

# Usporedba metoda za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima

---

Strinić, Luka

Master's thesis / Diplomski rad

2022

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:776454>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-22**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni diplomski studij računarstva**

**USPOREDBA METODA ZA IZGRADNJU KARTE I  
NAVIGACIJU MOBILNOG ROBOTA U  
UNUTRAŠNjim PROSTORIMA**

**Diplomski rad**

**Luka Strinić**

**Osijek, 2022.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMATIČKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 16.05.2022.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za diplomski ispit**

Ime i prezime Pristupnika:	Luka Strinić
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1084R, 06.10.2019.
OIB studenta:	39168964140
Mentor:	Prof.dr.sc. Robert Cupec
Sumentor:	Davor Damjanović, mag. ing. comp.
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. Josip Balen
Član Povjerenstva 1:	Prof.dr.sc. Robert Cupec
Član Povjerenstva 2:	Izv. prof. dr. sc. . Damir Filko
Naslov diplomskog rada:	Usporedba metoda za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima
Znanstvena grana diplomskog rada:	Umjetna inteligencija (zn. polje računarstvo)
Zadatak diplomskog rada:	Usporediti različite metode za izgradnju karte i navigaciju autonomnih robota u unutrašnjim prostorima. Sukladno algoritmima koji se koriste treba upotrijebiti odgovarajuće senzore koje razmatrane metode podržavaju. Senzori koji će se razmatrati su LiDAR i dubinska kamera te njihove kombinacije. Razmatrane metode treba usporediti s obzirom na točnost, preciznost i cjelovitost izgrađene karte te na mogućnost samostalne izrade karte. Sumentor s FERIT-a: Davor Damjanović
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	16.05.2022.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 16.06.2022.

Ime i prezime studenta:

Luka Strinić

Studij:

Diplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

D-1084R, 06.10.2019.

Turnitin podudaranje [%]:

5

Ovom izjavom izjavljujem da je rad pod nazivom: **Usporedba metoda za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima**

izrađen pod vodstvom mentora Prof.dr.sc. Robert Cupec

i sumentora Davor Damjanović, mag. ing. comp.

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.  
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

<b>1. UVOD</b> .....	<b>1</b>
1.1. Cilj diplomskog rada .....	2
<b>2. MOBILNI ROBOT</b> .....	<b>3</b>
2.1. TurtleBot 2.....	3
2.2. Lidar.....	4
2.3. Orbbec Astra kamera .....	5
<b>3. PROGRAMSKA PODRŠKA</b> .....	<b>7</b>
3.1. ROS .....	7
3.2. RVIZ.....	9
3.3. Gazebo.....	10
<b>4. SLAM ALGORITMI</b> .....	<b>11</b>
4.1. Simultana lokalizacija i mapiranje.....	11
4.2. Dynamic Window Approach.....	12
4.3. GMapping .....	14
4.3.1. GMapping lidar.....	16
4.3.2. GMapping dubinska kamera .....	17
4.4. RTAB-Map .....	19
4.4.1. RTAB-Map lidar.....	21
4.4.2. RTAB-Map dubinska kamera .....	22
4.4.3. RTAB-Map lidar i dubinska kamera .....	24
4.5. Cartographer .....	26
4.5.1. Cartographer lidar .....	31
4.5.2. Cartographer dubinska kamera .....	33
<b>5. USPOREDBA METODA</b> .....	<b>35</b>
5.1. Mapiranje prostora lidar senzorom .....	35
5.2. Mapiranje prostora dubinskom kamerom .....	36
5.3. Mapiranje prostora kombinacijom lidara i dubinske kamere.....	38
<b>6. ZAKLJUČAK</b> .....	<b>40</b>

<b>LITERATURA .....</b>	<b>43</b>
<b>SAŽETAK.....</b>	<b>44</b>
<b>ABSTRACT .....</b>	<b>45</b>
<b>ŽIVOTOPIS.....</b>	<b>46</b>
<b>PRILOG .....</b>	<b>47</b>
P.4.3.1.1 Parametri GMapping algoritma uz korištenje lidara.....	47
P.4.3.2.1 Parametri GMapping algoritma uz korištenje dubinske kamere.....	48
P.4.4.1.1 Parametri RTAB-Map algoritma uz korištenje lidara.....	49
P.4.4.2.1 Parametri RTAB-Map algoritma uz korištenje dubinske kamere.....	51
P.4.4.3.1 Parametri RTAB-Map algoritma uz korištenje lidara i dubinske kamere.....	52
P.4.5.1.1 Parametri Cartographer algoritma uz korištenje lidara .....	53
P.4.5.2.1 Parametri Cartographer algoritma uz korištenje dubinske kamere.....	55

## 1. UVOD

Sam termin robot prvi put upotrijebio je 1921. godine češki dramatičar Karel Čapek, prema češkoj riječi robotnik što u prijevodu znači radnik, rob i sl. Ideja o robotima datira još od Leonarda da Vinci koji je predvidio moderne robote napravivši pokretni stroj u obliku lava. Na američkim sveučilištima 1960-ih godina pojavljuju se prvi eksperimentalni primjerci robota. Razvojem tehnologije, tehničkih znanosti te industrije dolazi do sve većeg spektra upotrebe robota u današnjem društvu. Moderna industrijska postrojenja i skladišta nemoguće je zamisliti bez industrijskih robota te autonomnih mobilnih robota. Danas postoji mnogo primjera korištenja autonomnih mobilnih robota u skladištima i distribucijskim centrima. Jedan od primjera je Amazon koji koristi više od 80 tisuća robota u svojim distribucijskim centrima koji obavljaju različite zadatke. Kao što su premještanje robe unutar svojih skladišta. Navedeni primjer je posebno aktualan problem na kojem znanstvenici desetljećima rade kako bi mobilni roboti postali autonomni. To znači da je robot u mogućnosti izvršavati svoje zadatke i zadaće bez ljudskih smjernica i nadzora. Učenje mobilnog robota kako da se kreće kroz svoje okruženje jedan je od najvećih problema današnjice. Navigacija robota određuje planiranje kretanja robota u svrhu izvršenja njegovog zadatka. Navigacija u netrivialnim okruženjima, gdje postoje opasna područja i različite prepreke zahtjeva unaprijed planiranje puta za što je presudna karta prostora. Mobilni robot koji se nalazi u unutrašnjim prostorima npr. kući mora znati što su vrata i gdje se vrata nalaze kako bi se uspješno kretao kroz prostorije. Karta robotu pruža informacije o relativnom položaju prepreka i objekata potrebnih za orijentaciju u okolišu, naime karta mora sadržavati i opisati sve predmete koji su nužni za robota da se uspješno kreće kroz prostor. Prepreke označavaju mjesta gdje robot ne može ići dok različiti objekti služe za utvrđivanje lokacije i orijentacije. Robot mora precizno odrediti gdje se trenutno nalazi na svojoj karti te odlučiti kojim putem krenuti dalje kako bi uspješno izvršio zadani zadatak. Taj problem se naziva lokalizacija. Prilikom putovanja kroz prostor robot koristi svoju kartu i nadopunjuje ju novim informacijama. Mapiranje i lokalizacija su problemi koji se rješavaju zajedno, a kombinirani problem se naziva simultana lokalizacija i mapiranje SLAM (engl. *Simultaneous Localization And Mapping*). Zbog popularnosti problema navigacije mobilnog robota u unutrašnjim prostorima, izabrana je i tema ovog diplomskog rada, Usporedba metoda za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima.

Sam diplomski rad podijeljen je na 6 poglavlja. Prvo poglavlje je uvodno poglavlje u kojem je opisan zadatak diplomskog rada. U drugom poglavlju navedene su komponente mobilnog robota koje se koriste za rješavanje problema diplomskog rada. Treće poglavlje posvećeno je programskoj

podrški pomoću koje se rješava navedeni problem diplomskog rada. U četvrtom poglavlju prikazane su i detaljno objašnjene metode koje se koriste za rješavanje problema ovog diplomskog rada. U petom poglavlju prikazani su i uspoređeni rezultati dobiveni različitim metodama. Šesto poglavlje posvećeno je zaključku diplomskog rada u kojem su navedene prednosti i nedostaci pojedinih metoda za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima.

## **1.1. Cilj diplomskog rada**

Cilj diplomskog rada je izraditi, implementirati te usporediti različite algoritme za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima korištenjem programskih alata raspoloživih u ROS-u (engl. *Robot Operating System*). Sukladno algoritmima koji se koriste treba upotrijebiti različite senzore koje razmatrane metode podržavaju. Senzori koji će se razmatrati su LIDAR i dubinska kamera te njihove kombinacije. Razmatrane metode treba usporediti s obzirom na točnost, preciznost i cjelovitost izgrađene karte te na mogućnost samostalne izrade karte.



## 2. MOBILNI ROBOT

Postoji puno definicija mobilnog robota, jedna od definicija glasi : „Mobilni robot je mobilan i manipulativni fizički sustav koji se autonomno giba kroz nestrukturirani prostor, ostvarujući pritom interakciju s ljudskim bićima ili autonomno obavljajući neki posao umjesto njih“. Područje tehnike koje se bavi mobilnim robotima je mobilna robotika. Mobilna robotika integrira veliki broj znanstvenih disciplina: strojarstvo, elektroenergetika, elektronika, automatika, računarstvo, matematika, umjetna inteligencija, percepcija, neurologija [1]. Za rješenje zadanog zadatka diplomskog rada korišten je TurtleBot 2 mobilni robot.

### 2.1. TurtleBot 2

U studenom 2010 godine Willow Garage predstavlja mobilni robot pod nazivom TurtleBot 2. TurtleBot 2 jedan je od najpopularnijih jeftinih osobnih robota s programskom podrškom otvorenog koda za istraživanje i obrazovanje. Opremljen je sa Kobuki bazom robota, lidarom te Orbec Astra RGB-D kamerom. Najvažniji dio TurtleBot 2 robota je kobuki mobilna baza koja je proizvod tvrtke Yujin Robot. Kobuki baza koristi diferencijalni pogon kotača, što omogućava neovisno upravljanje s oba kotača. Ako kotači imaju istu brzinu rotacije TurtleBot 2 se kreće unaprijed ili unazad po ravnoj liniji u ovisnosti o smjeru rotacije kotača. Isto tako ako kotači imaju različitu brzinu rotacije TurtleBot 2 može skretati u različitim smjerovima. Baza sadržava još i bateriju te različite priključke za napajanje dodatne opreme, priključak za napajanje prijenosnog računala i USB priključak. TurtleBot 2 dolazi s odvojenom stanicom za punjenje kobuki baze [2].



Slika 1. Mobilni robot TurtleBot 2

Na slici 1 prikazan je TurtleBot 2 mobilni robot sa označenim glavnim dijelovima.

## 2.2. Lidar

LIDAR ( engl. „*Light Detection and Ranging*“) je uređaj koji pomoću laserskih zraka i senzora izračunava udaljenost do površine koju skenira. Radi tako da mjeri odbijene laserske zrake te računa udaljenost do površine koja je skenirana. Skenirane površine mogu biti različite prepreke, sobni zidovi, zemljina površina, ocean itd, što ovisi o primjeni za koju se koristi. Kako bi dobili točne rezultate prikupljeni podaci o udaljenosti skenirane površine moraju se kombinirati s različitim podacima kao što su kutovi skeniranja, položaj, podaci o kalibraciji, GPS i orijentacija. Kombinacijom navedenih prikupljenih podataka LIDAR generira i rekonstruira prikaz skeniranog područja te ga prikazuje u 2D ili 3D prostoru kao oblak točaka. Osnovna zadaća LIDAR-a je mapiranje okruženja koje skenira , a sastoji se od skenera, lasera i navigacijskog sustava. U današnje vrijeme primjena LIDAR-a je vrlo popularna u autonomnim automobilima i mobilnim robotima.



Slika 2. RPLIDAR A2

Za rješenje problema diplomskog rada korišten je RPLIDAR A2. Koji je prikazan na slici 2. RPLIDAR A2 proizvodi tvrtka SLAMTEC a razvio ga je tim RoboPeak. Predstavlja jeftini 2D laserski skener novije generacije [3]. Pogodan je za izgradnju 3D modela objekta, karte i SLAM-a. U tablici 1 prikazane su neke od osnovnih karakteristika RPLIDAR-a A2.

Tablica 1. Karakteristike RPLIDARA-a A2

Raspon mjerenja	0.2 – 16m
Frekvencija uzorkovanja	8K
Brzina vrtnje	5Hz – 15 Hz
Kutna rezolucija	0.9°
Napon	5V
Struja	450mA – 600mA
Snaga	2.25W – 3W
Izlaz	UART ( 3.3 V)
Temperaturni raspon	0°C-40°C
Kutni raspon	360°
Rezolucija raspona	≤1% raspona (≤12m) ≤2% raspona (12m ~ 16m)
Točnost	1% raspona (≤3m) 2% raspona (3-5m) 2.5% raspona (5-25m)

### 2.3. Orbbec Astra kamera

Orbbec Astra RGB-D kamera, odnosno 3D kamera sadrži računalni vid koji omogućuje različite funkcije kao što su prepoznavanje geste, prepoznavanje lica, trodimenzionalno mjerenje, trodimenzionalna rekonstrukcija scene, izgradnja karata prostora i percepcija okoliša. Astra, Astra S i Astra Pro pružaju mjerenje dubine, precizne konture i glatke gradijente [4]. Također imaju mogućnost filtriranja niskokvalitetnih dubinskih piksela. Različite verzije omogućuju različite mogućnosti RGB kamere kratkog dometa, dugog dometa i visoke rezolucije. Karakteristike kamere prikazane su u tablici 2, a na slici 3 prikazana je Orbbec Astra kamera.

Tablica 2. Karakteristike Orbbec Astra kamere

Domet	0.4m – 8m
RGB rezolucija slike	640 x 480 @30fps
Dubinska rezolucija slike	640 x 480 @30fps
Dimenzije	165mm x 30mm x 40mm
Temperatura	0°C-40°C
Napajanje	USB 2.0
SDK	Astra SDK ili OpenNI



Slika 3. Orbbec Astra kamera

### 3. PROGRAMSKA PODRŠKA

U ovom poglavlju je navedena i detaljno objašnjena različita programska podrška koja je korištena za izradu rješenja zadatka ovog diplomskog rada.

#### 3.1. ROS

ROS je robotski operacijski sustav koji korisnicima pruža zbirku svih paketa, usluga i alata potrebnih za razvoj robotike i programiranje robota. Glavne standardne usluge koje pruža ROS su:

- Slanje poruka između procesa
- Implementacija uobičajenih funkcija
- Hardverska apstrakcija
- Kontrola uređaja niske razine
- Upravljanje paketima

Glavna značajka ROS-a je njegova pristupačnost. Osim toga je besplatan i otvorenog koda (engl. *open source*) što ga čini još popularnijim među profesionalnim programerima, studentima, tvrtkama i znanstvenicima [5]. Neke od glavnih karakteristika zbog kojih ROS posjeduje toliku popularnost u svijetu robotike su :

- Besplatan je i otvorenog koda.
- Sadrži sve potrebne alate, pakete i biblioteke.
- Teoretski je neovisan o programskom jeziku. Provodi se u Pythonu, C++, Javi.
- Posjeduje sve veću zajednicu s velikom količinom akumuliranog znanja. Znanstvenici, istraživači i studenti dijele svoj rad te grade veliku bazu znanja.

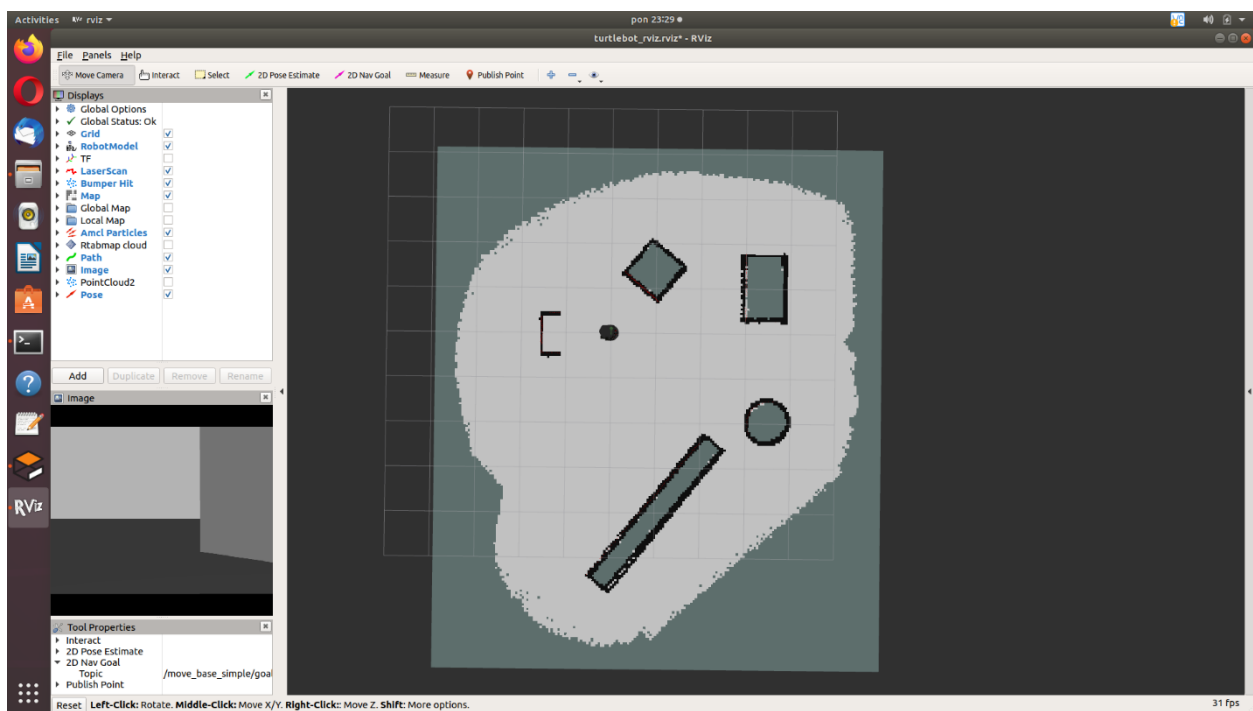
Osnovni zadatak ROS-a je da obavlja što više programa istovremeno, odnosno paralelno uz sinkronu izmjenu podataka. Da bi se omogućio takav način rada koristi se ROS i skup njegovih modela koji se zajedno zovu proračunski graf (engl. *Computation graph*). Proračunski graf je ROS-ov model koji se upotrebljava za programiranje i izgradnju robota. Radi tako da svaki ROS program predstavlja zaseban čvor (engl. *node*) proračunskog grafa koji je povezan s ostalim čvorovima preko tema (engl. *topics*). Teme omogućuju čvorovima izmjenu poruka, podataka te traženje usluga od drugih čvorova. Također mogu zatražiti i predati podatke od servera parametara. Kako bi se komunikacija između čvorova odvijala preko tema zaslužan je ROS proces pod nazivom glavni čvor (engl. *master*). Osim omogućivanja komunikacije između čvorova, služi još i za ažuriranje i kontroliranje podataka na serveru parametara. Ovakav način rada vrlo je pogodan

za programiranje i izgradnju robota koji se sastoje od više podsustava i hardvera. Metoda proračunskog grafa izgrađena je od više glavnih koncepata, a ti koncepti su: teme (engl. *topics*), čvorovi (engl. *nodes*), poruke (engl. *messages*), glavni čvor, usluge (engl. *services*), akcije (engl. *actions*), server parametara (engl. *parameter server*) i skupovi podataka (engl. *bags*). Čvorovi su pokrenuti ROS procesi, tj. programi. Predstavljaju procese vezane uz motore, senzore, obradu podataka. Svaki od tih čvorova ima posebnu identifikacijsku oznaku s kojom je registriran u master čvoru. Većina napisanog koda se nalazi unutar čvorova, te samim time predstavljaju središnji dio programiranja robota u ROS operacijskom sustavu. ROS čvorovi šalju podatke drugim čvorovima i primaju podatke od drugih čvorova. Isto tako primaju i šalju zahtjeve za izvršavanje nekih akcija nad drugim čvorovima. Čvorovi podatke međusobno šalju preko tema, koje predstavljaju poveznicu između njih. Podatke međusobno šalju sustavom dijeljenja temeljenom na principu pretplata/objava (engl. *subscribe/publish*). Ako čvor želi poslati neku poruku on mora objaviti tu poruku na određenu temu, a kako bi uspjeli primiti neku poruku isto tako moraju biti pretplaćeni na tu temu. Prijenos podataka na način pretplata/objava između čvorova je anonimn. Što znači da čvor neće znati od kojeg drugog čvora je poruka došla. Već samo zna preko koje teme će poslati i primiti poruku do i od tog čvora. Podaci koji se mogu slati su naredbe za pokretanje motora i različitih aktuatora, podaci sa senzora itd. Kako bi se komunikacija među čvorovima uspješno odvijala svi čvorovi moraju biti registrirani na master čvor. Master čvor omogućuje komunikaciju među čvorovima, odnosno slanje i primanje poruka i podataka. Poruke se sastoje od jednostavnijih tipova podataka npr. boolean, string, integer i samih poruka, a čine složenu podatkovnu strukturu. Razlika između usluga i tema je to što usluge imaju sinkroni prijenos podataka za razliku od tema koje imaju asinkroni. Koriste se najčešće za radnje koje imaju definirani početak i kraj te jedinstveno rješenje. Čvorovi usluge mogu pozvati ili pružiti u bilo kojem trenutku. Isto tako čvorovi mogu pozvati i akcije, no razlika između akcija i usluge je to što se akcije koriste za radnje koje se obavljaju kroz neko određeno vrijeme. Skupovi podataka služe za spremanje svih poruka poslanih među čvorovima. Također skupovi podataka omogućuju i ponovni prikaz podataka ukoliko je to potrebno. Takav način spremanja podataka uveliko doprinosi kod rješavanja problema prilikom programiranja, kao što su ispravljanje grešaka i simuliranje podataka. Na server parametara se spremaju podaci koji su podijeljeni među čvorovima. Najčešće se tu spremaju podaci koji se često koriste pri programiranju robota.

ROS također posjeduje programe i alate za stvaranje robota u simuliranom svijetu, kao što su alati RVIZ i Gazebo koji su opisani u daljnjem tekstu.

## 3.2. RVIZ

RVIZ je alat koji služi za 3D vizualizaciju ROS-a. Podaci koji su dobiveni iz različitih senzora kao što su Kinect, rplidar odnosno kamera mogu se vizualizirati pomoću RVIZ alata. RVIZ je jedan od važnijih i korisnijih alata pri radu s Turtlebotom i ROS-om. Alat dolazi sa instalacijom ROS-a. Pruža mogućnost vizualizacije podataka iz navigacijskog snopa. Također vizualizira svijet na način kako ga robot vidi, u ovom slučaju je to korisno jer je moguće vidjeti oblak točaka oko Turtlebota, izgrađenu kartu lidara te globalne i lokalne putanje koje je predvidio planer. Preko RVIZ-a moguće je odrediti lokaciju robota na karti, mjeriti udaljenost na karti te dobivati relativne koordinate robota [6].



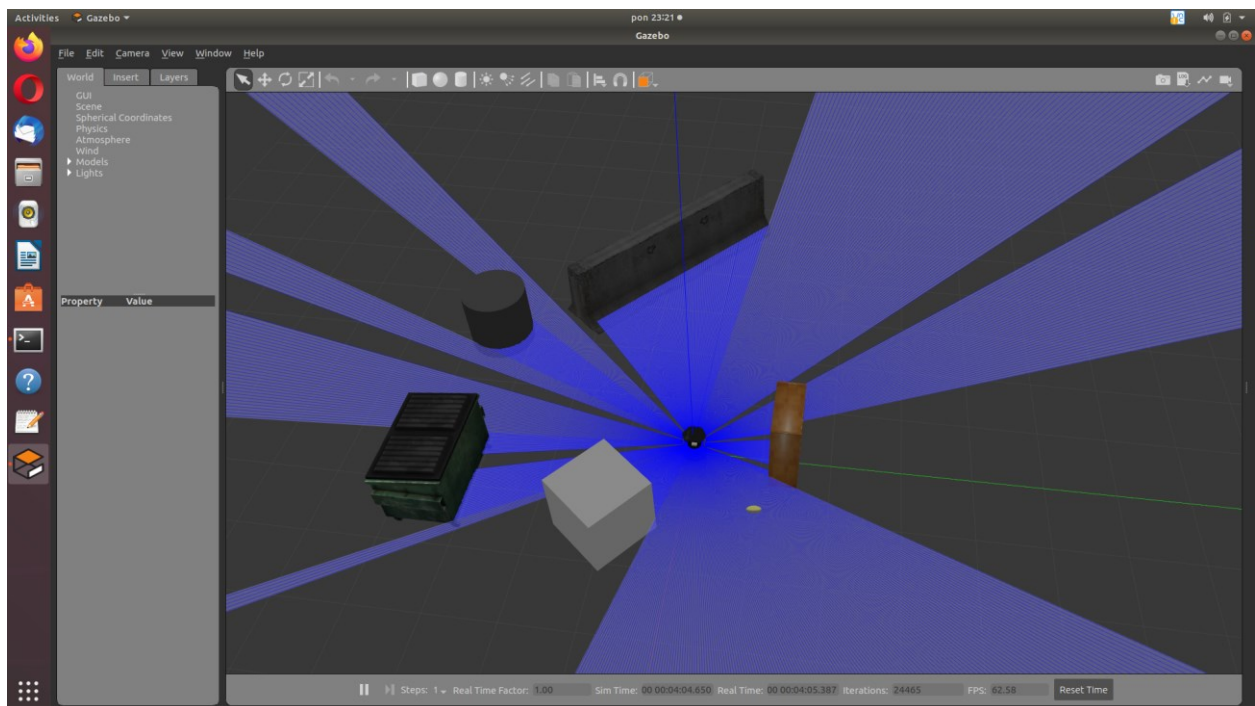
Slika 4. Prikaz RVIZ alata

Na slici 4 prikazan je RVIZ alat na kojem je vidljiva izgrađena karta okoline i pozicija robota unutar karte. Na lijevoj strani slike prikazane su različite opcije koje RVIZ omogućava kao što su stvarna slika koju robot vidi, te različite opcije koje je moguće prikazati npr. LaserScan, Path, Global Map, Local Map, itd. U gornjem dijelu slike vidljive su opcije koje služe za navigaciju i lokalizaciju robota. 2D Nav Goal omogućuje korisniku da pošalje cilj navigaciji postavljanjem željenog položaja koji robot treba postići, a 2D Pose Estimate omogućuje korisniku da inicijalizira sustav lokalizacije koji koristi navigacijski stog postavljanjem položaja robota u svijetu.

### 3.3. Gazebo

Gazebo je trodimenzionalni open-source robotski simulator koji omogućuje stvaranje i simuliranje robota u kompleksnom vanjskom ili unutrašnjem prostoru. Dolazi sa skupom ROS API-a koji nam omogućuju da promijenimo i dobijemo informacije o različitim aspektima simuliranog svijeta [7]. Neke od najvažnijih karakteristika Gazebo simulatora su:

- Otvorenog je koda
- Riječ je o multirobotičkom simulatoru koji omogućuje da izvodimo unutrašnje i vanjske simulacije
- Turtlebot može fizički ući u interakciju s predmetima
- Povratne informacije dobivene i prikupljene od senzora su realne
- Jednostavan je za korištenje, intuitivan i dobro dizajnirani simulator koji daje mogućnost pokretanja robota, algoritme ispitivanja i projektiranja robota
- Koristi URDF 3D model
- Moguće je dizajnirati prilagođeni svijet
- Robotu je moguće dodati senzore



Slika 5. Prikaz Gazebo alata

Na slici 5 prikazan je Gazebo alat. Na slici je prikazan simulirani mobilni robot Turtlebot2 u simuliranom svijetu. Simulirani robot sadrži sve simulirane senzore koje posjeduje i pravi robot. Te je moguće prikupljati različite podatke sa navedenih senzora. Na slici su vidljive laserske zrake od simuliranog RP Lidara.



## 4. SLAM ALGORITMI

U ovom poglavlju navedeni su i detaljno opisani SLAM algoritmi koji su korišteni za izradu ovog diplomskog rada. SLAM algoritmi koji su korišteni: GMapping, Dynamic Window Approach, RTAB-Map i Cartographer. Svaki od navedenih algoritama je u daljnjem tekstu detaljno opisan.

### 4.1. Simultana lokalizacija i mapiranje

Mapiranje i lokalizacija su problemi koji se rješavaju zajedno, a kombinirani problem se naziva simultana lokalizacija i mapiranje odnosno SLAM. Jedan od važnijih procesa SLAM algoritma je „zatvaranje petlje“ (engl. *loop closure*) u kojoj robot mora razumjeti da se njegova trenutna pozicija nalazi na mjestu na kojem je bio prije. Ako su prijašnji položaji robota točno izračunati, lanac pozicija na kojima se nalazio robot pretvara se u petlju koja poprima stvarni oblik sobe. Odometrija predstavlja robotsko uvjerenje o načinu kako se kreće. Izgradnja karte okoline neizvediva je bez promatranja okoline i odometrije robota. Mjerenja bi trebala biti točno povezana s pozicijom promatranja robota kako bi se dobio točan i cjelovit prikaz okoliša u kojem se nalazi robot. Još jedan težak i važan korak SLAM-a je „pridruživanje podataka“. U tom koraku robot mora shvatiti da promatra obilježja koja je već prije vidio. U stvarnom svijetu, promatranje objekta nije lagano bez znanja o tome gdje se objekt nalazi. Višestruka opažanja objekta mogu predstavljati višestruk prikaz jednog objekta ili nekoliko prikaza više objekata. Određivanje što je od toga stvarni slučaj u trenutnom robotskom svijetu je posao za probabilističku obradu svojstvenu grafičkom kartiranju, u kojoj se podaci uključuju i filtriraju kako bi se izgradila probabilistička distribucija svih mogućih slučajeva. Najbolje procijenjeni slučaj predstavlja uvjerenje robota, najvjerojatnije stanje svijeta koje se na kraju integrira u kartu koju će robot koristiti [8].

Ako je mjerenje robota potpuno točno, odnosno bez nesigurnosti u mjerenjima, tada bi se putanja robota mogla točno izračunati. Odometrija robota, postupak u kojem robot procjenjuje vlastito kretanje, utječe na sposobnost pravilnog lokaliziranja objekta. Bez nesigurnosti, SLAM bi predstavljao samo problem mapiranja, zato što bi lokalizacija bila dovršena. Mapiranje bi uključivalo postavljanje opažanja robota u odnosu na lokaciju robota uzduž rute, pa bi se i opažanja mogla točno pozicionirati.

Pošto svi senzori posjeduju određenu pogrešku u mjerenjima, najsuvremeniji SLAM algoritmi predstavljaju položaj robota i značajki kao slučajnu varijablu koja predstavlja lokaciju robota ili značajki. Uključivanje pogreške u strukturu karte omogućuje promjenu vjerovanja robota s

obzirom na novu informaciju. Kad god je potrebno izgraditi kartu, robot može izračunati najvjerojatniji trenutačni položaj značajki, prepreka te samoga sebe.

Postoji nekoliko SLAM algoritama visokih performansi, od kojih svaki zahtijeva korištenje različitih vrsta mjernih senzora kako bi ih se organiziralo u drugačiju strukturu. Što je više podataka koje robot preuzima iz okoliša, to točnije može izgraditi kartu.

Tablica 3. Popularni ROS kompatibilni LIDAR i vizualni SLAM algoritmi

	Inputs							Online Outputs			
	Camera			IMU	Lidar		Odom	Pose	Occupancy		Point Cloud
	Stereo	RGB-D	Multi		2D	3D			2D	3D	
GMapping					✓		✓	✓	✓		
TinySLAM					✓		✓	✓	✓		
Hector SLAM					✓		✓	✓	✓		
ETHZASL-ICP					✓	✓	✓	✓	✓		Dense
Karto SLAM					✓		✓	✓	✓		
Lago SLAM					✓		✓	✓	✓		
Cartographer					✓	✓	✓	✓	✓		Dense
BLAM						✓		✓			Dense
SegMatch						✓					Dense
VINS-Mono				✓				✓			
ORB-SLAM2	✓	✓									
S-PTAM	✓							✓			Sparse
DVO-SLAM		✓						✓			
RGBiD-SLAM		✓									
MCPTAM	✓		✓					✓			Sparse
RGBDSLAMv2		✓					✓	✓		✓	Dense
RTAB-Map	✓	✓	✓		✓	✓	✓	✓	✓	✓	Dense

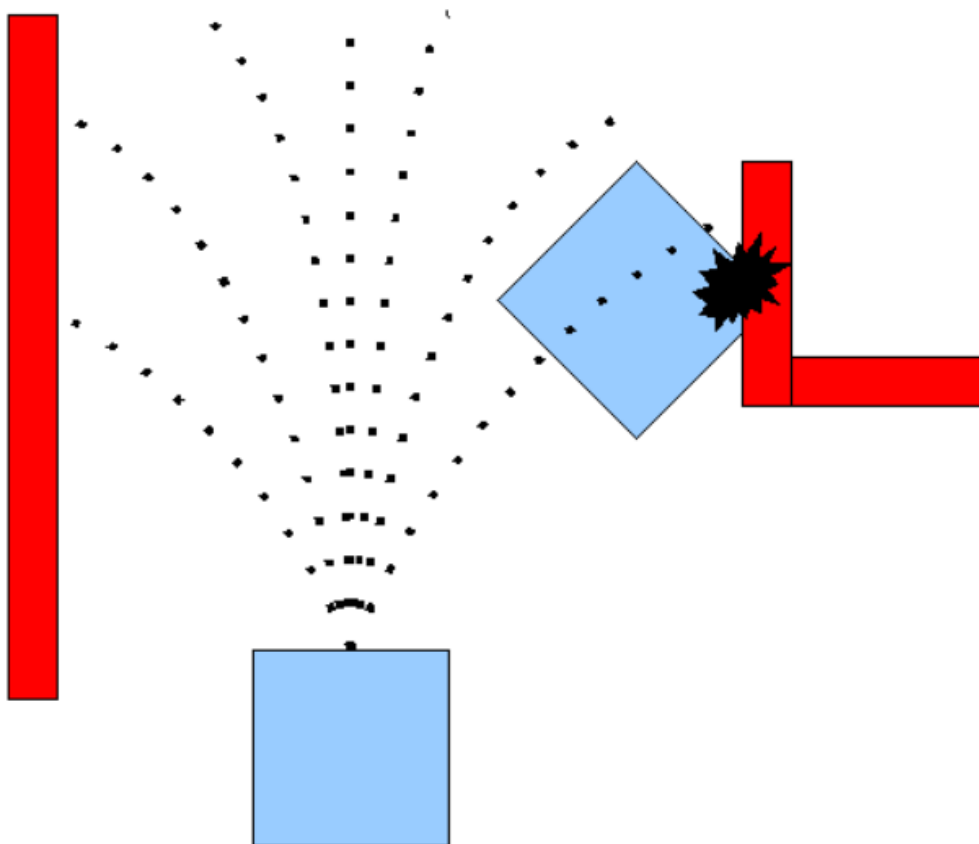
U tablici 3 prikazani su jedni od najpopularnijih ROS kompatibilni LIDAR i vizualni SLAM algoritmi s njihovim podržanim ulazima i izlazima. Ulazi (engl. *inputs*) predstavljaju senzore koji se mogu koristiti u algoritmima. To mogu biti dubinska kamera, stereo kamera, lidar, itd. Stupac multi označava algoritme koji mogu koristiti više kamera istovremeno kako bi se povećalo vidno polje sustava. Dok izlazi (engl. *outputs*) predstavljaju izlaze koji se dobiju algoritmima. Algoritmi koji su korišteni za izradu ovog diplomskog rada detaljnije su opisani u daljnjem tekstu.

## 4.2. Dynamic Window Approach

Dynamic Window Approach (DWA) je ROS-ov paket koji pruža robotu implementaciju dinamičkog prozorskog pristupa lokalnoj robotskoj navigaciji u ravnini. S obzirom na globalni plan koji treba slijediti i kartu vrijednosti, lokalni planer šalje naredbe brzine mobilnoj bazi robota. Ovaj paket podržava svakog robota čiji se otisak može predstaviti kao konveksni poligon ili kružnica, i prikazuje njegovu konfiguraciju kao ROS parametre koji se mogu postaviti u datoteci za pokretanje. Parametre DWA planera moguće je dinamički podešavati [9].

Paket `dwa_local_planner` osigurava kontroler koji upravlja mobilnom bazom robota u ravnini. Taj kontroler služi za povezivanje plana puta s robotom. Pomoću karte planer stvara kinematičku putanju kako bi robot stigao s početne na ciljnu lokaciju. Tijekom kretanja od početne do ciljne lokacije planer stvara lokalno oko robota, funkciju vrijednosti, predstavljenu kao kartu u obliku kvadratne rešetke (engl. *grid map*). Ova funkcija vrijednosti kodira troškove prolaska kroz ćelije mreže.

Posao kontrolera je koristiti ovu funkciju vrijednosti za određivanje  $dx, dy, d\theta$  brzina koje se šalju robotu. Na slici 6 prikazana je metoda unaprijednog simuliranja putanje robota pomoću DWA planera.



Slika 6. Dynamic Window Approach

Cilj ovog algoritma je maksimizirati ciljnu funkciju koja uzima u obzir napredak do cilja, izmicanje od prepreke (najbliža udaljenost između prepreke i puta) i robotovu brzinu naprijed. Kao rezultat toga, DWA proizvodi parove brzina ( $v, \omega$ ) koji predstavljaju kružne putanje koje su optimalne za lokalne uvijete robota. Prostor za pretraživanje sveden je na dinamički prozor koji sadrži samo brzine koje se mogu postići unutar sljedećeg vremenskog intervala prema zadanim

translacijskim i rotacijskim brzinama i ubrzanjima. Princip rada algoritma sastoji se od nekoliko glavnih koraka:

- DWA diskretno uzrokuje brzine u prostoru brzina
- Za svaku uzorkovanu brzinu, unaprijed izvesti simulaciju iz trenutnog stanja robota kako bi se predvidjelo što bi se dogodilo da se uzorkovana brzina promijenila kroz neko (kratko) vremensko razdoblje.
- Vrednuje svaku putanju koja je rezultat unaprijedne simulacije, koristeći metriku koja uključuje karakteristike kao što su: blizina preprekama, blizina cilju, blizina globalnoj putanji i brzina
- Odbaciti nedopuštene putanje (one koje se sudaraju s preprekama)
- Odabrati putanju s najvećim brojem bodova i poslati pridružene brzine mobilnoj bazi robota
- Ponavljati korake dok robot dostigne svoj cilj.

### 4.3. GMapping

GMapping je 2D SLAM algoritam koji koristi odometriju robota i mjerenja koja dolaze sa senzora (npr. Lidar i sonar) kako bi procijenio poziciju robota i napravio kartu u kojoj robot radi koja je predstavljena kao mreža popunjenosti (engl. *Occupancy grid*). Koristeći Bayesov pristup za spajanje promjene položaja i senzorskih mjerenja u probabilističkom okviru, algoritam rješava SLAM koristeći Rao-Blackwellized filter čestica [10]. Jednadžba (1) prikazuje traženje sljedeće vjerojatnosti  $p(x_t, m_t | z_t, u_t)$  nakon faktorizacije postupkom Rao-Blackwellization.

$$p(x_t, m_t | z_t, u_t) = p(x_t | z_t, u_t) p(m_t | x_t, z_t) \quad (1)$$

U jednadžbi (1), vjerojatnost robotske pozicije  $x_t$ , i mape  $m_t$  uz mjerenja koja dolaze sa senzora  $z_t$  i odometrije robota  $u_t$ , prikazana je kao produkt dvije vjerojatnosti, vjerojatnosti  $p(x_t | z_t, u_t)$  da ima poziciju  $x_t$  zadanu mjerenjima  $z_t$  i odometrijom  $u_t$ , puta vjerojatnost  $p(m_t | x_t, z_t)$  da je karta  $m_t$ , ako je pozicija robota  $x_t$  a mjerenja senzora  $z_t$ . U filteru čestica, hipoteza za poziciju robota i karta njegova okoliša se nalaze u nizu čestica. Svaka čestica  $\langle x^i, m^i, w^i \rangle$  skupa  $\{\langle x^1, m^1, w^1 \rangle \dots \langle x^N, m^N, w^N \rangle\}$  ima saznanje prošle pozicije robota  $x^i$ , izrađenu kartu od prošle pozicije  $m^i$  i težinu  $w^i$ . Težina  $w^i$  je vrijednost vjerojatnosti koja prikazuje koliko je vjerojatno da čestica predstavlja prethodnu poziciju robota i mapira mjerenja sa senzora i odometrije robota. Izazov je kako odabrati svaku česticu tako da stvarno predstavlja robotsko uvjerenje o svojoj poziciji i stvorenoj karti. Zbog toga se GMapping temelji na robotskoj odometriji i podudaranju skeniranja (registraciji između uzastopnih laserskih skeniranja) kako bi se procijenili parametri funkcije gustoće vjerojatnosti iz koje će se generirati skup čestica u svakoj iteraciji filtra. Tako se drastično smanjuje broj čestica  $N$  potrebnih za točnu SLAM procjenu.

Također tako se smanjuju troškovi računanja uključeni u proces. Algoritmom se izbjegava problem iscrpljivanja čestica, situacija svojstvena filterima čestica u kojima čestice velike težine mogu biti odbačene. Kako bi se to uskladilo, čestice se ponovno uzorkuju samo ako je procijenjeni efektivni broj čestica  $N_{\text{eff}}$  ispod zadanog praga. Efektivne čestice su one čestice koje dovoljno dobro aproksimiraju poziciju robota.

GMapping algoritam implementira se u ROS-u pomoću jednog ROS čvora koji se naziva `slam_gmapping`. Čvor se pretplaćuje na temu `scan` sa senzorskim podacima i temu `tf` s podacima o odometriji robota. Svaki put kada čvor primi poruku s navedenih tema, objavljuje poruku na temi `map` (koja objavljuje ažuriranu kartu) i ažurira transformaciju iz referentnog okvira `map` u referentni okvir `odom` u `robot_tree`. Tablica 4 prikazuje relevantne teme objavljene ili pretplaćene od strane ROS čvora `slam_gmapping`.

Tablica 4. Lista relevantnih pretplaćenih/objavljenih tema od GMapping ROS paketa

Topic	Message	Type
Tf	Tf/tfMessage	Subscribe
Scan	Sensor_msgs/LaserScan	Subscribe
Map	Nav_msgs/OcupancyGrid	Published

ROS omogućuje online podešavanje parametara algoritma. Neki od parametara koji se mogu podešavati za GMapping algoritam prikazani su u tablici 5.

Tablica 5. Lista parametara za GMapping ROS paket

Parametar	Vrijednost	Opis
Particles	30	Broj čestica filtra
linearUpdate	1.0	Pređena udaljenost koja pokreće obradu skeniranja
angularUpdate	0.5	Zakretni kut koji pokreće obradu skeniranja
resampleThreshold	0.5	$N_{\text{eff}}$ prag ponovnog uzorkovanja
Delta	0.05	Rezolucija mreže popunjenosti (m)
Occ_thresh	0.25	Prag za smatranje ćelije zauzetom

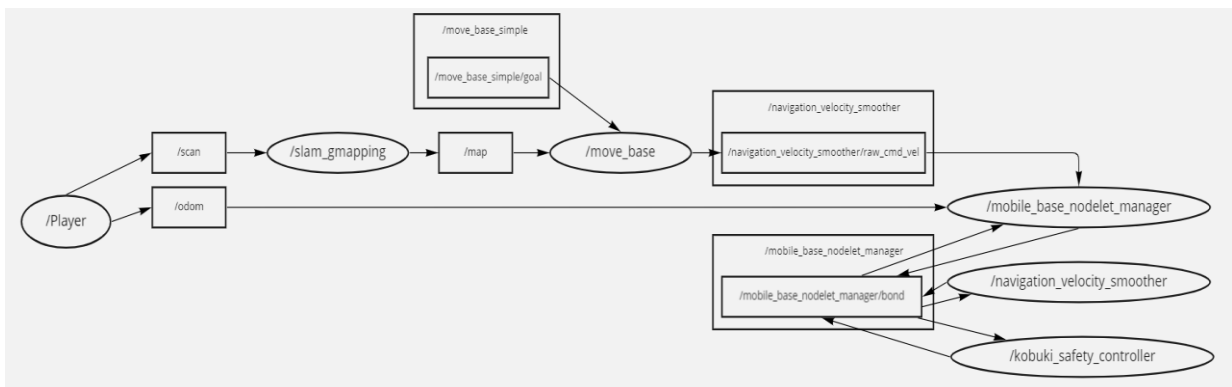
### 4.3.1. GMapping lidar

Za izgradnju karte korištenjem GMapping SLAM algoritma uz korištenje lidara potrebno je navedeni ROS čvor pretplatiti na određene teme koje su neophodne za uspješnu izgradnju karte okoline. Nakon što je ROS čvor pretplaćen na potrebne teme, SLAM algoritam koristi poruke sa tih tema te objavljuje određene teme u kojima su sadržane poruke sa informacijama o kreiranoj karti okoline. U tablici 6 prikazane su najvažnije potrebne teme na koje se ROS čvor pretplaćuje ili objavljuje. Parametri algoritma prikazani su u prilogu 4.3.1.1.

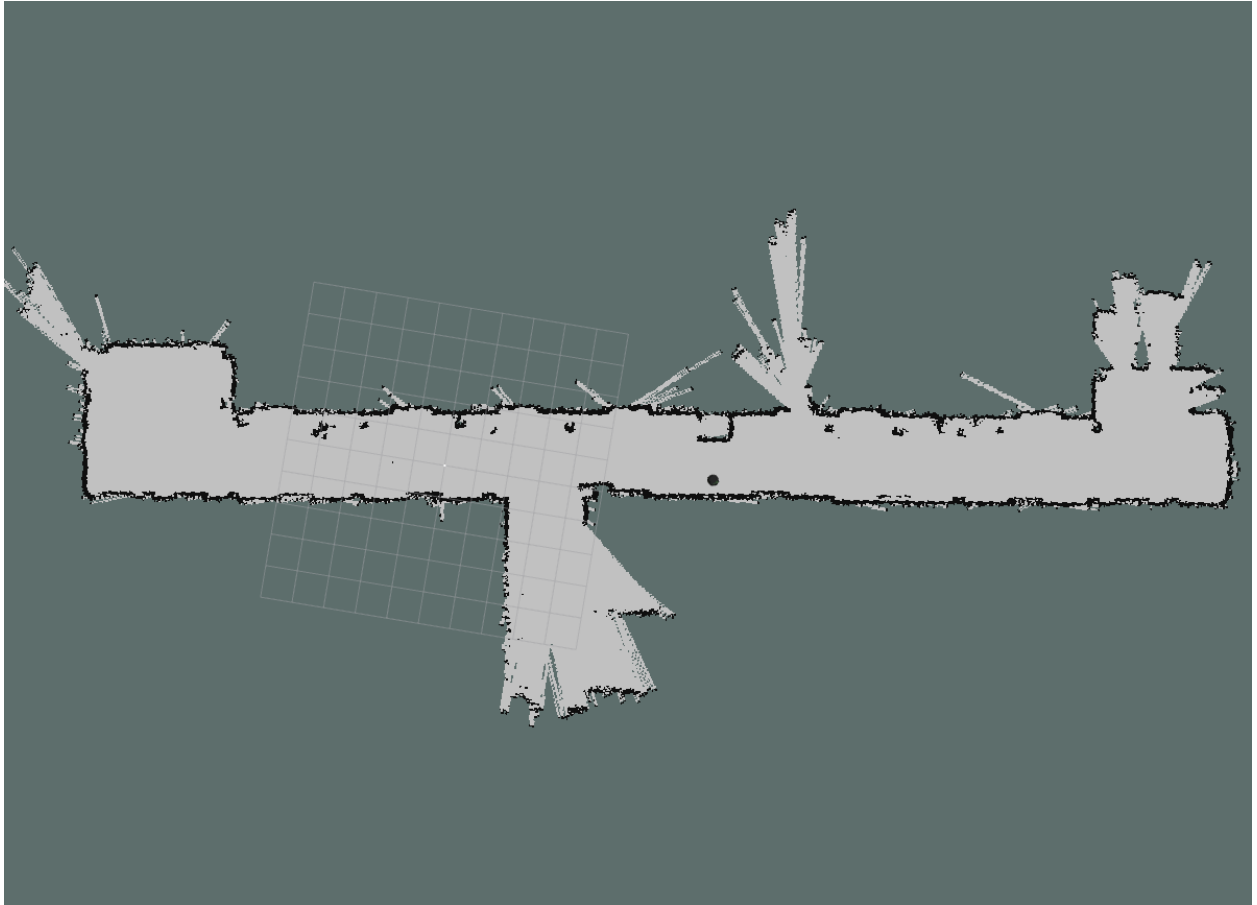
Tablica 6. Lista pretplaćenih/objavljenih tema za GMapping lidar

Topic	Message	Type
Tf	Tf/tfMessage	Subscribe
Scan	Sensor_msgs/LaserScan	Subscribe
Map	Nav_msgs/OcupancyGrid	Published
Map_metadata	Nav_msgs/MapMetaData	Published

Na slici 7 prikazan je rqt graf GMapping SLAM algoritma. Rqt graf je GUI dodatak iz Rqt alata. Pomoću rqt grafa moguće je prikazati ROS graf aplikacije. Iz rqt grafa vidljivi su svi pokrenuti čvorovi kao i komunikacija između njih. Svaki čvor i tema su prikazani unutar njihovog imenskog prostora.



Slika 7. RQT graf GMapping SLAM algoritma uz korištenje lidara



Slika 8. Karta hodnika izgrađena GMapping algoritmom korištenjem lidara

Na slici 8 prikazana je karta hodnika trećeg kata zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku snimljena GMapping SLAM algoritmom uz korištenje lidar senzora koji prikuplja i šalje sve potrebne informacije SLAM algoritmu za izgradnju karte i navigaciju robota.

#### 4.3.2. GMapping dubinska kamera

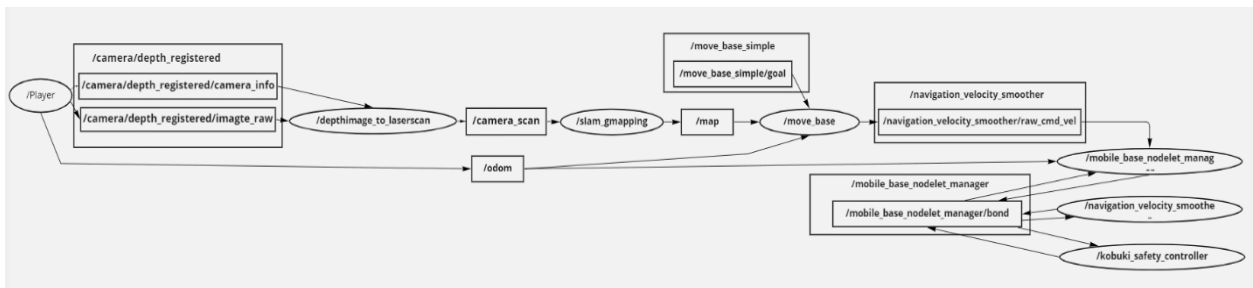
Algoritam GMapping prvenstveno je namijenjen za izradu karte pomoću laserskog skeniranja odnosno sa lidarom. Kako bi se navedeni algoritam mogao koristiti i sa dubinskom kamerom koja daje 3D oblak točaka umjesto lidara, potrebno je pretvoriti 3D podatke o dubini u lasersko skeniranje. Da bi se izvršila pretvorba 3D podataka o dubini u lasersko skeniranje potrebno je koristiti dodatan ROS čvor koji se naziva `depthimage_to_laserscan`. Navedeni ROS čvor uzima sliku dubine i generira 2D lasersko skeniranje na temelju određenih parametara. `Depthimage_to_laserscan` koristi lijenu (engl. *lazy*) pretplatu i neće se pretplatiti na sliku ili podatke o kameri ukoliko ne postoji pretplatnik za skeniranje. U tablici 7 prikazane su teme na

koje se depthimage\_to\_laserscan pretplaćuje odnosno objavljuje. Parametri algoritma prikazani su u prilogu 4.3.2.1.

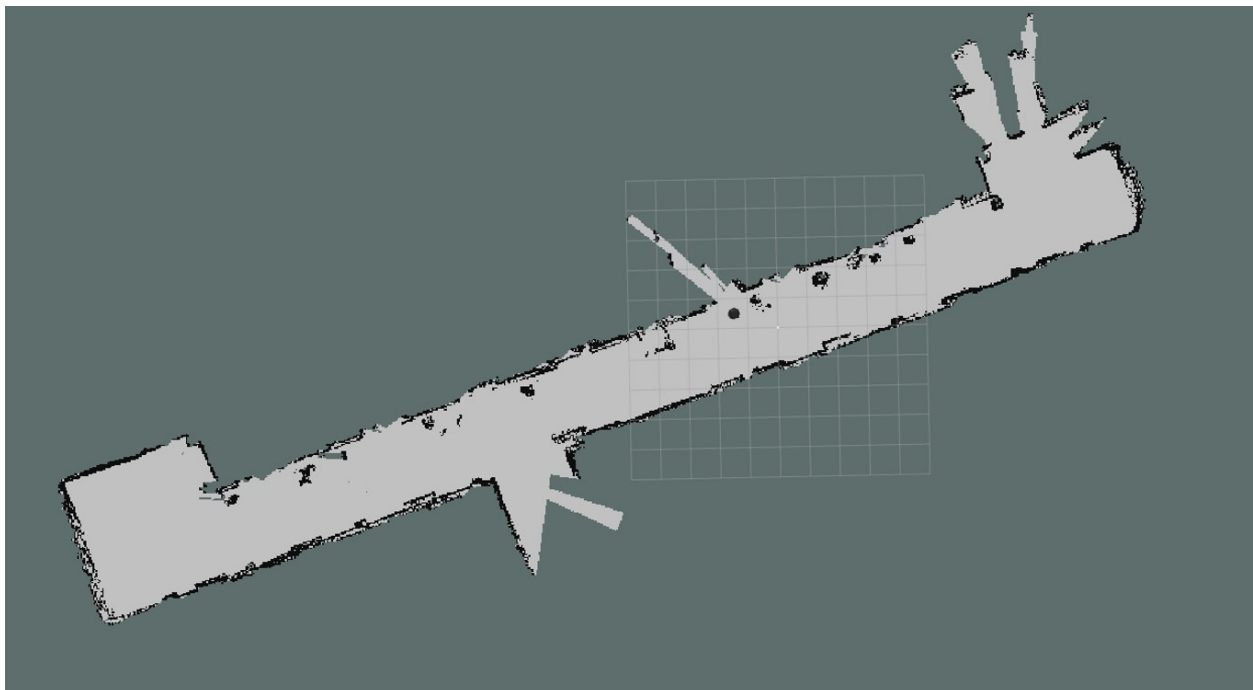
Tablica 7. Lista pretplaćenih/objavljenih tema za depthimage\_to\_laserscan čvor

Topic	Message	Type
Image	Sensor_msgs/Image	Subscribe
Camera_info	Sensor_msgs/CameraInfo	Subscribe
Scan	Sensor_msgs/LaserScan	Published

Na slici 9 prikazan je rqt graf GMapping algoritma sa korištenom dubinskom kamerom umjesto lidara.



Slika 9. Rqt graf GMapping algoritma sa dubinskom kamerom



Slika 10. Karta hodnika izgrađena GMapping algoritmom korištenjem dubinske kamere



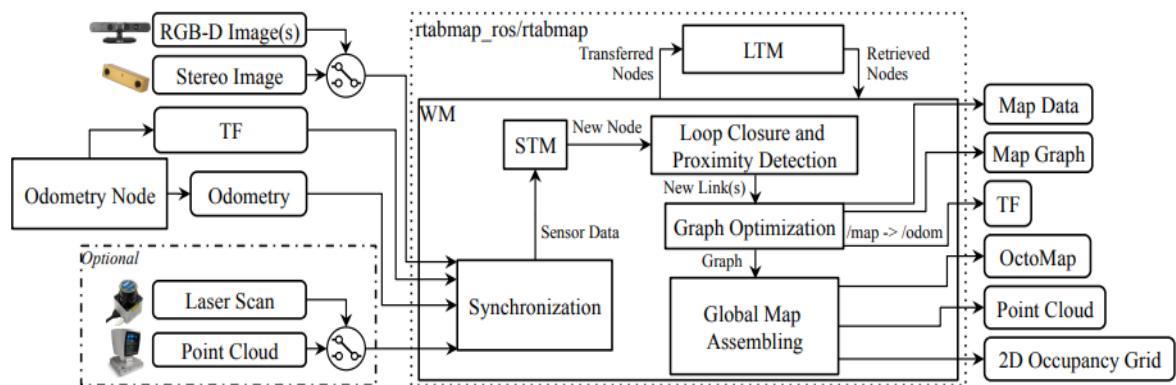
Na slici 10 prikazana je karta hodnika trećeg kata zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku snimljena GMapping algoritmom uz korištenje dubinske kamere kao senzora koji prikuplja i šalje sve potrebne informacije SLAM algoritmu za izgradnju karte i navigaciju robota.

#### 4.4. RTAB-Map

RTAB-Map je jedan od najpopularnijih SLAM algoritama koji je zasnovan na grafovima, od 2013 godine integriran je u ROS-u kao paket `rtabmap_ros`. Slika 11 prikazuje glavni ROS čvor `rtabmap`. Odometrija predstavlja vanjski ulaz u RTAB-Map, što znači da je SLAM moguće izvesti s bilo kojom vrstom odometrije kako bi se koristilo ono što je prikladno za datu aplikaciju i robota. Struktura karte je graf s čvorovima i vezama. Nakon sinkronizacije senzora, modul kratkotrajne memorije STM (engl. *Short-Term Memory*) kreira čvor koji sprema odometriju pozicije robota, podatke sa senzora i ostale dodatne informacije koje se koriste za različite module ( npr. Vizualne riječi za zatvaranje petlje, lokalnu mrežu zauzetosti za sastavljanje globalne karte). Čvorovi se kreiraju s fiksnom brzinom „`Rtabmap/DetectionRate`“ koja je u milisekundama a ovisi o tome koliko se podataka stvorenih iz čvorova treba međusobno preklapati. Na primjer, ako se robot kreće velikom brzinom a domet senzora je mali, potrebno je povećati brzinu detekcije kako bi se osiguralo preklapanje podataka uzastopnih čvorova. Međutim, ako se brzina detekcije postavi previsoko, nepotrebno se povećava upotreba memorije i vrijeme izračunavanja. Veze sadrže relativni položaj dva čvora. Postoje tri vrste veze: veze za susjedne čvorove, veze za zatvaranje petlje i veze za blizinu. Susjedske veze su dodane u STM između uzastopnih čvorova s transformacijom odometrije. Zatvaranje petlje i blizina veze se dodaju kroz detekciju zatvaranja petlje ili detekciju blizine. Sve veze se koriste kao ograničenja za optimizaciju grafa. Kada se na graf doda nova veza za zatvaranje petlje ili veza blizine, optimizacija grafa propagira izračunatu pogrešku na cijeli graf kako bi se smanjilo odometrijsko odstupanje. Uz optimizirani graf, OctoMap, Point Cloud i 2D Occupancy grid izlazi se mogu objaviti na vanjske module [11].

Pristup upravljanja memorijom RTAB-Map algoritma nadovezuje se na module upravljanja grafovima. Koristi se za ograničavanje veličine grafa, tako da se dugoročni online SLAM može postići u velikim okruženjima. Bez pristupa upravljanja memorijom, kako graf raste, vrijeme obrade za module kao što su modul za detekciju zatvaranja petlje, i detekciju blizine, optimizacija grafa i globalno sastavljanje karte mogu premašiti ograničenja u stvarnom vremenu, tj. vrijeme obrade može postati veće od vremena ciklusa izvođenja čvora. Memorija RTAB-Map podijeljena je na radnu memoriju (engl. *Working Memory (WM)*) i dugotrajnu memoriju (engl. *Long-Term*

*Memory (LTM)*). Kada se čvor prenosi na LTM, više nije dostupan za module unutar WM-a. Kada vrijeme ažuriranja RTAB-Map-a premašuje fiksni vremenski prag „Rtabmap/TimeThr“, neki čvorovi u WM se prenose na LTM radi ograničavanja veličine WM i smanjenja vremena ažuriranja. Slično kao kod fiksnog vremenskog praga, postoji i memorijski prag „Rtabmap/MemoryThr“ koji se koristi za postavljanje maksimalnog broja čvorova koje WM može spremati. Da bi se odredilo koje čvorove transferirati u LTM, mehanizam vrednovanja identificira lokacije koje su važnije od drugih, koristeći heuristiku kao što je duljina promatranja lokacije. Što je lokacija dulje promatrana to je lokacija vrjednija, te ju treba zadržati u radnoj memoriji. Prilikom stvaranja novog čvora STM inicijalizira težinu čvora na 0 i uspoređuje ju vizualno (dobivajući postotak odgovarajuće vizualne riječi) sa zadnjim čvorom u grafu. Ako su slični (s postotkom odgovarajuće vizualne riječi iznad praga sličnosti „Mem/RehearsalSimilarity“), težina novog čvora se povećava za 1 plus težina zadnjeg čvora. Težina zadnjeg čvora se resetira na 0, a zadnji čvor se odbacuje ako se robot ne kreće kako bi se izbjeglo nepotrebno povećavanje veličine grafa. Kada je dosegnut prag vremena ili memorije, najstariji od najmanje vrednovanih čvorova se prenosi u LTM. Kada se dogodi zatvaranje petlje s lokacijom u WM-u, susjedni čvorovi te lokacije mogu se vratiti iz LTM-a u WM za više zatvaranja petlje i detekciju blizine. Kako se robot giba po prethodno posjećenom području, tada se može postepeno prisjetiti prošlih lokacija kako bi proširio trenutačno sastavljenu kartu i lokalizirao se korištenjem prošlih lokacija.



Slika 11. Blok dijagram rtabmap ROS čvora

Slika 11 prikazuje blok dijagram rtabmap ROS čvora. Iz blok dijagrama je vidljivo da su nam potrebni sljedeći ulazi za rad rtabmap čvora. TF za definiranje položaja senzora u odnosu na bazu robota. Odometrija iz bilo kojeg izvora. Jedan od ulaza kamere (jedna ili više RGB-D slika ili stereo slika) s odgovarajućim porukama o kalibraciji. Ulazi koje možemo koristiti su još i lasersko skeniranje s 2D lidara ili oblak točaka iz 3D lidara. Sve poruke s ovih ulaza se zatim sinkroniziraju

i prosljeđuju SLAM algoritmu. Izlazi koje ovaj čvor daje su podaci o karti koji sadrže najnoviji dodani čvor s komprimiranim podacima senzora i grafom, map graph bez ikakvih podataka, korekciju odometrije, OctoMap (3D mreža popunjenosti), gusti oblak točaka i 2D mreža zauzetosti.

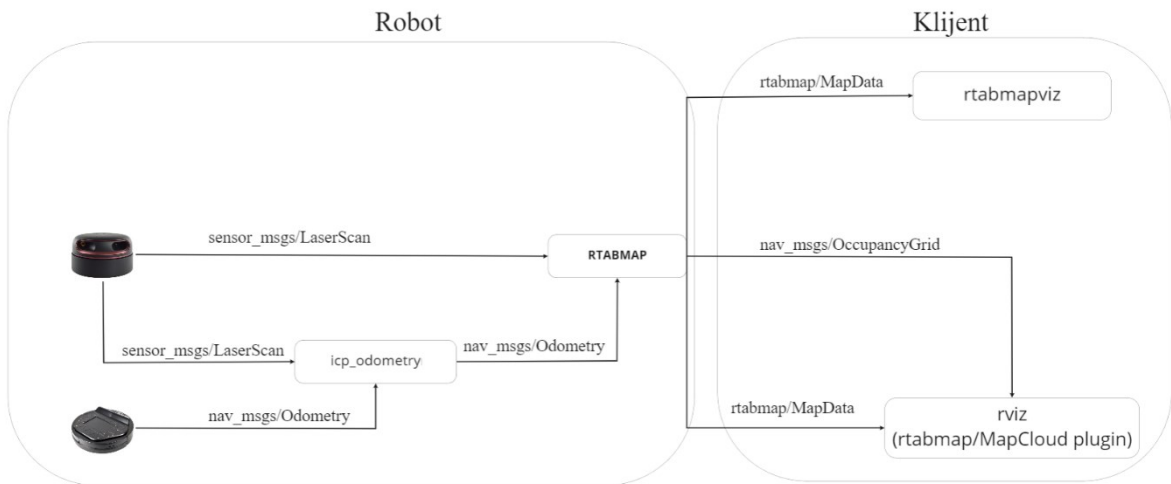
#### 4.4.1. RTAB-Map lidar

RTAB-Map je SLAM algoritam koji ima mogućnost korištenja različitih senzora za prikupljanje informacija koje su potrebne za mapiranje prostora. Moguće je koristiti lidar, dubinsku kameru ili kombinaciju lidara i dubinske kamere. U tablici 8 prikazan je popis relevantnih pretplaćenih/objavljenih tema od RTABMAP ROS paketa u kojem je korišten lidar senzor za mapiranje unutrašnjeg prostora.

Tablica 8. Popis relevantnih pretplaćenih/objavljenih tema od RTABMAP ROS paketa

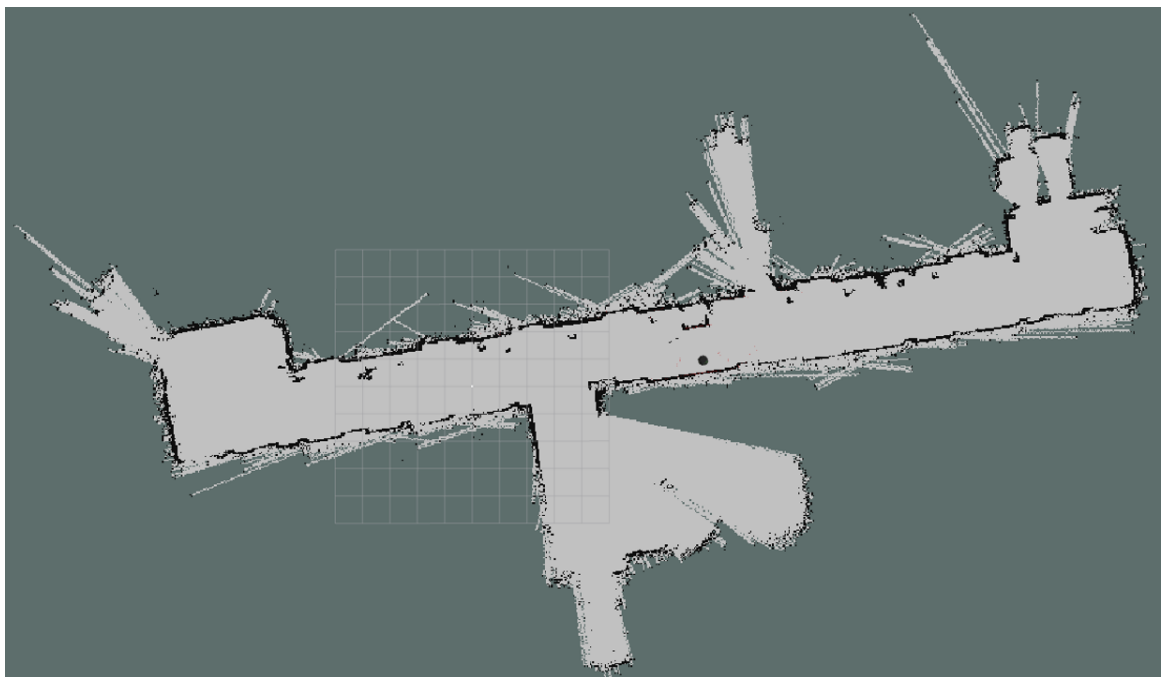
Topic	Message	Type
Tf	Tf/tfMessege	Subscribe
Scan	Sensor_msgs/LaserScan	Subscribe
Odom	Nav_msgs/Odometry	Subscribe
MapData	Rtabmap_ros/MapData	Published
Grid_map	Nav_msgs/OccupancyGrid	Published

Na slici 12 prikazana je konfiguracija robota za mapiranje unutrašnjeg prostora pomoću RTAB-Map SLAM algoritma uz korištenje lidar senzora. Na slici su vidljive sve teme i poruke na koje se RTAB-Map čvor pretplaćuje odnosno objavljuje. RTAB-MAP se pretplaćuje na teme koje sadrže senzorsko mjerenje i odometriju robota. A objavljuje teme koje sadrže informacije o karti prostora. Parametri algoritma prikazani su u prilogu 4.4.1.1.



Slika 12. Konfiguracija RTABMAP SLAM algoritma uz korištenje lidara

Na slici 13 prikazana je karta hodnika trećeg kata zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku snimljena RTAB-Map SLAM algoritmom uz korištenje lidara kao senzora za prikupljanje podataka potrebnih za izgradnju karte okoline.



Slika 13. Karta hodnika izgrađena RTAB-Map algoritmom korištenjem lidara

#### 4.4.2. RTAB-Map dubinska kamera

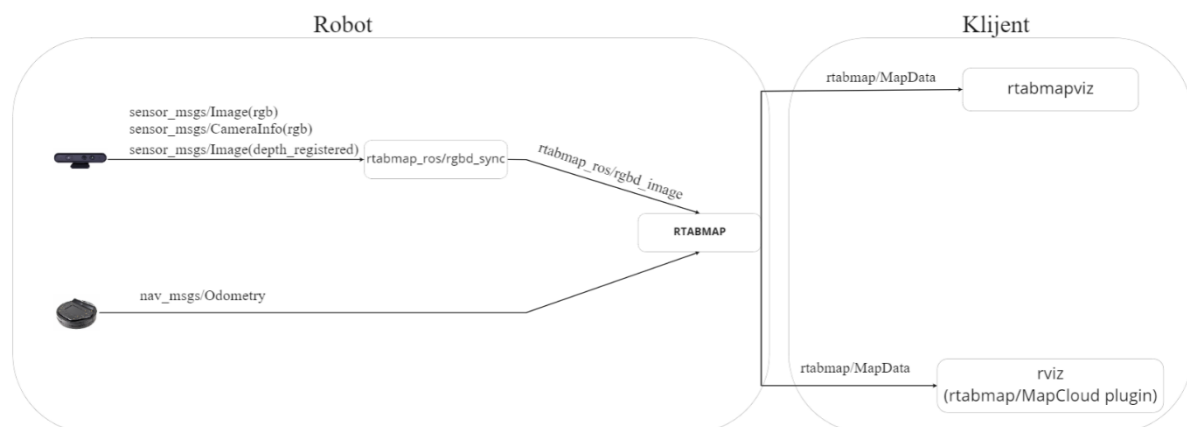
U tablici 9 prikazana je lista relevantnih pretplaćenih/objavljenih tema od RTABMAP ROS paketa u kojem je korištena dubinska kamera za mapiranje unutrašnjeg prostora. Za razliku od primjera iznad u kojem je korišten lidar za mapiranje prostora. Iz tablice je vidljivo da RTAB-Map

čvor objavljuje iste teme kao i u primjeru za lidar, dok su za mapiranje prostora potrebne informacije koje dolaze sa tema od dubinske kamere.

Tablica 9. Popis relevantnih pretplaćenih/objavljenih tema od RTABMAP ROS paketa

Topic	Message	Type
Tf	Tf/tfMessege	Subscribe
Odom	Nav_msgs/Odometry	Subscribe
Image	Sensor_msgs/Image(rgb)	Subscribe
CameraInfo	Sensor_msgs/CameraInfo	Subscribe
Image	Sensor_msgs/Image(depth_registered)	Subscribe
MapData	Rtabmap_ros/MapData	Published
Grid_map	Nav_msgs/OccupancyGrid	Published

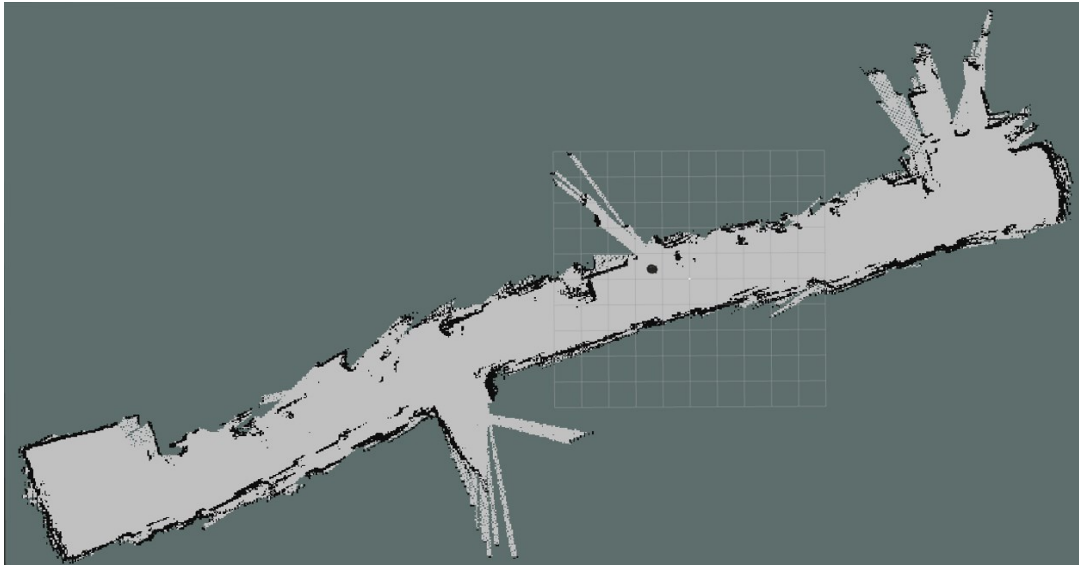
Na slici 14 prikazana je konfiguracije robota za mapiranje unutrašnjeg prostora pomoću RTAB-Map SLAM algoritma uz korištenje dubinske kamere kao senzora za mapiranje prostora [12]. Na slici su vidljive sve teme i poruke na koje se RTAB-Map čvor pretplaćuje odnosno objavljuje. RTAB-MAP se pretplaćuje na teme koje sadrže senzorsko mjerenje i odometriju robota. A objavljuje teme koje sadrže informacije o karti prostora. Teme koje dolaze sa dubinske kamere prije nego što se proslijede RTAB-Map čvoru potrebno je sinkronizirati. Za sinkronizaciju podataka koristi se rtabmap\_ros/rgbd\_sync čvor. Parametri algoritma prikazani su u prilogu 4.4.2.1.



Slika 14. Konfiguracija RTABMAP SLAM algoritma uz korištenje dubinske kamere

Na slici 15 prikazana je karta hodnika trećeg kata zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku snimljena RTAB-Map SLAM algoritmom uz

korištenje dubinske kamere kao senzora za prikupljanje podataka potrebnih za izgradnju karte okoline.



Slika 15. Karta hodnika izgrađena RTAB-Map algoritmom korištenjem dubinske kamere

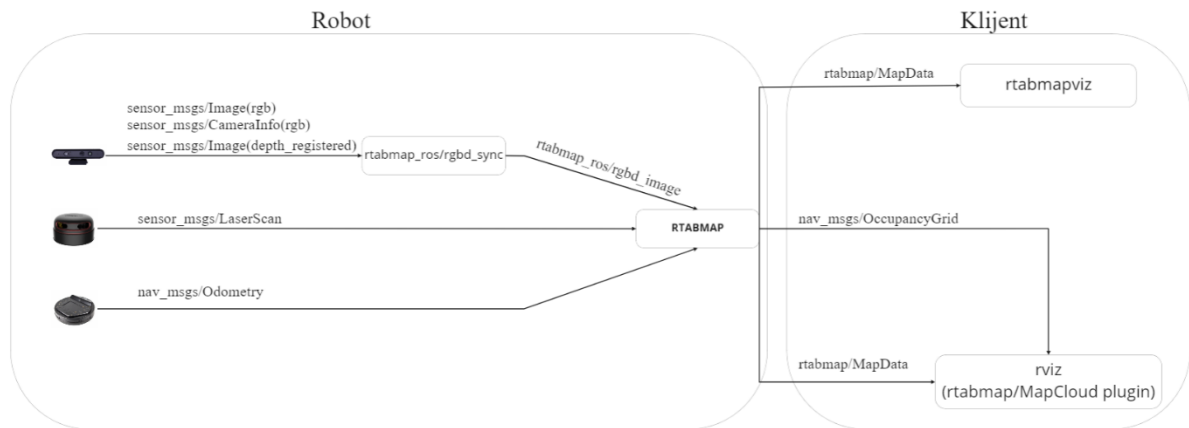
#### 4.4.3. RTAB-Map lidar i dubinska kamera

Za razliku od prethodna dva primjera mapiranja prostora pomoću RTAB-Map SLAM algoritma gdje su se odvojeno koristili lidar i dubinska kamera, u ovom primjeru korištena je njihova kombinacija za mapiranje prostora. U tablici 10 prikazana je lista relevantnih pretplaćenih/objavljenih tema od RTABMAP ROS paketa.

Tablica 10. Popis relevantnih pretplaćenih/objavljenih tema od RTABMAP ROS paketa

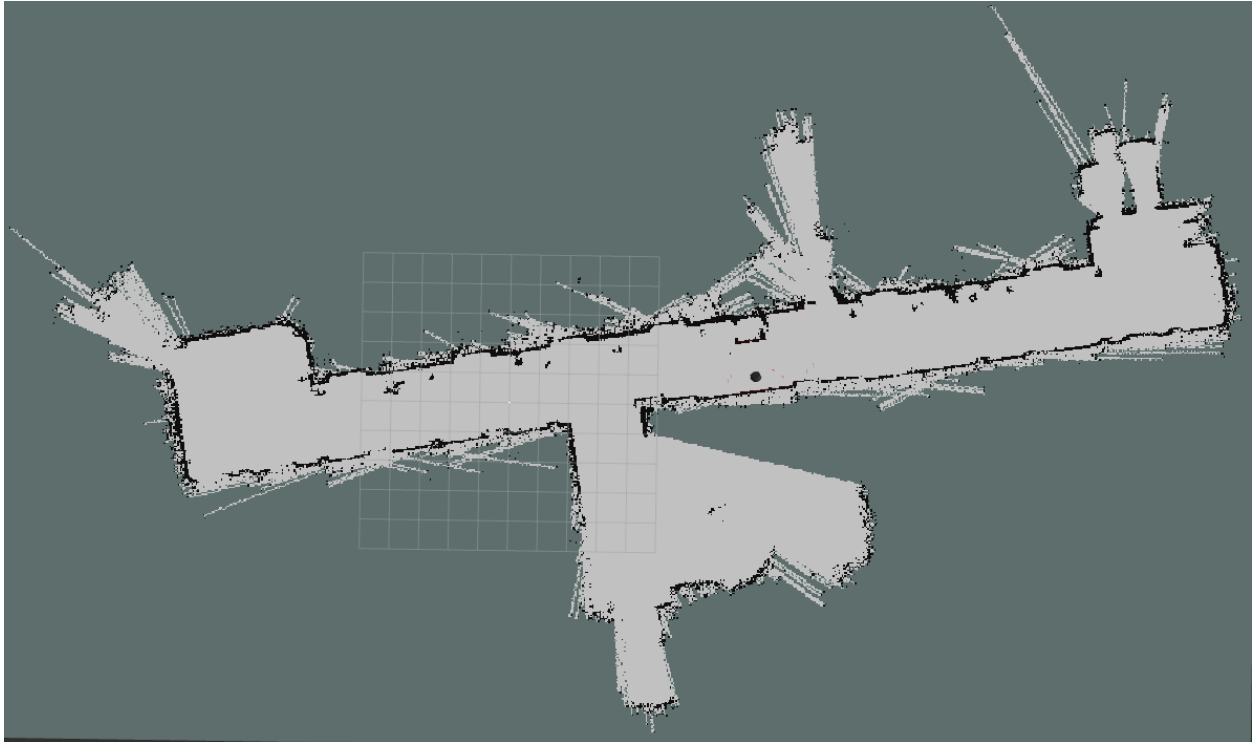
Topic	Message	Type
Tf	Tf/tfMessege	Subscribe
Odom	Nav_msgs/Odometry	Subscribe
Scan	Sensor_msgs/LaserScan	Subscribe
Image	Sensor_msgs/Image(rgb)	Subscribe
CameraInfo	Sensor_msgs/CameraInfo	Subscribe
Image	Sensor_msgs/Image(depth_registered)	Subscribe
MapData	Rtabmap_ros/MapData	Published
Grid_map	Nav_msgs/OccupancyGrid	Published

Na slici 16 prikazana je konfiguracija robota za mapiranje unutrašnjeg prostora pomoću RTAB-Map SLAM algoritma uz korištenje dubinske kamere i lidara kao senzora za prikupljanje podataka. Isto kao i u prošlom primjeru gdje je korištena dubinska kamera, podatke sa dubinske kamere je potrebno sinkronizirati uz pomoć `rtabmap_ros/rgbd_sync` čvora. Parametri algoritma prikazani su u prilogu 4.4.3.1.



Slika 16. Konfiguracija RTABMAP SLAM algoritma uz korištenje dubinske kamere i lidara

Na slici 17 prikazana je karta hodnika trećeg kata zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku snimljena RTAB-Map SLAM algoritmom uz korištenje dubinske kamere i lidara kao senzora za prikupljanje podataka potrebnih za izgradnju karte okoline.

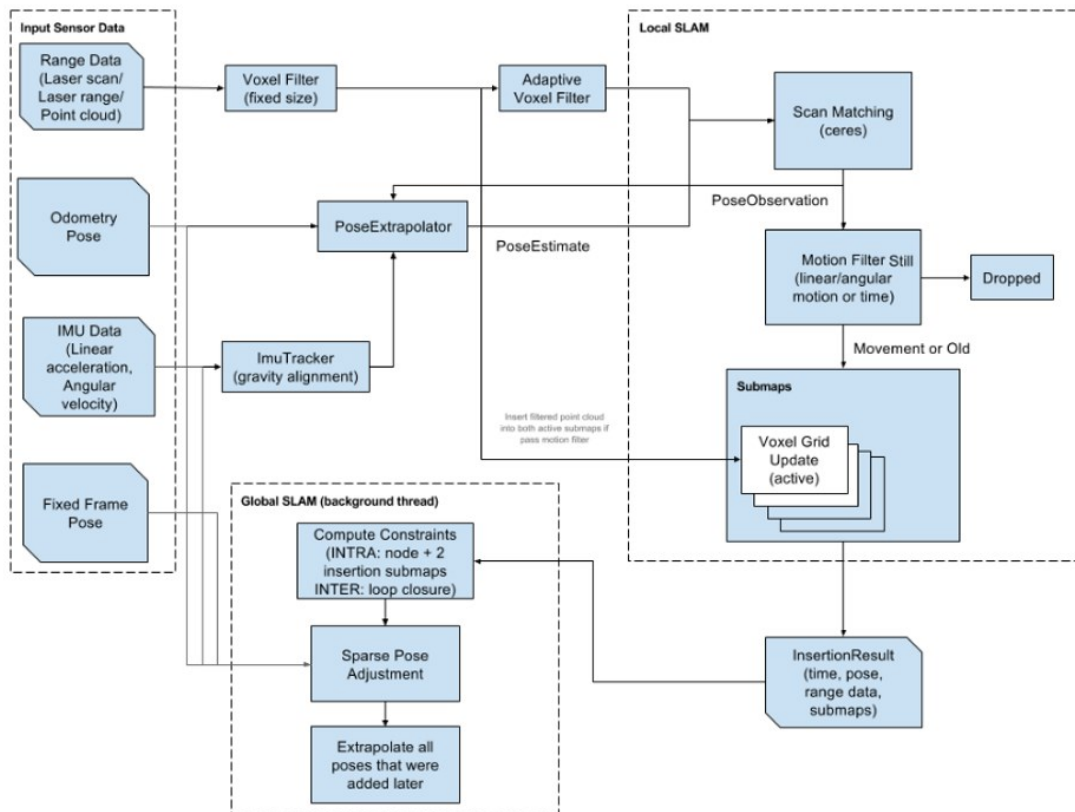


Slika 17. Karta hodnika izgrađena RTAB-Map algoritmom korištenjem dubinske kamere i lidara

#### **4.5. Cartographer**

Cartographer se može promatrati kao dva odvojena, ali povezana podsustava. Prvi je lokalni SLAM (ponekad se naziva i frontend ili lokalni graditelj putanje). Njegov je posao izgraditi niz podmapa. Svaka podmapa treba biti lokalno konzistentna, ali prihvatljivo je da se lokalni SLAM mijenja tijekom vremena. Drugi podsustav je globalni SLAM (ponekad se naziva pozadinski sustav). Radi u pozadinskim nitima, a glavni mu je zadatak pronaći ograničenja pri zatvaranju petlje. To radi uparivanjem skeniranja (sakupljenim u čvorovima) u odnosu na podmape. Uključuje i podatke s drugih senzora kako bi se dobio pogled na višu razinu i utvrdilo najdosljednije globalno rješenje. Jednostavnije rečeno, posao lokalnog SLAM-a je generirati dobre podmape, a posao globalnog SLAM-a je najdosljednije ih povezati. Na slici 18 prikazan je koncept Cartographer sustava. U daljnjem tekstu detaljnije su opisani najvažniji dijelovi i princip rada navedenog algoritma [13].





Slika 18. Koncept Cartographer sustava

Prema slici 18 vidljivo je da algoritmu treba predati ulazne podatke koji se dobiju s različitih senzora robota (npr. LIDAR, odometrija, IMU). Senzori za pronalaženje udaljenosti (na primjer LIDAR) pružaju informacije o dubini u više smjerova. Međutim, neka mjerenja nisu relevantna za SLAM. Ako je senzor djelomično prekriven prašinom, ili je usmjeren prema dijelu robota, dio izmjerene udaljenosti može se smatrati šumom za SLAM. S druge strane, neka od najudaljenijih mjerenja također mogu doći iz neželjenih izvora (refleksija, šum senzora) te nisu relevantna za SLAM. Kako bi se riješio taj problem, Cartographer počinje primjenom pojasno propusnog filtra i zadržava samo vrijednosti između određenog minimalnog i maksimalnog raspona. Te minimalne i maksimalne vrijednosti treba odabrati u skladu sa specifikacijama robota i senzora. Udaljenosti se mjere tijekom određenog vremenskog razdoblja, dok se robot kreće. Međutim, udaljenosti se isporučuju pomoću senzora „u paketu“ u velikim ROS porukama. Cartographer može neovisno razmotriti svaku vremensku oznaku poruka kako bi se uzele u obzir deformacije uzorkovane gibanjem robota. Što češće Cartographer dobiva mjerenja, to postaje bolji u obradi mjerenja kako bi sastavio jedan cjeloviti sken. Stoga je preporučljivo da se skeniranjem pruži što više podataka (ROS poruka) o udaljenosti (skup podataka o udaljenosti koji se mogu usporediti s drugim skeniranjima). Podaci o udaljenosti obično se mjere iz jedne točke na robotu

ali u više smjerova. To znači da su bliske površine (na primjer cesta ili pod) vrlo često pogođene i daju puno bodova. Nasuprot tome, udaljeni objekti se rjeđe pogađaju i nude manje bodova. Kako bi se smanjila računaska težina obrade podataka, obično moramo poduzorkovati oblake točaka. Međutim, jednostavno nasumično uzorkovanje uklonilo bi točke iz područja u kojima već imamo nisku gustoću mjerenja, a područja visoke gustoće i dalje bi imala više točaka nego što je potrebno. Kako bi se riješio taj problem gustoće, koristi se voxel filter koji pretvara uzorkovanje neobrađenih točaka u kocke konstantne veličine (voxele) i zadržava samo središte svake kocke. Mala veličina kocke rezultirat će gušćom reprezentacijom podataka, uzorkujući više izračuna. Velika veličina kocke, rezultirat će gubitkom podataka ali će njihova obrada biti puno brža. Nakon primjene voxel filtra fiksne veličine, Cartographer također primjenjuje prilagodljivi voxel filter. Ovaj filter pokušava odrediti optimalnu veličinu voxela (ispod maksimalne duljine) za postizanje ciljanog broja bodova. U 3D varijanti koriste se dva prilagodljiva voxel filtra za generiranje oblaka točaka visoke razlučivosti i niske razlučivosti. Inercijalna mjerna jedinica (engl. *Inertial Measurement Unit*) može biti koristan izvor informacija za SLAM jer pruža točan smjer gravitacije (tla) i šumovit ali dobar ukupni pokazatelj rotacije robota. Kako bi se filtrirao IMU šum, gravitacija se promatra tijekom određenog vremena. Ako se koristi 2D SLAM, podaci o udaljenosti mogu se obrađivati u stvarnom vremenu bez dodatnog izvora informacija tako da se može odabrati koristi li Cartographer IMU ili ne. Uz 3D SLAM, mora se osigurati IMU jer se koristi kao početna pretpostavka za orijentaciju skeniranja, uveliko smanjujući složenost uparivanja skeniranja.

Nakon što je skeniranje sastavljeno i filtrirano na temelju više raspona mjerenja, spremno je za lokalni SLAM algoritam. Lokalni SLAM algoritam stavlja novo skeniranje u svoju trenutnu konstrukciju podmape skeniranjem uparivanja pomoću početne procjene iz ekstrapolatora položaja. Ideja iza ekstrapolatora položaja je korištenje senzorskih podataka drugih senzora osim lidara da se predvidi gdje bi sljedeće skeniranje trebalo biti umetnuto u podmapu. Koriste se dvije strategije podudaranja skeniranja:

- CeresScanMatcher uzima početnu procijenu kao i prethodnu i pronalazi najbolje mjesto gdje se podudaranje skeniranja uklapa u podmapu. To čini interpolacijom podmape i podpikselskim poravnavanjem skeniranja. Ova metoda je brza, ali ne može popraviti pogreške koje su znatno veće od razlučivosti podmape. Ako su postavke senzora i vrijeme dobro podešeni, korištenje samo CeresScanMatchera obično je najbolji izbor.
- RealTimeCorrelativeScanMatcher se može omogućiti ako se ne koriste drugi senzori. Koristi pristup sličan načinu na koji se skeniranja uparuju s podmapama u zatvaranju petlje, ali se umjesto toga podudara s trenutnom podmapom. Najbolje podudaranje se tada koristi

kao prethodno za CeresScanMatcher. Ovakav način uparivanja skeniranja je vrlo skup i smatra se vjerodostojnijim od signala svih drugih senzora osim lidara, ali je robustan u okruženjima bogatim značajkama. U svakom slučaju CeresScanMatcher se može konfigurirati tako da svakom novom ulazu daje određenu težinu. Težina je mjera povjerenja u podatke, te se može promatrati kao statička kovarijanca. Što je veća težina izvora podataka, to će Cartographer staviti veći naglasak na taj izvor podataka prilikom uparivanja skeniranja. Izvor podataka uključuje zauzeti prostor (točke sa skeniranja), translaciju i rotaciju iz ekstrapolatora poze (ili RealTimeCorrelativeScanMatcher).

CeresScanMatcher je dobio ime po Ceres Solveru, biblioteci razvijenoj u Google-u za rješavanje nelinearnih problema najmanjih kvadrata. Problem uparivanja skeniranja modelira se tako da se nelinearni problem najmanjih kvadrata svede na najmanju moguću mjeru, pri čemu je pomak (transformacijska matrica) između dva skeniranja parametar koji se treba odrediti. Ceres optimizira pomak koristeći algoritam spuštanja za zadani broj iteracija. Ceres se može konfigurirati da prilagodi brzinu konvergencije prema vlastitim potrebama.

RealTimeCorrelativeScanMatcher može se mijenjati ovisno o senzoru koji se koristi. Radi tako što traži slične skenirane slike u prozoru za pretraživanje koji je definiran maksimalnim radijusom udaljenosti i maksimalnim radijusom kuta. Prilikom izvođenja uparivanja skeniranja sa skeniranjima koji se nalaze u ovom prozoru, moguće je odabrati drugačije težine za translacijske i rotacijske komponente.

Kako bi se izbjeglo umetanje previše skeniranja po podmapama, nakon što uparivač skeniranja detektira pomak između dva skeniranja, ono prolazi kroz filter pokreta. Skeniranje se odbacuje ako se pomak koji je doveo do njega ne smatra dovoljno značajnim. Skeniranje se ubacuje u trenutnu podmapu samo ako je njegov pomak iznad određene udaljenosti, kuta ili vremenskog praga. Podmapa se smatra završenom kada lokalni SLAM primi zadanu količinu podataka o pomaku. Pogreška lokalnog SLAM-a se akumulira tijekom vremena, globalni SLAM se koristi za popravljavanje te pogreške. Podmape moraju biti dovoljno male kako bi pomak robota unutar njih bio ispod rezolucije, tako da su lokalno ispravne. S druge strane, trebale bi biti dovoljno velike da se međusobno razlikuju kako bi zatvaranje petlje ispravno radilo. Podmape mogu pohraniti svoje podatke o rasponu u nekoliko različitih struktura podataka. Najrašireniji prikaz naziva se mreža vjerojatnosti. Međutim u 2D, također se može odabrati korištenje ograničena polja udaljenosti s predznakom (engl. *Truncated Signed Distance Fields (TSDF)*). Mreža vjerojatnosti dijeli prostor u 2D ili 3D tablicu gdje svaka ćelija ima fiksnu veličinu i sadrži izgled da bude

zanemarena. Koeficijenti se ažuriraju prema „pogocima“ (gdje se mjere podaci o udaljenosti robota i prepreke) i „promašajima“ (slobodni prostor između senzora i izmjerenih točaka). I pogoci i promašaji mogu imati različitu težinu u izračunima vjerojatnosti zauzetosti dajući više ili manje povjerenja mjerenjima zauzetog i slobodnog prostora.

Dok lokalni SLAM generira svoj niz podmapa, zadatak globalne optimizacije (koji se obično naziva „problem optimizacije“) izvodi se u pozadini. Njegova je uloga da prerasporedi podmape tako da tvore konzistentnu globalnu kartu. Na primjer, ova optimizacija je zadužena za izmjenu trenutno izgrađene putanje kako bi se pravilno poravnale podmape s obzirom na zatvaranje petlje. Optimizacija se izvodi u serijama nakon što je umetnut određeni broj čvorova putanje. Globalni SLAM je vrsta „graf SLAM-a“. To je u biti optimizacija pozadinskog grafa koja djeluje izgradnjom ograničenja između čvorova i podmapa, a zatim optimiziranje rezultirajućeg grafa ograničenja. Ograničenja se intuitivno mogu smatrati kao „mala užad“ koja povezuje sve čvorove. Problem optimizacije povezuje tu užad međusobno. Rezultirajuća mreža naziva se „graf položaja“. Neglobalna ograničenja (također poznata kao ograničenja unutar podmapa) izgrađuju se automatski između čvorova koji usko slijede jedan drugog na putanji. Intuitivno, ta „neglobalna užad“ zadržavaju lokalnu strukturu putanje konzistentnom. Globalna ograničenja (nazivaju se i ograničenja zatvaranja petlje) redovito se pretražuju između nove podmape i prethodnih čvorova koji se smatraju „dovoljno blizu“ u prostoru (dio određenog prozora za pretraživanje) i jakog uklapanja (dobro podudaranje kada se izvodi skeniranje podudaranja). Intuitivno, ta „globalna užad“ uvode čvorove u strukturu i čvrsto približavaju dvije niti. Kako bi se ograničila količina ograničenja (i izračuna), Cartographer razmatra samo poduzorkovani skup svih bliskih čvorova za izgradnju ograničenja. To se kontrolira konstantom omjera uzorkovanja. Uzorkovanje premalo čvorova moglo bi rezultirati promašenim ograničenjima i neučinkovitim zatvaranjem petlje. Uzorkovanje previše čvorova usporilo bi globalni SLAM i spriječilo zatvaranje petlje u stvarnom vremenu. Kada se čvor i podmapa razmatraju za izgradnju ograničenja, oni prolaze kroz prvo uparivanje skeniranja pod nazivom FastCorrelativeScanMatcher. Ovaj način usklađivanja skeniranja posebno je dizajniran za Cartographer i omogućuje usklađivanje skeniranja zatvaranja petlji u stvarnom vremenu. FastCorrelativeScanMatcher se oslanja na mehanizam „Branch and bound“ kako bi radio na različitim rezolucijama mreže i učinkovito eliminirao netočna podudaranja. Radi na stablu istraživanja čiju se dubinu može kontrolirati. Nakon što FastCorrelativeScanMatcher ima dovoljno dobar prijedlog (iznad minimalnog rezultata podudaranja), onda se ubacuje u CeresScanMatcher kako bi se poboljšao položaj.

Kada Cartographer pokrene problem optimizacije, Ceres se koristi za preuređivanje podmapa prema višestrukim rezidualima. Reziduali se računaju korištenjem vrednovanih težinskih funkcija (engl. *Cost functions*). Globalna optimizacija ima težinske funkcije kako bi se uzelo u obzir mnoštvo izvora podataka: globalna ograničenja (zatvaranje petlje), neglobalna (podudarna) ograničenja, mjerenja ubrzanja i rotacije IMU-a, lokalne procjene grube pozicije, odometriju robota. Težine i Ceres opcije mogu se konfigurirati kako bi se poboljšao SLAM. Kao dio IMU reziduala problem optimizacije daje određenu fleksibilnost IMU položaja. Ako IMU nije dovoljno pouzdan, rezultati globalne optimizacije Cerasa mogu se zabilježiti i koristiti za poboljšanja ekstrinzične kalibracije. Ako Ceras ne optimizira IMU položaj ispravno, a ekstrinzična kalibracija je dovoljno dobra, ta vrijednost se može postaviti da bude konstantna. Nakon što je putanja gotova, Cartographer pokreće novu globalnu optimizaciju s obično puno više iteracija od prethodnih globalnih optimizacija. To je učinjeno kako bi se poboljšao konačni rezultat Cartographera i obično ne mora biti u stvarnom vremenu pa je veliki broj iteracije često pravi izbor.

#### 4.5.1. Cartographer lidar

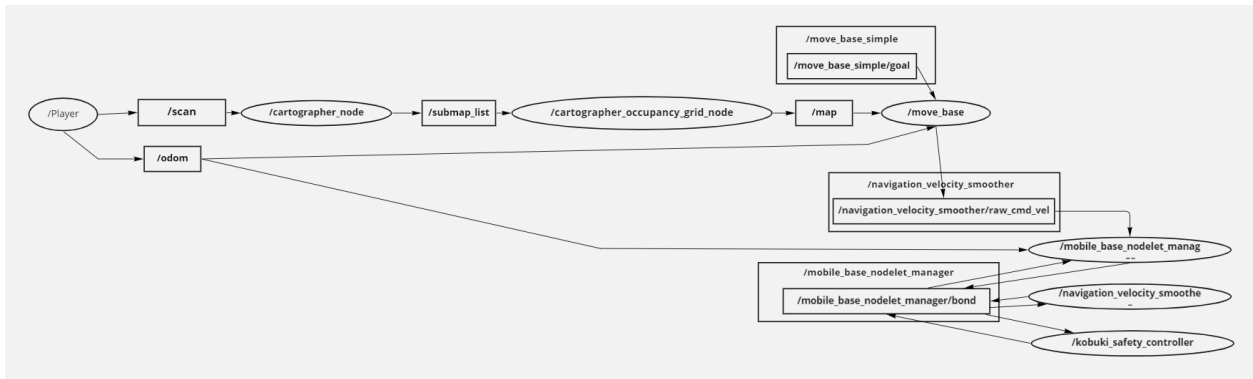
Za izgradnju karte odnosno mapiranje prostora uz pomoć Cartographer SLAM algoritma. Navedeni ROS čvor je potrebno pretplatiti na određene teme koje sadrže podatke koji su potrebni za SLAM algoritam. Lista relevantnih pretplaćenih/objavljenih tema od Cartographer ROS čvora prikazana je u tablici 11.

Tablica 11. Popis relevantnih pretplaćenih/objavljenih tema od Cartographer ROS paketa

Topic	Message	Type
Scan	Sensor_msgs/LaserScan	Subscribe
Echoes	Sensor_msgs/MultiEchoLaserScan	Subscribe
Points2	Sensor_msgs/PointCloud2	Subscribe
Odom	Nav_msgs/Odometry	Subscribe
Imu	Sensor_msgs/Imu	Subscribe
Scan_matched_points2	Sensor_msgs/PointCloud2	Published
Submap_list	Cartographer_ros_msgs/SubmapList	Published
Tracked_pose	Geometry_msgs/PoseStamped	Published

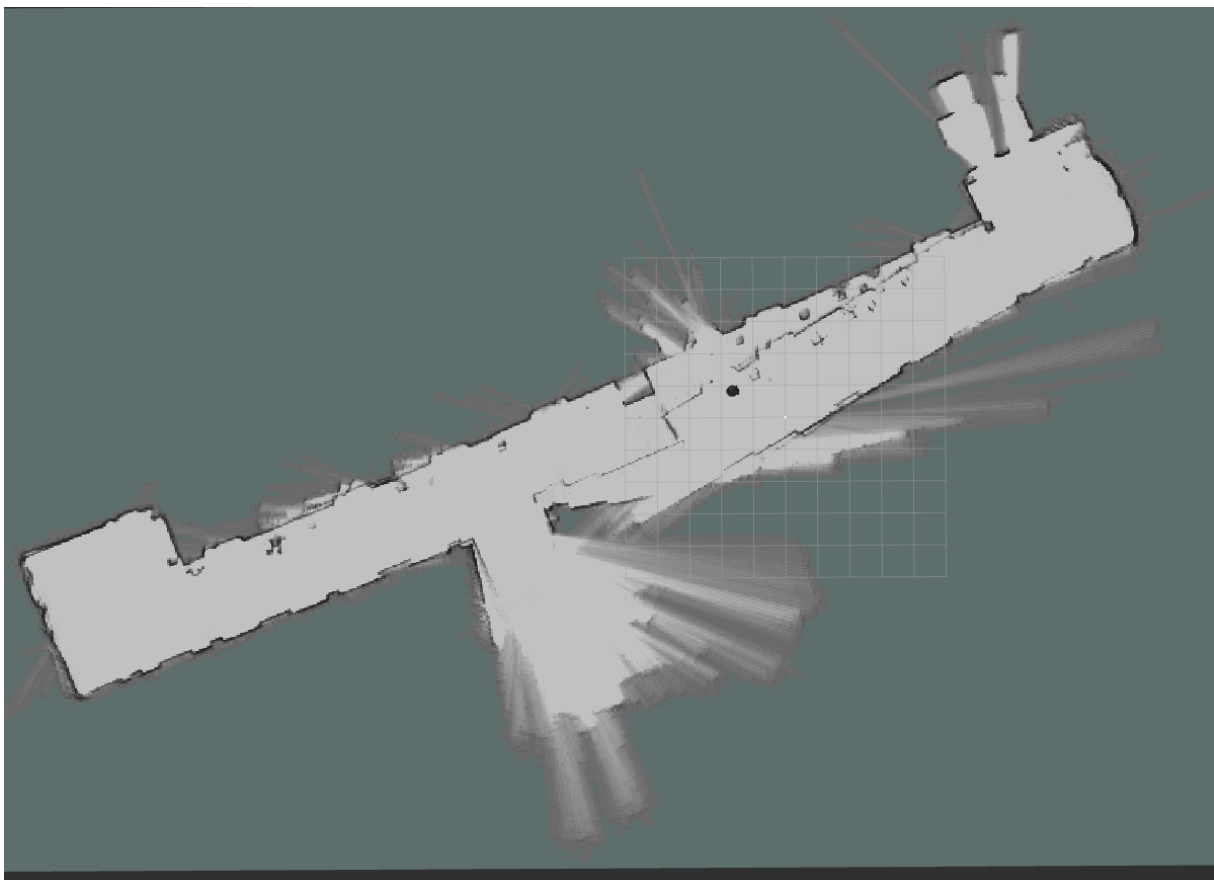
Tema Imu se može koristiti ali i ne mora u ovisnosti o tome koristi li se 2D ili 3D mapiranje. Ukoliko se koristi 2D mapiranje, tada se ne mora koristiti, a u suprotnom ako se koristi 3D

mapiranje tada se mora koristiti. Također tema `Tracked_pose` se ne mora objavljivati ali i može. Parametri algoritma prikazani su u prilogu 4.5.1.1.



Slika 19. Rqt graf Cartographer algoritma s korištenim lidarom

Na slici 19 prikazan je rqt graf Cartographer algoritma uz korištenje lidara kao senzora za prikupljanje podataka potrebnih za mapiranje prostora. Iz grafa je vidljivo na koje se sve teme Cartographer čvor pretplaćuje i koje teme objavljuje.

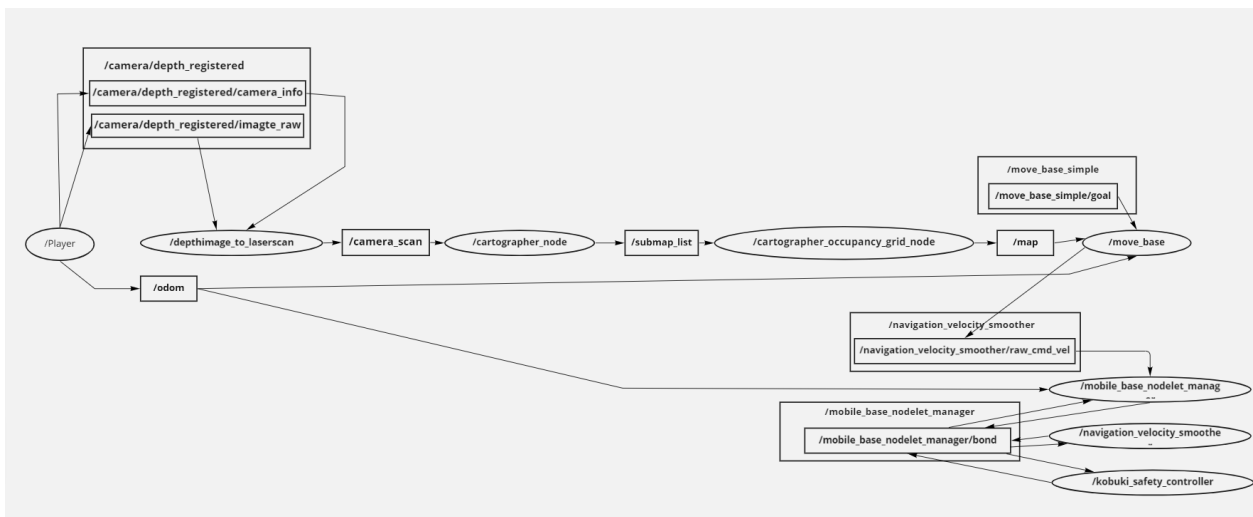


Slika 20. Karta hodnika izgrađena Cartographer algoritmom korištenjem lidara

Na slici 20 prikazana je karta hodnika trećeg kata zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku snimljena sa Cartographer SLAM algoritmom uz korištenje lidara kao senzora za prikupljanje podataka potrebnih za mapiranje prostora.

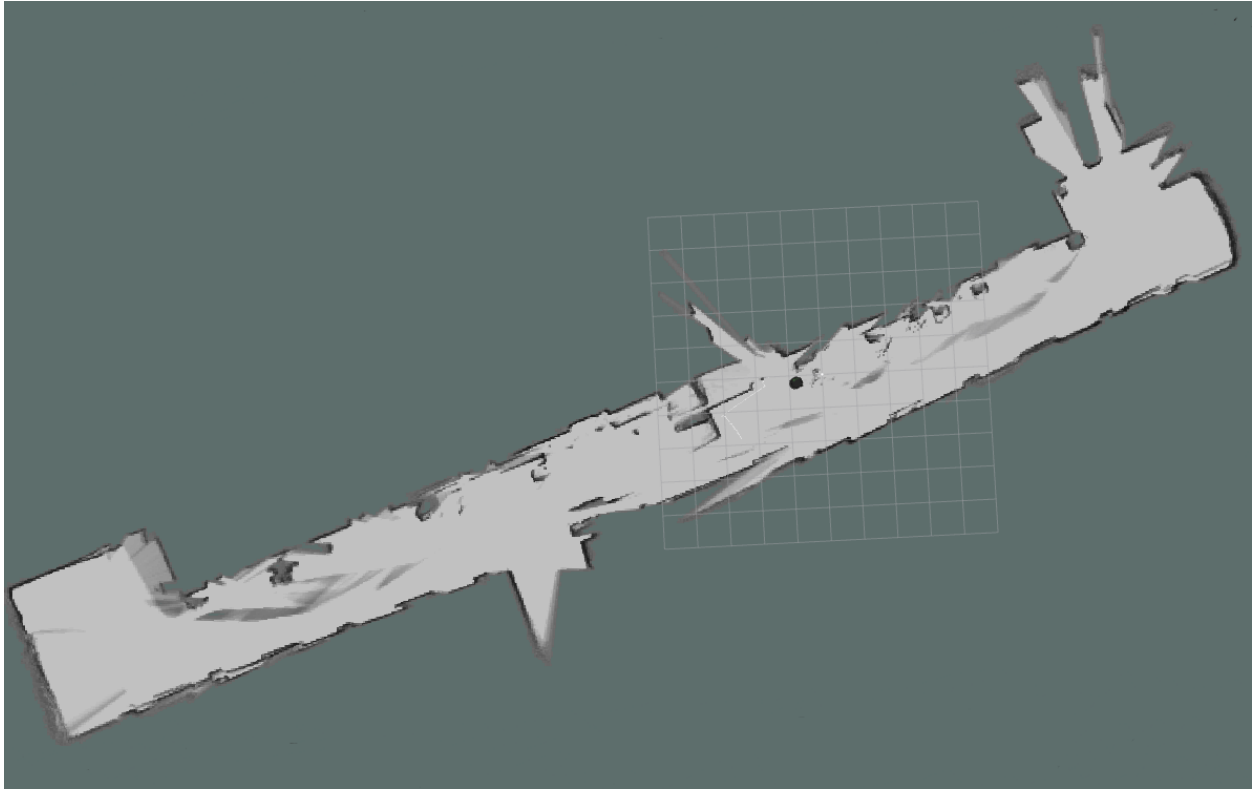
#### 4.5.2. Cartographer dubinska kamera

Isto kao i GMapper SLAM algoritam, Cartographer je prvenstveno namijenjen za mapiranje prostora pomoću lidar senzora. Kako bi se mogla koristiti dubinska kamera kao senzor za mapiranje prostora potrebno je koristiti ROS čvor `depthimage_to_laserscan` kako bi se dubinska slika pretvorila u laser i na taj način omogućilo mapiranje prostora pomoću dubinske kamere. Lista relevantnih pretplaćenih/objavljenih tema od Cartographer ROS paketa prikazana je u tablici 11. Na slici 21 prikazan je rqt graf Cartographer algoritma uz korištenje dubinske kamere kao senzora za prikupljanje podataka potrebnih za mapiranje prostora. Parametri algoritma prikazani su u prilogu 4.5.2.1.



Slika 21. Rqt graf Cartographer algoritma s korištenom dubinskom kamerom

Na slici 22 prikazana je karta hodnika trećeg kata zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku snimljena sa Cartographer SLAM algoritmom uz korištenje dubinske kamere kao senzora za prikupljanje podataka potrebnih za mapiranje prostora.



Slika 22. Karta hodnika izgrađena Cartographer algoritmom korištenjem dubinske kamere

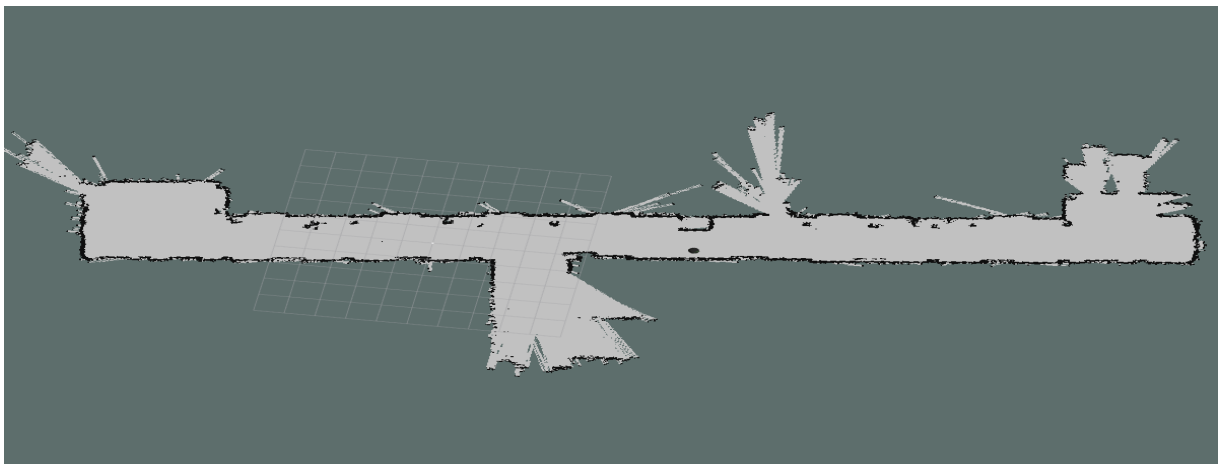


## 5. USPOREDBA METODA

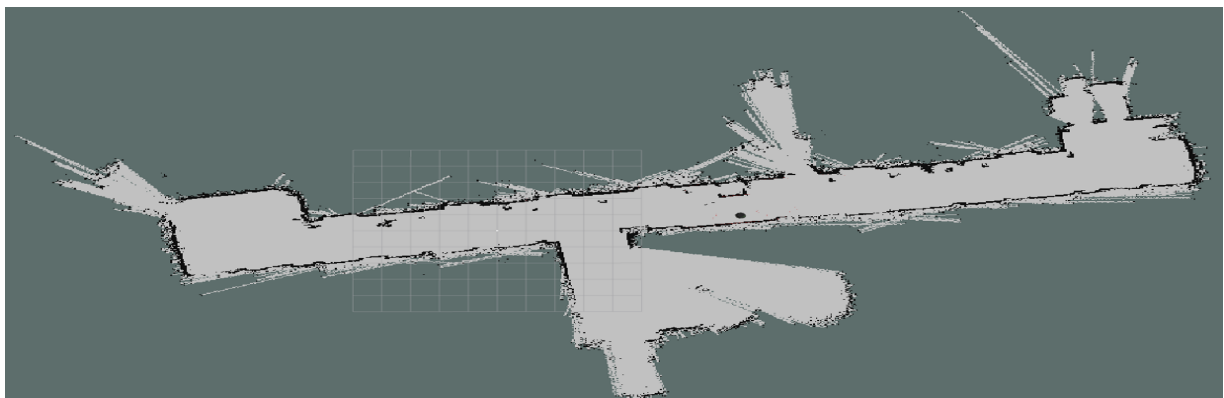
Kako bi se uspješno usporedile različite metode za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima, mobilni robot Turtlebot2 provozan je hodnikom zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Mobilni robot provozan je od sredine hodnika do jednog kraja, zatim do drugog kraja te na kraju vraćen u poziciju iz koje je krenuo. Prilikom kretanja robota snimana je rosbag datoteka koja sadrži sve potrebne informacije za uspješno stvaranje karti unutrašnjeg prostora. Na taj način postignuto je to da su sve metode ispitane u jednakim uvjetima te su vidljive prednosti i nedostaci određenih metoda. Metode koje su korištene u ovom diplomskom radu su navedene i detaljno opisane u prijašnjem poglavlju.

### 5.1. Mapiranje prostora lidar sensorom

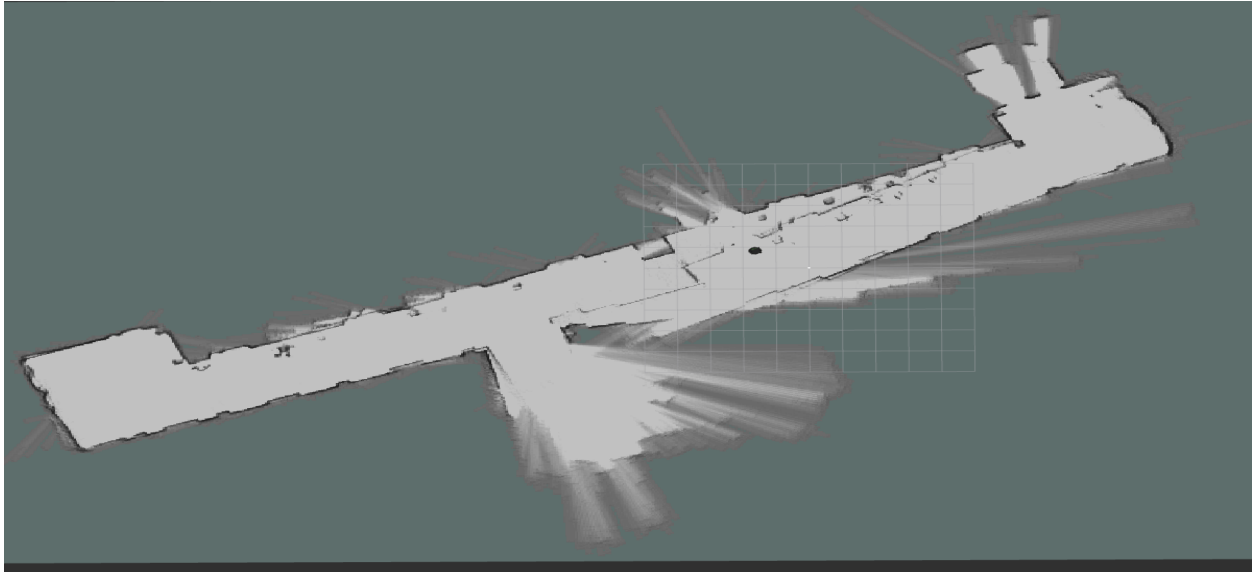
U ovom potpoglavlju prikazane su karte izgrađene različitim metodama za izgradnju karte okoline uz korištenje lidar senzora.



Slika 23. Karta hodnika izgrađena GMapping algoritmom uz korištenje lidara



Slika 24. Karta hodnika izgrađena RTAB-Map algoritmom uz korištenje lidara

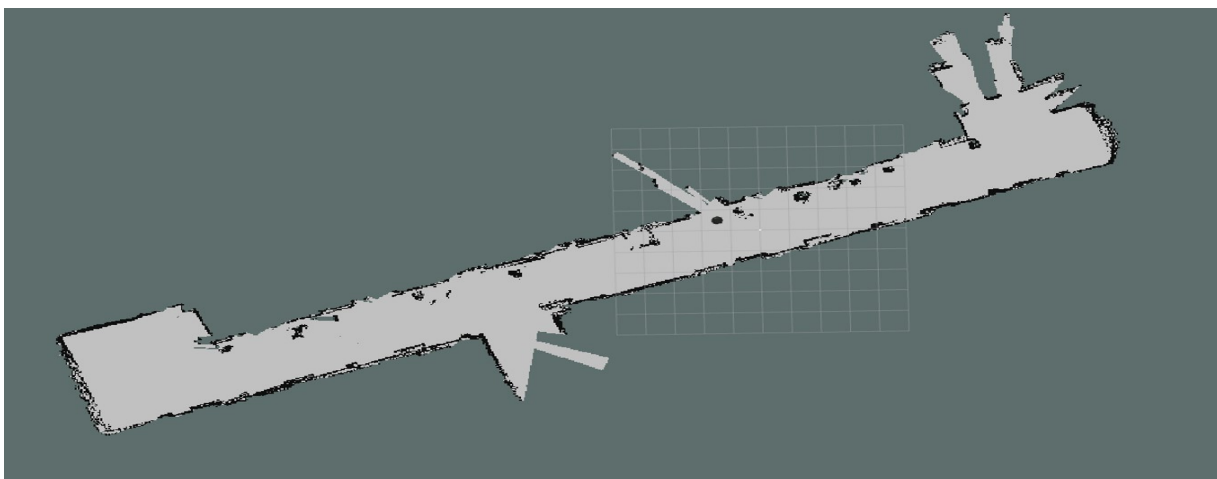


Slika 25. Karta hodnika izgrađena Cartographer algoritmom uz korištenje lidara

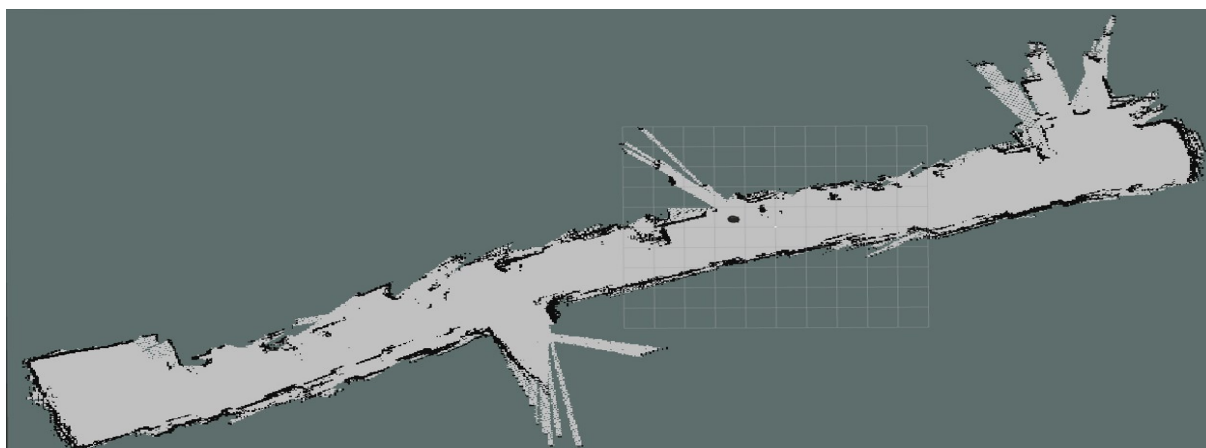
Nakon provedenog mapiranja prostora različitim metodama. Iz gore prikazanih slika vidljivo je da je GMapping algoritam uspio napraviti najtočniju kartu uz korištenje lidar senzora. Iz slika je vidljivo da je po preciznosti GMapping algoritam najprecizniji. Zidovi su najbolje iscrtani i najravniji su kao što bi i trebali biti. Sljedeći po preciznosti je RTAB-Map algoritam kod kojeg su zidovi isto poprilično dobro označeni dok ipak ima više smetnji i zakrivljenosti za razliku od GMapping algoritma. Najlošije rezultate mapiranja prostora korištenjem lidar senzora pokazao je Cartographer algoritam. Iz slike karte koju je napravio Cartographer algoritam vidljivo je da u zadnjem dijelu izgradnje karte nije uspjelo doći do zatvaranje petlje kako bi se karta ispravila. Da je algoritam uspio odraditi zatvaranje petlje i ispraviti kartu vjerojatno bi postigao bolji rezultat od RTAB-Map algoritma. RTAB-Map i Cartographer posjeduju više parametara od GMapper algoritma koje je moguće korigirati kako bi se postigli što bolji rezultati mapiranja prostora. Tako da bi se postavljanjem točnijih parametara ipak moglo utjecati na poboljšanje rezultata izgradnje karte.

## 5.2. Mapiranje prostora dubinskom kamerom

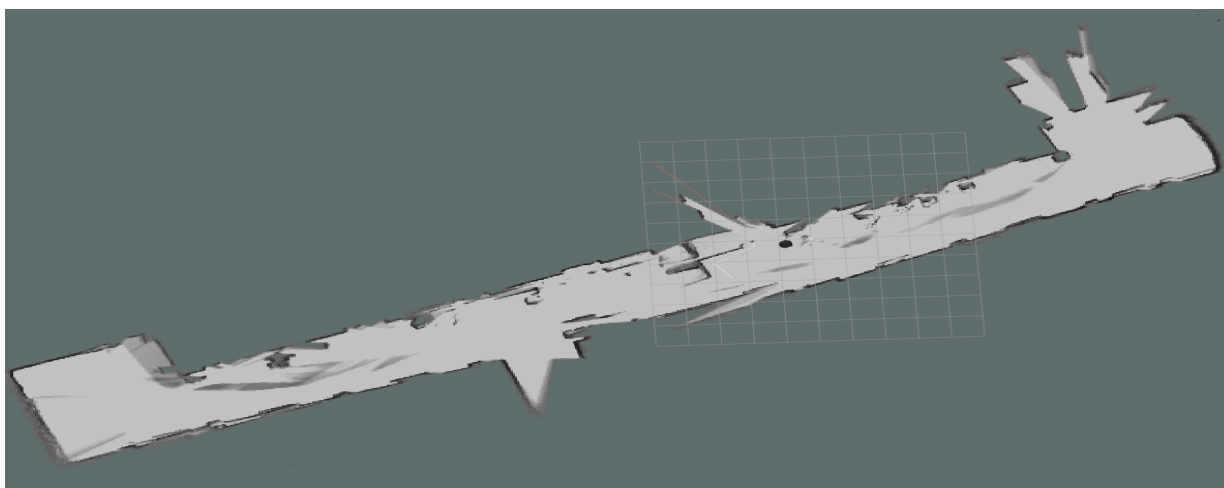
U ovom potpoglavlju prikazane su karte izgrađene različitim metodama za izgradnju karte okoline uz korištenje dubinske kamere. Te su uspoređene karte po preciznosti i cjelovitosti.



Slika 26. Karta hodnika izgrađena GMapping algoritmom uz korištenje dubinske kamere



Slika 27. Karta hodnika izgrađena RTAB-Map algoritmom uz korištenje dubinske kamere



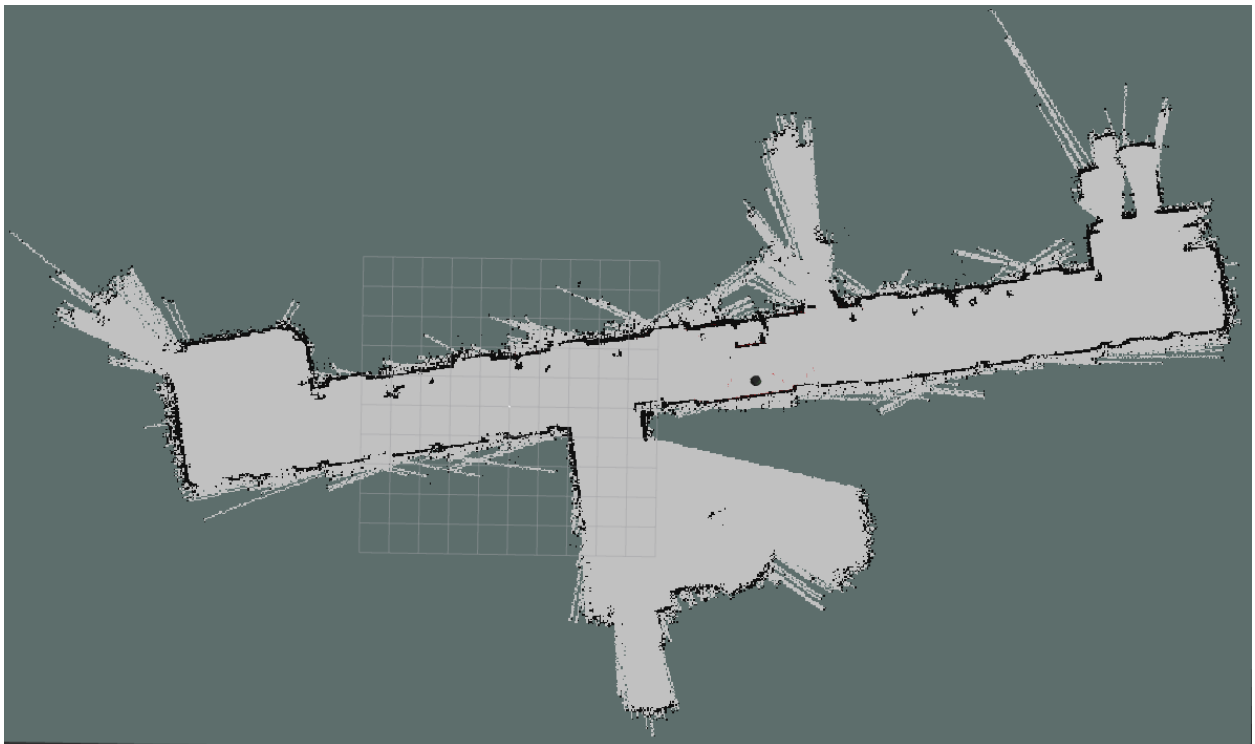
Slika 28. Karta hodnika izgrađena Cartographer algoritmom uz korištenje dubinske kamere

Nakon provedenog mapiranja prostora različitim metodama. Iz gore prikazanih slika vidljivo je da je najbolje rezultate opet postigao GMapper algoritam. Korištenjem navedenog algoritma

dobivena je najtočnije i najpreciznija karta za razliku od karata dobivenih sa ostale dvije metode. Iz slike je vidljivo da su zidovi najravniji, karta je najcjelovitija i ima najmanje šumova na karti. Drugi po točnosti karte je Cartographer algoritam kod kojeg zidovi izgledaju poprilično ravno, iako opet u zadnjem dijelu snimanja prostora algoritam nije uspio zatvoriti petlju i ispraviti kartu. Da je došlo do zatvaranje petlje dobili bi se bolji rezultati od GMapper algoritma. Najlošije rezultate u izgradnji karte pomoću dubinske kamere pokazao je RTAB-Map algoritam. Iz slike je vidljivo da je karta dosta izobličena. Problem kod mapiranja prostora sa dubinsko kamerom je to što robot može mapirati prostor samo u rasponu kamere. Također kod ovih metoda mapiranja problem isto stvara brzina kretanja robota i brzina rotacije. Da se robot kretao sporije i da se kretao bliže zidovima dobili bi se bolji rezultati.

### 5.3. Mapiranje prostora kombinacijom lidara i dubinske kamere

Pošto jedino RTAB-Map od ovih metoda može koristiti kombinaciju lidara i dubinske kamere u ovom potpoglavlju će biti prikazana samo njegova karta, a biti će uspoređena sa svim ostalim kartama.



Slika 29. Karta hodnika izgrađena RTAB-Map algoritmom uz korištenje dubinske kamere i lidara

Iz slike karte koja je napravljena kombinacijom lidara i dubinske kamere vidljivo je da je karta prostora cijelovita. Po preciznosti ipak nije bolja od karte koja se dobila s GMapper algoritmom

uz korištenje lidara. Dok je ipak točnija i preciznija od svih ostalih karata koje su dobivene različitim metodama. Teoretski karta prostora dobivena kombinacijom lidara i dubinske kamere bi trebala biti najtočnija i najpreciznija u usporedbi sa ostalim kartama. Preciznost i točnost karte bi se moglo popraviti sa boljim odabirom parametara i njihovim vrijednostima.

## 6. ZAKLJUČAK

Zadatak ovog diplomskog rada bio je izraditi, implementirati te usporediti različite algoritme za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima korištenjem programskih alata raspoloživih u ROS-u. Sukladno algoritmima koji se koriste trebalo je upotrijebiti različite senzore koje razmatrane metode podržavaju. Senzori koji su razmatrani u ovom diplomskom radu su lidar i dubinska kamera te njihova kombinacija. Za rješavanje ovog diplomskog rada korišten je mobilni robot Turtlebot2. Mapiranje prostora izvršeno je u hodniku zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Da bi se postigli jednaki uvjeti za sve metode, mobilni robot Turtlebot2 provozan je od sredine hodnika do jednog kraja, zatim do drugog kraja i na kraju vraćen u početnu točku snimanja. Tijekom kretanja robota po hodniku snimana je rosbag datoteka koja sadrži sve potrebne informacije za uspješno izvođenje algoritama mapiranja. Na taj način postignuti su jednaki uvjeti za sve metode. Metode koje su se koristile u ovom radu su GMapper, RTAB-Map i Cartographer. Sve metode mogu koristiti lidar kao senzor za prikupljanje podataka, dok dubinsku kameru može koristiti RTAB-Map ali je dubinsku kameru moguće koristiti i u metodama GMapper i Cartographer na način da se koristi dodatni ROS čvor `depthimage_to_laserscan` koji pretvara dubinsku sliku u laser i na taj način omogućava korištenje dubinske kamere i u metodama koje su namijenjene samo za mapiranje prostora pomoću lasera. Od svih navedenih algoritama mogućnost korištenja kombinacije senzora ima jedino metoda RTAB-Map. Nakon provedenih testiranja i mapiranja prostora. Iz rezultata je vidljivo da je najbolje rezultate postigao GMapper algoritam korištenjem lidar senzora. Karta snimljena tom metodom je najcjelovitija. Po preciznosti je isto pokazala najbolje rezultate, zidovi na karti su dobro označeni i ravni kao što bi i trebali biti, nema izobličenja kao u ostalim metodama. Druga metoda po rezultatima je RTAB-Map uz korištenje lidara. Karta snimljena tom metodom po preciznosti je dala dobre rezultate, zidovi su ravni isto kao i kod GMapper metode, ali po cjelovitosti izgrađene karte pokazala je ipak malo lošije rezultate u odnosu na GMapper metodu. U nekim dijelovima karte zidovi nisu dobro označeni. Najlošije rezultate mapiranja prostora lidarom pokazala je metoda Cartographer. Karta dobivena Cartographer metodom u pogledu cjelovitosti izrađene karte pokazala rezultate iste kao i GMapper, ali po točnosti je ipak pokazala lošije rezultate od ostalih metoda. Jedan od razloga zašto je metoda pokazala lošije rezultate je taj što pri kraju snimanja prostora odnosno, vraćanja robota u početnu točku, metoda nije uspjela zatvoriti petlju i ispraviti kartu koju je stvorila. Da se uspjelo izvršiti zatvaranje petlje metoda bi postigla puno bolje rezultate. Cartographer metoda zahtjeva više prolazaka kroz isto mjesto mapiranja kako bi što bolje uspjela napraviti kartu prostora. Na taj način bi se moglo značajno

utjecati na kvalitetu dobivene karte prostora. Metode RTAB-Map i Cartographer posjeduju puno više parametara koje je moguće podešavati kako bi se postigao što bolji rad algoritma i njegovi rezultati, tako da je moguće još poboljšati rezultate razmatranih metoda izborom odgovarajućih parametara potrebnih za njihov rad.

Kod mapiranja prostora dubinskom kamerom najbolje rezultate je opet pokazala metoda GMapper. Dok je najlošiju kartu napravila metoda RTAB-Map. Karta dobivena metodom GMapper je najcjelovitija i najpreciznija. Zidovi su dobro označeni i ravni. Dobre rezultate je pokazala i metoda Cartographer, ali opet u zadnjem dijelu mapiranja prostora nije došlo do zatvaranja petlje i ispravka karte, što bi puno popravilo pa čak i nadmašilo rezultate od GMapper metode. U mapiranju prostora s dubinskom kamerom najlošije rezultate pokazala je metoda RTAB-Map. Na slici dobivene karte navedenom metodom vidljivo je da karta nije cjelovita. Što se tiče preciznosti pokazala je dosta loše rezultate, zidovi nisu ravni već su izobličeni. Problem mapiranja prostora korištenjem dubinske kamere je taj što za mapiranje je potrebno puno više vremena. Zato što je prostor moguće mapirati samo u određenom rasponu kamere. Tijekom mapiranja robot bi se trebao kretati manjom brzinom i imati što manje rotacijskih kretnji. Tijekom kretanja robota hodnikom, robot se uglavnom kretao sredinom hodnika, dok se približavao zidovima jedino kada je zaobilazio prepreke na putu. Postigli bi se bolji rezultati da se robot kretao bliže zidovima tijekom snimanja rosbag datoteke. Također bi se postigli bolji rezultati da se prolazilo više puta kroz hodnik. Isto tako boljim odabirom parametara algoritama bi se moglo utjecati na kvalitetu mapiranja prostora.

Pošto je jedina metoda koja može koristiti kombinaciju lidara i dubinske kamere je RTAB-Map metoda, ta metoda će biti uspoređena sa svim ostalim metodama. Iz slike karte koja je dobivena tom metodom vidljivo je da ipak nije postigla najbolje rezultate. Što se tiče cjelovitosti karte bolje rezultate je pokazala metoda GMapper s korištenim lidarom. Po preciznosti karte metoda je pokazala zadovoljavajuće rezultate, zidovi su ravni. Teoretski ova metoda je trebala pokazati najbolje rezultate mapiranja prostora, zato što koristi kombinaciju lidara i dubinske kamere. S tom kombinacijom senzora metoda bi trebala dobiti najdetaljniju kartu prostora, zato što lidar skenira prostor u rasponu od 360 stupnjeva, dok dubinska kamera može označavati prepreka koje se nalaze iznad i ispod razine na kojoj se nalazi lidar, što je i problem mapiranja prostora sa 2D lidarom. Ukoliko se neka prepreka nalazi ispod ili iznad razine lidara ona neće biti označena na karti kao prepreka, što stvara probleme prilikom kretanja robota mapiranim prostorom. Zato se i koristi kombinacija lidara i dubinske kamere kako bi se dobili što točniji i precizniji rezultati izgrađene karte okoline. Ova metoda bi se mogla poboljšati sa boljim izborom

parametara algoritma. Također bi se dobili bolji rezultati mapiranja prostora ukoliko bi se robot kretao više puta po istom prostoru.

U ovom diplomskom radu nakon provedenih testiranja najbolje rezultate mapiranja prostora pokazala je metoda GMapper sa korištenjem lidara kao senzora za mapiranje. Ostale metode bi se mogle popraviti, odnosno njihovi rezultati mapiranja bi se mogli poboljšati boljim izborom parametara algoritama. Također boljim kretanjem robota kroz unutrašnji prostor koji se mapira mogli bi se poboljšati rezultati dobivenih algoritama.



## LITERATURA

- [1] Doc.dr.sc. Ivan Petrović, Mobilna robotika – predavanje, 2004./2005. Zagreb
- [2] Turtlebot 2, dostupno na : <https://www.turtlebot.com/turtlebot2/>, travanj 2022.
- [3] Lidar, dostupno na : <https://www.slamtec.com/en/Lidar/A2>, travanj 2022.
- [4] Astra Stereo S, Orbbec, dostupno na: <https://orbbec3d.com/astrastereos/> , travanj 2022.
- [5] Robot Operating System, Wikipedia, dostupno na: [https://en.wikipedia.org/wiki/Robot\\_Operating\\_System](https://en.wikipedia.org/wiki/Robot_Operating_System) , travanj 2022.
- [6] RVIZ, dostupno na: <http://wiki.ros.org/rviz> , travanj 2022.
- [7] GAZEBO, dostupno na : <https://gazebo.org/home> , travanj 2022.
- [8] R. Mata, Persistent Autonomous Exploration, Mapping and Localization, dostupno na : <https://dspace.mit.edu/handle/1721.1/113127>, svibanj 2022.
- [9] Dynamic Window Approach, dostupno na : [http://wiki.ros.org/dwa\\_local\\_planner](http://wiki.ros.org/dwa_local_planner), svibanj 2022.
- [10] Bruno M. F. da Silva, Rodrigo S. Xavier, Luiz M. G. Goncalves, Mapping and Navigation for Indoor Robots under ros: An Experimental Analysis, dostupno na: [Mapping and Navigation for Indoor Robots under ROS: An Experimental Analysis\[v1\] | Preprints](#), svibanj 2022.
- [11] M.L, F.M, RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation, dostupno na : [RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation - Labbé - 2019 - Journal of Field Robotics - Wiley Online Library](#), svibanj 2022.
- [12] RTAB-Map, dostupno na: [http://wiki.ros.org/rtabmap\\_ros/Tutorials/SetupOnYourRobot](http://wiki.ros.org/rtabmap_ros/Tutorials/SetupOnYourRobot), svibanj 2022
- [13] Cartographer, dostupno na: <https://google-cartographer-ros.readthedocs.io/en/latest/>, svibanj 2022.

## SAŽETAK

Zadatak ovog diplomskog rada je usporediti različite metode za izgradnju karte i navigaciju mobilnog robota u unutrašnjim prostorima. Metode koje su implementirane i uspoređene su GMapping, RTAB-Map i Cartographer. Za rješavanje problema diplomskog rada korišten je mobilni robot pod nazivom Turtlebot2. Koji sadrži senzore lidar i dubinska kamera, pomoću kojih su izgrađene karte hodnika trećeg kata zgrade Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Karte su izgrađene različitim SLAM algoritmima uz korištenje lidara, dubinske kamere te kombinacije lidara i dubinske kamere. Mobilni robot provozan je od središta hodnika do jednog kraja hodnika, zatim do drugog kraja hodnika i vraćen u početnu točku kretanja. Tijekom kretanja mobilnog robota hodnikom snimljena je rosbag datoteka koja sadrži sve potrebne podatke za mapiranje prostora. Tako je omogućeno implementiranje navedenih SLAM algoritama uz korištenje istih podataka, kako bi se dobila što bolja usporedba različitih metoda. Izgrađene karte uspoređene su po preciznosti, točnosti i cjelovitosti. Nakon provedenih testiranja najbolje rezultate mapiranja prostora pokazala je metoda GMapper s korištenjem lidara kao senzora za mapiranje, dok su ostale metode pokazale lošije rezultate.

**Ključne riječi:** Autonomni mobilni robot, dubinska kamera, lidar, SLAM, Turtlebot2

## ABSTRACT

The task of this master thesis is to compare different methods for building a map and navigating a mobile robot in indoor spaces. Implemented and compared methods are GMapping, RTAB-Map and Cartographer. Mobile robot called Turtlebot2 which contains sensors lidar and depth camera are used to solve the problem of this master thesis. The constructed maps of third floor hallway of the building Faculty of electrical engineering, computing and information technology in Osijek. The maps were built by different SLAM algorithms using lidar, depth camera and combination of lidar and depth camera. Mobile robot was driven from the center of hallway to one end of hallway, then to other end of hallway and return to the starting point of movement. While the mobile robot was moving down the hall, a rosbag file was recorded. The rosbag file contains all the necessary data for mapping the space. On that way it is possible to implement these SLAM algorithms using the same data, in order to obtain the best possible comparison of different methods. The constructed maps were compared in terms of precision, accuracy and completeness. After testing, the best results were shown by the GMapper method using lidar as a mapping sensor, while other methods showed poorer results.

**Keywords:** Autonomous mobile robot, depth camera, lidar, SLAM, Turtlebot2

## ŽIVOTOPIS

Luka Strinić rođen je 4. 4. 1996. godine u Vinkovcima. 2015 godine završio srednju tehničku školu „Ruđera Boškovića“ u Vinkovcima, smjer elektrotehničar, te godine 2015. upisao preddiplomski stručni studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku smjer automatika. Nakon završenog stručnog studija 2018. godine upisuje razlikovnu godinu. 2019. godine upisuje sveučilišni diplomski studij, smjer robotika i umjetna inteligencija.

Luka Strinić

---

## PRILOG

### P.4.3.1.1 Parametri GMapping algoritma uz korištenje lidara

Parametar	Vrijednost	Opis
map_update_interval	1.0	Vrijeme (u sekundama) između ažuriranja karte
maxUrange	14.0	Maksimalni upotrebljivi domet lasera
maxRange	16.0	Maksimalni domet senzora
sigma	0.05	Sigma koju koristi pohlepno podudaranje krajnje točke
kernalSize	1	Veličina jezgre za traženje podudaranja
lstep	0.05	Korak optimizacije u translaciji
astep	0.05	Korak optimizacije u rotaciji
iterations	5	Broj iteracija uparivanja skeniranja
lsigma	0.075	Sigma snopa upotrebljenog za izračun vjerojatnosti
ogain	3.0	Pojačanje koje se upotrebljava pri procjeni vjerojatnosti za izgladivanje učinaka ponovnog uzorkovanja
lskip	0	Broj zraka koje se preskoče u svakom skeniranju
minimumScore	800	Minimalni rezultat za uzimanje u obzir da je rezultat skeniranja dobar
srr	0.01	Odometrijska pogreška u translaciji kao rotacijska funkcija
srt	0.02	Odometrijska pogreška u translaciji kao translacijska funkcija
str	0.005	Odometrijska pogreška u rotaciji kao funkcija translacije
stt	0.02	Odometrijska pogreška u rotaciji kao funkcija rotacije
linearUpdate	0.05	Pređena udaljenost koja pokreće obradu skeniranja
angularAupdate	0.08726	Zakretni kut koji pokreće obradu skeniranja
temporalUpdate	-1.0	Obradi skeniranje ako je zadnje obrađeno skeniranje starije od vremena ažuriranja u sekundama. Vrijednost manja od nule isključit će ažuriranja temeljena na vremenu
resampleThreshold	0.5	$N_{\text{eff}}$ prag ponovnog uzorkovanja

particles	80	Broj čestica filtra
xmin	-1.0	Početna veličina karte (u metrima)
ymin	-1.0	Početna veličina karte (u metrima)
xmax	1.0	Početna veličina karte (u metrima)
ymax	1.0	Početna veličina karte (u metrima)
delta	0.05	Rezolucija mreže popunjenosti (u metrima)
llsamplerange	0.01	Translacijski raspon uzorkovanja za vjerojatnost
llsamplestep	0.01	Korak translacijskog uzorkovanja za vjerojatnost
lasamplerange	0.005	Kutni raspon uzorkovanja za vjerojatnost
lasamplestep	0.005	Kutni korak uzorkovanja za vjerojatnost

#### P.4.3.2.1 Parametri GMapping algoritma uz korištenje dubinske kamere

Parametar	Vrijednost	Opis
map_update_interval	1.0	Vrijeme (u sekundama) između ažuriranja karte
maxUrange	6.0	Maksimalni upotrebljivi domet lasera
maxRange	8.0	Maksimalni domet senzora
sigma	0.05	Sigma koju koristi pohlepno podudaranje krajnje točke
karnelSize	1	Veličina jezgre za traženje podudaranja
lstep	0.05	Korak optimizacije u translaciji
astep	0.05	Korak optimizacije u rotaciji
iterations	5	Broj iteracija uparivanja skeniranja
lsigma	0.075	Sigma snopa upotrebljenog za izračun vjerojatnosti
ogain	3.0	Pojačanje koje se upotrebljava pri procjeni vjerojatnosti za izgladivanje učinaka ponovnog uzorkovanja
lskip	0	Broj zraka koje se preskoče u svakom skeniranju
minimumScore	800	Minimalni rezultat za uzimanje u obzir da je rezultat skeniranja dobar
srr	0.01	Odometrijska pogreška u translaciji kao translacijska funkcija
srt	0.02	Odometrijska pogreška u translaciji kao rotacijska funkcija

str	0.005	Odometrijska pogreška u rotaciji kao funkcija translacije
stt	0.02	Odometrijska pogreška u rotaciji kao funkcija rotacije
linearUpdate	0.05	Pređena udaljenost koja pokreće obradu skeniranja
angularAupdate	0.08726	Zakretni kut koji pokreće obradu skeniranja
temporalUpdate	-1.0	Obradi skeniranje ako je zadnje obrađeno skeniranje starije od vremena ažuriranja u sekundama. Vrijednost manja od nule isključit će ažuriranja temeljena na vremenu
resampleThreshold	0.5	$N_{eff}$ prag ponovnog uzorkovanja
particles	80	Broj čestica filtra
xmin	-1.0	Početna veličina karte (u metrima)
ymin	-1.0	Početna veličina karte (u metrima)
xmax	1.0	Početna veličina karte (u metrima)
ymax	1.0	Početna veličina karte (u metrima)
delta	0.05	Rezolucija mreže popunjenosti (u metrima)
llsamplerange	0.01	Translacijski raspon uzorkovanja za vjerojatnost
llsamplestep	0.01	Korak translacijskog uzorkovanja za vjerojatnost
lasamplerange	0.005	Kutni raspon uzorkovanja za vjerojatnost
lasamplestep	0.005	Kutni korak uzorkovanja za vjerojatnost

#### P.4.4.1.1 Parametri RTAB-Map algoritma uz korištenje lidara

Parametar	Vrijednost	Opis
RGBD/NeighborLinkRefining	True	Kada se grafu doda novi čvor, transformacija njegove susjedne veze sa prethodnim čvorom preradi se pomoću odabranog registracijskog pristupa
RGBD/ProximityBySpace	True	Otkrivanje preko lokacija (u radnoj memoriji) u blizini u prostoru
RGBD/AngularUpdate	0.1	Minimalni kutni pomak (rad) za ažuriranje karte

RGBD/LinearUpdate	0.1	Minimalni linearni pomak (m) za ažuriranje karte
RGBD/OptimizeFromGraphEnd	False	Optimizacija grafa iz najnovijeg čvora. Ukoliko nije 1, graf se optimizira iz najstarijeg čvora trenutnog grafa.
Grid/FromDepth	False	Karta popunjenosti iz LIDARA-a
Reg/Force3DoF	True	Tranformacija sile 3 stupnja slobode
Reg/Strategy	1	0 = Vis, 1 = ICP, 2 = VisIcp
Icp/VoxelSize	0.05	Veličina voksel za uzorkovanje
Icp/MaxCorrespondenceDistance	0.1	Maksimalna udaljenost za korespondenciju točaka
Grid/MaxGroundHeight	0.03	Maksimalna visina tla (m)
Grid/MaxObstacleHeight	0.45	Maksimalna visina prepreke
Grid/NormalsSegmentation	False	Razlikovanje tla od prepreke pomoću normala točaka, u suprotnom se koristi brzi prijelaz
Grid/3D	0	1 = 3D karta, 0 = 2D karta
Grid/RangeMin	0.2	Najmanji raspon senzora
Odom/FilteringStrategy	0	0 = Bez filtra, 1 = Kalman filtar, 2 = čestični filtar
Grid/Sensor	0	Stvaranje mreže popunjenosti pomoću odabranog senzora. 0 = laser, 1 = dubinska kamera i 2 = laser i dubinska kamera
RGBD/ProximityPathMaxNeighbors	10	Maksimalan broj susjednih čvorova u usporedbi na svakom putu za detekciju blizine jedan-prema-više. Postaviti na 0 da biste onemogućili detekciju blizine jedan-prema-više



#### P.4.4.2.1 Parametri RTAB-Map algoritma uz korištenje dubinske kamere

Parametar	Vrijednost	Opis
RGBD/NeighborLinkRefining	True	Kada se grafu doda novi čvor, transformacija njegove susjedne veze sa prethodnim čvorom preradi se pomoću odabranog registracijskog pristupa
RGBD/ProximityBySpace	True	Otkrivanje preko lokacija (u radnoj memoriji) u blizini u prostoru
RGBD/AngularUpdate	0.1	Minimalni kutni pomak (rad) za ažuriranje karte
RGBD/LinearUpdate	0.1	Minimalni linearni pomak (m) za ažuriranje karte
RGBD/OptimizeFromGraphEnd	False	Optimizacija grafa iz najnovijeg čvora. Ukoliko nije 1, graf se optimizira iz najstarijeg čvora trenutnog grafa.
Grid/FromDepth	False	Karta popunjenosti iz LIDARA-a
Reg/Force3DoF	True	Tranformacija sile 3 stupnja slobode
Reg/Strategy	1	0 = Vis, 1 = ICP, 2 = VisIcp
Icp/VoxelSize	0.05	Veličina voksela za uzorkovanje
Icp/MaxCorrespondenceDistance	0.1	Maksimalna udaljenost za korespondenciju točaka
Grid/MaxGroundHeight	0.03	Maksimalna visina tla (m)
Grid/MaxObstacleHeight	0.45	Maksimalna visina prepreke
Grid/NormalsSegmentation	False	Razlikovanje tla od prepreke pomoću normala točaka, u suprotnom se koristi brzi prijelaz
Grid/3D	0	1 = 3D karta, 0 = 2D karta
Grid/RangeMin	0.4	Najmanji raspon senzora
Odom/FilteringStrategy	0	0 = Bez filtra, 1 = Kalman filtar, 2 = čestični filtar
Grid/Sensor	1	Stvaranje mreže popunjenosti pomoću odabranog senzora. 0 = laser, 1 = dubinska kamera i 2 = laser i dubinska kamera

RGBD/ProximityPathMaxNeighbors	10	Maksimalan broj susjednih čvorova u usporedbi na svakom putu za detekciju blizine jedan-prema-više. Postaviti na 0 da biste onemogućili detekciju blizine jedan-prema-više
--------------------------------	----	--

#### P.4.4.3.1 Parametri RTAB-Map algoritma uz korištenje lidara i dubinske kamere

Parametar	Vrijednost	Opis
RGBD/NeighborLinkRefining	True	Kada se grafu doda novi čvor, transformacija njegove susjedne veze sa prethodnim čvorom preradi se pomoću odabranog registracijskog pristupa
RGBD/ProximityBySpace	True	Otkrivanje preko lokacija (u radnoj memoriji) u blizini u prostoru
RGBD/AngularUpdate	0.1	Minimalni kutni pomak (rad) za ažuriranje karte
RGBD/LinearUpdate	0.1	Minimalni linearni pomak (m) za ažuriranje karte
RGBD/OptimizeFromGraphEnd	False	Optimizacija grafa iz najnovijeg čvora. Ukoliko nije 1, graf se optimizira iz najstarijeg čvora trenutnog grafa.
Grid/FromDepth	False	Karta popunjenosti iz LIDARA-a
Reg/Force3DoF	True	Transformacija sile 3 stupnja slobode
Reg/Strategy	1	0 = Vis, 1 = ICP, 2 = VisIcp
Icp/VoxelSize	0.05	Veličina vokseli za uzorkovanje
Icp/MaxCorrespondenceDistance	0.1	Maksimalna udaljenost za korespondenciju točaka
Grid/MaxGroundHeight	0.03	Maksimalna visina tla (m)
Grid/MaxObstacleHeight	0.45	Maksimalna visina prepreke
Grid/NormalsSegmentation	False	Razlikovanje tla od prepreke pomoću normala točaka, u suprotnom se koristi brzi prijelaz

Grid/3D	0	1 = 3D karta, 0 = 2D karta
Grid/RangeMin	0.2	Najmanji raspon senzora
Odom/FilteringStrategy	0	0 = Bez filtra, 1 = Kalman filtar, 2 = čestični filtar
Grid/Sensor	2	Stvaranje mreže potpunosti pomoću odabranog senzora. 0 = laser, 1 = dubinska kamera i 2 = laser i dubinska kamera
RGBD/ProximityPathMaxNeighbors	10	Maksimalan broj susjednih čvorova u usporedbi na svakom putu za detekciju blizine jedan-prema-više. Postaviti na 0 da biste onemogućili detekciju blizine jedan-prema-više

#### P.4.5.1.1 Parametri Cartographer algoritma uz korištenje lidara

Parametar	Vrijednos t	Opis
Num_laser_scans	1	Broj tema za lasersko skeniranje na koje se može pretplatiti
num_multi_echo_lasers_scans	0	Broj tema za multi-echo lasersko skeniranje na koje se može pretplatiti
Num_subdivisions_per_laser_scan	1	Broj oblaka točaka za razdiobu svakog primljenog (multi-echo) laserskog skeniranja
num_point_clouds	0	Broj tema oblaka točaka na koje se može pretplatiti
lookup_transform_timeout_sec	0.2	Vremenski odmak za traženje transformacija pomoću tf2
submap_publish_period_sec	0.3	Interval u sekundama za objavljivanje položaja podmapa
pose_publish_period_sec	5e-3	Interval u sekundama za objavljivanje položaja

trajectory_publish_period_sec	30e-3	Interval u sekundama u kojem se objavljuju markeri putanje
rangefinder_sampling_ratio	0.5	Fiksni omjer uzorkovanja za poruke mjerača udaljenosti
odometry_sampling_ratio	0.5	Fiksni omjer uzorkovanja za poruke o odometriji
fixed_frame_pose_sampling_ratio	1	Fiksni omjer uzorkovanja za poruke fiksnih okvira
imu_sampling_ratio	1	Fiksni omjer uzorkovanja za IMU poruke
landmarks_sampling_ratio	1	Fiksni omjer uzorkovanja za poruke o značajkama
min_range	0.2	Minimalni raspon senzora
max_range	14	Maksimalni raspon senzora
missing_data_ray_length	2	Točke izvan „max_range“ bit će umetnute s ovom duljinom kao prazan prostor
use_imu_data	False	Korištenje IMU podataka
use_online_correlative_scan_matching	True	Korištenje RealTimeCorrelativeScanMatcher
motion_filter.max_angle_radians	0.2	Prag iznad kojeg se podaci o rasponu umeću na temelju rotacijskog kretanja
ceres_scan_matcher.translation_weight	2e2	Vrijednost s kojom se penalizira uparivač skeniranja ukoliko pomakne rezultat od prethodnog
ceres_scan_matcher.rotation_weight	4e2	Vrijednost s kojom se penalizira uparivač skeniranja ukoliko pomakne rezultat od prethodnog
num_accumulated_range_data	1	Broj podataka o udaljenosti za akumulaciju u jedan kombinirani podatak o udaljenosti koji će se

		upotrebljavati za povezivanje skeniranja
constraint_builder.min_score	0.85	Prag ispod kojeg se rezultat podudaranja skeniranja smatra nevažećim
constraint_builder.global_localization_min_score	0.85	Prag ispod kojeg se globalna lokalizacija smatra nevažeća
optimization_problem.huber_scale	1e2	Parametar skaliranja za Huber funkciju gubitaka
optimize_every_n_nodes	35	Online zatvaranje petlje. Ako je pozitivan, pokrenit će zatvaranje petlje dok se karta gradi

#### P.4.5.2.1 Parametri Cartographer algoritma uz korištenje dubinske kamere

Parametar	Vrijednost	Opis
Num_laser_scans	1	Broj tema za lasersko skeniranje na koje se može pretplatiti
num_multi_echo_lasers_scans	0	Broj tema za multi-echo lasersko skeniranje na koje se može pretplatiti
Num_subdivisions_per_laser_scan	1	Broj oblaka točaka za razdiobu svakog primljenog (multi-echo) laserskog skeniranja
num_point_clouds	0	Broj tema oblaka točaka na koje se može pretplatiti
lookup_transform_timeout_sec	0.2	Vremenski odmak za traženje transformacija pomoću tf2
submap_publish_period_sec	0.3	Interval u sekundama za objavljivanje položaja podmapa
pose_publish_period_sec	5e-3	Interval u sekundama za objavljivanje položaja

trajectory_publish_period_sec	30e-3	Interval u sekundama u kojem se objavljuju markeri putanje
rangefinder_sampling_ratio	0.5	Fiksni omjer uzorkovanja za poruke mjerača udaljenosti
odometry_sampling_ratio	0.5	Fiksni omjer uzorkovanja za poruke o odometriji
fixed_frame_pose_sampling_ratio	1	Fiksni omjer uzorkovanja za poruke fiksnih okvira
imu_sampling_ratio	1	Fiksni omjer uzorkovanja za IMU poruke
landmarks_sampling_ratio	1	Fiksni omjer uzorkovanja za poruke o značajkama
min_range	0.4	Minimalni raspon senzora
max_range	8	Maksimalni raspon senzora
missing_data_ray_length	2	Točke izvan „max_range“ bit će umetnute s ovom duljinom kao prazan prostor
use_imu_data	False	Korištenje IMU podataka
use_online_correlative_scan_matching	True	Korištenje RealTimeCorrelativeScanMatcher
motion_filter.max_angle_radians	0.2	Prag iznad kojeg se podaci o rasponu umeću na temelju rotacijskog kretanja
ceres_scan_matcher.translation_weight	2e2	Vrijednost s kojom se penalizira uparivač skeniranja ukoliko pomakne rezultat od prethodnog
ceres_scan_matcher.rotation_weight	4e2	Vrijednost s kojom se penalizira uparivač skeniranja ukoliko pomakne rezultat od prethodnog
num_accumulated_range_data	1	Broj podataka o udaljenosti za akumulaciju u jedan kombinirani podatak o udaljenosti koji će se

		upotrebljavati za povezivanje skeniranja
constraint_builder.min_score	0.85	Prag ispod kojeg se rezultat podudaranja skeniranja smatra nevažećim
constraint_builder.global_localization_min_score	0.85	Prag ispod kojeg se globalna lokalizacija smatra nevažeća
optimization_problem.huber_scale	1e2	Parametar skaliranja za Huber funkciju gubitaka
optimize_every_n_nodes	35	Online zatvaranje petlje. Ako je pozitivan, pokrenit će zatvaranje petlje dok se karta gradi