

SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

*Ana Batinović, Ivona Jambrečić*

**Koordinirano višerobotsko istraživanje  
prostora zasnovano na *Graph SLAM*  
metodi i brzorastućim slučajnim stablima**

Zagreb, 2018.

Ovaj rad izrađen je u Laboratoriju za Robotiku i Inteligentne Sustave Upravljanja, na Zavodu za Automatiku i Računalno Inženjerstvo pod vodstvom prof. dr. sc. Stjepana Bogdana i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2017./2018.

# Sadržaj

<b>1. Uvod</b> . . . . .	<b>1</b>
<b>2. Mobilnost robota</b> . . . . .	<b>3</b>
2.1. Mobilni roboti . . . . .	3
2.2. Lokomocija . . . . .	4
2.3. Kinematika . . . . .	6
2.4. Odometrija . . . . .	7
<b>3. Istovremena lokalizacija i izgradnja karte prostora</b> . . . . .	<b>9</b>
3.1. Mrežasta karta zauzetosti prostora . . . . .	9
3.2. Istovremena lokalizacija i izgradnja karte prostora . . . . .	10
3.3. <i>Graph SLAM</i> . . . . .	11
3.4. Google Cartographer SLAM . . . . .	14
3.4.1. Stvaranje podkarte . . . . .	14
3.4.2. Usklađivanje rezultata laserskog očitavanja . . . . .	17
<b>4. Algoritam i strategija istraživanja prostora</b> . . . . .	<b>18</b>
4.1. RRT algoritam . . . . .	18
4.2. Strategija koordiniranog istraživanja prostora . . . . .	23
4.3. Transformacija stabla i točke fronte u koordinatni sustav karte . . . . .	26
<b>5. Eksperimentalni rezultati</b> . . . . .	<b>28</b>
5.1. Programsko okruženje . . . . .	28
5.1.1. Robot Operating System (ROS) . . . . .	28
5.1.2. Google Cartographer . . . . .	28
5.1.3. Ceres Solver . . . . .	29
5.1.4. 2D simulator Stage . . . . .	29
5.2. Planiranje i praćenje putanje uz ROS navigacijski slog . . . . .	30
5.2.1. Struktura . . . . .	30
5.2.2. Upravljanje robotom . . . . .	32

5.3. Izgradnja karte prostora i lokalizacija na stvarnom robotu . . . . .	33
5.3.1. Mobilni robot Pioneer P3-DX . . . . .	33
5.3.2. Povezivanje mobilnog robota i osobnog računala . . . . .	35
5.3.3. Rezultati uz <i>offline</i> čvor . . . . .	35
5.4. Koordinirano višerobotsko istraživanje prostora na stvarnom robotu . . . . .	36
5.4.1. Komunikacija centralnog računala i robota . . . . .	36
5.4.2. Eksperiment u stvarnom okruženju . . . . .	39
<b>6. Zaključak . . . . .</b>	<b>42</b>
<b>Zahvala . . . . .</b>	<b>43</b>
<b>Literatura . . . . .</b>	<b>46</b>
<b>Sažetak . . . . .</b>	<b>47</b>
<b>Summary . . . . .</b>	<b>48</b>



# 1. Uvod

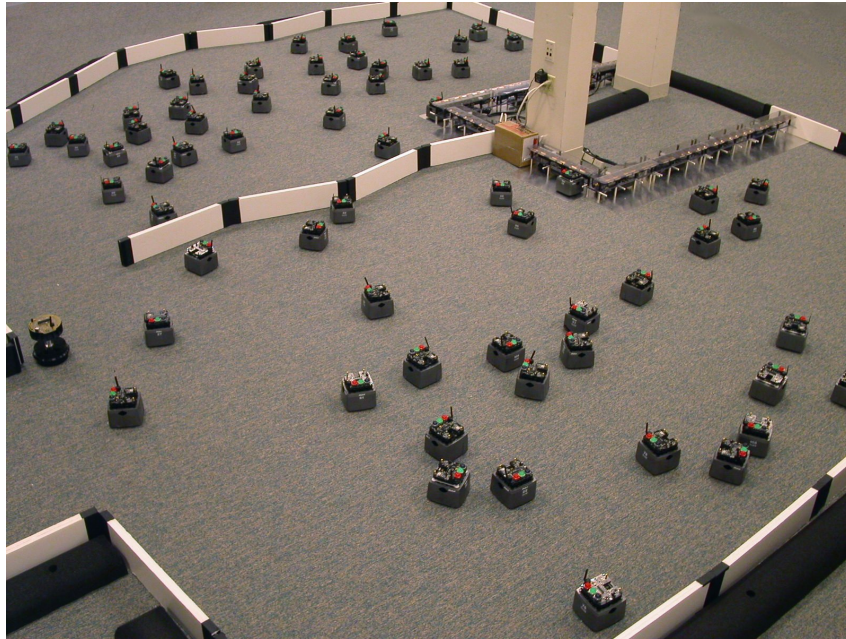
Razvoj civilizacije i tehnike kroz povijest doveo je do povećanja proizvodnje dobara. Kako bi se zadovoljile potrebe rastuće populacije za istima, iziskuje se ogromna količina rada. Taj rad je često težak, jednoličan i opasan. U želji za pronalaskom rješenja za ubrzanje i olakšanje rada, čovjek je razvio strojeve. Pojavom novih znanosti, koje omogućuju obradu informacija, strojevi postaju inteligentni i dobivaju naziv roboti.

Robotika je višedisciplinarna znanstvena grana koja objedinjuje znanja iz područja mehanike, elektronike, računarstva i automatike [1]. U modernom društvu značenje robotike postaje iz dana u dan sve veće zbog ubrzanog napretka znanosti i tehnologije. Sukladno tomu, pred robote se svakodnevno postavljaju zahtjevniji i složeniji zadaci te se povećavaju očekivanja korisnika.

Osim što se nastoje unaprijediti mogućnosti pojedinih robota za njihov samostalan rad, u svijetu robotike trenutno aktualno područje istraživanja su višerobotski sustavi autonomnog ponašanja. Takvi sustavi imaju velike prednosti pred sustavom jednog robota, no potrebne su napredne metode kojima je moguće postići željeno ponašanje. Akcije spašavanja, zajedničko čišćenje, nogometni tim te istraživanje prostora samo su neki od primjera primjene višerobotskih sustava. Primjer korištenja dan je na slici 1. Slika prikazuje višerobotski sustav koji koristi američka agencija DARPA za razvoj novih tehnologija za vojsku SAD-a [2].

Problem istraživanja okoline jedan je od temeljnih problema u mobilnoj robotici. Upravo iz tog problema nastala je ideja za ovaj rad te se razvila želja za povećanjem učinkovitosti koordiniranog višerobotskog istraživanja prostora nadogradnjom nekih od već postojećih rješenja. U ovom radu razmatra se problem istraživanja nepoznatog prostora s timom mobilnih robota.

Na samom početku rada dan je kratak opis mobilnih robota i njihove kinematike. U idućem poglavlju obrađuje se postupak istovremene lokalizacije i izgradnje karte prostora, na kojemu se zasniva koordinirano višerobotsko istraživanje prostora. Koristi se centralizirani pristup u kojem je centralno računalo u stalnoj komunikaciji sa svim robotima. Implementirani algoritam i strategija, koji omogućuju istraživanje prostora uz minimalni utrošak energije, opisani su u četvrtom poglavlju. U posljednjem poglavlju rada prikazani su rezultati ispitivanja u simulatoru i na stvarnim robotima.



Slika 1. Višerobotski sustav američke agencije DARPA-e u sklopu projekta Swarm

## 2. Mobilnost robota

### 2.1. Mobilni roboti

Prema ISO 8370 robot je definiran kao automatski upravljani, programibilni, višenamjenski manipulator koji može izvršavati zadaće u tri ili više osi te ovisno o primjeni može biti stacionaran ili mobilan. Međutim, ova definicija ne vrijedi za današnje robote jer su zahtjevi koji se stavljaju pred njih daleko složeniji [3]. Neki od zahtjeva za mobilne robote su:

- autonomno gibanje u nepoznatom prostoru
- obavljanje zadatka različite složenosti
- interakcija s drugim robotima i ljudima
- samostalno učenje i zaključivanje

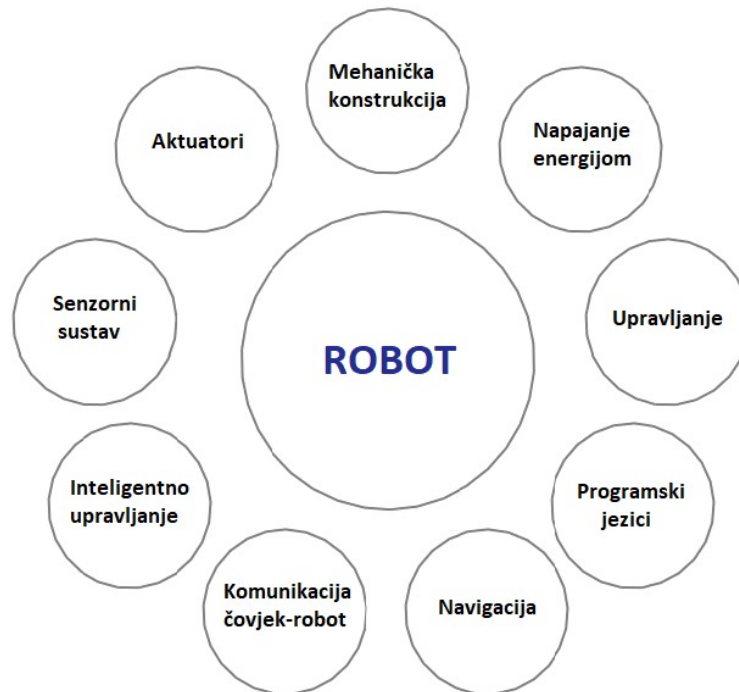
Sukladno ovim zahtjevima pretpostavlja se nova definicija robota koja kaže da su mobilni roboti mobilni i manipulativni fizički sustavi koji se autonomno gibaju kroz nestrukturirani prostor ostvarujući pritom interakciju s ljudskim bićima ili autonomno obavljaju neki zadatak umjesto njih. Nazvani su uslužnim robotima te se smatraju međukorakom između industrijskih robota današnjice i personaliziranih inteligentnih robota budućnosti koji će biti sposobni ponašati se kao ljudi [3]. Sastavni dijelovi robota prikazani su slikom 2.

Osim toga, pri razmatranju mobilnih robota moguće ih je gledati iz perspektive problema koje treba riješiti :

- primjena – industrija, kućanstva, svemir, poljoprivreda, vojne primjene
- realizacija gibanja – noga, kotač, ravnoteža, vrste kretanja
- interakcija s okolinom – kamere, stereo-kamere, senzori, manipulatori
- pronalaženje putanje kretanja – poznata okolina, nepoznata okolina

Mobilne robote je moguće primjeniti i na raznim drugim područjima (poljoprivreda, šumarstvo, inspekcija rizičnih područja, rudarstvo, građevina, istraživanje svemira i podmorja) zbog čega njihova uporaba s vremenom raste.

U ovom radu korišteni su mobilni roboti s diferencijalnim pogonom čije su karakteristike i funkcije opisane u nastavku.



Slika 2. Sastavni dijelovi robota

## 2.2. Lokomocija

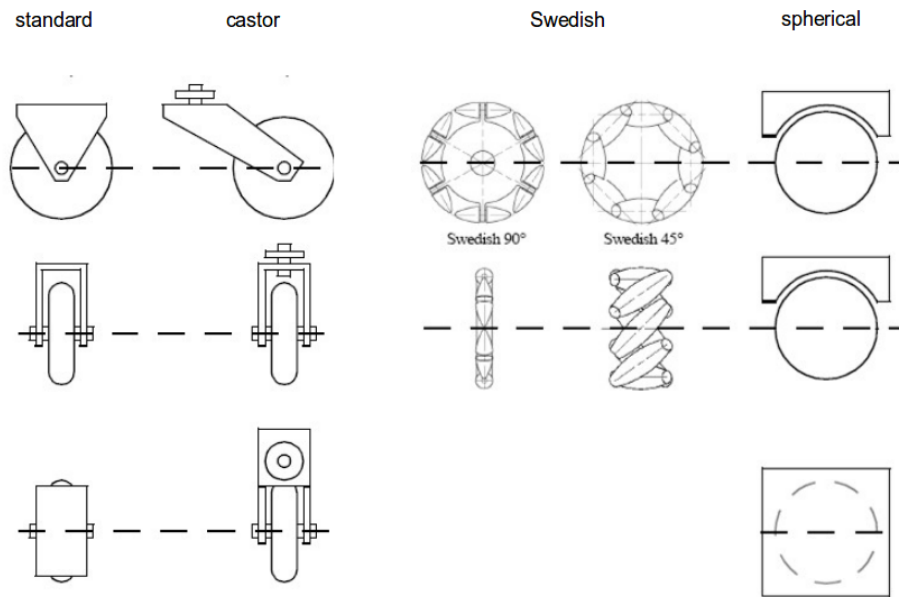
Glavni izazov kojime se bavi mobilna robotika je lokomocija. Lokomocija je proces koji uzrokuje gibanje ili sposobnost pomicanja s jednog mjesta na drugo. Postoje različiti načini pomicanja robota, ovisno o zadatku koji obavljaju. Takvi načini gibanja su često nadahnuti prirodom. Primjeri mehanizama lokomocije su hodanje, skakanje, trčanje i kotrljanje.

Mobilnom robotu potrebni su lokomocijski mehanizmi, koji mu omogućuju gibanje kroz prostor, kao što su aktuatori sile ili drugi mehanizmi sa željenim kinematičkim i dinamičkim svojstvima. Prilikom odabira mehanizma lokomocije razmatraju se različiti kriteriji, npr. brzina, stabilnost i efikasnost. Najčešće korišteni mehanizmi su kotači i noge.

Postoje četiri osnovna tipa kotača:

- standardni kotač - ima dva stupnja slobode, rotaciju oko osovine kotača i rotaciju oko točke kontakta
- kastorski kotač (engl. *castor wheel*) - ima dva stupnja slobode, rotaciju oko osovine kotača i kastorske osi
- švedski kotač - ima tri stupnja slobode, rotaciju oko osovine kotača, točke kontakta i oko rolera

- sferni kotač - tehnički teško izvediv

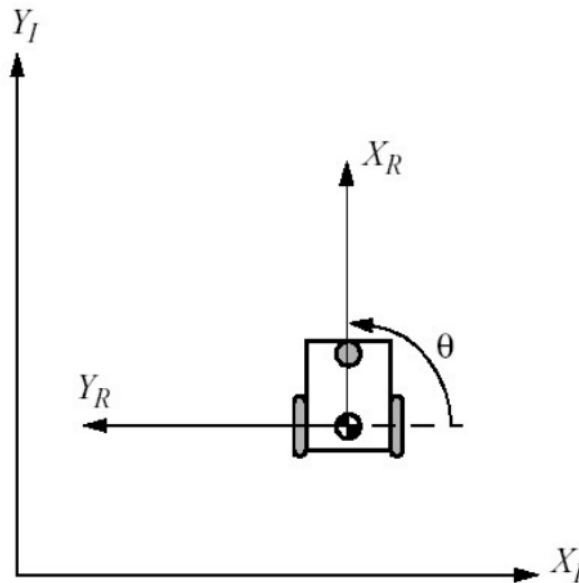


Slika 3. Vrste kotača

### 2.3. Kinematika

Kinematika je grana mehanike koja proučava gibanje tijela, odnosno mehaničkih sustava. U mobilnoj robotici potrebno je razumijevanje mehaničkog ponašanja kako bi se mobilni roboti najprikladnije konstruirali, a potom njima odgovarajuće upravljalo. Objašnjenje gibanja robota započinje opisivanjem doprinosa svakog kotača tom gibanju.

Sljedećim izrazima prikazat će se robotsko gibanje u globalnom referentnom sustavu i u lokalnom koordinatnom sustavu robota. Zatim će se izraziti direktna kinematika robota te kinematička ograničenja gibanja [4]. Poza robota određena je transformacijom između globalnog referentnog koordinatnog sustava  $I$  s ishodištem u  $[X_I, Y_I]$  i koordinatnog sustava robota s ishodištem u bazi robota  $[X_R, Y_R]$ .



Slika 4. Transformacija robota u globalni referentni sustav [4]

Poza robota definira se kao vektor s tri elementa, pri čemu  $I$  označava da je poza izražena u odnosu na globalni referentni sustav.

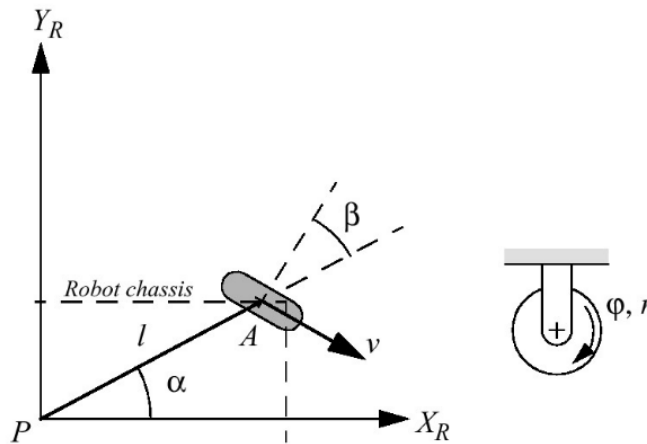
$$\mathbf{x}^I = \begin{bmatrix} x & y & \theta \end{bmatrix}^T, \quad (1)$$

$$\dot{\mathbf{x}}^I = \mathbf{R}_I^R(\theta) \dot{\mathbf{x}}^R = \mathbf{R}_I^R(\theta) \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T. \quad (2)$$

Homogena transformacija ovisi o globalnom kutu  $\theta$ .

$$\mathbf{R}_I^R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Kinematički model gibanja moguće je zapisati pomoću jednadžbi kinematičkih ograničenja. Kutevi  $\alpha, \beta, \varphi$  vidljivi su na slici 5.



Slika 5. Kinematika robota [4]

Postoje dvije vrste kinematičkih ograničenja:

- ograničenje kotrljanja: sva gibanja u ravnini kotača proiciraju se sa odgovarajućim kotrljanjem

$$\begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -l\cos(\beta) \end{bmatrix} \mathbf{R}(\theta)\dot{\mathbf{x}}^I - r\dot{\varphi} = 0, \quad (4)$$

- zabrana lateralnog gibanja (klizanja): gibanja ortogonalna na os kotača iznose nula

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l\sin(\beta) \end{bmatrix} \mathbf{R}(\theta)\dot{\mathbf{x}}^I = 0. \quad (5)$$

## 2.4. Odometrija

Odometrija je metoda za estimaciju položaja mobilnih robota u stvarnom vremenu. Korištenjem senzora pokreta moguće je izračunati relativnu promjenu pozicije robota. Poznavanje

pozicije i orijentacije robota u prostoru smatramo lokalizacijom. Za matematički izračun odometrije, pozicija robota je izražena kao vektor  $\mathbf{x}^I$ :

$$\mathbf{x}^I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}. \quad (6)$$

Za mobilnog robota s diferencijalnim pogonom pozicija se estimira početkom u poznatoj točki i integriranjem kretnje odnosno sumiranjem pređene udaljenosti. Za diskretni sustav s vremenom uzorkovanja  $\Delta t$  računaju se inkrementalne udaljenost  $(\Delta x, \Delta y, \Delta \theta)$ .

$$\Delta s = \frac{\Delta s_l + \Delta s_r}{2}, \quad (7)$$

$$\Delta \theta = \frac{\Delta s_r - \Delta s_l}{b}. \quad (8)$$

U izrazima  $\Delta s_r$  i  $\Delta s_l$  označavaju udaljenosti koje lijevi i desni kotač prijeđu,  $\Delta \theta$  promjenu orijentacije robota, a  $b$  je izmjerena udaljenost između dva kotača robota.

Ažuriranje estimacije poze računa se pomoću izraza:

$$\mathbf{x}_t^I = \mathbf{x}_{t-1}^I + \begin{bmatrix} \Delta s \cos(\theta_{t-1} + \frac{\Delta \theta}{2}) \\ \Delta s \sin(\theta_{t-1} + \frac{\Delta \theta}{2}) \\ \Delta \theta \end{bmatrix}. \quad (9)$$

Ažurirana poza  $\mathbf{x}_t^I$  dobiva se dodavanjem prethodno izračunatoj pozi  $\mathbf{x}_{t-1}^I$  inkrementalnih udaljenosti, izračunatih pomoću (7), (8) i orijentacije  $\theta_{t-1}$ .



### 3. Istovremena lokalizacija i izgradnja karte prostora

Glavni zadatci za postizanje autonomnosti robota su modeliranje okoline (izgradnja karte prostora), određivanje poze robota (lokalizacija) i mogućnost pomicanja na druge pozicije (navigacija). Metoda koja se koristi za lokalizaciju i navigaciju temeljena je na korištenju karte prostora te senzorskih i odometrijskih mjerenja.

#### 3.1. Mrežasta karta zauzetosti prostora

Mrežasta karta zauzetosti opisuje prostor pomoću diskretnih područja (ćelija) kvadratnog oblika, jednoliko raspoređenih na mrežastoj karti, gdje je svakom području dodijeljena binarna slučajna varijabla koja predstavlja zauzetost područja. Algoritam za izgradnju mrežaste karte zauzetosti prostora temelji se na procjeni vrijednosti zauzetosti pomoću posteriorne vjerojatnosti binarnih slučajnih varijabli [5].

Posteriorna vjerojatnost dana je oznakom

$$p(\mathbf{m} | \mathbf{z}_{1..t}, \mathbf{x}_{1..t}), \quad (10)$$

pri čemu je  $\mathbf{m}$  karta,  $\mathbf{z}_{1..t}$  sva mjerenja do uključivo vremena  $t$ , a  $\mathbf{x}_{1..t}$  je trajektorija robota. Karta  $\mathbf{m}$  podijeljena je u  $i$  ćelija s oznakama  $m_i$ . Svakoj ćeliji  $m_i$  dodijeljena je binarna vrijednost zauzetosti. Vjerojatnost zauzetosti ćelije ima oznaku  $p(m_i)$ , dok združena vjerojatnost karte ima oznaku  $p(\mathbf{m})$ .

Primjenom rekurzivnog Bayesovog pravila uz pretpostavku Markovljevog svojstva<sup>1</sup> dobiva se rekurzivni izraz za ažuriranje posteriorne vjerojatnosti zauzetosti ćelije:

$$p(m_i | \mathbf{z}_{1..t}, \mathbf{x}_{1..t}) = \frac{p(m_i | \mathbf{z}_t, \mathbf{x}_t)}{1 - p(m_i | \mathbf{z}_t, \mathbf{x}_t)} \frac{p(m_i | \mathbf{z}_{1..t-1}, \mathbf{x}_{1..t-1})}{1 - p(m_i | \mathbf{z}_{1..t-1}, \mathbf{x}_{1..t-1})} \frac{1 - p(m_i)}{p(m_i)}. \quad (11)$$

<sup>1</sup>uvjetna vjerojatnosna distribucija budućeg stanja slučajnog procesa ovisi samo o trenutnom stanju, tj. sva informacija je sadržana u trenutnom stanju

### 3.2. Istovremena lokalizacija i izgradnja karte prostora

Istovremena lokalizacija i izgradnja karte prostora (eng. *Simultaneous localization and mapping*), skraćeno SLAM, obuhvaća estimaciju poze robota i karte okoline u isto vrijeme. Složenost SLAM-a proizlazi iz činjenice da su lokalizacija i karta međusobno zavisne. Međusobna zavisnost očituje se u tome što lokalizacija obuhvaća estimiranje poze robota u izgrađenoj karti, dok se karta izgrađuje upravo prema pozama koje su rezultat lokalizacije.

SLAM algoritmi imaju primjenu u širom spektru robotskih sustava, kao što su autonomna ili od čovjeka upravljana vozila, u okolinama poput zatvorenog i otvorenog prostora, u zraku ili pod vodom. Na primjer, autonomni kućanski usisavač, kosač trave, nadzor neba uz pomoć bespilotnih letjelica ili pretraživanje mina.

U ovom odjeljku opisana su dva pristupa izvođenja SLAM-a: probabilistički pristup i pristup temeljen na konstrukciji i optimizaciji grafa (eng. *Graph SLAM*). Ulazni podatci za SLAM su senzorska mjerenja s robota te odometrija.

U probabilističkoj formulaciji SLAM-a estimiraju se pozicija i orijentacija robota (poza) te karta okoline. Trajektorija je zapisana kao vektor čvorova trajektorije

$$\mathbf{x}_{0:t} = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t]. \quad (12)$$

Odometrijska mjerenja, s oznakom  $\mathbf{u}_t$ , zapisana su u vektor mjerenja

$$\mathbf{u}_{1:t} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_t]. \quad (13)$$

Kada ne bi postojala pogreška u odometrijskim mjerenjima, ona bi točno pratila stvarnu trajektoriju. Zbog akumuliranja pogreške, mjerenja odstupaju od stvarne trajektorije. Uobičajeno se pretpostavlja da se pogreška može opisati kao bijeli šum, to jest, kao Gaussova slučajna varijabla s očekivanjem nula, koja ne korelira sa svojim prethodnim ili budućim vrijednostima. Za slučajni vektor  $\mathbf{x}$  s normalnom razdiobom očekivanja  $\boldsymbol{\mu}$  i kovarijance  $\boldsymbol{\Sigma}$ , gustoća vjerojatnosti  $p(\mathbf{x})$  iznosi

$$p(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right). \quad (14)$$

Uz odometriju, ulazni podatci su i senzorska mjerenja okoline  $z_t$ , zapisana u vektoru mjerenja

$$\mathbf{z}_{1:t} = [z_1, z_2, \dots, z_t]. \quad (15)$$

Rješenje SLAM-a definira se kao estimacija posteriorne združene vjerojatnosti poze robota i karte

$$p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}). \quad (16)$$

Primjenom Bayesovog pravila na (16) dobiva se faktorizirani izraz za posteriornu vjerojatnost:

$$p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \eta p(z_t | \mathbf{x}_{0:t}, \mathbf{m}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}), \quad (17)$$

gdje je  $\eta$  konstantan član koji obuhvaća izraz u nazivniku,  $p(z_t | \mathbf{x}_{0:t}, \mathbf{m}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$  senzorski model, a  $p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$  model gibanja robota.

Primjenom Markovljeve pretpostavke  $p(z_t | \mathbf{x}_{0:t}, \mathbf{m}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = p(z_t | \mathbf{x}_t, \mathbf{m})$  dobiva se konačni izraz za posteriornu vjerojatnost

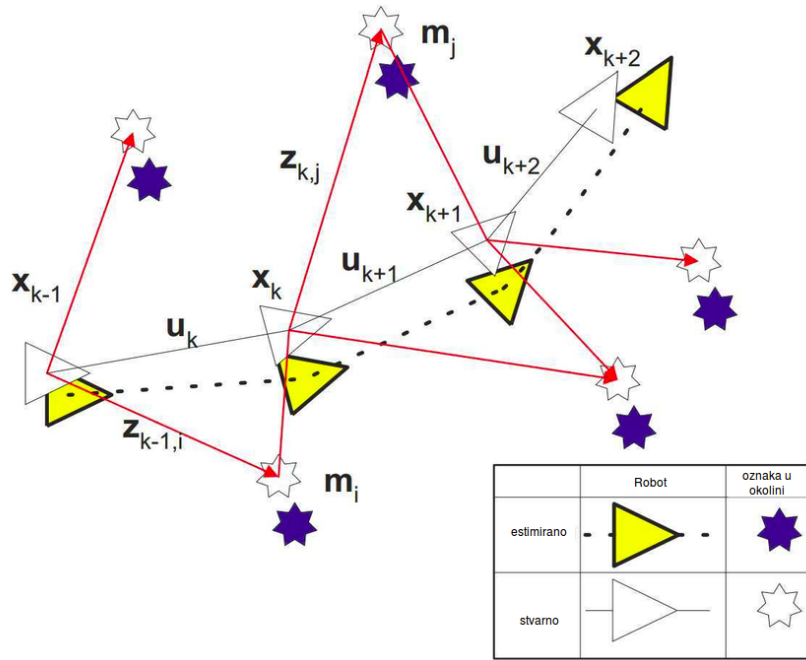
$$p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \eta p(\mathbf{x}_0) \prod_t p(z_t | \mathbf{x}_t, \mathbf{m}) p(\mathbf{x}_t, \mathbf{m} | \mathbf{x}_{t-1}, \mathbf{u}_t). \quad (18)$$

Na slici 6. ilustrirano je izvođenje SLAM-a. Robot se giba okolinom te se na temelju senzorskih mjerenja  $z_k$  i odometrije  $u_k$  estimira karta  $\mathbf{m}$  i poze čvorova trajektorije  $\mathbf{x}_k$ .

### 3.3. Graph SLAM

U implementaciji algoritma za istraživanje prostora korištena je varijanta SLAM-a poznata pod nazivom *Graph SLAM*, koja pomoću *grafa*<sup>2</sup> prikazuje međusobne ovisnosti čvorova trajektorije i senzorskih mjerenja [6]. U *Graph SLAM*-u uobičajeno se za čvorove grafa odabiru poze robota (čvorovi trajektorije)  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t$ , a bridovi grafa su prostorna ograničenja (eng. *constraint*) između čvorova trajektorije koja su dobivena iz senzorskih mjerenja  $z_t$  ili odometrije  $u_t$ .

<sup>2</sup>apstraktna matematička struktura koja prikazuje odnose između objekata, uobičajeno definirana kao uređeni par  $(V, E)$  skupa čvorova  $V$  i skupa bridova  $E$



Slika 6. Usporedba stvarnih i estimiranih poza i oznaka karte

Rješenje *Graph SLAM*-a, optimalna trajektorija i karta, dobivaju se minimizacijom greške prostornih ograničenja, koja je ilustrirana na slici 7.

Prostorno ograničenje je *virtualno mjerenje* poze čvora trajektorije  $x_j$  gledano iz čvora  $x_i$  [7]. Neka je  $z_{ij}$  poza virtualnog mjerenja, a  $\Omega_{ij}$  informacijska matrica virtualnog mjerenja. Na primjer, kod zatvaranja petlje, virtualno mjerenje može se dobiti iz transformacije koja usklađuje laserska mjerenja iz čvora  $i$  s onima iz čvora  $j$ .

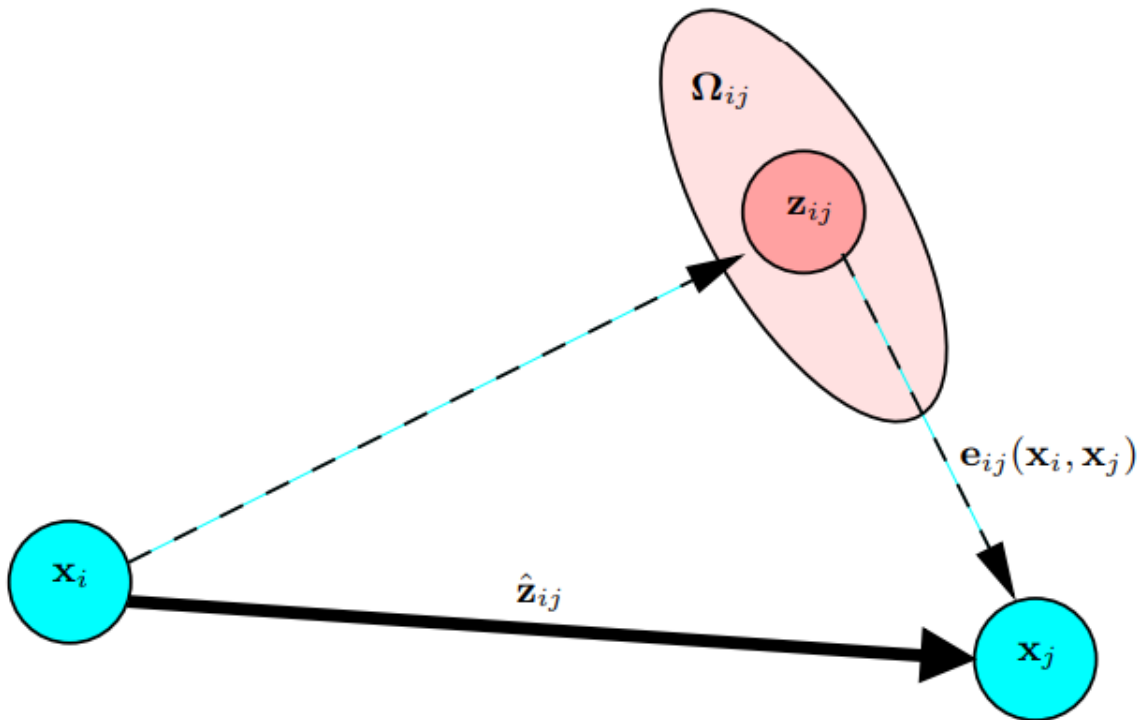
Relativna poza  $\hat{z}_{ij}$  dvaju čvorova  $x_i$  i  $x_j$  dana je *jednadžbom mjerenja*:

$$\hat{z}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \begin{bmatrix} \mathbf{R}_i^T \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \\ \theta_j - \theta_i \end{bmatrix}, \quad (19)$$

gdje je  $\mathbf{R}_i$  2x2 rotacijska matrica kuta  $\theta_i$ .

Greška prostornog ograničenja definirana je kao razlika poze virtualnog mjerenja i relativne poze dvaju čvorova:

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) := \mathbf{z}_{ij} - \hat{z}_{ij}(\mathbf{x}_i, \mathbf{x}_j). \quad (20)$$



Slika 7. Optimizacijski problem *Graph SLAM*-a [6]

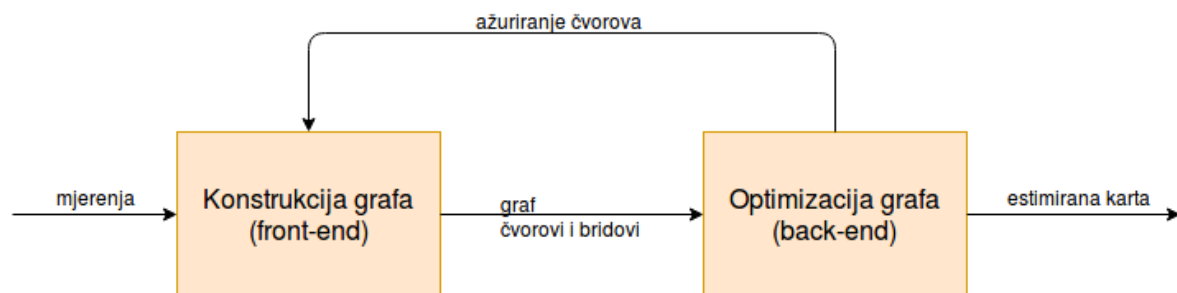
Optimizacijski postupak je minimizacija greške svih prostornih ograničenja:

$$F(\mathbf{x}_{0:t}) = \sum_{i,j} \mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}. \quad (21)$$

Minimizacija gornje kriterijske funkcije dat će optimalnu trajektoriju koja je rješenje *Graph SLAM*-a:

$$\mathbf{x}_{0:t}^* = \underset{\mathbf{x}_{0:t}}{\operatorname{argmin}} F(\mathbf{x}_{0:t}). \quad (22)$$

Arhitekturu SLAM sustava moguće je rastaviti na dvije veće komponente, pozadinski sustav (eng. *back-end*) i prednji sustav (eng. *front-end*) [7]. U prednjem sustavu se obrađuju senzorska mjerenja i izgrađuje graf, a u pozadinskom se rješava optimizacijski problem najmanjih kvadrata opisan u (22).



Slika 8. SLAM arhitektura

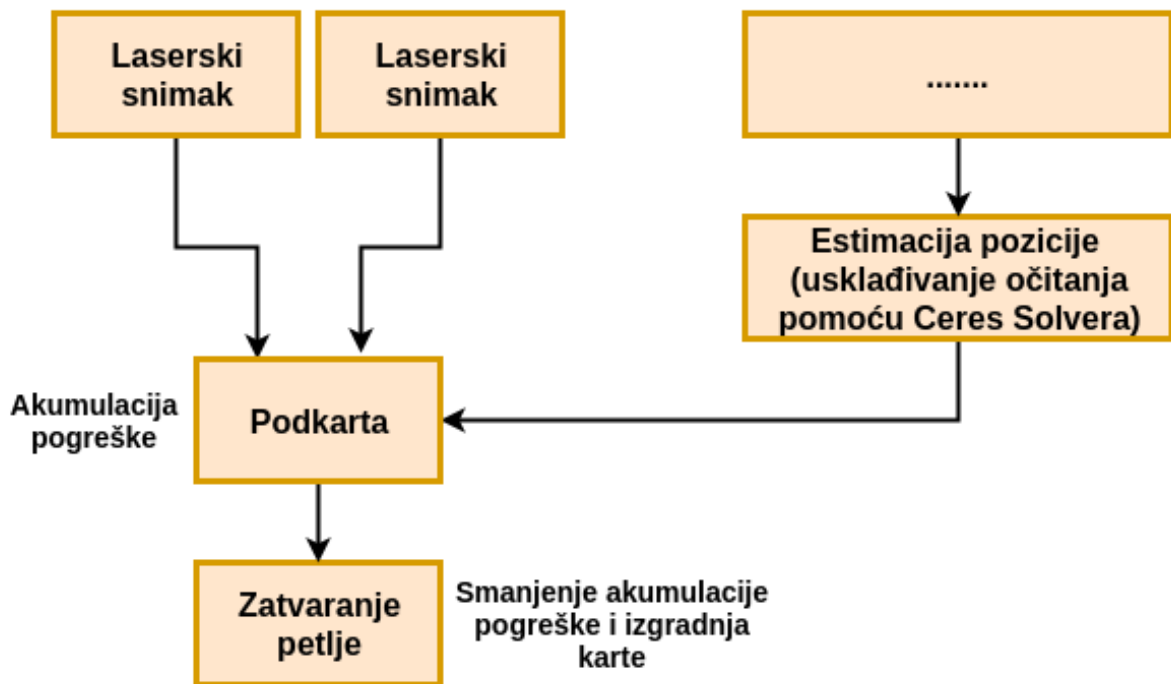
### 3.4. Google Cartographer SLAM

Google Cartographer je moderna i vrlo efikasna implementacija iz grupe *Graph SLAM* metoda [8]. Pomoću njega se provodi izgradnja karte zatvorenog prostora i lokalizacija u stvarnom vremenu uz mogućnost vizualizacije. Laserski snimci, koji se dobivaju od mobilnog robota, umeću se u podkarte, za koje se pretpostavlja da su dovoljno točne u kratkom vremenskom intervalu [9]. Usklađivanje laserskih očitavanja (eng. *scan-matching*) provodi se na trenutnoj podkarti. Usklađivanjem laserskog očitavanja dobiva se nova poza trajektorije. Dio Cartographera koji provodi usklađivanje laserskog očitavanja naziva se lokalni SLAM.

Međutim, trajektorija koju gradi lokalni SLAM podložna je pomicanju, što se dalje ispravlja u dijelu Cartographera koji se naziva globalni SLAM. Radi ispravljanja akumulirane greške, periodički se pokreće optimizacija poza podkarata i svih čvorova trajektorije. Jednom kada je podkarta dovršena, u nju se ne dodaju novi laserski snimci. Sve dovršene podkarte i laserski snimci uzimaju se u obzir za usklađivanje snimaka tijekom zatvaranja petlji. Optimizacija se vrši svakih nekoliko sekundi, a rezultat je zatvaranje petlje kad se ustanovi da je mobilni robot ponovno došao na mjesto koje je prethodno posjetio. Blokovski prikaz dan je slikom 9.

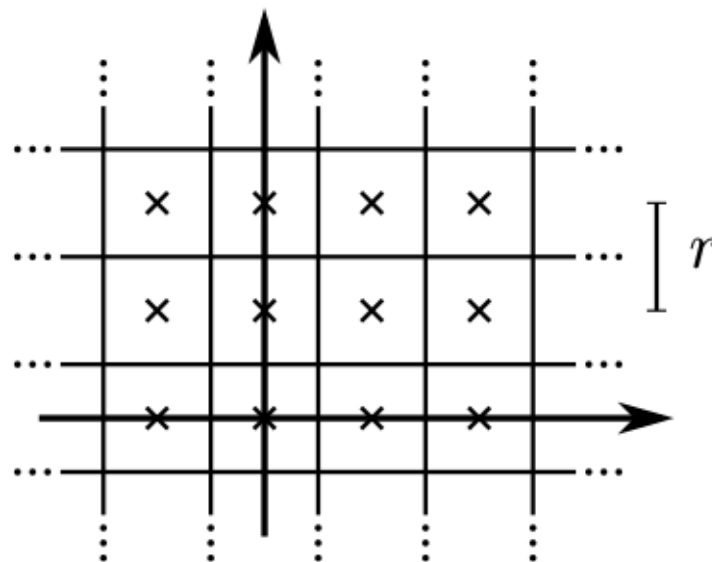
#### 3.4.1. Stvaranje podkarte

Jedna podkarta nastaje iz točno određenog broja laserskih snimaka, parametra algoritma koji se još naziva i veličina podkarte. Ovaj parametar odabire se tako da podkarte budu dovoljno malene, a da se unutar njih ne stigne akumulirati veća količina pogreške (engl. *drift*) lokalnog SLAM-a. Podkarte imaju formu mrežaste karte zauzetosti  $M: r\mathbb{Z} \times r\mathbb{Z} \rightarrow [p_{min}, p_{max}]$ , gdje je  $M$  funkcija koja preslikava iz skupa točaka mreže u vrijednosti iz intervala  $[p_{min}, p_{max}]$ .



Slika 9. Blokovski prikaz nastajanja podkarte i zatvaranja petlje

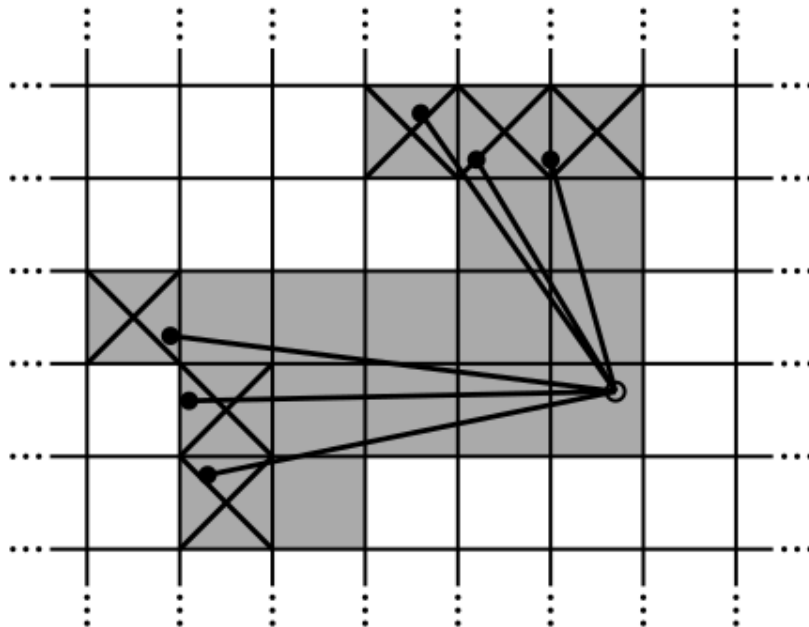
Rezoluciju mrežaste karte  $r$  definira se kao razmak između točaka mreže i uobičajeno iznosi 5 cm po ćeliji. Točke mreže i njima pridružene ćelije te rezolucija mrežaste karte  $r$  ilustrirani su na slici 10.



Slika 10. Točke mreže i pridružene ćelije te rezolucija mrežaste karte  $r$  [9]

Vrijednosti pridružene točkama mreže zauzetosti predstavljaju vjerojatnost da je određena točka mreže zauzeta. Svakoj točki mreže pridružena je ćelija koja obuhvaća područje u kojem je ta točka bliža od svih ostalih točaka mreže. Vjerojatnost zauzetosti pridružena točki mreže vrijedi za cijelu ćeliju.

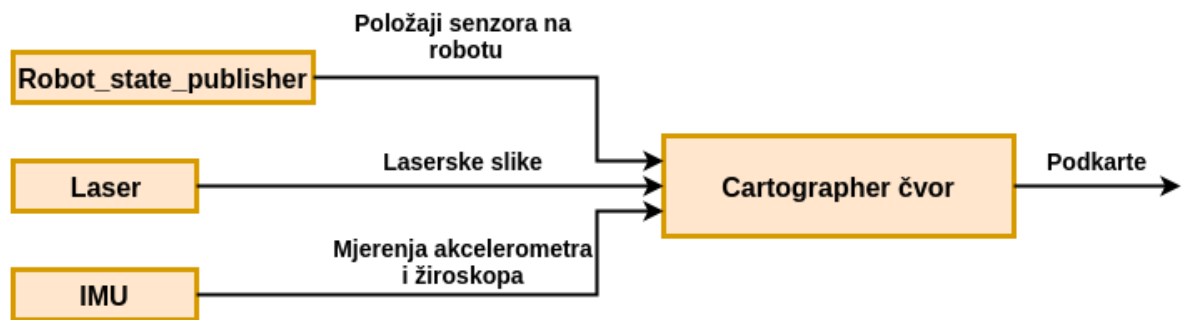
Svaki put kad se laserski snimak umetne u mrežastu kartu zauzetosti, računaju se dva disjunktna skupa točaka, skup "pogođeni" i skup "promašeni". Iz pozicije robota povuče se pravac u smjeru očitavanja na laserskoj slici. Sve ćelije ispod pravca koje se nalaze između robota i prepreke označavaju se kao "promašene", dok se točka prepreke označava kao "pogođena". "Pogođeni" su na slici 11. prikazani zatamnjeno i prekriženi su, a "promašeni" su samo zatamnjeni.



Slika 11. Skupovi "pogođeni" i "promašeni" [9]

Podkarte nastale na prethodno opisan način Cartographer sprema u listu podkarata, kako je prikazano na slici 12.





Slika 12. Tok podataka prilikom stvaranja podkarata u Cartographer SLAM-u

### 3.4.2. Usklađivanje rezultata laserskog očitavanja

Usklađivanje rezultata laserskog očitavanja radi na principu preklapanja laserskog očitavanja i podkarte. Prije umetanja očitavanja u podkartu, poza očitavanja  $x$  je optimizirana u odnosu na trenutnu lokalnu podkartu koristeći usklađivanje laserskog očitavanja. Funkcija za usklađivanje laserskog očitavanja služi za pronalazak poze robota koja maksimizira vjerojatnost zauzetosti u podkarti u točkama pogodnim očitanjem. Koristi se nelinearna kvadratna funkcija dana izrazom

$$\arg \min_x \sum_{k=1}^K (1 - M_{glatka}(\mathbf{T}_x \mathbf{h}_k))^2, \quad (23)$$

gdje je  $K$  broj točaka pogodnih laserskim očitanjem,  $\mathbf{T}_x$  transformacija poze robota prema koordinatnom sustavu podkarte, a  $\mathbf{h}_k$  pozicija pogođene točke očitavanja prema koordinatnom sustavu robota [9]. Funkcija  $M: \mathbb{R}^2 \rightarrow \mathbb{R}$  je glatka funkcija vjerojatnosti u lokalnoj podkarti. Optimizacija gornje funkcije može dati veću preciznost uz koju je pogreška manja nego što iznosi rezolucija mrežaste karte.

## 4. Algoritam i strategija istraživanja prostora

Globalna karta prostora nije dostupna u trenutku kada mobilni roboti započinju svoje kretanje. Cilj je da oni, u komunikaciji s ostalim robotima, izgrade kartu zadanog prostora. Do danas je predstavljen veliki broj algoritama za autonomno i koordinirano istraživanje prostora. Pojedini algoritmi namijenjeni su jednom robotu, a pojedini timu robota. Neke metode stavljaju naglasak na smanjenje vremena istraživanja ili na povećanje kvalitete izgradnje karte [10]. Važno je odabrati najbolje rješenje s obzirom na postavljene zahtjeve i željene ciljeve. U ovom radu stavlja se naglasak na koordinirani višerobotski sustav, u kojem roboti međusobno dijele saznanja o istraženom prostoru. Koristi se centralizirani pristup u kojem svi roboti šalju informacije u zajedničko središte, koje dodjeljuje zadatke pojedinim robotima primjenjujući odabranu strategiju istraživanja. Rezultat je koordinirano izgrađena globalna karta uz minimalno utrošeno vrijeme.

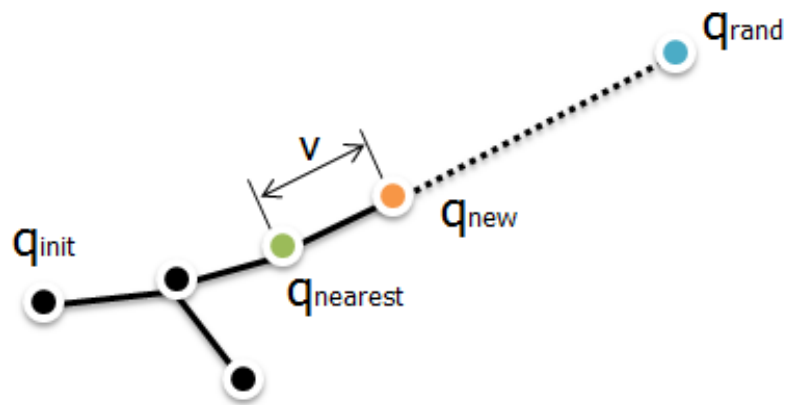
Čest pristup u strategijama autonomnog istraživanja prostora je usmjeravanje robota prema granici istraženog i neistraženog prostora. U ovom radu odabrana je istraživačka strategija koja se temelji na algoritmu detekcije granice istraženog prostora pomoću brzorastućih slučajnih stabala (eng. *Rapidly Exploring Random Tree*), skraćeno RRT-a. RRT algoritam je pristran prema neistraženim područjima te ga je lako proširiti na višedimenzionalni prostor [11]. Također, pomoću RRT-a se jednostavno osigurava zaobilazanje prepreka pri gradnji stabla i zadovoljavanje različitih prostornih ograničenja. RRT ima veliku primjenu u autonomnom robotskom planiranju kretanja. Slijedi detaljniji opis RRT algoritma.

### 4.1. RRT algoritam

RRT algoritam planiranja se temelji na inkrementalnoj gradnji stabla nasumičnim uzorkovanjem položaja iz prostora za pretraživanje [12]. Dizajniran je za učinkovito pretraživanje visokodimenzionalnih prostora slučajnom gradnjom stabala koja ispunjavaju prostor.

Grananjem se na nasumično uzorkovanom položaju  $q_{rand}$  pokušava izgraditi novi čvor koji će biti povezan kroz istražen prostor s najbližim čvorom u stablu  $q_{nearest}$ . Ako je veza izvediva, rezultat je stvaranje novog čvora u stablu. Duljina veze između najbližeg čvora  $q_{nearest}$  (slika 13.) i novog čvora  $q_{new}$  ograničena je faktorom rasta  $v$ . Ako je slučajni uzorak dalje od najbližeg

čvora  $q_{nearest}$  u stablu nego što to dopušta ograničenje, za stvaranje novog čvora  $q_{new}$ , umjesto slučajnog uzorka  $q_{rand}$ , koristi se položaj na maksimalnoj udaljenosti od najbližeg čvora  $q_{nearest}$  u smjeru slučajnog uzorka  $q_{rand}$ . Nasumični uzorci kontroliraju smjer rasta stabla, a faktor rasta određuje brzinu grananja. Time se održava pristranost ispunjavanja neistraženog prostora RRT-om, uz istovremeno ograničavanje rasta kako bi se osiguralo gusto ispunjavanje istraženog prostora.



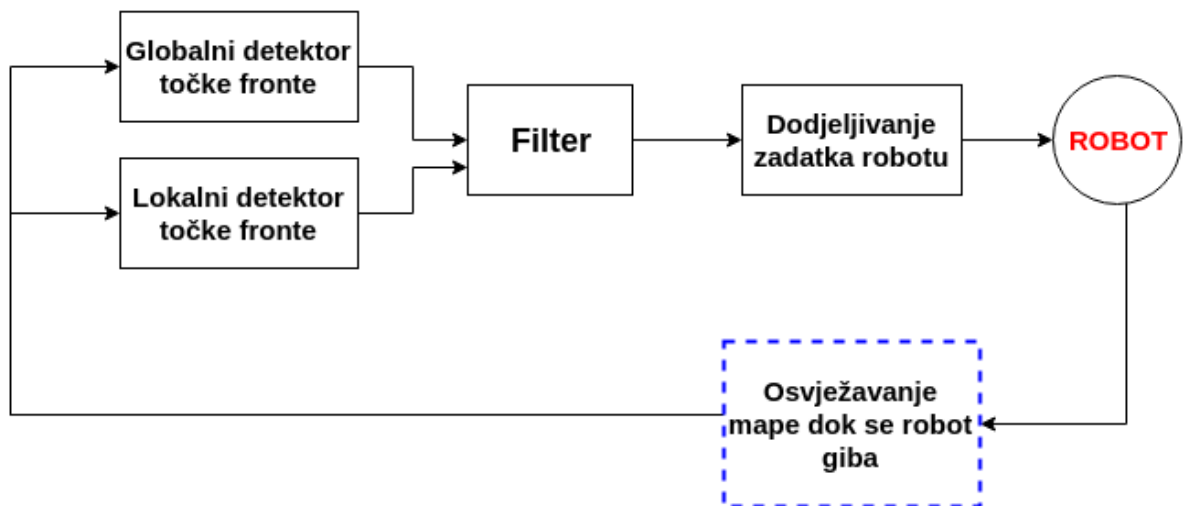
Slika 13. Određivanje najbližeg čvora RRT algoritmom

Kako se roboti kreću, istraženi prostor postaje sve veći. Na granici (fronti) neistraženog i istraženog prostora pronalaze se točke fronte. Iz skupa točaka fronte odabiru se centralne točke fronte te se dodjeljuju robotima kao ciljne točke.

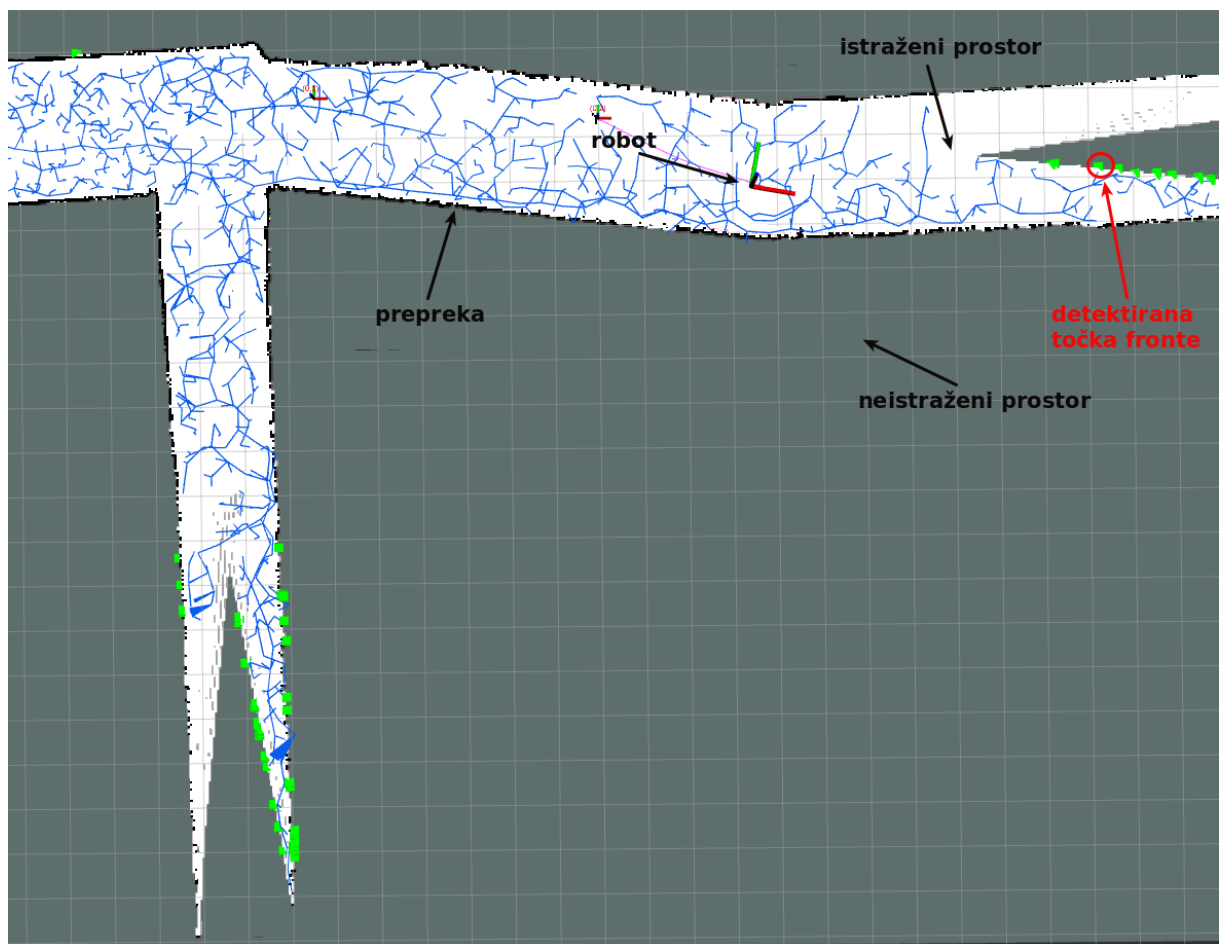
Implementacija algoritma za istraživanje prostora podijeljena je u tri ulančana modula: modul za detekciju točaka fronte temeljen na RRT-u (globalni i lokalni detektor točaka fronte), modul za filtriranje točaka fronte i modul za dodjeljivanje zadataka robotima. Blokovska shema algoritma prikazana je na slici 14.

Modul za detekciju točaka fronte vrši otkrivanje točaka fronte te ih šalje u modul za filtriranje. Moguća točka fronte je ona koja se nalazi na granici na kojoj je završilo širenje stabla, između poznatog (istraženog) i nepoznatog (neistraženog) dijela prostora. Vrijednosti iz mreža zauzetosti pripadajućih podkarti diskretizirane su u stanja "zauzeto" i "slobodno", uz posebno stanje "nepoznato" za ćelije za koje nema opažanja (slika 15.).

Detekcija točaka fronte odvija se na lokalnoj i globalnoj razini [11], kao što je prikazano na slici 16. Lokalna detekcija točaka fronte ima ulogu brze detekcije točaka fronte u blizini robota.



Slika 14. Blokovski prikaz implementacije algoritma za istraživanje prostora



Slika 15. Definicija dijelova prostora u RRT-u

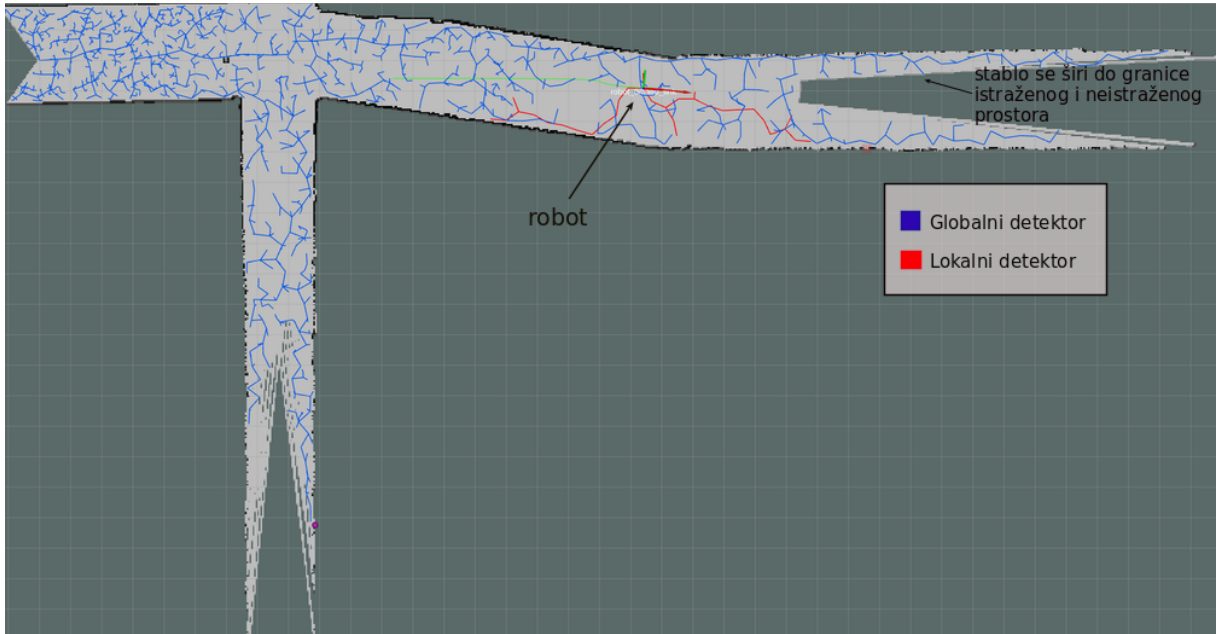
Lokalni detektor od svakog robot gradi stablo prema RRT algoritmu, a u trenutku kada stablo dosegne nepoznato područje, označava se detektirana točka fronte i stablo lokalnog detektora se briše. Cijeli proces se ponavlja za vrijeme gibanja robota. Nakon brisanja, stablo lokalnog detektora ponovno se gradi iz trenutnog položaja robota.

Globalni detektor točke fronte radi na istom principu kao i lokalni detektor, osim što se ne provodi brisanje stabla i ponovno započinjanje građenja stabla iz nove početne točke. Stablo globalnog detektora nastavlja rasti sve dok traje istraživanje prostora. Globalni detektor točaka fronte pronalazi točke na cijeloj karti uključujući i točke koje su udaljene od položaja robota.

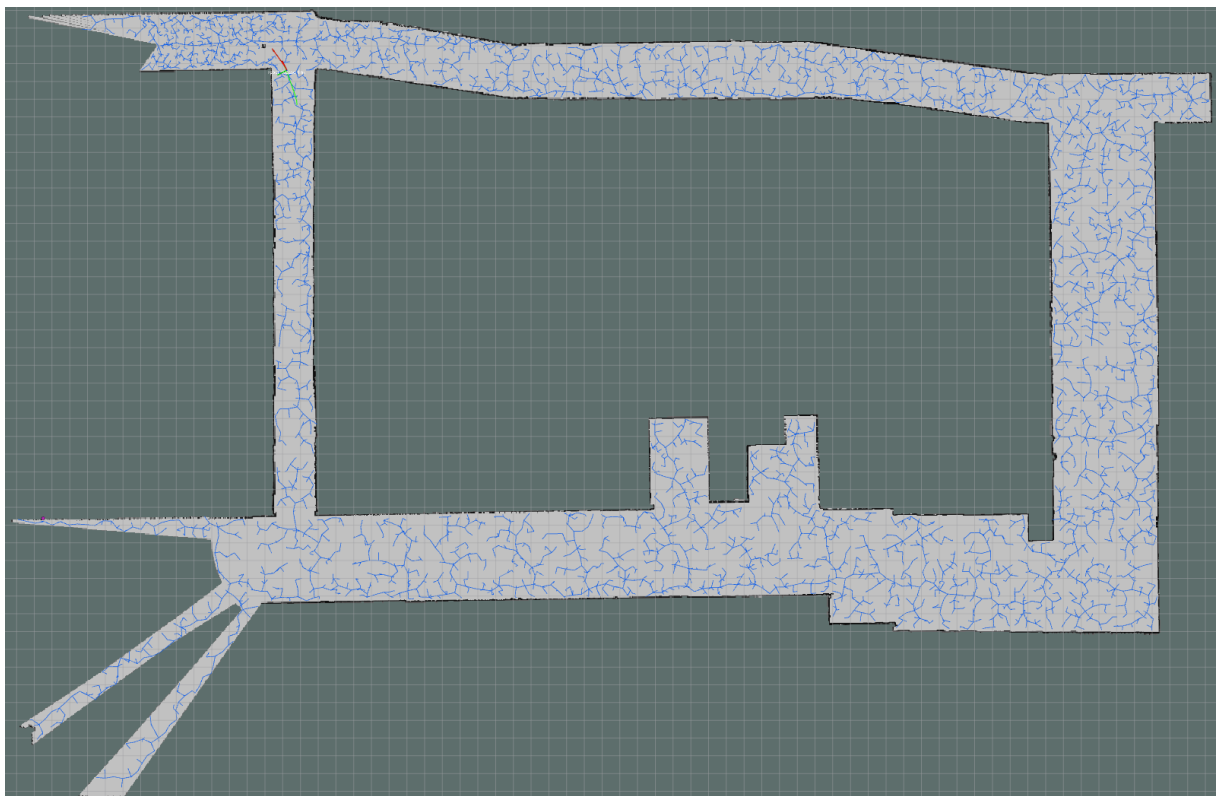
Modul za filtriranje dobiva kao ulazne podatke otkrivene točke fronte iz lokalnog i globalnog detektora. Provodi se postupak grupiranja točaka fronte metodom pomaka prema srednjoj vrijednosti (eng. *mean shift*), pri čemu se zadržavaju samo centri (centralne točke fronte ili centroidi) svake pojedine grupe, što ubrzava izvođenje algoritma. Modul za filtriranje briše stare točke fronte (točke u međuvremenu istraženom prostoru) u svakoj iteraciji.

Modul za dodjeljivanje zadataka dobiva kao ulazne podatke centralne točke fronte iz modula za filtriranje te ih dodjeljuje pojedinim robotima prema strategiji istraživanja. Detaljan postupak dodjeljivanja točaka fronte robotima opisan je u sljedećem poglavlju.

Slika 17. prikazuje globalno stablo izgrađeno tijekom istraživanja dijela prostora Fakulteta jednim robotom pomoću RRT algoritma.



Slika 16. Globalni i lokalni RRT



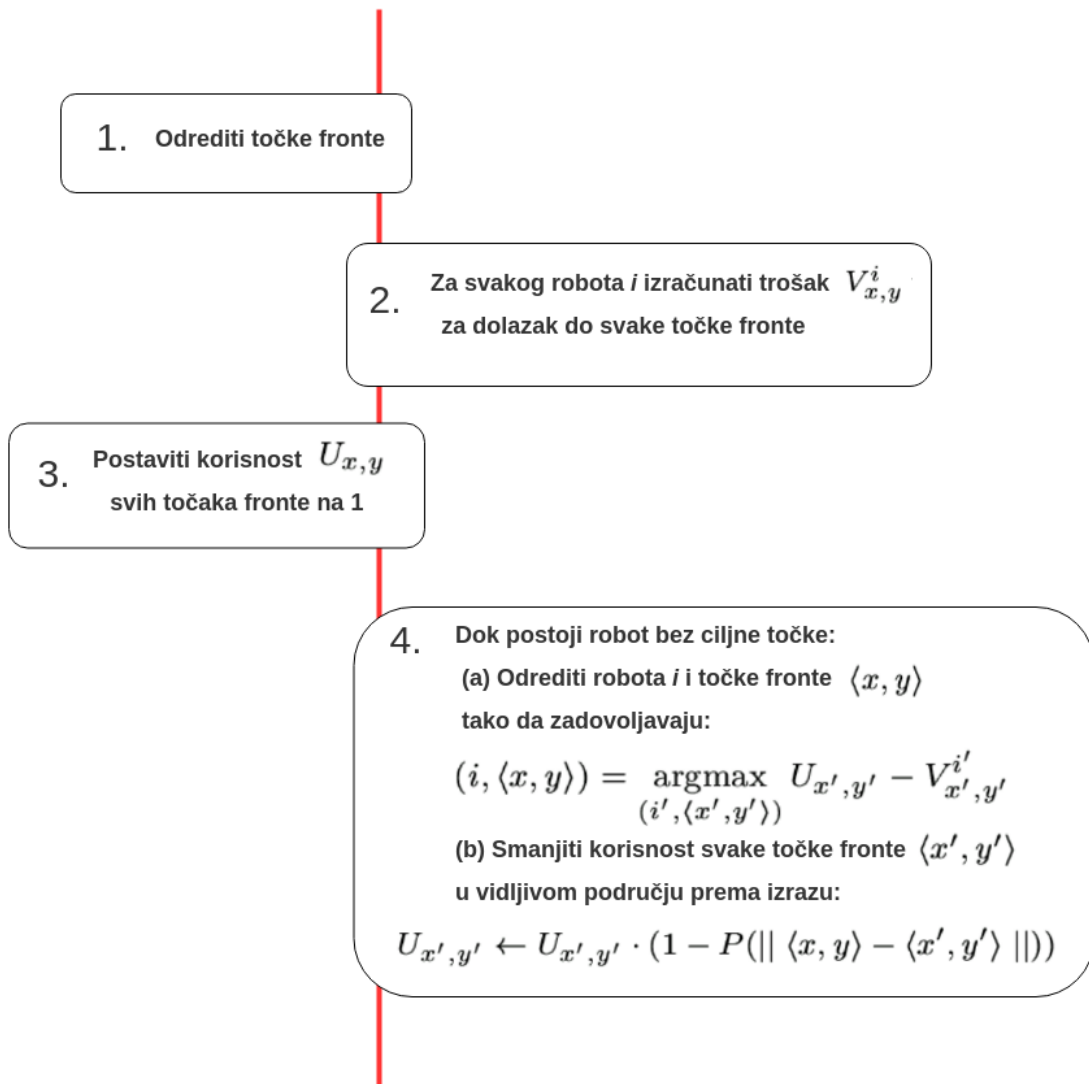
Slika 17. Izgrađeno globalno stablo tijekom istraživanja pomoću RRT-a

## 4.2. Strategija koordiniranog istraživanja prostora

Tijekom koordiniranog istraživanja nepoznatog prostora i gradnje zajedničke karte, koristi se centralizirani pristup upravljanja timom robota. Detektirane točke fronte na granici istraženog i neistraženog prostora su potencijalni ciljevi za navigaciju robota. Dodjeljivanje zadataka pojedinom robotu provodi se s pokušavajući minimizirati vrijeme potrebno za istraživanje cijelog prostora. Svakom robotu se dodjeljuje kao cilj točka fronte koja ima najveću korisnost  $U$  (eng. *information gain*) i najmanji trošak  $V$  [13]. *Korisnost* točke fronte je proporcionalna broju nepoznatih ćelija koje okružuju točku fronte u prethodno definiranom radijusu  $r$  prikazanom na slici 19. Pretpostavlja se da će robot opaziti sve nepoznate ćelije u okolini točke fronte nakon što dođe do nje. *Trošak* je definiran kao očekivana udaljenost koju bi robot trebao proći kako bi došao do ciljne točke fronte. Očekivana udaljenost se aproksimira euklidskom udaljenošću.

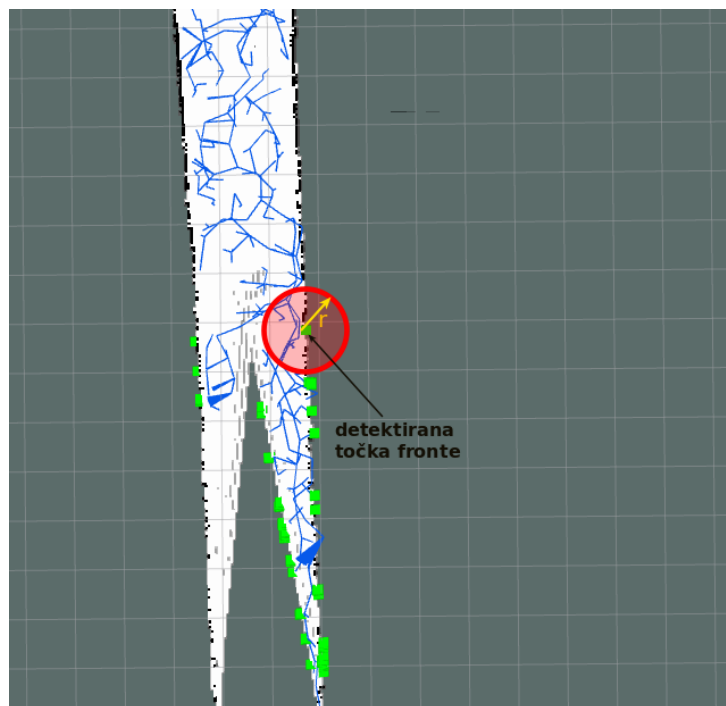
Algoritam koordiniranog upravljanja dan je na slici 18. U četvrtom se koraku algoritma odabire uređeni par robota i točke fronte za koji je maksimalna korisnost umanjena za trošak. Točka fronte iz odabranog uređenog para dodjeljuje se robotu s kojim čini uređeni par. Korisnost dodijeljene točke fronte, kao i vidljivih točaka fronte u blizini, umanjuje se kako bi se izbjeglo slanje više robota na isto mjesto.

Nakon što je robotu dodijeljena ciljna točka fronte, robot planira i prati putanju do ciljne točke, pri čemu se zatvara petlja sa SLAM-om, koji vrši lokalizaciju i izgradnju karte iz podataka s lasera i odometrije. Na slici 20. blokovski je prikazan redoslijed izvršavanja zadataka opisanih u ovom poglavlju.

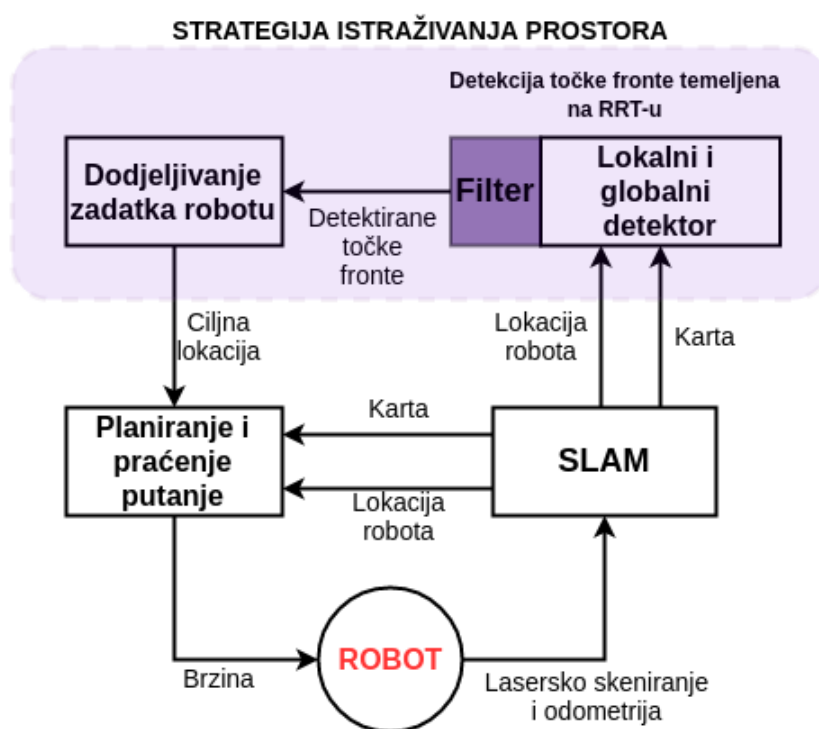


Slika 18. Algoritam koordiniranog upravljanja višerobotskim sustavom



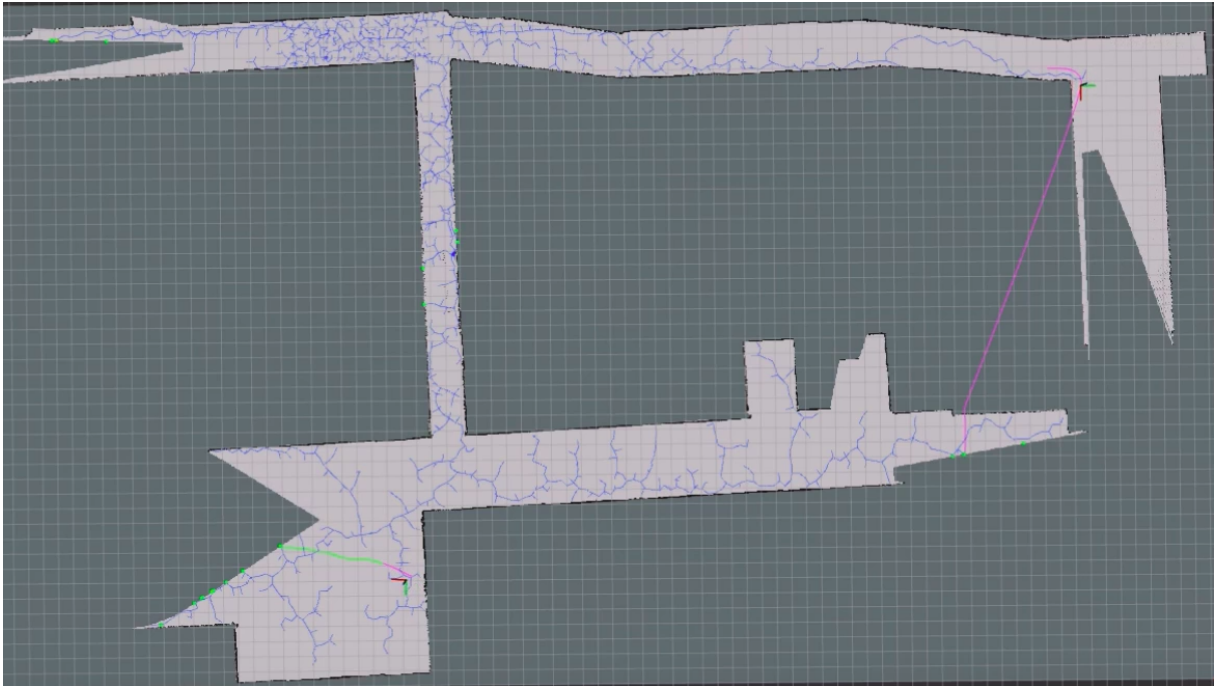


Slika 19. Detektirana točka fronte i radijus kojim je definirana



Slika 20. Blokovski prikaz sustava koji implementira strategiju istraživanja prostora

Strategija je ispitana pomoću dva robota, a rezultati su prikazani slikom 21. Na slici se može vidjeti RRT stablo, detektirane točke fronte te isplanirana putanja do ciljne točke fronte.

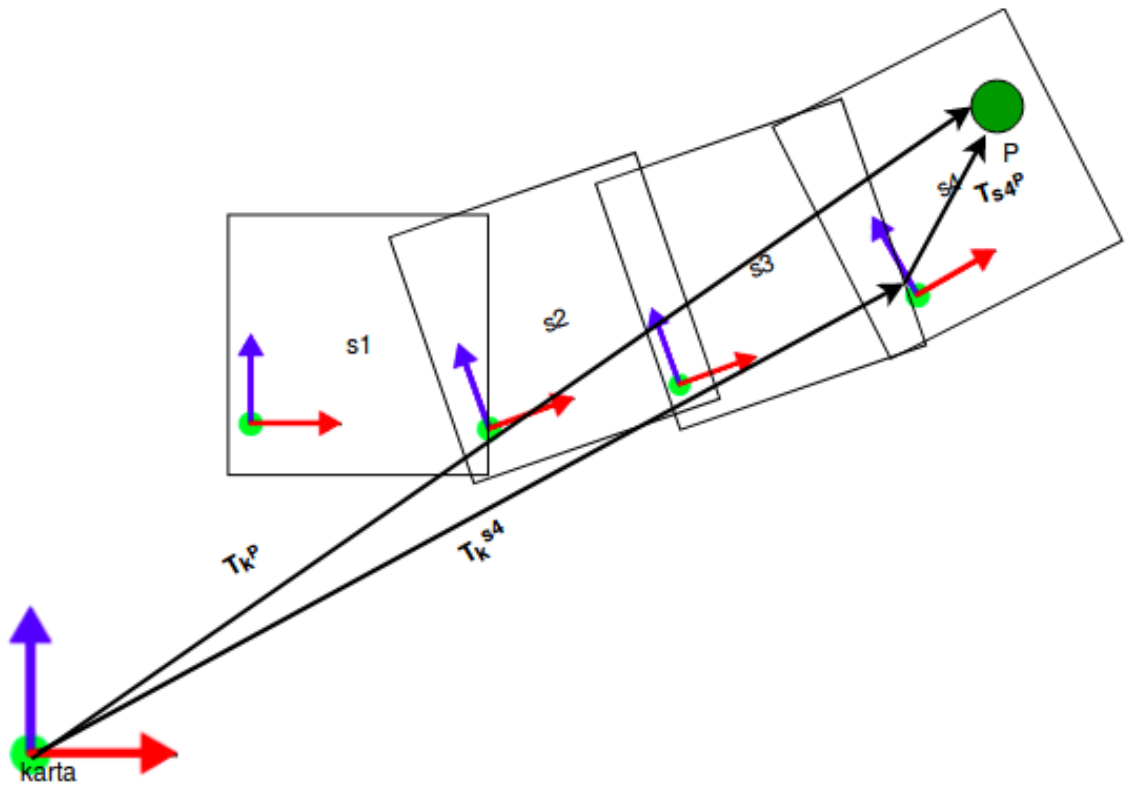


Slika 21. Dio prostora istražen pomoću dva robota

### 4.3. Transformacija stabla i točke fronte u koordinatni sustav karte

Metoda opisana u prethodnom poglavlju unaprijeđena je spajanjem izgrađenog stabla i detektiranih točaka fronte s podkartama kojima pripadaju. Na slici 22. prikazane su podkarte sa svojim ishodištima i točka unutar podkarte  $s_4$ . Spajanje podkarata i detektiranih točaka potrebno je zato što se prilikom SLAM postupka odvija zatvaranje petlje optimiziranjem poza podkarata i čvorova trajektorije. Zatvaranjem petlje pomiču se podkarte te se javila potreba za popratnim pomicanjem stabla i točaka fronte.

RRT algoritmom pronalaze se točke karte koje su na istraženom području i koje postaju dio stabla, ili su na granici istraženog i neistraženog te ih se proglašava točkama fronte. U originalnoj implementaciji [11], ovakvim točkama referentni sustav je globalni koordinatni sustav i njihova pozicija ne prati pomicanje podkarata prilikom zatvaranja petlje u SLAM-u. To može dovesti do situacije gdje spomenute točke mogu završiti u neistraženom području, što nepovoljno utječe na rad algoritma.



Slika 22. Transformacije točke P prema koordinatnom sustavu podkarte

Kako bi se izbjegao gore navedeni problem prilikom gradnje brzorastućeg stabla preko istraženog prostora karte, predlažemo pomicanje čvorova stabla i točaka fronte zajedno s podkartama kojima pripadaju.

Neka točka  $P$  predstavlja značajnu točku u RRT-u kao što je novoizgrađeni čvor stabla ili novodetektirana točka fronte. Točka  $P$  spaja se s najbližom podkartom  $s_i$  te se njena pozicija određuje fiksno prema koordinatnom sustavu podkarte, što omogućuje pomicanje točke  $P$  skupa s podkartom pomoću sljedećeg ulančavanja transformacija:

$$T_{karta}^P = T_{karta}^{s4} T_{s4}^P, \quad (24)$$

$$T_{s4}^P = (T_{karta}^{s4})^{-1} T_{karta}^P. \quad (25)$$

Na ovaj je način spojen *Graph SLAM* postupak s podkartama i algoritam istraživanja prostora temeljen na brzorastućem stablu.

## 5. Eksperimentalni rezultati

U ovom poglavlju opisano je korišteno programsko okruženje i predstavljeni rezultati u simulacijskom i stvarnom okruženju.

### 5.1. Programsko okruženje

Algoritme i njihovu učinkovitost potrebno je provjeriti u simulacijskom i stvarnom okruženju. Za implementaciju te provođenje eksperimentalne provjere kvalitete algoritma i strategije koordiniranog višerobotskog istraživanja prostora korišteno je ROS (eng. *Robot Operating System*) [14] okruženje na Linux Ubuntu 16.04 operacijskom sustavu.

#### 5.1.1. Robot Operating System (ROS)

ROS je fleksibilna razvojna cjelina (eng. *framework*) koja se koristi za razvoj robotskih aplikacija. ROS je skup alata, paketa i konvencija kojima je cilj pojednostaviti izradu kompleksnih i robusnih robotskih aplikacija na različitim platformama. To je sustav čije se funkcionalnosti neprestano razvijaju i proširuju.

Jedna od glavnih značajki ROS-a su gotove biblioteke za komunikaciju između pojedinih čvorova, odnosno dijelova sustava. Protokoli komunikacije su skriveni od korisnika. Time se pojednostavljuje njihovo korištenje pri izradi aplikacija te je korisniku omogućeno brzo i jednostavno povezati dijelove jednostavnih i kompliciranih sustava.

ROS je sustav koji, svojim karakteristikama, omogućuje povezivanje s poznatim realističnim simulatorima s cjelovitom podrškom kao što su Stage [15] za 2D okruženje te Gazebo [16] za 3D okruženja.

U radu je korištena verzija ROS Kinetic.

#### 5.1.2. Google Cartographer

Google Cartographer je moderna SLAM implementacija temeljena na *Graph SLAM* sustavu koji omogućuje istovremenu lokalizaciju i izgradnju karte u stvarnom vremenu u 2D i 3D prostoru.

SLAM algoritmi kombiniraju podatke različitih senzora kako bi istovremeno pronašli lokaciju jednog ili više senzora i izgradili kartu njegove okoline. Konačna karta prostora dobije se spajanjem podkarata u jednu jedinstvenu kartu.

Google Cartographer je implementiran unutar ROS-a omogućujući tako jednostavno instaliranje i pokretanje alata na vlastitim računalima te na mobilnim robotima [17]. Načini korištenja Google Cartographera, 2D simulatora i stvarnih robota u svrhu koordiniranog višerobotskog istraživanja prostora opisani su u nastavku rada.

### 5.1.3. Ceres Solver

Ceres Solver je nelinearni optimizator otvorenog koda napisan u C++-u [18]. Ceres Solver može riješiti dvije vrste problema:

- nelinearne probleme najmanjih kvadrata s definiranim ograničenjima
- općenite probleme koji nemaju definirana ograničenja

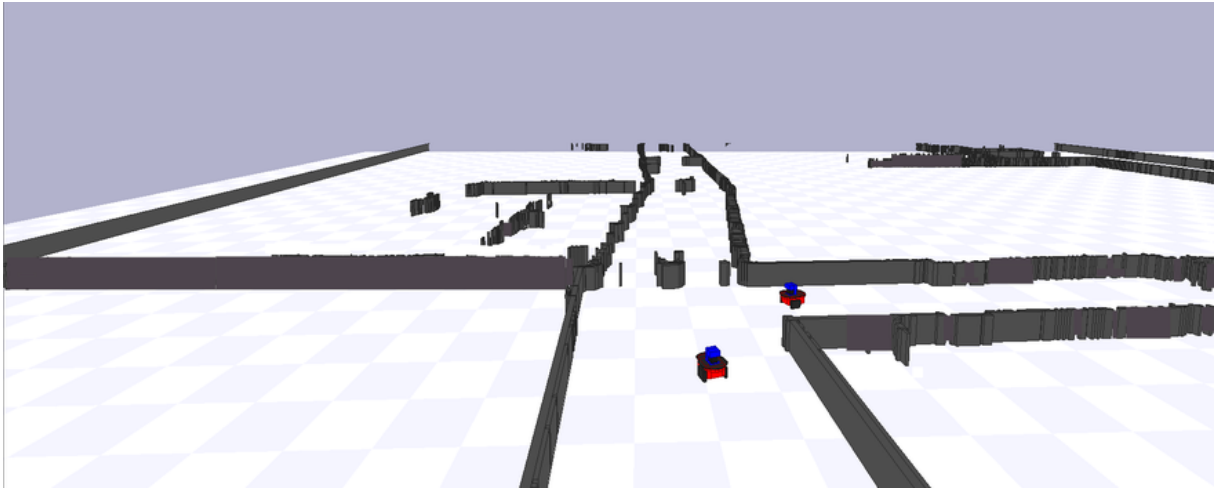
Google Cartographer koristi Ceres Solver za rješavanje *Graph SLAM* optimizacijskog problema (21).

### 5.1.4. 2D simulator Stage

Google Cartographer može biti korišten u 2D i 3D prostoru. Iako Gazebo nudi mogućnost precizne i učinkovite simulacije jednog ili više različitih vrsta robota u složenim unutarnjim i vanjskim okruženjima, zbog brzine izvođenja algoritma simulacija je obavljena u 2D simulatoru Stage. Stage simulator je jednostavan za korištenje. Pokretanjem Stage simulatora otvara se karta željenog prostora s dodanim mobilnim robotima. Slika 23. prikazuje Stage simulatora s dva mobilna robota.

Koriste se sljedeći ROS čvorovi:

- *stage\_ros* - pokretanje okruženja u Stage simulatoru
- *p2os\_driver* i *sicklms\_node* - potpora za pogon robota i čitanje laserske slike
- *teleop\_node* - izračun referentne brzine robota iz očitavanja s upravljačke palice



Slika 23. Prikaz okruženja u Stage simulatoru

Stage simulira mjerenja i upravljačko sučelje kao prema stvarnim robotima. Za vizualizaciju podataka s robota, izgrađene karte i ostalih podataka vezanih uz strategiju istraživanja kao što su detektirane točke fronte i isplanirane putanje, koristi se ROS-ov vizualizacijski alat RViz. .

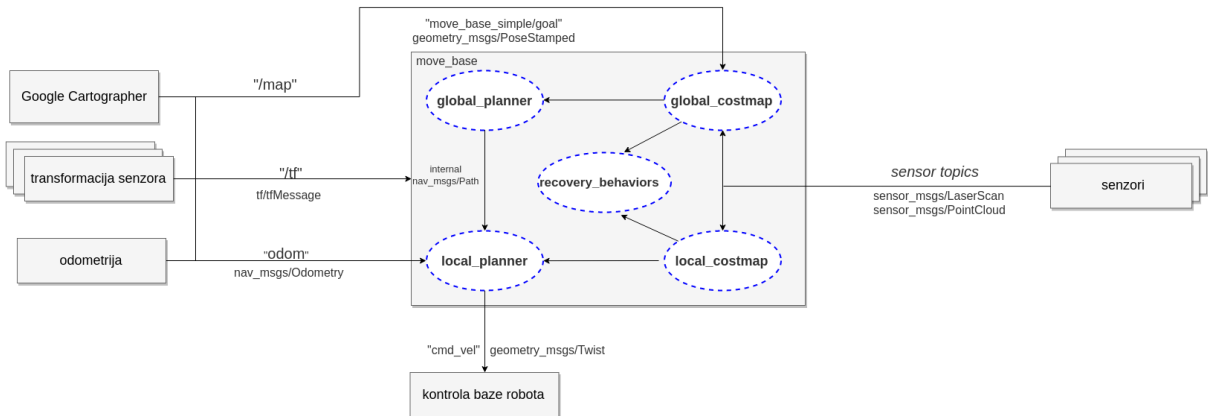
## 5.2. Planiranje i praćenje putanje uz ROS navigacijski slog

Planiranje putanje omogućava autonomnim mobilnim robotima pronalazak najkraćeg ili optimalnog puta. Algoritmi za planiranje putanje temelje se na izračunu akumuliranog troška od početne do ciljne pozicije.

ROS navigacijski slog (eng. *navigation stack*) je paket koji na temelju lokalizacije iz SLAM-a, senzora i karte računa optimalnu putanju za robota prema zadanom cilju te osigurava praćenje putanje slanjem odgovarajuće reference za brzinu kotača robota [19]. Navigacijski slog sadrži implementacije algoritama za planiranje putanje kao što su A\*, Dijkstra i dr. Navigacijski slog navodi robota do zadane ciljne pozicije u prostoru, pritom izbjegavajući statičke (vidljive na karti) i dinamičke (ljudi i sl.) prepreke [20].

### 5.2.1. Struktura

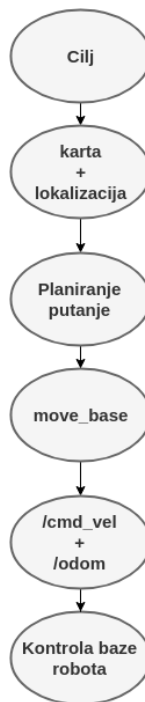
Na slici 24. prikazana je struktura ROS navigacijskog sloga s korištenim čvorovima te tipovima poruka potrebnima za njihovu međusobnu komunikaciju [19].



Slika 24. ROS navigacijski slog

Navigacijski slog koristi informacije sa senzora za izbjegavanje prepreka na koje robot može naići gibajući se u prostoru.

Google Cartographer stvara kartu koja se koristi za planiranje putanje slanjem u čvor navigacijskog sloga *move\_base* koji obuhvaća tzv. globalni i lokalni planer. Osnovni koraci prikazani su na slici 25.



Slika 25. Glavni koraci u ROS navigacijskom slogu

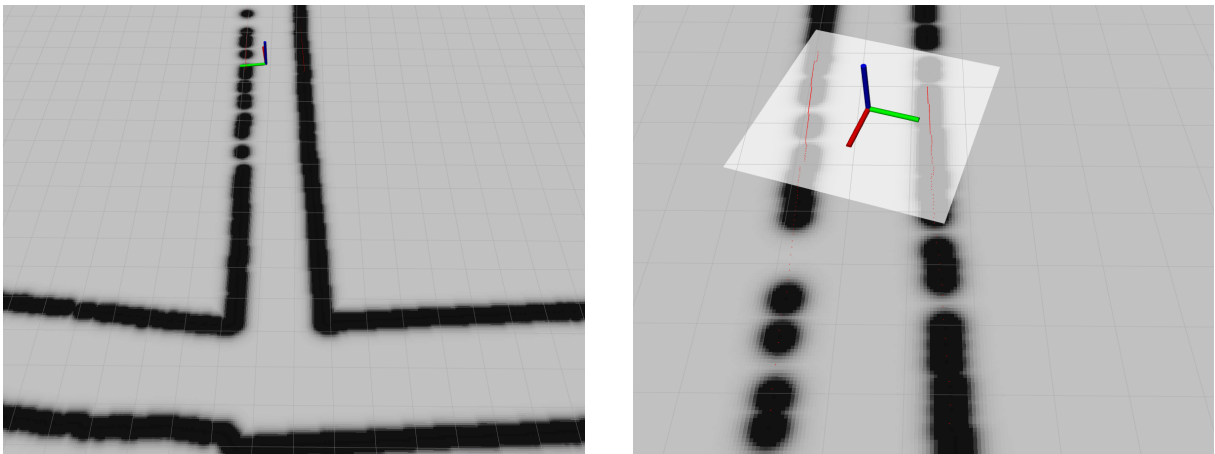
### 5.2.2. Upravljanje robotom

Robot se giba kartom koristeći dvije vrste navigacije, globalnu i lokalnu. Globalna navigacija koristi se za stvaranje putanja prema cilju na većim udaljenostima, a lokalna služi za generiranje putanja u neposrednoj blizini robota te za izbjegavanje prepreka.

Karta troška (eng. *costmap*) je struktura podataka koja predstavlja mjesta (ćelije) u rešetki koja su sigurna za robota. Sigurna mjesta su definirana kao ona na kojima nije prepreka. Karta troška zasnovana je na mrežastoj karti zauzeća i radijusu inflacije funkcije troška, koji je unaprijed definiran od strane korisnika. Unutar područja kraj prepreke definiranog inflacijskim radijusom, trošak ćelije se povećava funkcijom troška. Postoje dvije vrste karte troška u ROS-u:

- globalna karta troška koja se koristi u globalnoj navigaciji
- lokalna karta troška koja služi za lokalnu navigaciju

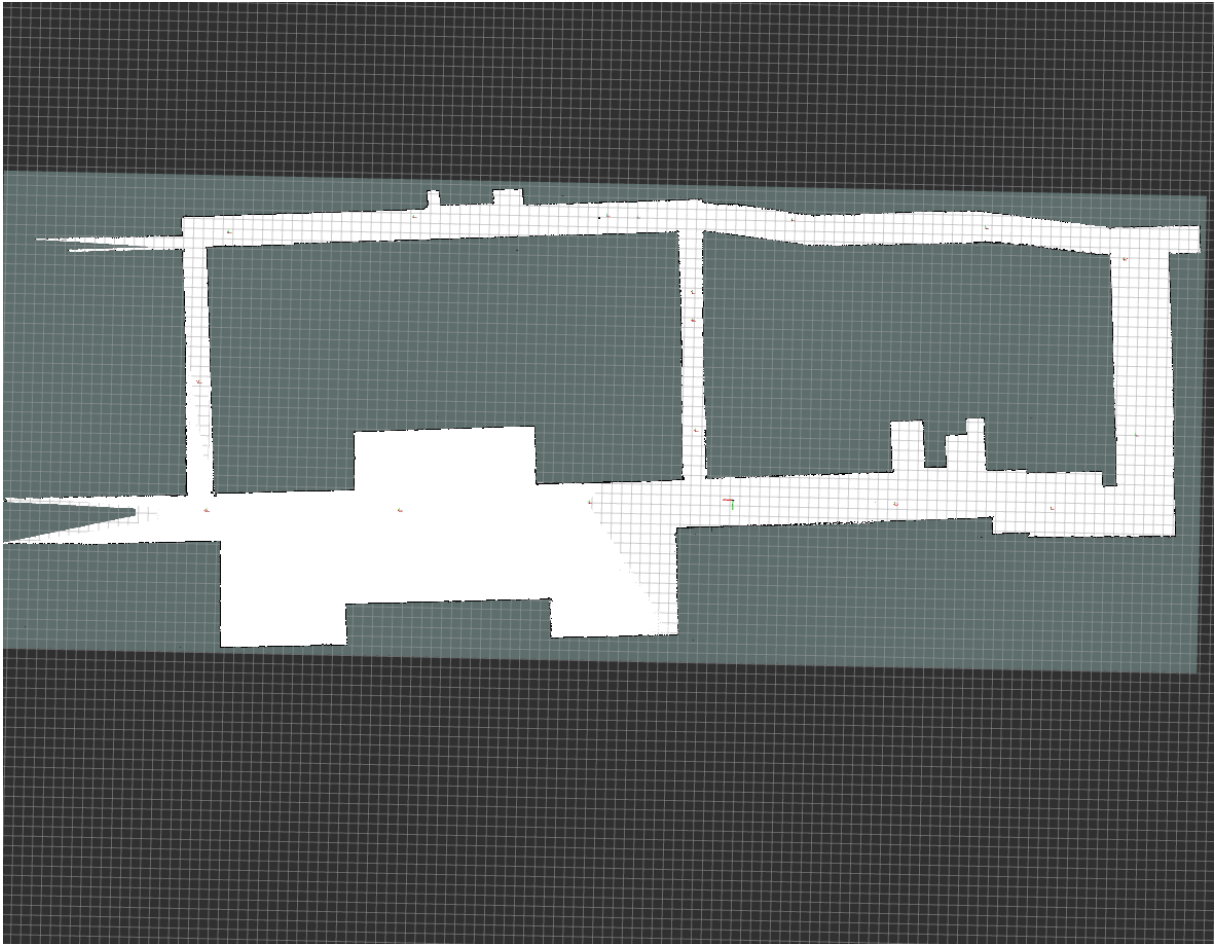
Obje karte prikazane su na slici 26.



Slika 26. Globalna i lokalna karta troška

Navigacijski se cilj šalje robotu, navigacijski slog računa putanju, a lokalni planer osigurava praćenje putanje temeljem podataka dobivenih od robota. Navigacijski slog ispitan je s Cartographerovom lokalizacijom u povratnoj vezi na dijelu prostora Fakulteta koristeći jednog robota. Rezultat je prikazan na slici 27.





Slika 27. Izgrađena karta prostora FER-a koristeći jednog robota i ručno zadavanje cilja

### **5.3. Izgradnja karte prostora i lokalizacija na stvarnom robotu**

U ovom dijelu opisan je mobilni robot Pioneer P3-DX, primijenjen je Google Cartographer za izgradnju karte prostora i lokalizaciju te prikazani rezultati korištenja robota Pioneer.

#### **5.3.1. Mobilni robot Pioneer P3-DX**

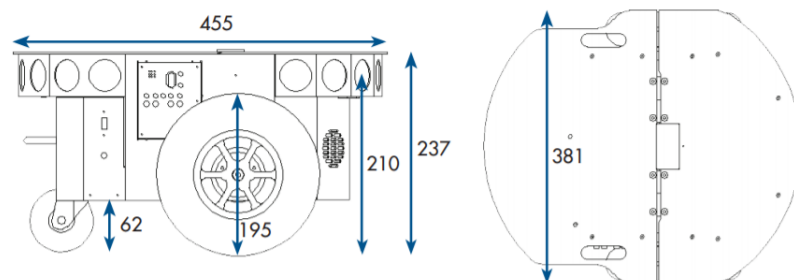
Tijekom rada korišten je mobilni robot Pioneer P3-DX prikazan na slici 28. Naziv Pioneer odnosi se na obitelj mobilnih robota s tri ili četiri kotača te sadrži sve osnovne komponente potrebne za mjerenje i navigaciju kroz okolinu. Svi roboti iz Pioneer serije sadrže motore, enkodere pozicije i brzine, baterijski paket, integrirane senzore te imaju mogućnost nadogradnje raznim dodacima. Svim komponentama upravlja integrirani mikrokontroler baziran na Siemens C166 mikrokontroleru [21]. Upravljački algoritmi implementirani su na platformi Pioneer

P3-DX s kojom su isti i provjereni u stvarnom okruženju. Korišteni su mobilni roboti Pioneer P3-DX opremljeni laserskim senzorom udaljenosti SICK LMS-100. Dimenzije mobilnog robota



Slika 28. Mobilni robot Pioneer P3-DX

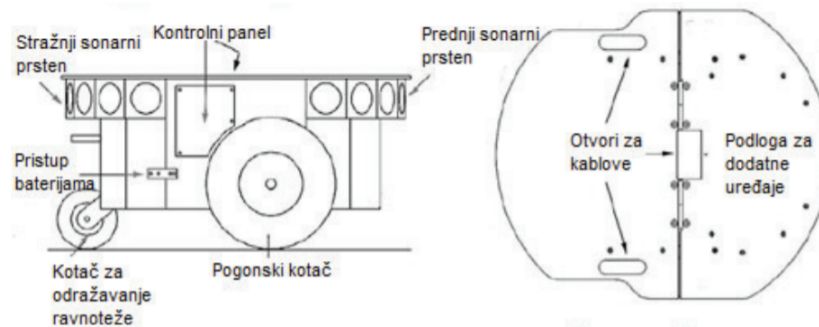
Pioneer P3-DX iskazane u milimetrima, tlocrt i bokocrt prikazani su na slici 29.



Slika 29. Dimenzije Pioneera P3-DX

Glavne komponente od kojih je sastavljen robot su niz sonarnih senzora, kotači (pogonski i za održavanje ravnoteže), baterija i sučelje za napajanje električnom energijom, površina s

korisničkim sučeljem te motori i enkoderi. Ove komponente i još nekoliko sporednih prikazano je na slici 30.



Slika 30. Komponente Pioneera P3-DX

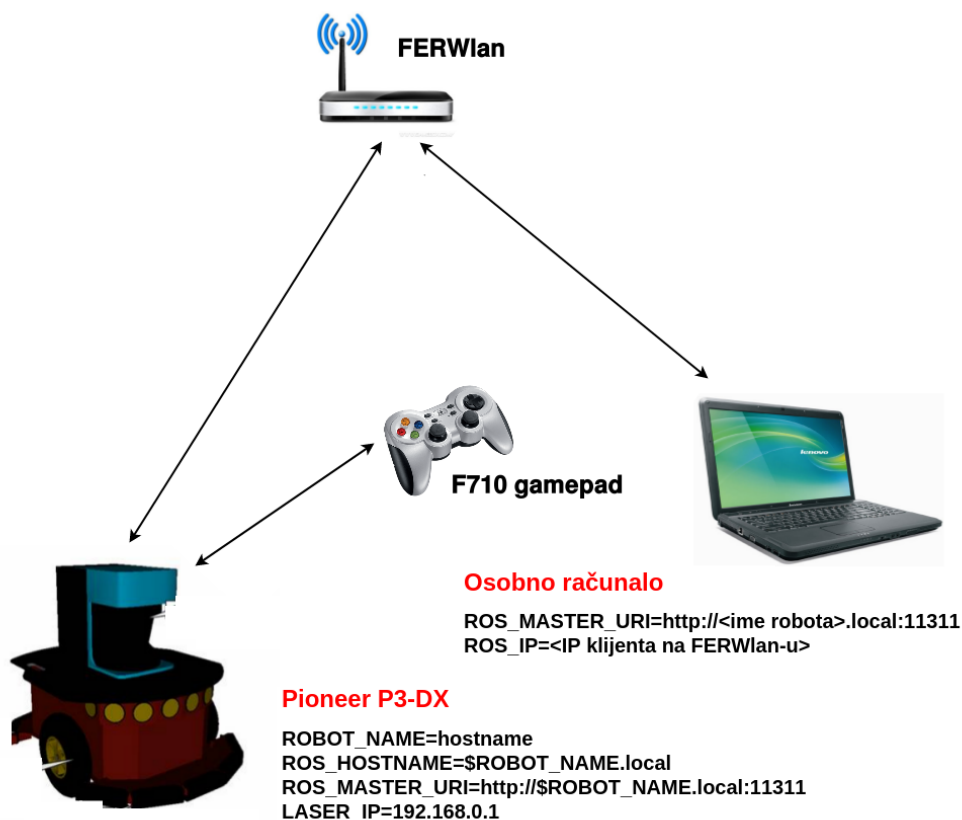
### 5.3.2. Povezivanje mobilnog robota i osobnog računala

Računalo je bežično spojeno s mobilnim robotom koristeći bežičnu mrežu Fakulteta elektrotehnike i računarstva FERwlan, koja je dostupna na prostorima FER-a. Način spajanja osobnog računala i Pioneera na FERwlan prikazan je na slici 31. Za komunikaciju koriste se postavke kao na slici. Upravljačka palica (eng. *joystick*) povezana je na mobilni robot direktno, bežičnom vezom.

### 5.3.3. Rezultati uz *offline* čvor

Postoje dva načina na koje Google Cartographer može prikupljati podatke s mobilnog robota: *online* i *offline* način. Za razliku od *online* načina, koji u stvarnom vremenu prikuplja podatke s robota, *offline* način čita podatke iz spremljene datoteke. *Offline* čvor korišten je s ciljem testiranja SLAM-a i podešavanja parametara. Dva mobilna robota upravljana su upravljačkom palicom te su prikupljena senzorska i odometrijska mjerenja snimljena u datoteku koja je naknadno obrađena pomoću *offline* čvora na računalu. Rezultati izgrađene karte prizemlja B i C zgrade FER-a dani su na slici 32.

Plavom i zelenom bojom prikazane su trajektorije robota ručno vođenih upravljačkom palicom.



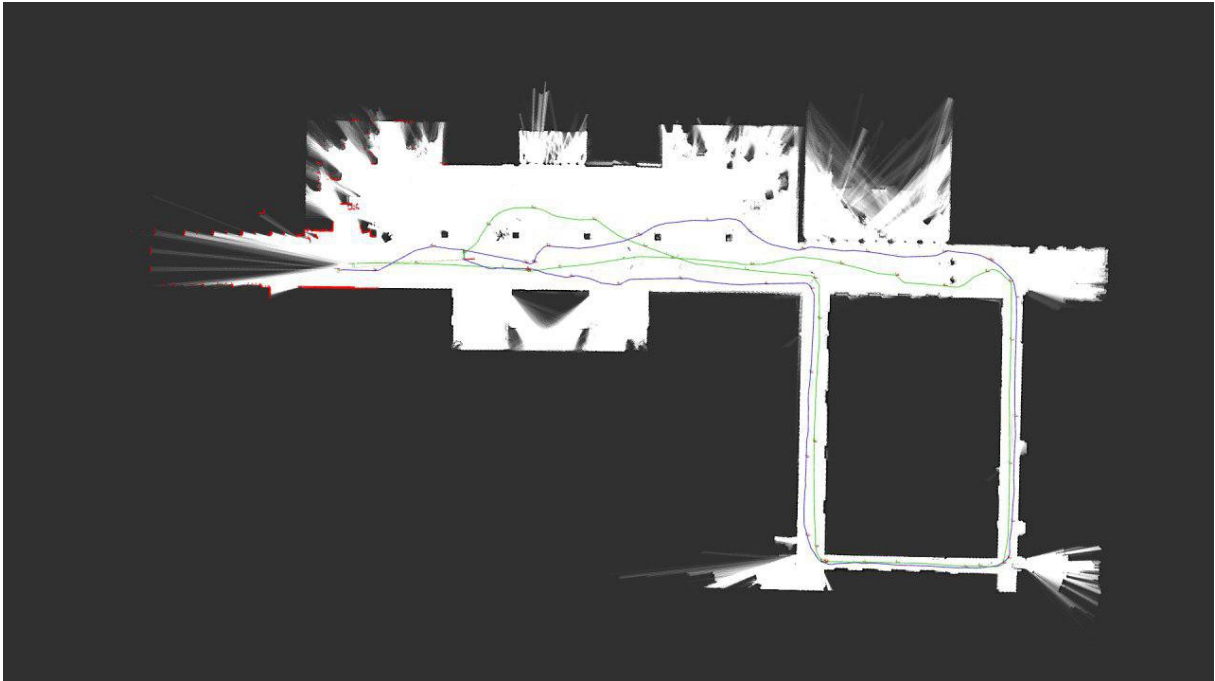
Slika 31. Spajanje osobnog računala i mobilnog robota na FERwlan

## 5.4. Koordinirano višerobotsko istraživanje prostora na stvarnom robotu

U ovom dijelu rada objašnjen je način uspostave komunikacije između robota i računala kako bi se ostvarila prethodno opisana strategija istraživanja. Komunikacija je ispitana na robotima u stvarnom okruženju te je prikazano koordinirano istraživanje prostora Fakulteta pomoću dva Pioneer robota.

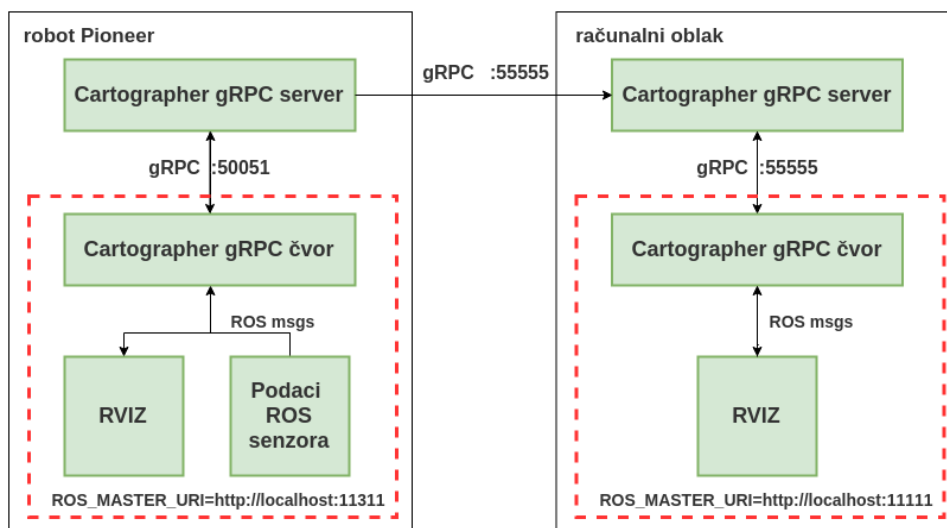
### 5.4.1. Komunikacija centralnog računala i robota

Budući da je cilj koordinirano upravljati višerobotskim sustavom, potrebno je implementirati i ostvariti komunikaciju između računala i odabranog broja robota. To jest, treba povezati jedno računalo s više robota, kako bi međusobno prenosili potrebne informacije. Za njihovu što robusniju komunikaciju koristi se kombinacija gRPC protokola [22] i paketa *nimbro\_network* [23] za prenošenje ROS tema između različitih mastera (tzv. *multi-master*). Kako bi se ostvarila



Slika 32. Izgrađena karta dijela prostora FER-a pomoću dva robota uz *offline* čvor

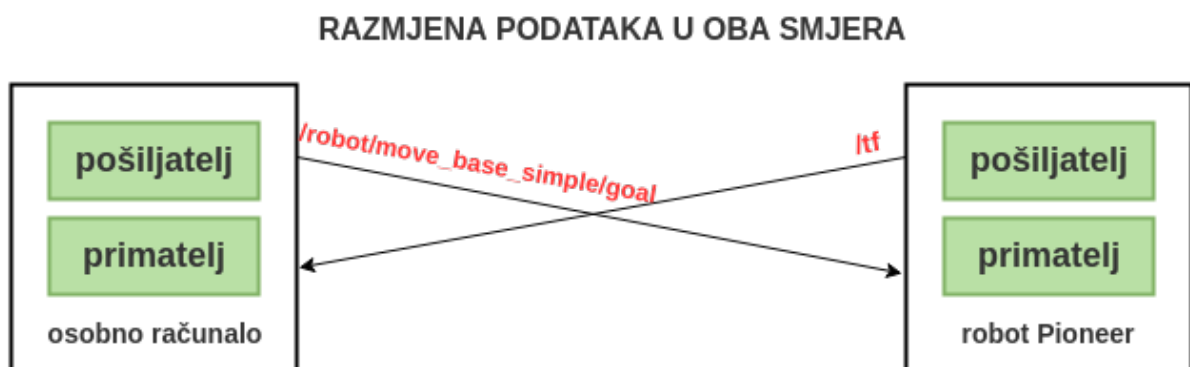
brza lokalizacija u povratnoj vezi prema navigacijskom slogu, na svakom je robotu pokrenut Cartographer server koji izvodi lokalni SLAM. Nadalje, serveri pokrenuti na robotima uspostavljaju vezu prema nadređenom centralnom serveru, odnosno računalnom oblaku putem bežične veze FERwlan. Na svakom od robota te na računalnom oblaku pokrenuti su gRPC čvorovi koji predstavljaju "mostove" prema ROS-u (slika 33.).



Slika 33. Komunikacija robota i računala uz gRPC

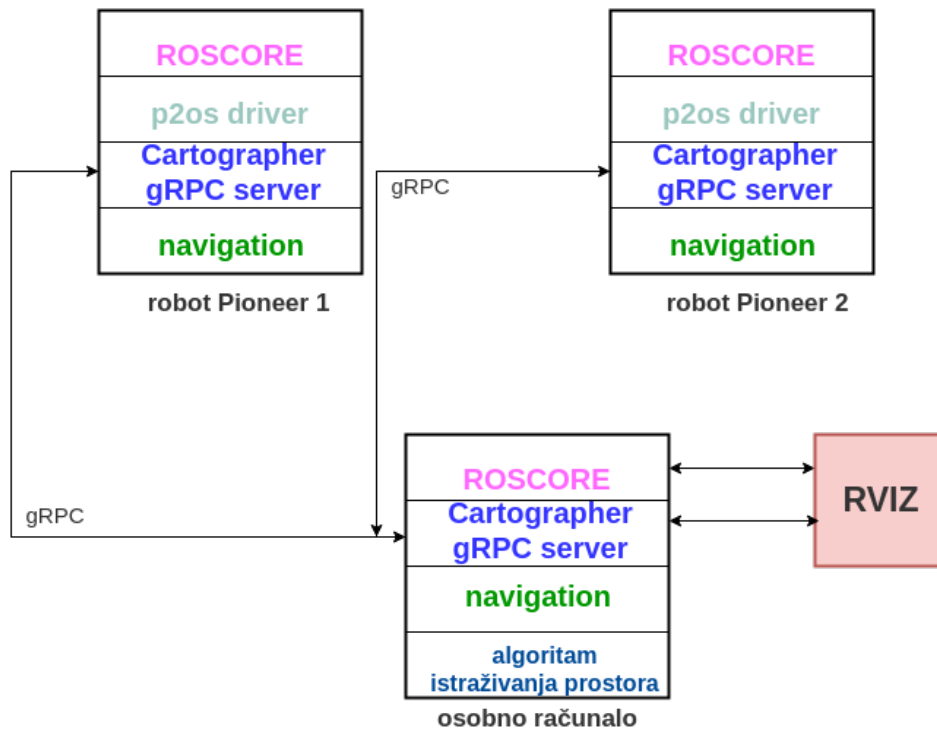
Kada se uspostavi veza prema nadređenom serveru, moguće je u centralnom RViz-u pokrenutom u računalnom oblaku vidjeti informacije primljene s robota.

Za prijenos podataka putem ROS tema od robota do računala koristi se *nimbro\_network*, skup ROS paketa koji služe za robustan prijenos željenih informacija između više ROS mastera. Prednost načina komunikacije putem *nimbro\_networka* je robusnost na prekide u vezi. Informacije se prenose s robota na računalo, gdje se pomoću algoritma istraživanja prostora određuju točke fronte i dodjeljuju ciljevi pojedinom robotu. Naposljetku se ciljna točka šalje robotu kao novi cilj za navigaciju. Razmjena podataka pomoću *nimbro\_network-a* prikazana je na slici 34.



Slika 34. Razmjena podataka pomoću *nimbro\_network* servisa

Nakon uspostavljanja komunikacije (slika 35.), roboti započinju istraživanje prostora vođeni centralnim računalom.



Slika 35. Komunikacija dvaju robota s centralnim računalom

#### 5.4.2. Eksperiment u stvarnom okruženju

Nakon ispitivanja sustava u simulaciji i realizacije komunikacije za neometano razmjenjivanje potrebnih informacija, mobilni roboti su postavljeni u prizemlje Fakulteta (slika 36. i slika 37.). Korištena su dva mobilna robota Pioneer, a strategija i upravljački algoritmi jednaki su kao u simulacijskom eksperimentu.





Slika 36. Eksperiment u auli Fakulteta





Slika 37. Eksperiment na hodniku Fakulteta

Provedenim eksperimentom potvrđena je uspješna realizacija koordiniranog višerobotskog istraživanja prostora. U [24] se nalazi implementacija rada, a video snimka rezultata eksperimentalnog rada može se pogledati na Youtube servisu [25].

## 6. Zaključak

U ovom radu ostvareno je koordinirano višerobotsko istraživanje nepoznatog prostora pomoću mobilnih robota s diferencijalnim pogonom u što kraćem vremenu.

Za istovremenu lokalizaciju i izgradnju karte prostora korišten je sustav Google Cartographer temeljen na *Graph SLAM* pristupu. Konačni izgled karte prostora dobiven je spajanjem podkarata u jednu jedinstvenu kartu. Nastala karta korištena je za planiranje putanje robota. Upravljanje višerobotskim sustavom je centralizirano, a strategija autonomnog istraživanja prostora, koja usmjerava robote prema granicama istraženog i neistraženog prostora, temeljena je na algoritmu brzorastućeg slučajnog stabla i maksimizaciji korisnosti točaka fronte. Postojeća implementacija detekcije točaka fronte pomoću RRT-a nadograđena je pomicanjem čvorova brzorastućeg stabla i detektiranih točaka fronte zajedno s pripadajućim podkartama iz Google Cartographer SLAM-a. Strategija istraživanja prostora uključuje lokalnu i globalnu detekciju točaka fronte, filtriranje te dodjeljivanje točaka fronte robotima kao ciljeva za navigaciju. Dodjeljivanje ciljne točke fronte pojedinom robotu provodilo se prema strategiji koja minimizira vrijeme potrebno za istraživanje cijelog prostora. Planiranje i praćenje putanje prema ciljnim točkama provodi se pomoću ROS navigacijskog sloga.

Algoritmi i strategije ispitani su simulacijski uz *online* način rada Google Cartographera te su rezultirali uspješnim istraživanjem okoline. Osim simulacijski, jednako uspješan rezultat dali su i mobilni roboti Pioneer P3-DX koji su u stvarnom okruženju autonomno i koordinirano istražili prizemlje Fakulteta elektrotehnike i računarstva u Zagrebu.

## **Zahvala**

Zahvaljujemo se mentoru prof. dr. sc. Stjepanu Bogdanu za nesebičnu pomoć koju nam je pružio tijekom izrade ovog rada. Zahvaljujemo dr. sc. Damjanu Mikliču na savjetima i pomoći pri izradi rada. Veliku zahvalnost dugujemo Juraju Oršuliću, mag. ing. koji je uložio ogroman trud i izdvojio puno vremena kako bi nam olakšao izradu rada.

## Literatura

- [1] Z. Kovačić, S. Bogdan, V. Krajči. *Osnove robotike*. Graphis, Zagreb, 2002.
- [2] Pioneer 3-DX. <http://www.mobilerobots.com>, 3. ožujka 2018.
- [3] I. Petrović. *Mobilna robotika - predavanja*. Mobilna robotika, Fakultet elektrotehnike i računarstva, Zagreb 2017.
- [4] R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series)*. The MIT Press, 2011.
- [5] S. Thrun, W. Burgard, D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*. The MIT Press, 2005.
- [6] G. Grisetti, R. Kümmerle, C. Stachniss, W. Burgard. A tutorial on graph-based slam. 2:31–43, 12 2010.
- [7] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, R. Vincent. Efficient sparse pose adjustment for 2d mapping. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [8] Google Cartographer. <https://github.com/googlecartographer/cartographer>, 29. studenog 2017.
- [9] W. Hess, D. Kohler, H. Rapp, D. Andor. Real-time loop closure in 2d LIDAR SLAM. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, svibanj 2016.
- [10] M. Juliá, A. Gil, O. Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33(4):427–444, svibanj 2012.
- [11] H. Umari S. Mukhopadhyay. Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1396–1402, rujana 2017.

- [12] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. *Computer Science Department, Iowa State University*, listopad 1998.
- [13] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, , H. Younes. Coordination for multi-robot exploration and mapping. page 852–858, 2000.
- [14] H. Giang. The robot operating system. <http://wiki.ros.org>, 3. ožujka 2018.
- [15] R. Vaughan. Stage simulator. <http://rtv.github.io/Stage/authors.html>, 3. ožujka 2018.
- [16] Gazebo simulator. <http://gazebosim.org>, 3. ožujka 2018.
- [17] Google Cartographer ROS. [https://github.com/googlecartographer/cartographer\\_ros](https://github.com/googlecartographer/cartographer_ros), 29. studenog 2017.
- [18] Ceres Solver. <https://github.com/ceres-solver/ceres-solver>, 19. ožujka 2018.
- [19] Navigation. <http://wiki.ros.org/navigation/Tutorials/Robot>, 25. ožujka 2018.
- [20] J. Lentin. *Mastering ROS for Robotics Programming*. Packt Publishing, prosinac 2015.
- [21] ActivMedia Robotics. Pioneer 3 operations manual, siječanj 2006. [http://www.Courses/Robotics/manual\\_pioneer.pdf](http://www.Courses/Robotics/manual_pioneer.pdf), 3. ožujka 2018.
- [22] gRPC. <https://github.com/grpc/grpc>, 5. travnja 2018.
- [23] M.Schwarz. Nimbro\_network. [https://github.com/AIS-Bonn/nimbro\\_network](https://github.com/AIS-Bonn/nimbro_network), 5. travnja 2018.
- [24] A. Batinović, I. Jambrečić. Repozitorij projekta za Rektorovu nagradu. <https://github.com/ivonaj/exploration>, 25. ožujka 2018.
- [25] A. Batinović, I. Jambrečić. Snimke rezultata projekta za Rektorovu nagradu. [https://www.youtube.com/playlist?list=PLiPLN9-Y4CIBlrH1kEyQ4\\_3AFgY2lq6ZO](https://www.youtube.com/playlist?list=PLiPLN9-Y4CIBlrH1kEyQ4_3AFgY2lq6ZO), 3. svibnja 2018.
- [26] Rapidly exploring random tree. <https://en.wikipedia.org/wiki/Rapidly-exploring-random-tree>, 19. ožujka 2018.

[27] Multi-robot Systems. [https://people.csail.mit.edu/jamesm/project-MultiRobotSystems Engineering.php](https://people.csail.mit.edu/jamesm/project-MultiRobotSystemsEngineering.php), 25. ožujka 2018.

## Sažetak

Autori: Ana Batinović, Ivona Jambrečić

Ime rada: Koordinirano višerobotsko istraživanje prostora zasnovano na *Graph SLAM* metodi i brzorastućim slučajnim stablima

U radu je implementirano koordinirano višerobotsko istraživanje prostora temeljeno na *Graph SLAM* metodi i brzorastućim slučajnim stablima uz korištenje mobilnih robota Pioneer P3-DX. Algoritam istraživanja prostora uključuje detekciju točaka fronte temeljenu na RRT algoritmu i nadograđenu za *Graph SLAM*, filtriranje detektiranih točaka i dodjeljivanje ciljnih točaka robotima koristeći strategiju koja minimizira vrijeme potrebno za istraživanje prostora. ROS navigacijski slog koristi se za planiranje i praćenje putanje. Eksperimenti su provedeni u simulacijskom i stvarnom okruženju.

Ključne riječi: mobilni robot, višerobotsko istraživanje prostora, točke fronte

## Summary

Authors: Ana Batinović, Ivona Jambrečić

Paper: Coordinated multi-robot exploration based on Graph SLAM method and rapidly exploring random trees

This paper presents coordinated multi-robot exploration based on Graph SLAM method and rapidly exploring random trees (RRT). The implementation is tested on Pioneer P3-DX mobile robots. The exploration algorithm of an unknown environment includes an RRT-based frontier detector module extended for use with Graph SLAM, the frontier point filter module and the robot task allocator module which uses an exploration time minimization strategy. The ROS navigation stack is used for path planning and following. Experiments were conducted in a simulation environment as well as in a real-world setup.

Keywords: mobile robots, multi-robot exploration, frontier points