

## Lokalizacija robota pomoću optičkog senzora

---

*Izvršeno je pozicioniranje robota Surveyor SRV-1 u dvodimenzionalnom prostoru sa podlogom koja se sastoji od crno-belih polja jednakih dimenzija, omogućena je rotacija robota za određen ugao kao i pomeranje za određeni iznos. Kamera koja se nalazi na robotu upotrebljena je kao senzor za prikupljanje informacija za određivanje položaja. Određivanje ugla rotacije vrši se detekcijom i analizom pravih linija sa podloge, dok se za određivanje rastojanja koje je robot prešao koriste optical flow i prebrojavanje polja sa podloge.*

---

### Uvod

Roboti danas imaju primenu u gotovo svim granama industrije. Poslednjih godina intenzivno se razvijaju spasilački roboti koji se upotrebljavaju pri pretrazi ruševina nakon elementarnih nepogoda, eksplozija, požara. Oni zamenjuju ljude u opasnim istraživanjima nepoznatog terena i predstavljaju veliku pomoć spasilačkim ekipama. Da bi se samostalno kretali kroz nepoznati prostor kao što su ruševine potrebno je da na osnovu podataka sa senzora koji se nalaze na njima dobiju informaciju o putu koji su prešli kako bi uspešno izvršili mapiranje tog prostora.

Tokom kretanja robota sa gusenicama moguće su dve vrste pomeranja: rotiranje i translacija napred i nazad. Da bi se kretanjem moglo upravljati neophodne su informacije o uglu rotacije i pređenom rastojanju prilikom translacije, koje je potrebno odrediti na osnovu podataka sa senzora robota. U ove svrhe koriste se različite vrste senzora, sonari, optički enkoderi, laserski daljinometri, kamere. Surveyor SRV-1 na sebi ima fiksiranu kameru koja posmatra okolinu

ispred robota. Pomoću slika dobijenih sa kamere određivane su nepoznate veličine – ugao prilikom rotacije, odnosno pomeraj prilikom translacije. Kako bi se izračunale nepoznate veličine, korišćena je podloga izgleda šahovske table, izdvojena na polja crne i bele boje poznate veličine, na kojoj je bilo potrebno izdvojiti linije između polja.

### Metod

Surveyor robot se povezuje sa računarom preko WiFi konekcije. Slike sa kamere robota se šalju računaru koji potom vrši njihovu obradu kako bi došao do informacija o pređenom uglu i putu, na osnovu kojih se vrši upravljanje. Obrada slike je vršena u programskom jeziku C++ sa Intel-ovom bibliotekom OpenCV. Pokušan je i rad u Matlabu, ali se nije pokazao kao dobar izbor za obradu slike u realnom vremenu.

Zbog nesavršenosti sočiva kamere dolazi do distorzije slike, njenog zakrivljenja na stranama poznatom pod nazivom „fish eye” (slika 1A). Da bi se sa slike efikasno izdvojile linije polja podloge potrebno je otkloniti ovu nepravilnost.

U cilju ispravljanja distorzije slike vrši se kalibracija kamere. Na osnovu određenog broja slika izračunavaju se koeficijenti distorzije pomoću kojih se ona na sledećim slikama otklanja (slika 1). Kalibracija je izvršena pomoću Camera Calibration Toolbox-a u Matlabu (web 1), na setu od petnaest slika. Koeficijenti dobijeni tom kalibracijom se potom koriste u funkcijama biblioteke OpenCV za korekciju distorzije svake slike dobijene sa kamere na sledeći način (Bradski i Kaehler 2008):

---

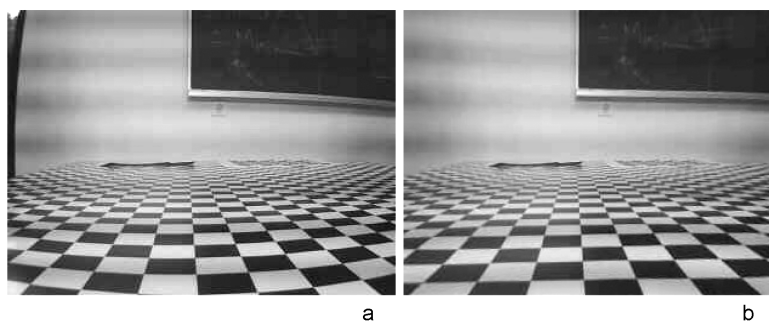
*Vladimir Ranković (1992), Beograd, Dr Ivana Ribara 146, učenik 4. razreda Matematičke gimnazije u Beogradu*

*Lazar Timotić (1992), Badovinci, Miloša Obilića 176, učenik 4. razreda Tehničke škole u Šapcu*

**MENTORI:**

*Filip Panjević, MDCS, Beograd*

*Miroslav Bogdanović, MDCS, Beograd*



Slika 1. a) slika sa robota bez ispravljene distorzije; b) slika sa robota sa ispravljenom distorzijom

Figure 1. a) An image taken by the robot with distortion; b) an image taken by the robot with corrected distortion

$$X = x \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$Y = y \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

pri čemu su  $X$  i  $Y$  nove, ispravljene, koordinate tačke,  $x$  i  $y$  koordinate tačke sa originalne slike na kojoj je uočena distorzija,  $k_1$ ,  $k_2$  i  $k_3$  koeficijenti distorzije, a  $r$  promenljiva polinoma.

Pored „fish eye” distorzije na slikama se uočava i perspektivna distorzija, pojava da sve linije koje se protežu od robota na slici skreću ka centru zbog različitih uglova pod kojim svetlost dolazi do optičkog senzora u kameri. Dakle, koordinate tačaka na slici ne odgovaraju koordinatama tačaka u prostoru zbog čega mora biti izvršena transformacija. Preslikavanje koordinata tačaka sa slike u realne koordinate u prostoru vrši se pomoću sledećih formula:

$$X = f \cdot \frac{x}{y}$$

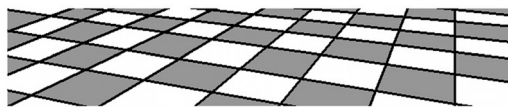
$$Y = f \cdot \frac{h}{y}$$

gde je  $X$  realna udaljenost tačke od robota po  $x$  osi, tj. levo i desno od njega,  $Y$  realna udaljenost tačke od robota po  $y$  osi, tj. ispred njega,  $(x, y)$  – pozicija tačke na slici, gledano na centar slike kao koordinatni početak,  $f$  žižna daljina sočiva kamere, a  $h$  udaljenost kamere od podloge (Hartley 2004).

Da bi se povećala brzina obrade slike u daljoj obradi posmatra se samo donji deo slike na kojoj se šahovnica najbolje vidi.

Određivanje ugla pod kojim se robot nalazi u odnosu na podlogu vrši se pomoću linija šahovske table. Koordinate početnih i krajnjih tačaka ovih linija dobijaju se primenom Hafove transformacije, koja svakoj tački na slici dodeljuje liniju u parametarskom prostoru. Tako sve tačke koje se nalaze na istoj liniji na slici u parametarskom prostoru imaju linije koje se seku u istoj tački (Bradski i Kaehler 2008). Detekcija linija na slici se vrši traženjem preseka većeg broja

parametarskih linija. Pre izdvajanja linija, slika se konvertuje iz RGB formata u grayscale format gde su boje reprezentovane određenom količinom crne i bele boje, a potom se vrši izdvajanje ivica na slici nekim od algoritama koje poseduje biblioteka OpenCV.



Slika 2. Linije izdvojene pomoću Hafove transformacije (označene debljim linijama)

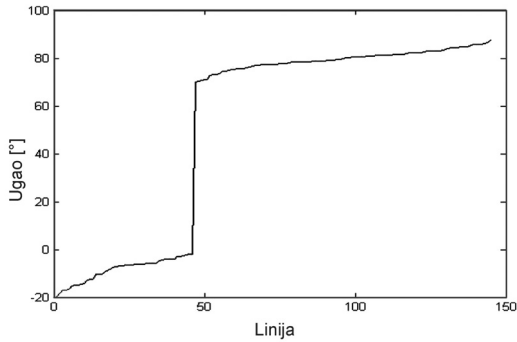
Figure 2. Lines extrapolated using Hough transform (marked as thick black lines)

Nakon izdvajanja linija (slika 2) sa slike one se preslikavaju u realan koordinatni sistem, posle čega se za sve linije računaju uglovi koje one zaklapaju sa  $x$  ili  $y$  osom. Uglovi se računaju prema formuli:

$$\theta = \arctg \frac{Y_2 - Y_1}{X_2 - X_1},$$

gde su  $X_1$  i  $Y_1$  koordinate početne, a  $X_2$  i  $Y_2$  koordinate krajnje tačke linije.

Da bi se odstranile pogrešno izdvojene linije vrši se filtriranje tako što se opseg od  $0^\circ$  do  $180^\circ$  deli na segmente od  $5^\circ$ . Ukoliko u nekom segmentu ne postoji više od četiri ugla, svi uglovi tog segmenta se brišu. Posle filtriranja ostaju samo dve grupe uglova (slika 3), pri čemu su svi uglovi u istoj grupi približno jednaki. Usrednjavanjem vrednosti u grupi dobija se ugao pod kojim se nalazi robot u odnosu na podlogu.



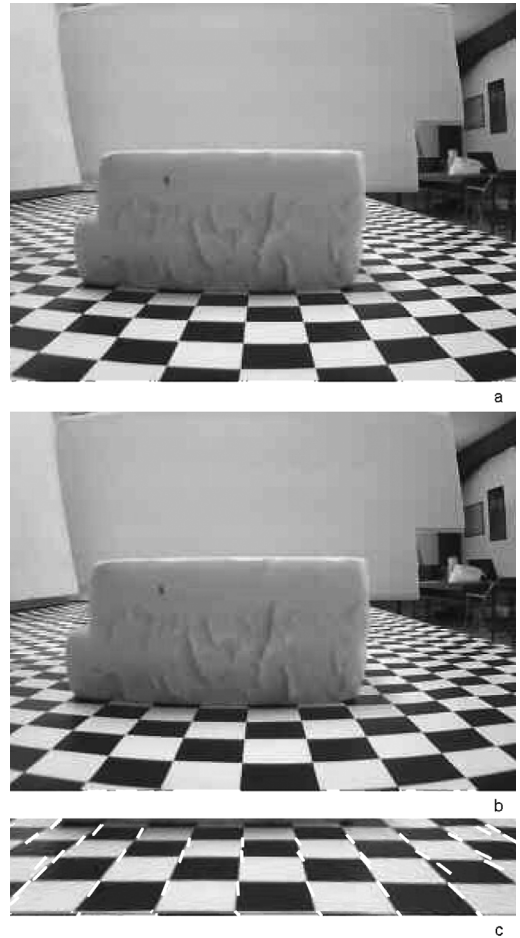
Slika 3. Preostale dve grupe uglova nakon filtriranja

Figure 3. Two remaining groups of the angles after filtering

U slučaju pravolinijskog kretanja potrebno je odrediti put koji robot pređe. Upoređena su dva metoda kojima je moguće odrediti pređeni put. Prilikom upotrebe optical flow metode računa se pomeraj između dve uzastopno dobijene slike sa kamere a ukupan put predstavlja zbir svih pomeraja (Davies 2005). Postoji više načina za implementaciju optical flow-a. Piramidnalna Lucas-Kanade metoda (LK) je prihvaćena kao standard (Szeliski 2008). Na prvoj slici se izdvajaju tačke od značaja (karakteristične tačke) čije se kretanje posmatra. Oko svake tačke LK metoda definiše prozor određene veličine. Na sledećoj slici značajne tačke se traže u okviru prethodno definisanih prozora, čime se postiže ušteda na vremenu obrade. Ovakav pristup ima nedostatak kada dođe do većih pomeraja i tačka na sledećoj slici izađe iz okvira prozora. Problem se rešava praćenjem tačaka u širim prozorima lošijeg kvaliteta (niži nivoi piramide), a potom u užim prozorima boljeg kvaliteta (viši nivoi piramide), te se tako detektuju i brzi pokreti koji prave veće pomeraje (Gonzalez i Woods 2008).

Posle primene LK algoritma poznate su koordinate značajnih tačaka sa prve kao i sledeće slike. Računa se rastojanje za svaki par značajnih tačaka (slika 4).

Ova rastojanja bi trebalo da budu jednaka i da predstavljaju pomeraj između dve slike, međutim usled grešaka pri praćenju značajnih tačaka neka izra-

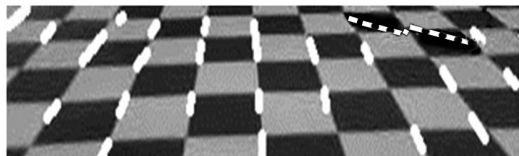


Slika 4. Dve uzastopne slike sa robota i pomeraj izmeren između njih (bele linije)

Figure 4. Two consecutive images from the robot, and movement detected between them (white lines)

čunata rastojanja su pogrešna i predstavljaju šum. Otklanjanje šuma postiže se upotrebom RANSAC algoritma (RANdom SAMple Consensus). Od svih izračunatih rastojanja formira se niz. Iz niza se nasumično odabere nekoliko vrednosti (npr. pet) čiji se proseki izračuna. Potom se prolazi kroz čitav niz i ispituje se svaki elemenat. Ukoliko se element razlikuje za manje od definisanog praga, njegova vrednost se prihvata i uključuje se u računanje proseka sa svim prethodnim vrednostima, u suprotnom se odbacuje. Posle prolaska kroz ceo niz dobija se proseki koji

predstavlja konačan pomeraj između dve slike. Kako bi se izbegla mogućnost da se pri biranju nasumičnih vrednosti izaberu sve pogrešne, postupak se ponavlja više puta i kao konačan rezultat uzima se srednja vrednost dobijenih pomeraja. Na slici 5 prikazano je odstranjivanje pogrešno spojenih tačaka pomoću RANSAC algoritma.



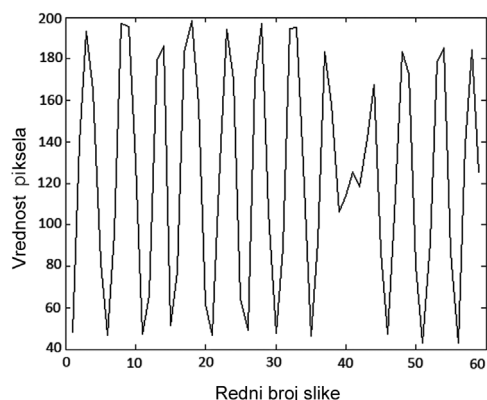
Slika 5. RANSAC algoritam – primer (isprekidana linija odstranjena iz niza)

Figure 5. And example of the action done by the RANSAC algorithm (movement represented as dashed line is removed from the sample)

Pored optical-flow-a testiran je i metod brojanja pređenih polja sa podloge. Kako se na podlozi naizmenično smenjuju polja crne i bele boje moguće je detektovati prelaze između boja i tako odrediti broj polja koje je robot prešao. Polja su kvadrati čija je stranica fiksna i iznosi 2.5 cm, pa se množenjem sa brojem pređenih polja dobija rastojanje pređeno u

centimetrima. Između vrednosti bele i crne boje na slici postoji razlika. Kako bi se efikasno detektovao prelaz iz jedne boje u drugu potrebno je što preciznije odrediti razliku između boja. Određivanje razlike je vršeno tako što je određen prag za crnu odnosno belu boju. Prag je određen tako što je boja usrednjena na uzorku od 180 prozora  $10 \times 10$  piksela preuzetih sa 60 različitih slika. Na slici 6 prikazana su dva uzastopna merenja, na x osi je broj frejma, a na y osi je vrednost jednog piksela na frejmu. Vidi-mo kako vrednost piksela raste i opada naizmenično. Male vrednosti označavaju crna polja, dok velike vrednosti označavaju bela polja. Manje razlike nastaju kad polje obuhvati sam prelaz između polja. Da bi izbegli situaciju da neki prelaz ne bude primećen, posmatrali smo tri susedna polja.

Za oba metoda pomoću kojih se određuje pređeni put izvršeno je po deset merenja za nekoliko različitih rastojanja. Rezultati dobijeni brojanjem polja prikazani su u tabeli 1. Prosečna vrednost merenja upoređena je sa zadatom i izražena u tabeli. Ukoliko se svi prelazi između crne i bele boje detektuju prosečno odstupanje ne može biti veće od dimenzije polja (2.5 cm), jer greška može nastati samo ukoliko se početak vidnog polja kamere nalazi negde u polju, a ne na granici između polja. Do veće greške može doći ukoliko nisu detektovani svi prelazi što je najčešće izazvano uticajem osvetljenja na razliku vrednosti između crne i bele boje. Nedostatak ovakvog pristupa jeste nemogućnost određivanja rastojanja ukoliko robot nije paralelno postavljen sa nekom od



Slika 6. Određivanje razlike između belih i crnih polja

Figure 6. Calculating the differences between black and white fields

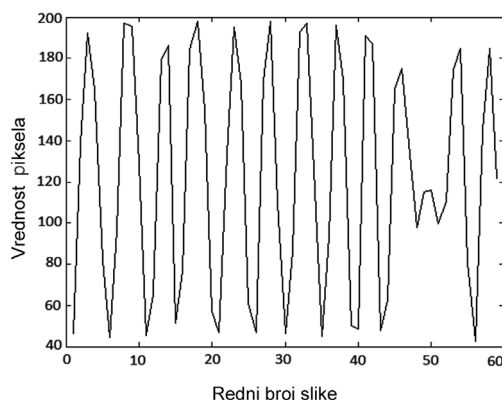


Tabela 1. Merenja prilikom određivanja rastojanja upotrebom metode brojanja pređenih polja

Tačno rastojanje	Merenje										SV	SD	Ods.
	1	2	3	4	5	6	7	8	9	10			
2.5	2.8	2.5	2.7	2.5	3.4	2.9	2.9	2.6	2.9	3.2	2.84	0.29	0.34
5.0	5	5.2	5.4	4.9	5.5	5.5	5	5.3	6.1	5.1	5.30	0.35	0.30
10.0	7.5	11.8	11.7	11	10	10.7	12.9	11.2	11.2	10.6	10.86	1.41	0.86
12.5	13.4	12.6	11.2	13.7	14.4	12.1	12.7	13.1	13.2	11	12.74	1.61	0.24
20.0	19.4	20.2	19.4	20.1	21.1	18.9	21.6	23.4	22.6	18.6	20.53	2.00	0.53
25.0	26.8	23.7	27.1	24.3	27.5	24.2	21	23.2	24.5	23.8	24.61	2.19	0.39
37.5	34.5	38.8	39.2	38.1	35.2	39.1	39.3	37	34	39.9	37.51	2.30	0.01
50.0	46.8	50.3	47.2	48.9	46.2	51.1	49.4	47.6	50.2	53.7	49.14	1.95	0.86
62.5	64.1	61.6	60.5	63.4	65	65	61	59.5	63.5	61.4	62.50	2.84	0.00
75.0	73.8	68.6	71.3	74.1	76.5	75.2	73.7	77.3	77.8	76.3	74.46		0.54

SV – srednja vrednost, SD – standardna devijacija, Ods. – odstupanje srednje od tačne vrednosti.  
Sve vrednosti su izražene u centimetrima

Tabela 2. Merenja prilikom određivanja rastojanja upotrebom metode optical-flow

		Merenje										SV	SD	Ods.
		1	2	3	4	5	6	7	8	9	10			
ZR	10.0													
PR		10.3	10.8	10.1	12.3	9.7	9.75	11	10.6	13.7	10.7	10.89	1.17	0.89
PIR		9.79	9.6	9.9	10.7	11.7	9.97	9.62	11	10.5	10.1	10.28	0.65	0.28
ZR	12.5													
PR		13.9	14.4	14.2	15.2	13.2	12.2	16.7	15.6	14.7	14.3	14.44	1.25	1.94
PIR		12.2	11.6	11.8	11.7	12	11.8	12.6	12.8	12.6	12.9	12.21	0.49	0.29
ZR	20.0													
PR		18.9	21.6	20.3	20.1	21.5	18.2	19.5	22.3	22.4	20.3	20.51	1.42	0.51
PIR		20.3	19.7	20.2	20.4	20	20.7	19.7	19.6	20.5	19.6	20.07	0.41	0.07
ZR	25.0													
PR		24.8	25.2	23.8	24.9	25.8	24.9	22.8	23.3	23.2	21.2	23.99	1.39	1.01
PIR		25.5	24.7	24.6	25.3	26.2	25.4	25.6	25.2	24.5	24.8	25.18	0.53	0.18
ZR	37.5													
PR		33.4	34.8	33.9	33.6	33.6	32.9	30.9	35	34.6	32.6	33.53	1.21	3.97
PIR		37.6	38.1	37.5	37.9	36.6	37.8	37.6	36.9	36.6	37	37.34	0.54	0.46
ZR	50.0													
PR		42.6	39.9	42	43.5	42.8	43.5	42.1	41.9	43.8	42.6	42.47	1.12	7.53
PIR		49.9	51.1	50.3	50.5	50.2	49.8	49.8	49.9	50	50.3	50.17	0.40	0.17

ZR – zadato rastojanje, PR – pređeno rastojanje, PIR – programski izračunato rastojanje, SV – srednja vrednost, SD – standardna devijacija, Ods. – odstupanje srednje od zadate vrednosti.  
Sve vrednosti su izražene u centimetrima

linija na podlozi. U takvom slučaju se za jedno polje može detektovati više prelaza, pa je određeno rastojanje neupotrebljivo.

Metoda koja se oslanja na optical flow može određivati rastojanje nezavisno od položaja robota u odnosu na linije na podlozi, rezultati dobijeni ovom metodom prikazani su u tabeli 2. Predeno rastojanje poređeno je sa rastojanjem koje je robot izmerio i odstupanje je prikazano u tabeli. Povećanjem rastojanja koje robot prelazi povećava se greška koja nastaje usled inercije jer je posle izvršenja poslednje komande za kretanje robotu potrebno neko vreme da se zaustavi. Ipak, najveću grešku pravi optical-flow kada pogrešno poveže parove većeg broja značajnih tačaka na susednim slikama pri čemu ni RANSAC algoritam nije u mogućnosti da otkloni grešku. Greške napravljene pri svakom pojedinačnom merenju se akumuliraju, povećavajući na taj način grešku.

Metod brojanja pređenih polja je brži i precizniji, ali neupotrebljiv u slučaju gde kretanje nije pod makar približno pravim uglom, otklanjanjem tog nedostatka ta metoda bi bila optimalna. Greške do kojih dolazi prilikom određivanja pomeraja mogu biti umanjene kretanjem robota u prostoru gde je osvetljenje nepromenljivo kao i finijim podešavanjem parametara koje algoritmi koriste.

## Zaključak

Realizovan je sistem koji se sastoji od robota i računara koji vrši kontrolu robota, određivanje pomeraja pri pravolinijskom kretanju i orijentacije u prostoru. Ustanovljene su prednosti i mane različitih metoda za izvršenje algoritma kao i različitih programskih jezika. Najveća mana sistema je potreba za specifičnom podlogom za određivanje orijentacije robota kao i greške do kojih dolazi pri merenju pređenog puta.

## Literatura

- Bradski G, Kaehler A. 2008. *Learning OpenCV*. Sebastopol: O'Reilly Media
- Davies E. R. 2005. *Machine Vision – Theory, Algorithms, Practicalities*. Elsevier
- Gonzalez R. C., Woods R. E. 2008. *Digital Image Processing*. Prentice Hall
- Hartley R. 2004. *Multiple View Geometry in Computer Vision*. Cambridge University Press
- Szeliski R. 2008. *Computer Vision: Algorithms and Applications*. Dostupno na: [http://research.microsoft.com/en-us/um/people/szeliski/book/drafts/szeliski\\_20080330am\\_draft.pdf](http://research.microsoft.com/en-us/um/people/szeliski/book/drafts/szeliski_20080330am_draft.pdf)
- web 1. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/htmls/example.html](http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html)

---

Vladimir Ranković and Lazar Timotić

## Robot Localization Using Video Sensor

Positioning of the robot Surveyor SRV-1 in two-dimensional space with checkerboard beneath him was performed. Moving the robot for a certain angle or length was enabled. The camera on the robot was used for gathering information, and the information about the robot's position is derived from it. For calculating the angle, the lines on the checkerboard are used and the distance that the robot has moved is calculated by using the Optical-flow algorithm. Using that information and the starting point we can calculate the current location. 