názvom uloženým v reťazci. Na tieto témy sa môže napojiť veľké množstvo uzlov buď ako odoberatelia alebo publikovatelia.



Obr. 3.2: Ukážka funkčnosti správ v systéme ROS

Komunikácia na úrovni tém však nie je vhodná pri súbežných tranzakciách a preto existuje ďalší typ uzlov, ktorý sa nazývajú **služby** (z angl. *services*). Služby definujú ich názvy v reťazcoch a ich striktné typované návratové správy. Názvy služieb musia byť unikátne.

3.3.2 Dôležité moduly

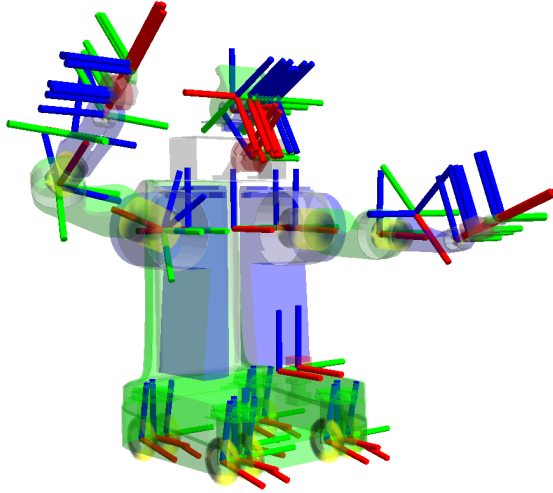
Všetky jednotlivé moduly sú uložené v **balíkoch** (z angl. *packages*). Každý balík musí obsahovať vlastný XML súbor, v ktorom si nesie údaje o autorovi. Taktiež tu sú zaznamenané všetky balíky, na ktorých daný modul závisí.

Keďže v robotike nie je možné mať vždy pri práci prístup k robotom, ROS poskytuje ako riešenie tohoto problému utilitu **roscat**, ktorá slúži na zaznamenávanie a následovný prehrávanie vybranej komunikácie vzniknutej za behu systému. Takýmto spôsobom sa dá plne nahradiť nutnosť prístupu k zariadeniu, na ktoré sa práve tvorí softvér. Týmto spôsobom sa dajú zachytiť všetky správy a ich obsah.

Na komunikáciu programu z užívateľom je možné využiť jednoduché a krátke programy, ktoré sa napoja na určitý prúd správ, ktoré sú vyslané programom, s ktorým chce užívateľ komunikovať. ROS však obsahuje modul **rviz**, pomocou ktorého vie užívateľ jednoducho nastaviť, ktoré správy chce odoberať použitím intuitívneho grafického užívateľského rozhrania. ROS takýmto spôsobom dokáže zobrazíť mnoho typov dát, medzi ktoré patrí napríklad *mračno bodov*, výstupy z kamier alebo zmenu pozície pri pohybe robota.

Jedným z najužitočnejších a najdôležitejších typov správ sú **transformácie** (*tf*). Sú to správy, ktoré plnia dôležitú funkciu informovania o zmene polohy prvku zariadenia v relácií ku ostatným zariadeniam na robotovi. Transformácie sú dôležité na správnu vizualizáciu.

Rovnako sa využívajú napríklad aj na správne natočenie a posunutie prijatých mračien bodov. Vytvárajú sa stromy transformácií, väčšinou existuje jedna správa pre celého robota, ktorá sa prirovnáva k začiatočnej polohe pri spustení programu a správy pre každý kĺb a koncové zariadenie robota.



Obr. 3.3: Strom transformácií na robotovi PR2

[11]

3.4 Octomap

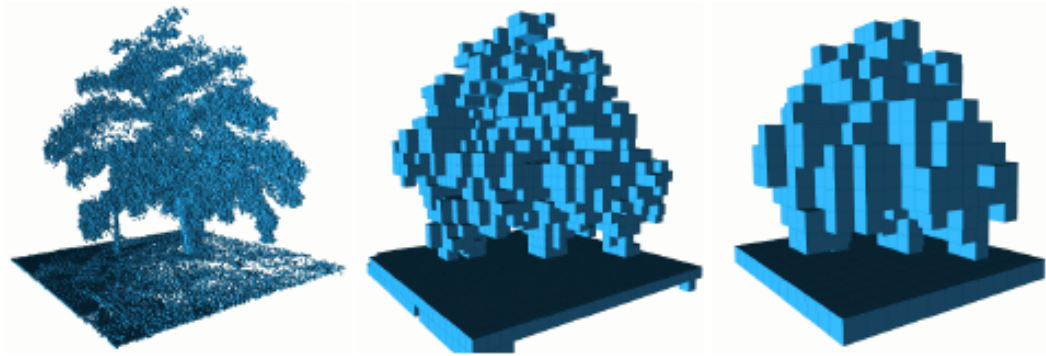
Rozvoj robotiky v poslednej dobe si vyžiadal uvedenie 3D modelov do používania na zaznamenanie priestoru. Framework Octomap slúži ako open-source varianta efektívneho ukladania trojrozmernej geometrickej reprezentácie nameraného prostredia.

Je to framework založený na využívaní *oktálových stromov* (octree) vytvorený v jazyku C++. Jeho cieľom je vytvorenie voxelového modelu z dát ako sú mračná bodov, z lúčov, atď..

Pre jednoduchú a bezpečnú navigáciu v 3D priestore si framework *Octomap* ukladá do pamäte nielen body, ktoré boli zaznamenané napríklad laserovým skenom, ale aj body v ktorých nebolo nič nájdené. Vkladať informácie je možné kedykoľvek, preto modelovanie a aktualizovanie prebieha za pomoci *pravdepodobnostnej funkcie*, ktorá berie v úvahu šum senzora a jeho merania, zapríčiňujúce dynamické zmeny v prostredí. Taktiež umožňuje viacerým robotom prispievať do tej istej mapy a to súčasne, ale aj postupne. Každá takto vytvorená mapa si zachováva svoje dáta vo viacerých rozlíšeniach. Toto umožňuje rýchle prehliadanie častí dát, kde nie je potrebné veľké množstvo detailov. Oktomapy nie sú obmedzené zásobníkmi, pamäť im je priradzovaná dynamicky. Mapy sú dostatočne kompresované na to, aby nezaberali na disku ani v pamäti veľa miesta, a aby bolo zaručené praktické zdieľanie informácií aj v prípadoch s obmedzenou schopnosťou prenosu dát. [12]

Rozlíšenie

Rozlíšenie oktomapy jednoznačne udáva veľkosť najmenšieho voxelu. Tento oktál je špecifický tým, že už nemá žiadne dcérske bunky.



Obr. 3.4: Postupné znižovanie rozlíšenia oktálového stromu

3.4.1 Pravdepodobnosť obsadenia

Na zisťovanie, ktoré body na mape sú obsadené sa využíva *occupancy grid*, mriežka zaznačujúca obsadenosť.[2] Na základe nasledujúceho vzorca sa vypočítava obsadenosť jednotlivého voxelu v modele.

$$P(n|z_{1:t}) \left[1 + \frac{1 - P(n|z_t)}{P(n|z_t)} \frac{1 - P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1 - P(n)} \right]^{-1}$$

Vzorec na výpočet aktuálnej pravdepodobnosti závisí na predchádzajúcom výpočte. Hodnota $P(n)$ vyznačuje predchádzajúcu vypočítanú pravdepodobnosť, $P(n|z_{1:t-1})$ predchádzajúci predpoklad. $P(n|z_t)$ sa vzťahuje na pravdepodobnosť obsadenosti voxelu n v danom meraní z_t . Táto hodnota je špecifická zariadeniu, ktoré ju nameralo. Priebežne sa vypočítaná pravdepodobnosť porovnáva s hodnotou prahu, podľa ktorej októmata určí, či sa daný bod na mape nachádza alebo je súčasťou vypočítanej chyby alebo pohybujúceho sa objektu. [7]

Framework Octomap je priamo podporovaný na systémoch ROS, jeho aktuálny balík sa nachádza v repozitároch ROS-u³ spolu s rôznymi podpornými knižnicami ako napríklad knižnica *octomap_rviz_plugins*, ktorá umožňuje vizualizačnému programu ROS-u, *rviz* zobrazovať dáta typu *mapa(2D)* a *mriežka(3D)* obsadenosti.[6]

3.5 Implementačné riešenie

Vzhľadom na to, že všetky potrebné nástroje už boli doposiaľ opísané, nasledujúca časť sa venuje ich skutočnému použitiu v programe.

Aplikácia je zhotovená v programovacom jazyku C++ na operačnom systéme Ubuntu16.04. Spúšťa sa s jedným alebo viacerými vstupnými parametrami. Medzi vstupné parametre patria:

- h vypíše nápovedu,
- d nastavenie názvu správ vstupných dát. Predvolené nastavenie je „/velodyne_points“,
- e nastavenie názvu správ transformácií mračna bodov. Predvolené nastavenie je „/base_link/velodyne“,

³wiki.ros.org/octomap

- f** nastavenie názvu správ transformácií sveta. Predvolené nastavenie je „/world“. Používať spolu s *-e*,
- v** nastavenie názvu výstupných správ oktálnych stromov. Predvolené nastavenie je „/map“,
- w** nastavenie názvu výstupných správ transformácií 3D mapy. Predvolené nastavenie je „/map“,
- o** nastavenie výstupného súboru. Predvolené nastavenie je „/octomap“,
- t** nastavenie dátového typu vstupného súboru. Vkladá sa za názov výstupného súboru. Možnosť **ot**, čo je bezstratový výstup alebo **bt**, čo je dátový výstup uchovávajúci iba binárne dáta stromu. Predvolené nastavenie je „ot“,
- m** nastavenie módu vizualizačného výstupu. Prijateľné hodnoty sú 0 (žiadna filtrácia), 1(iba statické objekty) a 2(iba dynamické objekty). Predvolené nastavenie je „0“,
- p** nastavenie intervalu prahovania. Uvádza sa v celočíselných hodnotách 0 až 100 Predvolené nastavenie je „70“.
- r** nastavenie rozlíšenia oktomapy v centimetroch. Predvolené nastavenie je „50“,

Všetky funkcie jadra ROSu sa nachádzajú v mennom priestore s rovnakým menom, „ros“. Na začiatku behu programu je potrebné z tohto menného priestoru zavolať metódu inicializácie `init`, vďaka ktorej ju bude systém schopný rozpoznať nakoľko sa tu zádava unikátny názov programu v ROSe.

Bez komunikácie s ostatnými ROS modulmi by program nemohol pracovať. Z toho dôvodu je potrebné aby sa prihlásil na odber správnych správ a odoslal vybraný výstup. Prichádzajúce dáta sú typu `PointCloud2`, mračno bodov a odoslané sú dáta vo formáte `octomap_msg`, čo je vlastne rozšírenie dátového typu mriežky obsadenia (*occupancy grid*). Na publikovanie a odoberanie správ slúžia metódy `publish` a `subscribe` nachádzajúce sa v triede správ uzlov `NodeHandle`.

Aplikácia pracuje na asynchrónnom princípe, čo znamená, že pokiaľ sa aplikácií podarí vyhotoviť mapu a už nemá na vstupe žiadne ďalšie mračná bodov, záznamy z laserového skeneru, uspí sa a bude čakať až pokým jej jadro systému ROS neobnoví činnosť upozorením na zmenu dát v toku, téme, na ktorej sa nachádzajú nové mračná bodov.

V cykle spracovania samotných vstupných dát, vo funkcií zvanej `pointcloud2_callback`, sa po získaní dát zisťujú *transformácie* patriace k novo získaným dátam pomocou časovej známky nachádzajúcej sa v správe pomocou `TransformListener::lookupTransform` metódy. ROS si uchováva neuchováva iba posledné, aktuálne správy ale udržiava si záznam o tisíckach správ.

Transformácie mračen bodov prebiehajú vzlodom ku počiatočnému bodu, najskôr posunom o správnu vzdialenosť a následne sa rotujú podľa kvaterniónu, informácií dostupnej z transformačnej správy. Následne sú vložené do oktálového stromu. Toto všetko riadi framework `octomap` metódou `insertPointCloud`.

Vďaka využitiu nástroja `octomap` je možné ušetriť času riešením získavania finálneho modelu neobsahujúce mobilné objekty. `Octomap` dokáže vyhodnotiť pravdepodobnosť obsadenia daného voxelu v modele vyvolaním metódy `UpdateInnerOccupancy`, ktorá využitím integrovaného modelu pravdepodobnosti obsadenia nastaví do každého voxelu zapíšu jeho pravdepodobnosť, že sa v danom mieste niečo mobilného nachádza. Toto je zaznačené nízkou pravdepodobnosťou, podľa ktorej s pomocou nastavenia prahu dokážeme určiť statické a dynamické objekty.

Do systému ROS sa ako výstup z vytvorenej aplikácie vracia správa obsahujúca serializovaný oktálový strom, ktorý je sprevádzaný správou *transformácie* vysielanej metódou triedy `transformBroadcaster` zvanou `sendTransform`.

Zápis na disk program vykonáva v momente, kedy dostane príkaz na ukončenie. Príkaz na ukončenie sa zadáva pomocou kombinácie tlačítiek `Ctrl+C`, čo vyvolá signál prerušenia, *SIGINT*.

Kapitola 4

Zhodnotenie

Na svete sa v dnešnej dobe nenachádza veľa aplikácií schopných konvertovať mračná bodov na trojrozmerné modely. Väčšina z nich vytvára práve polygonálne modely. Tento daný druh modelov však v robotike za účelom autonómnej navigácie vhodný nakoľko má vysoké nároky na správu a pamäť.

4.1 Nástroj ReCap

Medzi jeden z nástrojov, ktoré sú schopné spracovávať mračná bodov patrí napríklad aj **ReCap** vyvíjaný spoločnosťou *Autodesk*. Ako väčšina podobných programov, aj tento je však určený pre strojárstvo, na vizualizáciu rôznych súčiastok. Rovnako ako ostatné vytvára práve polygonálne modely.

ReCap je schopný pracovať s mračnami bodov zakódovanými iba vo formátoch *.rcs*. Výhodou mnou navrhovaného programu je, že nie je závislý od presných formátov vstupných súborov, ale podpory systému ROS, v ktorej je možné za potreby vytvoriť jednoduchý modul čítania daného formátu, pokiaľ už nie je vytvorený.

Polygonálne modely sú vhodné na presnú vizualizáciu skutočných predmetov. Ich využitie na snímanie prostredia nie je práve najvhodnejšie. Zaberajú príliš veľa pamäte. Ich porovnanie s voxelovými modelmi sa dá nájsť v nasledujúcej tabuľke.

Typ dát	Veľkosť modelu 1[MB]	Veľkosť modelu 2[MB]	Veľkosť modelu 3[MB]
Mračno bodov v ASCII kódovaní	82	267	1610
Polygonový model	39	126	779
OcTree	1, 1	6, 8	13

Tabuľka 4.1: Porovnanie veľkostí 3D modelov

Z tabuľky vypláva, že prevedenie modelu na voxelový model uložený v oktálových stroch má iba zlomkové nároky na pamäť oproti porovnateľným polygonálnym modelom. Polygonálne modely majú presnejšiu reprezentáciu skutočného sveta avšak voxelové modely, ktorých mriežka vytvára diskretný priestor je užitočnejšia pre autonómnu navigáciu robotických zariadení, potrebuje menej procesorového času.

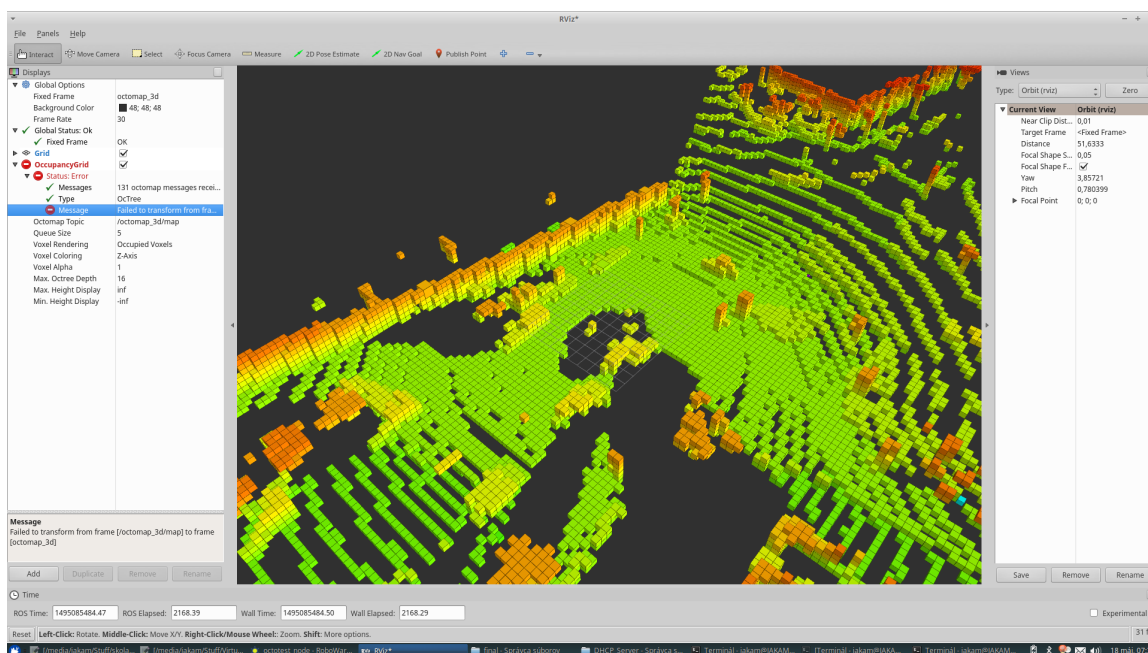
Názov objektu	Rozsah objektu[m ³]	Rozlíšenie[cm]	Veľkosť oktomapu[MB]
freiburg1_360	7, 9x7, 3x4, 6	2	21, 59
		5	2, 11
Kampus Freiburg	292x167x28	10	379, 70
		20	49, 68
		80	1, 53
Chodba FR-079	43, 7x18, 2x3, 3	5	15, 76
		10	2, 70

Tabuľka 4.2: Veľkosti máp vytvorených na univerzite Freiburg[7]

4.2 Testovanie

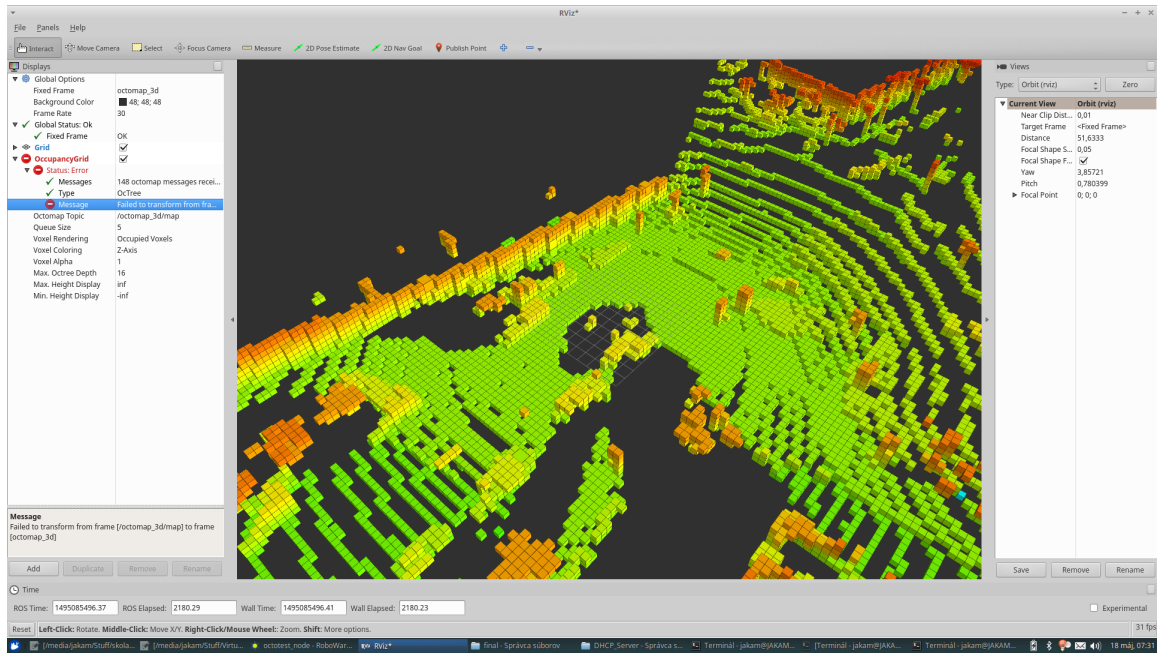
Pri vypracovaní práce som využíval dáta získané z robota Toad¹ a datasetov pochádzajúcich z porovnávacích nástrojov Kitti[5].

Testovanie som vykonával na viacerých počítačoch, najvýkonnejší z nich mal procesor s taktovacou frekvenciou 2,5GHz. Vyššie zmienené dáta mali frekvenciu snímok 10Hz, čo bolo pre môj počítač príliš náročné na spracovanie, často dochádzalo k zahlteniu systému. Pri znížení frekvencie je však možné dosiahnuť uspokojivé výsledky aj na slabších zariadeniach. Nároky sú vďaka inteligentnému systému ROS a octomap nízke, zariadenie s veľkosťou pamäte 4GB nemalo problém program používať.



Obr. 4.1: Ukážka tvorby voxelového modelu časť 1.

¹<http://www.fit.vutbr.cz/research/groups/robo/eqp.php.cs>



Obr. 4.2: Ukážka tvorby voxelového modelu časť 2.

Kapitola 5

Záver

V poslednom období zaznamenáva 3D laserové skenovanie veľký rozmach vďaka úspore času, realizácií merania a spracovania údajov a vytvorenia výsledného 3D modelu objektov a krajín. Využitie mobilného skeneru umožňuje zameranie a zosnímanie údajov o budovách, objektoch, celkoch objektov, čo umožní naskenovať celú ulicu, obec, mesto, krajinu. Zo získaných údajov je možné vytvoriť komplexné 3D mapy pozostávajúce z voxelov vytvorených z nameraných mračien bodov. Je to jedna z najefektívnejších metód získavania priestorových údajov.

V druhej kapitole práce som popísal spôsoby skenovania prostredia pomocou laserového a optického skenovaním a ich vzájomné porovnanie. Vzároveň som sa v tejto kapitole venoval efektívnemu spôsobu uloženia mračien bodov do pamäte a detekcií chýb vzniknutých pri zaznamenávaní a pohyblivých objektov.

V tretej kapitole som sa zaoberal návrhom riešenia práce, popísal vhodné implementačné nástroje použité pri vývoji a behu programu. Taktiež sa tu nachádza priblíženie skutočného riešenia vytvárania programu.

V štvrtej časti som sa venoval zhodnoteniu efektivity programu a jeho prirovnaniu a porovnaniu k doposiaľ vytvorených podobných riešení.

Výsledkom celej práce je funkčný nástroj na tvorbu 3D modelu prostredia.

Literatúra

- [1] Dekan, M.: *Autonómna navigácia mobilného robota*. Dizertační práce, Slovenská Technická Univerzita v Bratislave, 2014.
- [2] Elfes, A.: Using occupancy grids for mobile robot perception and navigation. *Computer*, ročník 22, č. 6, 1989: s. 46–57.
- [3] Fraštia, M.: Laserové verzum optické skenovanie skalných masívov. *Mineralia Slovaca*, ročník 44, č. 2, 2012: s. 177–184.
- [4] Fritz, T.: *Aplikace s 3D laserovým sd álkoměrem SICK*. Diplomová práce, Vysoké Učení Technické v Brne, 2009.
- [5] Geiger, A.; Lenz, P.; Stiller, C.; aj.: Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [6] Hornung, A.: Octomap - ROS wiki. 2014.
URL wiki.ros.org/octomap
- [7] Hornung, A.; Wurm, K. M.; Bennewitz, M.; aj.: OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, ročník 34, č. 3, 2013: s. 189–206.
- [8] Laurent, S. S.; Johnston, J.; Dumbill, E.; aj.: *Programming web services with XML-RPC*. Ö'Reilly Media, Inc.", 2001.
- [9] Luettel, T.; Himmelsbach, M.; Wuensche, H.-J.: Autonomous ground vehicles—Concepts and a path to the future. *Proceedings of the IEEE*, ročník 100, č. Special Centennial Issue, 2012: s. 1831–1839.
- [10] Meagher, D.: Geometric modeling using octree encoding. *Computer graphics and image processing*, ročník 19, č. 2, 1982: s. 129–147.
- [11] Quigley, M.; Conley, K.; Gerkey, B.; aj.: ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, ročník 3, Kobe, 2009, str. 5.
- [12] Wurm, K. M.; Hornung, A.; Bennewitz, M.; aj.: OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, ročník 2, 2010.
- [13] Žára, J.; Beneš, B.; Sochor, J.; aj.: *Moderní počítačová grafika*. Computer press, 1998, ISBN 80-251-0454-0.

- [14] Čermák, J.: Metody 3D skenování objektů. 2015.
- [15] ČVUT: Laser - Teorie 3D. [Online; navštíveno 14.05.2017].
URL lfgm.fsv.cvut.cz/data/vvt/s1/laserteorie3d.pdf

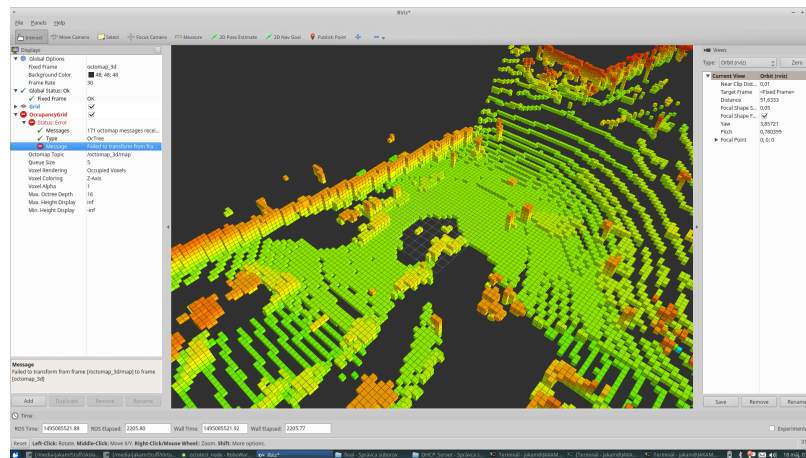
Príloha A

Obsah príloženého pamäťového média

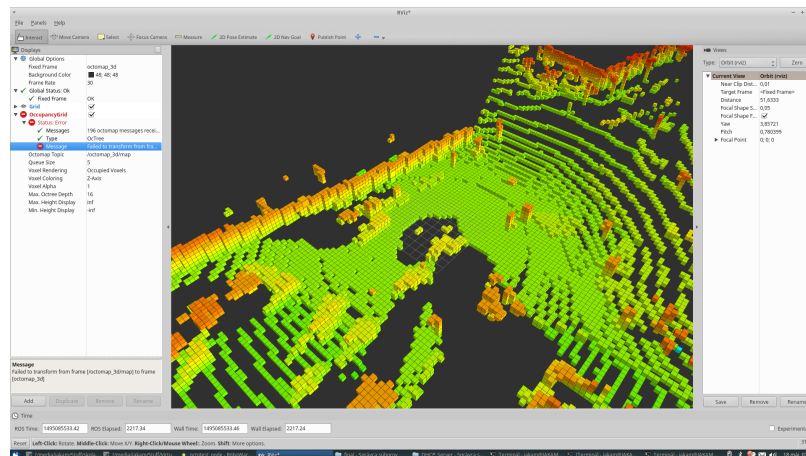
- Zdrojové kódy technickej správy bakalárskej práce
- Zdrojové kódy programu bakalárskej práce
- Súbor readme
- Plagát

Príloha B

Dodatočné snímky programu



Obr. B.1: Ukážka tvorby voxelového modelu časť 3.

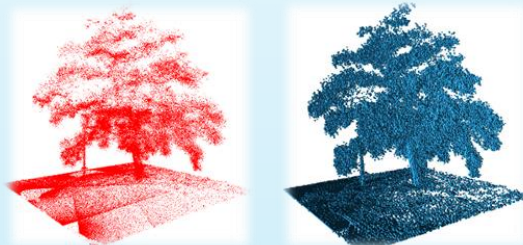


Obr. B.2: Ukážka tvorby voxelového modelu časť 4.

Príloha C

Plakát

Pořizování 3D modelu ulice pomocí laserového skenování



Konverzia mračien bodov senzorov
LiDAR na voxelové 3D modely.
Možné používať aj v obytných
častiach sveta. Dokáže
ignorovať pohyblivé
objekty zo snímok.

Jednoduché na používanie
Riešenie nenáročné na pamäť
Tvorba 3D modelov skutočných scenérií



Rok: 2017

Ústav: UPGM

Vedúci práce: Ing. Martin Veľas

Autor práce: Michal Jakabovič