Zaznavanje ovir s strukturirano svetlobo – lasersko črto

Aleš Medvešek, Peter Lepej
Simon Klančnik, Karel Benkič, doc.dr. Peter Planinšič (mentorji)
Fakulteta za elektrotehniko računalništvo in informatiko, Fakulteta za strojništvo
Smetanova 17, 2000 Maribor, Slovenija
ales_medvesek@yahoo.com, peter.lepej@gmail.com

Obstacle detection with structured light – laser line

Obstacle detection is needed to increase safety of speech-controlled wheelchair in case of third person would give unwanted command witch could be recognised by wheelchair. To detect an obstacle we project structured light on the floor and shot it by camera. Our method of image processing is used to recognise structured light. Position of laser line on captured picture defines obstacles.

1 Uvod

Na študijski smeri mehatronika že nekaj let razvijamo avtonomen glasovno vodeni invalidski voziček VOIC [1]. Voziček je namenjen predvsem ljudem, ki so močno gibalno ovirani in zato ne morejo uporabljati klasičnega invalidskega vozička. Razpoznava govora se izvaja s pomočjo usmerjene nevronske mreže, ki pa v vseh situacijah ni popolnoma zanesljiva. Zato je bil na voziček za prepoznavanje in izogibanje oviram dodan varnostni sistem (VS) zgrajen na podlagi ultrazvočnih senzorjev. Rezultati testiranj VS so pokazali, da sami ultrazvočni senzorji v vseh situacijah niso zanesljivi. Tako smo se odločili za dopolnitev ultrazvočnih senzorjev s sistemom, ki zaznava ovire s pomočjo strukturirane svetlobe.

Zaznavanje ovir na podlagi strojnega vida je najpogosteje izvedeno z detekcijo robov. Sami robovi na zajeti sliki pa ne povedo ničesar o globini, razen, če uporabimo najmanj dve kameri oz. stereo vid, kar deluje zelo podobno kot človeško dojemanje tretje dimenzije.

V tem članku bomo predstavili delo, ki je nadaljevanje članka: »Zaznavanje ovir s pomočjo strukturirane svetlobe« [2]. V omenjenem članku je opisana strojna in programska oprema, princip zaznavanja ovir s pomočjo strukturirane svetlobe z znanimi geometrijskimi lastnostmi, obdelava slike s progovno metodo in zgoščevanjem, detekcija robov ovire z mehko logiko, razdelan je bil primer zaznavanja laserske pike. Naše nadaljnjo delo temelji predvsem na razširitvi sistema na zaznavanje ovir s projicirano lasersko črto. V ta namen smo za uporabo v naši aplikaciji razvili dva algoritma obdelave slike, ki sta sposobena zanesljivo določiti položaj laserske črte na sliki in ju bomo v tem delu tudi podrobneje predstavili. Do učinkovitega in zanesljivega algoritma smo prišli z opazovanjem obnašanja laserske svetlobe pri različnih okoljih in pogojih. Po ugotovitvi, da za naš problem HSL barvni prostor [4] ni bistveno učinkovitejši smo ostali pri RGB barvnem modelu [3].

Članek je organiziran v naslednjem vrstnem redu. V drugem poglavju bomo kratko predstavili princip zaznavanja ovir s pomočjo strukturirane svetlobe. Algoritem obdelave slike bomo podrobneje prestavili v tretjem poglavju in sicer bomo predstavili dva različna načina obdelave slike (metoda trendov in algoritem sledenja laserski črti), ki smo jih razvili in sta se oba izkazala, kot učinkovita. Od metode trendov pričakujemo malo večjo učinkovitost prepoznave laserske svetlobe, od algoritma sledenja pa pričakujemo hitrejšo obdelavo. Rezultate in ugotovitve bomo predstavili v četrtem poglavju.

2 Princip zaznavanja ovir

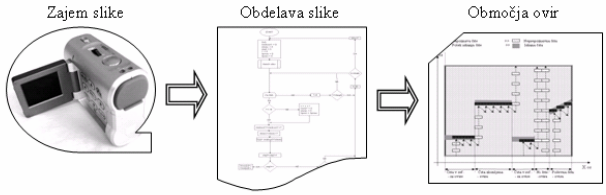
Princip zaznavanja ovir je natančno opisan v članku »Zaznavanje ovir s pomočjo strukturirane svetlobe« [2], zato bomo tukaj podali le kratek povzetek. Ovire zaznavamo s pomočjo kamere in izvora laserske svetlobe, ki v našem primeru projicira lasersko črto. Laserski izvor je nameščen nad kamero in projicira črto na tla pred voziček. V primeru, ko se pred vozičkom pojavi ovira, se na sliki, ki jo zajamemo s kamero, spremeni pozicija laserske črte. Če poznamo velikost te spremembe, lahko iz znanih geometrijskih lastnosti sistema določimo položaj zaznane ovire Prototip sistema prikazuje slika 1.

Sistem je v prvi fazi potrebno kalibrirati ali drugače povedano, moramo določiti položaj laserske črte na sliki, ki jo zajamemo v trenutku, ko pred sistemom ni ovir. Ta položaj predstavlja referenčni položaj črte na zajeti sliki in z njim primerjamo črto na vsaki novo zajeti sliki. Iz tega podatka lahko ugotovimo ali je ovira prisotna in določimo njen položaj.

Slika 1: Laboratorijski prototip sistema z linijskim izvorom laserske svetlobe.

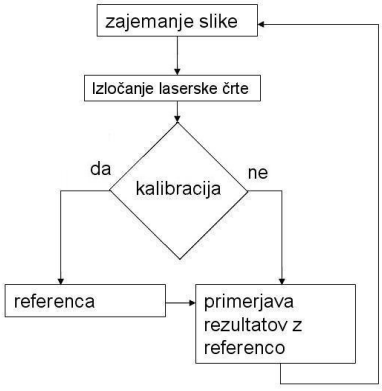
3 Obdelava slike

Za lažje razumevanje opisanih algoritmov, ki bodo podani v nadaljevanju, bomo najprej podali nekaj splošnih osnov. Slika je v digitalni obliki predstavljena kot matrika, v kateri je vsak njen element (piksel) predstavljen s tremi byti. Vsak byte predstavlja eno iz med RGB (rdeča, zelena in modra) barv ter kombinacije od 0 do 255, kar pomeni odtenke posamezne barve [3]. Za pravilno delovanje celotnega sistema je potrebno lasersko črto izločiti iz zajete slike, kar pa ni vedno enostavno, saj se v realnem okolju lahko srečamo z različnimi motnjami, ki vplivajo na uspešnost izločanja laserske črte. Največje težave predstavlja neenakomerna osvetljenost področja na katerem ugotavljamo prisotnost ovir in različne barve podlag iz katerih je potrebno izločiti položaj laserske črte.



Slika 2: Potek obdelave slike

Na sliki 2 je prikazan postopek delovanja sistema. Vsakih 60 ms zajamemo novo sliko in jo obdelamo z algoritmom. V začetku algoritem kalibriramo, da določimo referenčni položaj laserske črte na sliki zajeti v trenutku, ko pred sistemom ni ovire. V normalnem načinu delovanja pa na novo zajeti sliki določimo položaj laserske črte in ga primerjamo z referenčnim slika 3.



Slika 3: Diagram poteka izvajanja algoritma

3.1 Izločanje laserske črte z metodo trendov

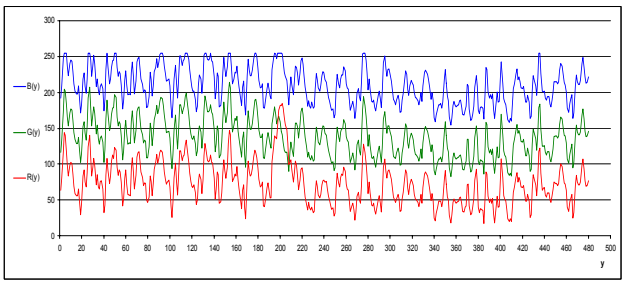
To metodo smo razvili za izločanje laserske črte ob predpostavki, da laserska črta na sliki zavzema približno prečno lego glede na smer vožnje. Dobra lastnost metode je zmanjševanje vpliva neenakomerne osvetljenosti in

neobčutljivost na širok spekter barv, ki jih laserska črta lahko zavzame pri različnih svetlobnih pogojih in barvah podlage. Za zmanjševanje vpliva neenakomerne osvetljenosti na rezultat filtriranja laserske svetlobe smo razvili lasten postopek pri katerem barvne kanale R, G in B (RGB barvni model [3]) pretvorimo v eno vrednost, ki jo bomo v nadaljevanju članka označevali z C, po enačbi:

$$C = R - \frac{G+B}{2} \qquad (1)$$

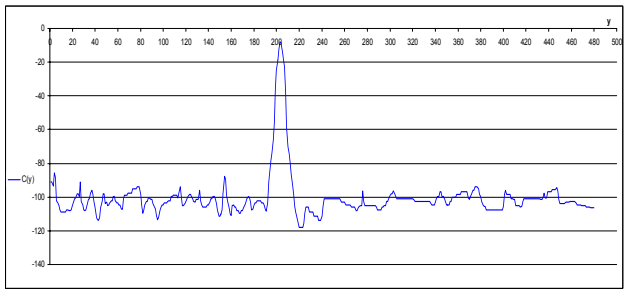
S preslikavo poudarimo rdečo komponento in istočasno zmanjšamo vpliv dejavnikov kot so: neenakomerna osvetljenostost, hrapavost, ukrivljenost površin, razni položaji izvorov svetlobe, ki otežujejo zaznavanje laserja (primer prikazuje slika 8).

Na sliki 4 je graf odtenkov R, G in B barvnih komponent za 264-ti stolpec slike 8, ki smo ga za prikaz označili z belo navpično črto. Vidimo lahko osciliranje vsake komponente posebej, kar predstavlja problem pri postavljanju fiksnih pravil za izločanje laserske svetlobe. Opazimo pa lahko, da se posamezne komponente pri prehodu iz manj osvetljenega področja v bolj osvetljeno povečajo za približno enako vrednost. Če te barvne komponente 264-tega stolpca transformiramo z enačbo (1) dobimo graf, ki je prikazan na sliki 5. Iz njega že na prvi pogled mnogo lažje opazimo odstopanje, ki predstavlja laser.



Slika 4: Barvni kanali za 264-ti stolpec slike 8.

Pojavijo pa se nekatere dodatne težave, saj C lahko zavzema vrednosti med 255 in -255. Na z laserjem osvetljenih mestih, so vrednosti običajno med -20 in 220. Zato še ne moremo določiti enostavnega praga, po katerem bi zanesljivo izločili lasersko svetlobo. Na izrazito rdečih podlagah, kjer ima R komponenta že zaradi podlage visoko vrednost, pod vplivom laserja ne more bistveno narasti, zato pa se lahko povečajo G in/ali B barvna komponenta, kar pa zaradi minusa v enačbi (1) privede do obrnitve grafa na sliki 5.



Slika 5: Preslikava komponent RGB v graf C odpravi večino težav zaradi neenakomerne osvetljenosti.

Pred voziček projiciramo lasersko črto, zato smo se pri snovanju algoritma po metodi trendov, osredotočili predvsem na izločanje laserske svetlobe v vsakem stolpcu posebej (obravnavali bi lahko tudi dva ali več sosednjih stolpcev). Iz slike 5 je razvidno, da laser povzroča nastanek ozkega pulza, ki se lahko pojavi na precej širokem intervalu. Zato uvedemo trend poteka vrednosti C, ki pa mora ignorirati nenadna odstopanja. Torej potrebujemo nekakšno glajenje signala C. Izvedli smo ga po enačbi (2).

$$T1(y) = T1(y-1) + \frac{C(y) - T1(y-1)}{K} \qquad (2)$$

V enačbi (3) *T1* predstavlja trend, ki sledi *C* od vrha slike proti dnu, *y* je koordinata piksla, konstanta *K* pa določa stopnjo glajenja in jo določimo glede na širno laserske črte. Na vrhu slike *T1* postavimo na vrednost *C*.

$$T1(0) = C(0) \qquad (3)$$

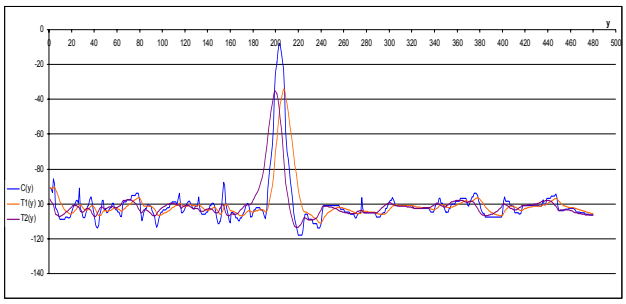
Trende lahko izračunamo v obe smeri. Od dna proti vrhu računamo trend po enačbi (4).

$$T2(y) = T2(y+1) + \frac{(C(y) - T2(y+1))}{K} \qquad (4)$$

T2 na dnu slike postavimo na enako vrednot kot *C*.

$$T(H) = C(H) \qquad (5)$$

H v enačbi (5) predstavlja višino slike v pikslih. Tako dobljeni algoritem trenda se obnaša podobno kot člen prvega reda. Trenda *T1* in *T2* določata interval, znotraj katerega se načeloma nahaja *C*. Dovolj velika odstopanja signala *C* izven trendov pa pomenijo, da je na tem območju laserska črta, kot lahko vidimo na sliki 6, kjer je signal *C* moder, *T1* je oranžen, *T2* pa je označen z vijoličasto barvo.

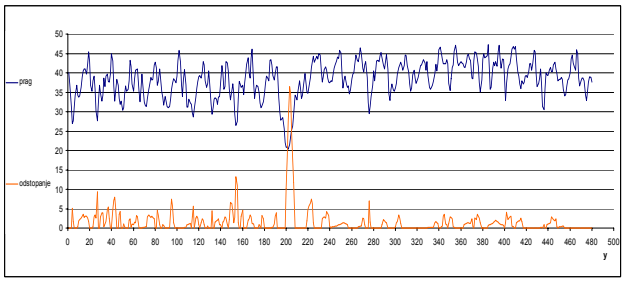


Slika 6: Signal C z dodanimi trendi

Iz vrednosti *C*, *T1* in *T2* izrišemo graf na katerem so odstopanja C izven intervala, ki ga določata oba trenda, prikazana absolutno na sliki 7. Vrednosti C znotraj intervala pa so postavljene na nič. Upoštevati pa je potrebno tudi, da na dobro osvetljenih slikah ne moremo pričakovati tako velikih odstopanj, kot pri slabše osvetljenih. Iz tega razloga smo vpeljali adaptiven prag, glede na R komponento za vsak piksel posebej, po enačbi 6.

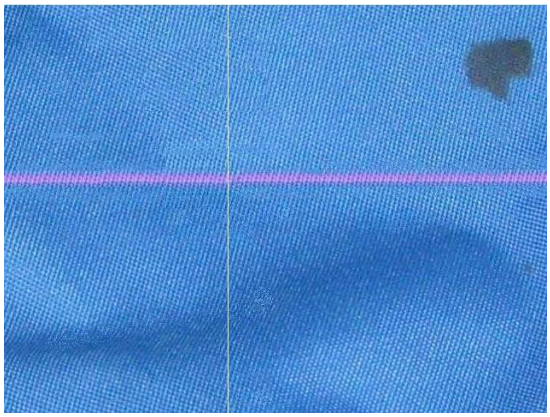
$$prag = -k \cdot R + n\,, \qquad (6)$$

kjer so k in n konstante izbrane eksperimentalno. V našem primeru smo jim priredili vrednosti $k = 0.16$ in $n = 50$.



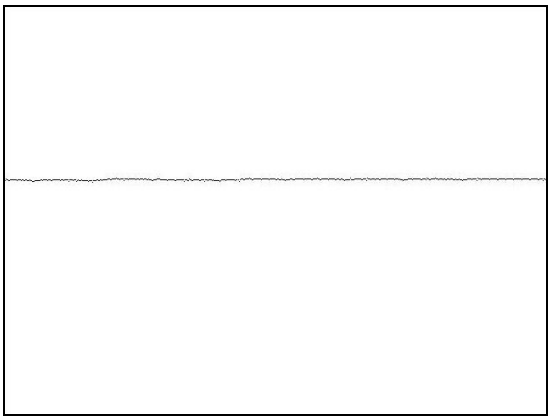
Slika 7: Odstopanja C izven intervala (T1, T2) prikazana absolutno in prag določen na podlagi rdeče barve

Za večjo zanesljivost izločanja laserske črte pa poleg praga sproti beležimo tudi velikost odstopanja preko praga in če se presežki pojavijo na več mestih upoštevamo kot posledico laserja samo piksel, kjer je presežek največji. Na ta način upoštevamo samo eno lasersko črto in izločimo tudi morebitne odboje. Gornji grafi prikazujejo delovanje algoritma in se na nanašajo na 264-ti stolpec spodnje slike 8.



Slika 8: Vrhovi hrapave podlage so skoraj beli, doline so temnomodre, bela vertikala označuje 264-ti stolpec.

Rezultat, ki ga dobimo, ko sliko 8 obdelamo z metodo trendov, je prikazan na sliki 9.



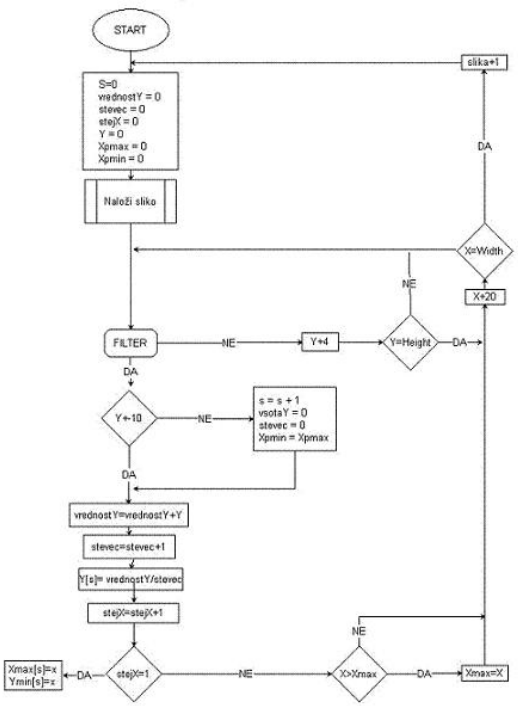
Slika 9: Črna barva označuje izločeno lasersko črto bela pa ozadje (robovi so dodani zaradi preglednosti).

3.2 Izločanje laserske črte z algoritmom sledenja

Določiti je potrebno tudi višino, kje se nahaja laserska črta in kako dolga je. Podatke širine in višine uporabimo za identifikacijo prisotnosti ovire tako, da primerjamo vrednosti z

referenčno pozicijo laserske črte, ki jo določimo z algoritmom kalibracije. Tukaj so pojavijo različni scenariji: lom črte, črta izgine iz vidnega polja, poševna črta in spremenjena širina črte glede na naklon terena, na katerega je črta projicirana. Vsa ta stanja smo morali upoštevati in jih zadovoljivo rešiti.

Prepoznavo laserske črte smo izvedli z pragom (threshold), ki je pogojeval nadaljevano obdelavo pikslov oziroma njihovih položajev v matriki. Prag je seštevek vseh maksimalnih vrednostih R, G ali B posameznih pikslov v oknu:



Slika 10: Diagram poteka izločanja laserske črte z algoritmom sledenja

$$\mathrm{MaxB} = \mathrm{M} \times \mathrm{n} \qquad (7)$$

V enačbi 7 *n* predstavlja število pikslov v oknu, *M* maksimalno vrednost odtenkov posamezne barve, ki je 255 in *MaxB* je maksimalna možna vrednost barve: rdeče, zelene ali modre. Tako lahko izvajamo prepoznavo rdeče barve v idealnem okolju, ki pa v realnosti ni mogoča, zaradi motenj iz okolice. Zato to dobljeno vrednost prilagodimo tako, da jo ustrezno zmanjšamo s faktorjem, ki ga določimo s poizkušanjem. Na takšen princip poteka zaznavanje črte, z razliko, da je *M* zaznan oziroma prebran iz pikslov slike. Nato se izračunana vrednost primerja z *MaxB*, če je vrednost večja ali enaka od *MaxB* je to črta.

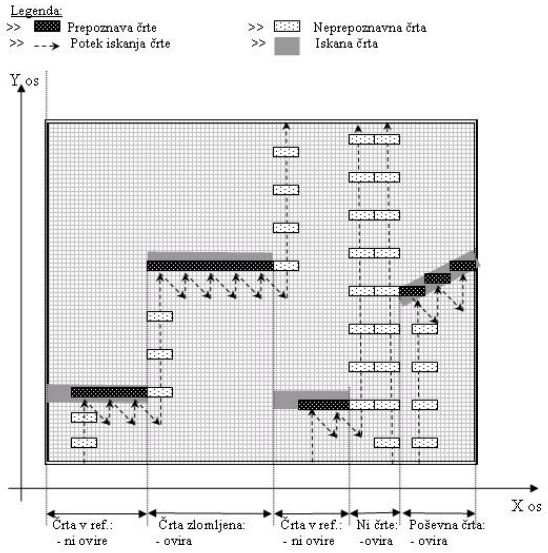
Algoritem prepoznavanja poteka s "potovanjem filtra" v obliki okna (matrika uteži pikslov) po y osi navzgor (slika 11). Ko okno zazna črto je pogoj izpolnjen in podatek se zapiše v obliki položaja zaznanega piksla (x,y). Tu Y poda višino črte, ki je enaka referenčni vrednosti y, če ni prisotne ovire. X pa poda začetek črte. Nadaljnjo obdelavo smo poimenovali sledenje črte. Algoritem najprej prepozna začetek črte, nakar sledi črti tako, da se filtrirno okno prestavi v naslednji stolpec in nekaj korakov po vrstici nazaj ter ponovno preveri če je pogoj prag izpolnjen. V primeru ko, je pogoj izpolnjen se koordinate shranijo, filter pa skoči v naslednji stolpec, če pa pogoj ni izpolnjen nadaljuje pot navzgor, dokler ne zazna črte ali doseže vrha. Ta postopek prikazuje slika 11. Zaradi možnih odstopanj (različna osvetlitev okolice, različna barvna podlaga, manjši nakloni črte) povzroči slabši učinek filtra in črta ni zaznana na dejanski višini. Zato pri manjših vrednostih y od začetne oziroma referenčne vrednosti (npr: y=+-10), določimo pogoj, s katerim predpostavimo, da to še spada v osnovno črto, kar pa je večje od tega pogoja pa smatramo za lom črte, oziroma prisotnost ovire. Manjša odstopanja rešimo s preprostim izračunom povprečne vrednosti pikslov z manjšimi odstopanji. Piksle, ki imajo večje odstopanje se zapišejo v novo matriko, kar predstavlja oviro. To nadaljujemo čez celotno širino slike. Tako smo določili lome črte, imamo torej več matrik in za vsako vrednost y. Sedaj moramo določiti od kod do kod traja lom črte za določen y, to naredimo z iskanjem največjega in najmanjšega števila x, ki vstopa v okno.

Za zanesljivejšo prepoznavo črte smo izvedli naslednje:

Za prepoznavo črte smo vzeli matriko sosednjih pikslov v obliki horizontalno podaljšanega pravokotnika (npr: 5×2, 10×3..), saj s tem približamo obliko črti in večjo verjetnost zaznave prave črte.

Pomagamo si lahko tudi z določitvijo maksimalne širine črte. Postopka se lotimo tako, da najprej testiramo laser pri različnih kotih projicirana črte, rezultat tega testa bo maksimalna širina, ki je mogoča. Tako algoritmu dodamo pogoj, da črta ne more biti širša.

Z zaznavo pikslov z oknom, lahko dejansko določimo naklon črte, ki je določen z dolžino matrike glede na maksimalno možno debelino črte.



Slika 11: Potek prepoznavanja sledenja laserske črte

Z algoritmom sledenja dejansko sledimo črti in s tem obdelamo približno 35% slike. Tako pridobimo na procesorskem času, ki je bistvenega pomena pri obdelavi v realnem času.

4 Rezultati

Proces obdelave zajetih slik smo implementirali v programskem jeziku C #. Slike, ki jih obdelujemo so resolucije 720×567 pikslov. Snovanje algoritmov smo izvajali v programskem paketu Matlab 7.1 Algoritma smo testirali na osebnem računalniku AMD Athlon2400+, s 512 MB delovnega spomina.

Izvedli smo ločena testiranja algoritmov trenda in sledenja. Z algoritmom "Trend" smo lahko vsakih 80 ms obdelali novo sliko, z algoritmom "Sledenja" pa 50 ms. Za uporabo v naši aplikaciji potrebujemo čim krajši čas obdelave, zato je v tem pogledu sledenje boljše. Algoritma smo prev tako testirali v različnih realnih okoljih in pogojih pri čemer se je algoritem trenda izkazal kot zanesljivejši in učinkovitejši.

5 Zaključek

V tem članku sta predstavljena dva algoritma, ki sta se oba izkazala kot izredno robustna in učinkovita. Prednosti pri izločanju črte po metodi trendov so, da zelo učinkovito loči lasersko črto od ozadja, vendar porabi preveč procesorskega časa, za uporabo v realnem času v naši aplikaciji. Algoritem sledenja črti je časovno ugoden, vendar manj zanesljiv.

V prihodnosti bomo poiskali kompromis med obema algoritmoma in to implementirali na zmogljivem digitalnem signalnem procesorju (DSP), kar bo omogočalo dovolj hitro obdelavo slike in uporabo v aplikaciji vodenje vozička. Lažje in zanesljivejše izločanje bomo skušali tudi zagotoviti z uporabo optičnega laserskega filtra, ki izboljša vidnost laserske črte.

Zahvala:

Za vzpodbude, mentorstvo in koristne nasvete se zahvaljujeva Simonu Klančnik, Karlu Benkič in Petru Planinšič.

6 Literatura

[1] Aleš Tetičkovič, Simon Klančnik, »Razpoznava govora z usmerjeno nevronsko mrežo«, Zbornik konference ERK' 05, Portorož, 26-28 sept., 2005

[2] Simon Klančnik, »Zaznavanje ovir s pomočjo strukturirane svetlobe « Zbornik konference ERK' 06, Portorož, 25-27 sept., 2006

[3] http://en.wikipedia.org/wiki/RGB_color_model

[4] http://hercules.uni-mb.si/articles/Znanstveni%5CIzvedba%20parametricnega%20nelinearnega%20filtra%20za%20iskanje%20ko%C5%BEnih%20znacnic%20v%20digitalni%20sliki%20z%20FPSLIC%20tehnologijo.pdf

[5] http://www.cescg.org/CESCG-2004/web/Sedlacek-Marian/

[6] Žarko Čučej, Dušan Gleich in Peter Planinšič Signali: Povzetki teorije z zbirko rešenih nalog, UM-FERI, 2005