

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 317

**DETEKCIJA TEKSTA U SLIKAMA
PRIBAVLJENIM IZ PERSPEKTIVE VOZAČA**

Matija Gulić

Zagreb, lipanj 2011.

Sadržaj

Uvod.....	1
1. Pregled korištenih tehnika.....	2
1.1. Binarizacija sive slike	2
1.2. Pronalaženje kontura.....	2
1.3. Pravokutnici koji opisuju konture.....	3
1.4. Stvaranje slike akumuliranih pravokutnika opisanih konturama.....	3
1.5. Eliminacija lažnih detekcija teksta.....	4
1.5.1. Eliminacija po veličini pravokutnika	4
1.5.2. Eliminacija po omjeru visine i širine pravokutnika	5
1.5.3. Eliminacija na temelju periodičnosti elemenata	5
1.6. Odabrani pristup detekcije teksta.....	5
2. Biblioteka OpenCV.....	7
3. Programska implementacija.....	9
3.1. Modul <code>main</code>	9
3.2. Modul <code>parametri</code>	9
3.3. Modul <code>predobradaSlike</code>	10
3.4. Modul <code>konture</code>	10
3.5. Modul <code>detekcijaTeksta</code>	11
3.6. Modul <code>histogram</code>	11
3.7. Modul <code>anotacija</code>	11
3.8. Modul <code>dohvacanjeSlike</code>	12
4. Eksperimentalni rezultati	13
4.1. Postavljanje potrebnih parametara.....	13
4.2. Ilustracija koraka obrade.....	13
4.3. Utjecaj pojedinih parametara na rezultate postupka	18

4.4. Procjena ispravnosti postupka.....	21
Zaključak	25
Literatura.....	26
Sažetak	27
Summary	28

Uvod

Računalni vid je područje usmjereno na ekstrakciju korisnih podataka iz slika ili videa. Neke od osnovnih tema koje istražuje računalni vid su detekcija, raspoznavanje i praćenje objekata. Detekcija teksta može se smjestiti unutar većeg područja detekcije objekata. Sam postupak detekcije teksta za čovjeka je vrlo jednostavan dok je za računalo vrlo složen.

Kako bi bilo moguće osmisliti i realizirati postupak za detekciju teksta potrebno je koristiti neke od biblioteka za rad sa slikama. OpenCV je javno dostupna biblioteka za rad sa slikama, uključuje znatan broj već postojećih algoritama i pomoćnih funkcija. Upravo zahvaljujući navedenom i izvršnoj dokumentaciji odlučeno je korištenje biblioteke OpenCV u ovom radu.

U ovom radu razmatra se postupak za detekciju i lokalizaciju teksta. Prvo provodimo binarizaciju po svim pragovima (od 0 do 255) kako bi dobili akumulirane vrijednosti za pravokutnike opisane detektiranim konturama. Nakon toga slijedi ponovna binarizacija akumulirane slike i eliminacija potencijalnih pravokutnika teksta na temelju informacija iz histograma. Informacije iz histograma omogućuju glasanje na temelju periodičnosti elemenata. Rezultati glasanja izravno eliminiraju potencijalne pravokutnike koji nisu tekst.

Rad je strukturiran kako slijedi. U 1. poglavlju opisane su korištene tehnike. Iduće poglavlje (2.poglavlje) opisuje osnove biblioteke OpenCV. Nakon toga u 3. poglavlju slijedi programska implementacija i u 4. poglavlju su eksperimentalni rezultati te procjena ispravnosti dobivenih rezultata.

1. Pregled korištenih tehnika

1.1. Binarizacija sive slike

Prije samog pronalaženja potencijalnih pravokutnika koji opisuju tekst potrebno je izvornu sliku pretvoriti u crno bijelu sliku. Postupak binarizacije [1] slike je jednostavan, ali ovisan o odabranom pragu binarizacije. U poglavlju 4.3 raspravljani su rezultati ovisno o spomenutom parametru.

Kako bi izbjegli ovisnost o pragu binarizacije može se koristiti Otsuova metoda binarizacije [2]. Ova metoda se koristi za automatsko određivanje optimalnog praga binarizacije. Osnova na kojoj se temelji Otsuova metoda je korištenje histograma uz pretpostavku da slika sadrži dvije kategorije piksela (tekst i pozadina).

1.2. Pronalaženje kontura

Konture opisuju rubove objekata na binarnim slikama. Postoji mnogo algoritama koji se koriste za pronalaženje kontura. U ovom radu koristi se postupak djelomično baziran na Suzukijevom algoritmu [3]. Na slijedećim slikama (Slika 1, Slika 2) može se vidjeti rezultat pronalaženja kontura.



Slika 1. Slika za detekciju kontura



Slika 2. Invertirana slika detektiranih kontura

1.3. Pravokutnici koji opisuju konture

Nakon što su pronađene konture potrebno je na neki način opisati njihovu lokaciju na slici. Jedan od načina je prikaz položaja konture opisanim pravokutnikom, odnosno minimalnim pravokutnikom koji sadrži detektiranu konturu. Pogledajmo na sljedećem primjeru (Slika 3) kako se koriste navedeni pravokutnici.



Slika 3. Pravokutnici koju opisuju detektirane konture

Neke od alternativnih metoda za označavanje lokacije konture su minimalni poligoni koji prate orijentaciju konture i nakošeni pravokutnici. No u ovom konkretnom slučaju ne postoji potreba za kompliciranijim metodama.

1.4. Stvaranje slike akumuliranih pravokutnika opisanih konturama

Akumulirana slika pravokutnika opisanih konturama jedna je od originalnih ideja nastalih u ovom radu. Kao što je spomenuto u poglavlju 1.1 postoji problem određivanja praga binarizacije i upravo se na ovaj način rješava navedeni problem. Postupak djelomično sličan navedenom koristi se u algoritmu MSER [4].

Postupak stvaranja akumulirane slike sastoji se od binarizacije ulazne sive slike po svim pragovima binarizacije (vrijednosti od 0 do 255). Za svaku iteraciju dobiva se crno bijela slika, na toj slici provodi se postupak pronalaženja kontura (i opisanih minimalnih pravokutnika). Na mjestu svakog piksela unutar pravokutnika, u akumuliranoj slici se vrijednost piksela poveća za jedan. Ako se neki pravokutnik pojavljuje prilikom više različitih pragova binarizacije to će vrijednost piksela unutar pravokutnika u akumuliranoj slici biti veća.

Na stvarnim slikama, upravo zahvaljujući pravokutnicima, dolazi do spajanja područja oko pojedinih znakova u riječi, odnosno tekst. Navedena pojava je osnova za daljnje odbacivanje područja koja nisu tekst.



Slika 4. Slika prije postupka akumulacije



Slika 5. Primjer invertirane akumulirane slike

1.5. Eliminacija lažnih detekcija teksta

Nakon što smo detektirali potencijalne pravokutnike s tekstom potrebno je na neki način riješiti problem lažne detekcije. Za rješavanje ovog problema koriste se mnogobrojne metode [5], no često su ovisne o pretpostavkama usko vezanim za specifično područje istraživanja. No ipak mogu poslužiti kao referenca prilikom istraživanja prvotne detekcije. U ovom radu razvijen je specifičan postupak eliminacije lažnih pravokutnika koji opisuju tekst. Navedeni postupak opisan je u nastavku ovog poglavlja (1.5.3).

1.5.1. Eliminacija po veličini pravokutnika

Eliminacija po veličini pravokutnika je jedan od često upotrebljavanih pristupa eliminacije lažnih detekcija. Loša strana ovog postupka je što konačan postupak detekcije neće biti neovisan o veličini teksta.

Koristan postupak za brzo i efikasno rješavanje problema pravokutnika koji veličinom prekrivaju pola izvorne slike. Ovaj postupak koristi se prilikom generiranja akumulirane slike.

1.5.2. Eliminacija po omjeru visine i širine pravokutnika

Jedna od pretpostavki da bi neki pravokutnik bio tekst je omjer dužine i širine mora biti veći od neke granične vrijednosti (primjerice 1). Ovaj postupak je robustan, ali na konkretnim slikama koje su analizirane u ovom radu nije u potpunosti primjenjiv (prvenstveno zbog problema parametra binarizacije akumulirane slike, opisano u 4.3).

1.5.3. Eliminacija na temelju periodičnosti elemenata

Ovaj postupak nije toliko često spominjan u literaturi, a stoga ni korišten. No u konkretnom slučaju pokazao se izvrstan. Uglavnom ne eliminira ispravne detekcije, a eliminira preko 95% krivo detektiranih pravokutnika.

Ideja ovog postupka je provjeriti da li dva pravokutnika (u kojima su smještene konture) imaju približno jednaku visinu (uz prethodno definiranu toleranciju). Nakon toga provjerava se da li su ta dva pravokutnika međusobno susjedna, odnosno jedan pokraj drugog na horizontalnoj osi.



Slika 6. Potencijalni pravokutnici teksta (lažna detekcija)



Slika 7. Potencijalni pravokutnici teksta (ispravna detekcija)

Na slikama iznad (Slika 6, Slika 7) može se vidjeti da je pretpostavka na kojoj počiva navedeni postupak potpuno ispravna i pravokutnici koji nisu tekst neće imati periodičkih elemenata.

1.6. Odabrani pristup detekcije teksta

Pseudokod:

```

za pb=1 do 255
    binariziraj originalnu ulaznu sliku s pragom binarizacije pb
    pronadi sve konture
    označi konture minimalnim pravokutnicima
    dodaj vrijednost 1 za svaki piksel unutar pravokutnika u početno
    praznu akumuliranu sliku
kraj

binariziraj akumuliranu sliku (Otsuova metoda binarizacije)
pronadi sve konture
označi konture minimalnim pravokutnicima (skup pravokutnika sp)
za svaki pravokutnik p u skupu dobivenih pravokutnika sp
    odredi histogram pravokutnika p na akumuliranoj slici
    pronadi lokalne minimume na dobivenom histogramu
    za svaki lokalni minimum lm
        binariziraj pravokutnik p (prag binarizacije lm)
        ako (postoji periodičnost unutar pravokutnika p)
            tekst++
        inače
            netekst++
    kraj
    ako (tekst / netekst >= 4)
        pravokutnik p je tekst, ostavi u listi pravokutnika sp
    inače
        pravokutnik p nije tekst, izbacij iz liste pravokutnika sp
kraj

za svaki pravokutnik p u skupu pravokutnika sp
    anotiraj pravokutnik p na originalnoj ulaznoj slici
kraj

```

2. Biblioteka OpenCV

Biblioteka OpenCV [6], [7], [8] može se podijeliti na 4 osnovna područja: funkcionalnost jezgre (cxcore), procesiranje slika i računalni vid (cv), korisničko sučelje i čitanje/zapis slika (highgui) i strojno učenje (ml).

Funkcije jezgre sadrže osnovne strukture, operacije nad nizovima, funkcije za iscertavanje, XML/YAML perzistenciju.

Procesiranje slika i računalni vid sastoji se od filtriranja slika, transformacija nad slikama, histograma, detekcija značajki, analiza pokreta i detekcija objekata, kalibracije kamere i 3D rekonstrukcija.

Highgui se sastoji od korisničkog sučelja, čitanja i pisanja slika i videa, odnosno stvaranje videa iz niza slika.

Open Computer Vision library je open source biblioteka sa preko nekoliko stotina algoritama vezanih za računalni vid. Biblioteka je multiplatformna, odnosno moguće je koristiti na nekoliko najpopularnijih platformi (Windows, Linux, Mac).

Ova dobro dokumentirana biblioteka je jedan od najčešće korištenih alata u razvoju novih algoritama računalnog vida. Jedan od razloga je dostupnost sučelja za brojne programske jezike i dovoljno snažna zajednica koja usavršava ovu biblioteku.

Korištene OpenCV funkcije u ovom radu: cvRectangle (anotacija), cvCopy (kopiranje slika), cvZero (pražnjenje slike), cvGet2D (dohvaćanje vrijednosti pojedinog piksela), cvSet2D (postavljanje vrijednosti pojedinog piksela), cvSetImageROI (postavljanje regije interesa za pojedini dio slike), cvThreshold (binarizacije slike), cvDestroyWindow (oslobađanje memorije koju je zauzeo prozor za prikazivanje slike), cvResetImageROI (postavljanje regije interesa na cijelu sliku), cvReleaseImage (oslobađanje memorije koju je zauzela slika), cvFindContours (pronalaženje kontura na crno bijeloj slici), cvBoundingRect (pronalaženje minimalnog pravokutnika oko konture), cvReleaseMemStorage (oslobađanje memorije kontura), cvCreateFileCapture (dohvaćanje videa), cvLoadImage (dohvaćanje slike), cvGetCaptureProperty (dohvaćanje okvira po sekundi u videu), cvQueryFrame (dohvaćanje slike iz videa), cvCreateImage (stvaranje slike), cvNamedWindow (stvaranje prozora za prikaz slike), cvShowImage (prikazivanje

slike na ekranu), cvWaitKey (dohvaćanje pritisnute tipke), cvSmooth (korištenje Gauss zamućenja), cvConvertImage (pretvaranje iz slike u boji u sivu sliku), cvEqualizeHist (izjednačavanje sive slike po histogramu), cvCanny (detekcija rubova Cannyevim postupkom).

3. Programska implementacija

Eksperimentalni dio diplomskog rada izrađen je u programskom jeziku C++, uz korištenje biblioteke OpenCV.

Sustav učitava ulaznu sliku, primjenjuje odabrani postupak i prikazuje rezultate obrade. Podijeljen je na nekoliko modula: `main`, `parametri`, `predobradaSlike`, `konture`, `detekcijaTeksta`, `histogram`, `anotacija` i `dohvacanjeSlike`.

3.1. Modul `main`

Glavna komponenta sustava, dohvaća nazive slika iz modula `dohvacanjeSlike` i prosljeđuje tako dobivene slike do modula `detekcijaTeksta`. Također ova komponenta se koristi za interakciju s korisnikom, odnosno ona je sučelje prema korisniku.

3.2. Modul `parametri`

U ovom modulu smješteni su parametri koji definiraju ponašanje sustava, odnosno koje metode će se koristiti ili koje slike će se prikazivati.

Primjer dijela parametara:

```
char* imageDirPath = "E:\\ljuska_cv\\slike_video\\diplomski_rad\\znakovi\\baza001";
char* resultDirPath =
"E:\\ljuska_cv\\slike_video\\diplomski_rad\\znakovi\\baza001\\rezultati";

// korištenje Gauss blur na izvornoj slici
int gauss = 0;

// korištenje histogram equalization: 1 - da, 0 - ne
int histogramequalization = 0;

// prikaži crno bijelu sliku: 1 - da, 0 - ne
int prikazcrnobijela = 1;

// prikažu crno bijelu sliku s detektiranim boxovima
int prikazprijesusjednih = 0;
```

3.3. Modul predobradaSlike

Predobrada slike sastoji se od funkcija koje se primjenjuju na sliku prije samog početka detekcije teksta. Kako bi mogli primijeniti željeni postupak potrebno je dobiti crno bijelu sliku. Prije same binarizacije ulaznu sliku moguće je obraditi nekim od filtara dostupnih u biblioteci OpenCV. Konkretno u ovom modulu implementirane su slijedeće funkcije:

```
void GaussBlur(IplImage* ulazSlika, IplImage* izlazSlika);  
void PretvoriUSivuSliku(IplImage* ulazSlika, IplImage* izlazSlika);  
void HistogramEqualize(IplImage* ulazSlika, IplImage* izlazSlika);  
void Binariziraj(IplImage* ulazSlika, IplImage* izlazSlika, double prag);  
void BinarizirajOtsu(IplImage* ulazSlika, IplImage* izlazSlika, double prag);  
void CannyEdgeDetection(IplImage* ulazSlika, IplImage* izlazSlika, double lowThresh,  
double highThresh, int aperture);
```

Prilikom istraživanja koji postupak bi davao najbolje rezultate testirane su funkcije Gaussovog glaćenja [9], izjednačavanje po histogramu [10] i Cannyeva detekcija rubova [11]. Omjer dobiveno (kvaliteta rezultata) i uloženo (vrijeme, resursi) nije zadovoljavajući za navedene funkcije i stoga je odlučeno da se one neće koristiti u ovoj verziji postupka.

3.4. Modul konture

Modul konture sastoji se od funkcija za detekciju kontura i eliminaciju kontura.

```
void PronadiKonture(IplImage* ulazSlika, std::vector<CvRect>& izlazBoxes);  
void PronadiKonture(IplImage* ulazSlika, IplImage* izlazKonture);  
void EliminirajKonturePoVelicini(std::vector<CvRect>& boxes, int minWidth, int  
maxWidth, int minHeight, int maxHeight);  
void EliminirajKonturePoHeightWidthRatio(std::vector<CvRect>& boxes, double  
widthHeightRatio);  
void EliminirajKonturePoWidthHeightRatio(std::vector<CvRect>& boxes, double  
widthHeightRatio);  
void EliminirajKonturePomocuHistograma(IplImage* ulazSlika, std::vector<CvRect>&  
boxes);
```

Nakon početnog detektiranja kontura javlja se izrazito velik broj kontura koje nisu tekst i potrebno je takve konture ukloniti iz dobivenih rezultata. U ovom slučaju funkcija EliminirajKonturePomocuHistograma se pokazala odlična. Ideja ove funkcije je na temelju podataka iz histograma [12] odrediti prag binarizacije optimalan za detektiranu konturu, odnosno pravokutnik u koji je upisana. Zatim na tako dobivenoj crno bijeloj slici

provjeriti periodičnost elemenata. Ova metoda je iznimno robusna i kao takva može se koristiti u većini postupaka za detekciju teksta.

3.5. Modul **detekcijaTeksta**

DetekcijaTeksta je glavni modul sustava. Sastoji se od samo jedne funkcije Detektiraj koja slijedno provodi odabrani postupak. Prvo ulaznu sliku pretvara u sivu sliku. Zatim tako dobivenu sivu sliku binarizira za sve vrijednosti pragova. Na svakoj crno bijeloj slici detektiraju se konture. Za svaku konturu opisuje se pravokutnik koji ima utjecaj na vrijednost piksela u akumuliranoj slici. Ideja akumulirane slike temelji se na pretpostavci da je tekst kontrastniji u odnosu na ostale pojave na slici. Zbog toga što je kontrastniji imat će veću vrijednost na akumuliranoj slici. Na kraju funkcije slijedi eliminacija lažnih pravokutnika (kontura) i anotacija detektiranih područja teksta.

3.6. Modul **histogram**

Ovaj modul je okosnica eliminacije velikog broja lažnih detekcija teksta. Za svako područje koje je potencijalni tekst kreira se histogram vrijednosti piksela sive slike. Zatim se traže lokalni minimumi u tako dobivenom histogramu. Svaki lokalni minimum postaje prag binarizacije za akumuliranu sliku. Na takvoj slici dobivaju se konture pojedinačnih slova ili nekih drugih pojava na slici. Zatim se na temelju periodičnosti određuje da li je pravokutnik tekst ili ne. Definirane su dvije funkcije:

```
bool OdrediHistogram(IplImage* ulazSlika, CvRect ulazBox);
```

```
bool ProvjeriPeriodicnost(std::vector<CvRect>& boxes);
```

3.7. Modul **anotacija**

Anotacija detektiranih područja slike je krajnji korak u svakom postupku detekcije teksta. Zadaća ovog jednostavnog modula je na ulaznoj slici prikazati koja područja su proglašena tekstom. Funkcija koja sadrži tu funkcionalnost je AnotirajPravokutnike.

3.8. **Modul dohvatanjeSlika**

Modul dohvatanjeSlika koristi se prilikom pokretanja sustava. Jedina zadaća ovog modula je dohvatiti sva imena slika koje se nalaze u direktoriju definiranom u modulu parametri. Koristi se funkcija DohvatiImenaSlika.

4. Eksperimentalni rezultati

Kako bi ocijenili valjanost i značaj algoritma potrebno je iscrpno testiranje. Automatsko testiranje nije omogućeno i stoga je potrebno za svaku sliku gledati rezultate i objektivno zabilježiti dobivene rezultate.

4.1. Postavljanje potrebnih parametara

Za ispravan rad sustava potrebno je prije pokretanja postaviti određene parametre. Parametri se mogu postaviti u modulu `parametri`.

Neki od potrebnih parametara su određivanje rada sa slikom ili videom, put do direktorija koji sadrži slike ili video, da li želimo koristiti Gaussovo gladenje prije rada s ulaznom slikom, koju vrstu binarizacije želimo koristiti, koje postupke eliminacije potencijalnih pravokutnika s tekstom želimo koristiti i koje slike želimo prikazivati prilikom detekcije teksta (sive slike, crno-bijele slike, prikaz histograma, akumulacijsku sliku).

Važno je napomenuti da se određeni parametri mogu dinamički mijenjati prilikom izvršavanja programa, konkretno svi parametri vezani uz prag binarizacije i korištenje željenih metoda eliminacije.

Modul `parametri` je dobro dokumentiran (komentari) i stoga je lako definirati novo ponašanje sustava.

4.2. Ilustracija koraka obrade

Prvo se prikazuje ulazna slika (Slika 8), kako bismo lakše pratili rad algoritma.



Slika 8. Ulazna slika

Iduća slika (Slika 9) dobiva se akumuliranjem detektiranih pravokutnika za sve pragove binarizacije (ukupno 255 pragova).



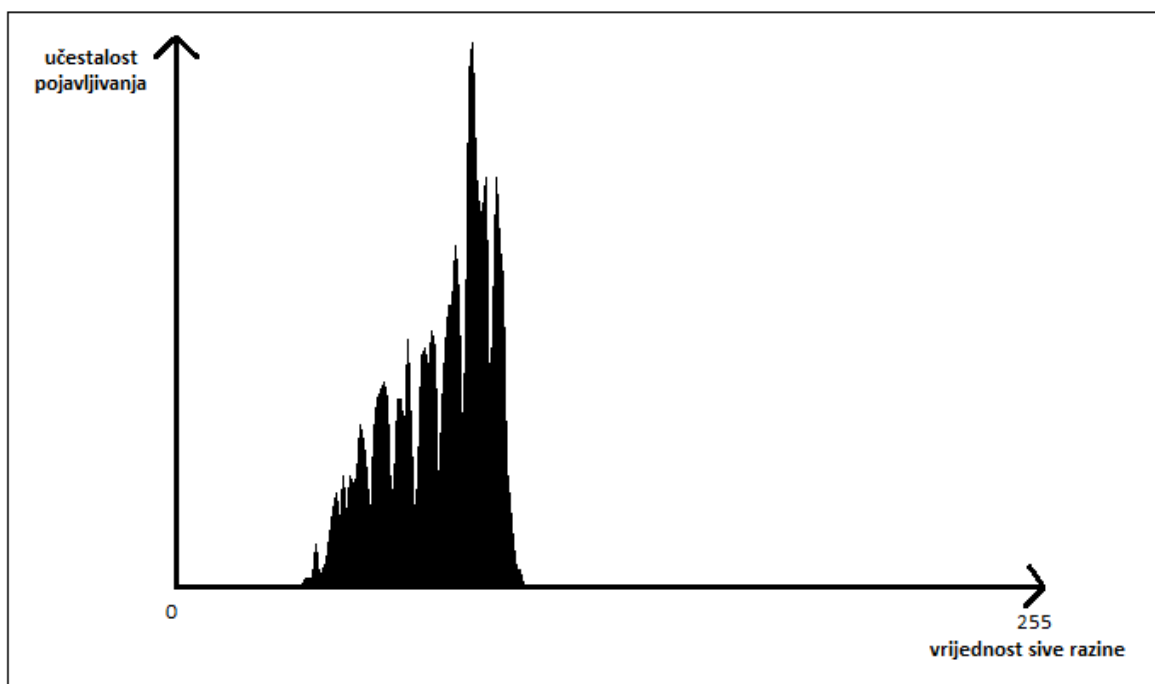
Slika 9. Slika akumuliranih pravokutnika opisanih konturama

Zatim se ovako dobivena akumulirana slika pretvara u crno bijelu sliku. Uz Otsuovu metodu binarizacije i postupak detekcije potencijalnih pravokutnika teksta anotiraju se dobiveni pravokutnici na početnoj slici (Slika 10).



Slika 10. Anotirani potencijalni pravokutnici teksta

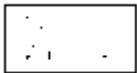
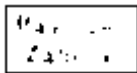
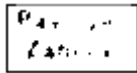
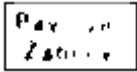
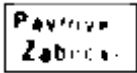
Iduća slika (Slika 11) nastaje fokusiranjem na pojedine potencijalne pravokutnike teksta. Za svaki pravokutnik kreira se siva slika i radi histogram.



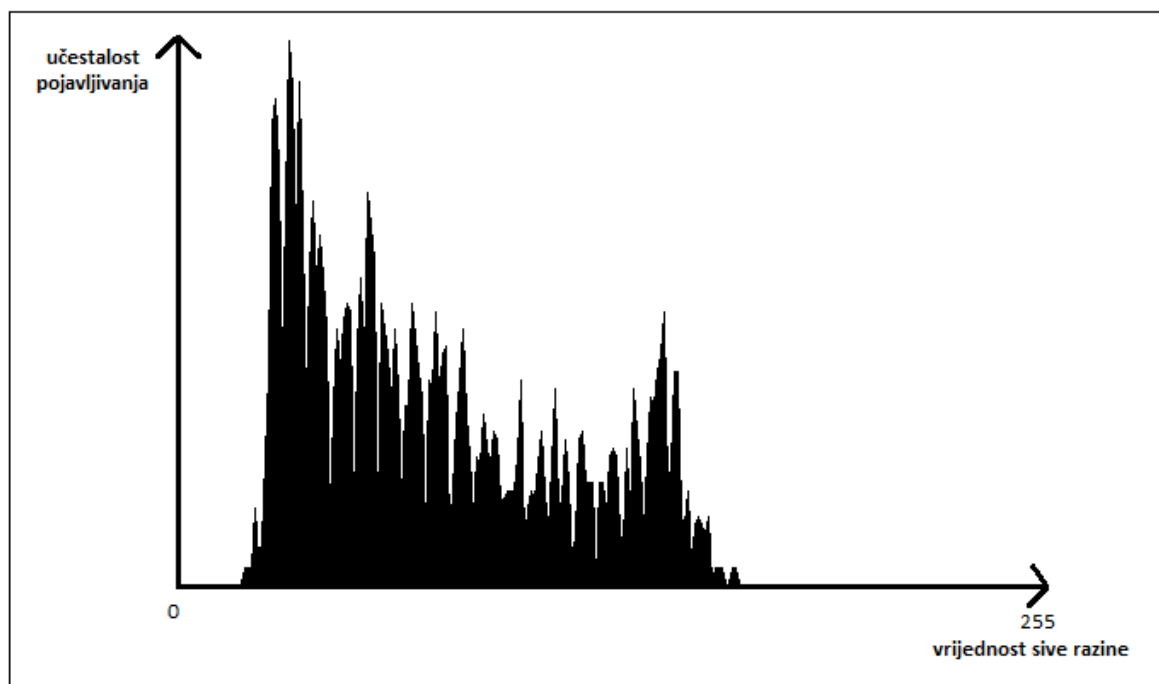
Slika 11. Histogram potencijalnog pravokutnika teksta (označen plavo na slici 10)

Na temelju dobivenog histograma traže se svi lokalni minimumi koje se nalaze lijevo od globalnog maksimuma. Pretpostavka je da će u pravokutniku koji sadrži tekst pozadina biti dominantna i zato pokušavamo naći prag binarizacije koji će eliminirati pozadinu. Za svaki od pronađenih minimuma provodi se binarizacija potencijalnog pravokutnika teksta i testira se da li postoje periodični elementi unutar pravokutnika. Za histogram na slici 11 dobivaju se slijedeći pragovi binarizacije: 43, 49, 51, 53, 58, 64, 68, 71, 75, 78 i 85. Pogledajmo u tabeli 1 kako izgleda binarizacija po dobivenim vrijednostima pragova binarizacije.

Tabela 1. Rezultati binarizacije za lokalne minimume za ispravnu detekciju

Prag binarizacije	Binarna slika	Anotirani pravokutnici	Prag binarizacije	Binarna slika	Anotirani pravokutnici
43			68		
49			71		
51			75		
53			78		
58			85		
64					

Pokažimo na primjeru zeleno označenog pravokutnika sa slike 10 kako izgleda histogram za pravokutnik koji ne sadrži tekst (Slika 12).



Slika 12. Histogram pravokutnika koji ne sadrži tekst (označeno zeleno na slici 10)

Iz navedenog histograma može se vidjeti da će pragovi binarizacije imati vrijednosti 22 i 29. Pogledajmo u tabeli 2 kako izgleda binarizacija za lažnu detekciju.

Tabela 2. Rezultati binarizacije za lokalne minimume lažne detekcije

Prag binarizacije	Binarna slika	Anotirani pravokutnici	Prag binarizacije	Binarna slika	Anotirani pravokutnici
22			29		

Završni korak je glasanje da li je pravokutnik tekst. Jedini kriterij u odlučivanje da li je pravokutnik tekst je provjera periodičnosti elemenata. Ukoliko prevladaju pozitivni glasovi pravokutnik se proglašava tekstom. Na idućoj slici (Slika 13) prikazano je koji pravokutnici su proglašeni tekstom.

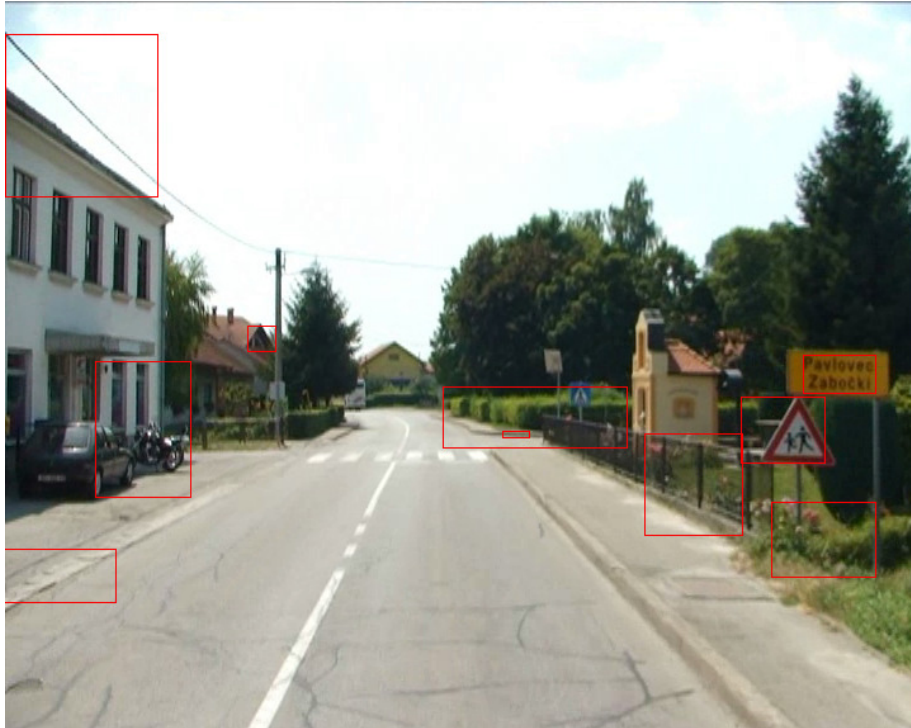


Slika 13. Slika s označenim pravokutnicima teksta

4.3. Utjecaj pojedinih parametara na rezultate postupka

Na rad sustava znatno utječe prag binarizacije akumulirane slike. Ovo je jedini parametar koji djelomično sprječava robusnost samog algoritma.

Pogledajmo na nekoliko primjera (Slika 14, Slika 15, Slika 16) kako utjecaj odabira praga binarizacije utječe na detekciju teksta. Ukoliko je prag premali dolazi do spajanja kontura, a ukoliko je prag prevelik ni jedna kontura neće proći binarizaciju.



Slika 14. Detektiran tekst uz prag binarizacije 20

Odabirom praga binarizacije 20 dolazi do spajanje više slova teksta u jednu veću konturu i rezultat je detekcija dvije odvojene riječi kao jedna cjelina. Također jedna od posljedica premalog praga binarizacije je i povećan broj krivo detektiranih pravokutnika teksta.



Slika 15. Detektiran tekst uz prag binarizacije 30

Odabirom praga binarizacije 30 dobili smo skoro savršenu detekciju teksta. Broj krivo detektiranih pravokutnika sveden je na svega 3 pravokutnika.



Slika 16. Detektiran tekst uz prag binarizacije 40

Povećanjem praga binarizacije na 40 nemamo krivo detektiranih pravokutnika, ali puno značajnije nemamo ni točno detektirani pravokutnik.

Jedno od rješenja odabira dobrog praga binarizacije je korištenje Otsuove [2] metode binarizacije. Zahvaljujući ovom postupku sustav za detekciju teksta je posve neovisan o svim parametrima. Ovom metodom dobiva se točna detekcija, ali i nekoliko lažnih detekcija kao što se može vidjeti na sljedećoj slici (Slika 17).



Slika 17. Detektiran tekst uz Otsuovu binarizaciju

4.4. Procjena ispravnosti postupka

Postupak možemo podijeliti na dva dijela: stvaranje slike akumuliranih pravokutnika i eliminacija lažnih detekcija. Postupak je evaluiran na 52 slike.

Rezultati preciznosti nakon stvaranje akumulirane slike su manji od 0.6%, dok su rezultati odziva 76.5%. Ako primijenimo Gaussovo glađenje, uz vrijednost parametra glađenja 0.5, rezultati odziva padaju na 69%, dok se preciznost povećava na 0.8%.

Pogledajmo rezultate nakon eliminacije lažnih detekcija. Ispitne slike moguće je podijeliti u nekoliko grupa. Za prvu grupu karakterističan je veći tamni tekst na svijetloj pozadini. Ilustracija prve grupe može se vidjeti na slici 18. Za ovu grupu rezultati detekcije su u gotovo svim ispitnim slikama bili jako dobri. Odziv je 50%, a preciznost 37.5%.



Slika 18. Tipičan predstavnik prve grupe slika ispitnog slijeda

Drugu grupu karakteriziraju bijela slova na svijetlo plavoj pozadini. Primjer druge grupe može se vidjeti na slici 19. U ovom slučaju nije dolazilo do točne detekcije. Postupak za pronalaženje teksta se bazira na kontrastu između slova i pozadine, a u ovom slučaju taj uvjet nije ispunjen i zbog toga nemamo rezultate detekcije. Ovi slučajevi neće biti razmatrani u daljnjoj procjeni ispravnosti postupka.



Slika 19. Tipičan predstavnik druge grupe slika ispitnog slijeda

Treća skupina (Slika 20) može se opisati kao veći putokazi koje se uglavnom tretira kao spojeni tekst. Uz dobar prag binarizacije tekst je gotovo uvijek točno detektiran. I u ovom slučaju javlja se nešto veći broj lažnih detekcija. Odziv za ovu grupu je 37.5%, preciznost je 30%.



Slika 20. Tipičan predstavnik treće grupe slika ispitnog slijeda

Četvrtu grupu karakteriziraju svijetla slova na tamnoj pozadini. Slika 21 je tipičan predstavnik četvrte grupe. Postupak eliminacije na temelju histograma nema rješenje za ove slike i gotovo uvijek eliminira točne detekcije. Razlog eliminacije točne detekcije je taj što u histogramu očekujemo da će tamna pozadina biti dominantna. U ovom slučaju pragovi binarizacije će biti pogrešno izabrani. Moguće rješenje ovog problema je prilikom odabira minimuma iz histograma odrediti dominantnu boju pozadine (svijetla ili tamna), i na temelju tog podatka dohvaćati lokalne minimume lijevo ili desno od globalnog maksimuma.



Slika 21. Tipičan predstavnik četvrte grupe slika ispitnog slijeda

Petu grupu možemo definirati kao manja tamna slova na svijetloj pozadini. Slično kao prva grupa, uz glavnu razliku u veličini teksta. Odziv za ovu grupu je 14.1%, a preciznost 34.7%. Peta grupa ilustrirana je na slici 22.



Slika 22. Tipičan predstavnik pete grupe slika ispitnog slijeda

Ukupni rezultati odziva za prvu, treću i petu grupu je 28.7% i preciznost 35.8%.

Zaključak

Predloženi postupak za detekciju teksta temelji se na pronalaženju kontura, stvaranju slike akumuliranih pravokutnika opisanih konturama i eliminaciji lažnih detekcija. Ulaznu sliku prvo pretvorimo u sivu sliku. Zatim na takvoj slici slijedi postupak dobivanja akumulirane slike (binarizacija po pragovima od 0 do 255 i bilježenje dobivenih rezultata). Nakon toga za svaki potencijalni pravokutnik teksta primjenjujemo postupak eliminacije temeljen na vrijednostima histograma. Navedeni postupak svodi se na računanje histograma i određivanje novih pragova binarizacije za akumuliranu sliku, nakon čega slijedi glasanje kako bismo odredili da li je pravokutnik zaista tekst. Opisani postupak za detekciju teksta uvodi nove metode (ideje) u ovo područje i dobivaju se ohrabrujući rezultati. Većina postojećih postupaka za detekciju teksta bazira se na pojedinačnim znakovima (slovima) i zatim se kreće u stvaranje riječi iz dobivenih slova. Postupak korišten u ovom radu ima suprotni smjer, prvo se detektiraju potencijalne riječi i zatim provjerava da li te riječi sadrže slova.

Opisani postupak za detekciju teksta u većini slučajeva na ispitnim slikama daje dobre rezultate. Važno je napomenuti da je prava detekcija gotovo uvijek prisutna, no ovisno o slici javlja se do stotinjak lažnih detekcija. Uz određene prilagodbe algoritma eliminacije specifičnoj slici dolazi do znatnog smanjenja lažnih detekcija.

Najbolji rezultati dobivaju se kada postoji znatan kontrast između slova i pozadine. Jedan od problema su bijela slova na svijetlo plavoj pozadini. Također potencijalni problemi javljaju se kada je kvaliteta slike loša, ili slova prilično mala.

Razvijeni postupak bi mogao poslužiti kao temelj za daljnji rad na području detekcije teksta. Neki od postupaka koje bi se moglo istražiti su: predobrada slike pomoću lokalnih pragova binarizacije, nove metode eliminacije lažnih rezultata temeljene na algoritmima strojnog učenja.

Literatura

- [1] JURIBAŠIĆ, M. *Detekcija registarskih oznaka na vozilima*, Završni rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, (2008), 4-9
- [2] OTSU, N. *A threshold selection method from gray-level histograms*, IEEE Trans.Sys., Man., Cyber. 9, 1, (1979), 62-66
- [3] SUZUKI, S. I ABE, K. *Topological structural analysis of digital binary image by border following*, Computer Vision, Graphics and Image processing, 30, 1, (1985), 32-46.
- [4] MATAS, J., CHUM, O., URBA, M. I PAJDLA, T. *Robust wide baseline stereo from maximally stable extremal regions*, Proc. of British Machine Vision Conference, (2002), 384-396
- [5] JUNG, K., KIM, K.I. I JAIN, A.K. *Text information extraction in images and videos: A Survey*, Elsevier Science B.V., Pattern Recognition, 37, 5, (2004), 977-997
- [6] IZVORNI KOD OPENCV BIBLIOTEKE:
<http://sourceforge.net/projects/opencvlibrary/>
- [7] WIKI STRANICE OPENCV PROJEKTA:
<http://opencv.willowgarage.com/wiki/>
- [8] BRADSKI, G. I KAEHLER, A. *Learning OpenCV*, O'Reilly Media, Sebastapol, (2008)
- [9] GAUSSIAN FILTERING, COMPUTER VISION DEMONSTRATION WEBSITE, ELECTRONICS AND COMPUTER SCIENCE, UNIVERSITY OF SOUTHAMPTON (12.04.2011.):
http://users.ecs.soton.ac.uk/msn/book/new_demo/gaussian/
- [10] PAPIĆ, V. *Obrada slike i računalni vid*, 2005, 49-50
- [11] DOSTAL, D. ET AL, *Cannyev detektor rubova*, tehnička dokumentacija predmeta Projekt, Fakultet elektrotehnike i računarstva, Zagreb, 2009
- [12] HISTOGRAM, WIKIPEDIA (12.05.2011.):
<http://en.wikipedia.org/wiki/Histogram>

Sažetak

Detekcija teksta u slikama pribavljenim iz perspektive vozača

U ovom radu bavimo se detekcijom teksta u slikama pribavljenim iz vozila u pokretu. Postupak koji koristimo temelji se na pronalaženju kontura, stvaranju akumulirane slike pravokutnika opisanih konturama i eliminaciji lažnih detekcija. Postupak je izveden u programskom jeziku C++, korištenjem biblioteke Open Computer Vision. Razvijena implementacija je evoluirana na stvarnim slikama, a dobiveni rezultati su prikazani i komentirani.

Ključne riječi: računalni vid, detekcija teksta, lokalizacija teksta, akumulirana slika, pronalaženje kontura, eliminacija histogramom, openCV, obrada slike

Summary

Text detection in images recorded from vehicle in motion

In this work we consider text detection in images recorded from vehicle in motion. Algorithm which we use is based on contour finding, creating contour based accumulated image and false positive elimination. This algorithm is implemented in programming language C++, using Open Computer Vision library. Developed component was evolved in real images, and results are provided and discussed.

Keywords: computer vision, text detection, text localization, accumulated image, contour finding, histogram based elimination, openCV, image processing