SVEUČILIŠTE U ZAGREBU FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA ZAVOD ZA AUTOMATIKU I PROCESNO RAČUNARSTVO

DIPLOMSKI RAD Br. 1359

# Izgradnja lokalne vjerojatnosne karte zauzeća prostora na temelju ultrazvučnih senzora

Dino Čakija

Zagreb, rujan 2003.

#### SVEUČILIŠTE U ZAGREBU FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 05. studenog 2002.

#### Zavod: Zavod za automatiku i procesno računarstvo

Predmet: Mobilna robotika

## DIPLOMSKI ZADATAK br. 1359

Pristupnik: Dino Čakija

Studij: Elektrotehnike Smjer: Automatika

#### Zadatak: Izgradnja lokalne vjerojatnosne karte zauzeća prostora na temelju ultrazvučnih senzora

Opis zadatka:

Izgradnja karte prostora od vitalne je važnosti u navigaciji mobilnih robota, kako za estimaciju položaja mobilnog robota tako i za planiranje njegove putanje. U radu je potrebno implementirati algoritme za izgradnju lokalne vjerojatnosne karte zauzeća prostora u Saphira 8 razvojnom okruženju. Isprobati nekoliko inačica modela ultrazvučnih senzora i algoritama izgradnje karte. Algoritme eksperimentalno provjeriti na mobilnom robotu Pioneer DX2.

Zadatak uručen pristupniku: 07. studenog 2002. u 12:00 sati

Rok za predaju rada:

najkasnije do kraja akademske godine u kojoj je zadan diplomski zadatak

Mentor:

Dran Petronie Doc.dr.sc. Ivan Petrović

Djelovođa: Iran Petronic

Predsjednik povjerenstva za

diplomski ispit:

Prof.dr.sc. Nedjeljko Perić

Doc.dr.sc. Ivan Petrović

Zahvaljujem se mentoru doc.dr.sc. Ivanu Petrović i Edouardu Ivanjko, dipl.ing., na pomoći.

> Zahvaljujem se roditeljima na financijskoj i moralnoj podršci i bratu Dejanu na savjetima.

Zahvaljujem se i djevojci Ivani na podršci i razumijevanju.

# SADRŽAJ

1. UVOD	4
2. ACTIVMEDIA PIONEER 2 MOBILNI RO	BOT6
2.1. Fizičke karakteristike 2.2. Programska podrška	
2.2.1. Saphira	
2.2.2. Simulator	
2.2.3. Programski jezik Colbert	14
3. SENZORIKA MOBILNOG ROBOTA	
3.1. Propriocepcijski senzori	
3.1.1. Inkrementalni enkoder	
3.2. PERCEPCIJSKI SENZORI	
5.2.1. Olirazvucni senzor uaaljenosti	
4. KARTE PROSTORA	24
4.1. PROBLEMI PRI IZGRADNJI KARTE PROSTORA	
4.2 Upotreba načela vjerojatnosti	
4.2.1. Kalmanov filter	
4.2.2. EM familija algoritama 4.2.3 Mrežaste karte zauzeća	
	~~
5. LOKALIZACIJA MOBILNOG ROBOTA	
5.1. MODELI PREZENTACIJE	
5.1.1. Model okoline	
5.1.2. Model pozicije	
5.2. SAMO-LOKALIZACIJA	
5.3. REZULTATI	
5.3.1. Simulacija	
5.3.1. Realan robot Pioneer 2	
6. ZAKLJUČAK	
7. LITERATURA	
8. SAŽETAK	
	<b>F</b> 1
<b>7. LIVUIUFIS</b>	

# 1. UVOD

Još od začetka robotike kao zasebne znanstvene discipline primarni cilj bio je izraditi mobilni robot koji će se samostalno kretati prostorom i obavljati postavljene mu zadatke. Međutim, za ispunjenje toga cilja potrebno je riješiti mnoge probleme (izbjegavanje prepreka, izgradnja karte prostora, lokalizacija..) koji su razvojem računalne tehnologije postali rješivi. Naime, u idealnom slučaju, navigacija u prostoru sastoji se od uočavanja i prepoznavanja fizičkih obilježja i promjena prostora (sonar) i prikupljanju podataka o gibanju mobilnog robota (odometrija). Na osnovu tih prikupljenih podataka izračunava se točan položaj mobilnog robota u prostoru. U realnom svijetu pojavljuje se mnogo problema. Jedan od problema pri autonomnom gibanju mobilnog robota jest promjena njegovog fizičkog modela tokom vremena. Do promjene fizičkog modela mobilnog robota dolazi uslijed djelovanja trenja (promjena promjera kotača, istrošenost raznih prijenosnih mehanizama) te radne okoline (trešnja doprinosi labavosti vijaka...). Sve te promjene fizičkog modela mobilnog robota utječu na njegovu odometriju, a samim time i na njegovu položaj u prostoru. U svakom trenutku autonomnog gibanja mobilnog robota potrebno je znati njegov trenutni fizički model.

Određivanje trenutnog fizičkog modela mobilnog robota naziva se *kalibracija*. Kalibracija je postupak estimacije fizičkog modela mobilnog robota iz podataka dostupnih tj. prikupljenih sa odometrije i ostalih osjetila *(senzora)* sa samog mobilnog robota. Potreba za kalibracijom stara je kao i sama znanost o mobilnoj robotici te postoji mnogo razvijenih metoda kalibracije mobilnog robota. Međutim, većini tih metoda zajedničko je to da se kalibracija mobilnog robota ne može izvršiti bez intervencije čovjeka. Da bi se mobilni robot kalibrirao, čovjek *(ili neki vanjski uređaj)* mora mjeriti točno gibanje *(putanju)* mobilnog robota te iz tih mjerenja odrediti njegov fizički model. Takav pristup kalibracije ima dva velika nedostatka. Pošto se fizički model mobilnog robota mijenja tokom vremena potrebno je mjerenje gibanja izvršavati u redovnim vremenskim razmacima te obnavljati fizički model robota. Drugi nedostatak je taj da se mobilni robot mora prekidati u radu da bi se izvršila kalibracija. U ovom diplomskom radu opisana je metoda samo-kalibracije mobilnog robota u radu i bez intervencije čovjeka, a u svrhu njegove lokalizacije.

U drugom poglavlju opisana je fizička struktura i programska podrška mobilnog robota *Pioneer 2* proizvođača *ActivMedia*.

U trećem poglavlju opisani su *propriocepcijsk*i i *percepcijski* senzori mobilnog robota kojima se vrši prikupljanje podataka o gibanju i okolini u kojoj se mobilni robot nalazi te problemi koji se pojavljuju upotrebom tih senzora.

U četvrtom poglavlju govori se o vrstama karata prostora i opisani su najčešće korišteni algoritmi za njihovu izgradnju.

U petom poglavlju opisan je algoritam razvijen za samo-lokalizaciju *Pioneer 2* mobilnog robota, izneseni su rezultati simulacije i rezultati rada mobilnog robota u realnim uvjetima pomoću razvijenog algoritma.

U šestom i sedmom poglavlju dani su zaključak i popis korištene literature, a u osmom i devetom sažetak i životopis..

# 2. ACTIVMEDIA PIONEER 2 MOBILNI ROBOT

Mobilni robot *Pioneer 2* potječe iz porodice malih i inteligentnih robota Pioneer koji imaju pogon na 2 ili 4 kotača. Arhitekturu Pioneer familije robota razvio je Kurt Konolige, Ph.d., sa sveučilišta Stanford.



Slika 2.1. Pioneer2 mobilni robot

## 2.1. Fizičke karakteristike

Porodica Pioneer posjeduje sve potrebne komponente za navigaciju i interakciju sa okolinom realnog svijeta što ih čine pravim '*plug and play*' mobilnim robotima [6]. Svi su oni upravljani pomoću mikrokontrolera, koji je ugrađen u njih, i pripadne programske podrške. Glavni dijelovi *Pioneer 2* mobilnog robota su kućište, nos, konzola, paluba, pogonski kotači, pomoćni kotač, mreža sonara, poklopci za priključke i kontrolne lampice. Konzola i paluba čine jedan dio robota, ravnu ploča na gornjoj strani na koju se mogu ugraditi različiti uređaji kao kamera ili laserski senzori. Unutar čvrstog i laganog aluminijskog kućišta nalaze se baterija, pogonski motori, elektronika i druge standardne komponente poput prednje i stražnje mreže sonara. Također, unutar kućišta ima mjesta za standardni PC, Ethernet, modem, itd.



Slika 2.1.1. Fizičke komponente Pioneer2 mobilnog robota

*Pioneer 2* mobilni robot podržava do dvije mreže sonara. Jedna mreža je ugrađena u prednji dio mobilnog robota, a druga, dodatna, u stražnji dio mobilnog robota. Moguće je ugraditi do 24 sonara za snimanje okoline oko mobilnog robota. Snimanje sonarima vrši se u intervalu od 40 ms *(25 kHz),* a udaljenost snimanja je od 10 cm do 5 m.



Slika 2.1.2. Položaj mreže sonara

*Pioneer 2,* osim navedenih svojstva, ima I/O priključke za spajanje različitih perifernih uređaja kao i mogućnost ugradnje različitih vrsta napajanja. Na sabirnicu se može priključiti do 16 uređaja, ugrađeno je i osobno računalo sa četiri RS-232 serijska priključka, osam digitalnih I/O priključka, pet AD priključka, Ethernet, PC104

sabirnicom sa dodatnim priključcima i PSU kontroler, sve dohvatljivo preko uobičajenih aplikacija povezanih sa operativnim sustavom mobilnog robota.

## 2.2. Programska podrška

Samo upravljanje mobilnim robotom *Pioneer 2* izvršava se preko server-klijent strukture gdje *Pioneer 2 (odnosno, operativni sustav P2OS)* ima ulogu servera, a aplikacije koje se koriste za programiranje tj. upravljanje mobilnim robotom *(npr., razvojno okruženje Saphira)* imaju ulogu klijenta.



Slika 2.2.1. Server-klijent arhitektura

Glavna prednost server/klijent arhitekture je da različiti serveri (*tj. mobilni roboti*) mogu biti povezani i upravljani pomoću jednog klijenta. Za primjer se može uzeti dobiveni simulator pomoću kojeg se simulira ponašanje realnog mobilnog robota. Programer može razviti aplikaciju koja se bez ikakvih izmjena može koristiti u upravljanju realnog mobilnog robota nakon što se obave sva potrebna testiranja

pomoću dobivenog simulatora. Na *slici 2.2.2.* prikazane su mogućnosti povezivanja servera *(mobilni robot)* sa klijentom *(osobno računalo)* ili autonomno gibanje sa osobnim računalom, kao klijentom, na samom robotu.



Slika 2.2.2. Povezivanje servera sa klijentom

#### 2.2.1. Saphira

*Saphira* je okruženje za razvoj aplikacija pomoću kojih se upravlja mobilnim robotom u klijent/server strukturi. Ona sadrži set rutina potrebnih za razvoj različitih klijenata, funkcije za komunikaciju, odnosno, slanje naredaba serveru, funkcije za prikupljanje podataka sa različitih senzora koji se nalaze na mobilnom robotu te prikaz istih na grafičkom sučelju.

*Grafičko sučelje (slika 2.2.3)* sastoji se od komponenata koje se pojavljuju u okolini mobilnog robota i prikazano je iz ptičje perspektive. Postoji mogućnost prikaza okoline sa mobilnim robotom u centru ili u globalnim koordinatama. Mobilni robot prikazan je pomoću romba, koji se, ovisno o vrsti prikaza, ili giba *(okolina miruje)* ili miruje *(okolina se giba)*. Smjer ili rotacija robota označena je pravokutnikom prema kojem se robot giba. Točke prostora snimljene sonarima označene su plavim kvadratićima. Na lijevoj strani grafičkog sučelja nalaze se informacije :

- status da li stoji, da li se kreće ili je zapeo;
  - brzina translacijska i rotacijska;
- **položaj**  $-x, y, \theta$  koordinate izmjerene odometrijom;
- komunikacija broj paketa primljenih od servera;
- **razno** status baterije, zauzetost CPU-a.

Na dnu grafičkog sučelja nalazi se područje za interakciju sa *Saphirom*. U tom području ispisuju se važnije informacije o sustavu, a korisnik može direktno upisivati naredbe za upravljanje mobilnim robotom.

Osim toga, korisnik može:

- mijenjati radni direktorij i aktivnost robota ( activity file );
- uspostavljati i prekidati vezu sa serverom;
- definirati, pokretati i zaustavljati aktivnosti robota;
- pratiti aktivnosti robota;
- mijenjati interne varijable Saphire;
- dobiti pomoć o funkcijama i API-u.



Slika 2.2.3. Saphira grafičko sučelje

Na vrhu grafičkog sučelja nalaze se izbornici:

- Connect uspostavljanje veze sa serverom preko serijskog, lokalnog i TCP protokola, prekid veze i izlaz iz Saphire;
- *Files* odabir parametara robota i modela prostora iz gotovih datoteka;
- Grow uvećanje prikaza prostora i robota u njemu;
- *Shrink* smanjenje prikaza prostora i robota u njemu;
- Display prikaz prostora u lokalnim ili globalnim koordinatama, prikaz korak po korak ili sa kartom zauzeća;
- Sonars upravljanje sonarima;
- Functions prikaz ponašanja, aktivnosti i procesa u novom prozoru.

Također, postoje funkcije više razine za kontrolu mobilnog robota i senzorike poput neizrazitog ponašanja (*fuzzy-control behaviour*), upravljanja pomoću mapa i sustava registracije. *Saphira* se može podijeliti na dvije arhitekture povezane jedna na drugu. Prva arhitektura je *sustavna arhitektura* (*system arhitecture*) koja se koristi za kontrolu robota pomoću računala (*host computer*) pri čemu se korisnik ne mora zamarati sa tokom podataka unutar hardvera nego koristi gotove procedure i podatke sa senzorike za kontrolu mobilnog robota. Na *slici 2.2.4.* plavom bojom označene su procedure koje se izvršavaju svakih 100 ms, a crvenom bojom označene su procedure i rutine korisnika.



Slika 2.2.4. Sustavna arhitektura

Druga arhitektura je arhitektura *kontrole robota (robot control arhitecture)* kod koje korisnik ima mogućnost upravljanja robotom od najniže razine *(upravljanje pogonskim motorima i senzorima)* do najviše razine *(planiranje putanje i prepoznavanje objekata)*. Arhitektura kontrole robota prikazana je na *slici 2.2.5*. Prostor u kojem se mobilni roboti kreću i obavljaju određene zadatke je geometrijski i točna prezentacija tog prostora je od iznimne važnosti za pravilno funkcioniranje mobilnog robota.



Slika 2.2.5. Arhitektura kontrole robota

Unutar Saphire koriste se dva koordinatna sustava za prikaz geometrijskog prostora (označeno svjetlo-plavom bojom na slici 2.2.5.). Lokalni koordinatni sustav (local perceptual space ili LPS) je sustav u kojem je mobilni robot uvijek u ishodištu, odnosno, sve što se nalazi u prostoru ima koordinate prema mobilnom robotu. LPS je koristan za praćenje gibanja robota u kratkim vremenskim intervalima, fuziju podataka sa senzora i otkrivanje prepreka. Globalni koordinatni sustav (global map space ili GMS) ima ishodište fiksirano u nekoj početnoj točci mobilnog robota ili u nekoj proizvoljnoj točki (obično 0,0,0) i sve što se nalazi u prostoru, pa i sam robot, ima koordinate prema toj točki.

Arhitektura *Saphire* izgrađena je na osnovu operativnog sustava (*OS*) koji ima rutine za prekid i sinkronizam. OS podržava *mikro – radnje (micro – tasks)* koje se izvršavaju korak po korak sinkrono svakih 100 ms. Mikro – radnje su procedure, koje nakon izvršenja, poprime neko konačno stanje (*finite state machines* ili FSM) i korisne su za određivanje točno određenog vremenskog kašnjenja koje je korisno u kontroli robota.

### 2.2.2. Simulator

*Simulator* je vrlo korisna alternativa fizičkog mobilnog robota koji se koristi za razvoj programa za upravljanje mobilnim robotom. Iako je vrlo teško simulirati realan svijet koji je najpouzdaniji test za projekte vezane uz mobilni robot, simulator omogućuje ispitivanje projekata korak po korak pri čemu se može otkriti većina grešaka koje bi se mogle pojaviti u realnom svijetu. U simulator su implementirana i fizička ograničenja mobilnog robota poput realnih modela pogrešaka sonara i inkrementalnih enkodera. U većini slučajeva, ako projekt radi na simulatoru, raditi će i u realnom svijetu. Simulator podržava kreiranje vlastitih svjetova u kojem će se robot kretati i vrstu ponašanja. Pošto je cijelo sučelje i komunikacija simulatora jednaka fizičkom robotu, nije potrebno vršiti nikakve dodatne preinake samog projekta pri prebacivanju rada sa simulatora na fizički robot. Grafičko sučelje simulatora dano je na *slici 2.2.6*.



Slika 2.2.6. Grafičko sučelje simulatora

Najveći dio sučelja zauzima prikaz mobilnog robota, koji je u centru, i svijeta u kojem se kreće. Svijet (*zidovi, vrata, prepreke…*) može odrediti sam korisnik prema realnom ili izmišljenom svijetu tako da izradi pripadajuću datoteku u koju ga pohrani i kasnije koristi u simulaciji. Na dnu sučelja nalaze se stvarne ili točne koordinate robota koje se mijenjaju prema početnom položaju robota u danom svijetu, status komunikacije između robota i klijenta te naziv datoteke svijeta koja se koristi u simulaciji. Na vrhu sučelja nalaze se slijedeći izbornici:

- Connect provjera i prinudni prekid komunikacije sa klijentom;
- Files određivanje ponašanja robota i svijeta u kojem se kreće;
- Grow uvećanje prikaza simulacije;
- Shrink smanjivanje prikaza simulacije;
- Wake prikaz trajektorije kretanja robota zadnjih par sekundi;
- Recenter postavljanje robota u centar prikaza;

## 2.2.3. Programski jezik Colbert

Programski jezik *Colbert* sličan je programskom jeziku C i baziran je na semantici mikro – radnji. Kao i C, *Colbert* podržava programiranje poput iteracija, uvjetnih grananja ili petlji koje se izvršavaju u smislu kontrole robota. Glavna procedura programskog jezika *Colbert* naziva se *shema aktivnosti* (*activity schema*) koja opisuje sve radnje koje mobilni robot mora izvršiti u svome radu. Prednost takvog programiranja je što su radnje međusobno neovisne tako da greška u jednoj od radnji ne uzrokuje prekid rada cjelokupne sheme aktivnosti već se problematična radnja isključi iz sheme i javlja se korisniku poruka o grešci.

## **3. SENZORIKA MOBILNOG ROBOTA**

*Odometrija* mobilnog robota, u sprezi sa percepcijskim senzorima, omogućava određivanje položaja mobilnog robota u prostoru [1]. Mobilni robot ima 3 stupnja slobode u prostoru u kojem se giba. *Poziciju* mobilnog robota određuju koordinate u prostoru *x* i *y*. *Položaj* mobilnog robota određuju koordinate u prostoru *x*, *y* i  $\theta$  (*razlika kuta u orijentaciji mobilnog robota između početnog i trenutnog položaja*):

$$x(t) = \int_{0}^{t} v(t) \cos[\theta(t)] dt$$
  

$$y(t) = \int_{0}^{t} v(t) \sin[\theta(t)] dt$$
 (3-1)  

$$\theta(t) = \int_{0}^{t} \omega(t) dt$$

Pozicija i orijentacija mobilnog robota određuju njegov položaj u prostoru. Mobilni roboti u većini slučajeva nemaju mogućnost neovisnog upravljanja sa te tri varijable nego se mora izvršiti složeno manevriranje da bi došli na određeni položaj u prostoru. Takvi roboti nazivaju se *neholonomski* roboti.

Složeno manevriranje mobilnog robota vrši se preko *direktne* i *inverzne kinematike*. Direktna kinematika opisuje kako iz brzina pogonskih kotača  $v_L$  i  $v_R$  odrediti brzinu v i položaj ( $x, y, \theta$ ) mobilnog robota.



Slika 3.1. Direktna kinematika

Kutna brzina robota *w* određena je izrazom

$$\omega = \frac{v_R - v_L}{2d}, \qquad (3-2)$$

dok je polumjer rotacije *R* oko trenutačnog centra zakrivljenosti (*Instatanous Center* of *Curvature ili ICC je točka oko koje svi kotači prave kružnu putanju*) određen izrazom

$$R = d \, \frac{v_R + v_L}{v_R - v_L}.$$
 (3-3)

Iz izraza (3-2) i (3-3) slijedi brzina v(t)

$$v(t) = R\omega = \frac{v_R + v_L}{2}$$
(3-4)

iz koje se pomoću izraza (3-1) i kutne brzine  $\omega$  mogu izračunati koordinate i orijentacija robota.

Inverzna kinematika opisuje kako upravljati brzinama kotača  $v_L$  i  $v_R$  da bi se došlo određenom brzinom v u željeni položaj ( $x, y, \theta$ ). Dok direktna kinematika ima jednoznačno rješenje, inverzna kinematika ima više mogućih rješenja jer treba iz sustava jednadžbi određenog izrazima (3-1) ÷ (3-4) odrediti brzine kotača  $v_R$  i  $v_L$ . Izbor optimalnog rješenja ovisi o kriterijima poput najkraćeg vremena gibanja robota, minimalnoj potrošnji energije, linearnoj brzini kotača itd. Najčešće rješenje inverzne kinematike provodi se u tri koraka.

U prvom koraku mobilni robot rotira u mjestu dok ne postigne orijentaciju prema željenom položaju *(slika 3.2.),* pri čemu su brzine kotača jednake nekoj maksimalnoj brzini koju robot može postići u rotaciji na mjestu i različitog predznaka.



Slika 3.2. Prvi korak inverzne kinematike

U drugom koraku robot se pomakne u željeni položaj (*slika 3.3.*), a u trećem koraku opet se rotira u mjestu dok ne postigne željenu orijentaciju (*slika 3.4.*).



Dijagram brzine kotača prikazan je na slici 3.5.



Slika 3.4. Dijagram brzine kotača pri inverznoj kinematici

Pogon korišten u opisu direktne i inverzne kinematike naziva se *diferencijalni pogon*. Diferencijalni pogon se sastoji od dva nezavisno pokretana kotača pri čemu razlika brzine lijevog i desnog kotača određuje kut zakreta ili rotaciju robota *(slika 3.4.)*. Dobra svojstva diferencijalnog pogona su jednostavnost izvedbe, pokretljivost i niska cijena dok je loše svojstvo teško održavanje pravocrtnog gibanja zbog pogrešaka u brzinama kotača koje mogu nastati uslijed neravnog terena, nejednakoj istrošenosti kotača *(različiti promjer kotača),* proklizavanja kotača i promjene razmaka kotača pri zakretu.

## 3.1. Propriocepcijski senzori

Svaki mobilni robot ima *propriocepcijske* senzore koji mjere unutarnja stanja mobilnog robota. Propriocepcijski senzori mogu se podijeliti na *odometrijske* senzore i *orijentacijske* senzore. Odometrijski senzori omogućuju izračunavanje položaja mobilnog robota, odnosno estimaciju prijeđenog puta mobilnog robota preko mjerenja brzine vrtnje kotača dok orijentacijski senzori određuju orijentaciju mobilnog robota prema magnetskoj ili zemljopisnoj osi Zemlje. Odometrijski senzori su inkrementalni enkoderi, apsolutni enkoderi, akcelerometri, senzori bazirani na Dopplerovu efektu itd. Orijentacijski senzori su žiroskopi, kompasi i žirokompasi.

#### 3.1.1. Inkrementalni enkoder

Mjerenje brzine odnosno zakreta lijevog i desnog kotača mjeri se pomoću inkrementalnog enkodera.



Slika 3.5. Inkrementalni enkoder

Inkrementalni enkoder se u prošlih 50 godina vrlo malo promijenio. Konvencionalni inkrementalni enkoder sastoji se od izvora svjetlosti, stacionarne maske, rotirajućeg diska, foto-detektora i procesora koji obrađuje dobiveni signal *(slika 3.6.).* 



Slika 3.6. Dijelovi inkrementalnog enkodera

Kako se rotirajući disk kreće, svijetlo ili prolazi kroz njega i detektira se u foto-diodi ili biva blokirano od diska. Prolaz svijetla, odnosno njegovu detekciju procesor tumači kao vrijednost jedan, a blokiranu svijetlost kao vrijednost nula. Tim vrijednostima pridodaju se naponske razine i dobije se pravokutni signal koji je proporcionalan kutu zakreta osovine na kojoj se nalazi inkrementalni enkoder.



Slika 3.7. Signali inkrementalnog enkodera

Međutim, da bi se znala trenutna pozicija osovine mora se znati prijašnja pozicija pa se zbog toga na vanjskom rubu kodnog diska nalazi referentni kod koji služi za postavljanje enkodera u nultu poziciju. Inkrementalni enkoder koji se koristi u *Pioneer* 2 mobilnom robotu postavlja se u nulti položaj svakih prijeđenih 3000 mm.

## 3.2. Percepcijski senzori

Za točnu estimaciju položaja mobilnog robota potrebno je rezultate o gibanju, dobivene iz odometrije, korigirati pomoću rezultata dobivenih iz *percepcijskih* senzora mobilnog robota. Percepcijski senzori mjere stanje okoline u kojem se robot giba i uglavnom rade na načelu zračenja. Klasifikacija percepcijskih senzora prema vrsti

izvora i vrsti zračenja, pretraživanju i prikazu informacija o sceni, dometu i tehnici određivanja udaljenosti dana je u *tablici 3.1.* 

Izvor zračenia	Pasivni – koriste zračenje okoline		
·_···j.	Aktivni – koriste vlastiti izvor zračenja		
Vrsta zračenia	Elektromagnetsko zračenje		
<b>,</b>	Zvučno zračenje		
Pretraživanie scene	Skenirajući – pretražuju scenu gibanjem		
	Neskenirajući – ugrađeno polje senzora		
Informacije o sceni	Slika – generiraju sliku prostora		
	Točka – generiraju jednu točku prostora		
Domet	Blizinski – detekcija i izbjegavanje prepreka		
	Daljinski – mjere udaljenost do prepreke		
	Triangulacija		
Tehnika određivanja udaljenosti	Vrijeme putovanja zrake		
	Sažimanje scene		
	Interferometrija		

Tablica 3.1. Karakteristike percepcijskih senzora

Percepcijski senzori su korisni za planiranje putanje mobilnog robota, detekciju i izbjegavanje prepreka, pretraživanje prostora, estimaciju pozicije robota, manipulaciju predmetima itd. Prednosti radijacijskih senzora su u percepciji širokog prostora, izbjegavanju kontakata i unaprijednom dobivanju informacija dok su nedostaci problemi radijacijskog načela rada, računska složenost obradbe informacija i problemi interpretacije percepcijskih informacija.

#### 3.2.1. Ultrazvučni senzor udaljenosti

*Ultrazvučni senzor udaljenosti* ili sonar (*Sound Navigation and Ranging*) je jedan od najčešće korištenih percepcijskih senzora zbog jednostavnosti, male mase, male potrošnje, niske cijene i odličnih svojstva pri izbjegavanju prepreka. Najveći nedostaci su velika širina zrake, mala brzina zvuka, gušenje signala, velika

minimalna udaljenost mjerenja, ograničena maksimalna udaljenost, spekularne refleksije itd. Sonar se sastoji od dijafragme preko koje se odašilju ultrazvučni signali različitih frekvencija, dobiveni pomoći piezoelektričnog efekta, u prostor te se snimaju povratni signali iz kojih se može zaključiti o karakteristikama prostora u kojem se nalazi mobilni robot.



Slika 3.8. Načelo rada sonara

Maksimalni domet određen je intenzitetom odašiljanih signala koji opada sa udaljenošću, a ovisi i o samom mediju kroz koji putuju signali, temperaturi i vlažnosti.



Slika 3.9. Ovisnost intenziteta o uvjetima okoline

Slabljenje signala je približno 10 dB/m pa bi za domet od 5 m intenzitet odašiljanog signala trebao biti veći od 50 dB. Minimalan domet *(minimalna mjerljiva udaljenost)* ovisi o trajanju slanja impulsa i frekvencijskom opsegu metalne dijafragme, odnosno,

piezoelektričnog kristala. Frekvencija uzorkovanja signala određena je brzinom zvuka i maksimalnim dometom. Za domet od 5 m iznosi 33 Hz. Točnost mjerenja udaljenosti određena je promjenama brzine zvuka uslijed različite temperature i vlažnosti okoline. Ovisnost brzine zvuka o temperaturi dana je izrazom

$$c = 331.4\sqrt{T/273} = 331.4 + 0.607\Theta \tag{3-5}$$

pri čemu je

- c brzina zvuka,
- T temperatura u K,
- $\Theta$  temperatura u °C.

Preciznost mjerenja udaljenosti određena je valnom duljinom nosećeg signala. Intenzitet povratnog signala određuje se integracijom intenziteta zrake preko geometrije prostora od kojeg se zraka reflektira. Slaba kutna preciznost utječe i na slabu preciznost mjerenja udaljenosti.

Problem pomiješanih točaka javlja se zbog zaobljavanja kutova prepreka koje sonar registrira. Zaobljavanje kutova je posljedica integracije intenziteta zrake preko čitavog prostora refleksije zrake (*slika 3.10.*).



Slika 3.10. Zaobljavanje kutova prepreka

Problem spekularnih i višestrukih refleksija jedan je od većih problema koji se javljaju pri upotrebi sonara. *Spekularna refleksija* javlja se kad emitirana zraka udari u vrlo glatku površinu predmeta pod kutem manjim od 90° i zatim se odbije u prostor *(slika 3.11.).* Samim time ne postoji povratni signal i sonar ne može detektirati predmet u

prostoru. Uvjet za spekularnu refleksiju dan je izrazom (3-6) i poznat je pod imenom *Rayleighov uvjet (slika 3.12.).* 



Slika 3.12. Rayelighov uvjet

*Višestruke refleksije* pojavljuju se kada se povratna zraka reflektira sa sekundarnog objekta u prostoru nazad u sonar.



Slika 3.13. Višestruka refleksija

# 4. KARTE PROSTORA

Izgradnja karte prostora u kojem se giba mobilni robot jedan je od najvažnijih uvjeta za postizanje autonomnog gibanja mobilnog robota [3]. Za izgradnju takve karte prostora mobilni robot mora imati niz različitih senzora za interakciju sa okolinom i pripadne mehanizme da se izdvoje korisne informacije od svih podataka dobivenih pomoću senzora. Karta prostora predstavlja model svijeta u kojem se giba mobilni robot i kao takva može se koristiti za planiranje putanje, izbjegavanje prepreka, estimaciju položaja i gibanja itd.

Karte prostora mogu se podijeliti na *metričke karte* i *topološke karte*. Metričke karte prikazuju prostor u geometrijskom smislu dok topološke karte opisuju povezanost različitih mjesta u prostoru. Različita mjesta u prostoru predstavljaju čvorove koji su povezani spojnicama koje određuju način na koji se robot mora gibati da bi došao iz jednog čvora u drugi.

Povijesno gledano, u početku su se primjenjivale karte kojima je centar bio robot, a okolina je bila opisana informacijama koje su dobivene iz mjerenja samog robota. Iako se činilo da je izgradnja takve karte jednostavna jer se nije trebalo uzimati u obzir gibanje robota u prostoru, veliki problem bilo je razlikovati dva geometrijski različita mjesta u prostoru koja su imala isti informacijski zapis. Kasnije su se počele koristiti globalne karte prostora u kojima su svi objekti u prostoru, uključujući i robot, imali geometrijske koordinate prema nekoj početnoj fiksnoj točki. Objekti nisu opisani informacijama dobivenih iz mjerenja koja su ih otkrila.

## 4.1. Problemi pri izgradnji karte prostora

Prvi problem pri izgradnji karte prostora su postojanje mjernog šuma i fizička ograničenja senzora robota. Npr, zvuk i svjetlost ne mogu prolaziti kroz zidove. Ta fizička ograničenja zahtijevaju gibanje robota kroz prostor da bi se izgradila karta prostora. Kada bi odometrija robota bila točna, tada bi se gibanjem robota kroz prostor mogla izgraditi karta na temelju podataka dobivenih odometrijom. Postojanje mjernog šuma ne bi bio problem da je šum u različitim mjerenjima statistički neovisan. Tada bi se sa povećanjem broja mjerenja utjecaj šuma smanjivao. Međutim, mjerni šum je statistički ovisan i sa vremenom se akumulira te utječe na

interpretaciju sljedećih mjerenja. Takav problem prikazan je na *slici 4.1.* Na slici je vidljivo da je mala rotacijska pogreška na jednom kraju hodnika uzrokovala veliku pogrešku u odometriji na drugom kraju hodnika. Daljnim gibanjem robota greške su se akumulirale i ukupna pogreška je postala velika.



Slika 4.1. Greška u odometriji

Smanjenje utjecaja pogrešaka, odnosno mjernog šuma, je glavni problem izgradnje karte prostora. Zbog toga su razvijeni mnogi algoritmi izgradnje prostora koji su i matematički i implementacijski kompleksni.

Drugi problem pri izgradnji karte prostora nalazi se u opisu objekata u prostoru. Svaki objekt u prostoru određuje njegova dimenzija. Najjednostavniji opis prostora dobiva se upotrebom nekoliko brojeva kojima se mogu opisati objekti poput vrata, zidova, križanja, soba... Detaljan dvodimenzionalni opis prostora iziskuje upotrebu nekoliko stotina brojeva. 3D opis zgrade u kojoj se robot giba iziskuje nekoliko milijuna brojeva. Za rješavanje ovog problema potrebno je imati snažnu hardversku podršku. Treći problem, i najveći, je poznat pod imenom *problem korespondentnosti*. To je problem određivanja da li se senzorska mjerenja izvršena tokom vremena odnose na isti objekt u prostoru. *Slika 4.2.* prikazuje pokušaj izgradnje karte kružnog prostora pri čemu dolazi do problema korespondentnosti. Kada robot prolazi cijeli krug, mora znati gdje se nalazi relativno prema prošloj izgrađenoj karti. Kako se bliži početnom položaju, akumulirana pogreška je sve veća i broj mogućnosti ili hipoteza gdje se nalazi je sve veći.



Slika 4.2. Problem korespondentnosti

Četvrti problem je dinamičko okruženje robota. Okruženje robota podložno je promjenama tokom vremena. Neke promjene su spore, poput izgleda drveća tokom godine ili promjene strukture zgrada uslijed starenja, a neke su brže, poput prolaska automobila ili ljudi. Npr., ako robot dođe ispred vrata koja su trenutno zatvorena, a bila su otvorena. Iz te situacije mogu se zaključiti dvije mogućnosti. Ili su vrata promijenila svoj status *(otvorena-zatvorena)* ili robot nije na mjestu na kojem misli da se nalazi. Ne postoji algoritam za izgradnju karata dinamičkih okruženja, nego se se postojeći algoritmi baziraju na pretpostavci da je sve što se giba običan šum.

Peti i zadnji problem proizlazi iz činjenice da robot mora odlučivati kojim će se putem gibati kroz prostor. Istraživanje prostora je gibanje robota u svrhu izgradnje karte prostora. Dok je gibanje robota u potpuno modeliranim prostorima vrlo dobro savladano, gibanje robota u nepoznatim prostorima u svrhu izgradnje karte u realnom vremenu još je uvijek prepuno izazova. Pri planiranju putanje robota kroz prostor potrebno je uzeti u obzir činjenice poput gubitka vremena i energije pri prikupljanju informacija i mogućnost gubitka informacija o položaju tijekom vremena.

Izgradnja karte prostora usko je povezana sa lokalizacijom robota. Kada bi se te dvije problematike razdvojile tada bi jedna na drugu djelovale poput sistemske pogreške. Estimacijom pojedinih problematika, izgradnje karte te lokalizacije u isto vrijeme, eliminira se sistemska pogreška, odnosno, pogreške postaju neovisne. Kada bi se znao točan položaj robota u svakom trenutku, tada bi se vrlo lako mogla izgraditi karta prostora. Isto tako, kada bi se znala točna karta prostora tada bi se mogao odrediti točan položaj robota u njemu. Međutim, vršiti lokalizaciju i izgradnju karte u isto vrijeme je puno teže.

## 4.2 Upotreba načela vjerojatnosti

Svi današnji vrhunski algoritmi za izgradnju karte prostora imaju jednu zajedničku osobina, a ta je da se svi temelje na *vjerojatnosti*. Svi oni koriste vjerojatnostni model robota i njegove okoline i pretvorba senzorskih mjerenja u karte prostora također se temelji na vjerojatnosti. Razlog takvom pristupu leži u činjenici da se izgradnja karte prostora temelji na šumu i nesigurnosti. Pomoću vjerojatnosti algoritmi modeliraju različite izvore šuma i njihov utjecaj na mjerenja.

Princip na kojem se temelje algoritmi vjerojatnosti naziva se Bayesovo pravilo:

$$p(x \mid d) = \eta p(d \mid x) p(x)$$
. (4-1)

Neka *x* predstavlja dio karte prostora, a *d* podatke dobivene mjerenjem sa sonara ili odometrije. Cilj je izgraditi kartu prostora na osnovu podataka dobivenih sa sonara ili odometrije. Bayesovo pravilo nam kaže da je vjerojatnost da je dio prostora *x* zauzet, uz uvjet da je dobiven podatak *d*, jednaka umnošku vjerojatnosti da je dobiven podatak *d*, uz uvjet da je dio prostora *x* zauzet i vjerojatnosti da je dio prostora *x* zauzet prije dolaska podatka *d*. Koeficijent  $\eta$  služi za normiranje dobivene vrijednosti. Ako se sa *m* označi karta prostora, a sa *z* pozicija robota tada se može napisati relacija za Bayesov filter:

$$p(z_{t}, m \mid s^{t}, o^{t}) = \eta p(s^{t} \mid z_{t}, m) \int p(z_{t} \mid o^{t}, z_{t-1}) p(z_{t-1}, m \mid s^{t-1}, o^{t-1}) dz_{t-1}$$
(4-2)

pri čemu je  $d = \{ s^0, o^0, s^1 \dots s^t, o^t \}$ . Podatak d sastoji se od očitanja sonara s i odometrije o u vremenskim intervalima t. Da bi se estimator (4-2) mogao koristiti potrebno je definirati vjerojatnosti  $p(s^t | z_t, m)$  i  $p(z_t | o^t, z_{t-1})$ . Pretpostavlja se da su te dvije vjerojatnosti invarijante tj. da ne ovise o vremenu t. Obje predstavljaju generički model robota i okoline. Prva vjerojatnost naziva se *percepcijski model* i opisuje kakva mjerenja s se dobivaju za različite karte prostora m i pozicije robota z. Druga vjerojatnost se naziva *model gibanja* i opisuje vjerojatnost da se robot pomakne sa pozicije  $z_{t-1}$  u poziciju  $z_t$  ako je robot izvršio gibanje  $o^t$ .

Najznačajniji algoritmi izgradnje karte prostora i njihova usporedba dani su u tablici 4.1.

	Kalman	Lu/Milios	EM	Incremen-	Hybrid	Occupancy	Multi-	Dogma
				tal ML		Grids	Planar	
							Maps	
Representation	landmark	point ob-	point ob-	landmark	point ob-	occupancy	objects	occupancy
	locations	stacles	stacles	locations	stacles	grids	and	grids
				or grid			polygons	
				maps				
Uncertainty	posterior	posterior	maximum	(local)	maximum	posterior	maximum	posterior
	poses	poses	likeli-	max-	likeli-	map	likeli-	map
	and map	and map	hood	1mum	hood		hood	
			map	likeli-	map		map	
				hood				
				map				
Convergence	strong	no	weak?	no	no	strong	weak	weak
Local Minima	no	yes	yes	yes	yes	no	yes	yes
Incremental	yes	no	no	yes*	yes	yes	no	no
Requires Poses	no	no	no	no	no	yes	yes	yes
Sensor Noise	Gaussian	Gaussian	any	any	any	any	Gaussian	any
Can map cycles	yes	no	yes	no	yes,	n/a	n/a	n/a
					but not			
					nested			
Map dimensionality	$\sim 10^3$	unlimited						
Correspondence	no	yes						
Handles raw data	no	yes						
Dynamic env's	limited	no	no	no	no	limited	no	yes

Tablica 4.1. Najznačajniji algoritmi izgradnje karte prostora

U nastavku će ukratko biti dan pregled nekih od gore navedenih algoritama.

#### 4.2.1. Kalmanov filter

Klasičan pristup izgradnje karte prostora temelji se na Kalmanovom filtru [3]. Postoje algoritmi koji se temelje na modificiranom Kalmanovom filtru i svi su oni poznati pod kraticom *SLAM ( Simultaneous localization and mapping ).* Zajedničko im je da estimiraju rekurzivno i da kombiniraju višestruka mjerenja.

Kalmanov filter je Bayesov filter koji opisuje vjerojatnost  $p(z_t, m | s^t, o^t)$  uz dodani *Gaussov model šuma*. Gaussov model šuma može se izraziti vektorom stanja *x* dimenzije 2*K*+3 koji opisuje poziciju robota *z* i kartu *m*:

$$x = (z_{x,t}, z_{y,t}, z_{\theta,t}, m_{K,x,t}, m_{K,y,t})^T$$
(4-3)

pri čemu je sa K = 1, 2, 3... određen broj karakterističnih objekata u karti prostora. Ti objekti mogu biti neka karakteristična obilježja ili oblici u prostoru. Izgradnja karte pomoću Kalmanova filtra temelji se na dvije pretpostavke. Prva pretpostavka je da model gibanja mora biti linearan sa dodanim Gaussovim modelom šuma što znači da pozicija i karta u trenutku *t* ovise linearno o poziciji i karti u trenutku t -1, a također i o odometriji u trenutku t -1. Dok je slučaj sa linearnom ovisnošću o karti ispunjen, linearna ovisnost o poziciji i odometriji nije ispunjena zbog pojave nelinearne trigonometrijske funkcije. Zbog toga Kalmanov filter aproksimira model gibanja pomoću linearne Taylorove funkcije. Takav Kalmanov filter naziva se *prošireni Kalmanov filter*. Rezultat linearizacije je funkcija varijabli stanja koja se može pisati kao linearna funkcija sa dodani Gaussovim šumom:

$$p(x \mid x', o) = Ax' + Bo + \varepsilon_c \tag{4-4}$$

Linearne matrice *A* i *B* povezuju varijable stanja x' i odometriju *o* sa slijedećom varijablom stanja *x*. Šum je modeliran varijablom  $\varepsilon_c$  te se smatra da pripada normalnoj razdiobu sa nula-značenjem i kovarijancom  $\sigma_c$ .

Druga pretpostavka je da iste karakteristike mora imati model percepcije. Mjerenja dobivena senzorima su također nelinearna sa osobinama Gaussovog šuma te se vrši linearizacija pomoću Taylorove funkcije:

$$p(s \mid x) = Cx + \varepsilon_m \tag{4-5}$$

pri čemu je x=(z,m), C je linearna matrica, a  $\varepsilon_m$  je normalno distribuiran šum mjerenja sa nula-značenjem i kovarijancom  $\sigma_m$ .

Na *slici 4.3.* prikazan je rezultat estimacije pozicije i karte prostora Kalmanovim filtrom. U prostoru je detektirano 14 obilježja od kojih je 5 odgovara tankim i vertikalnim objektima, a ostali potječu od reflektirajućih objekata u prostoru. Sa plavim elipsama označena je rezidualna nesigurnost mjerenja koja se pojavljuje i pri gibanju robota, označena sa crvenim isprekidanim elipsama. Na *slici 4.4.a* prikazana su obilježja prostora dobivena simulacijom, na *slici 4.4.b* je korelacijska matrica nakon 278 iteracija, a na *slici 4.4.c* normalizirana inverzna matrica kovarijance koja prikazuje korelaciju između 3D koordinata robota i 2D koordinata objekata u prostoru.

Korelacija između estimiranih x i y koordinata robota i objekata u prostoru je vrlo velika, međutim, karta na *slici 4.4.a* je još uvijek nesigurna.



Slika 4.3. Estimacija pozicije robota i karte prostora Kalmanovim filtrom



Slika 4.4. Rezultati simulacije dobiveni Kalmanovim filtrom

Bayesov filter dan izrazom (4-2) može se izračunati slijedećim standardnim izrazima Kalmanovog filtra:

$$\eta'_{t-1} = \eta_{t-1} + Bo_t$$
  

$$\Sigma'_{t-1} = \Sigma_{t-1} + \Sigma_c$$
  

$$K_t = \Sigma'_{t-1}C^T (C\Sigma'_{t-1}C^T + \Sigma_m)^{-1}$$
  

$$\eta_t = \eta'_{t-1} + K_t (o - C \eta'_{t-1})$$
  

$$\Sigma_t = (I - K_t C)\Sigma'_{t-1}$$
(4-6)

Prednost Kalmanovog filtra je puna *a posteriori* estimacija stanja u realnom vremenu koja se izvodi ciklički i obuhvaća predikciju i korekciju. Nedostatak leži u pretpostavci Gaussovog šuma.

#### 4.2.2. EM familija algoritama

Alternativa Kalmanovom filtru naziva se *EM* (*expectation maximization*) porodica algoritama [3]. Za razliku od Kalmanovog filtra, EM algoritmi se temelje na statistici. Njihova efikasnost dolazi do izražaja u okolini u kojoj postoji mnogo sličnih obilježja. Ne koriste nesigurnost nego izvršavaju tkz. *hill – climbing* proceduru unutar prostora karti u potrazi za najvjerojatnijom kartom okoline. *Hill – climbing* procedura iziskuje obradu podataka nekoliko puta i snažne hardverske resurse.

EM algoritmi izvršavaju se u dva koraka:

- E korak ili korak očekivanja u kojem se izračunavaju moguće pozicije robota temeljeno na trenutnoj karti prostora,
- M korak ili korak maksimizacije u kojem se određuju moguće karte prostora temeljene na očekivanim pozicijama robota iz prvog koraka.

Rezultat je serija karata prostora čija točnost se povećava brojem karata. Inicijalna karta je prazna, odnosno, ima vrijednost nula.

EM algoritam dan je izrazom:

$$m^{[i+1]} = \arg\max_{m} \sum_{T} \int p(o_T \mid m^{[i]}, d^t) \log p(z_T \mid o_T, m) do_T \quad (4-7)$$

E korak u izrazu (4-7) nalazi se u rješavanju sume, a M korak u maksimizaciji te sume. Prednost EM algoritama u odnosu na Kalmanov filter je u rješavanju problema korespodentnosti. To rješava tako da u E koraku kontinuirano vrši relokalizaciju robota relativno prema trenutnoj karti prostora. Pozicije robota izračunate u E koraku odgovaraju različitim hipotezama gdje bi se robot mogao nalaziti i impliciraju različite korespodentnosti. Kada se u M koraku izgrađuju karte prostora, te korespodentnosti određuju objekte u karti koji se u slijedećem E koraku ili potvrđuju ili postupno nestaju.



Slika 4.5. Rezultat simulacije primjenom EM algoritma

Na *slici 4.5.a* prikazan je set podataka o 28 objekata u prostoru, koji su percepcijski neprepoznatljivi, i putanja robota. Ti objekti predstavljaju skretanja i raskrižja, ali robotu nije dana percepcijska informacija o tim objektima kako bi ih mogao razlikovati. Greška estimacije pozicije robota upotrebom danog seta podataka i odometrije (*slika 4.5.b*) je prevelika da bi se upotrijebila za rješavanje problema

korespodentnosti. Na *slici 4.5.c* upotrijebljen je EM algoritam na istom setu podataka te se vidi da je karta i putanja robota topološki ispravna. Na *slici 4.5.d* prikazana je mrežasta karta zauzeća dobivena upotrebom podataka na koji je primijenjen EM algoritam i mjerenja sa sonara pri čemu je estimirana pogreška pozicije robota vrlo mala.

#### 4.2.3. Mrežaste karte zauzeća

Gore navedeni algoritmi izgradnje karata prostora pripadaju *SLAM* algoritmima jer pri izgradnji karata ne znaju položaj robota. Jednostavniji slučaj pri kojem se zna položaj robota koristi se pri izgradnji *mrežastih karti zauzeća* [3]. Taj algoritam razvili su Elfes i Moravec sredinom 90–tih i uživa veliku popularnost. Mrežaste karte zauzeća koriste se u ovom diplomskom radu u svrhu lokalizacije mobilnog robota.

Glavni problem pri izgradnji mrežaste karte zauzeća predstavlja izgradnja konzistentne metričke karte iz nepotpunih ili šumom zagušenih podataka sa senzora. Čak i ako se zna položaj robota, teško je reći da li je dio prostora zauzet ili ne. Da bi izgradili mrežastu kartu zauzeća potrebno je imati robot sa sonarima ili laserskim senzorima udaljenosti. Osim što se u njihovim mjerenjima nalazi šum, ti senzori imaju i fizička ograničenja koja su navedena u prošlom poglavlju.

Da bi se riješio taj problem, pri izgradnji mrežastih karti zauzeća koriste se karte vjerojatnosti. Mrežasta karta zauzeća, kao što i samo ime govori, predstavlja prostor pomoću mreže koja je u većini slučajeva 2D, ali može biti i 3D. Mreža se sastoji od segmenata jednake veličine od kojih svaki predstavlja jedan mali dio prostora.

Neka su (*x*,*y*) koordinate segmenta, a  $m_{x,y}$  zauzetost segmenta. Zauzetost segmenta je binarna varijabla : ili je segment zauzet ili je slobodan. Problem izgradnje mrežaste karte zauzeća je problem izračunavanja *a posteriori* vrijednosti binarne varijable odnosno segmenta zauzetosti, pri čemu svaki od njih ima vjerojatnost  $p(m_{x,y} | s^t, x^t)$ . Bayesov filter omogućuje izračunavanje *a posteriori* vrijednosti binarne varijable pomoću šansi (*odds*). Šansa da se dogodi događaj *x* sa vjerojatnošću *p*(*x*) definirana je sa izrazom  $\frac{p(x)}{1-p(x)}$ .

Bayesov filter za izračun statičkih karata uz poznatu poziciju robota  $z^t$ , napisan u notaciji šansi, glasi:

$$\frac{p(m_{x,y} \mid s^{t}, z^{t})}{1 - p(m_{x,y} \mid s^{t}, z^{t})} = \frac{p(m_{x,y} \mid s^{t}, z^{t})}{1 - p(m_{x,y} \mid s^{t}, z^{t})} \frac{1 - p(m_{x,y})}{p(m_{x,y})} \frac{p(m_{x,y} \mid s^{t-1}, z^{t-1})}{1 - p(m_{x,y} \mid s^{t-1}, z^{t-1})}$$
(4-8)

Izraz (4-8) najčešće se implementira u logaritamskoj strukturi koja je brža za izračunavanje i sprečava numeričke nestabilnosti kada su vjerojatnosti blizu nule:

$$\log \frac{p(m_{x,y} | s^{t}, z^{t})}{1 - p(m_{x,y} | s^{t}, z^{t})} = \log \frac{p(m_{x,y} | s^{t}, z^{t})}{1 - p(m_{x,y} | s^{t}, z^{t})} + \log \frac{1 - p(m_{x,y})}{p(m_{x,y})} + \log \frac{p(m_{x,y} | s^{t-1}, z^{t-1})}{1 - p(m_{x,y} | s^{t-1}, z^{t-1})}$$
(4-9)

Iz danih izraza može se zaključiti da je algoritam za izgradnju mrežastih karata zauzeća rekurzivan te omogućuje osvježavanje vrijednosti segmenata kako pristižu novi podaci sa senzora. *A priori* vjerojatnost  $p(m_{x,y})$  ima vrijednost 0.5, dok se  $p(m_{x,y} | s^t, z^t)$  naziva *inverzni model* senzora i određuje vjerojatnost da li je segment  $m_{x,y}$  zauzet uz očitanje sonara  $s^t$  na lokaciji  $z^t$ .

Na *slici 4.6.* dan je primjer mrežaste karte zauzeća dobiven pomoću laserskog senzora udaljenosti. Segmenti crne boje (binarno 1) predstavljaju prepreke poput zidova i namještaja, a segmenti bijele boje (binarno 0) predstavljaju prostor bez prepreka. Segmenti sive boje (obično imaju vrijednost 0.5) predstavljaju dijelove prostora za koje se ne zna da li su zauzeti ili slobodni.



Slika 4.6. Mrežasta karta zauzeća

Za potrebe ovog diplomskog korištena je metoda izgradnje mrežaste karte zauzeća po Olle Wijku [2]. Njegova metoda se, također temelji na Bayes-ov pristupu. Bayesov pristup izgradnje mrežaste karte zauzetosti *G* počiva na Bayes-ovom pravilu

$$P(A_i | B) = \frac{P(A_i \cap B)}{P(B)},$$
(4-10)

koje se može napisati kao

$$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{\sum_{A_i} P(B \mid A_i)P(A_i)} .$$
(4-11)

 $A_i$  predstavlja događaj da li je segment karte zauzeća zauzet ili nije. Zbog toga je svaki segment  $C_{ij}$  karte G povezan sa binarnom varijablom  $S_{ij}$  koja može poprimiti vrijednosti O (*Occupied*) ili E (*Empty*) tako da vrijedi:

$$P(S_{ii} = O) + P(S_{ii} = E) = 1.$$
(4-12)

Ako se sa *B* označi novo očitanje sonara *R* tada se može napisati rekurzivna formula za osvježavanje vjerojatnosti  $p_{ij}$  da je segment zauzet:

$$P(S_{ij} = O \mid R) = \frac{P(R \mid S_{ij} = O)P(S_{ij} = O)}{\sum_{s=O,E} P(R \mid S_{ij} = s)P(S_{ij} = s)}.$$
(4-13)

Inicijalno, vjerojatnosti  $p_{ij}$  svih segmenata u karti *G* poprimaju vrijednost tako da vrijedi

$$p_{ij} = 0.5 \quad \forall C_{ij} \in G , \qquad (4-14)$$

odnosno, stanje segmenta je nepoznato. Kada se dobije novi podatak R sa sonara, vrši se osvježavanje vjerojatnosti  $p_{ij}$  segmenta koji se nalazi unutar područja pokrivenog djelovanjem sonara slijedećim pravilom:

$$p_{ij} = \frac{P(R \mid S_{ij} = O)p_{ij}}{P(R \mid S_{ij} = O)p_{ij} + [1 - P(R \mid S_{ij} = O)][1 - p_{ij}]}$$
(4-15)

U izrazu (5.7) potrebno je odrediti *model senzora*  $P(R|S_{ij} = O)$ . Najčešće korišten model senzora dan je slijedećim izrazom

$$P(R \mid S_{ij} = O) = \begin{cases} 0.5 + (p_E - 0.5)f_1(\theta, \rho), & \rho \prec r - 2\Delta r \\ 0.5 + (p_E - 0.5)f_2(\theta, \rho, \tau), & r - 2\Delta r \le \rho \prec r - \Delta r \\ 0.5 + (p_O - 0.5)f_3(\theta, \rho, \tau), & r - \Delta r \le \rho \prec r + \Delta r \\ 0.5, & \rho \ge r + \Delta r \end{cases}$$
(4-16)

$$f_{1}(\theta, \rho) = \alpha(\theta)\Delta\rho,$$
  

$$f_{2}(\theta, \rho, \tau) = \alpha(\theta)\Delta\rho(1 - (2 + (\frac{\rho - r}{\Delta r})^{2})),$$
  

$$f_{3}(\theta, \rho, \tau) = \alpha(\theta)\Delta\rho(1 - (\frac{\rho - r}{\Delta r})^{2}).$$
  
(4-17)

Vrijednosti  $p_E = 0.4$  i  $p_O = 0.6$  određuju donju i gornju granicu od  $P(R|S_{ij} = O)$ , a  $\rho$  je udaljenost od senzora do polja  $C_{ij}$ . Parametar  $\Delta r = 0.1$  *m* određuje područje u kojem se mjerenje udaljenosti *r* smatra točnim. Funkcijama  $f_1(\theta,\rho)$ ,  $f_2(\theta,\rho,\tau)$ ,  $f_3(\theta,\rho,\tau)$  model senzora podijeljen je na kutni i radijalni dio kroz funkcije  $\alpha$  ( $\theta$ ) i  $\Delta(r)$  koje su dane izrazima:

$$\alpha(\theta) = \begin{cases} 1 - \left(\frac{\theta}{12.5^{\circ}}\right)^2, & 0 \le |\theta| \le 12.5^{\circ} \\ 0 & |\theta| > 12.5^{\circ} \end{cases}$$
(4-18)

$$\Delta(\rho) = 1 - \frac{1 + \tanh(2(\rho - \rho_{\nu}))}{2}.$$
 (4-19)

# 5. LOKALIZACIJA MOBILNOG ROBOTA

Problem samo-lokalizacije robota može se podijeliti na probleme *globalne lokalizacije* i *praćenje pozicije* [4], [5]. Praćenje pozicije podrazumijeva poznavanje početne pozicije mobilnog robota, dok globalna lokalizacija sprečava da se mobilni robot izgubi zbog globalne nesigurnosti prostora. Oba problema su pojedinačno razrađena, ali mali ih broj mobilnih robota može rješavati istovremeno u velikim i kompleksnim prostorima. Algoritam razvijen za potrebe ovog diplomskog rada koristi lokalne mrežaste karte zauzeća, dobivene iz mjernih podataka sa sonara, i podatke sa odometrije za određivanje točne pozicije mobilnog robota u prostoru. Podacima sa sonara vrši se korekcija pozicije mobilnog robota dobivena odometrijom. Problem globalne lokalizacije riješen je promatranjem višestrukih Gaussovih hipoteza temeljenih na mogućim lokacijama robota u karti prostora. Problem praćenja pozicije riješen je izračunavanjem najvjerojatnije koordinate za svaku od mogućih pozicija.

## 5.1. Modeli prezentacije

Korištena su dva modela prezentacije prostora. Prvi model naziva se *model okoline* i opisan je u poglavlju 5.1.1. Drugi model naziva se *model pozicije* i opisan je u poglavlju 5.1.2.

#### 5.1.1. Model okoline

Model okoline sastoji se od mrežaste karte zauzeća dobivene mjerenjem okoline sonarima. Mrežasta karta zauzeća sastoji se 120 x 120 segmenata širine 50 mm i opisuje okolinu 6 x 6 m. Korisnik može mijenjati broj segmenata, ali površina prostora koju karta prikazuje mora ostati ista. U ovom algoritmu ne upotrebljava se mrežasta karta zauzeća u izvornom obliku nego se iz nje kreira par *histograma*, jedan u *x* smjeru, a jedan u *y* smjeru (*slika 5.1.*). Par histograma predstavlja brojčani opis dijela okoline u kojoj je izvršeno mjerenje sonarima. Svaki segment može imati tri stanja; zauzet (*Occupied*), prazan (*Empty*) i nepoznat (*Unknown*), što ovisi o vjerojatnosti zauzeća za taj segment. Na *slici 5.1.* segment koji je zauzet prikazan je crnom bojom, prazan sa bijelom bojom, a nepoznat sa sivom. Za primjer, dan je izraz (5-1)

u kojem  $p(c_{xy})$  označava vjerojatnost da je segment *c* sa koordinatama *x* i *y* zauzet. Robot se nalazi u centru karte i ima koordinate (60,60), gledano u koordinatnom sustavu karte koji ima ishodište u segmentu (0,0).



Slika 5.1. Mrežasta karta zauzeća i pripadni histogrami

$$Segment(c_{xy}) = \begin{cases} O \ if \ p(c_{xy}) \succ 0.5 \\ U \ if \ p(c_{xy}) = 0.5 \\ E \ if \ p(c_{xy}) \prec 0.5 \end{cases}$$
(5-1)

Histogrami su dobiveni zbrajanjem zauzetih, praznih i nepoznatih segmenata po svakom od 120 redaka i stupaca. Za svaki redak i stupac zna se koliko ima zauzetih, praznih i nepoznatih segmenata. Inicijalna vrijednost svih segmenata je 0.5.

#### 5.1.2. Model pozicije

Model pozicije robota sastoji se od 9 hipoteza  $H = \{h_1, h_2...h_9\}$ , odnosno, lokacija na kojoj se robot stvarno nalazi (*slika 5.2.*). Vjerojatnost da se robot nalazi na lokaciji i=1...9 dana je distribucijom vjerojatnosti  $P = \{p(h_1), p(h_2)...p(h_9)\}$ . Svaka lokacija sastoji se od kartezijevih koordinata ( $x_{hi}, y_{hi}$ ) i varijance  $v_{hi}$  koja se koristi za praćenje pozicije mobilnog robota. Pretpostavka je da se mjerni šum oko estimirane lokacije jednako raspoređuje u svim smjerovima i da ima oblik Gaussove raspodjele. To znači da se model aktuatora robota ne estimira točno, odnosno, ne razdvajaju se komponente translacijskog i rotacijskog gibanja.



Slika 5.2. Greška u poziciji robota zbog odometrije

#### 5.2. Samo-lokalizacija

U početku lokalizacije iz prvog mjerenja sonara izgrađuje se početna ili inicijalna mrežasta karta zauzeća. Iz te karte izračuna se par histograma his(0) i pohranjuje u memoriju. Nakon toga robot prijeđe put od 500 mm i izvrši novo mjerenje sonarima. Iz novog mjerenja izgradi se set mrežastih karata zauzeća  $M = \{ m_1, m_2...m_9 \}$  u kojima koordinate robota predstavljaju set hipoteza H na slijedeći način:

- 1. pozicija na kojoj je obavljeno mjerenje je centralna pozicija i pripadnu mrežastu kartu zauzeća možemo označiti sa  $m_5$ , pri čemu robot ima koordinate (60,60) koje određuju hipotezu  $h_5$ ;
- ostalih 8 karata izgrađuje se oko centralne pozicije tako da se koordinate robota prividno mijenjaju po jedan segment u svakom

smjeru, odnosno, pretpostavka je da se robot ne nalazi na poziciji dobivenoj odometrijom. Npr., za izgradnju mrežaste karte  $m_1$  koordinate robota iznose (59,59) gledano prema karti  $m_5$ , te one određuju hipotezu  $h_1$  i uzimaju se u obzir pri izgradnji karte, ali prema karti  $m_1$  robot i dalje ima iste koordinate (60,60) (*slika* 5.3.);

iz dobivenih mrežastih karata zauzeća izračunaju se pripadni setovi histograma.



Slika 5.3. Izgradnja seta mrežastih karata M

#### 5.2.1. Usporedba histograma

Setovi *x* i *y* histograma  $his_i(1)$  dobivenih iz mrežaste karte prostora nakon prijeđenih 500 mm uspoređuju se sa *x* i *y* histogramom his(0) sa prošle lokacije pohranjene u memoriju. Funkcija koja pronalazi najbolje poklapanje između dva seta histograma  $T^a$  i  $T^b$  dana je izrazom:

$$Match(T^{a}, T^{a}) = \frac{1}{w} \sum_{j} \left[ \min(O_{j}^{a}, O_{j}^{b}) + \min(U_{j}^{a}, U_{j}^{b}) + \min(E_{j}^{a}, E_{j}^{b}) \right]$$
(5-2)

gdje *O*, *U* i *E* određuju broj zauzetih, nepoznatih i praznih segmenata na *j-om* elementu histograma, pri čemu je j=1,2...120, a konstanta w = 120x120 služi za normalizaciju rezultata tako da vrijedi  $0 \le Match() \le 1$ . U danoj funkciji pohranjeni histogram *his(0)* drži se stacionarnim, a novi histogram se translatira preko njega i izračunava se najbolje poklapanje svih 120 elemenata sa pohranjenim histogramom. Svi elementi novog histograma koji se zbog translacije ne poklapaju sa elementima starog histograma imaju vrijednost 0.5.



Slika 5.4. Usporedba x i y histograma

*Podudarnost L(s|h<sub>i</sub>)* izračuna se izrazom:

$$L(s \mid h_i) = M_x^{i^*} \times M_y^{i^*}$$
(5-3)

gdje su sa  $M_x^{i^*}$  i  $M_y^{i^*}$  najbolji rezultati preklapanja *x* i *y* histograma za lokaciju *i*. Najvjerojatnija hipoteza  $h_i$  određena je najvećom podudarnošću, a samim time mogu se odrediti i najvjerojatnije koordinate mobilnog robota slijedećim izrazima:

$$x_{hi} = x + r_{xi}$$
  

$$y_{hi} = y + r_{vi}$$
(5-4)

pri čemu su x i y globalne koordinate robota u trenutku mjerenja sonarima, a  $r_{xi}$  i  $r_{yi}$  su pomaci određeni dobitnom hipotezom. Npr., ako je potvrđena hipoteza  $h_1$ , tada je

estimirana stvarna pozicija robota pomaknuta za jedan segment u -*x* smjeru i jedan segment u -*y* smjeru te su  $r_{xi}$  i  $r_{yi}$  određeni veličinom segmenta. U ovom slučaju, ako se uzme da su globalne koordinate robota, kada je izvršeno mjerenje sonarima, bile x = 250 mm i y = 400 mm, tada su estimirane stvarne globalne koordinate robota jednake:

$$x_{h1} = x + r_{x1} = 250 - 50 = 200mm$$
  

$$y_{h1} = y + r_{y1} = 400 - 50 = 350mm$$
(5-5)

Nakon što se odredi dobitna hipoteza  $h_i$ , njeni histogrami  $his_i(1)$  pohranjuju se u memoriju i koriste za usporedbu sa setom novih histograma  $his_i(2)$  hipoteza  $h_i$  nakon prijeđenih daljnih 500 mm i novog mjerenja sonarima, a pozicija mobilnog robota osvježi se estimiranim koordinatama.

#### 5.3. Rezultati

#### 5.3.1. Simulacija

Rezultati simulacije dani su u *tablicama 5.1. i 5.2..* Mobilni robot giba se u simuliranom prostoru Zavoda za automatiku na 9. katu i ima početne globalne koordinate x = 22.7 m i y = 3.0 m, a zadana udaljenost koju mora prijeći je 7500 mm. Nakon što prijeđe zadanu udaljenost, očitavaju se globalne koordinate robota na novoj poziciji. U simulatoru (*poglavlje 2.2.2.*) se mogu očitati stvarne globalne koordinate mobilnog robota, dok se u *Saphirinom* grafičkom sučelju (*poglavlje 2.2.1.*) mogu očitati lokalne koordinate mobilnog robota koje se moraju prebaciti u globalne. Te koordinate sadrže odometrijsku pogrešku i zadatak razvijenog algoritma je da fuzijom informacija sa odometrije i sonara smanji tu pogrešku na najmanju moguću mjeru.



Slika 5.5. Vjerojatna distribucija pozicije robota

Na *slici 5.5.a* prikazan je model pogreške odometrije nakon translacijskog i rotacijskog gibanja. Na *slici 5.5.b* prikazana je velika translacijska, a na *slici 5.5.c* rotacijska pogreška.

U simulaciji je izvršeno 5 mjerenja pri čemu se u prvom slučaju ne koristi algoritam estimacije pozicije robota, dok se u drugom slučaju koristi gore opisani algoritam estimacije pozicije robota. Treba napomenuti da su stvarne globalne koordinate sa jednom decimalom izvorno u metrima, dok su u tablicama zaokružene vrijednosti u milimetrima. Korištena je mrežasta karta sa 240x240 segmenata.

Bez estimaciie	Stvarna	a pozicija	Pozicija dobivena odometrijom	
	X <sub>s</sub> [ mm ]	Y <sub>s</sub> [ mm ]	X <sub>o</sub> [ mm ]	Y <sub>o</sub> [ mm ]
1. mjerenje	15200	2700	15241	3000
2. mjerenje	15200	2500	15201	3000
3. mjerenje	15200	3100	15261	3000
4. mjerenje	15200	3200	15231	3000
5. mjerenje	15200	2800	15160	3000

Tablica 5.1. Rezultati simulacije bez estimacije pozicije robota

Sa estimacijom	Stvarna	a pozicija	Estimirana pozicija	
	X <sub>s</sub> [ mm ]	Y <sub>s</sub> [ mm ]	X <sub>e</sub> [ mm ]	Y <sub>e</sub> [ mm ]
1. mjerenje	15200	2800	15150	2825
2. mjerenje	15200	2800	15180	2800
3. mjerenje	15200	3000	15160	3000
4. mjerenje	15200	3200	15220	3175
5. mjerenje	15200	3200	15160	3150

Tablica 5.2. Rezultati simulacije sa estimacijom pozicije robota

	Bez estimacije		Sa estimacijom		
	δ <sub>xo</sub> [ % ]	δ <sub>yo</sub> [ % ]	δ <sub>xe</sub> [ % ]	δ <sub>ye</sub> [ % ]	
1. mjerenje	0.27	1.94	0.32	0.16	
2. mjerenje	0.01	3.25	0.13	0.00	
3. mjerenje	0.40	0.64	0.26	0.00	
4. mjerenje	0.20	1.28	0.13	0.16	
5. mjerenje	0.39	1.30	0.26	0.32	

Tablica 5.3. Relativna pogreška pozicije robota

Na slijedećem setu slika (5.5.a do 5.5.n) prikazane su dobitne mrežaste karte zauzeća (odnosno, hipoteze) pri estimaciji položaja mobilnog robota. Karte su izgrađene približno svakih 500 mm.







c.

g

Bayesian

3

E -





Slike 5.5.a – 5.5.n Mrežaste karte zauzeća korištene u simulaciji

*Slika 5.5.a* prikazuje početnu kartu prostora. Domet sonara podešen je na 3000 mm. Na *slici 5.5.c* i *d* sonari su detektirali prva otvorena vrata, a na slikama *k,l* i *m* druga otvorena vrata.

#### 5.3.1. Realan robot Pioneer 2

Rad algoritma isproban je na realnom robotu. Mjerenje na robotu *Pioneer 2* izvršeno je na pravocrtnoj putanji dužine 5 m u hodniku na 9. katu Zavoda za automatiku i procesno računarstvo *(slika 5.6)*. Korištena je mrežasta karta sa 120x120 segmenata. Rezultati mjerenja dani su u *tablici 5.4*. i *5.5*.



Slika 5.6. Hodnik Zavoda za automatiku

Realni robot	Stvarna	a pozicija	Estimirana pozicija	
	X <sub>s</sub> [ mm ]	Y <sub>s</sub> [ mm ]	X <sub>e</sub> [ mm ]	Y <sub>e</sub> [ mm ]
1. mjerenje	5080	152	5099	160
2. mjerenje	5095	280	5125	252
3. mjerenje	5100	100	5150	83
4. mjerenje	5098	138	5131	102
5. mjerenje	5067	115	5114	98

Tablica 5.4. Rezultati mjerenja na realnom robotu

Realni robot	Relativna	pogreška	Apsolutna pogreška		
	δ <sub>xe</sub> [ % ]	δ <sub>ye</sub> [ % ]	$\delta_{xea}$ [mm]	$\delta_{xea}$ [mm]	
1. mjerenje	0.38	0.16	19	8	
2. mjerenje	0.59	0.54	30	28	
3. mjerenje	0.98	0.33	50	17	
4. mjerenje	0.64	0.70	33	36	
5. mjerenje	1.12	0.34	57	17	

Tablica 5.5. Pogreška mjerenja na realnom robotu

Y koordinata ima uvijek pozitivnu vrijednost što je rezultat nagiba hodnika na desnu stranu mobilnog robota u pozitivnom smjeru *y* koordinate. Rezultati dobiveni na realnom robotu potvrđuju rezultate dobivene simulacijom.

# 6. ZAKLJUČAK

Da se postigne pouzdana samo-lokalizacija, mobilni robot mora se osloniti na sposobnost prepoznavanja prostora pomoću objekata u njemu, a ne na odometriju. Iako je odometrija relativno pouzdana samo za mjerenje udaljenosti, ali ne i za određivanje pozicije, najbolje rezultate samo-lokalizacije daje fuzija informacija sa percepcijskih senzora, poput sonara, i odometrije.

Mjerna nesigurnost, odnosno šum sonara ne predstavljaju veliki problem pri izgradnji mrežastih karata prostora, jer se slične pogreške u izgradnji prve karte nalaze i u svakoj slijedećoj karti pa, u konačnici, ne utječu puno na rezultat usporedbe. Utjecaji mjernog šuma smanjuju se i upotrebom načela vjerojatnosti u izgradnji mrežastih karti prostora.

Upotrebom histograma dobije se jedinstveni brojčani opis lokacije na kojoj je izvršeno mjerenje, a samim time drastično se smanjuju i zahtjevi na hardverske resurse jer nema više potrebe za manipulacijom i pohranom velikih mrežastim karata zauzeća. Pohrana i manipulacija vrši se samo sa lokalnim kartama i histogramima i u svakom slijedećem koraku stare karte i histogrami se osvježavaju novima.

Iz rezultata simulacije može se zaključiti da samo-lokalizacija na temelju odometrije nema smisla *(tablica 5.1.)* jer postoji velika pogreška u *y* koordinati koja je nastala rotacijskom pogreškom u gibanju mobilnog robota. Uzrok rotacijske pogreške može biti neravan teren i iako inkrementalni enkoderi ne bilježe promjenu brzina kotača, mobilni robot ipak izgubi pravocrtno gibanje. Fuzijom mjerenja sonara i odometrije može se detektirati promjena pravocrtnog gibanja te je pogreška u *y* koordinati znatno manja od 1% *(tablica 5.3.).* 

Rezultati simulacije potvrđeni su mjerenjem na realnom robotu *Pioneer 2*, iako je relativna pogreška veća nego ona dobivena simulacijom Jedini problem je hardverski slabo ugrađeno računalo u robotu koje nije moglo izvršavati algoritam u realnom vremenu (zastajkivanje robota), pa se mjerenje izvršilo sa mrežastom kartom zauzeća od 120x120 segmenata, za razliku od simulacije gdje je korišteno 240x240 segmenata. U ovom algoritmu nije implementirana estimacija rotacije ( $\theta$ ).

# 7. LITERATURA

- [1] Doc.dr.sc. Ivan Petrović, *Mobilna robotika predavanja*, Fakultet elektrotehnike i računarstva, Zagreb, 2002.
- [2] Olle Wijk, *Triangulation Based Fusion of Sonar Data with Application in Mobile Robot Mapping and Localization*, Royal Institute of Technology, Stockholm, 2001.
- [3] Sebastian Thrun, *Robotic Mapping: A Survey,* Carnegie Mellon University, Pittsburgh, 2002.
- [4] Tom Duckett, Ulrich Nehmzow, Mobile Robot Self-Localisation Using Occupancy Histograms and A Mixture of Gaussian Location Hypotheses, University of Manchester, Manchester, 2001.
- [5] Nicholas Roy, Sebastian Thrun, *Online Self-Calibration For Mobile Robots*, Carnegie Mellon University, Pittsburgh, 2002.
- [6] ActivMedia Pioneer 2 Operating Manual, ActivMedia Robotics, 2002.

# 8. SAŽETAK

Za razliku od drugih pristupa kod kojih je potrebno neposredno mjeriti stvarni položaj pri lokalizaciji robota, ovdje opisana metoda koristi senzore robota za automatsku lokalizaciju u toku rada mobilnog robota. Ova metoda kombinira novu tehniku usporedbe histograma dobivenih iz lokalnih mrežastih karti zauzeća sa algoritmom praćenja vjerojatnih lokacija robota tokom vremena. Metoda je potvrđena u simulaciji i u stvarnom svijetu pri čemu je smanjena odometrijska pogreška za red veličine.

Ključne riječi: lokalizacija mobilnog robota, histogrami, mrežaste karte zauzeća.

# ABSTRACT

In contrast to previous approaches, which require explicit measurements of actual motion when localisating a robot, the method proposed here uses the robot's sensors to automatically localisate the robot as it operates. This method combines a new scan matching technique, using histograms extracted from local occupancy grids, with an algorithm for tracking location hypotheses over time. The method was validated with experiments in simulation and in real world environment where the amount of odometric error is reduced by an order of magnitude.

Key words: Mobile Robot Localisation, Histograms, Occupancy grids.

Title: Local Occupancy Map Building using Ultrasonic Sensors

# 9. ŽIVOTOPIS

Rođen sam 21.12.1977. u Čakovcu. U Koprivnici sam pohađao osnovnu školu i Gimnaziju "Fran Galović", prirodoslovno-matematički smjer. Maturirao sam 1996. g. odličnim uspjehom i iste godine upisao sam Fakultet elektrotehnike i računarstva u Zagrebu. Na kraju 2. godine odabrao sam smjer automatika. Fakultet završavam u rujnu 2003. godine.