

Segmentacija dužice oka

Jasmina Smailović i Milan Todorović

2009



Predmet: Digitalna obrada slike

**Elektrotehnički fakultet
Univerzitet u Banjaluci**

**Profesor: dr Zdenka Babić
Asistent: mr Vladimir Risojević**

Sadržaj

Definicija problema.....	3
Uvod u biometrijsko prepoznavanje.....	3
Objašnjenje izbora teorije i tehnologije za rješavanje problema	4
Daugmanov algoritam.....	5
Optimizovani Dougmanov algoritam	6
Detalji rješenja na nivou implementacije	7
Thresh.m	7
Partiald.m.....	12
Lineint.m	12
Search.m	12
Drawcircle.m	12
Tresh1.m	13
Eyelid.m.....	13
traceLine.m	14
Diskusija i prijedlozi daljeg poboljšanja	15
Literatura.....	16

Definicija problema

Naš zadatak u okviru semestralnog rada iz predmeta **Digitalna obrada slike** je bio da izvršimo segmentaciju dužice¹ oka sa slike oka. Ovaj postupak predstavlja prvi korak za biometrijsko prepoznavanje na osnovu slike oka.

Uvod u biometrijsko prepoznavanje

Pitanje svih mogućih aspekata razlikovanja, prepoznavanja i identifikacije javlja se još u dalekoj istoriji. U plemenskom sistemu prvobitne zajednice, neposlušne članove plemena koji su se ogriješili o plemenska pravila uz progon kao mjeru s vrlo izvjesnom smrću (nije se moglo preživjeti sam, bez vatre, hrane i lovačke opreme), određivalo se i označavanje istih sakaćenjem, ožiljcima ili žigosanjem, kako bi svi znali da se radi o prognaniku, pa ga obično, nakon što je prepoznat kao takav, nije prihvaćalo niti drugo pleme. Poznati su slični načini označavanja ubojica, lopova i nemoralnih u srednjovjekovnoj Evropi, ali i širom svijeta.

Pojedine metode identifikacije koristile su se i u humanije svrhe, pa je poznato da su prije četiri hiljade godina u pisanim dokumentima Asiraca i Babilonaca kao znak pisca i dokaz autorstva na dokument otiskivani otisci papilarnih linija prsta (tzv. Supur), a još stari Kinezi označavali su novorođenčad kako bi izbjegli zamjenu djece.

U novom vijeku, a posebno razvojem humanističkih i prirodnih nauka došlo je do procvata brojnih metoda koje su trebale pridonijeti identifikaciji, prepoznavanju i razlikovanju određenih osoba. Prije svega radi se o korištenju medicinskih nauka u području kriminalističke identifikacije osoba i o traseološkoj identifikaciji.

Neke od klasičnih identifikacijskih metoda, koje su prema opštim karakteristikama biometrijske, dobivaju potpuno novo značenje i kvalitetu i doživljavaju svoj *remake* u informatičko-digitalnom okruženju. Naime, primjenom modernih tehnologija iz domene elektrotehnike i računarstva, kroz savremene hardverske uređaje i softverske alate, neke svojevremeno kroz istoriju odbačene identifikacijske metode, doživljavaju svoj procvat i novu afirmaciju. To je posebno značajno u području antropološke identifikacije koja je zbog tehnoloških ograničenja tokom 20. vijeka gotovo napuštena, osim u zoni sudsko-medicinskih ekspertiza. Također, savremena

¹ Dužica = šarenica. U tekstu se koriste oba pojma.

tehnološka i naučna dostignuća omogućuju nastanak i razvitak potpuno novih metoda identifikacije, koje se temelje na tjelesnim i ponašajnim individualnim karakteristikama osoba koje prije nije bilo moguće registrovati, izdvojiti i koristiti kao i na tehnološkim, prije svega elektroničkim dostignućima za apliciranje istih. Savremena tehnologija omogućava pouzdana, brza i nenametljiva sredstva za automatsko prepoznavanje identitet osoba. Mnogo je primjera potrebe za visokim stepenom sigurnosti u našim životima, kao što je kontrola pasoša, licence, upotreba ključeva i kartica, sigurnost na aerodromima i slično. Neke od metoda za biometrijsku identifikaciju su geometrija ruke, otisak prsta i dinamika potpisa. Za ove vrste identifikacije problem se javlja u jedinstvenosti, kao i u imunosti na vrijeme. Ideja da se šarenica oka koristi kao optički otisak prsta za identifikaciju osobe je prvi put predložena od strane oftalmologa koji su iz iskustva primijetili da svaka šarenica ima jedinstvenu teksturu i visok nivo detalja koji ostaju nepromijenjeni decenijama. Sljedeće karakteristike šarenice naglašavaju njenu pogodnost za korištenje u automatskoj identifikaciji:

- Inherentna izolacija i zaštita od okoline jer je interni organ oka, iza rožnjače i vodene tečnosti
- Nemogućnost promjene operacijom bez rizika da se ošteti vid
- Fiziološka osjetljivost na svjetlost koja omogućava prirodni test protiv varke (postavljanje lažnog oka umjesto pravog). Osobina koju šarenica dijeli sa otiskom prsta je slučajna struktura detalja. Tekstura šarenice je po sebi stohastička i moguće haotična jer genetika nema uticaja na ovaj organ osim u anatomske forme, fiziologiji, boji i uopštenoj pojavi. S obzirom da detaljna struktura zavisi od početnog stanja mezoderma embriona od kojeg se razvija, dvije šarenice identičnih blizanaca ili dvije šarenice jedne osobe imaju međusobno nepovezane detalje.
- Lakoća registrovanja slike šarenice na distanci od osobe bez fizičkog kontakta, nenametljivo i moguće neupadljivo
- Polarna geometrija koja dozvoljava prirodni koordinatni sistem i originalne coordinate

Objašnjenje izbora teorije i tehnologije za rješavanje problema

Za rješavanje problema segmentacije dužice oka sa slike oka izabrali smo Daugmanov algoritam.

Lokalizacija šarenice oka se smatra najtežim dijelom algoritma za biometrijsko prepoznavanje na osnovu slike oka jer definiše unutrašnju i vanjsku granicu šarenice oka koja se koristi za dalju analizu. Ovdje je glavni cilj odstraniti nekorisne informacije, kao što su zjenica i dio izvan šarenice (bionjača, trepavice, koža). R. Wildes koristi Hough-ove transformacije za detekciju konture šarenice. Daugman predlaže integro-diferencijalni operator za pronalaženje kontura zjenice i šarenice. Za Daugmanov algoritam se tvrdi da je naefikasniji.

Daugmanov algoritam

Daugman predlaže integro-diferencijalni operator za pronalaženje kontura zjenice i šarenice:

$$\max(r, x_0, y_0) \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r, x_0, y_0} \frac{I(x, y)}{2\pi} ds \right|$$

gdje su x_0, y_0 i r : koordinate centra i poluprečnik grube procjene kružnice (za zjenicu i šarenicu). $G_\sigma(r)$: Gausova funkcija. $I(x, y)$: originalna ulazna slika.

Zhavaljujući kružnom obliku šarenice, lokalizacija njenih granica se primjenjuje na ulaznu sliku $I(x, y)$ pomoću integro-diferencijalnog operatora koji traži maksimum zamućene parcijalne derivacije, povećavajući poluprečnik r , normalizovanog integrala određenog centralnim koordinatama (x_0, y_0) i poluprečnikom r . $*$ označava konvoluciju, a Gausova funkcija je funkcija zamućenja (smoothing function) sa skalom σ . Cijeli operator se ponaša kao detektor kružnih ivica, koji iterativno traži maksimum definisane konvolucije gdje se poluprečniku povećava vrijednost, a tri parametra (coordinate centra (x_0, y_0) i poluprečnik r) definišu putanju integracije.

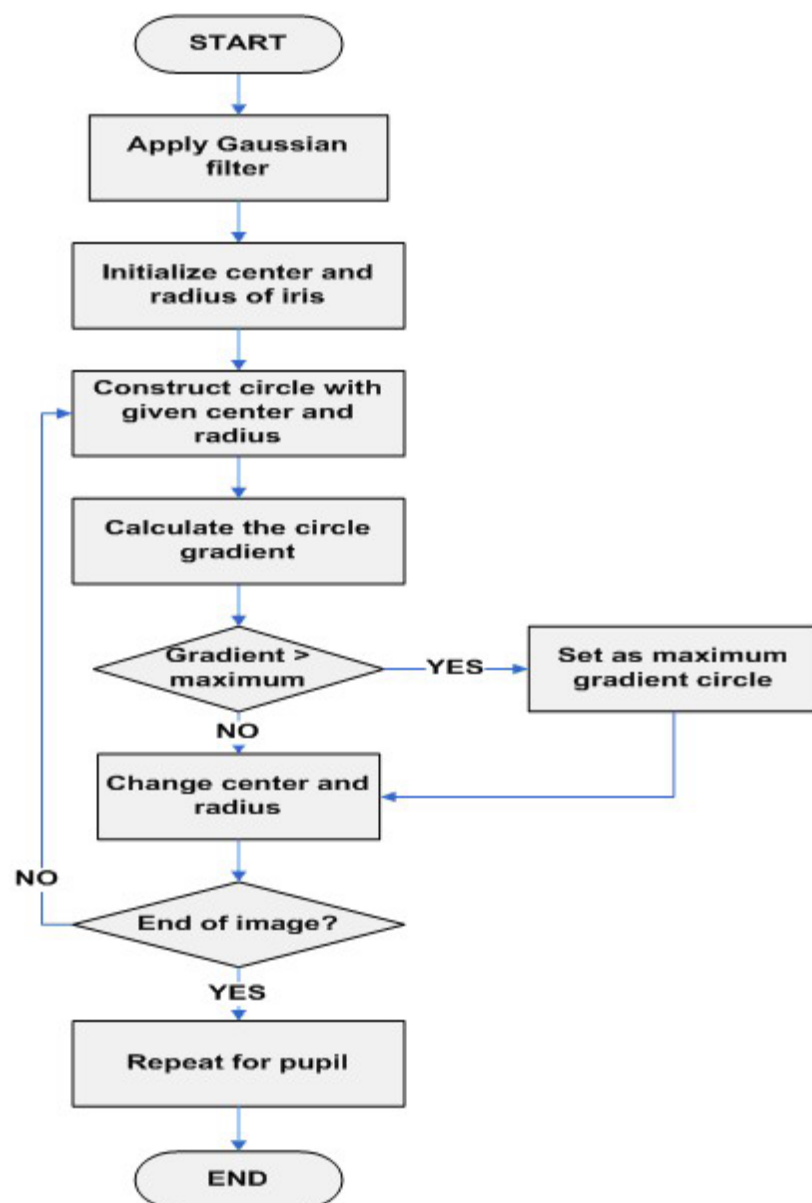
Navedeni algoritam se radi dva puta, prvi put da se detektuje valjska ivica šarenice, a drugi put da se detektuje ivica zjenice. Potrebno je naglasiti problem osvjetljenja unutar zjenice koji se javlja u vidu kruga sa veoma visokim nivoom osvjetljenosti (blizu čiste bijele boje). Algoritam ovaj bijeli krug može shvatiti kao konturu zjenice. Zbog toga, mora se definisati minimalni poluprečnik zjenice.

Optimizovani Dougmanov algoritam

Jedno rješenje ovog problema je modifikacija integro-diferencijalnog operatora tako da ignoriše sve krugove koji imaju maker jedan piksel čija je vrijednost veća od neke granične vrijednosti. Ova granična vrijednost je 200 za sive slike. Drugo rješenje je da se svi pikseli čija je vrijednost veća od granične pretvore u crne piksele.

Optimizovani Dougmanov algoritam daje mnogo bolje rezultate od Dougmanovog algoritma bez optimizacije.

Dijagram toka optimizovanog Dougmanovog algoritma dat je na sljedećoj slici:



Detalji rješenja na nivou implementacije

Implementacija obuhvata procesiranje dvije vrste slika oka. Prva grupa obuhvata slike oka gdje se vidi cijela zjenica i šarenica, bez preklapanja od strane kapaka ili trepavica. U drugu grupu spadaju slike gdje je šarenica djelimično preklopljena kapcima i trepavicama. Za drugu grupu slika implementirali smo segmentaciju koja odstranjuje donji kapak. Za gornji kapak su potrebni statistički podaci, sa kojima ne raspolažemo, pa ga nismo ni mogli odstraniti.

Za segmentaciju prve grupe slika potrebni su sljedeći fajlovi: `thresh.m`, `partiald.m`, `lineint.m`, `search.m`, `drawcircle.m`.

Za segmentaciju druge grupe slika potrebni su sljedeći fajlovi: `thresh1.m`, `partiald.m`, `lineint.m`, `search.m`, `drawcircle.m`, `eyelid.m`, `traceLine.m`.

Implementacija rješenja radi za slike u boji dimenzija 200*150, kao i za neke slike dimenzija 300*400. Glavna funkcija kojom se poziva rješenje je **thresh.m**, odnosno **thresh1.m**.

Thresh.m

Thresh.m je funkcija koja traži centralne koordinate zjenice i šarenice kao i njihove poluprečnike. Koristi 'Camus&Wildes' metodu da selektuje moguće koordinate centra. Ova metoda se sastoji od određivanja praga iza čega slijedi provjera da li je selektovana tačka lokalni minimum u prozoru od 3x3 piksela. Ove tačke su moguće koordinate centra šarenice. Jednom kada je šarenica detektovana (korištenjem Dougmanovog metoda) prelazimo na detekciju centra zjenice, tako što pretražujemo prozor od 10x10 piksela u okolini centra zjenice varirajući radijus dok se ne nađe maksimum (koristeci Daugmanov integrodiferencijalni operator).

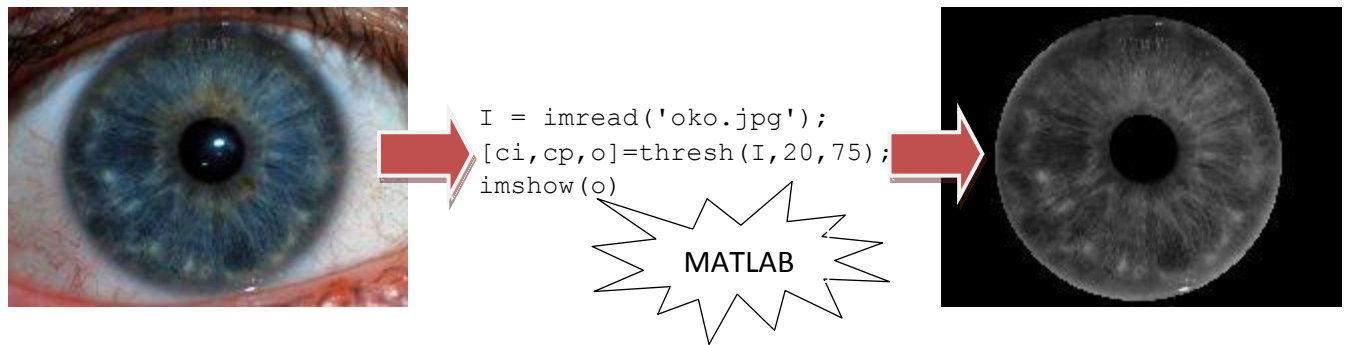
ULAZI:

- **I**: slika za segmentaciju
- **rmin ,rmax**: minimalna i maksimalna vrijednost poluprečnika šarenice

IZLAZI:

- **cp**: parametri[xc,tc,r] granica zjenice
- **ci**: parametri[xc,yc,r] granica šarenice
- **out**: segmentirana slika

Segmentacija slike pozivom ove funkcije data je na sljedećem dijagramu.



Sada slijedi detaljniji opis zanimljivih dijelova ove funkcije.

Na samom početku, funkcija sliku u boji pretvara u sivu sliku.

```
I = rgb2gray(I);
```

Nakon skaliranja slike, slika se pretvara u double.

```
I=im2double(I);
```

U promjenljivoj pimage čuva se originalna slika koja će poslije poslužiti za prikaz segmentirane slike.

```
pimage=I;
```

Zatim, odstranjujemo odsjaj koji se može javiti u zjenici. Kao što se može primijetiti sa prethodnog dijagrama, izlazna slika više nema onaj odsjaj koji je bio prisutan na originalnoj slici.

```
I=imcomplement(imfill(imcomplement(I),'holes'));
```

BW2 = IMFILL(BW1,'holes') popunjava šupljine u ulaznoj slici. Šupljina je skup piksela pozadine kojima se ne može pritupiti popunjavajući pozadinu od ivica slike.

Sada je potrebno naći koordinate centra šarenice. Kreiramo vektor kolonu elemenata slike koji su izabrani pragom:

```
rows=size(I,1);  
cols=size(I,2);  
[X,Y]=find(I<0.5);  
s=size(X,1);
```


U sljedećoj for petlji skeniramo susjedstvo selektovanog piksela da bismo provjerili da li je on lokalni minimum. Ako nije, dodjeljuje mu se vrijednost NaN.

```
for k=1:s
    if (X(k)>rmin) & (Y(k)>rmin) & (X(k)<=(rows-rmin)) & (Y(k)<(cols-
rmin))
        A=I( (X(k)-1):(X(k)+1), (Y(k)-1):(Y(k)+1));
        M=min(min(A));
        if I(X(k),Y(k))~=M
            X(k)=NaN;
            Y(k)=NaN;
        end
    end
end
```

Brišemo sve piksele koji nisu lokalni minimum (koji su postavljeni na NaN).

```
v=find(isnan(X));
X(v)=[];
Y(v)=[];
```

Zatim, brišemo sve piksele koji su tako blizu granice da nije moguće da budu koordinate centra.

```
index=find( (X<=rmin) | (Y<=rmin) | (X>(rows-rmin)) | (Y>(cols-rmin)) );
X(index)=[];
Y(index)=[];
```

Ponovo računamo dimenzije nakon brisanja nepotrebnih elemenata.

```
N=size(X,1);
```

Definišemo dva niza maxb i maxrad u koje će se smjestiti maksimalna vrijednost zamućenja za svaku selektovanu centralnu tačku i odgovarajući poluprečnik. Zatim se nađe maksimalna vrijednosti zamućenja skeniranjem svih koordinata centra.

```
maxb=zeros(rows,cols);
maxrad=zeros(rows,cols);
for j=1:N
```

```
[b,r,blur]=partiald(I,[X(j),Y(j)],rmin,rmax,'inf',600,'iris');%c
oarse search
    maxb(X(j),Y(j))=b;
    maxrad(X(j),Y(j))=r;
end
[x,y]=find(maxb==max(max(maxb)));
```


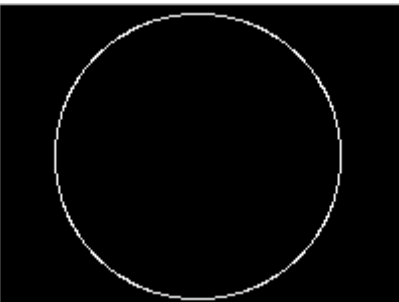
Nalaženje maksimalne vrijednosti zamućenja skeniranjem svih koordinata²:

```
ci=search(I,rmin,rmax,x,y,'iris');
```

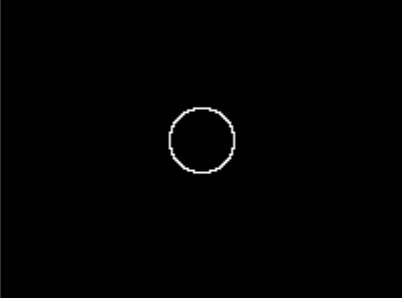
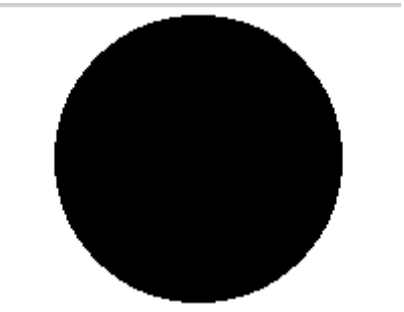
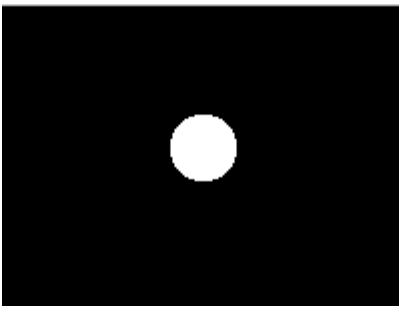

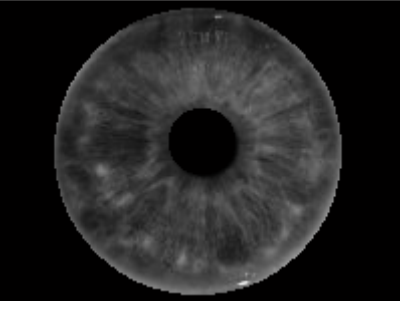
Sada je potrebno naći koordinate centra zjenice i njen poluprečnik tako što skeniramo prozor dimenzija 10*10 oko centra šarenice.

```
cp=search(I,round(0.1*r),round(0.8*r),ci(1)*scale,ci(2)*scale,
'pupil');
```

Koraci dobijanja konačne segmentirane slike prikazani su sljedećom tabelom:

<pre>slika = zeros(150, 200);</pre>	
<pre>out1=drawcircle(slika,[ci(1),ci(2)],ci(3),600);</pre>	

² Ostali m-fajlovi će biti naknadno detaljnije objašnjeni.

<pre>out2=drawcircle(slika,[cp(1),cp(2)],cp(3),600);</pre>	
<pre>out1=1-imfill(out1,'holes');</pre>	
<pre>out2=imfill(out2,'holes');</pre>	
<pre>out = 1-xor(out1,out2);</pre>	
<pre>out = out.*pimage;</pre>	

Partiald.m

Izračunava parcijalni izvod integrala koji je normalizovan po kružnoj putanji, za svaki od kandidata za centre. Kako je sama slika digitalizovana i kvantizovana te ima samo ograničen broj vrijednosti na domenu u kojem je predstavljena, slijedi da je dovoljno izračunati sumu vrijednosti svjetlina po kružnim putanjama, koje su aproksimirane odgovorajućim n-touglovima.

Pomenuti proračun suma koje predstavljaju integral se izvodi funkcijom lineint.m

Lineint.m

Izvršava izračun zatvorneog integrala po kružnoj putanji. Ima mogućnost proračuna integrala, posebno za zjenicu, a posebno za šarenicu, gdje se proračun za šarenicu ne izvodi po cijeloj dužini zbog mogućeg preklapanja sa kapkom. Za slučaj da pomenuta putanja iskače iz domena slike, vrijednost integrala se automatski izjednačava sa 0.

Search.m

Izvršava finu pretragu centra zjenice, u susjednih 10x10 piksela centra šarenice. Prilikom ove pretrage primjenjuje se identičan postupak, kao pri traženju centra šarenice, pri čemu se pretraži pomoću parcijalnog izvoda niza linijskih integrala podvrgava svih 10x10 piksela iz susjedstva centra šarenice.

Drawcircle.m

Funkcija koja na osnovu proslijeđenog centra šarenice, radijusa i broja željenih stranica iscrtava krug na slici, koju proljeđuje na izlaz. Pri ovome se na osnovu broja ivica n, računa korak prostornog ugla theta, koji se koristi pri proračunu koordinata za iscrtavanje kruga.

```
theta=(2*pi)/n;  
angle=theta:theta:2*pi;  
x=C(1)-r*sin(angle);  
y=C(2)+r*cos(angle);
```

Tresh1.m

Tresh1.m je funkcija koja se koristi za rad sa šarenicom, tj. dužicom, koja je dijelom prikivena kapkom. U trenutnoj implementaciji omogućen je rad samo sa donjim kapkom, jer je za uklanjanje gornjeg kapka i trepavica potrebno poznavanje statistike, s kojom nismo raspolagali.

U svojoj strukturi, tresh1.m je potpuno identičan tresh.m funkciji osim što na samom kraju poziva funkciju eyelid.m, koja vrši detekciju donjeg kapka i vraća kao rezultat masku kojom se donji kapak može isključiti iz daljeg poređenja.

```
out1=eyelid(pimage,C,cp(3),ci(3));
```

Nakon ovoga, se prethodna maska segmentacije, na kojoj kapak nije uklonjen, pomnoži sa novodobijenom maskom i kao rezultat daje segmentiranu sliku bez donjeg kapka.

```
out = out.*out1;
```

Razlog za uuspješan rad sa donjim kapkom leži u činjenici da je on rijetko narušen šumovima, tj. trepavicama.

Eyelid.m

Ideja ove funkcije je da razdvoji donji kapak u dvije regije, donji lijevu i donju desnu. Obe regije graniče sa centrom zjenice.

Početak se zasniva na uzimanju jednog okomitog vektora sa granicama od kraja same zjenice, do kraja same šarenice i detektovanja maksimalne promjene(koja se podudara sa prelaskom šarenice na kapak) korištenjem parcijalnog izvoda.

```
vektor=I(C(1)+rPupil+4:C(1)+ rIris-1,C(2));  
D=diff(vektor);  
D=D';  
D=[0 D];  
D=D';  
f=fspecial('gaussian',[1 5]);  
blur=convn(D,f,'same');  
blur=abs(blur);  
[b,i]=max(blur);  
ref=C(1)+i+rPupil+4;  
refPocetno = ref;
```

Detekcijom samog maksimuma, uzima se njegov indeks i postavlja za referentnu tačku kapka, s koje se vrši pretraga lijevo i desno.

Od sada pa na dalje, uzimaju se mali prozori od po 6 piksela vertikalno i unutar njih vrši detekcija maksimalne promjene koja u teoriji označava granicu šarenice i kapka. Ovaj prozor se u jednoj for petlji pomijera u lijevo dok se ne stigne do granice šarenice, a onda se isti postupak ponavlja za desnu stranu.

Međutim zbog nesavršenosti samog oka, često smo dolazili u situaciju da dođe do naglog „propadanja piksela“, pa bi rezultata maskiranja donjeg kapka često bio nepotpun.



Pomenuti problem je riješen funkcijom traceLine.m čija će uloga biti objašnjena naniže.

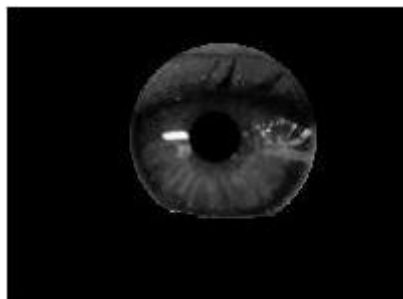
traceLine.m

Bez ulaženja u detalje koda, koji je priložen na samom popratnom CD-u zajedno sa komentarima, ideja je da se naredna dva susjedna pomjeraja u lijevo, odnosno u desno ispituju, prije prihvatanja nove tačke za referentnu. Zbog prirode donjeg kapka nije moguće da do propadnaja dođe dva puta zaredom jer bi to narušilo njegov oblik. Upravo zbog toga, ako se i detektuju dva propadanja, vrši se korekcija referentne tačke, izborom sljedećeg manjeg izvoda koji u sebi sadrži korektnu informaciju.

Slika dobivena pomenutim postupkom se odlikuje korektnom segmentacijom.



Množenjem sa prethodnom maskom za segmentaciju, segmentira samo šarenicu oka, dobija se sljedeća slika.



Diskusija i prijedlozi daljeg poboljšanja

U toku našeg istraživanja problema segmentacije šarenice slike oka, uočili smo sljedeće mogućnosti poboljšanja:

- Detekcija autentičnosti ljudskog oka. Ovo se odnosi na mogućnost razlikovanja pravog ljudskog oka od podmetnutog lažnog oka (stakleno oko, slika oka ili sočivo). Prijedlog implementacije: uzimanje sekvence uzastopnih slika oka u kratkom vremenskom intervali na kojima bi se jednostavno vidjelo širenje zjenice koja pokazuje da se radi o pravom ljudskom oku.
- Detekcija da li je slika dobar kandidat za segmentaciju (slika oka može biti izvan fokusa, isuviše zaklonjena kapkom ili zamućena uslijed naglog pokreta). Prijedlog implementacije: ispitivanje raspodjele frekvencija dijela slike oka. Najčešće, prisustvo velikog broja visokih frekvencija ukazuju na prisustvo trepavica koje zaklanjaju šarenicu, a prisustvo velikog broja niskih frekvencija ukazuju na zamućenost slike.
- Automatsko pozivanje funkcija `thresh` i `thresh1` u zavisnosti od toga da li je šarenica zaklonjena kapkom ili ne.
- Detekcija gornjeg kapka. Prijedlog implementacije: statističke metode bi omogućile uklanjanje trepavica, a ostatak postupka bi se sveo na našu funkciju koja detektuje donji kapak.
- Kako šarenica i zjenica nisu savršeni krugovi, u cilju povećanja efikasnosti identifikacije osobe, trebalo bi ih izdvojiti u njihovom originalnom obliku. Implementacija rješenja: varijacijom pojedinih dijelova kružnice koje aproksimiraju šarenicu i zjenicu i njihovim

uglašavanjem dobila bi se putanja koja predstavlja originalnu putanju šarenice odnosno zjenice.

Literatura

- John G. Daugman, *High Confidence Visual Recognition of Persons by a Test of Statistical Independence*
- Dr. Mohamed A. Hebaishy, *Optimized Daugman's Algorithm for Iris Localization*
- MATLAB-ov help
- Želimir Radmilović (2008), *Biometrijska identifikacija*, stručni članak