Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Robert Pečavar

Preiskovanje prostora z mobilno platformo in omejenimi senzorji

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: doc. dr. Danijel Skočaj

Ljubljana 2012

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00258/2012
Datum: 10.04.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ROBERT PEČAVAR**

Naslov: **PREISKOVANJE PROSTORA Z MOBILNO PLATFORMO IN OMEJENIMI SENZORJI**

EXPLORATION OF SPACE WITH A MOBILE PLATFORM AND LIMITED SENSORS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Zanesljivi samolokalizacija in navigacija po prostoru sta zelo pomembni sposobnosti mobilnih robotov, ki naj bi se samostojno premikali po prostoru. Pri tem so takšni roboti zelo odvisni od nabora senzorjev, ki so jim na voljo. Tipično se za določanje lege mobilnega robota uporabljajo podatki odometrije obogateni z dodatno informacijo, ki jo zajamejo z bolj naprednimi senzorji kot so laserski merilnik razdalje ali barvne in globinske kamere. V diplomski nalogi poizkusite problema lokalizacije in navigacije rešiti z bolj preprostimi senzorji, ki so na voljo na mobilni platformi iRobot Roomba. Izdelajte navigacijski algoritem, ki bo s pomočjo tako pridobljene informacije usmerjal mobilno platformo, da bo sistematično preiskala prostor. Z na strop pritrjeno kamero nato ocenite učinkovitost razvitih algoritmov tako, da primerjate kako hitro uspe mobilna platforma z uporabo različnih algoritmov preiskati čim večji del prostora.

Mentor:

doc. dr. Danijel Skočaj

Fakulteta za računalništvo in informatiko
UNIVERZA V LJUBLJANI
2

Dekan:

prof. dr. Nikolaj Zimic

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Robert Pečavar, z vpisno številko **63970116**, sem avtor diplomskega dela z naslovom:

Preiskovanje prostora z mobilno platformo in omejenimi senzorji

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Danijela Skočaja,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 24. septembra 2012 Podpis avtorja:

Zahvaljujem se mentorju doc.dr. Danijelu Skočaju za prijaznost in vso strokovno pomoč pri izdelavi diplomskega dela. Zahvaljujem se tudi moji ženi za podporo.

Kazalo

Povzetek

V diplomski nalogi je opisan razvoj aplikacije za sistematično preiskovanje prostora na robotskem sesalcu iRobot Roomba. Cilj razvoja aplikacije je bilo izboljšanje osnovnega delovanja sesalca Roomba, ki deluje nenačrtno znotraj pred-programiranih načinov obnašanja. S pomočjo okolja Robot Operating System (ROS) in sistematičnega načina obnašanja se je razvilo aplikacijo, ki je preko serijskega vmesnika nadzorovala sesalec Roomba. Za to se je uporabilo samo že vgrajene senzorje. Aplikacija je vsebovala dva načina obnašanja. Prvi način je uporabljal senzorje samo za odkrivanje ovir. Drugi način pa je senzorje uporabljal tudi za preračunavanje gibanja. Temu je sledilo testiranje aplikacije v praksi in na koncu primerjava z osnovnim delovanjem sesalca.

Ključne besede: iRobot Roomba, ROS, preiskovanje prostora, senzorji

Abstract

This thesis will discuss the development of application for systematic exploration of space on iRobot Roomba robotic vacuum cleaner. The goal of development of application was to improve the basic operation of Roomba vacuum cleaner. Basic operation works unplanned within preprogrammed behaviors. The application for controlling the Roomba vacuum cleaner with the use of serial interface was developed with the help of Robot Operating System (ROS) environment and systematic pattern of behavior. The application was using only the integrated sensors. The application included two ways of behavior. The first way used sensors for detecting obstacles and the second one used the sensors to calculate the movement. The application was later tested in real test environment and compared with the basic operation of the vacuum cleaner.

Keywords: iRobot Roomba, ROS, exploration of space, sensors

Poglavje 1

Uvod

Človeštvo si že od nekdaj prizadeva olajšati vsakodnevna opravila. Za ta namen so bila razvita različna orodja in pripomočki. Eden od zadnjih orodij so roboti. Ti so si v zadnjem času utrli pot tudi v gospodinjstva. Zoprno delo sesanja prostorov je olajšal prihod robotskih sesalcev. Ti inteligentni sesalci nam prihranijo čas, ker samostojno opravljajo rutinsko delo, kar nam omogoča, da ta čas posvetimo drugim opravilom ali počitku.

Eden od pionirjev na tem področju je bilo podjetje iRobot s svojo družino robotskih sesalnikov Roomba. Njihovi novejši sesalci preko računalniškega vmesnika omogočajo uporabniku dostop do vseh njihovih zmožnosti.

Kljub vsem prednostim takega sesalca se v praksi izkaže, da ima tudi svoje pomanjkljivosti. Zaradi vgrajenih osnovnih senzorjev in omejenega načina delovanja se pojavlja več težav. V prostoru lahko ostanejo mesta, ki jih sesalec ne posesa. Sesalec se zadržuje samo v enem delu prostora. Sesalec se večkrat vrača na isti del prostora, ostale pa ignorira. Pri opazovanju samega delovanja je moteče dejstvo, da v določenem trenutku sesalec deluje organizirano, v naslednjem pa čisto naključno. To je pogojeno s tem, da sesalec izbira med nekaterimi prednastavljenimi načini delovanja.

Na trgu že obstajajo modeli robotskih sesalcev, ki se tega problema lotevajo z izdelavo zemljevidov prostorov. Ti uporabljajo dodatne, bolj napredne senzorje, s pomočjo katerih med delovanjem gradijo zemljevid prostora. Iz

zemljevida sesalec ve, kje v prostoru se nahaja in katere dele prostora je že posesal. Slabost tega pristopa je, da dodatno podražijo napravo.

V diplomski nalogi smo se osredotočili na cenejši, bolj dostopen model sesalca Roomba. Cilj diplomske naloge je napisati aplikacijo za vodenje sesalca po prostoru. Ta aplikacija bi bolj načrtno in s tem tudi bolj učinkovito posesala prostore. Pri tem bi uporabila samo senzorje, ki so že vgrajeni v sesalec Roomba. Na koncu sledi primerjava z običajnim delovanjem sesalca.

V drugem poglavju si bomo ogledali sesalec, njegove senzorje, opremo, ki je potrebna za komunikacijo, programsko opremo in osnovno delovanje Roombe. V drugem poglavju je opisan način premikanja in aplikacija za krmiljenje. Tretje poglavje pa opisuje proces testiranja.

Poglavje 2

Mobilna platforma

2.1 iRobot Roomba

Robotski sesalec je avtonomna naprava, ki se je s pomočjo senzorjev zmožna izogibati oviram v prostoru. iRobot Roomba (na Sliki 2.1) je ena izmed najpopularnejših družin robotskih sesalcev na tržišču. Prvič so se predstavili leta 2002, trenutno so že v tretji generaciji. V tretji generaciji je več modelov. Od osnovnega modela, do modelov, ki znajo uporabljati virtualne svetilnike za sprehajanje med prostori in imajo časovno nastavljivo čiščenje. Za diplomsko nalogo je bil uporabljen osnovni model Roombe tretje generacije, model 521, ki omogoča dostop do elektronike preko serijskega protokola ROI (Roomba Open Interface). Dostop je omogočen preko Mini-DIN konektorja, ki se nahaja pod pokrovom na vrhu sesalca.

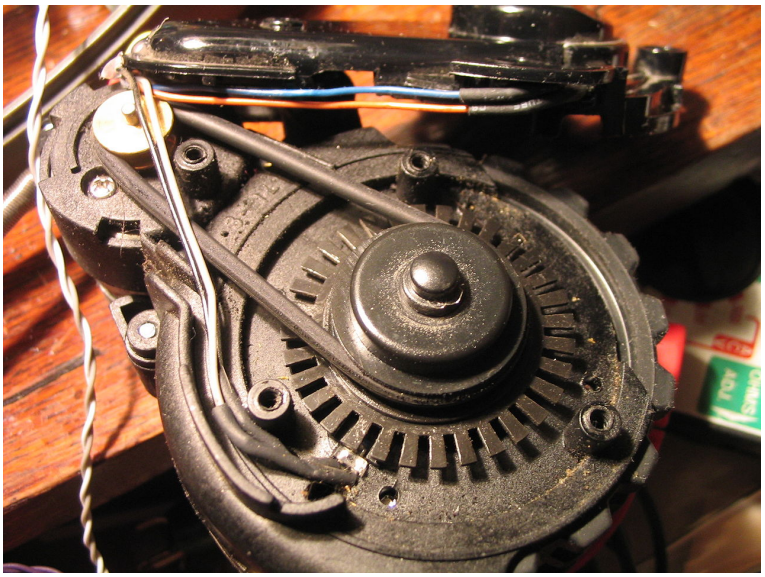
Slika 2.1: iRobot Roomba [6]

2.2 Senzorji

Sesalec Roomba ima vgrajenih 38 senzorjev ki so razporejeni v več skupin:

- odmetrija - meri razdaljo, ki so opravijo kolesa,
- odbijač - zaznava ovire z dotikom,
- infrardeči senzorji - zaznavajo ovire brez dotika,
- talni senzor - zaznava oddaljenost od tal in preprečuje padec po stopnicah,
- gumbi - Roomba ima več gumbov, ki se lahko uporabijo v aplikacijah,
- senzor za padec kolesa - zaznava, kdaj kolo nima več stika s podlago,
- senzor za baterijo - nadzoruje napetost baterije in tok, ki ga zahtevajo motorji,
- senzor za zaznavanje umazanije - zaznava območja tal z večjo umazanijo.

Za potrebe diplomske naloge smo uporabili tri tipe senzorjev: odometrijo, odbijač in infrardeče senzorje. Senzor za odometrijo nam sporoča razdaljo, ki jo opravi pogonsko kolo na sesalcu. Je elektro-mehanska naprava, ki spreminja rotacijsko gibanje osi kolesa v elektronske impulze. Iz teh impulzov se lahko s pomočjo dodatnih parametrov (premer kolesa, število pulzov na premer) izračuna kolikšno razdaljo kolo prevozi med vsakim impulzom. Senzor je sestavljen iz zobate plošče, ki je pritrjena na os kolesa in para foto detektor-izvor svetlobe. Foto detektor zaznava svetlobo, ki jo oddaja vir svetlobe (navadno LED dioda). Ko se zobata plošča vrti, prekinja tok svetlobe in preko foto detektorja ustvarja električne pulze, ki se jih nato uporabi za izračun prevožene razdalje. Senzor za odometrijo na Roombi je prikazan na Sliki 2.2.



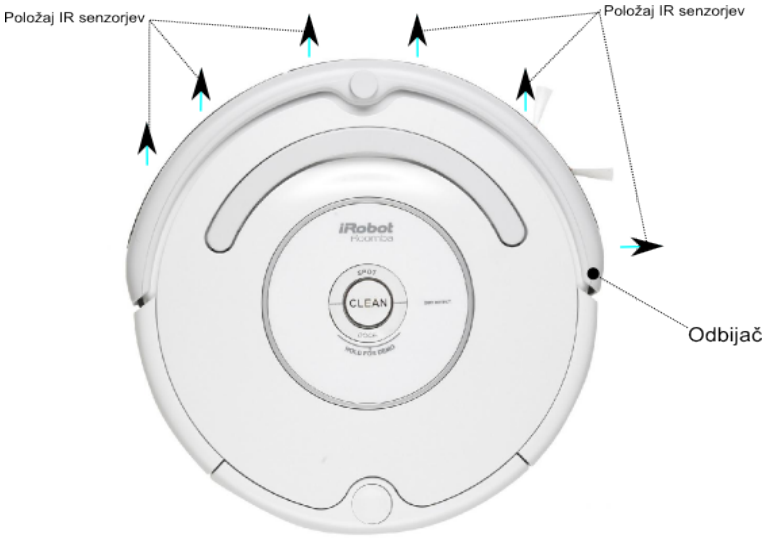
Slika 2.2: Senzorji za odometrijo [10]

Odbijač je pritrjen na sprednji strani sesalca in ima dva senzorja, ki sta občutljiva na dotik. Senzorja zaznavata dotike z desne ali leve strani. Ob dotiku s sprednje strani pa se aktivirata oba.

Roomba ima šest infrardečih senzorjev, ki so postavljeni v dveh skupinah (leva in desna) na zunanjem sprednjem obodu sesalca. Senzorji v levi skupini so vsi usmerjeni v smer gibanja. V desni skupini pa sta dva usmerjena v smer gibanja, skrajni desni je usmerjen pravokotno na smer gibanja. Pozicije senzorjev so označene na Sliki 2.3.

Infrardeči senzor deluje na principu odboja svetlobe. Vir infrardeče svetlobe oddaja pulze svetlobe, ki jih foto detektor poskuša ujeti. Ko se svetloba odbije od predmeta in jo foto detektor zazna, senzor sporoči, da je predmet v dometu.

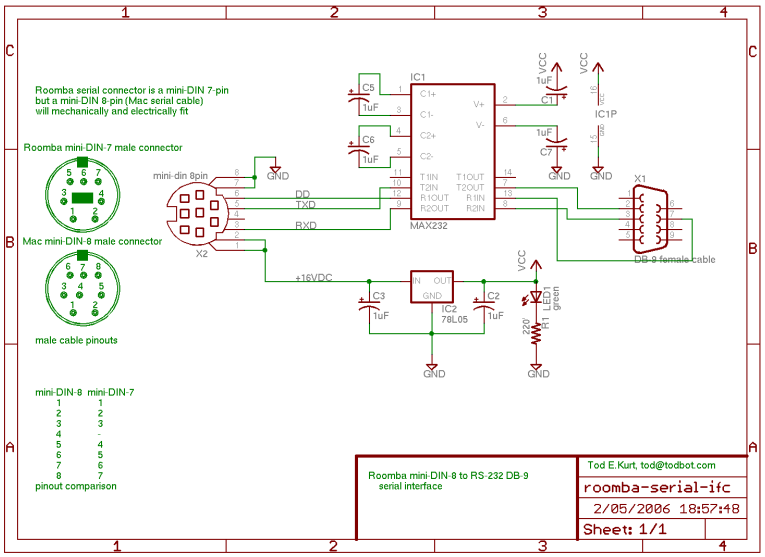
Natančnost infrardečih senzorjev je odvisna predvsem od površine. Pri površinah, ki ne odbijajo zadostne količine svetlobe se lahko zgodi, da jih senzor ne zazna.



Slika 2.3: Senzorji na Roombi

2.3 Serijski vmesnik

Ker Roomba podpira samo signale TTL (napetosti med 0 in 5V), je za priklop na osebni računalnik potrebno uporabiti vmesnik. Najprej smo uporabili RS232 serijski vmesnik proizvajalca Bitbox, ki se je nato izkazal za pomanjkljivega. Vmesnik je zaradi Roombine omejitve pri jakosti signala omogočal samo enosmerno komunikacijo v smeri proti Roombi. Zato smo namesto tega za dvosmerno komunikacijo kasneje uporabili vmesnik, ki smo ga izdelali v samogradnji [2]. Vmesnik smo zgradili na prototipni plošči, ki omogoča hitro izgradnjo elektronskih vezij. Na ploščo smo prispajkali regulator napetosti, vezje MAX232 in ostale elemente potrebne za delovanje. Na vmesnik smo na koncu prispajkali še kabla, ki skrbita za povezavo na osebni računalnik in na Roombo. Roomba za priklop uporablja nestandardni mini-DIN konektor s sedmimi kontakti, zato smo za naš pretvornik uporabili mini-DIN konektor z osmimi kontakti, kateremu smo odstranili en kontakt. Za priklop na osebni računalnik smo uporabili standardni serijski konektor z devetimi kontakti. Shemo vmesnika prikazuje Slika 2.4. Vmesnik uporablja vezje MAX232, ki



Slika 2.4: Shema serijskega vmesnika [2]

skrbi za pretvarjanje signalov iz nivoja RS232 (0-12V) na TTL nivo, ki ga uporablja Roomba. Vezje omogoča dva sprejemna in dva oddajna kanala. Uporabili smo oba oddajna kanala in enega sprejemnega.

Na tržišču je veliko programskih paketov, ki omogočajo krmiljenje robotov. Eden izmed njih je tudi Robot Operating System.

2.4 ROS

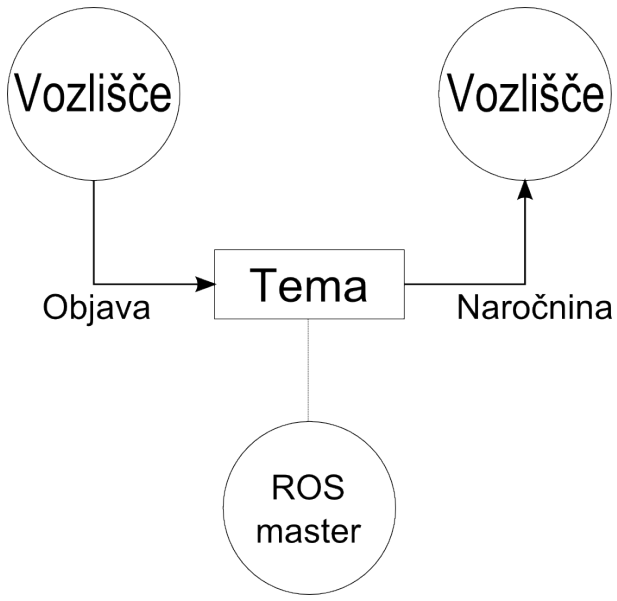
Robot Operating System (ROS) [9] je odprtokodni razvojni sistem, ki ponuja funkcionalnost operacijskega sistema, namenjenega robotiki. Originalno je bil razvit leta 2007 v Stanford Artificial Intelligence Laboratory, leta 2008 ga je prevzelo podjetje Willow Garage.

ROS nudi vso podporo za razvoj robotskih aplikacij. Deluje na principu omrežja enakovrednih vozlišč, kjer posamezna vozlišča v omrežju komunicirajo med sabo. *Vozlišče* (ang. node) je aplikacija, ki opravlja določeno funkcijo (nadzor senzorja, izračuni, premikanje...). Vozlišča komunicirajo med sabo s *sporočili* (ang. message), ki se objavljajo na *temah* (ang. topic).

Sporočila so podatkovna struktura, ki je sestavljena iz polj. Polje je lahko osnovni podatkovni tip ali niz osnovnih tipov. Vozlišče objavi sporočilo na temi, na kateri poslušajo ostala vozlišča. Teme se ločijo med seboj s poimenovanji. Vozlišče, ki je zainteresirano za pridobivanje sporočil z določeno tematiko, se naroči na ustrezno temo. Na določeni temi lahko hkrati objavlja ali posluša več vozlišč. Vozlišče pa lahko pošilja ali sprejema sporočila na več temah hkrati. Shema delovanja je predstavljena na Sliki 2.5. Vsa vozlišča nadzoruje ROS Master, ki vodi seznam vseh trenutnih tem in s tem omogoča komunikacijo in lokalizacijo med vozlišči.

Osnovna organizacijska enota v sistemu ROS je *paket* (ang. package). Ta združuje vozlišča, knjižnice, sporočila, nastavitvene datoteke, ter vse ostalo, kar tvori zaključeno celoto. Paketi, ki vsebujejo sestavne dele za večje celote, na primer navigacijo, so združeni v *skladih* (ang. stack).

V ROS je vključenih več skladov. Ti pokrivajo področja navigacije, premikanja, nadzora, zaznavanja... Aplikacije za ROS se lahko razvija v dveh programskih jezikih: C++ in Python.



Slika 2.5: Osnovno delovanje sistema ROS

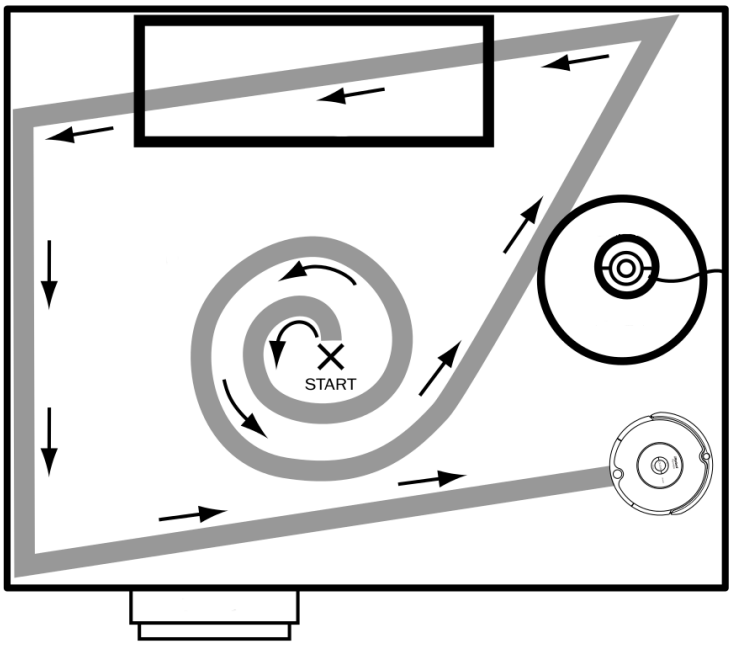
2.5 Osnovno delovanje sesalca Roomba

Eden izmed ciljev diplomske naloge je bila tudi primerjava delovanja naše aplikacije z Roombinim privzetim načinom delovanja. V tem podpoglavju ga bomo na kratko opisali.

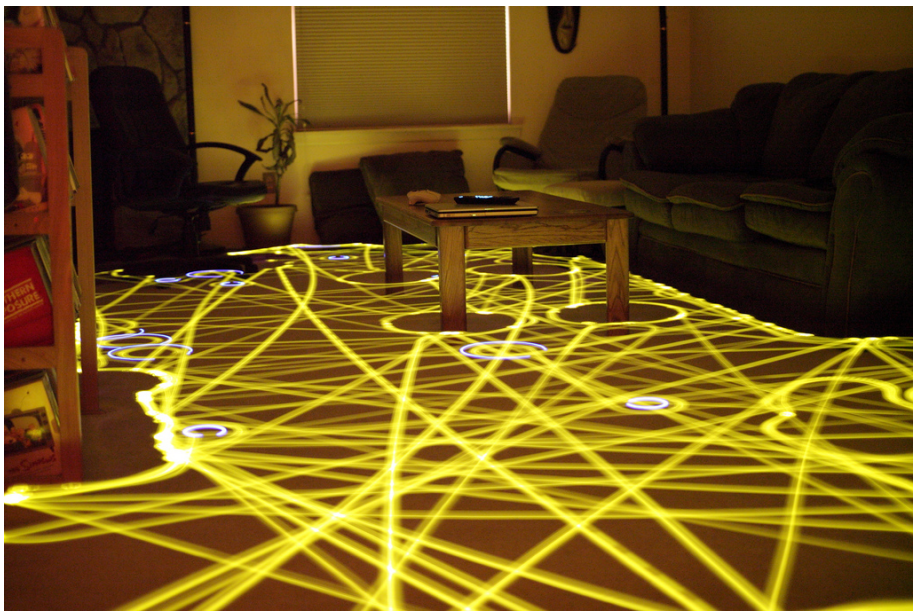
Sesalec Roomba izračunava pot, po kateri sesa prostor med samim delovanjem. Pri tem naključno izbira med nekaterimi vgrajenimi načini obnašanja. Vgrajeni načini obnašanja so [4]:

- spiralno gibanje; Roomba uporablja spiralen vzorec za čiščenje dela prostora. Sesalec se v krožnem gibanju vrti okrog točke v prostoru v vedno večjem radiju.
- sledenje zidu; Roomba v tem načinu sledi robu prostora in oviram. Sesalec se s pomočjo infrardečega senzorja na desni strani premika vzporedno s steno.
- prečkanje sobe; Roomba križemkražem prečesava sobo. Sesalec naključno izbere kot, se obrne in se premika do ovire. Nato spet izbere naključen kot in ponovi premikanje.

Načini so prikazani na Sliki 2.6. Slika 2.7 prikazuje primer poti, ki jo Roomba naredi v dejanskem prostoru.



Slika 2.6: Osnovno delovanje sesalca Roomba [4]



Slika 2.7: Primer dejanske poti Roombe [10]

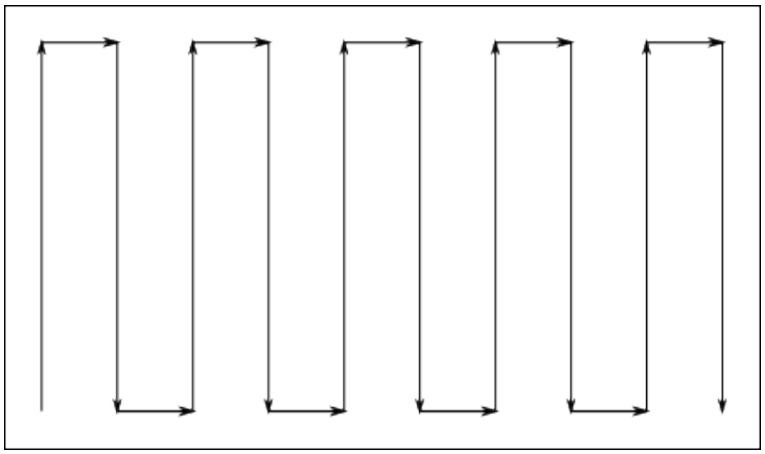
Poglavje 3

Zasnova in razvoj sistema

Osnovno obnašanje sesalca Roomba ima nekaj pomanjkljivosti. Za njim lahko ostanejo neposesana mesta, sesalec se zadržuje samo v enem delu prostora oziroma se večkrat vrača na isti del prostora, ostale pa ignorira. Osnovno vedenje sesalca Roomba smo poskušali izboljšati z novim načinom premikanja, ki bi bolj učinkovito pokril površino sesanja in uporabljal samo vgrajene senzorje.

3.1 Boustrofedonov način gibanja

Za preiskovanju prostora smo izbrali način premikanja, ki se imenuje bustrofedon [3]. V grščini ta izraz pomeni volovsko obračanje. Premikanje sesalca je podobno premikanju volov pri oranju. Ko vola prideta do konca brazde, se obrneta in nadaljujeta oranje na vzporedni brazdi v nasprotni smeri. Način gibanja je prikazan na Sliki 3.1. Ta način omogoča bolj sistematično in deterministično premikanje po prostoru. Pogoj za čim boljše pokritje prostora je, da se sesalec giblje pravokotno na prostor tako, da so posamezni premiki čim bolj vzporedni.



Slika 3.1: Boustrofedonov način gibanja

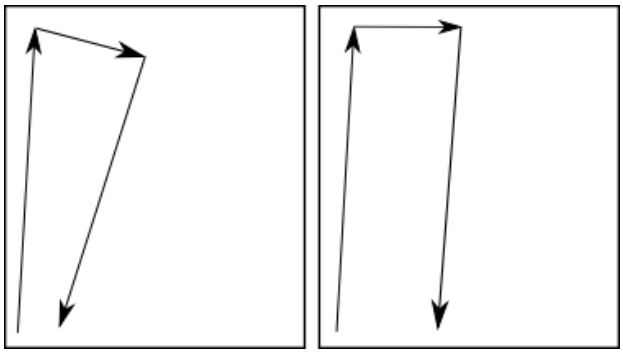
3.2 Nova načina obnašanja

Aplikacija za vodenje sesalca Roomba je sestavljena iz dveh niti: glavne in stranske. V glavni niti se izvajajo ukazi za premikanje, v stranski pa se berejo podatki iz senzorjev. V aplikaciji smo sprogramirali dva načina delovanja, ki se ju izbira s parametrom pri zagonu:

- zaznavanje sten z odbijačem,
- zaznavanje sten z odbijačem in s popravki z infrardečimi senzorji.

Pri načinu brez popravkov smeri se zaradi nenatančnosti senzorjev za odometrijo v praksi izkaže, da vzorec gibanja ne poteka pravokotno na prostor (na Sliki 3.2(a)). Zaradi tega smo izdelali še drugi način, ki poskuša to pomanjkljivost odpraviti s popravljanjem smeri (na Sliki 3.2(b)).

Prvi način delovanja: sesalec se giblje v bustrofedonovem vzorcu premikanja. Gibanje je sestavljeno iz dveh akcij: glavne in stranske. Glavna akcija je sestavljena iz premikanja naravnost po korakih z določeno hitrostjo do zaznane stene. Pred steno se upočasni do dotika odbijača. Ob dotiku se ustavi in zapelje nazaj za nekaj centimetrov. Premik nazaj je pomemben



(a) Način brez popravkov smeri

(b) Način s popravki smeri

Slika 3.2: Prikaz razlike med načini delovanja

zaradi sprostitve odbijača. V nasprotnem primeru senzor stalno javlja oviro. Nato se na mestu obrne z določeno hitrostjo obračanja za kot 90 stopinj in izvede stransko akcijo. Ta je sestavljena iz premikanja po korakih za širino sesalca naprej. Na koncu spet sledi obrat za kot 90 stopinj. Ti dve akciji se izmenjujeta tako, da je smer glavne akcije vedno obratna od prejšnje smeri. Aplikacija si smer shranjuje in izbira novo smer glede na prejšnjo. Hitrost linearnega gibanja, hitrost obračanja in dolžina koraka so podani kot parametri pri zagonu.

Drugi način delovanja: sesalec se prav tako giblje v bustrofedonovem vzorcu premikanja s to razliko, da se sedaj ob približevanju steni shrani stanje infrardečih senzorjev. Po dotiku odbijača se iz stanja senzorjev ugotovi, pod kakšnim kotom se je sesalec približal steni in se nato izračuna nov kot zasuka.

Pri obeh načinih delovanja se sesalec v primeru, ko se znajde v situaciji, kjer ne more več izvesti gibanja po bustrofedonovem vzorcu (na primer vogal prostora), namesto obrata za 90° zavrti za 180° in nadaljuje pot.

Samo gibanje je izvedeno s pomočjo dveh funkcij v aplikaciji. Prva skrbi za linearno gibanje. Kot parametra se poda smer premika in hitrost. Druga funkcija skrbi za rotacijsko gibanje, s podano smerjo zasuka, kotom zasuka in hitrostjo. Funkciji v zanki pošiljata ukaze za premikanje gonilniku, ki

poskrbi za dejanske premike sesalca. Funkcija za rotacijsko gibanje podatke, pridobljene iz senzorjev za odometrijo uporablja za to, da oceni kot zasuka. Premikanje se ustavi, ko se glede na podatke iz odometrije doseže kot zasuka, ki je bil podan ob klicu funkcije.

Za gonilnik, ki skrbi za povezavo sistema ROS in dejanskega sesalca smo izbrali paket roomba_500_series [8]. Gonilnik sprejema ukaze za premikanje in skrbi, da se podatki iz senzorjev objavljajo na njihovih temah.

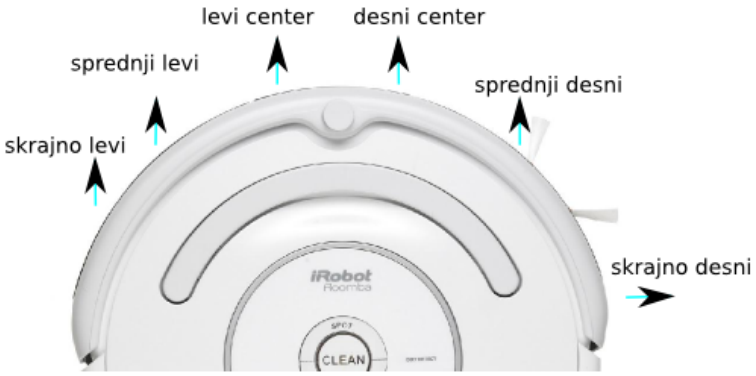
3.3 Izračun kota zasuka

Podatki iz senzorjev (infrardeči in odbijač) se berejo v ločeni niti. Pri prvem načinu delovanja se pri odbijaču in infrardečih senzorjih preverja samo, če so sproženi. Sprožen odbijač pomeni, da je sesalec zadel steno, kar povzroči ustavitev in premik nazaj. Pri infrardečih senzorjih sprožitev kateregakoli senzorja izmed šestih pomeni, da se sesalec približuje steni in je potrebna upočasnitev približevanja. To prepreči nalet v steno s polno hitrostjo.

Pri drugem načinu delovanja si aplikacija v trenutku, ko sesalec zadane steno, zapomni kateri izmed infrardečih senzorjev so bili sproženi. Iz tega podatka se lahko približno oceni, pod kakšnim kotom se je sesalec približeval steni. Koti so bili izmerjeni eksperimentalno, s pomočjo Roombe, postavljene pred ravno steno. Med rotacijo Roombe pred steno se je spremljalo kombinacije sproženih senzorjev. Pri spremembah kombinacij se je nato izmerilo kote, ki jih pokriva določena kombinacija. Iz izbranega območja se je za našo aplikacijo kasneje izbrala vrednost, pri kateri sesalec najbolje popravi smer. Senzorji so poimenovani: skrajno levi, sprednji levi, levi center, desni center, sprednji desni in skrajno desni (Slika 3.3). Rezultati meritev so predstavljeni v Tabeli 3.1.

Iz dobljenih podatkov se v aplikaciji izračuna za kolikšen kot se mora sesalec obrniti ob koncu glavnega giba, da se giblje čim bolj pravokotno na prostor oziroma se v stranskem gibu premika vzporedno s steno.

Če se sesalec približuje steni, ki je na levi strani sesalca, se novi kot zasuka



Slika 3.3: Oznake infrardečih senzorjev

Kombinacija senzorjev - levi	Kot v stopinjah	Izbran kot
skrajno levi	10-30	30
skrajno levi in sprednji levi	30-55	45
skrajno levi, sprednji levi in levi center	55-90	60
Kombinacija senzorjev - desni	Koti v stopinjah	Izbran kot
skrajno desni	0-12	5
skrajno desni in sprednji desni	15-35	30
skraj. desni, spred. desni in desni cent.	35-60	45
sprednji desni in desni center	60-90	65

Tabela 3.1: Tabela meritev senzorjev

β izračuna glede na zahtevan zavoj po Enačbi (3.1). Če sesalec zavija levo, je novi kot β enak kotu α, pridobljenem iz senzorjev. To je prikazano na Sliki 3.4(a). Pri zavoju na desno je novi kot β enak kotu π zmanjšanem za kot α (Slika 3.4(b)).

Kadar se sesalec približuje steni z desne strani, sta izračuna obrnjena (Enačba (3.2)). Pri levem zavoju novi kot β dobimo iz kota π zmanjšanega za kot α (Slika 3.5(a)). Novi kot β za desni zavoj pa je enak kotu α (Slika 3.5(b)).

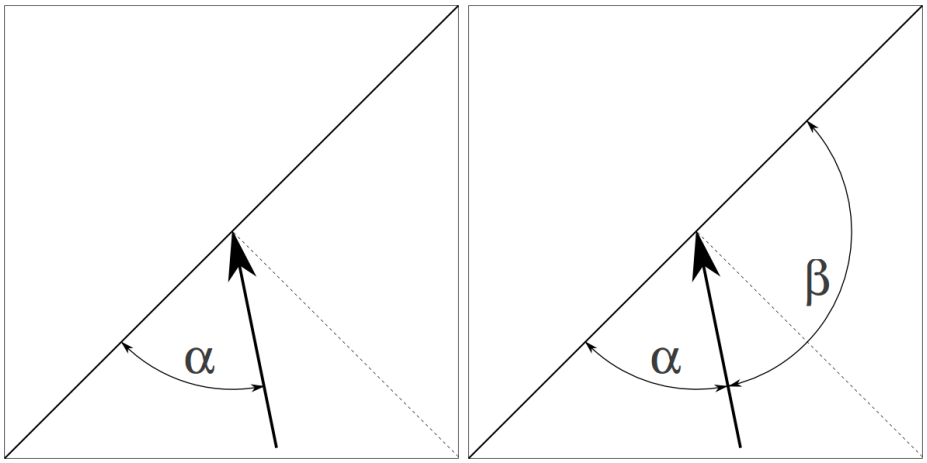
Enačba za izračun popravkov smeri:

$$\beta = \begin{cases} \alpha & \text{obračanje v desno} \\ \pi - \alpha & \text{obračanje v levo} \end{cases} \tag{3.1}$$

$$\beta = \begin{cases} \Pi - \alpha & \text{obračanje v desno} \\ \alpha & \text{obračanje v levo} \end{cases} \tag{3.2}$$

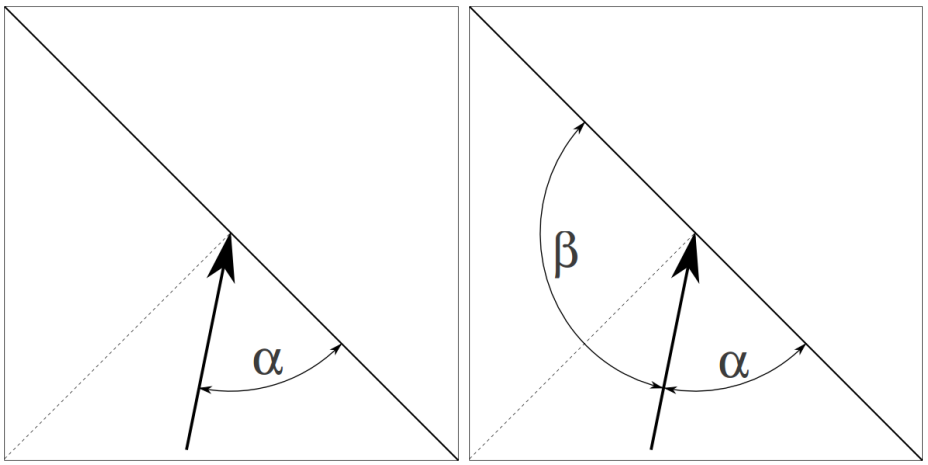
kjer je:

- α: kot približevanja steni, pridobljen iz senzorjev,
- β: novi kot zasuka.



(a) Stena na levi strani - levi zavoj (b) Stena na levi strani - desni zavoj

Slika 3.4: Prikaz kotov - Stena na levi strani



(a) Stena na desni strani - desni zavoj (b) Stena na desni strani - levi zavoj

Slika 3.5: Prikaz kotov - stena na desni strani

Zgoraj opisana načina delovanja smo na koncu izdelali v sistemu ROS. Temu je sledilo testiranje algoritma z odpravo napak in prilagajanje parametrov aplikacije. Na koncu je sledilo le še ovrednotenje aplikacije, ki je opisano v naslednjem poglavju.

Poglavje 4

Eksperimentalni rezultati

Sistem smo ovrednotili v dveh načinih delovanja. Prvi način je uporabljal samo odbijače za zaznavanje sten, drugi način je poleg odbijačev uporabil še infrardeče senzorje za izračunavanja popravkov smeri. Samo vrednotenje smo izvedli s kamero, ki je bila pritrjena na strop, in z aplikacijo, ki je merila pokritost površine na video posnetku, pridobljenem iz kamere. Rezultate smo na koncu primerjali s pokritostjo površine pri Roombinem osnovnem delovanju.

4.1 Merjenje učinkovitosti delovanja

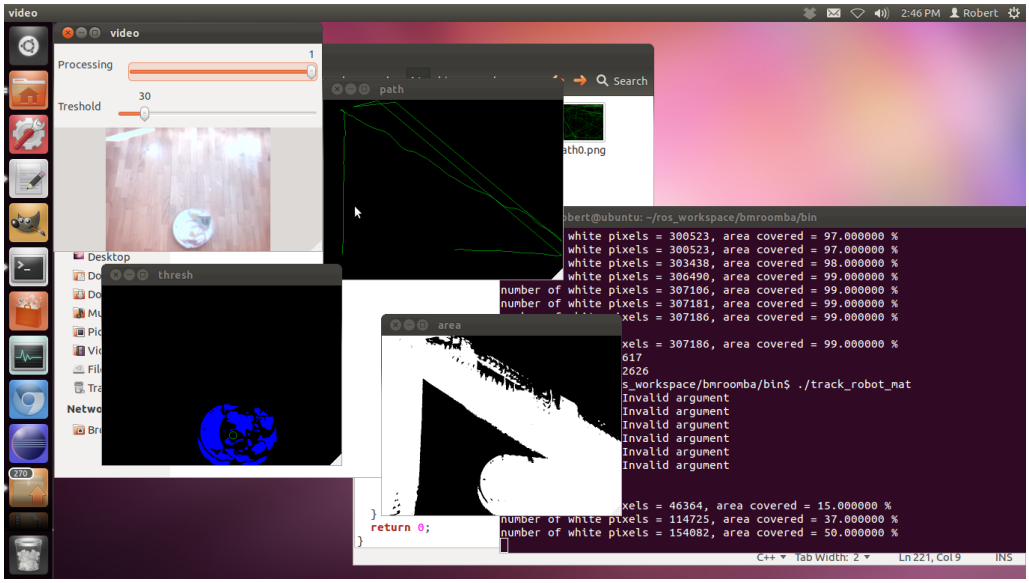
Vrednotenje se je izvedlo na ograjeni pravokotni površini treh različnih velikosti. Za snemanje smo uporabili spletno USB kamero Sony EyeToy z resolucijo 640x480 pik (Slika 4.1). Kamera je bila pritrjena na določeni višini tako, da je pokrivala celotno površino. Kamera je bila pred uporabo skalibirirana zaradi odprave popačenja slike. Kalibracija je bila opravljena z uporabo paketa camera_calibration [5], ki je sestavni del sistema ROS. Proces poteka s pomočjo šahovnice znanih dimenzij (velikost in število polj na šahovnici), ki jo premikamo pred kamero. Šahovnico se premika v vertikalni in horizontalni smeri pod različnimi nagibi. S pomočjo tega aplikacija za kalibracijo dobi dovolj podatkov za odpravo popačenja kamere.

Slika 4.1: Spletna kamera Sony

Aplikacija za vrednotenje je bila napisana v jeziku C++ z uporabo knjižnice OpenCV. OpenCV [7] je knjižnica programskih funkcij, ki so namenjene predvsem področju računalniškega vida. Osredotoča se v glavnem na obdelavo slik v realnem času.

Uporabniški vmesnik aplikacije (Slika 4.2) je sestavljen iz štirih oken. Glavno okno vsebuje živo sliko iz kamere, drsnik za nastavitev praga in drsnik za vklop meritve. Ostala okna prikazujejo bitno sliko obrisa, ki ga zazna kamera, celotno pokritost površine in pot centroida. Po končanem merjenju aplikacija izpiše procent pokritosti prostora in pretečeni čas od začetka meritve.

Aplikacija je na začetku inicializirala USB kamero in začela zajemati barvne slike. Barvne slike se je nato pretvorilo v sivinske slike in se jih shranjevalo v začasne matrike. Iz absolutnih razlik [1] med dvema zaporednima sivinskima slikama se je dobilo obris premikajočega se objekta. Absolutna razlika je izračunana kot absolutna vrednost razlike med dvema matrikama za vsak element matrike posebej. Ker je tako pridobljena slika razlik vsebovala veliko šuma, se jo je pretvorilo v bitno sliko. Vsaka točka slike razlik, ki je padla pod prag, ki se ga določi v glavnem oknu aplikacije, se je obarvala črno. Vse ostale točke pa se je obarvalo z belo barvo. Tako pridobljena bitna slika je vsebovala samo gibanja sesalca, brez šumov. Gibanju sesalca se je



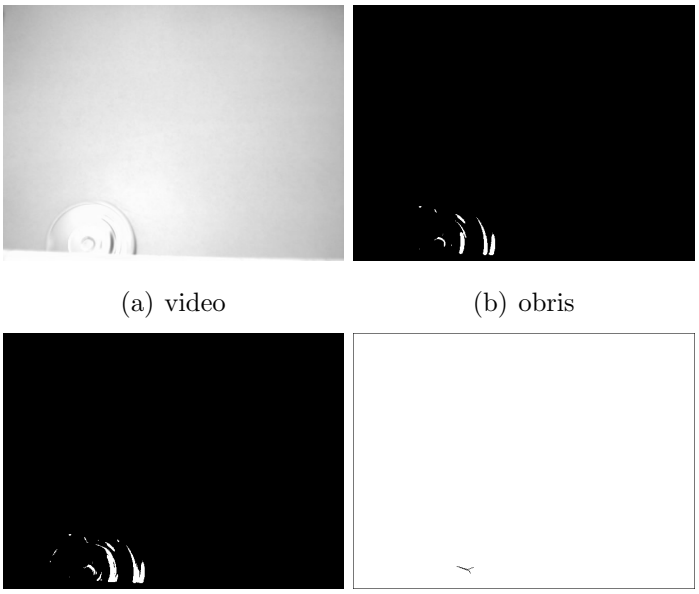
Slika 4.2: Uporabniški vmesnik

sledilo na dva načina.

Prvi način je sledenje centroidu obrisa sesalca. Iz obrisa je aplikacija izračunala koordinate centroida. Koordinate centroida so se izračunale kot vsota koordinat vseh točk obrisa v horizontalni in vertikalni smeri. Dobljeni vsoti se je delilo s številom vseh točk obrisa. Iz razlike med koordinatami centroida med dvema zaporednima slikama se je izračunalo vektor premika. S pomočjo vektorjev se je v okno programa risala pot, ki jo je opravil sesalec.

Drugi način je točke obrisa v vsaki sliki shranjeval v začasno matriko. Iz teh točk se je dobilo sled, ki jo je za sabo puščal sesalec. Pokritost površine se izračunava kot razlika med celotno površino slike in površino, ki jo pokriva sled sesalca.

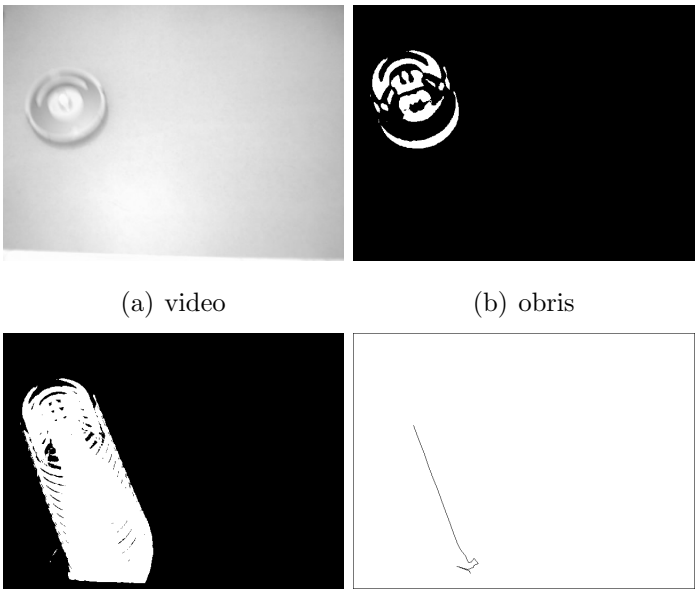
Na Slikah 4.3, 4.4, 4.5, 4.6 je prikazano delovanje aplikacije v zaporednih časovnih intervalih. Sklop (a) prikazuje sliko, zajeto iz kamere, sklop (b) prikazuje obrise sesalca, sklop (c) prikazuje pokritost prostora in sklop (d) prikazuje pot sesalca.



(a) video (b) obris

(c) pokritost (d) pot

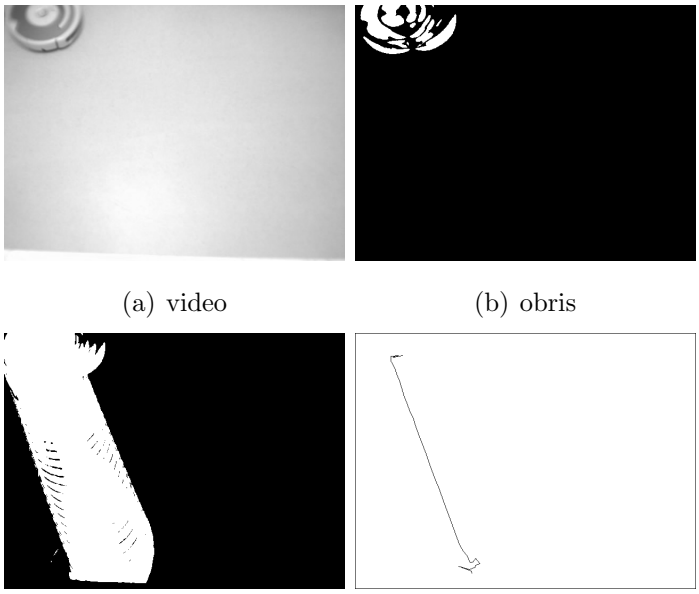
Slika 4.3: Delovanje aplikacije: prvi časovni interval



(a) video (b) obris

(c) pokritost (d) pot

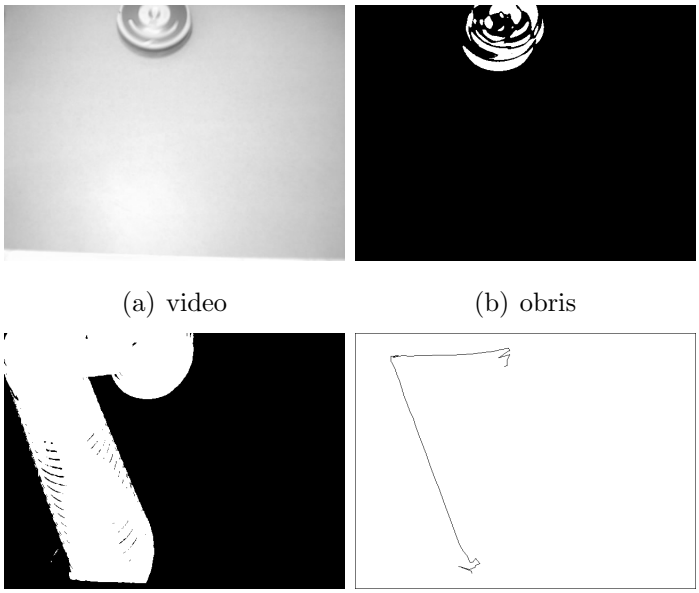
Slika 4.4: Delovanje aplikacije: drugi časovni interval



(a) video (b) obris

(c) pokritost (d) pot

Slika 4.5: Delovanje aplikacije: tretji časovni interval



(a) video (b) obris

(c) pokritost (d) pot

Slika 4.6: Delovanje aplikacije: četrti časovni interval

4.2 Eksperimentalni poligon

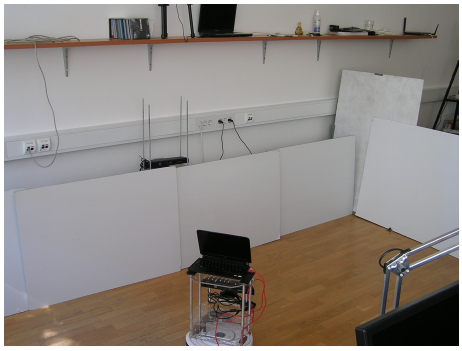
Eksperimentalni poligon je bil sestavljen iz treh testnih površin različnih velikosti. Površine so bile ograjene z panoji tako, da se je sesalcu omejilo gibanje. Velikosti površin, na katerih se je izvedlo vrednotenje, so bile:

1. mala površina: 180cm x 115cm, višina kamere 165cm,
2. srednja površina: 230cm x 175cm, višina kamere 200cm,
3. velika površina: 260cm x 230cm, višina kamere 270cm.

Testne površine so predstavljene na Slikah 4.7(a), 4.7(b) in 4.7(c).

(a) Mala testna površina

(b) Srednja testna površina



(c) Velika testna površina

Slika 4.7: Testne površine

4.3 Eksperimentalni rezultati

Eksperiment je potekal na vseh treh testnih površinah po naslednjem zaporedju. Začeli smo z Roombo v privzetem načinu delovanja, nato smo Roombo ovrednotili z našim algoritmom. Enkrat brez popravkov smeri in drugič s popravki smeri. Za vsak način smo naredili tri meritve. V nadaljevanju so predstavljeni rezultati meritev. V tabelah so za vsako testno površino in način delovanja predstavljeni časi, poti in pokritost. Ti podatki so izračunana povprečja meritev za vsak način delovanja.

Pri vrednotenju našega algoritma smo se zaradi višjega težišča sesalca, ki ga prinese nadgradnja s prenosnim računalnikom, odločili za bolj konzervativne hitrosti premikanja. Tako se sesalec premika in obrača počasneje kot v privzetem načinu delovanja.

Vrednotenje smo prekinili, ko je pokritost prostora dosegla 99% oziroma po določenem času, ko je bilo iz poteka razvidno, da se je pokritost ustavila pri določenem procentu.

4.3.1 Vrednotenje na majhni testni površini

Na najmanjši testni površini so vsi trije načini delovanja pokrili 99% površine. Odstopanja pri odometriji so pri načinu delovanja brez popravkov privedle do ponavljajočega se vzorca gibanja v zanki. Pri načinu s popravki smeri se je sesalec večkrat ujel v kot in zaradi tega porabil več časa za popravljanje smeri in posledično opravil najdaljšo pot. Privzeti način delovanja je porabil najmanj časa in opravil najkrajšo pot. Rezultati so zbrani v Tabeli 4.1.

Način delovanja	Čas v sekundah	Pot	Pokritost
privzeto delovanje	107	9180	99%
brez popravkov smeri	169	10348	99%
s popravki smeri	272	19857	99%

Tabela 4.1: Tabela meritev - velikost prostora 180cm x 115cm

4.3.2 Vrednotenje na srednji testni površini

Na srednji testni površini je način brez popravkov smeri pokril najmanjšo površino. Sesalec se je zaradi napak pri odometriji ujel v zanko, ki je zaradi večje testne površine pokrila manjši del. Pokritost se je ustavila na 67%. Način delovanja s popravki je pokril 99% testne površine in porabil največ časa, obenem pa opravil najkrajšo pot. Najhitrejši je bil privzeti način delovanja, ki pa zaradi pogostega vračanja na že pokriti del ni opravil najkrajše poti. Rezultati so zbrani v Tabeli 4.2.

Način delovanja	Čas v sekundah	Pot	Pokritost
privzeto delovanje	222	24377	99%
brez popravkov smeri	330	20638	67%
s popravki smeri	435	22794	99%

Tabela 4.2: Tabela meritev - velikost prostora 230cm x 175cm

4.3.3 Vrednotenje na veliki testni površini

Na največji testni površini se je najslabše izkazal način brez popravkov smeri. Zaradi napak odometrije se je sesalec hitro ujel v ponavljajoči se vzorec, zato se ga je po določenem času ustavilo, ker pokritost prostora ni presegla 27%. Privzeti način delovanja in način s popravki smeri sta se izkazala za dokaj enakovredna, kar se tiče pokritosti prostora in opravljene poti. Privzeti način se je izkazal za hitrejšega, a se je pokritost ustavila pri 98%. Način s popravki smeri je pokril 99% površine, potreboval pa je precej več časa. Rezultati so zbrani v Tabeli 4.3.

4.3.4 Prikaz in razlaga rezultatov

Pri vodenju sesalca z našim algoritmom so se v praksi pokazale težave z natančnostjo odometrije, ki se je odražala v odstopanju želenega kota pri obračanju. Pri želeni rotaciji za kot 90°je bila napaka +/- 5%. To se je najbolj

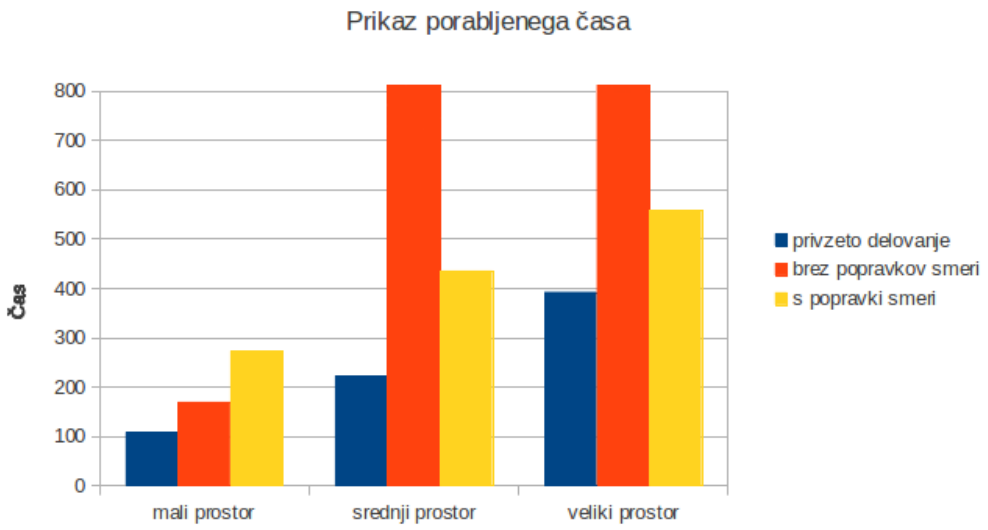
Način delovanja	Čas v sekundah	Pot	Pokritost
privzeto delovanje	392	22647	98%
brez popravkov smeri	173	5320	27%
s popravki smeri	558	22958	99%

Tabela 4.3: Tabela meritev - velikost prostora 260cm x 230cm

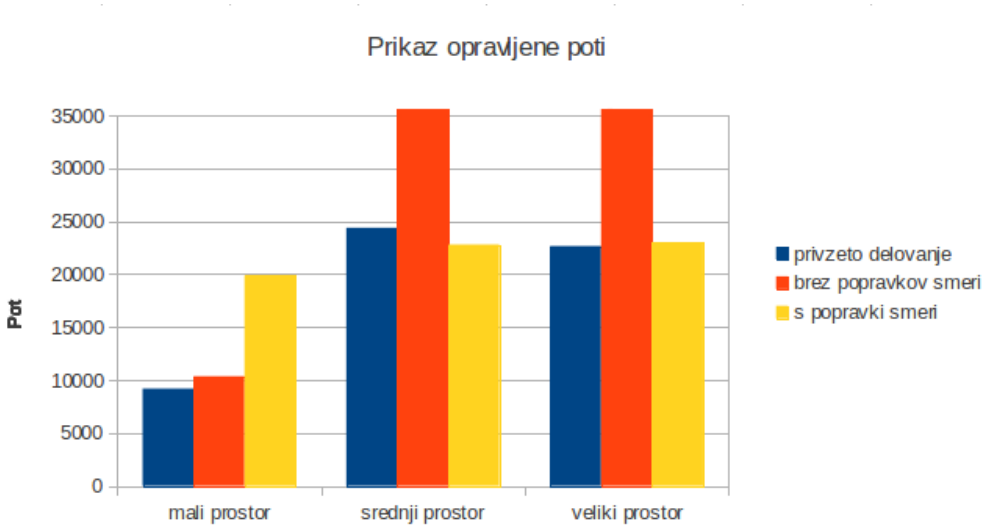
pokazalo pri načinu brez popravkov smeri, ker se je sesalec ujel v ponavljajoči se vzorec. Na najmanjši testni površini je to zadostovalo za pokritje celotnega prostora, na večjih testnih površinah pa samo za del. Način s popravki smeri se je temu problemu izognil tako, da je vedno pokril celotno površino. Pokazala pa so se odstopanja od načrtovane poti premikanja. Odstopanja so posledica nenatančnosti odometrije v povezavi z nenatančnostjo meritev infrardečih senzorjev. Meritve so v praksi lahko odstopale tudi do 15° od dejanskega kota. To je privedlo do elementa naključnosti, ki je pripomogel, da se sesalec ni ujel v zanko.

Graf porabljenega časa (na Sliki 4.8) prikazuje čas, ki ga je posamezni način delovanja porabil za pokritje celotne površine na vseh treh testnih prostorih. Način brez popravkov smeri je bil na srednjem in velikem testnem prostoru predčasno prekinjen, ker bi za pokritost celotne površine porabil precej več časa kot ostala dva načina. Način s popravki smeri je zaradi nastavljene manjše hitrosti gibanja porabil več časa kot privzeti način gibanja.

Graf (na Sliki 4.9) prikazuje opravljene poti. Za način brez popravkov smeri podatki na grafu ne odražajo prave slike. Na srednjem in velikem testnem prostoru se je sesalec ujel v zanko. Posledično je pot naraščala, pokritost prostora pa se je ustavila pri določeni vrednosti, zato je bilo delovanje predčasno prekinjeno.



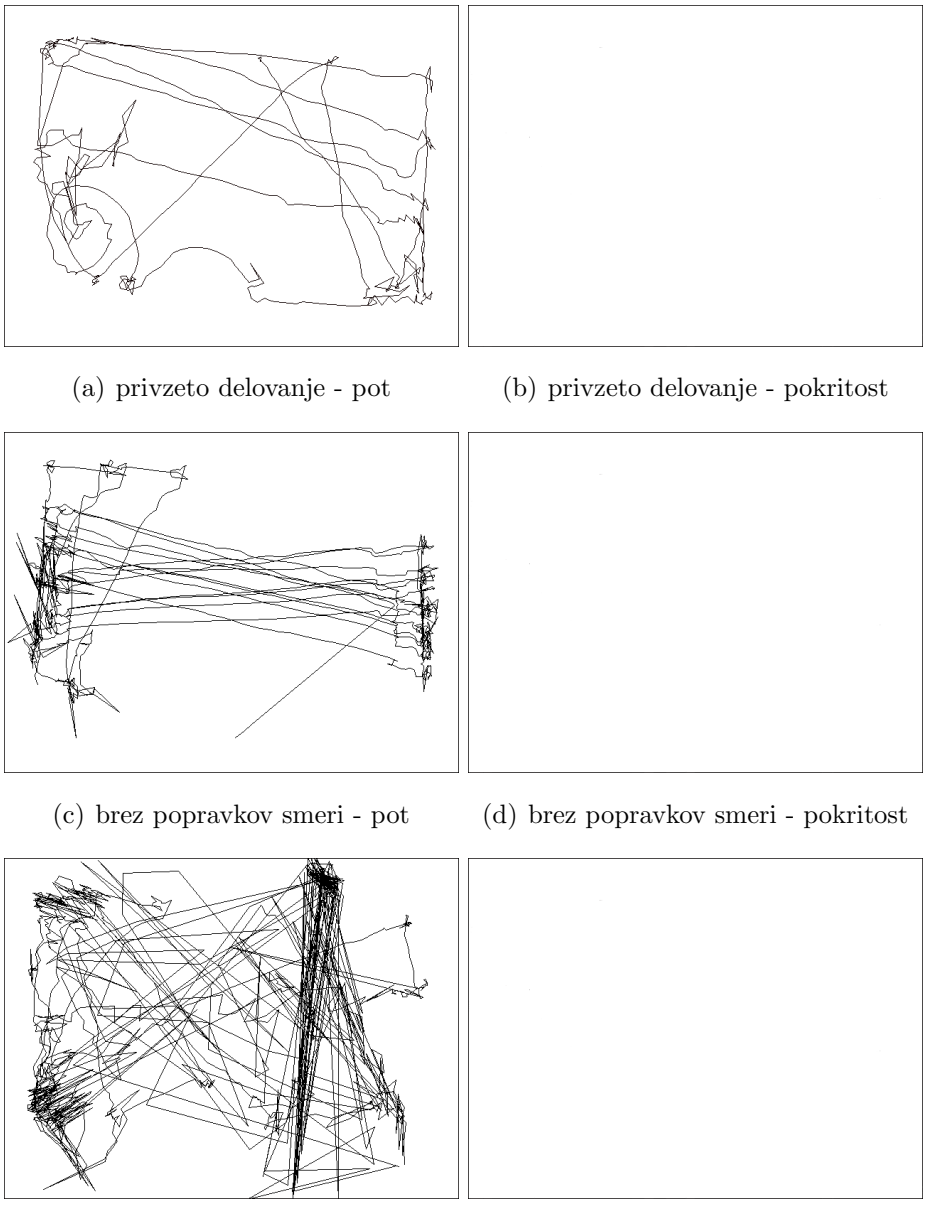
Slika 4.8: Prikaz porabljenega časa



Slika 4.9: Prikaz opravljene poti

4.3.5 Slike opravljenih poti in pokritosti

Slike spodaj predstavljajo izbrane poti gibanja obrisa sesalca in pokritost prostora. Ker se vzorci poti med seboj niso pretirano razlikovali, so prikazani samo najbolj reprezentativni vzorci. Pri slikah pokritosti prostora bela barva predstavlja prostor, ki ga je sesalec že obiskal. Slika 4.10 predstavlja poti in pokritost na najmanjšem testnem prostoru. Vsi trije načini delovanja so pokrili celotno površino, zato se slike pokritosti med seboj ne razlikujejo. Na Sliki 4.11 so predstavljene poti in pokritost na srednji testni površini. Izstopa Slika 4.11(d), ki prikazuje pokritost pri načinu brez popravkov smeri, ker sesalec ni uspel pokriti celotne površine. Pokritost največje testne površine je prikazana na Sliki 4.12. Slika 4.12(b) prikazuje manjšo površino, ki je privzeti način delovanja ni uspel pokriti, na Sliki 4.12(b) pa je večja površina, ki v načinu brez popravkov smeri ni bila pokrita.

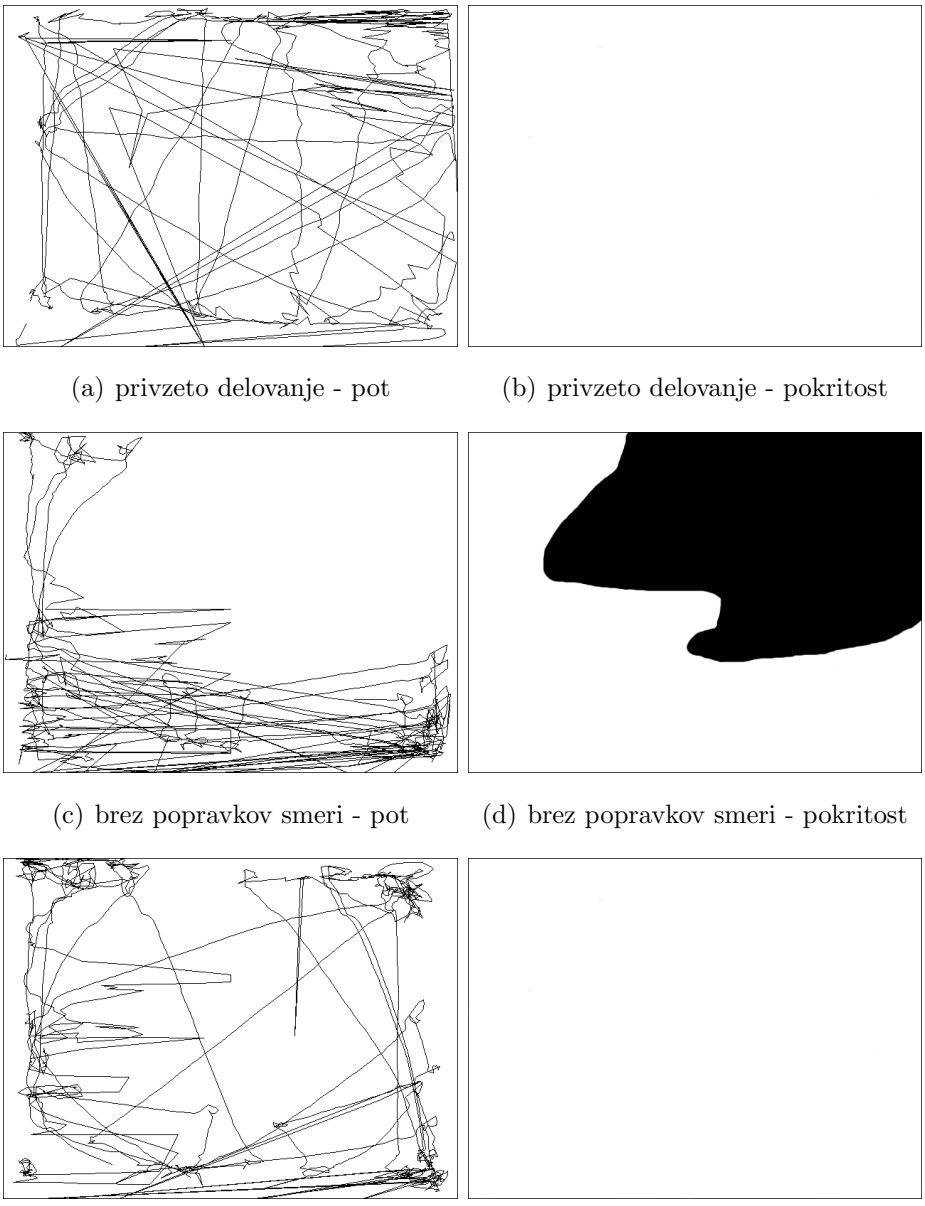


(a) privzeto delovanje - pot (b) privzeto delovanje - pokritost

(c) brez popravkov smeri - pot (d) brez popravkov smeri - pokritost

(e) s popravki smeri - pot (f) s popravki smeri - pokritost

Slika 4.10: Opravljene poti in pokritost - velikost prostora 180cm x 115cm

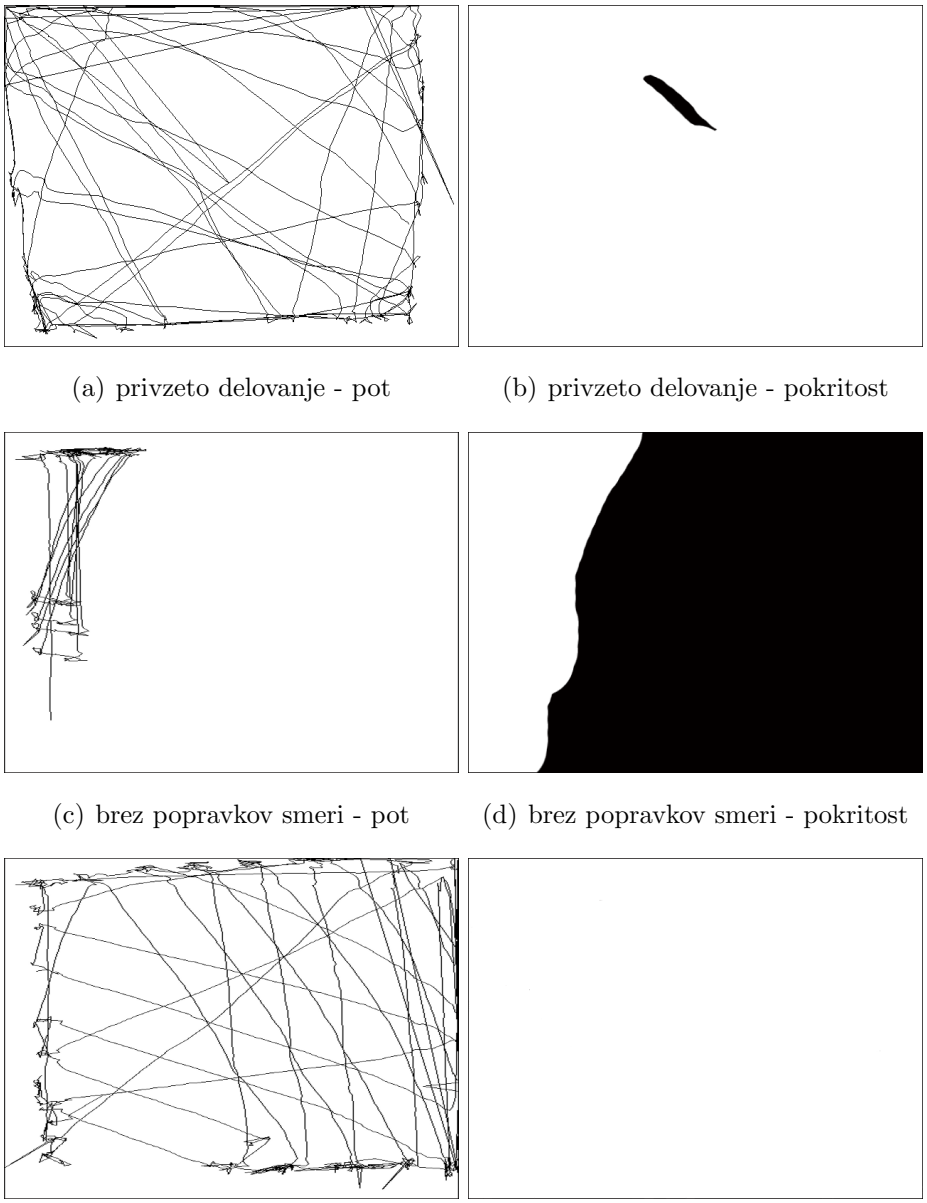


(a) privzeto delovanje - pot (b) privzeto delovanje - pokritost

(c) brez popravkov smeri - pot (d) brez popravkov smeri - pokritost

(e) s popravki smeri - pot (f) s popravki smeri - pokritost

Slika 4.11: Opravljene poti in pokritost - velikost prostora 230cm x 175cm



(a) privzeto delovanje - pot

(b) privzeto delovanje - pokritost

(c) brez popravkov smeri - pot

(d) brez popravkov smeri - pokritost

(e) s popravki smeri - pot

(f) s popravki smeri - pokritost

Slika 4.12: Opravljene poti in pokritost - velikost prostora 260cm x 230cm

Poglavje 5

Zaključek

V diplomski nalogi smo se lotili izboljšave osnovnega delovanja mobilne platforme iRobot Roomba. Osnovno delovanje izbira med nekaterimi predprogramiranimi načini obnašanja. Izbira je v veliki večini primerov naključna. Posledica tega je, da je premikanje po prostoru nesistematično. Roomba se lahko dlje časa zadržuje na delu prostora, pogosto se vrača na že obiskan prostor, nekaterih delov prostora pa niti ne obišče. Te pomanjkljivosti smo poskušali odpraviti z izdelavo algoritma, ki bi bolj sistematično in s tem bolj učinkovito preiskal prostor. Večina se tega problema loteva z uporabo dragih dodatnih senzorjev, na primer z laserskimi merilniki razdalje ali kamerami. Mi smo se tega lotili na drug način in se osredotočili samo na senzorje, ki so že vgrajeni v mobilno platformo.

Za sistematično preiskovanje prostora smo izbrali način premikanja poimenovan boustrofedon. Ta poteka tako, da se Roomba giblje po prostoru, dokler ne naleti na steno, se nato obrne in se po vzporedni poti giblje v nasprotni smeri nazaj. Ta vzorec se ponavlja, dokler ne preišče vsega prostora. Zamislili smo si dva načina delovanja aplikacije: prvi način delovanja uporablja senzorje samo za zaznavanje sten, drugi pa uporablja senzorje tudi za popravljanje smeri. Aplikacijo smo ovrednotili na treh različno velikih površinah. Eksperiment se je opravil s kamero, ki je bila pritrjena na ustrezni višini, da je pokrila celoten testni prostor. Aplikacija za vrednotenje je

preko kamere zasledovala gibanje Roombe in beležila čas, opravljeno pot in pokritost prostora.

Pri vrednotenju so se pokazala velika odstopanja v natančnosti senzorskih meritev. Prihajalo je do velikega razkoraka med zastavljenim ciljem gibanja in dejanskim gibanjem. To je pri načinu delovanja brez popravkov smeri privedlo do zanke, v katero se je ujela Roomba. Na najmanjšem testnem prostoru se je ta način kljub temu dobro obnesel, na večjih testnih površinah pa bil pokrit samo manjši del. Način delovanja s popravki smeri s tem ni imel težav in je preiskal celotno površino na vseh testnih površinah. Zaradi odstopanj v meritvah senzorjev je tudi v tem načinu delovanja prihajalo do odstopanj od začrtane poti. Na koncu se je rezultate primerjalo z privzetim načinom delovanja Roombe. Privzeti način se je izkazal kot najhitrejši, ni pa vedno preiskal celotne površine, niti ni opravil najkrajše poti. Način delovanja s popravki smeri je potreboval več časa kot privzeti način, vendar pa je preiskal celotno površino in v določenih primerih to opravil po najkrajši poti.

Za največji problem se je izkazala nenatančnost senzorjev, tako za odometrijo, kot za prepoznavanje ovir. Z bolj natančnimi senzorji bi se razviti algoritmi lahko izkazali za še bolj učinkovite, tako časovno kot po opravljeni poti. Vendar bi to prineslo tehnološke težave in višjo ceno.

Literatura

[1] G. Bradski, A. Kaehler, *Learning OpenCV*, O'Reilly Media, Inc., 2008

[2] Todd E. Kurt, *HackingRoomba*, Wiley Publishing, Inc., 2007

[3] Steven M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006

[4] iRobot Roomba –navodila za uporabo, 2009-2010

[5] Spletna stran paketa camera_calibration
Dostopno na: http://www.ros.org/wiki/camera_calibration (zadnji obisk 01.08.2012)

[6] Uradna spletna stran podjetja iRobot
Dostopno na: http://www.irobot.com/ (zadnji obisk 10.09.2012)

[7] Uradna spletna stran OpenCV
Dostopno na: http://opencv.willowgarage.com/ (zadnji obisk 31.08.2012)

[8] Spletna stran paketa roomba_500_series.
Dostopno na: http://www.ros.org/wiki/roomba_500_series (zadnji obisk 30.06.2012)

[9] Uradna spletna stran Ros.org.
Dostopno na: http://www.ros.org/ (zadnji obisk 15.09.2012)

[10] spletna stran Wikimedia Commons
Dostopno na: http://commons.wikimedia.org/ (zadnji obisk 03.09.2012)