Bojana Nenezić

Sistem za automatsko parkiranje Robo-PICA robota

U ovom radu je opisan sistem za automatsko parkiranje Robo-PICA robota na unapred zadato parking mesto. Princip rada se zasniva na obradi podataka dobijenih sa kamere koja snima robota iz ptičije perspektive. Sistem je implementiran u programskom jeziku Matlab. Rad predstavlja jednostavan sistem za automatsko upravljanje.

Uvod

Sistem za automatsko parkiranje Robo-PICA robota (web1) predstavlja jednostavan sistem za automatsko upravljanje. Princip rada sistema zasniva se na pronalaženju koordinata centra mase robota i određivanju njegovog položaja u odnosu na parking mesto. Kao senzor za određivanje trenutnog položaja robota korišćena je kamera. Podaci dobijeni sa kamere šalju se računaru, koji ih obrađuje i šalje komande za upravljanje motorima. Pik robot u sebi ima ugrađena dva DC motora, kojima se upravlja preko mikrokontrolera. Motori pokreću gusenice i na taj način robot menja svoj položaj.

Cilj rada je dovođenje robota sa bilo kog mesta na površini koju snima kamera na obeleženo parking mesto, odnosno dovođenje robota u pravilan položaj. U toku samog procesa parkiranja, robot ne bi trebalo da pregazi ni jednu liniju, pošto linije predstavljaju zamišljene zidove garaže.

Usmeravanje robota predstavlja zadavanje putanje kojom će robot stići do željenog položaja. Neki primeri putanja kojima robot može da se dovede u pravilan položaj su:

a) kretanje najkraćim putem uz izbegavanje prelaska preko ivica parking mesta
b) dovođenje robota do određene tačke na podlozi sa koje se pravolinijski kreće do parking mesta

U ovom radu korišćen je primer putanje pod b). Ovaj primer izabran je zbog jednostavnije realizacije. Prvi primer putanje je zahtevniji i efikasniji, pa se njegovom realizacijom može unaprediti sistem.

Aparatura i metod

Aparatura (slika 1) sastoji se od:

– veb kamere
– podloge sa obeleženim parking mestom
– Robo-PICA robota
– računara

Radi lakšeg pronalaženja i određivanja položaja, na Robo-PICA robota pričvršćen je dodatni deo, gde je crnim markerom označen prednji kraj robota, a zelenim markerom zadnji kraj robota. Korišćeno je nekoliko različitih podloga, kako bi se utvrdilo na kojoj će kretanje robota biti najuspešnije (što podrazumeva najmanje proklizavanje). Iznad podloge postavljena je web kamera na nepokretnom stalku koja snima podlogu iz ptičije perspektive. Kamera je podešena da šalje YUV 640×480 sliku.

Robo-PICA robot u sebi ima ugrađenu RBX-887 V2.0 (web2) ploču za mikrokontroler PIC 16F887 (web3). Preko mikrokontrolera se vrši upravljanje DC motorima koji pokreću gusenice.

Za upravljanje robotom koriste se već gotove komande za pokretanje motora, čiji je kod napisan u programskom jeziku MikroC, a komunikacija između robota i računara ostvarena je preko serijskog porta.

Blok šema sistema prikazana je na slici 2.

Postupak rada sastoji se iz dve celine:

– obrade slike
– implementacije algoritma za kretanje robota

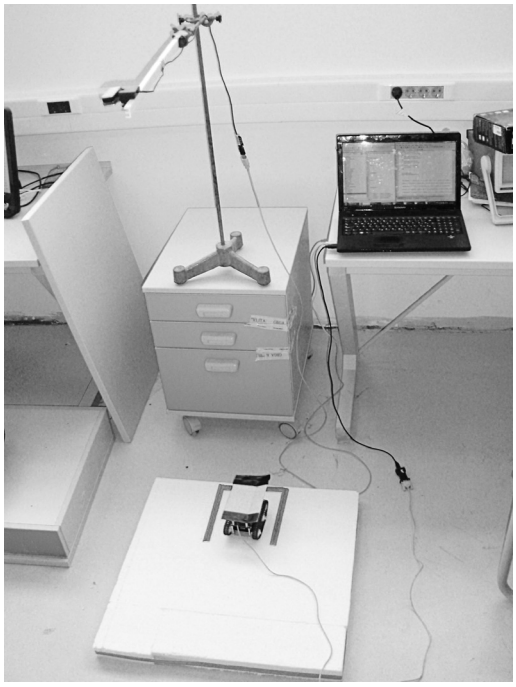
Bojana Nenezić (1993), Beograd, Dr. Ivana Ribara 68/3, učenica 4. razreda III beogradske gimnazije

MENTORI:

Gavrilo Andrić, student Elektrotehničkog fakulteta Univerziteta u Beogradu

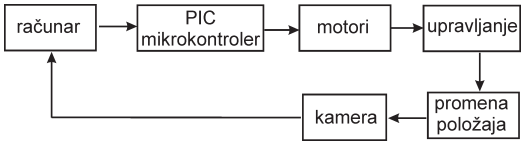
Vladimir Ranković, student Elektrotehničkog fakulteta Univerziteta u Beogradu

Darko Todorović, asistent na Elektronskom fakultetu Univerziteta u Nišu



Slika 1. Aparatura

Figure 1. Apparatus



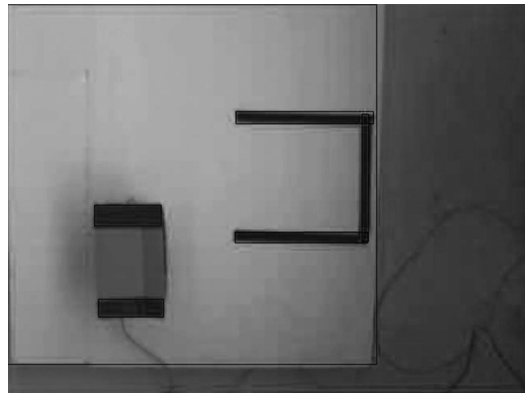
Slika 2. Blok šema sistema

Figure 2. Principle diagram

Obrada slike vršena je u programskom jeziku Matlab. Obrada slike izvedena je u nekoliko faza i zahteva mnogo vremena. Korišćene su neke ugrađene funkcije (kako bi se minimizovalo vreme obrade), kao što su pronalaženje najvećeg objekta na slici i pronalaženje centra mase objekta. U obradi je korišćena slika u YUV formatu (web4). Slike nisu konvetovane u drugi format, kako se ne bi povećalo vreme potrebno za njihovu obradu. YUV format sadrži tri komponente, od kojih prva određuje osvetljenost piksela, a druge dve nose informacije o boji. Sama obrada slike može se podeliti u sledeće faze:

- uzimanje uzoraka boje podloge, prednjeg i zadnjeg markera na robotu i parking mesta
- izdvajanje podloge i binarizacija slike
- pronalaženje najvećeg objekta svake boje
- pronalaženje centara mase izdvojenih objekata
- određivanje koeficijenata pravca robota i parking mesta

Da bi se na slici dobijenoj sa kamere precizno mogli izdvojiti objekti različitih boja, neophodno je odrediti opseg vrednosti piksela za svaki objekat. Određivanje opsega piksela vrši se prilikom inicijalizacije tako što se svaki objekat obeleži (slika 3).



Slika 3. Inicijalizacija

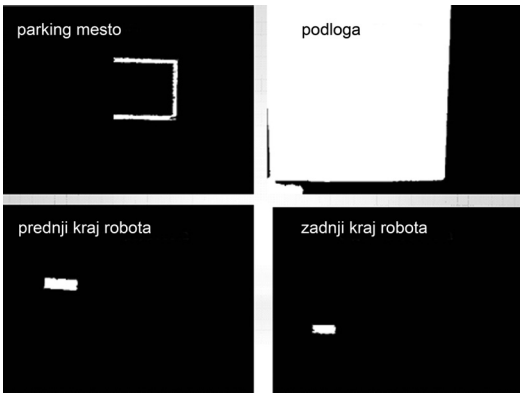
Figure 3. Initialization

Markeri na objektima se ručno označe u početnim položajima za koje se određuje standardna devijacija za vrednosti piksela. Standardna devijacija se koristi kako bi se što tačnije odredili opsezi, sa najmanjim udelom greške. Na taj način je izbegnuto empirijsko unošenje vrednosti, koje bi se često morale menjati usled promene intenziteta svetlosti u okruženju. Omogućeno je dovoljno precizno izdvajanje objekata, koje je neophodno za rad sistema, bez obzira na uticaj promene svetlosti u okolini objekta.

Drugu fazu obrade slike predstavlja izdvajanje podloge i binarizacija. Podloga za parkiranje manja je od površine koju snima kamera. Osim podloge, na slici se uočavaju i drugi objekti. Zbog toga se na slici pojavljuje šum. Šum može da poremeti određivanje vrednosti koordinata centara mase objekata. Ovaj

šum se javlja ukoliko boje objekata koji su van podloge ulaze u opseg posmatranih objekata na podlozi. Da bi se to izbeglo, osmišljen je algoritam, pomoću kojeg se izdvajaju pikseli koji čine podlogu. Ostali pikseli, koji nisu obuhvaćeni ovim opsegom dobijaju vrednost 0 i postaju crni. Na ovaj način se izbegavaju nepotrebni objekti, koji se nalaze na površini obuhvaćenoj kamerom. Ovaj algoritam se primenjuje kako bi na slici svi pikseli imali jednu od dve moguće vrednosti, čime se omogućuje izdvajanje objekata. Ovaj postupak naziva se binarizacija slike. Izdvajaju se sledeći objekti: zeleni marker (prednji kraj robota), crni marker (zadnji kraj robota), parking mesto i podloga za parkiranje.

Pošto se uklone nepotrebni delovi slike neophodno je izdvojiti objekte i pronaći najveći objekat za sve boje na slici. Posmatrani objekti su jednobojni, ali zbog različitog upadnog ugla svetlosti i senki predmeta iz okoline na njima može doći do promene intenziteta boje na nekom delu. Taj deo posle binarizacije može da se manifestuje kao šupljina u objektu. Šupljine mogu da uzrokuju promenu u vrednosti koordinata centara mase, što dovodi do greške i smanjuje preciznost sistema. Radi preciznije obrade praznine u objektima se popunjavaju. Ukoliko posle binarizacije slike u belim objektima ostanu skupine crnih piksela u vidu šupljina, oni menjaju svoju vrednost. Pikseli postaju beli, ispunjavajući šupljine da bi objekti bili prikazani kao celina. Na obrađenim slikama nalaze se izdvojeni beli objekti na crnoj pozadini (slika 4). Na ovim slikama moguće je primeniti funkcije za pronalaženje centara mase belih objekata, koje su ugrađene u Matlabu.



Slika 4. Izdvojeni objekti na obrađenim slikama

Figure 4. Featured objects: top left – parking spot; top right – surface; bottom left – robot front side; bottom right – robot rear side.

Koeficijent pravca putanje po kojoj se robot kreće određen je na osnovu koordinata centra mase njegovog prednjeg i zadnjeg dela:

$$k_{\text{robota}} = \frac{y_1 - y_2}{x_1 - x_2}$$

gde su x_1 i y_1 koordinate prednjeg, a x_2 i y_2 koordinate zadnjeg dela robota

Koeficijent pravca određuje osa parking mesta određen je na osnovu koordinata centra parking mesta i njegovog centra mase:

$$k_{\text{parking mesta}} = \frac{y_1 - y_2}{x_1 - x_2}$$

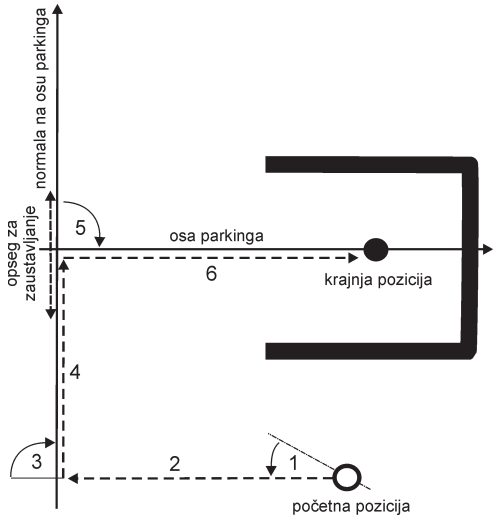
gde su x_1 i y_1 koordinate centra parking mesta, a x_2 i y_2 koordinate centra mase parking mesta

Centar parking mesta određen je upotrebom funkcije ugrađene u Matlab-u, kojom se opiše pravougaonik oko parking mesta i određuju vrednosti koordinata temena tog pravougaonika. Koordinate centra parking mesta određuju se kao algebarska sredina vrednosti koordinata temena tog pravougaonika.

Na osnovu koeficijenta pravaca putanje po kojoj se robot kreće i ose parkinga određeni su uglovi koje oni zaklapaju sa x-osom. Kao smernice za kretanje robota koriste se osa parkinga i pomoćna osa (slika 5). Pomoćna osa normalna je na osu parkinga. Osa parkinga predstavlja referencu u odnosu na koju se određuje u kom se položaju nalazi robot, na koju stranu i za koliki ugao treba da vrši rotaciju. Pomoćna osa služi za određivanje mesta do kog robot treba da vrši translaciju.

Algoritam za kretanje robota realizovan je tako da se robot iz bilo kog početnog položaja dovede do određene pozicije na podlozi, od koje nastavlja pravolinijsko kretanje do centra parking mesta. Algoritam je sastavljen iz 6 delova (slika 5): rotacija 1, translacija 1, rotacija 2, translacija 2, rotacija 3 i translacija 3.

Rotacija robota predstavlja njegovo okretanje u levu ili desnu stranu oko svoje ose. Rotacija se izvršava sve dok koeficijent pravca kretanja robota ne uđe u određeni opseg. Rotacijom se robot okrene za prav ugao. Translacija robota podrazumeva njegovo jednosmerno kretanje napred dok koordinate njegovog centra mase ne zadovolje zadati opseg. Pri-

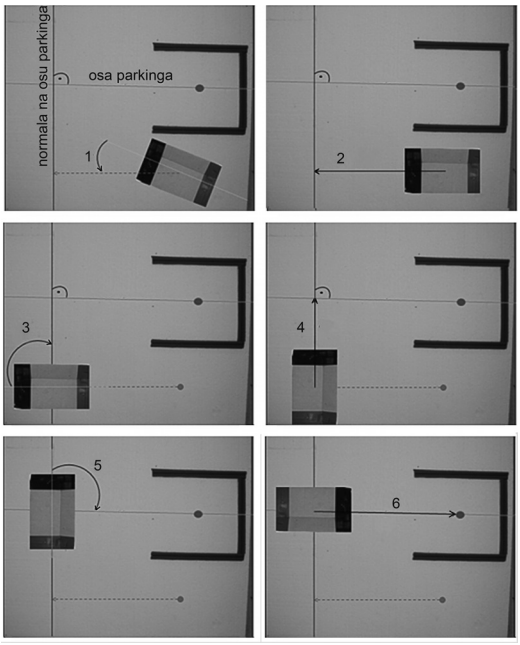


Slika 5. Osnovni elementi algoritma kretanja: 1 – rotacija 1, 2 – translacija 1, 3 – rotacija 2, 4 – translacija 2, 5 – rotacija 3, 6 – translacija 3.

Figure 5. Basic elements of the motion algorithm: 1 – rotation 1, 2 – translation 1, 3 – rotation 2, 4 – translation 2, 5 – rotation 3, 6 – translation 3.

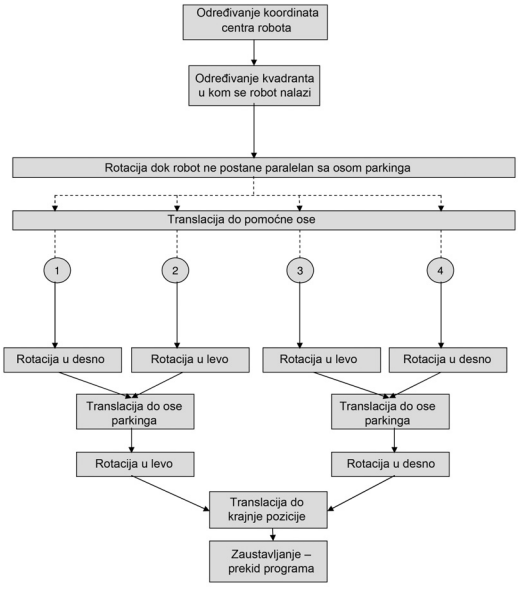
likom izvršavanja ovih komandi javlja se određena greška tako da se robot ne zarotira za zadati ugao i ne kreće u potpunosti pravolinijski. Do greške dolazi zbog merenja koja nisu idealna i mehanike samog robota. Na kretanje robota utiče spora obrada slike, usled čega se robot kreće intermitentno. Zbog toga su uvedeni opsezi za položaj koje robot mora da zadovolji. Ukoliko robot prilikom rotacije u jednom smeru prekorači zadati opseg za rotaciju, počinje da se izvršava korekcija pravca i menja smer rotacije. Vrednost opsega za rotaciju stavljena je na 30°, tj. ±15° u odnosu na osu parkinga. Ista vrednost važi i za opseg pomoćne ose. Prosečna brzina robota pri translaciji je približno 1 cm/s, a brzina rotacije iznosi 9°/s. Brzina robota zavisi od vrste podloge, jer trenje utiče na pokretljivost robota.

Kretanje napred odvija se dok koordinate centra mase robota ne uđu u odrgovarajući opseg za rotaciju u zavisnosti od njegovog položaja na slici. Početni položaj robota može da bude u jednom od 4 kvadranta. Na osnovu orijentacije robota rotacija se vrši u levu ili desnu stranu, tako da se izabere najkraća putanja. Rotacija i translacija se izvršavaju sve dok koeficijent pravca robota ne dostigne zadate opsege.



Slika 6. Primer situacije (oznake iste kao na slici 5)

Figure 6. Parking example (same notation as Figure 5)



Slika 7. Grafik algoritma kretanja

Figure 7. Flow chart of the motion algorithm

Tabela 1. Koordinate za vrednosti koordinata centra mase objekata

	Koordinate centra mase prednjeg dela robota				Koordinate centra mase zadnjeg dela robota				Koordinate centra mase parking mesta			
	x		*y*		*x*		*y*		*x*		*y*	
	*	**	*	**	*	**	*	**	*	**	*	**
1	203	202	61	62	275	275	148	148	563	535	231	227
2	100	100	179	177	189	189	247	248	563	535	231	228
3	85	86	298	308	81	63	195	196	563	535	231	228
4	297	391	273	227	258	509	170	225	563	536	231	227
5	344	341	344	344	402	399	245	244	563	535	231	229
6	248	244	196	195	248	243	312	313	563	535	231	227
7	331	333	172	172	239	240	106	100	563	536	231	227
8	304	307	160	115	229	419	246	150	563	536	231	228
9	308	304	119	111	211	221	149	170	563	536	231	227

* – koordinate izračunate u programu; ** – stvarne koordinate

Ukoliko robot izađe iz oblasti koju pokriva kamera, podaci o njegovom položaju se gube i izvršavanje programa se prekida.

Algoritam je realizovan tako da robot naizmenično vrši komande rotacije i komande translacije. Ukupno se izvrše po tri komande rotacije i translacije. Uporedo sa translacijom vrši se i korekcija pravca. Robot započinje kretanje kada se koeficijent pravca njegove putanje i koordinate centra ne nalaze u potrebnim opsezima. Po završetku poslednje translacije robot se zaustavlja i izvršavanje programa se prekida. Primer jedne situacije prikazan je na slici 6, a algoritam je prikazan na slici 7.

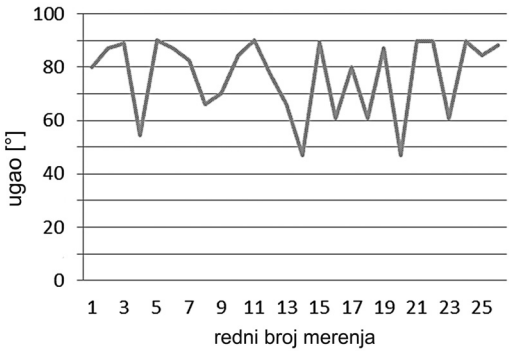
Rezultati i diskusija

Realizovan je algoritam za kretanje robota po zadatoj putanji. Obrada slike je vršena u Matlabu, pa srednja brzina obrade iznosi 0.246 sekundi (odnosno 4.065 slike po sekundi). Vremenski najzahtevnije funkcije su funkcija za binarizaciju slike i pronalaženje centara mase objekata. Zbog relativno spore obrade slike do robota stiže upravljanje u dužim vremenskim razmacima, pa se on kreće intermitentno.

Prilikom kretanja robot uspešno izvršava komande rotacije, pri čemu napravi odstupanje od zadatog ugla. Srednja vrednost ugla za koji se robot zarotira pri komandi za rotaciju za 90° iznosi 76.3° stepena. Standardno odstupanje od srednje vrednosti iznosi 14.4°, odnosno 6.56%. Greška se javlja jer je zadat opesg tolerancije za komandu rotacije, koji iznosi ±15 stepeni. Ova širina opsega bila je neophodna kako bi robot bio u mogućnosti da prekine rotaciju u potrebnom trenutku. Zbog spore obrade slike, informacije o položaju robota program dobija sa zakašnjenjem. Ukoliko bi opseg bio manji, do robota ne bi stizale komande za prestanak rotacije, pa izvršavanje algoritma kretanja ne bi bilo moguće (slika 8). Na preciznost okretanja takođe utiče i mehanika, jer robot ne može da se zarotira u mestu. Pri rotaciji kotur na prednjem delu gusenice robota napravi pomeraj od 7.2 cm po *x* i 5.42 cm po *y* osi.

Centar mase određen je uz pomoć funkcije ugrađene u Matlabu. Program izračunava vrednosti koordinata centra mase sa odgovarajućom preciznošću, tako da se robot parkira na zadato mesto. Odstupanje od stvarnih vrednosti koordinata centra mase izazvano je promenljivim osvetljenjem koje utiče na oblik i veličinu površine izdvojenih objekata. Na preciznost sistema utiče i položaj kamere, odnosno ugao pod kojim se snima podloga. Usled iskošenosti kamere, odnos dimenzija objekta na slici razlikuje se od stvarnog odnosa, što dovodi do greške. Upoređivane su vrednosti koje program izračunava za koordinate centra mase sa stvarnim koordinatama centara mase. Rezultati su prikazani u tabeli 1.

Osnovni algoritam realizovan je u programskom jeziku Matlab, zbog jednostavnosti sintakse i velikog



Slika 8. Vrednosti za ugao rotacije

Slika 8. Rotation angle

broja ugrađenih funkcija koje olakšavaju izradu. Za komunikaciju između robota i računara implementiran je kod u programskom jeziku MikroC. Ovaj sistem bi se mogao unaprediti realizacijom u drugom programskom jeziku kao što je C, gde bi se obrada slike vršila brže. Tada bi se radi preciznosti mogla povećati rezolucija kamere. Povećanjem njene rezolucije objekti bi sadržali veći broj piksela i jasnije bi se izdvajali na slici, pa bi se njihov položaj mogao preciznije odrediti. Preciznost se može povećati upotrebom LE dioda umesto markera. LE diode bi davale stalan intenzitet svetlosti određene boje i bilo bi ih jednostavno izdvojiti na slici, jer bi se smanjio uticaj spoljašnje svetlosti. Kretanje robota bi se moglo olakšati ostvarivanjem bežične komunikacije, jer bi se na taj način izbegao kabl koji mu ograničava kretanje. Optimizacijom algoritma, odnosno kreiranjem algoritma za kretanje po najkraćoj putanji smanjilo bi se vreme neophodno za parkiranje robota.

Zaključak

Sistem je testiran za nekoliko položaja robota i uspešno funkcioniše, tj. robot se iz svih položaja ispravno parkira na zadato parking mesto. Osnovni problem je neravnomerno osvetljenje i spora obrada slike. Neravnomerno osvetljenje menja intenzitet boje piksela, što dovodi do pojave šuma, smanjenja preciznosti obrade i uspešnosti rada sistema. Ovaj nedostatak se može izbeći upotrebom LED-ova umesto markera i inicijalizacijom. Zbog nesavršene mehanike robota i vremena koje je zadato kao radno stanje motoru, neophodno je uvođenje opsega vrednosti koordinata koje robot prilikom kretanja mora da zadovolji. Sistem bi se mogao unaprediti realizacijom u drugom programskom jeziku kao što je C, gde bi se obrada slike vršila brže ili realizacijom algoritma za kretanje po kraćoj putanji.

Literatura

web 1. http://www.hometrainingtools.com/robo-pica-2-0-robot-kit-with-c-programming/p/KT-IROBPIC

web 2. http://www.inexglobal.com

web 3. http://www.microchip.com

web 4. http://softpixel.com/čcwright/programming/colorspace/yuv

web 5. https://www.sparkfun.com/datasheets/Robotics/Robo-PICA2007.pdf

Bojana Nenezić

Self Parking Algorithm for PICA Robot

This paper describes a system for automated car parking. The principle of the system is based on finding the coordinates of the center of mass of the robot and determining its position relative to the parking place. A camera is used as a sensor for determining the current position of the robot. Data obtained from the camera are sent to a computer, which processes them and sends commands to control the motor. PIC robot contains two DC motors, which are controlled by the microcontroller. Engines run tracks and thus the robot changes its position.

In this paper, our approach was to find a trajectory that leads the robot to a certain point on the surface where it begins to move straight to the center of the parking space.

The apparatus consists of a camera, PIC robot (web1), PC and parking surface (Figure 1). The principle diagram is shown on Figure 2. For the better lo-

cating and positioning an additional part is attached to the robot. The front of the robot is marked with a black label, and the back is marked with a green label. For robot control the ready-made code written in a programming language MikroC is used. It controls the engine. The communication between the robot and the computer is established through a serial port. The system is implemented in Matlab.

The work procedure consists of two parts:

1. image processing
2. implementation of algorithms for robot motion

The image processing can be divided into the following phases:

A. sampling colors of the surface, front and back markers on the robot and parking lots
B. surface separation and image binarization
C. finding the largest object of each color
D. finding the center of mass of allocated objects
E. determining coefficients axis of the robot and parking lots.

The image is processed in the YUV format (web4).

For the accurate allocation of different color objects, it is necessary to determine the range of pixel values for each object. Determining the scope of pixels is performed during initialization as each object is marked (Figure 3).

In order to determine the position of the robot it is necessary to create a binary image and allocate objects on it. The following objects are separated: a green marker, black marker, parking place and parking surface (Figure 4).

The algorithm enables the robot to be parked from any position on the given parking place. The Algorithm is composed of 6 steps (Figure 5), so the robot alternately uses rotation and translation movement commands.

An example of a situation is shown in Figure 6, and the algorithm is shown in Figure 7.

The system has been tested for several robot positions and it was successful. The robot can be properly parked from any position at the given parking place. The main problems are the uneven lighting and the slow image processing. Uneven illumination changes color intensity pixels, which causes ghosting and image noise and the reduction of the accuracy and efficiency of the system.

The average value of image processing speed is 0246 seconds (4065 frames per second). The most time-demanding functions are functions for creating binary images and the determination of the center of mass. Due to the relatively slow image processing, the move control for the robot arrives at longer intervals, so the robot moves intermittently.

The robot successfully executes rotation commands, but a certain deviation occurs. The mean value of the angle for the rotation command of 90 degrees is 76.3 degrees. The standard deviation from the mean is 14.4 degrees, or 6.56%. The error occurs because the range of tolerance is specified for the rotation command, which is ±15 degrees (Figure 1).

The center of mass is determined with the help of Matlab built-in functions. The program calculates the values of the coordinates of the center of mass with sufficient precision so the robot is parked on a given place. The deviation from the actual coordinates of the center of mass values is caused by the uneven lighting that affects changes of the shape and size of allocated objects. Actual values for center of mass were compared with the program calculated values of the center of mass.