

# LOKALIZACIJA RAVNE LINIJE U SLIKOVNOJ SEKVENCI

Vrhovski Z.<sup>1</sup>, Herčeki R.<sup>2</sup>

<sup>1</sup>Visoka tehnička škola u Bjelovaru, Bjelovar, Hrvatska

<sup>2</sup>Tehnička škola Bjelovar, Bjelovar, Hrvatska

**Sažetak:** U ovom radu opisana je lokalizacija ravne linije u slikovnoj sekvenci. Za dohvata slike korištena je USB kamera. Dohvaćena slika predstavljena je u obliku matrice RGB boja. Nakon dohvata slike slijedi obrada slike: dohvaćena slika prebacuje se u sivo područje i filtrira usrednjavanjem. Nakon usrednjavanja slike, detektiraju se rubovi Sobelovim gradijentnim postupkom, čime je slika spremna za lokalizaciju objekata. Za lokalizaciju ravne linije koristi se linijska Houghova transformacija.

**Gljučne riječi:** vizualna povratna veza, detekcija rubova, Sobelov algoritam, Houghova transformacija

**Abstract:** This paper describes straight line localization from an image sequence. The images are captured using a USB camera. The captured images are represented by an RGB matrix. After the image acquisition, the images are processed using various algorithms. The captured image is first transformed into grayscale and filtered out using a median filter. After the filtering, the edge detection procedure is done using the Sobel gradient method, making the images ready for object localization. For straight line localization, a Hough transformation is applied.

**Key words:** visual feedback, Edge Detection, Sobel algorithm, Hough transform

## 1. UVOD

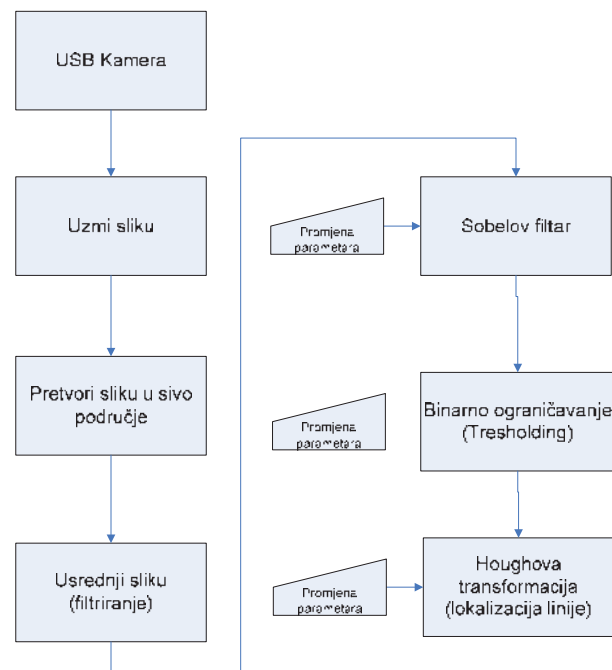
Jedan od zahtjeva koji se nameće mobilnim robotima je autonomnost njihova gibanja u nepoznatom prostoru. Mogućnost autonomije gibanja u nepoznatom prostoru omogućuje lokalizacija ravne linije u slikovnoj sekvenci. Mobilni roboti kod kojih se može koristiti ovaj vid autonomije su uslužni roboti usisavači, baloni tipa Zeppelin [1], roboti za pranje podova, autonomni viljuškari i drugi. Ravna linija u nepoznatom prostoru gibanja mobilnog robota je marker (nije nužno korištenje markera tipa ravne linije) koji se može nalaziti ili na podu ili na stropu nepoznatog prostora. Ako se pod ili strop snima kamerom tada se može napraviti algoritam koji će slijediti ravnu liniju na podu ili stropu. Mobilni robot potrebno je opremiti USB kamerom (ETHERNET kamerom ili kamerom nekog drugo tipa sučelja) koja je priključena na mikroručaladni sustav na kojem se izvršava algoritam lokalizacije ravne linije. Nakon

lokalizacije ravne linije treba napraviti algoritam kojim će se mobilni robot kretati duž ravne linije, što je tema sljedećeg rada koji će se nadovezati na ovaj rad.

U prvom dijelu rada opisan je način dohvaćanja slike s kamere. Iz dohvaćene slike izdvajaju se zanimljivi objekti. Drugi dio rada bavi se eksperimentima i testiranjima algoritma za lokalizaciju ravne linije iz slikovne sekvence. Za izradu algoritma lokalizacije ravne linije na slici korišten je programski paket *Visual Studio 2010* i sintaksa programskog jezika C++. Za dohvaćanje slike s kamere korištene su biblioteke *OpenCV* [3] koje se koriste u računalnom vidu.

## 2. OBRADA SLIKE U SLIKOVNOJ SEKVENCI

Dijagram toka algoritma za lokalizaciju ravne linije prikazan je na slici 1.



Slika 1. Dijagram toka algoritma za lokalizaciju ravne linije

Nakon što se dohvati slika s kamere potrebno je sliku pretvoriti u sivo područje. Siva slika prolazi kroz filtar za usrednjavanje koji smanjuje utjecaj šumova na slici. Na

filtriranu sliku primjeni se Sobelov gradijentni filtar koji služi za detekciju rubova na slici. Nakon gradijentne metode slijedi binarno ograničavanje slike, čime su istaknuti svi rubovi na slici te se na rubove može primijeniti Houghova transformacija za lokalizaciju ravne linije.

## 2.1. Dohvaćanje slike USB kamerom

Sekvenca slika dohvaća se USB kamerom *Logitech QuickCam* [2] (slika 2.). USB kamera čini samostalni sustav s ugrađenom centralnom procesnom jedinicom.



Slika 2. USB kamera Logitech

Slika se dohvaća kamerom preko USB porta koji je smješten na mikroročalnom sustavu kojim je opremljen autonomni mobilni robot (robot usisavač, čistač, autonomni viljuškar i dr.). Dohvaćena slika je u format RGB matrice intenziteta crvene, zelene i plave boje s vrijednostima od 0 do 255. Slika se dohvaća pomoću funkcija *OpenCV* (*Open Source Computer Vision Library*) [3]. *OpenCV* biblioteka sadrži velik broj funkcija (algoritama), od onih u čestoj upotrebi (crtanje geometrijskih likova i pisanje po slici, pretvorba između više formata slike i raznih filtara – računanje gradijenta, rubova, usrednjavanje slike medijan filtrom...) do specifičnih algoritama (proračun optičkog toka slike, računanje histograma, momenata, Houghove transformacije, Kalmanovih filtara...). Osim brojnih funkcija za obradu slika, *OpenCV* sadrži funkcije za dohvaćanje slike s kamere. Ova biblioteka funkcija za obradu slika dostupna je na internetu i uključena je u rad.

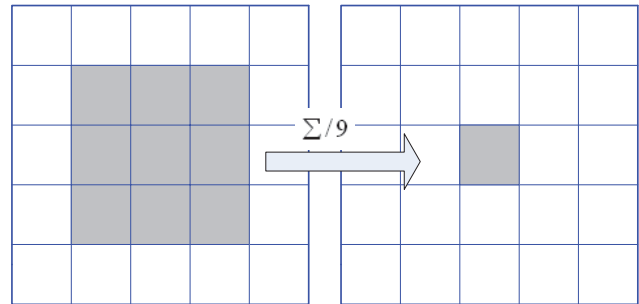
## 2.2. Pretvaranje slike u sivo područje

Pretvaranjem slike u sivo [4] područje smanjen je utjecaj sjene na liniju (ne javljaju se poteškoće pri prepoznavanju linije). Uz to, jednostavnije je i brže koristiti jednu matricu intenziteta boja nego tri matrice intenziteta boja (crvena, zelena i plava). Od tri matrice intenziteta pojedinih boja dobivena je jedna matrica intenziteta sive boje prema relaciji (1):

$$E[x][y] = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (1)$$

## 2.3. Usrednjavanje slike

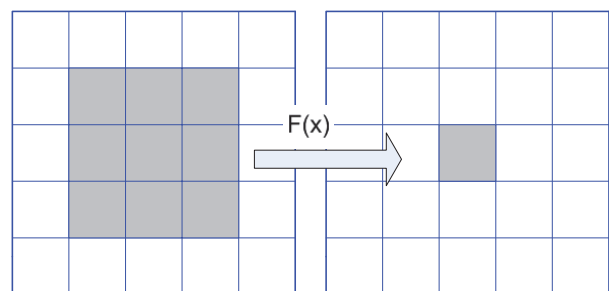
Pojava šuma na slici (pikseli slike koji predstavljaju poremećaj) nastoji se smanjiti korištenjem prostornog usrednjavanja slike. Najbolje filtriranje slike [4] dobiveno je primjenom medijan filtra (korišten je algoritam medijan filtra iz *OpenCV* biblioteke), ali kako dolazi do značajnog usporenja algoritma lokalizacije linije, primijenjen je jednostavniji filtar usrednjavanja. On se temelji na postupku koji u središnji piksel matrice 3\*3 stavlja srednju vrijednost intenziteta sive boje svih piksela u matrici (slika 3.).



Slika 3. Filtar usrednjavanja

## 2.4. Sobelov filtar

Pri analizi objekata na slici neophodno je razlučiti "objekte od interesa" od ostatka slike. Za pronalaženje linije na slici prvi korak je detekcija rubova ([1],[5],[6]). Da bi se dobili rubovi objekata na slici, treba primijeniti jedan od algoritama za detekciju rubova. Zbog jednostavnosti i brzine izvođenja odabran je Sobelov gradijentni filtar za detekciju rubova koji se temelji na postupku kojim se u središnji piksel matrice 3\*3 stavlja gradijent drugog stupnja promjene intenziteta na slici (slika 4.).



Slika 4. Djelovanje Sobelovog filtra rubova na matricu slike

Gradijent se računa u oba smjera prema relaciji (2):

$$\begin{aligned} G_x &= E[y-1][x-1] + 2E[y-1][x] + E[y-1][x+1] + \dots \\ &\quad - (E[y+1][x-1] + 2E[y+1][x] + E[y+1][x+1]) \\ G_y &= E[y-1][x+1] + 2E[y][x+1] + E[y+1][x+1] + \dots \\ &\quad - (E[y-1][x-1] + 2E[y][x-1] + E[y+1][x-1]) \end{aligned} \quad (2)$$

Ukupni gradijent se određuje prema izrazu:

$$G = \sqrt{G_x^2 + G_y^2} \quad (3)$$

Ovaj način usporava izvođenje algoritma (korjenovanje i kvadriranje decimalnih brojeva uzima više procesorskog vremena, što je posebno vidljivo kod slabijih mikroracunala). Da bi se povećala brzina proračuna, koristi se aproksimacija gradijenta zbrajanjem:

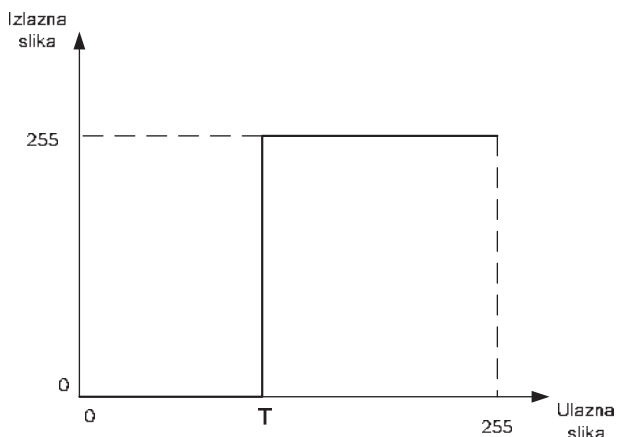
$$G = G_x + G_y \quad (4)$$

Kada se odredi matrica gradijenata, potrebno je odvojiti zanimljive objekte od podloge. Najjednostavnije je koristiti binarno ograničenje (engl. *thresholding*). Binarnim ograničenjem dobiva se binarna slika na kojoj su željeni objekti prikazani jedinicom (u ovom slučaju bijelom bojom, 255), a ostali nulom (u ovom slučaju crnom bojom, 0). Ovaj postupak može se opisati relacijom (5) i slikom 5.

$$v(x, y) = \begin{cases} 0 & , \text{ za } u(x, y) \leq T \\ 255 & , \text{ za } u(x, y) > T \end{cases} \quad (5)$$

Pri tome su:

1.  $T$  – prag odluke
2.  $u(x,y)$  – ulazna slika
3.  $v(x,y)$  – izlazna slika



Slika 5. Graf binarnog ograničenja (thresholding)

Relacija (5) se može opisati riječima: u matrici gradijenata svi pikseli, čija je vrijednost manja i jednaka od praga odluke ( $T$ ), proglašavaju se crnim pikselima (intenziteta 0), a svi ostali pikseli čija je vrijednost veća od praga odluke ( $T$ ) proglašavaju se bijelim pikselima (intenziteta 255). Na taj način je dobiven rub objekata. Što je prag odluke veći, intenzitet rubova na slici će biti manji.

## 2.5. Houghova transformacija

U automatiziranoj obradi slike česti problem na koji se nailazi je lokalizacija objekata kao što su linije, elipse, kružnice i slično. Često korištena Houghova transformacija [8] omogućuje lokalizaciju rubova objekata koji se nalaze na slici. Pretpostavka je da su prije transformacije detektirani rubovi. Kombiniranjem

rubnih piksela detektiraju se linije, a njihovim kombiniranjem i složenije konture. Za potrebe daljnje analize koristit će se osnovna linijska Houghova transformacija. Nakon detekcije rubova Sobelovim gradijentnim filtrom dobiva se slika rubova koja predstavlja ulazni podatak za Houghovu transformaciju. Radni prostor transformacije je ravnina s dvije vrste piksela:

- pikselima koji predstavljaju rub
- pikselima koji predstavljaju pozadinu

Zato se obično koriste monokromatske slike, iako to nije uvjet (u ovom slučaju koristi se slika u sivom području). Općenito se kroz svaki piksel ruba može provući beskonačno mnogo pravaca koje možemo prikazati kao:

$$y = ax + b \quad (6)$$

Sljedeći zadatak je odrediti koeficijente  $a$  i  $b$  tako da na tom pravcu leži što više rubnih piksela (ili dovoljan broj što može biti i je jedan od parametara transformacije). Zbog vrijednosti parametara  $a$  i  $b$  koji mogu poprimiti sve vrijednosti iz  $\mathbf{R}$ , češće korištena transformacija je transformacija pomoću polarnih koordinata. Ako se varijable  $x$  i  $y$  zamijene sljedećim relacijama:

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \end{aligned} \quad (7)$$

slijedi relacija za Houghovu transformaciju u parametarskom prostoru ( $\theta, r$ ):

$$r = x \cos \theta + y \sin \theta \quad (8)$$

Ako se kut  $\theta$  mijenja diskretno ( $1^\circ, 2^\circ, \dots, 360^\circ$ ) tada se dobiju različite vrijednosti  $r$ . Točke  $(x,y)$  koje leže na istom pravcu u parametarskom prostoru ( $\theta, r$ ) predstavljaju jednu točku. Cilj ovog algoritma je pronaći najdužu liniju na slici, odnosno liniju koju predstavlja najveći broj piksela. Prije samog algoritma pronalaska linije stvoreno je tzv. akumulatorsko polje Houghove transformacije koje se na početku inicijalizira na nulu ( $\text{Hough}[\theta][r]=0$ ). Za svaki rubni piksel  $(x,y)$  računaju se vrijednosti za  $r$ , pri čemu se  $\theta$  kreće u granicama od  $(1, 360^\circ)$ . Pri tome se  $r$  kreće u granicama od  $(0, 400)$ , što je posljedica rezolucije slike od  $320 \times 240$ . Za svaki rubni piksel  $(x, y)$  i za svaki  $\theta$  u granicama od  $(0, 360^\circ)$  računa se  $r$  iz relacije (8) te se vrijednosti svaki put u akumulatorskom polju (za trenutni  $r$  i  $\theta$ ) uvećaju za 1 ( $\text{Hough}[\theta][r]++$ ). Lokalni maksimum unutar akumulatorskog polja ( $r_m, \theta_m$ ) određuje zapis pravca:

$$y = -\frac{\cos \theta_m}{\sin \theta_m} x + \frac{r_m}{\sin \theta_m} \quad (9)$$

To je ujedno najduža i najizraženija linija na slici, odnosno linija predstavljena najvećim brojem rubnih (bijelih) piksela. Da bi eliminirali linije koje su manje u odnosu na liniju koju očekujemo, koristimo uvjet za postojanje linije, a to je broj glasova za pojedinu liniju.

Uvjet se svodi na definiranje donje vrijednosti lokalnog maksimuma koji prihvaća liniju kao traženi objekt.

U  $(\theta, r)$  prostoru uvijek će postojati barem jedan maksimum  $(\theta_m, r_m)$ . On će definirati najveći pravac unutar slike, ako je njegov broj glasova veći od donje granice lokalnog maksimuma. Sljedeći (lokalni) maksimumi definirat će pravce koji su uži i kraći (za njih je glasovalo manje točaka).

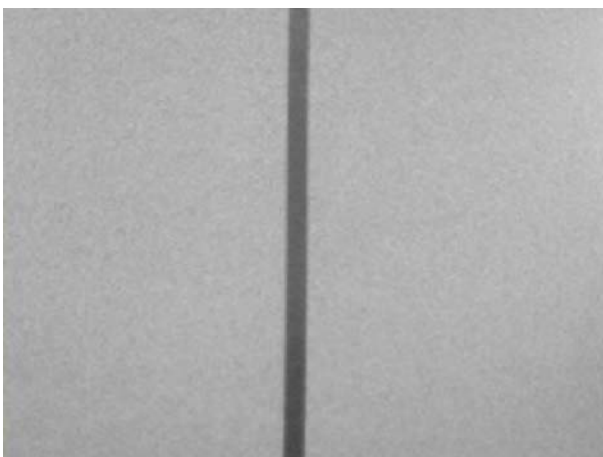
### 3. PROVOĐENJE ALGORITMA ZA LOKALIZACIJU RAVNE LINIJE

Slika dohvaćena s kamere (slika 6.) pretvara se najprije u sivo područje (slika 7.). Siva slika ulazni je podatak za detekciju rubova Sobelovim filtrom.



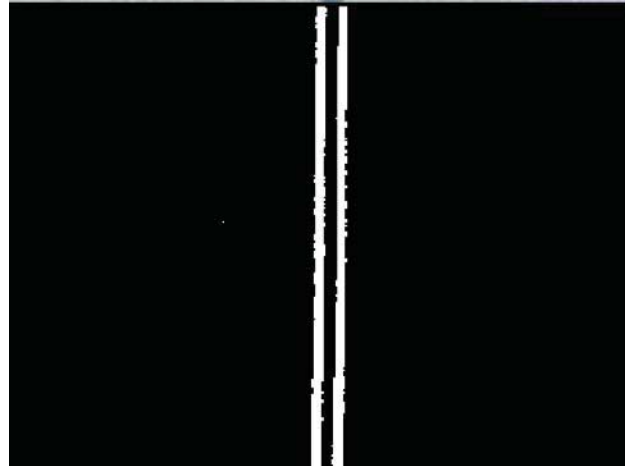
Slika 6. Slika dohvaćena s kamere

Problemi koji se javljaju pri korištenju Sobelovog gradijentnog filtra je osvijetljenje. Zbog jako osvijetljene podloge (ako podloga blješti) vrijednosti izračunatog gradijenta prema relaciji (2) će biti malog iznosa. Zato će zbog karakteristike binarnog ograničenja (slika 5.) za relativno velik prag odluke  $T$  slika biti bez rubova ili će oni biti jako slabo izraženi. Jedino rješenje koje se tada nalaže je smanjenje praga odluke  $T$ . S druge strane, ako je podloga dobro osvijetljena uz mali iznos praga odluke  $T$  i lažni rubovi će biti prikazani kao rubovi objekata na slici. Sobelov gradijentni filtar daje jako dobre rezultate uz dobru osvijetljenost i optimalni je prag odluke  $T$ .

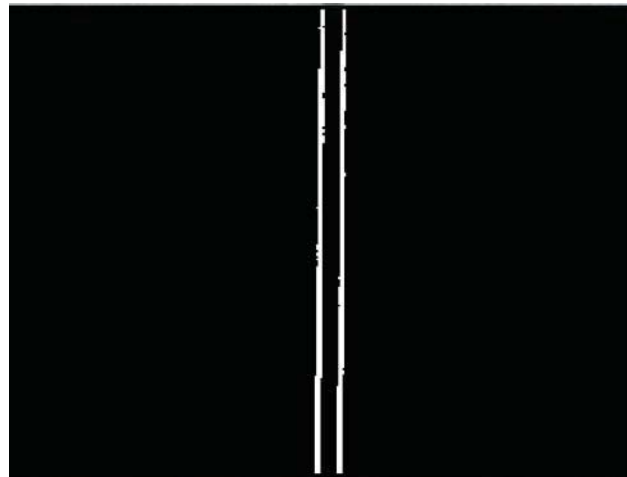


Slika 7. Slika dohvaćena s kamere u sivom području

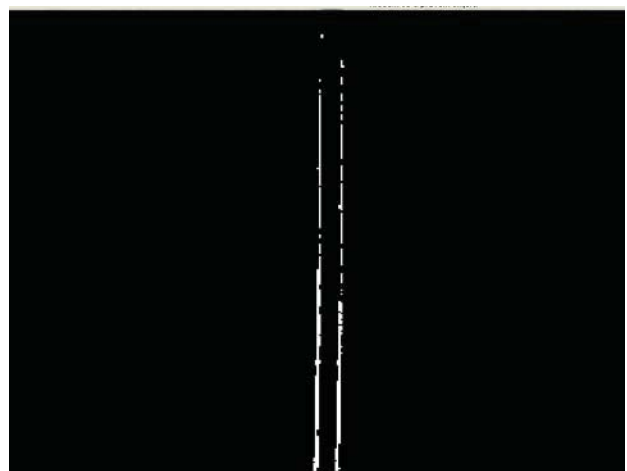
Optimalni prag odluke može se odrediti eksperimentalno. Primjeri na kojima se mogu vidjeti utjecaj različitih pragova odluke  $T$  kod algoritma binarnog ograničenja prikazani su na slikama 8., 9. i 10.



Slika 8. Sobelov filtar rubova (binarno ograničenje 70)



Slika 9. Sobelov filtar rubova (binarno ograničenje 150)

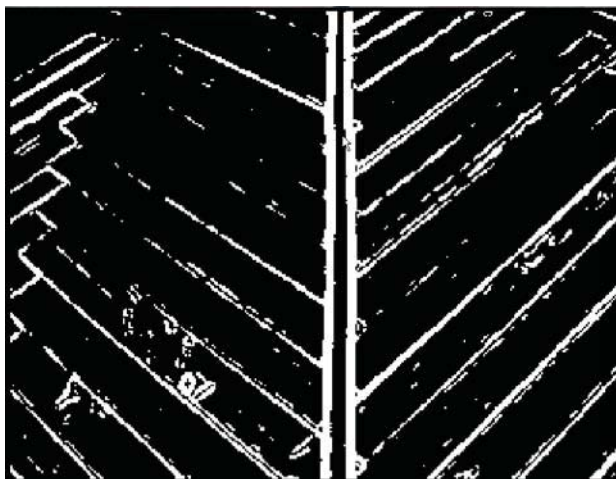


Slika 10. Sobelov filtar rubova (binarno ograničenje 190)

Na slici 8. prikazan je rad Sobelovog gradijentnog filtra uz prag osjetljivosti 70. Budući da je podloga uniformna, rezultat algoritma je prihvatljiv. Na slikama 9. i 10. prikazan je rad algoritma za još dvije vrijednosti praga



odluke (150, 190). Najbolji rezultati ostvareni su s pragom odluke 150. Kada se uniformna podloga zamijeni s podlogom parketa, tada dolazi do problema detekcije mnoštva linija koje predstavljaju rubove elemenata parketa. Na slici 11. prikazana je detekcija rubova uz mali prag odluke (70). Da bi se postigli prihvatljivi rezultati treba povećati prag osjetljivosti. Na slici 12. prikazan je rad Sobelovog gradijentnog filtra uz prag odluke 150. Na ovaj način riješen je problem višeznačnosti linije.



Slika 11. Sobelov filter rubova (binarno ograničenje 70, podloga parket)



Slika 12. Sobelov filter rubova (binarno ograničenje 150, podloga parket)

Lokalizacija ravne linije na temelju detekcije rubova prikazana je na tri primjera. Slika 13. prikazuje lokalizaciju linije paralelne s kamerom, a slike 14. i 15. prikazuju lokalizaciju linija koje nisu paralelne s kamerom.



Slika 13. Lokalizacija ravne linije (1)



Slika 14. Lokalizacija ravne linije (2)



Slika 15. Lokalizacija ravne linije (3)

#### 4. ZAKLJUČAK

U današnje vrijeme upravljanje vizualnom povratnom vezom nalazi veliku primjenu u području praćenja objekata i navođenja mobilnih robota prema nekom cilju. Razlog tomu je velika računalna moć koja omogućuje procesiranje složenih algoritama za obradu slike. Prvi korak izrade algoritma za lokalizaciju ravne linije je razlučivanje unaprijed zadanog objekta (u ovom slučaju

radi se o ravnoj liniji) od ostalih objekata na slici. Korištenjem Sobelovog gradijentnog postupka detektiraju se rubovi objekata na slici. Rubovi na slici u tom slučaju mogu biti kontinuirani ili isprekidani. Širina rubova direktno ovisi o parametru binarnog ograničenja. Nakon detekcije rubova slijedi lokalizacija ravne linije na slici. Pri tome je korištena linijska Houghova transformacija. Cilj Houghovog algoritma je pronaći pravac koji je predstavljen najvećim brojem piksela koji leže na tom pravcu. Donju granicu broja piksela koji predstavljaju pravac moguće je mijenjati s obzirom na praćeni prostor. Prednost Houghove transformacije je u tome što ona može lokalizirati pravac, iako se on sastoji od više segmenata (razlomljeni pravac). Lokalizirani pravac ulazni je podatak za algoritam navođenja mobilnog robota u smjeru ravne linije.

## 5. LITERATURA

- [1]Vrhovski, Z. ; Jakus, I. ; Kovjanović, N. :*UPRAVLJENJE BALONOM TIPRA ZEPPELIN POMOĆU VIZUALNE POVRATNE VEZE*, REKTOROVA NAGRADA 2008., Zagreb, 2008.
- [2][http://www.nzoczone.com/product\\_info.php?cPath=70&products\\_id=4503&osCsid=0c2f92e8ef3892c54c0486d0c52902e5](http://www.nzoczone.com/product_info.php?cPath=70&products_id=4503&osCsid=0c2f92e8ef3892c54c0486d0c52902e5), USB kamera, Logitech QuickCam Chat
- [3]<http://opencvlibrary.sourceforge.net/>, SourceForge.net: Open Computer Vision Library

[4]Medenjak, M.: *VIZUALNA POVRATNA VEZA U SUSTAVIMA UPRAVLJANJA FORMACIJOM POKRETNIH OBJEKATA*, diplomski rad, Zagreb, 2007.

[5]<http://www.pages.drexel.edu/%7Eweg22/edge.html>, Green, B.: Edge Detection Tutorial, 2002.

[6]Arbenz, P.: *Edge detection with MMX technology*, Swiss Federal Institute of Tehnology Zurich, 2007.

[7]Rahim, et al.:*The Use of Hybrid Tehnique: Tresholding and Edge Detection for Indentifying River from Aeral Photo*, Jurnal Teknologi, Universiti Teknologi of Malasya, <http://eprints.utm.my/1939/1/JURNALDIS%5BB1%5D41.pdf2004>.

[8]<http://www.cee.hw.ac.uk/hipr/html/hough.html>, Hough Transform

### Kontakt:

Zoran Vrhovski, mag. ing. eit.  
Visoka tehnička škola u Bjelovaru  
Trg Eugena Kvaternika 4  
43000 Bjelovar  
mob: 0917842199  
tel: 043/241-185

mail:  
[zvrhovski@vtsbj.hr](mailto:zvrhovski@vtsbj.hr)  
[robert.herceki@skole.hr](mailto:robert.herceki@skole.hr)