Identifikacija osoba pomoću računalnog vida

Adamčević, Matej

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:513941

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

IDENTIFIKACIJA OSOBA POMOĆU RAČUNALNOG VIDA

Diplomski rad

Matej Adamčević

Osijek, 2017.

SADRŽAJ

1. Uvod 1
2. Prijedlog sustava za identifikaciju i kontrolu pristupa 4
3. Realizacija sustava za identifikaciju i kontrolu pristupa 8
3.1. Dohvaćanje fotografije 8
3.2. Programsko rješenje za identifikaciju 11
3.2.1. Predobrada slike 11
3.2.2. Lokalizacija zjenice 17
3.2.3. Optimizacija položaja zjenice 21
3.2.4. Lokalizacija šarenice 25
3.2.5. Uzorkovanje šarenice 30
3.2.6. Generiranje binarnog koda 34
3.2.7. Uspoređivanje binarnih kodova 37
3.3. Kontrola prava pristupa 39
3.3.1. Baza podataka 39
3.3.2. Provjera prava pristupa korisnika 40
3.4. Korisničko sučelje i administracija korisnika 42
4. Rezultati 44
5. Zaključak 50
Reference 52
Sažetak 54
Abstract 55
Životopis 56

1. UVOD

U suvremenim organizacijskim jedinicama (npr. poduzeća, projektni timovi), ključno je kontrolirati koja osoba, ovisno o svojoj ulozi, ima koje ovlasti. Najvažniji su pritom pravo pristupa i izmjene podataka te pravo pristupa prostorijama. Prije korištenja računalnih sustava za ovu namjenu, bio je potreban čovjek koji upravlja pravima pristupa (npr. portir koji čuva ključeve određenih prostorija). Razvojem računalne tehnologije, čovjek je postao zamjenjiv. Sustavi autentikacije prešli su na jednokratne lozinke, korisnička imena i osobne lozinke, a u posljednjim desetljećima pojavljuju se sustavi zasnovani na biometriji.

Biometriju čine sve metode očitavanja fizikalnih karakteristika ljudskog tijela od kojih vrijedi izdvojiti otisak prsta, značajke lica i uzorak šarenice oka kao obilježja jedinstvena pojedincu. Dok je boja šarenice izravno određena genetikom, uzorak šarenice nastaje pod dodatnim utjecajem slučajnih događaja i okolnosti za vrijeme morfogeneze, tj. formiranja šarenice. Ti događaji i okolnosti poznati su kao epigenetski mehanizmi nasljeđivanja, a posljedica njihovog djelovanja je jedinstvenost uzorka svake pojedine šarenice. To vrijedi i za šarenice iste osobe – uzorci šarenice lijevog i desnog oka formiraju se pod utjecajem međusobno neovisnih epigenetskih mehanizama. Uzorak šarenice je izuzetno pogodno svojstvo za analizu biometrijom. Razlozi tome su zaštićenost šarenice od vanjskih utjecaja i njena gotovo neprestana vidljivost te relativna nepromjenjivost samog uzorka šarenice u vremenu [1] (za razliku od npr. karakteristika lica na koje mogu utjecati boranje ili opuštanje kože, nastanak većih ožiljaka ili deformacija i sl.).

Svaki sustav identifikacije ima dva cilja, a to su minimizirati stopu lažnog prepoznavanja (engl. *false recognition rate*) i lažnog odbijanja (engl. *false rejection rate*). Stopa lažnog prepoznavanja, tj. postotak osoba bez prava pristupa kojima je pristup odobren, može se smatrati najvažnijom veličinom svih sustava identifikacije s aspekta sigurnosti. Stopa lažnog odbijanja odnosno postotak osoba kojima pristup nije odobren, a trebao je biti, više govori o kvaliteti raspoznavanja danog sustava. U idealnom sustavu obje stope iznose 0%, odnosno sustav nikad neće prihvatiti korisnika bez prava pristupa niti odbiti korisnika s pravom pristupa.

Sustavi identifikacije osoba na temelju uzorka šarenice postižu znatan napredak 90-ih godina 20. stoljeća. Tada su predstavljena dva generalna rješenja za problem identifikacije osoba temeljem fotografije oka. Prvo rješenje je predstavio današnji profesor na Sveučilištu u Cambridgeu, John Daugman [1]. Rješenje koje je tada predložio prof. Daugman uključivalo je i poseban matematički operator koji se koristio za lokalizaciju granica šarenice, a ostalo je jedna od

vodećih smjernica u razvoju sustava ovakvog tipa. Drugo rješenje predstavio je današnji profesor na Sveučilištu York u Torontu, Richard P. Wildes [2]. Njegovo rješenje zasniva se isključivo na mogućnostima računalnog vida, što je vodeća misao i ovog rada. Od tada su ovi sustavi znatno napredovali, pa čak i doživjeli komercijalnu upotrebu. Primjerice, upravo je Daugman 2011. vodio projekt uspostavljanja ovakvog sustava na državnoj razini u Indiji s ciljem povećanja dostupnosti državnih usluga. Početkom 2017. sličan sustav po prvi put je primijenjen u potrošačkoj elektronici. Južnokorejska tehnološka tvrtka Samsung predstavila je prvi mobilni telefon s opcijom otključavanja uređaja temeljem analize šarenice. Razlika između dana dva primjera je ta što sustav primijenjen u Indiji ima zadatak prepoznati bilo koga iz baze od preko milijardu aktivnih korisnika, a mobilni uređaj prepoznati najčešće samo jednog aktivnog korisnika.

Nagli razvoj računalne tehnologije učinio je potrošačku elektroniku pristupačnijom, jeftinijom i kvalitetnijom. Posljednjih godina javljaju se projekti kojima je cilj samogradnju u elektronici dovesti široj publici, a kao primjer mogu se istaknuti Micro:bit, Arduino i Raspberry Pi. Dok je prvi prvenstveno namijenjen kao edukativna platforma za djecu, Arduino odnosno Raspberry Pi predstavljaju seriju mikroupravljača odnosno mikroračunala koja su sposobna obavljati kompleksnije zadatke. Raspberry Pi uređaji imaju dovoljno računalne moći za pogoniti operacijski sustav (kao npr. *Raspbian,*prilagođena distribucija Linux OS-a), a omjerom cijene i računalne moći pokazuju se kao vrlo dobro rješenje za ugradbene računalne sustave (engl. *embedded computer systems*) [3].

Cilj ovog rada je razviti konceptualni model cjelovitog sustava identifikacije koji bi se primjenom našao između onih koje je primijenio Daugman u Indiji odnosno tvrtka Samsung u mobilnom uređaju, to jest omogućiti bazu korisnika veću nego što je to moguće na mobilnom uređaju, ali ograničiti se realniju situaciju manjeg poduzeća (naspram Daugmanovog rješenja na državnoj razini). Izvedba rješenja temeljena je na postupcima koje su predložili upravo Daugman i Wildes. Takav sustav uključuje podsustav za fotografiranje, algoritam za izdvajanje i uspoređivanje uzoraka šarenice, podsustav za nadzor i upravljanje korisnicima te grafičko sučelje prema krajnjem korisniku. Razvijeno rješenje je konceptualne prirode, tj. za potrebe ovog rada neće dobiti konkretnu primjenu. Izvedba sustava za stvarnu primjenu iz razvijenog modela uključivala bi manje dodatke karakteristične za konkretnu primjenu (npr. automatizacija sustava ključanja vrata u slučaju primjene na određenoj prostoriji).

U drugom poglavlju ovog rada dan je pregled problematike i prijedlog rješenja razvoja opisanog sustava, a njegova realizacija po danom prijedlogu opisana je u trećem poglavlju. Četvrto

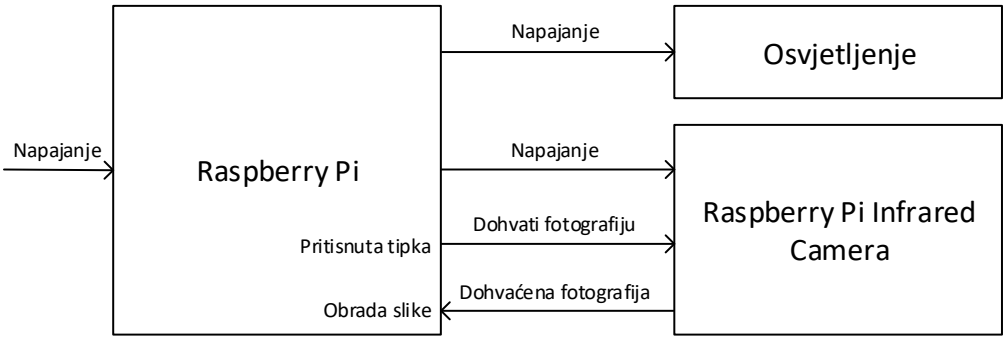
poglavlje sadrži pregled rezultata identifikacije dobivenih korištenjem predstavljenog rješenja zajedno sa vremenima izvođenja programskog rješenja na dvije različite platforme (osobno računalo sa Windows OS i mikroračunalo sa Linux OS).

2. PRIJEDLOG SUSTAVA ZA IDENTIFIKACIJU I KONTROLU PRISTUPA

Problematika razvoja opisanog sustava može se podijeliti na tri glavna funkcijska odjeljka:

1. sklopovsko i programsko rješenje za dohvaćanje fotografije,
2. programsko rješenje za identifikaciju (analizu uzorka šarenice) i
3. programsko rješenje za kontrolu pristupa i
4. programsko rješenje za korisničko sučelje i upravljanje korisnicima.

Sklopovsko rješenje ostvareno je koristeći *Raspberry Pi* (verzija *3B)* mikroračunalo, zajedno sa *Raspberry Pi Infrared Camera* (verzija *1.3*) [4] modulom i sustavom osvjetljenja (Sl. 2.1.). Samostalno, kamera ne daje mogućnost optičkog uvećanja slike kao ni mogućnost promjene fokalne duljine, što fotografiranje objekata malih dimenzija čini neizvedivim. Kako bi fotografija bila dovoljno kvalitetna odnosno sadržavala dovoljno informacija o uzorku šarenice, potrebno je uvesti dodatne leće koje mijenjaju optička svojstva kamere. Mehanizam dohvaćanja fotografije je pojednostavljen fotografiranjem u trenutku pritiska određene tipke na tipkovnici, stoga neće biti detaljnije opisivan.



Sl. 2.1. Blok dijagram sustava za dohvaćanje fotografije

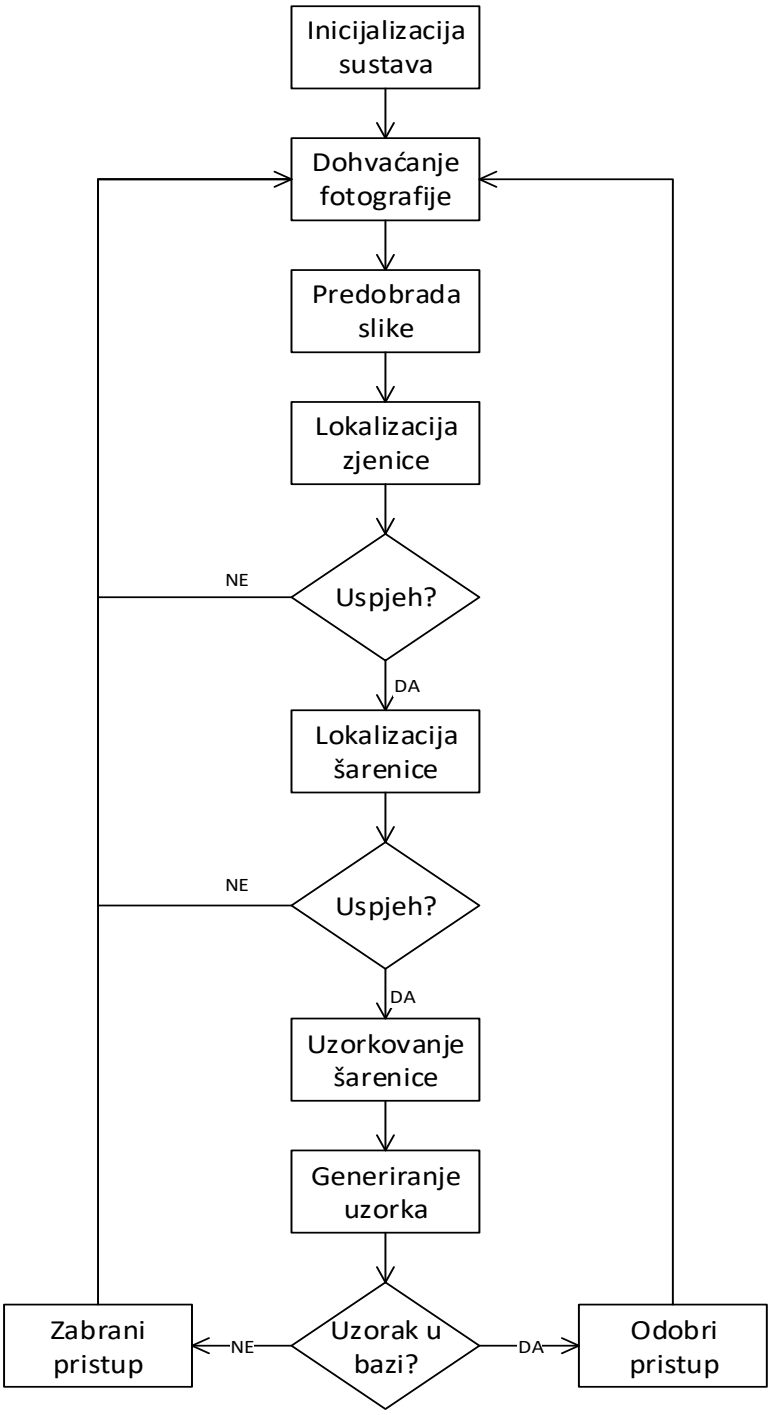
Programsko rješenje razvija se na osobnom računalu, koristeći *C++* razvojno okruženje *Qt Creator 4.2.1.* (*Qt 5.8.0.*) [5] te biblioteku za implementaciju metoda računalnog vida *openCV 3.2.0* [6]. Za održavanje i upravljanje bazom podataka korisnika odabrana je *SQLite* biblioteka (verzija *3.21.0*) [7].

Postupak identifikacije ima par manjih ciljeva ispred konačnog – raspoznavati osobe temeljem uzorka njihove šarenice. Prvi je precizno izdvojiti područje šarenice od ostatka slike, a drugi iz područja šarenice generirati binarni kod unaprijed određenih dimenzija.

Idejno programsko rješenje za identifikaciju odnosno analizu uzorka šarenice, prikazano blok dijagramom na slici 2.2., napravljeno je po uzoru na rješenja koja su predložili profesori Daugman i Wildes, a predstavlja sustav sastavljen od 8 funkcijskih blokova. Prvi blok predstavlja inicijalizaciju sustava. Ona je potrebna u slučaju da je radi gubitka napajanja ili iz nekih drugih tehničkih razloga, sustav stao s radom. Ukoliko baza korisnika nije bila prazna prije zastoja rada sustava, ona se tijekom inicijalizacije učitava. Blok za dohvaćanje slike predstavlja sve korake između dolaska korisnika u vidno polje *Raspberry Pi Camera* modula do učitavanja fotografije sa modula u radnu memoriju programskog rješenja. Potom slijede postupci predobrade slike, lokalizacije i optimizacije zjenice te šarenice (od trećeg do šestog bloka) čiji je cilj precizno odrediti unutarnju (pupilarnu) i vanjsku (cilijarnu) granicu šarenice.

Ukoliko postupci lokalizacije odnosno optimizacije uspiju, područje unutar granica šarenice (omeđeno sa dvije kružnice, jedna unutar druge) se uzorkuje u pravokutan oblik. Iz uzorkovanog područja šarenice se koristeći Gaborove 2D filtre i metodu binarnog praga generira binarni kod šarenice u obliku matrice unaprijed određenih dimenzija. Generirani kod se uspoređuje sa kodovima svih korisnika pohranjenih u bazi podataka sustava računanjem Hammingove udaljenosti, čiji se rezultat nalazi u intervalu [0, 1]. Hammingova udaljenost od 0 označava da svi pikseli obje matrice imaju iste vrijednosti na istim mjestima, a udaljenost od 1 označava da su dvije binarne matrice komplementarne tj. ne dijele niti jedan podudaran piksel po mjestu i vrijednosti. Cilj ovog rada je i eksperimentalno utvrditi prag prepoznavanja koji bi označavao maksimalnu Hammingovu udaljenost između dvije binarne matrice za koju bi se smatralo da uspoređene matrice binarnih kodova pripadaju oku iste osobe.

Grafičko korisničko sučelje mora omogućiti komunikaciju sustava s korisnikom u vidu ispisivanja poruka o stanju programa (u slučaju neuspjele identifikacije – gdje se pojavio problem i sl.). Dodatno, korisnicima s administrativnim pravima pristupa potrebno je omogućiti pristup i pregled baze korisnika zajedno sa podacima o povijesti odobrenih pristupima.

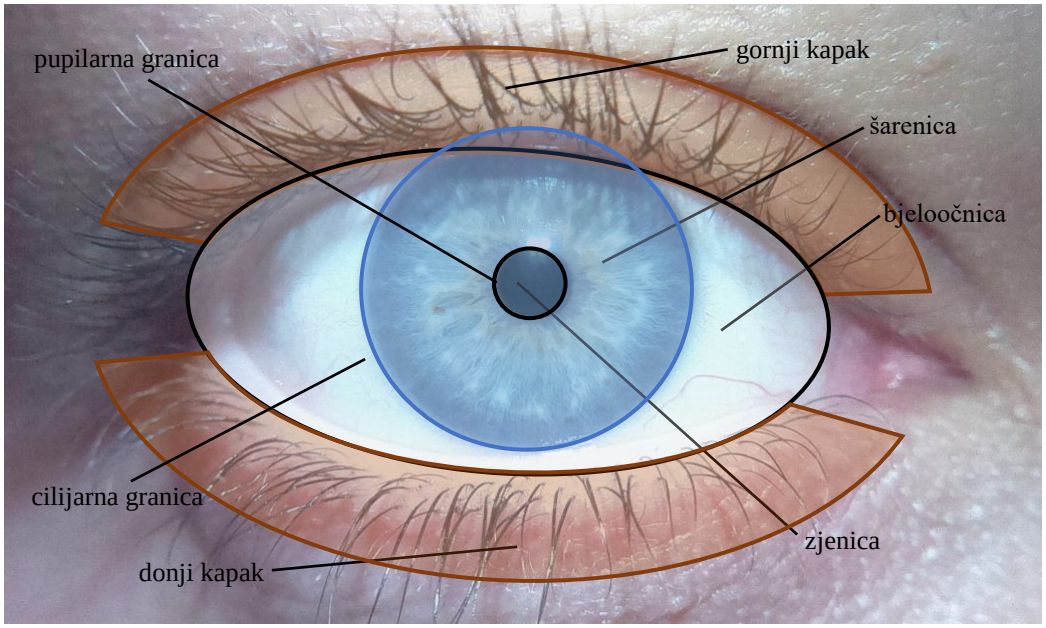


Sl. 2.2. Pojednostavljeni funkcionalni blok dijagram idejnog rješenja za analizu šarenice

U ovom radu nisu korišteni algoritmi lokalizacije bazirani na Daugmanovom integro-diferencijalnom operatoru nego isključivo na mogućnostima funkcija računalnog vida. Prvenstveno misli se na Cannyjev detektor rubova i Houghovu metodu detekcije krugova.

Polazne pretpostavke za ovaj rad su da je na fotografiji oka (primjer na Sl. 2.3.):

1) Područje zjenice tamnije od područja šarenice, tj. sadrži piksele nižih vrijednosti i promjera je unutar nekog praga (2-5 mm [8]),
2) Područje bjeloočnice svjetlije od područja šarenice, tj. sadrži piksele s višim vrijednostima,
3) Refleksija osvjetljenja izrazito svijetla, tj. ima izrazito visoke vrijednosti (iznad određenog praga),
4) Vidljiva pupilarna (uz zjenicu) i cilijarna (uz bjeloočnicu) granica u visini zjenice, s lijeve i desne strane. Cilijarna granica može biti zaklonjena kapcima i s gornje i s donje strane.



Sl. 2.3. Građa i vidljivi elementi oka

Nakon realizacije algoritama za analizu i usporedbu šarenice, potrebno je implementirati sustav upravljanja korisnicima i grafičko sučelje prema krajnjem korisniku. Pristup sustavu upravljanja korisnicima imaju određeni korisnici, tzv. administratori. Potrebno je osigurati opcije dodavanja i brisanja korisnika iz baze, pregled aktivnih korisnika i pregled zadnjih odobrenih pristupa.

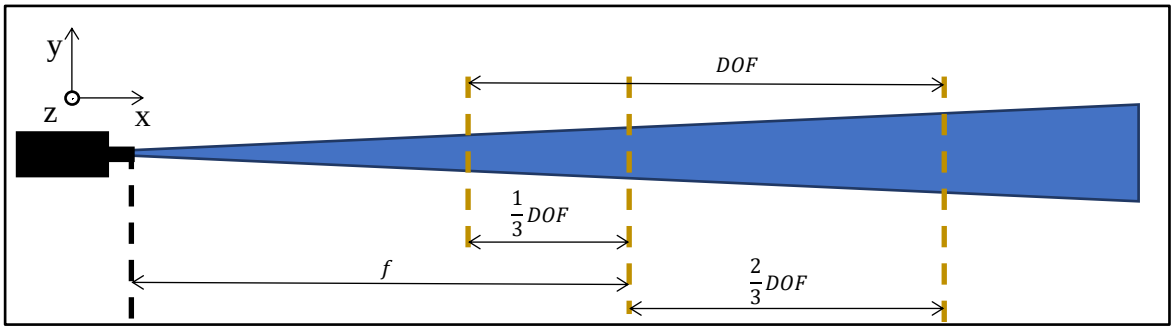
3. REALIZACIJA SUSTAVA ZA IDENTIFIKACIJU I KONTROLU PRISTUPA

Prema prijedlogu danom u drugom poglavlju, realizacija sustava za identifikaciju i kontrolu pristupa razdvojena je na četiri glavna funkcijska odjeljka:

1. Dohvaćanje fotografije oka,
2. Programsko rješenje za analizu šarenice,
3. Kontrola prava pristupa i
4. Korisničko sučelje i administracija korisnika

3.1. Dohvaćanje fotografije

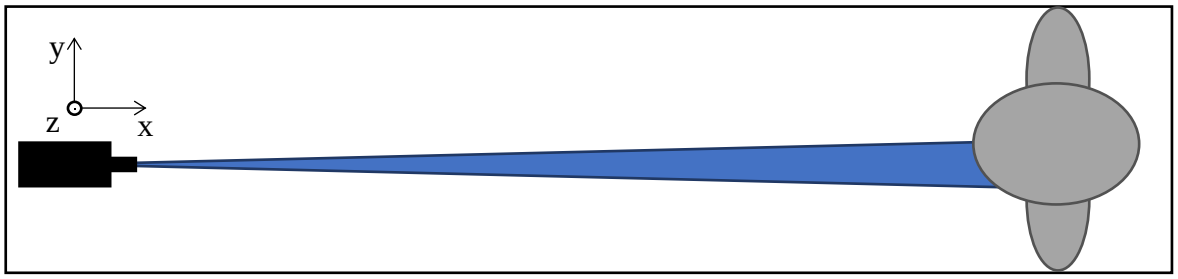
Za dohvaćanje fotografije oka, korišten je kamera modul za *Raspberry Pi* (NoIR kamera, v.1.3) [4] sa fiksnom fokalnom duljinom. Na kvalitetu fotografije utječu prvenstveno fokalna duljina i dubina vidnog polja kamere te razlučivost slike. Pod dubinom vidnog polja smatra se onaj prostor ispred i iza fokalne duljine. U fotografiji se kao pravilo koristi omjer prikazan na slici 3.1. [9].



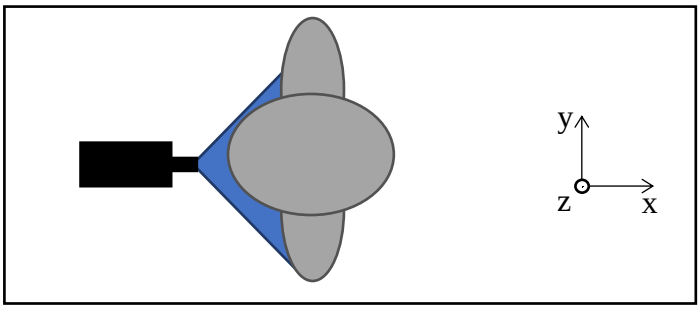
Sl. 3.1. Pojednostavljeni prikaz dubine vidnog polja kamere

Rješenje može biti jedna od dvije mogućnosti, a prva je koristiti teleskopske leće koje povećavaju fokalnu duljinu, gdje korisnik može biti udaljen nekoliko metara od kamere ali ima vrlo uzak prostor na kojem ga kamera vidi. Dodatno, korištenje teleskopskih leća bez opcije prilagodljivog fokusa smanjuje dubinu vidnog polja na svega par centimetara. U slučaju korištenja teleskopske leće s fokalnom duljinom 100 cm, dubina vidnog polja bez prilagodljivog fokusa bila bi svega par centimetara, a raspon vidnog polja manji od 20° [10].

Druga opcija su makroskopske leće. Uporaba makroskopskih leća označava manju fokalnu duljinu, gdje korisnik može biti udaljen tek nekoliko centimetara od kamere, ali šire vidno polje. Dodatno, osvjetljenje subjekta je puno manji problem na manjim udaljenostima. Najveće ograničenje je x-os, odnosno mala dubina vidnog polja. Subjekt se može kao i u slučaju teleskopskih leća može kretati svega nekoliko centimetara ispred i iza fokalne ravnine. Razlika je u tome što pojedinac teže pozicionira na udaljenost od npr. 98-102 cm (Sl. 3.2.) od teleskopske nego 4-8 cm (Sl. 3.3.) od makroskopske leće.



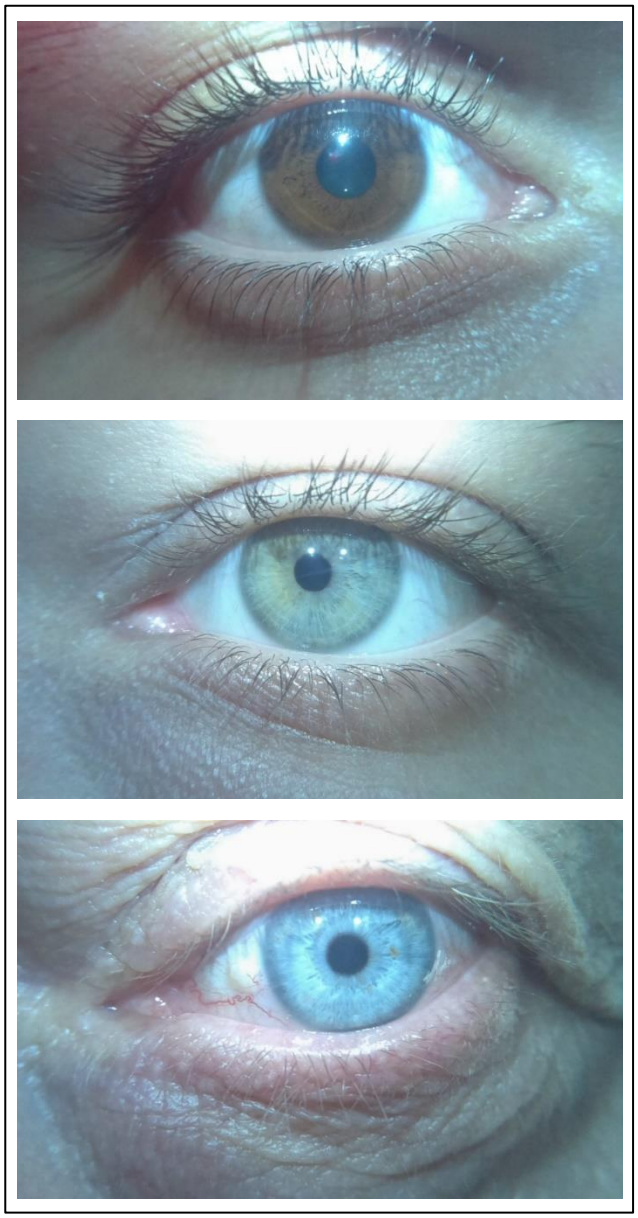
Sl. 3.2. Ilustracija vidnog polja kamere uporabom teleskopskih leća



Sl. 3.3. Ilustracija vidnog polja kamere uporabom makroskopskih leća

Pri izradi ovog rada primijenjene su makroskopske leće prvenstveno radi lakšeg pozicioniranja subjekta u vidno polje kamere (sklopovlje za dohvaćanje fotografije prikazano je prilogu 3.1.). Prema izvoru [11], dovoljno je i 30 piksela između kraja zjenice i kraja šarenice za mogućnost dobrog kodiranja uzorka. Fotografiranjem rezolucijom 1280x800 piksela postiže se polumjer šarenice između od 160 i 200 piksela, a zjenice između 35 i 70 piksela. *Raspberry Pi* kameri moguće je programski podešavati unutarnje parametre kao što je *exposure* odnosno vrijeme izloženosti, tj. prikupljanja svjetla. Koristeći osvjetljenje od jedne bijele LED diode snage 0.1W i nisko vrijeme prikupljanja, moguće je dobiti vrlo detaljnu sliku šarenice sa minimalnom refleksijom. Nisko vrijeme prikupljanja direktno doprinosi minimiziranju refleksije svjetlosti s

reflektivne površine oka na kameru. Fotografije korištene pri izradi ovog rada slikane su raznim vremenima izloženosti iz intervala [50, 250] μs kako bi se razvio algoritam manje osjetljiv na promjenu svjetline. Na slici 3.4. dani su primjeri fotografija smeđeg, zelenog i plavog oka. Na fotografijama je moguće vidjeti kako su uvjeti postavljeni u drugom poglavlju ispunjeni – zjenica je izrazito tamnija od šarenice, šarenica od bjeloočnice, a refleksija osvjetljenja je najsvjetlije područje na slici.



Sl. 3.4. Fotografije smeđeg (gore), zelenog (sredina) i plavog (dolje) oka

3.2. Programsko rješenje za identifikaciju

Programsko rješenje za identifikaciju osoba kao ulaznu veličinu prima fotografiju dohvaćenu sa *Raspberry Pi Camera* modula. Konačan cilj identifikacije je odrediti odgovara li fotografirano oko jednom od korisnika pohranjenih u bazu. Izlazna veličina ovog programskog rješenja je cijeli broj koji označava rezultat identifikacije. U slučaju pozitivnog rezultata neki korisnik iz baze je uspješno identificiran, a u slučaju negativnog osoba nije identificirana.

Realizacija programskog rješenja za analizu šarenice – od dohvaćene fotografije do autentikacije korisnika, podijeljeno je u sedam odjeljaka:

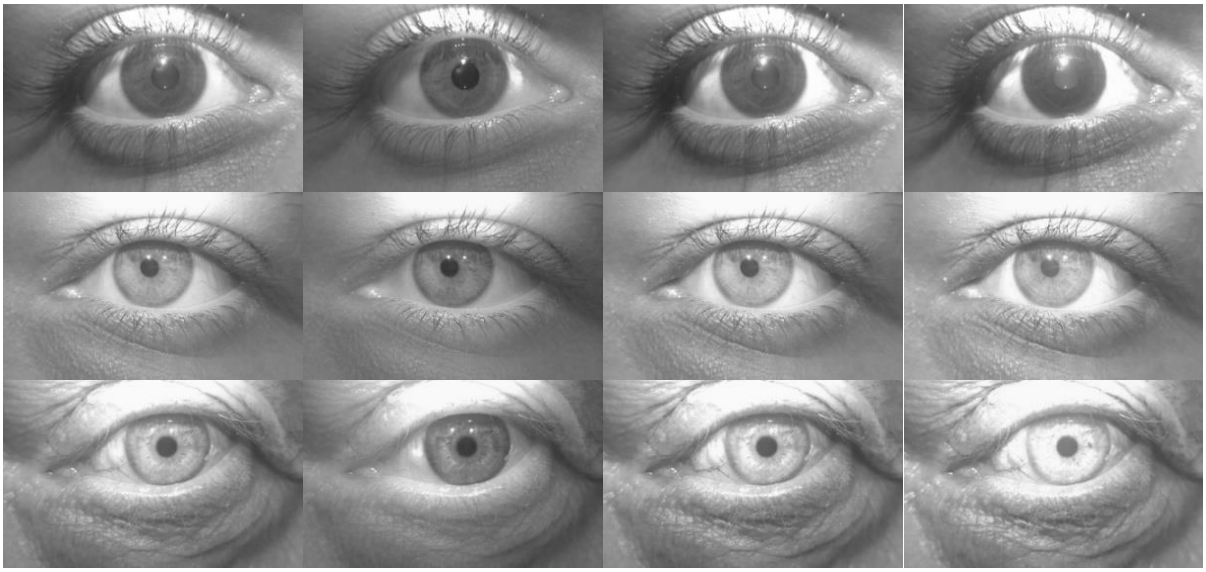
1. Predobrada slike – priprema za lokalizaciju zjenice,
2. Lokalizacija zjenice – određivanje užeg područja oko zjenice,
3. Optimizacija zjenice – određivanje granica zjenice,
4. Lokalizacija šarenice – određivanje granica šarenice,
5. Uzorkovanje šarenice – transformaciju područja šarenice u pravokutni oblik,
6. Generiranje binarnog koda – kodiranje uzorka šarenice u binarnu matricu i
7. Uspoređivanje binarnih kodova – razlikovanje istih od različitih očiju.

3.2.1. Predobrada slike

Prije lokalizacije zjenice, obavlja se predobrada slike kako bi se istaknula željena područja na fotografiji (prvenstveno prijelazi svjetline) te smanjilo računsko vrijeme izvođenja narednih koraka. Za potrebe lokalizacije, gdje kvaliteta uzorka šarenice nije bitna, originalna slika se s početne rezolucije 1280x800 piksela linearnom interpolacijom (*openCV* funkcija *resize* [12]) smanjuje na veličinu 512x320 piksela što uvelike smanjuje cjelokupno vrijeme obrade. Nakon što se originalna slika smanji na željenu veličinu, nad njom se obavlja niz operacija iz područja obrade slike. Dobra predobrada je ključna za daljnji rad algoritma jer o njoj uvelike ovisi uspješnost prvog koraka analize šarenice – gruba lokalizacija zjenice. Iz tog razloga je i prije predobrade dobro znati što se od nje želi dobiti, a u ovom slučaju to je binarna slika rubova koja dobro opisuje granicu zjenice.

Obrada slike u RGB (ili bilo kojem drugom višekanalnom) prostoru boja je i memorijski i vremenski zahtjevnija od obrade jednokanalne slike. Iz tog razloga potrebno je pronaći kanal (ili jednokanalnu kombinaciju više kanala) u kojem je granica zjenice i šarenice sadržana konzistentno, za različite razine osvjetljenja i pigmentacije šarenice. Trokanalna slika u RGB

prostoru boja može se transformirati u jednokanalnu sliku sivih tonova (engl. *grayscale*). Dodatno, slika predstavljena u bilo kojem višekanalnom (*n*-kanalnom) prostoru boja može se razdvojiti na *n* jednokanalnih *grayscale* slika.



Sl. 3.5. S lijeva na desno: sivi tonovi, crveni, zeleni, plavi kanal za smeđe (gore), zeleno (sredina) i plavo (dolje) oko

Primjer rezultata takve transformacije odnosno rastavljanja dan je na slici 3.5. Za oko smeđe (prvi red), zelene (drugi red) i plave (treći red) boje prikazani su rezultati transformacije slike iz RGB u *grayscale* područje (prvi stupac) te pojedinačno crveni (drugi stupac), zeleni (treći stupac) i plavi (četvrti stupac) kanal rezultata rastavljanja RGB slike. Pri obradi slike korisno je na ovakav način razdvojiti originalnu sliku zbog mogućnosti da jedan od kanala sadrži većinu smetnje (kao što npr. refleksija osvjetljenja ima najveći utjecaj u plavom, a najmanji u crvenom kanalu).

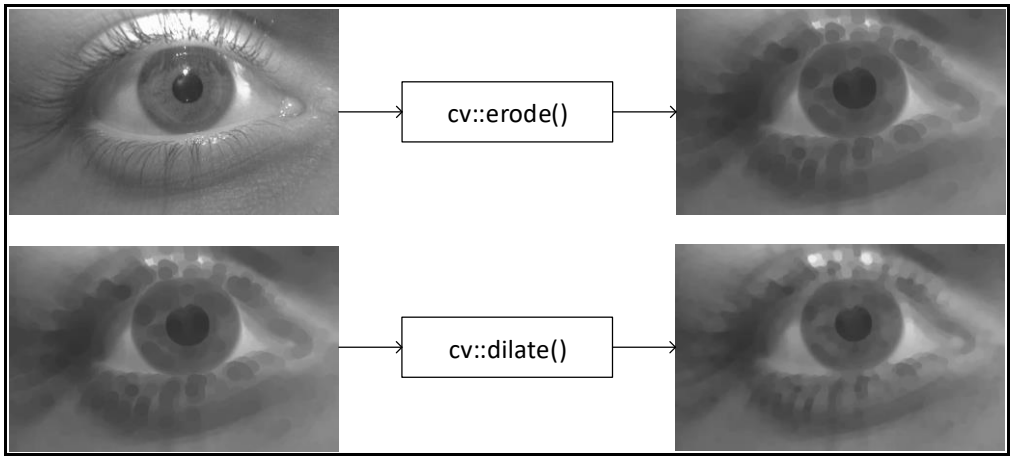
U postupku predobrade koristi se niz *openCV* funkcija za obradu slike. Korištene funkcije i redoslijed pozivanja istih određen je eksperimentalno. Izlazna slika svake funkcije predstavlja ulaznu sliku sljedeće funkcije. Najznačajnije su pritom:

1. Erozija i dilatacija bijele boje (*openCV: erode*); korištene za širenje područja tamnih i sužavanje područja svijetlih nijansi na slici (erozija, Sl. 3.6, gore), odnosno širenje područja svijetlih i sužavanje područja tamnih nijansi (dilatacija, Sl. 3.6., dolje) [13]. Ideja iza erozije, a potom dilatacije je sljedeća:

1.1. Erozijom se širi područje tamnih nijansi (zjenica) a sužava područje svijetlih nijansi (refleksija osvjetljenja, bjeloočnica). Ukoliko se erozija primijeni s prozorom veličine 10x10 piksela, izrazito tamno područje zjenice proširit će se za 5 piksela u svim smjerovima. Pri tome se područje refleksije osvjetljenja smanji za 5 piksela, od vanjskih rubova prema centru.

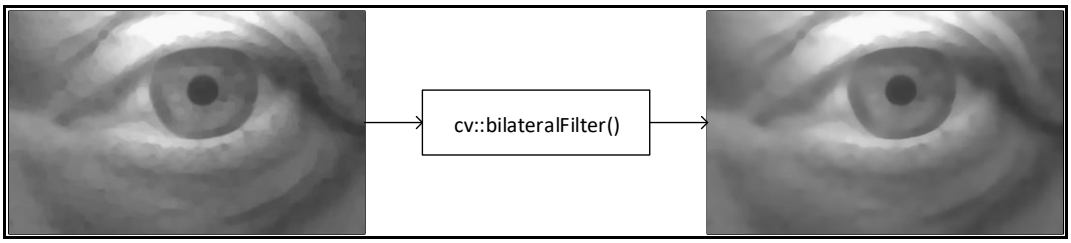
1.2. Ako je prozor erodiranja dovoljno velik, može se u potpunosti ukloniti (erodirati) područje refleksije osvjetljenja. Time se uklanja utjecaj refleksije na lokalizaciju, ali područje zjenice ostalo je prošireno za pola veličine prozora.

1.3. Dilatacijom erodirane slike istom veličinom prozora područje šarenice se vraća u originalne granice, a područje osvjetljenja se „ne vraća“ pošto su erozijom svi svijetli pikseli s tog područja uklonjeni.



Sl. 3.6. Erozija i dilatacija bijele boje

2. Bilateralni filtar (*openCV: bilateralFilter*) grupira piksele uzimajući u obzir dva kriterija: međusobnu udaljenost piksela na slici i sličnost vrijednosti tih piksela. Rezultat je slika prikazana s desne strane na slici 3.7., zamućena u područjima blagih promjena intenziteta svjetline, a izoštrena u područjima većih promjena intenziteta (zjenica – šarenica; šarenica – bjeloočnica). [14]

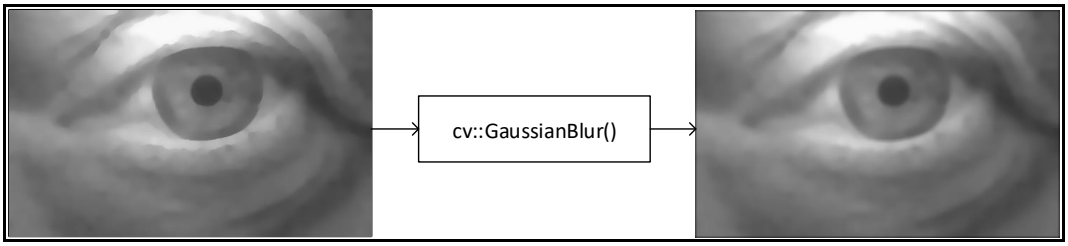


Sl. 3.7. Filtriranje bilateralnim filtrom

3. Gaussova metoda za uklanjanje šuma (*openCV: GaussianBlur*); za uklanjanje Gaussovog šuma na slici. Postupak filtriranja slike Gaussovim filtrom obavlja se piksel po piksel, gdje se svakom pikselu rezultantne matrice R, prikazane na slici 3.8., pripisuje vrijednost ovisna o radijalnoj okolini tog piksela originalne slike [15]. Vrijednost svakog piksela rezultantne slike u središtu okoline dimenzija $m \times m$ na slici računa se kao srednja vrijednost Hadamardovog produkta H matrice okoline O i Gaussove matrice težina G (3-1) prema izrazima i (3-2).

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} = O \circ G = \begin{bmatrix} O_{11} & O_{12} & O_{13} \\ O_{21} & O_{22} & O_{23} \\ O_{31} & O_{32} & O_{33} \end{bmatrix} \circ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & O_{12} & 0 \\ O_{21} & 2 \times O_{22} & O_{23} \\ 0 & O_{32} & 0 \end{bmatrix} \quad (3\text{-}1)$$

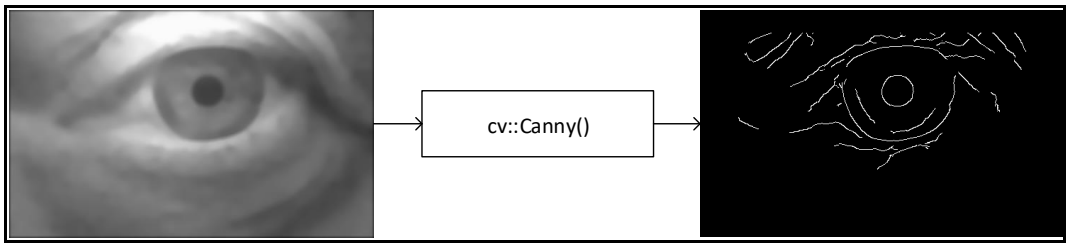
$$R_{i,j} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} H_{i,j}}{\sum_{i=1}^{m} \sum_{j=1}^{m} G_{i,j}} \quad (3\text{-}2)$$



Sl. 3.8. Filtriranje Gaussovim filtrom

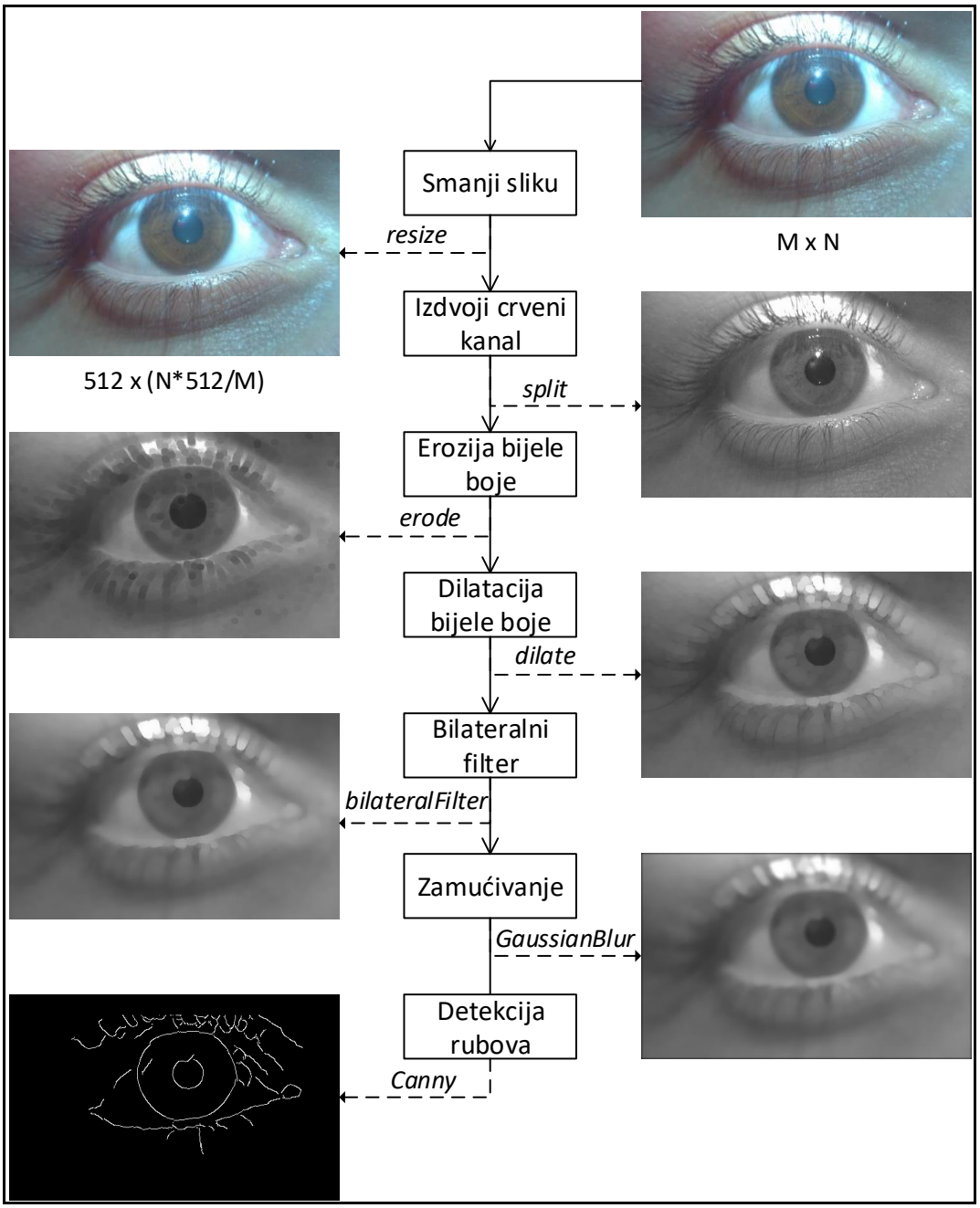
4. Cannyjev detektor rubova (*openCV: Canny*) za detekciju rubova na jednokanalnoj slici, s podesivim parametrima osjetljivosti [16]. Parametri osjetljivosti ovise prvenstveno o ulaznim slikama. Na primjer, u slučaju mutnije slike parametri trebaju biti niži tj. osjetljiviji, jer će svi prijelazi biti blaži nego u slučaju izoštrene slike. U ovom radu

parametri osjetljivosti detektora određeni su empirijskim postupkom, a primjer detektiranih rubova dan je na slici 3.9.



Sl. 3.9. Detekcija rubova Cannyjevim detektorom

Nakon razmatranja nekoliko prostora boja i pojedinačnih kanala, za početnu lokalizaciju zjenice odabran je crveni kanal slike u RGB prostoru boja. Postupak predobrade slike prikazan je na slici 3.10. Na iscrtanim linijama označene su openCV funkcije korištene za generiranje pojedine slike. Ulazna slika rezolucije $M \times N$ piksela skalira se na veličinu 512×320 piksela. Iz skalirane slike u RGB prostoru boja izdvaja se crveni kanal, nad kojim se obavlja funkcija erozije kako bi se u potpunosti uklonilo područje refleksije osvjetljenja i proširilo područje zjenice. Dilatacijom se područje zjenice vraća na originalnu veličinu dok refleksija ostaje uklonjena. Bilateralni i Gaussov filtar koriste se za ublažavanje neželjenih rubova (npr. unutar uzorka šarenice). Cannyjev detektor metodom binarnog praga izdvaja granice prijelaza intenziteta svjetline na slici.



Sl. 3.10. Postupak predobrade slike za lokalizaciju zjenice

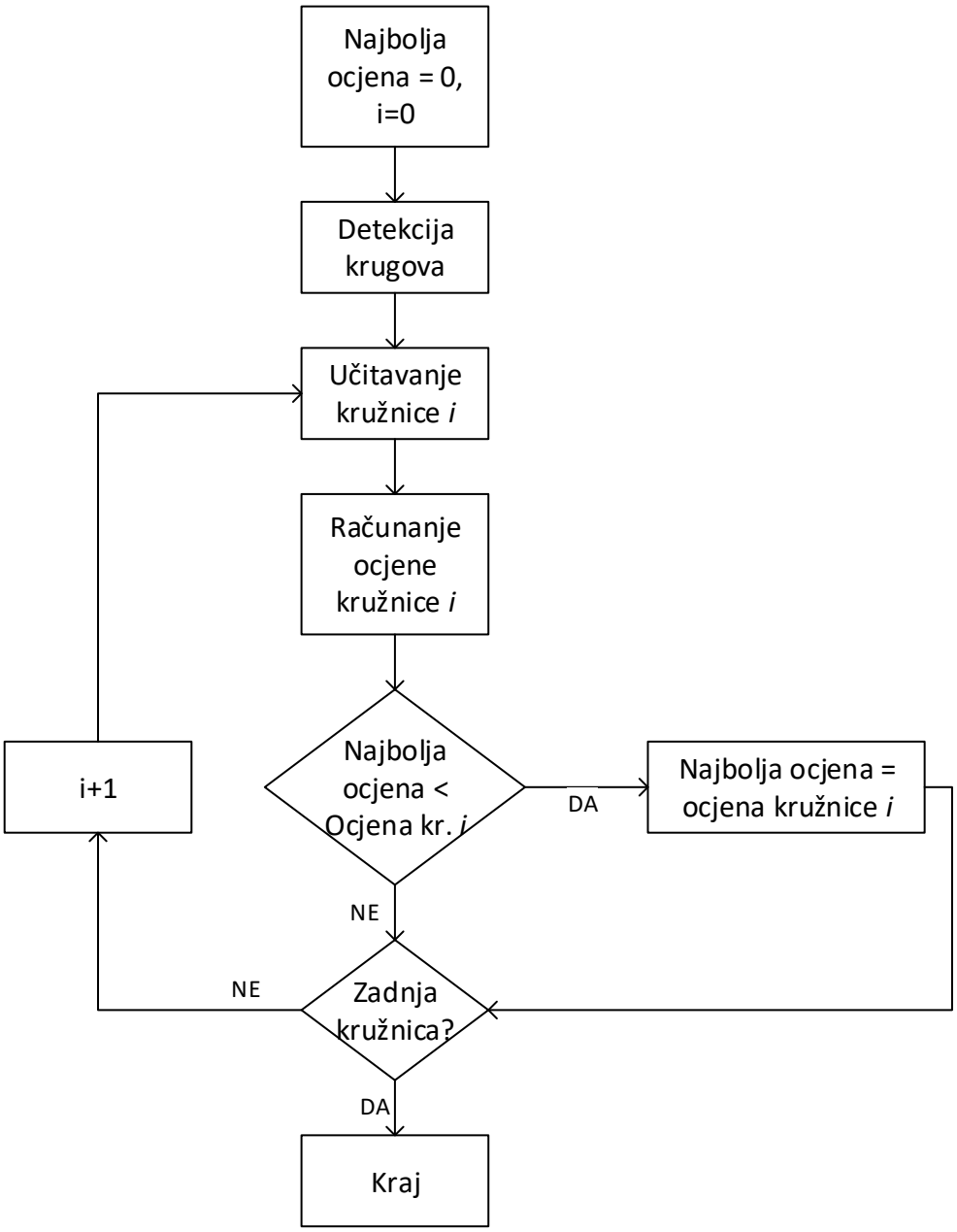
3.2.2. Lokalizacija zjenice

Bitno je razlikovati lokalizaciju od optimizacije položaja zjenice. Lokalizacija zjenice kao ulaznu sliku prima detektirane rubove cijele slike (Sl. 3.9., desno), a bira onaj krug koji najviše odgovara profilu zjenice od svih detektiranih kružnica na cijeloj slici. Postupak optimizacije kao ulaznu sliku prima samo područje oko lokalizirane zjenice (prikazano na Sl. 3.12.), te u njemu traži kružnicu koja najbolje opisuje zjenicu. Cilj lokalizacije je što brže naći položaj zjenice na slici i tako smanjiti prostor pretraživanja u postupku optimizacije, a cilj optimizacije što preciznije odrediti granice zjenice.

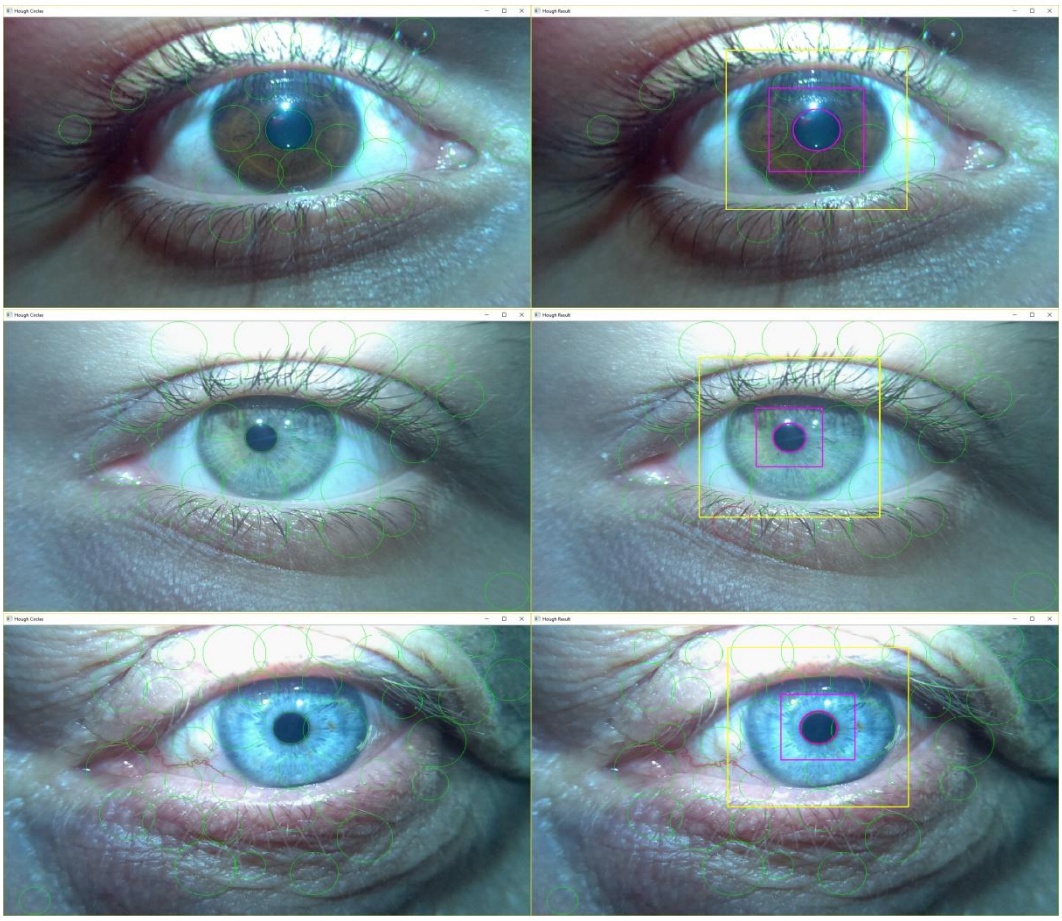
Na slici 3.11. prikazan je funkcionalni blok dijagram koji vrijedi i za postupak lokalizacije zjenice i za postupak optimizacije položaja zjenice, budući da su postupci vrlo slični. Najveća razlika je u sustavima ocjenjivanja kružnica koji su detaljnije opisani u nastavku. Oba postupka svode se na detekciju krugova Houghovom metodom te pojedinačno ocjenjivanje svakog kruga. Ona kružnica s najboljom ocjenom označava kružnicu koja najbolje opisuje područje zjenice (u postupku lokalizacije) odnosno granicu zjenice (u postupku optimizacije).

Nakon što su detektirani rubovi Cannyjevim detektorom, slijedi primjena Houghovog algoritma za detekciju krugova. Potrebno je znati očekivani polumjer tražene kružnice, u ovom slučaju zjenice. Kratkim pregledom niza fotografija uočeno je da se pri korištenoj rezoluciji, polumjeri zjenica kreću između 35 i 70 piksela originalne slike. Radi povećanja fleksibilnosti algoritma, Houghova metoda traži krugove polumjera između 30 i 80 piksela originalne slike. Dodatno, potrebno je zadati minimalnu udaljenost *minD* između središta najbližih detektiranih kružnica, prvenstveno kako bi se smanjilo područje pretraživanja metode HoughCircles. To u suštini znači da ako je detektirana kružnica sa središtem u točki S, središte sljedeće detektirane kružnice mora biti udaljeno barem *minD* piksela od točke S.

Houghova metoda stavlja središte kružnice polumjera *r* iz zadanog intervala na neku rubnu točku (dobivenu Cannyjevim detektorom rubova). Tada svakoj točki (pikselu) koja leži na hipotetskoj kružnici polumjera *r* pridodaje jedan glas u Houghovom prostoru parametara. Taj postupak se ponavlja za svaki rubni piksel. Za točke „žarišta“, tj. za one piksele kojima je dodijeljen broj glasova iznad nekog praga, pretpostavlja se da su središta kružnica.



Sl. 3.11. Funkcionalni blok dijagram lokalizacije i optimizacije zjenice

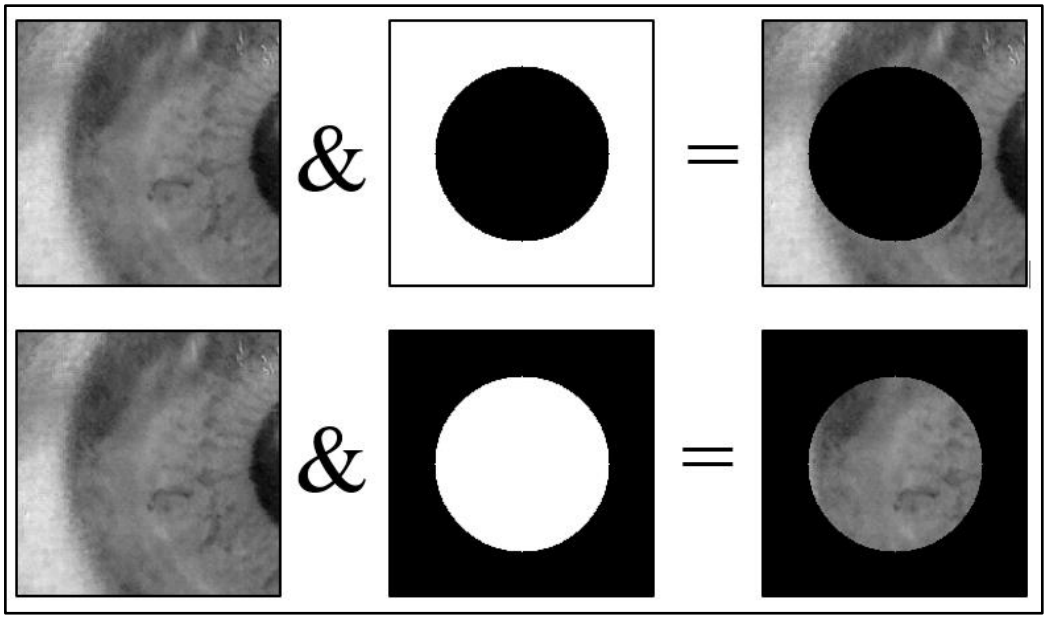


Sl. 3.12. Rezultati metode Hough Circles (lijevo) i lokalizirana zjenica (desno) za smeđe (gore), zeleno (sredina) i plavo (dolje) oko

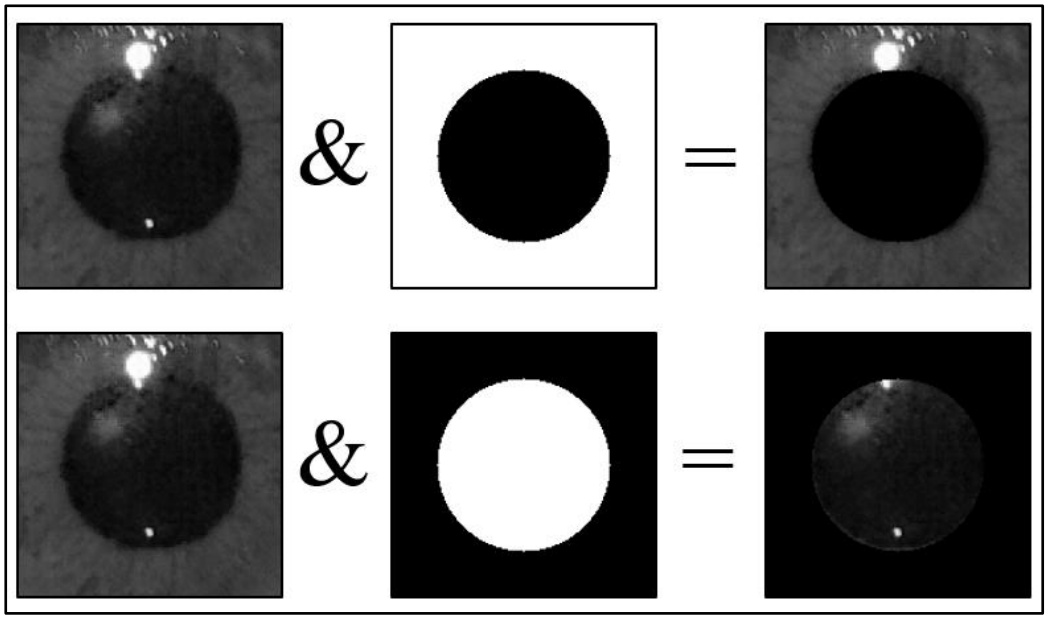
Za lokalizaciju zjenice realiziran je algoritam koji se bazira na histogramima intenziteta svjetlosti. Algoritam iterativno prolazi kroz sve kružnice $k=1,2,\ldots,n$, uzima kvadrat dimenzija $3r_k \times 3r_k$, u daljnjem kontekstu područje interesa (*engl. ROI - Region Of Interest*, područje interesa), sa sjecištem dijagonala u središtu kružnice. Pomoću binarnih maski određuju se histogrami intenziteta svjetlosti unutar i van kružnice. Ideja iza histograma intenziteta svjetlosti je sljedeća: unutar zjenice trebali bi se nalaziti izrazito tamni pikseli, uz mogući izuzetak malog broja izuzetno svijetlih zbog odsjaja osvjetljenja. U užem prostoru izvan zjenice, nužno unutar šarenice, trebali bi se nalaziti pikseli uglavnom srednjih vrijednosti intenziteta (karakterističnih za šarenicu, opet uz iznimku manjeg broja izrazito svijetlih piksela). Ocjena kružnice je veća što su pikseli unutar nje tamniji (karakteristično za zjenicu), a izvan nje bliže srednjim vrijednostima karakterističnim za šarenicu.

Prvi korak lokalizacije zjenice je dohvatiti parametre detektirane kružnice te generirati novu matricu koja sadrži područje interesa. Nakon toga, kreiraju se dvije binarne maske, obje

dimenzija jednakih području interesa. Maska za područje van kružnice je bijeli kvadrat (jedinice) s ucrtanom crnom kružnicom (nule). Maska za područje unutar kružnice je crni kvadrat (nule) s ucrtanom bijelom kružnicom (jedinice). Logičkom „I“ operacijom između područja interesa i maski dobiju se dvije nove matrice – s prepisanim vrijednostima područja interesa gdje su bile jedinice maske, odnosno nulama gdje su bile nule maske. Slika 3.13. prikazuje područje oko jedne od neispravno detektiranih kružnica koje ne odgovaraju zjenici. Na slici 3.14. prikazana je kružnica detektirana ispravno, na području zjenice. Međusobnom usporedbom područja izvan i unutar pojedine detektirane kružnice može se zaključiti kako je raspodjela intenziteta svjetline unutar i van kružnice vrlo dobar kriterij za raspoznavanje zjenice na slici.



Sl. 3.13. Područje interesa (ROI, lijevo), područje unutar "zjenice" (sredina), područje izvan "zjenice" (desno)



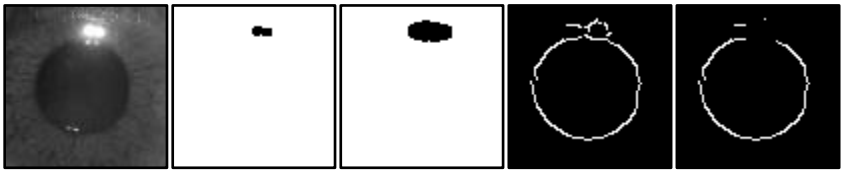
Sl. 3.14. Područje interesa (ROI, lijevo), područje unutar zjenice (sredina), područje izvan zjenice (desno)

Točnost lokalizacije zjenice je zadovoljavajuća, ali vjerojatnost preciznog opisivanja granica zjenice nije velika, jer dominantna kružnica ne mora nužno točno opisivati cijelu granicu zjenice i šarenice. Moguće je da Houghova metoda kružnicu stavi na dobar lijevi ili desni rub zjenice, no premalog ili prevelikog polumjera. Takav slučaj se i dalje smatra uspjehom lokalizacije jer je uspješno izdvojeno uže područje oko zjenice. Upravo radi povećanja preciznosti opisivanja granica provodi se dodatan proces optimizacije položaja zjenice koji promatra samo područje označeno ljubičastim (unutarnjim) kvadratom na slici 3.12. odnosno ROI na slici 3.14., te ponavlja postupak detekcije krugova s nižim kriterijima za detekciju.

3.2.3. Optimizacija položaja zjenice

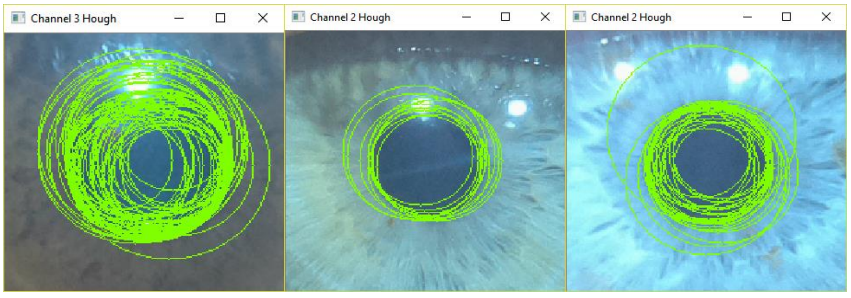
Prvotni plan za optimizaciju granica zjenice bio je zasebno promatrati sva tri kanala RGB prostora boja, pod pretpostavkom da će za plave oči granica biti najvidljivija u plavom, za zelene u zelenom, a smeđe u crvenom kanalu. Razmatranjem pojedinih kanala, utvrđeno je da čak niti kod plavih očiju razlika između „vidljivosti“ ruba zjenice u zelenom i plavom kanalu nije velika. Iz tog razloga se u postupku optimizacije područje oko zjenice promatra u dva kanala – zelenom i crvenom kanalu RGB prostora boja. Sljedeći koraci su vrlo slični onima prikazanim blok dijagramom na slici 3.10., a rezultiraju dvjema binarnim slikama rubova (po jedna za svaki promatrani kanal), koje se koriste kao ulazni parametri metode *HoughCircles*. Kako bi se umanjio

utjecaj refleksije osvjetljenja koje će detektori rubova nužno deketirati jer se radi o naglom prijelazu svjetline, implementirana je detekcija regije osvjetljenja metodom praga. Metoda praga rezultira binarnom slikom gdje su crnim pikselima označene ekstremne vrijednosti očekivane za refleksiju osvjetljenja. Erozijom bijele boje (kao na primjeru na sl. 3.6.) to se područje dodatno poveća i logičkom „I“ operacijom sa slikom rubova rezultira rubovima bez rubova osvjetljenja. Cijeli postupak uklanjanja utjecaja refleksije prikazan je na slici 3.15.



Sl. 3.15. Uklanjanje refleksije: područje zjenice; ekstremne vrijednosti; prošireni ekstremi; originalni rubovi; filtrirani rubovi

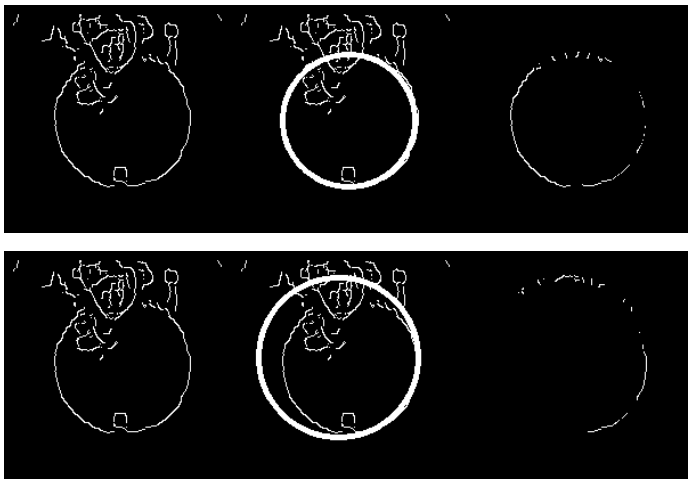
Odabirom dva umjesto jednog kanala uvelike se povećava fleksibilnost algoritma – jedan kanal daje bolje rezultate za svjetlije slike, a drugi za tamnije. Parametri za funkciju *HoughCircles* su nešto drukčije postavljeni nego u slučaju lokalizacije – središta detektiranih kružnica mogu biti udaljena samo 1 piksel i kriteriji za detekciju su znatno niži. Time se znatno povećava broj detektiranih kružnica, a samim time i vjerojatnost da će jedna od detektiranih kružnica precizno opisivati rub zjenice (Sl. 3.16).



Sl. 3.16. Detektirane kružnice za smeđe (lijevo), zeleno (sredina) i plavo (desno) oko uz znatno niže kriterije detekecije u postupku optimizacije položaja zjenice

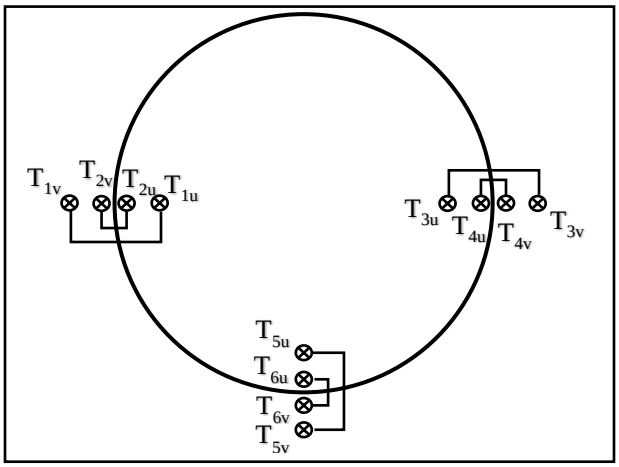
Svakoj detektiranoj kružnici algoritam dodjeljuje ocjenu prema kojoj se određuje najbolja granica zjenice i šarenice. Histogramom binarne slike rezultata detekcije rubova odredi se broj bijelih piksela, a za svaku detektiranu kružnicu generira se maska – crna pozadina s ucrtanom

bijelom kružnicom. Između te dvije slike provodi se logička „I“ operacija – a rezultat su rubovi koji leže u neposrednoj blizini polumjera detektirane kružnice. Određuje se broj preostalih bijelih piksela, a kao ocjena kružnice upisuje se omjer broja preostalih i originalnih bijelih piksela (prikazano na Sl. 3.17.).



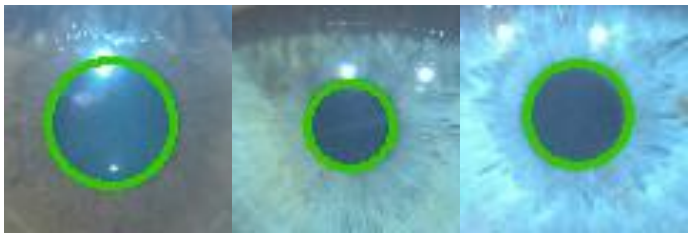
Sl. 3.17. Vrednovanje kružnice podudaranjem s rubovima: vjerojatniju granicu zjenice opisuje gornja kružnica.

Kao dodatan parametar pri ocjenjivanju kružnice kao zjenice uzima se gradijent svjetline piksela oko rubova zjenice. Pikseli izvan kružnice bi u slučaju zjenice trebali biti svjetliji od onih unutar kružnice. Optimizacija granica zjenice provedena je sa šest parova točaka odnosno šest razlika intenziteta svjetline. Odabrane su točke kod krajnjeg lijevog i krajnjeg desnog ruba kružnice. Promatrana je razlika intenziteta svjetline piksela na dvije različite udaljenosti točaka od ruba kružnice. Izračun ocjene prema ilustraciji na Sl. 3.18. dan je u izrazu (3-3).



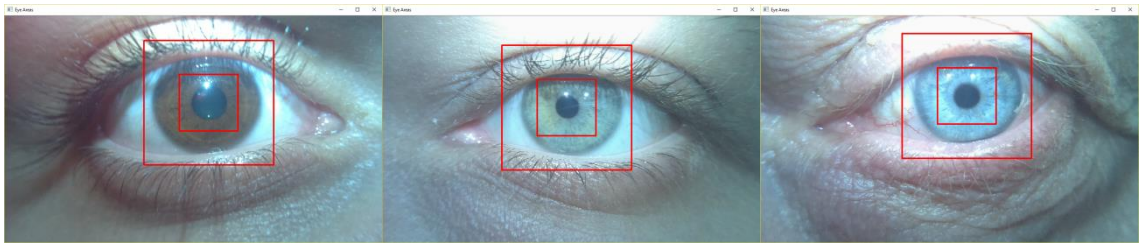
Sl. 3.18. Točke korištene pri ocjenjivanju zjenice

$$Ocjena = \prod_{i=1}^{6}(Intenzitet_{T_{iv}} - Intenzitet_{T_{iu}}) \quad (3\text{-}3)$$

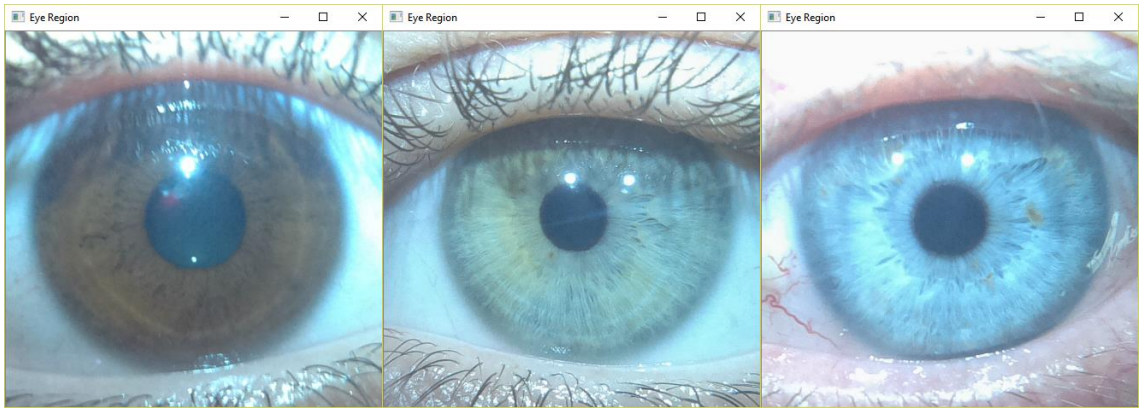


Sl. 3.19. Optimizirani položaj zjenice

Nakon optimiziranja položaja zjenice (Sl. 3.19.) određuju se nove granice područja oko središta zjenice u kojem će se tražiti vanjski rub šarenice (Sl. 3.20., izdvojena na Sl. 3.21.).



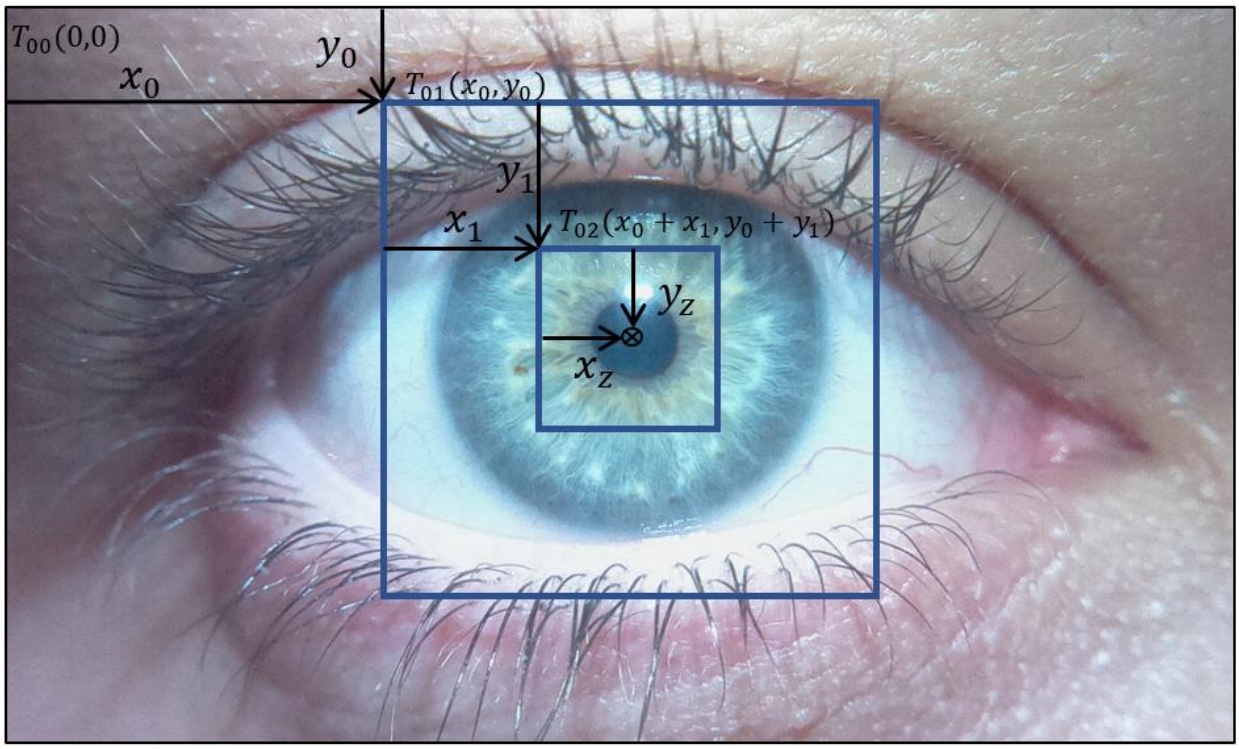
Sl. 3.20. Podešena područja interesa prema optimiziranom položaju zjenice



Sl. 3.21. Područja interesa originalnih slika korištena u daljnjoj obradi, tj. lokalizaciji šarenice

Pri daljnjoj obradi korišteno je samo izdvojeno područje interesa (Sl. 3.21., u daljnjem kontekstu područje interesa šarenice) prvenstveno radi smanjivanja područja pretraživanja

odnosno podataka za obradu. Veličina tog područja nije fiksna, nego se određuje prema razlučivosti originalne slike. Prikazano područje interesa generira se tako da se središte detektirane zjenice nalazi na sjecištu dijagonala područja interesa. Takav način olakšava daljnju obradu, prvenstveno uzorkovanje šarenice opisano u odjeljku 3.2.5. Nužan korak pri odvajanju na manje područje interesa je prebacivanje središta optimizirane zjenice iz koordinata s obzirom na ROI zjenice (Sl. 3.19.) u koordinate s obzirom na ROI cijelog oka (Sl. 3.21). Princip transformacije koordinata središta prikazan je na slici 3.22. Točke T_{0x} označavaju lokalna ishodišta koordinatnih sustava. Konačna rješenja lokalizacije transformirana su i korištena s obzirom na područje interesa oka odnosno točku T_{01}.

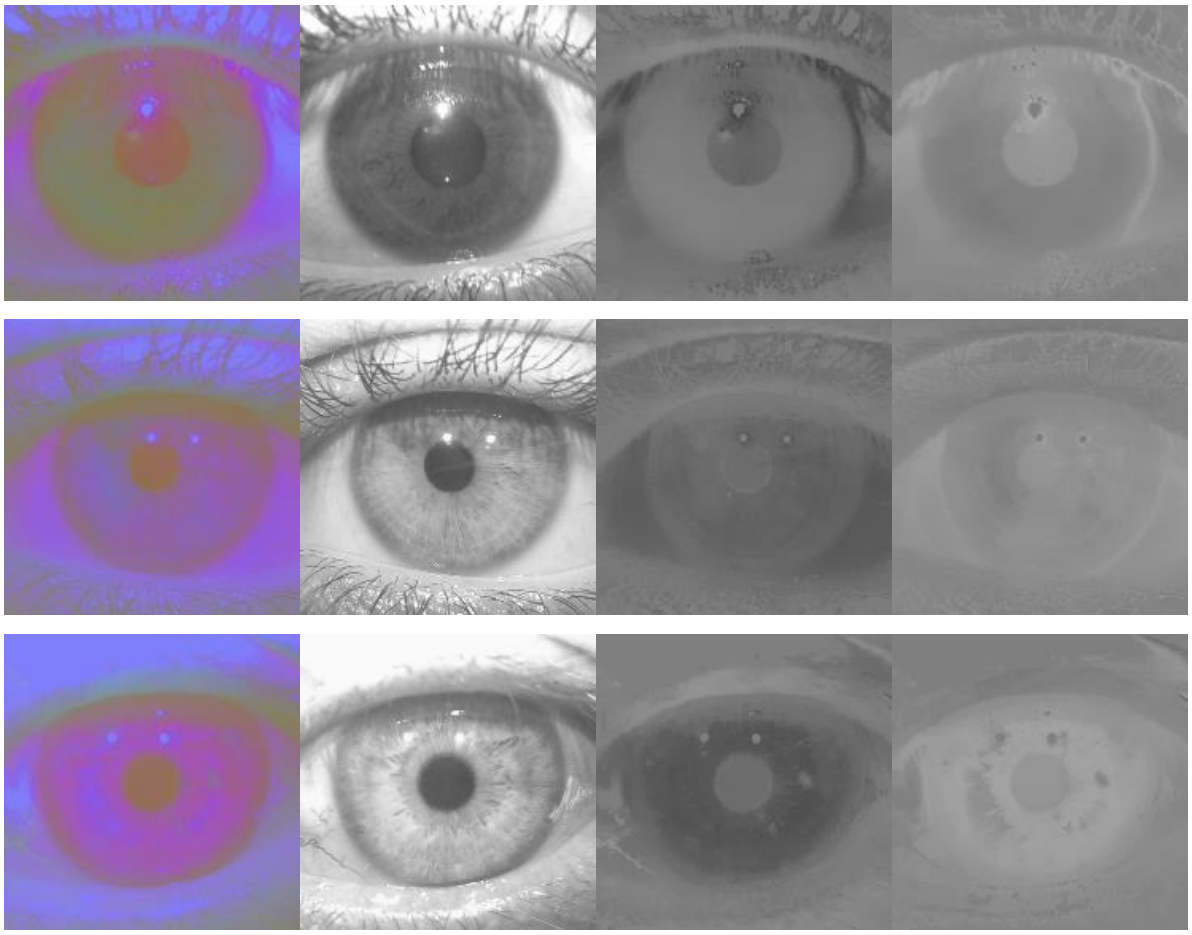


Sl. 3.22. Pregled koordinata potrebnih pri transformaciji središta iz jednog područja interesa u drugi

3.2.4. Lokalizacija šarenice

Za lokalizaciju šarenice korišten je isti princip kao i dosad – obrada područja interesa šarenice s ciljem dobivanja što kvalitetnije binarne slike rubova, detekcija krugova (Hough) s nižim kriterijima i u konačnici odabir najbolje šarenice po određenom kriteriju.

Istraživanjem različitih prostora boja (RGB, HSV, HLS, YCrCb, XYZ, LAB, LUV) i primjenom algoritma za lokalizaciju šarenice na niz različitih kanala, odabran je prvi kanal YCrCb prostora boja. Prvi kanal, poznatiji kao *luma*, predstavlja akromatsku komponentu osvijetljenosti, dok se kromatske informacije (o plavoj i crvenoj boji) nalaze u preostala dva kanala (Sl. 3.23.).



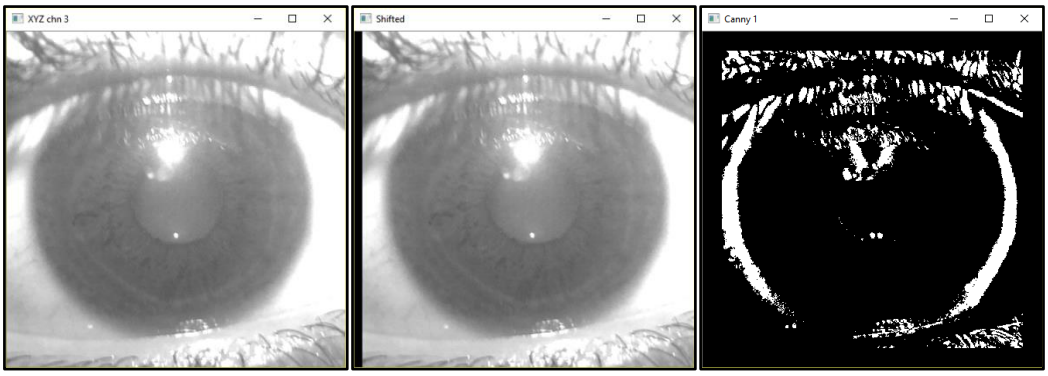
Sl. 3.23. Područje interesa šarenice u YCrCb kanalu, te pojedinačno u Y, Cr, Cb za smeđe (gore), zeleno (sredina) i plavo (gore) oko

Zbog očekivanja kako će lijevi i desni rubovi šarenice biti vidljiviji od gornjih i donjih (jer će gornji odnosno donji rub biti prikriven gornjim odnosno donjim kapkom), veća je vjerojatnost da će kružnica dobro opisivati rub šarenice ako je detektirana na bočnim, izraženijim rubovima. Iz tog razloga je primijenjen jednostavan algoritam za detekciju promjene intenziteta svjetlosti u horizontalnom smjeru. Početna slika, tj. *Y* kanal *YCrCb* prostora boja se kopira u pomoćnu matricu i pomiče (engl. *shift*) za *n* piksela u desno. Potom se traže razlike između te dvije slike (Sl. 3.24.) prema izrazu (3-4):

$$Sobel_{horiz} = abs(Y - Y_{pd}) > prag, \tag{3-4}$$

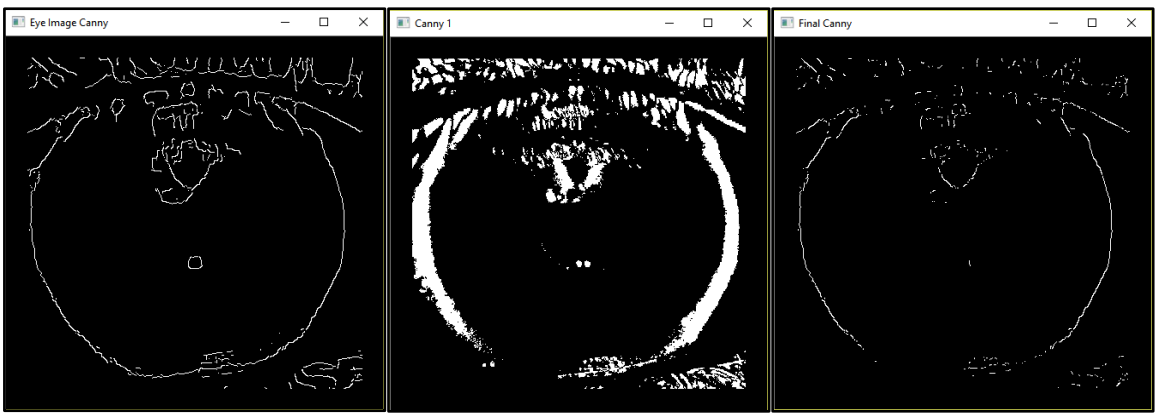
gdje je Y matrica prvog kanala slike područja oka u YCrCb prostoru boja, a Y_{pd} matrica Y pomaknuta u desno za određeni korak k, a može se izračunati pomoću izraza (3-5).

$$Y_{pd_{i,j}} = \begin{cases} Y_{i,j-k}, & za\ j \geq k \\ 0, & za\ j < k \end{cases} \tag{3-5}$$



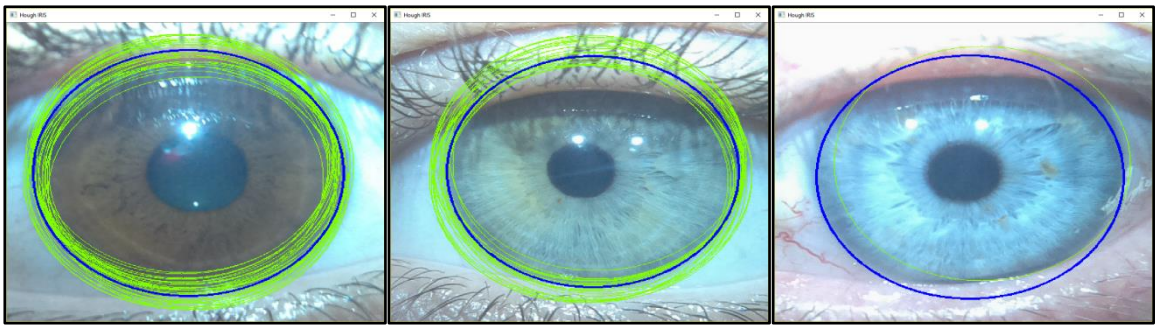
Sl. 3.24. Originalna slika (lijevo), Pomaknuta (sredina), rezultat po izrazu (3-4) (desno)

Rubovi se također detektiraju i Cannyjevim detektorom. Konačna slika koja se prosljeđuje metodi *Hough Circles* dobije se logičkom „I“ operacijom između Cannyjevih i horizontalnih rubova. Ta slika zadržava bočne rubove šarenice, a u isto vrijeme otklanja dio šuma u podacima (rubovi trepavica, kapaka; Sl. 3.25.).



Sl. 3.25. Cannyjevi rubovi (lijevo), Sobel-ovi rubovi (sredina), rezultat „I“ operacije između rubova

Zbog niskih kriterija za detekciju krugova, *Hough Circles* uglavnom detektira više od 5 kružnica koje potencijalno opisuju vanjski rub šarenice. Što manje šuma sadrže detektirani rubovi, to je manji broj kružnica koje će biti detektirane (Sl. 3.26).



Sl. 3.26. Rezultat metode Hough Circles za smeđe (lijevo), zeleno (sredina) i plavo (desno) oko

Kriterij za kružnicu koja najbolje opisuje rub šarenice od ukupno n_{iris} detektiranih krugova sa središtima u točkama $S_{iris}(x_{iris},y_{iris})$ te polumjera r_i sastavljen je od dva faktora, a prvi je primijenjen i u postupku optimizacije granica zjenice – razlika između intenziteta svjetline piksela neposredno van kružnice i neposredno unutar je, odnosno na točkama za *korak* piksela udaljenih od ruba kružnice. Ta razlika promatra se na nekoliko parova točaka (T_{i1}, T_{i2}):

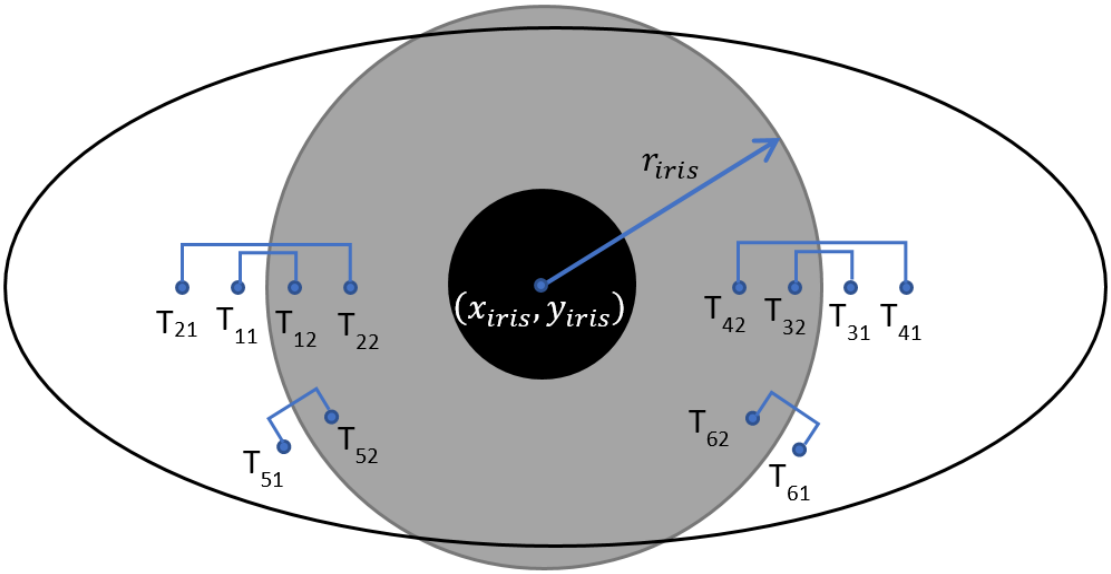
- $T_{i1}(x - r - korak, y)$, $T_{i2}(x - r + korak, y)$

Za četiri različite udaljenosti (koraka) svakog para od ruba kružnice, a ukupna ocjena kružnice računa se prema izrazu (3-6).

$$O = \sum_{k=1}^{6}\left(I(T_{k1}) - I(T_{k2})\right), \tag{3-6}$$

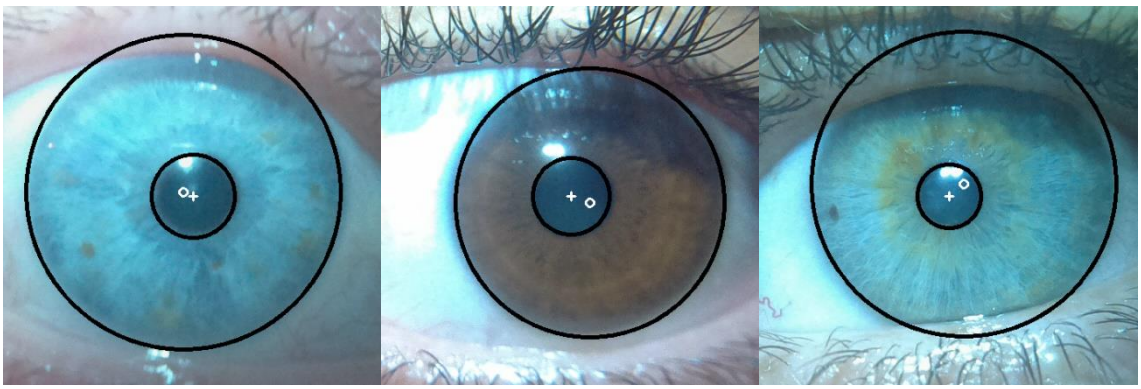
gdje je O ocjena kružnice, T_{kx} jedna od točaka a $I(T_{kx})$ intenzitet svjetline slike u točki T_{kx}.

Ovaj postupak se ponavlja i za područje šarenice zakrenuto za 30° i 150°. Te dodatne točke označene su kao $T_{6(1/2)}$ odnosno $T_{5(1/2)}$, kao što je prikazano na slici 3.27. Točke i kutovi su odabrani iz područja s najvećom vjerojatnošću sadržavanja granice između bjeloočnice i šarenice. Krajnji gornji odnosno donji rub šarenice će najčešće biti prekriven gornjim odnosno donjim kapkom.



Sl. 3.27. Ilustracija odabira točaka za ocjenjivanje šarenice

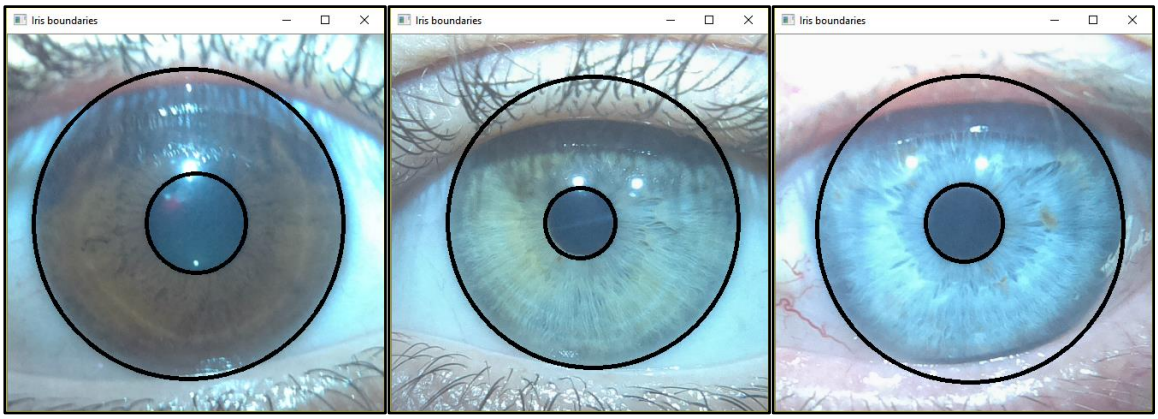
Drugi faktor u konačnoj ocjeni šarenice je već primijenjen u postupku optimizacije granica zjenice. Generira se binarna maska s jedinicama na rubu detektirane kružnice, nakon čega se provodi logička „I“ operacija između maske i binarne slike rubova. Na rezultantnoj slici ostaju samo podudarni pikseli (tj. oni pikseli rubova koji leže na kružnici), a ocjena kružnice je veća što je broj takvih piksela veći. Na slici 3.28. prikazane su kružnice koje najbolje opisuju zjenicu odnosno šarenicu. Središte zjenice označeno je znakom plus, a središte šarenice kružićem. Vidljivo je kako vanjsku i unutarnju granicu šarenice najčešće ne opisuju dvije koncentrične kružnice. Sljedeće poglavlje objašnjava zašto je to bitno.



Sl. 3.28. Detektirani rubovi šarenice s označenim središtima kružnica

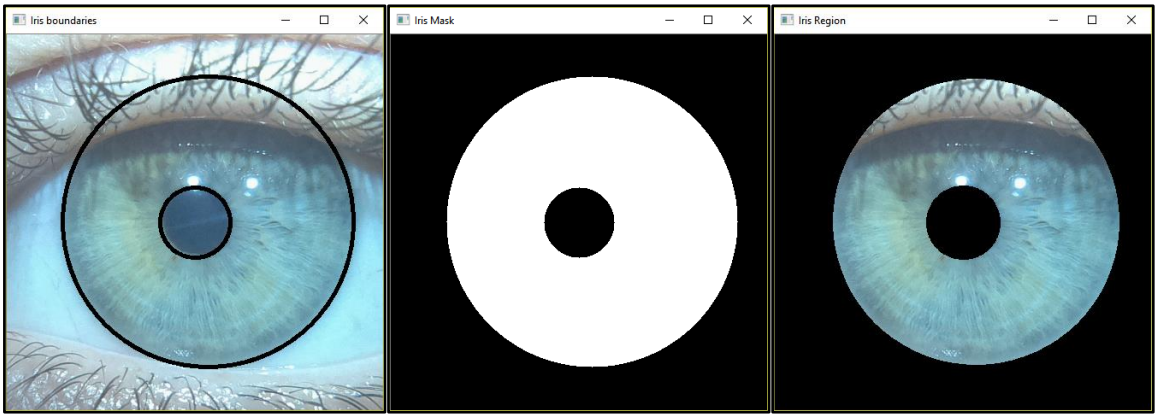
3.2.5. Uzorkovanje šarenice

Nakon što su određene kružnice koje označavaju unutarnju i vanjsku granicu šarenice, potrebno je transformirati podatke između njih, koji su radijalno raspoređeni, u oblik pogodniji za računalnu analizu – pravokutnu matricu. Potrebno je u obzir uzeti regije šarenice koje bi trebale biti vidljive u većini fotografija. Osvjetljenje je pri fotografiranju postavljeno iznad optičke osi kamere, pa se odsjaj osvjetljenja na slici nalazi u gornjoj polovini šarenice. Dodatno, gornji kapak u većini slučajeva prekriva veću površinu šarenice nego donji kapak. Programsko rješenje je realizirano tako da se sjecište dijagonala područja interesa šarenice nalazi u središtu kružnice koja opisuje granicu zjenice polumjera $r_{zjenice}$ (Sl. 3.28, 3.29). Pošto se uzorkovanje šarenice zasniva se na iterativnoj rotaciji slike oko središta, a posljedica rotacije je promjena koordinata svih piksela slike osim središnjeg. Ukoliko se središte zjenice nalazi na središtu slike, ono se rotacijom neće pomicati pa se granice uzorkovanja šarenice sa strane zjenice neće mijenjati, odnosno neće ih biti potrebno računati za svaki korak rotacije.



Sl. 3.29. Unutarnji i vanjski rubovi šarenice

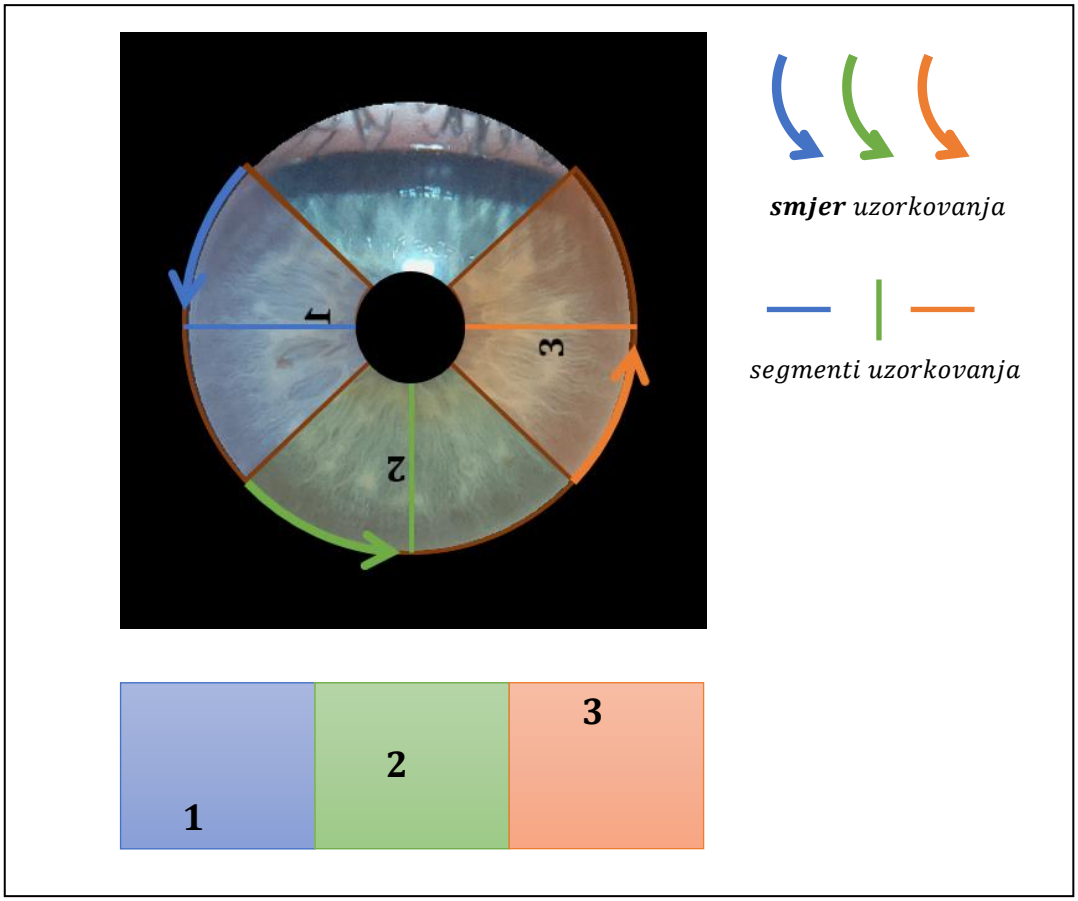
U postupku uzorkovanja šarenice vrijednosti piksela izvan granica šarenice nisu bitne i prepisuju se nulama. Za to je ponovno korišten princip binarne maske i logičkog operatora „I“. Na crnu sliku (nule) veličine područja interesa šarenice ucrta se detektirani krug šarenice bijelom bojom (jedinicama), na koju se potom nulama ucrta crni krug zjenice (Sl. 3.30.).



Sl. 3.30. Izdvajanje područja šarenice

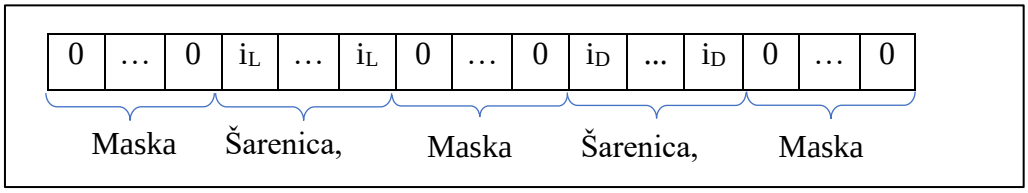
Cilj uzorkovanja šarenice je iz radijalno raspršenih podataka u matrici demonstriranoj na slici 3.30. (desno) dobiti pravokutnu matricu. Prema [1], šarenica se uzorkuje obrnuto od smjera kazaljke na satu, od unutarnjeg prema vanjskom rubu šarenice.

Pošto je iz matrice puno jednostavnije čitati redove i stupce nego pravce s nagibom različitim od 0° i 90°, princip uzorkovanja sveden je na iterativno zakretanje slike šarenice oko središta zjenice i uzorkovanje srednjeg reda i srednjeg stupca. Uzorkovanje srednjeg reda odnosno stupca označava dohvaćanje, skaliranje i pohranu onih podataka u srednjem redu odnosno stupcu područja interesa šarenice koji su isključivo unutar granica šarenice. Pošto svaki srednji red sadrži podatke šarenice s lijeve i desne strane zjenice (2 segmenta), a stupac s gornje i donje (2 segmenta), moguće je uzorkovati svaki od tih segmenata šarenice pojedinačno te ih nizati u četiri zasebne matrice. Odmak između segmenata je 90°, što znači da se rotacijom slike 90 puta s korakom od 1° pokriva cijelo područje šarenice, odnosno $90 * \frac{1°}{\text{segment}} * 4\ segmenta = 360°$. Pošto se refleksija osvjetljenja na fotografijama slikanim za potrebe ovog rada nalazi u gornjoj četvrtini područja šarenice, uzorkovana su tri segmenta po koraku (Sl. 3.31.).



Sl. 3.31. Ilustracija područja i smjera uzorkovanja šarenice (gore) te trostrukog rezultata uzorkovanja (dolje)

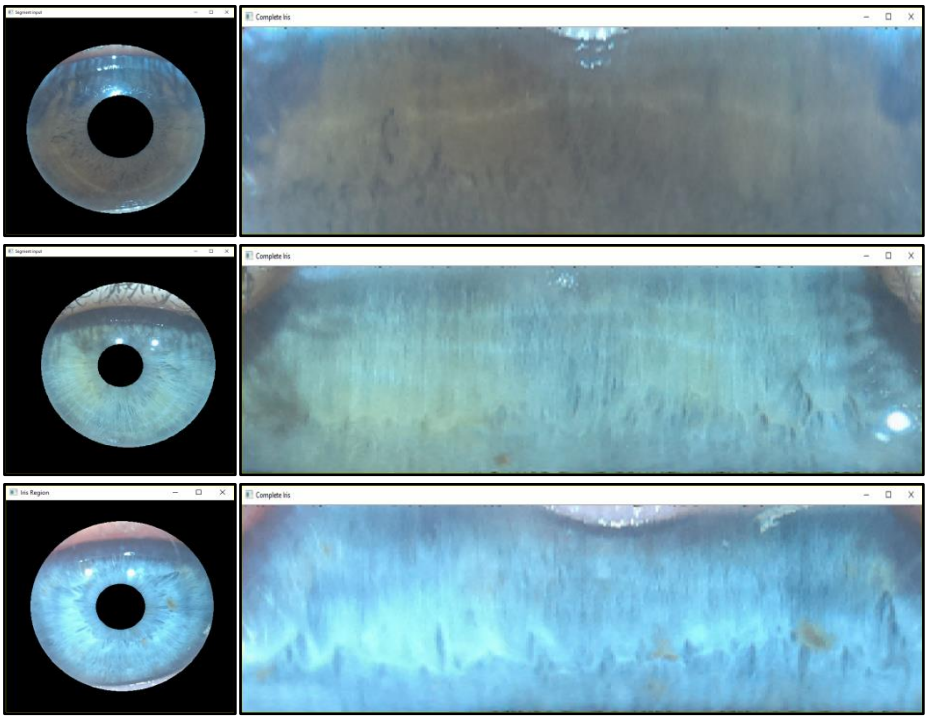
Algoritam za uzorkovanje šarenice iterativno zakreće sliku suprotno smjera kazaljke na satu za korak $k_u = 1°$ i pritom uzorkuje srednji red i srednji stupac, čija je struktura prikazana na slici 3.32.



Sl. 3.32. Struktura uzorkovanog srednjeg reda i rotiranog srednjeg stupca

Lijeva strana šarenice uzorkuje se od prvog polja s vrijednošću većom od 0 do polja s indeksom $\left(\frac{broj_{stupaca}}{2} - r_{zjenice}\right)$. Desna strana (segment) šarenice uzorkuje se od polja s

indeksom $\left(\frac{broj_{stupaca}}{2} + r_{zjenice}\right)$ do zadnjeg polja s vrijednošću većom od 0. Duljina lijevog i desnog segmenta šarenice (u pikselima) neće biti jednaka jer šarenica i zjenica nisu koncentrične. Iz tog razloga se svaki uzorkovani red linearnom interpolacijom dovodi na unaprijed određenu duljinu l. Prema tome, segmentirana šarenica bit će dimenzija $\frac{270°}{k_u} \times l$. Svakom iteracijom tj. svakim zakretanjem slike uzorkuju se tri segmenta šarenice (redovi lijevo i desno od zjenice te stupac ispod). Nakon zakretanja slike za raspon od 90° korakom 1°, rezultat čine tri matrice dimenzija $90 \times 100px$. Te matrice se horizontalno nižu jedna do druge (ilustrirano na Sl. 3.31., dolje) čime je dobivena konačna rezultantna matrica uzorkovane šarenice dimenzija $\frac{270°}{1°} \times 200px = 270 \times 100px$ (Sl. 3.33.).

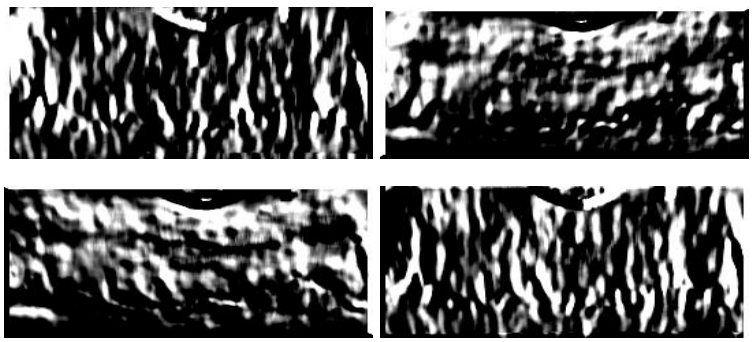


Sl. 3.33. Područje interesa šarenice (lijevo) i rezultati uzorkovanja (desno) prema ilustraciji na slici 3.31.

3.2.6. Generiranje binarnog koda

Jedinstveni kod, odnosno uzorak šarenice koji će se uspoređivati radi identifikacije mora biti dovoljno visoke rezolucije kako bi sadržavao dovoljno informacije za identifikaciju. U isto vrijeme mora biti dovoljno male veličine u memoriji – pošto se svaka šarenica mora uspoređivati sa svima u bazi dok se eventualno ne nađe odgovarajući uzorak. Prema tome, vrijeme identifikacije uvelike ovisi o broju i veličini uzoraka u bazi.

Iz tog razloga generira se binarni kod – matrica ispunjena nulama i jedinicama. Proučavanjem literature [1] [2] [11] [17], utvrđeno je kako je 2D adaptacija Gaborovog filtra, koju je realizirao upravo prof. John Daugman, najraširenij i najpouzdaniji oblik izvlačenja karakterističnih informacija iz uzorka šarenice. Uzorak šarenice se filtrira nizom 2D Gaborovih filtara, realiziranih pomoću *openCV* funkcija *getGaborKernel* i *filter2D*. Radi se o orijentiranom filtru, tj. jedan od parametara mu je smjer „djelovanja“. Taj parametar, kut *theta* označava smjer u kojem se filtriranje izvodi tj. smjer u kojem se traže tranzicije. U ovom radu uzorak šarenice se filtrira u 12 smjerova (od 0° do 165°, s koracima od 15°), a rezultati nekih dani su na slici 3.34. Prije samog filtriranja, generirana je akumulacijska matrica dimenzija uzorka šarenice, inicijalno popunjena nulama.



Sl. 3.34. Rezultati filtriranja uzorka šarenice Gaborovim filtrom za kutove: Gore: 0°i 45°; Dolje: 90° i 135°

Rezultat svakog filtriranja se zbrajanjem dodaje u akumulacijsku matricu koja se na kraju normalizira u raspon od 0 do 255 (Sl. 3.35.). Prof. Daugman je predložio i primijenio filtriranje 2D Gaborovim filtrom, no kodiranje rezultata filtriranja je zaseban problem. Za ovaj rad razmatrana su rješenja u obliku histograma intenziteta svjetline. Prijedlog je sliku rastaviti na nekoliko manjih kvadrata, te uspoređivati histograme intenziteta svjetline tih kvadrata. Druga je mogućnost histogram orijentiranih gradijenata intenziteta svjetline i taj bi pristup vjerojatno dao

najbolje rezultate jer takav način prezentacije podataka daje informaciju i o smjeru i o intenzitetu gradijenta svjetline u okolini pojedinog piksela. Međutim, zbog jednostavnosti je odabrana treća opcija – jednostavna segmentacija slike (akumulacijske matrice) binarnom metodom praga. Izračun akumulacijske matrice obavlja se prema izrazima (3-6) do (3-8), gdje *A* predstavlja akumulacijsku matricu, $gabor_{i*\frac{180°}{n}}$ rezultat filtriranja uzorka šarenice Gaborovim filtrom za određeni kut, *i* smjer filtriranja (od 1 do 12), a *n* je ukupan broj smjerova filtriranja odnosno 12.

$$A = \sum_{0}^{n-1} gabor_{i*\frac{180°}{n}} \tag{3-6}$$

$$A = rez - \min(rez) \tag{3-7}$$

$$A = rez * \left(\frac{255}{\max(rez)}\right) \tag{3-8}$$



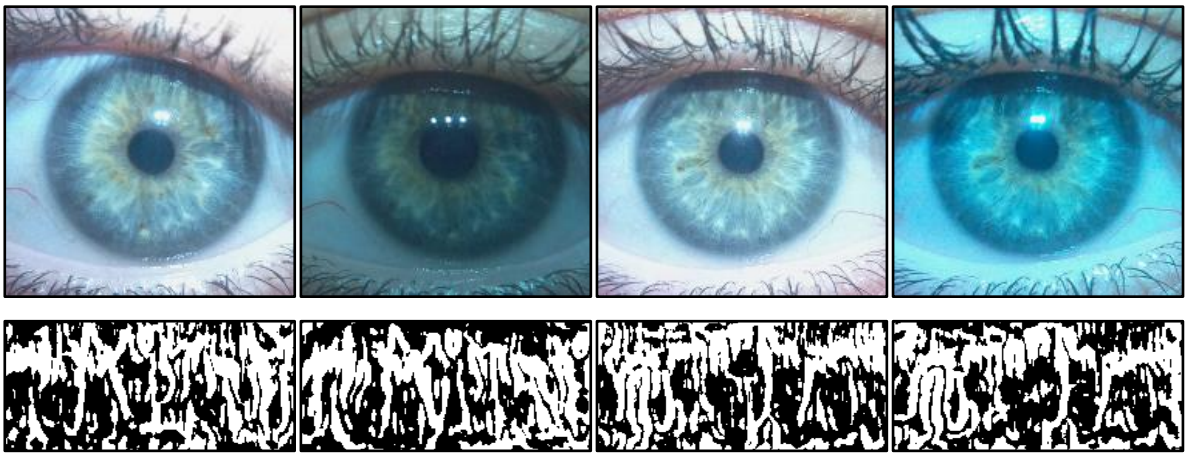
Sl. 3.35. Akumulacijska matrica

Segmentacija piksela akumulacijske matrice binarnom metodom praga odvaja piksele na one s vrijednostima iznad i ispod nekog praga (engl. *threshold*). Ako bi taj prag bio fiksna vrijednost, algoritam bi bio izuzetno osjetljiv na osvjetljenje oka. Iz tog razloga se kao vrijednost praga koristi medijan akumulacijske matrice. Tako bi rezultantna matrica uvijek trebala imati oko 50% crnih i 50% bijelih piksela. A ako se radi o uzorcima istog oka, pretpostavka je da će iste regije biti crne odnosno bijele. Nakon što je akumulacijska matrica segmentirana uz vrijednost praga jednakoj medijan vrijednosti matrice, generiran je binarni kod uzorka šarenice.



Sl. 3.36. Konačan binarni kod uzorka šarenice

Na slici 3.37. prikazane su dvije fotografije lijevog i dvije fotografije desnog oka iste osobe, slikane u različita vremena i s različitim parametrima kamere, zajedno sa binarnim kodom generiranim za pojedinu sliku. Kratkim pregledom moguće je zaključiti kako binarni kodovi istog oka, bez obzira na razliku u boji, znatno više odgovaraju međusobno, nego u usporedbi sa kodom drugog oka.

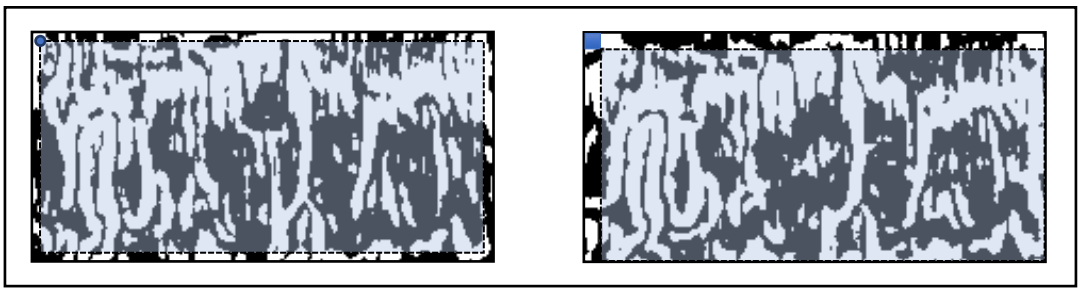


Sl. 3.37. Dvije fotografije lijevog oka (lijevo) i desnog oka (desno) iste osobe, zajedno sa generiranim binarnim kodovima

3.2.7. Uspoređivanje binarnih kodova

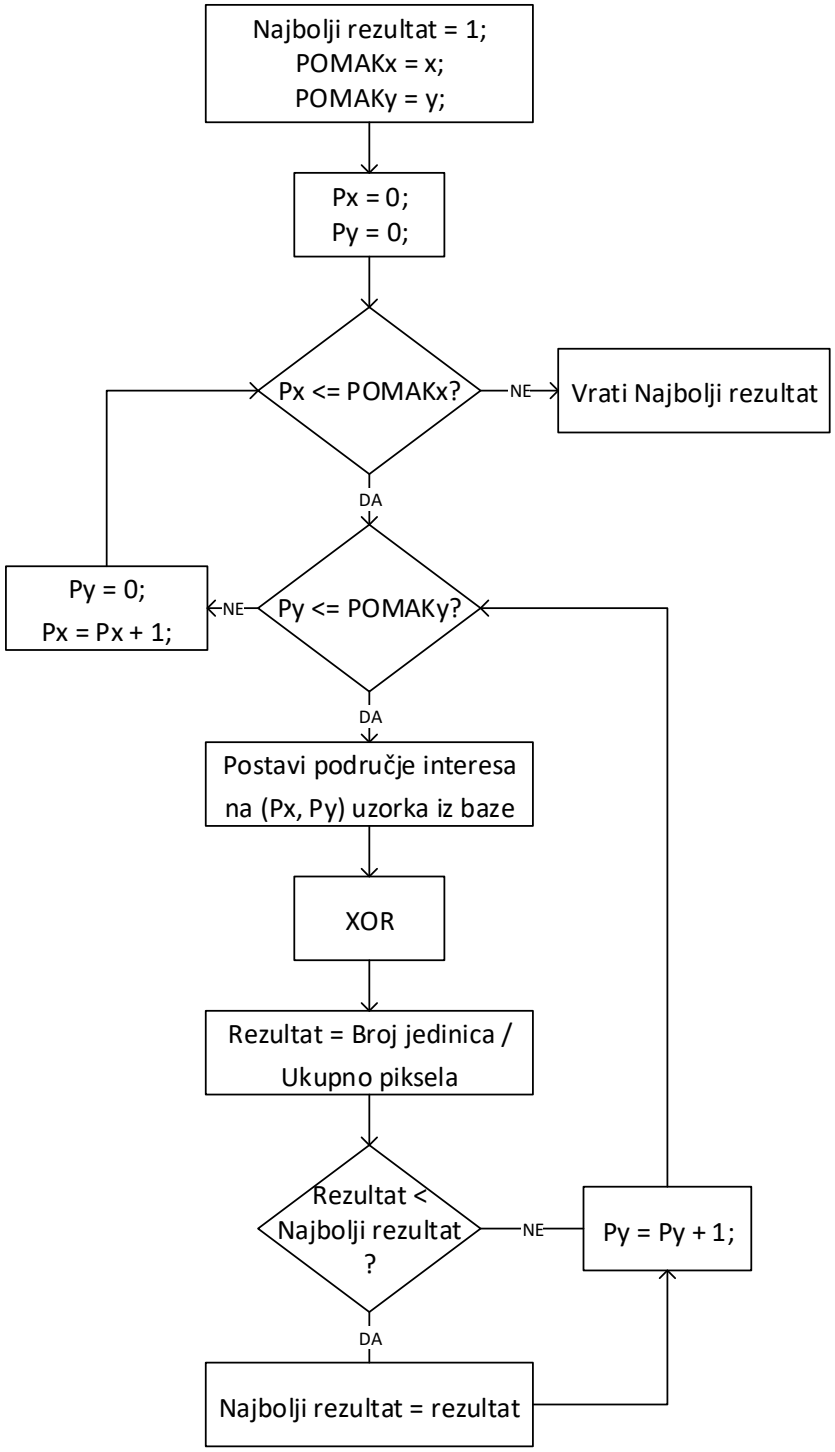
Uspoređivanje binarnih matrica (kodova) zasniva se na računanju Hamming-ove udaljenosti dvije binarne matrice. Hamming-ovu udaljenost predstavlja broj različitih piksela dvije binarne slike istih dimenzija. Računa se jednostavnom isključivo-ILI (XOR) logičkom operacijom između dvije binarne matrice. Rezultantna matrica sadrži jedinice gdje su pikseli dva uzorka različiti, a nule gdje su isti. Histogramom rezultantne binarne matrice izrazito brzo se mogu prebrojati sve nule i jedinice rezultantne matrice. Hammingova udaljenost predstavlja omjer broja bijelih i ukupnog broja piksela rezultantne matrice. Ukoliko je Hamming-ova udaljenost između uzoraka ispod određenog praga, može se smatrati kako se radi o uzorcima istog oka iste osobe.

Kako bi se smanjio utjecaj neprecizne lokalizacije ili rotacije samog oka na slici, novi uzorak se pomiče po uzorku iz baze za određeni niz koraka u svim smjerovima, a za svaki od međusobnih položaja računa Hammingova udaljenost. Dva uzorka se uspoređuju tako da se od jednog uzme područje interesa koje obuhvaća barem 80% binarnog koda. Odabrano područje se iterativno uspoređuje sa „pomičnim“ područjem interesa drugog uzorka. Za svaki pomak računa se Hamming-ova udaljenost, a na kraju izdvaja najmanja. Primjer takvog područja interesa dan je na slici 3.38., gdje desni kod predstavlja kod korisnika iz baze, a lijevi je generiran prilikom pokušaja pristupa. Osjenčana područja predstavljaju područja interesa koja se uspoređuju. Na lijevoj slici to je statično područje interesa novog uzorka, a na desnoj područje uzorka iz baze koje se iterativno pomiče. Ishodište područja interesa uzorka iz baze se pomiče piksel po piksel od gornjeg lijevog do donjeg desnog kuta plavog kvadrata označenog u gornjem lijevom kutu uzorka iz baze. Osjenčano područje interesa uzorka iz baze predstavlja zadnju iteraciju pomaka.



Sl. 3.38. Princip uspoređivanja binarnih kodova (lijevo: novi uzorak; desno: uzorak iz baze)

Slika 3.39. prikazuje funkcionalni blok dijagram računanja najmanje Hammingove udaljenosti između dva binarna uzorka. Pretpostavlja se da je područje interesa novog uzorka već određeno, a područje interesa uzorka iz baze postavlja se iterativno, pomicanjem ishodišta pri svakoj iteraciji.



Sl. 3.39. Funkcionalni blok dijagram računanja Hammingove udaljenosti

3.3. Kontrola prava pristupa

Kontrola prava pristupa, kao programsko rješenje, objašnjena je kroz dvije cjeline:

1. Generiranje i struktura baze podataka i
2. Provjera prava pristupa novog uzorka.

3.3.1. Baza podataka

Nakon realizacije algoritma za generiranje i uspoređivanje binarnog koda, generirana je SQL baza podataka sa tri tablice. Prva tablica sadrži opće informacije o korisniku (ime, prezime, razina prava pristupa), druga binarni kod generiran iz šarenice korisnika, a treća informacije o posljednjim odobrenim pristupima (kome i kada). Podaci prve i druge tablice povezane su jedinstvenim identifikacijskim ključem. Struktura svake tablice, zajedno sa SQL i Qt/C++ tipom podatka svakog stupca, dana je u tablicama 3.1. – 3.3.

Tab. 3.1. Struktura tablice s podacima o korisnicima

Tablica	Podaci o korisnicima			
Stupac	Ključ	Prezime	Ime	Razina prava pristupa
SQLite Tip	INTEGER	TEXT	TEXT	INTEGER
Qt/C++ Tip	int	QString	QString	int

Tab. 3.2. Struktura tablice s binarnim kodovima

Tablica	Binarni kodovi	
Stupac	Ključ	Kod
SQLite Tip	INTEGER	BLOB
Qt/C++ Tip	int	QByteArray

Tab. 3.3. Struktura tablice s podacima o povijesti pristupa

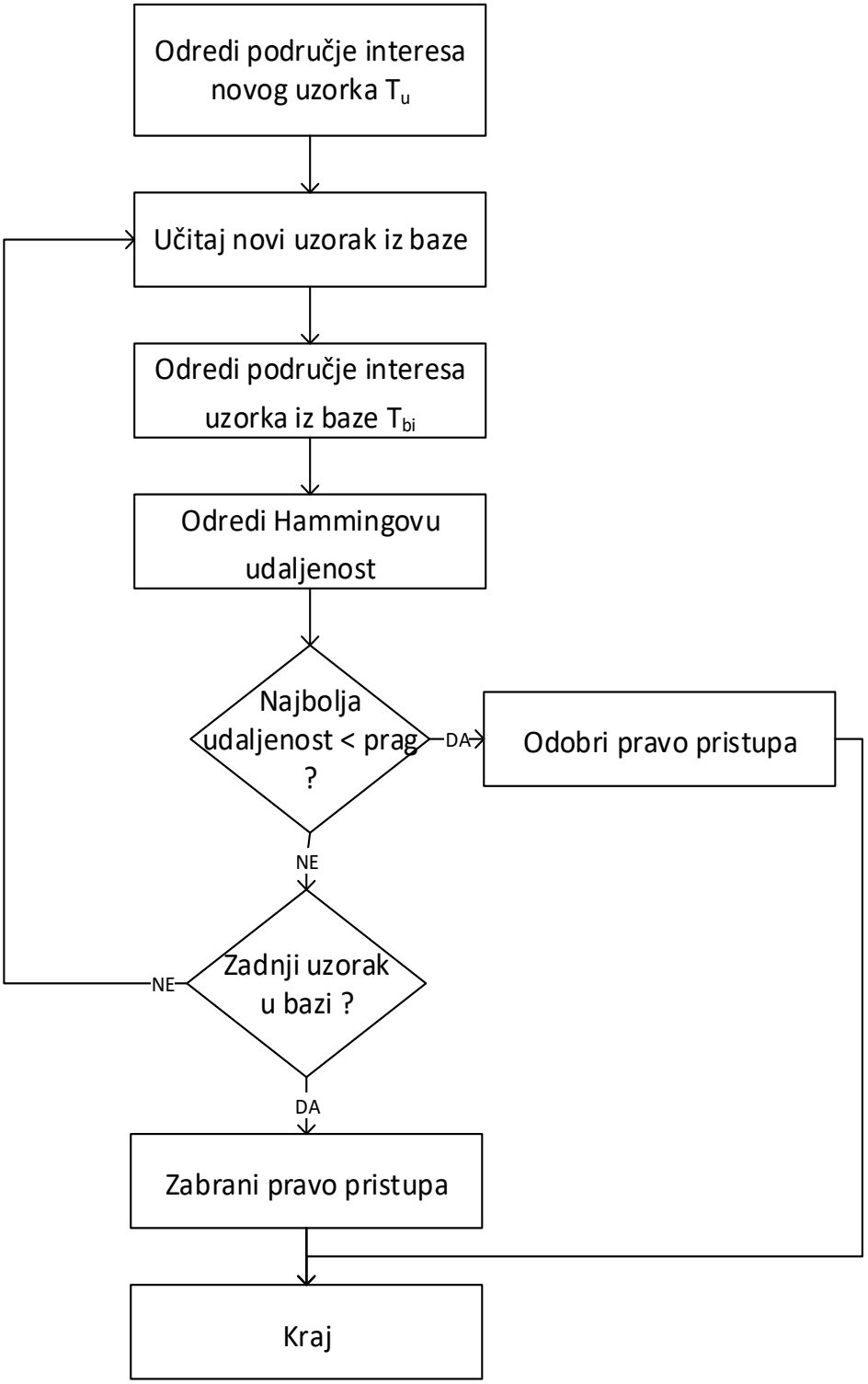
Tablica	Povijest pristupa		
Stupac	Ključ	Datum	Korisnik
SQLite Tip	INTEGER	TEXT	TEXT
Qt/C++ Tip	int	QString	QString

3.3.2. Provjera prava pristupa korisnika

Pri inicijalizaciji sustava, uzorci svih aktivnih korisnika iz baze se dohvaćaju i spremaju u radnu memoriju. To je izvedivo za manji broj uzoraka, a korisno jer smanjuje vrijeme identifikacije – nije potrebno za svaku provjeru slati upit bazi podataka kako bi se dohvatio kod. Prosječno vrijeme izvršavanja nekog upita SQLite bazi podataka iznosi 300ms, što znači da bi dohvaćanje svih korisnika jednim upitom prilikom svake prijave vrijeme identifikacije odužilo za 300ms. Ako bi se uzorci dohvaćali pojedinačno, vrijeme identifikacije linearno bi raslo brojem uzoraka u bazi.

Prilikom dohvaćanja nove fotografije (ostvareno pritiskom tipke na tipkovnici), ona se pohranjuje u radnu memoriju, a zatim se pokreću postupci lokalizacije i generiranja binarnog koda, a ako svi uspiju generirani kod se iterativno uspoređuje sa svim uzorcima u bazi. Ukoliko pristigli uzorak odgovara postojećem korisniku, dobiva pravo pristupa u skladu sa svojim ovlastima. U ovom radu implementirane su tri razine ovlasti – korisnik bez prava pristupa, korisnik s pravima pristupa i administrator. Korisnika bez prava pristupa će biti identificiran ali mu pravo pristupa neće biti odobreno kao što bi bilo odobreno korisniku s pravima pristupa. Administrator ujedno ima prava pristupa i upravljanja bazom.

U slučaju da je pristup odobren, potrebno je u bazu unijeti informacije o identitetu korisnika i vremenu prijave. Slanje informacija u bazu se obavlja neposredno nakon odobravanja pristupa. Na slici 3.40. nalazi se funkcionalni blok dijagram postupka uspoređivanja binarnog koda s bazom kodova aktivnih korisnika. Računanje Hammingove udaljenosti obavlja se prema blok dijagramu na slici 3.39.

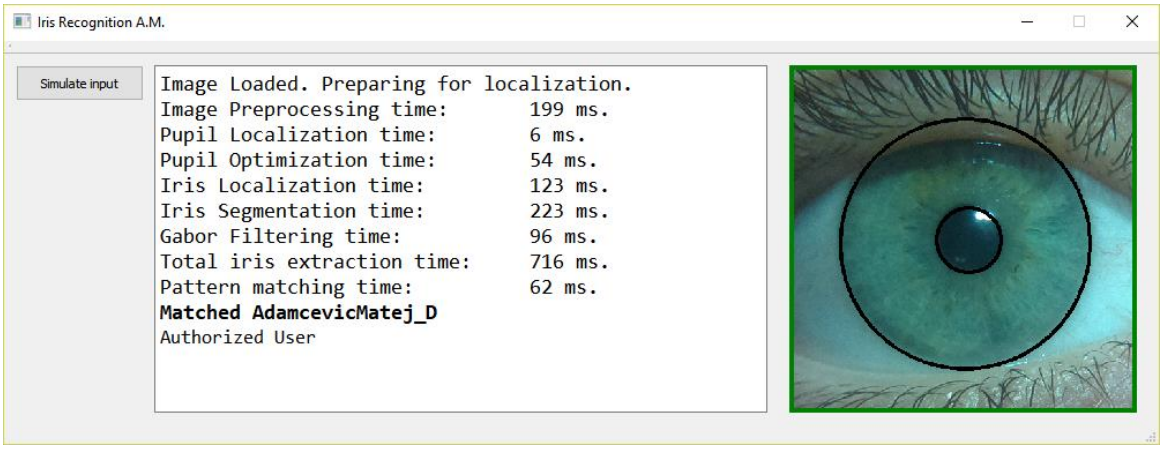


Sl. 3.40. Funkcionalni blok dijagram provjere prava pristupa

3.4. Korisničko sučelje i administracija korisnika

Kako je navedeno prilikom razrade rješenja, potrebno je osigurati grafičko sučelje prema krajnjem korisniku kao i sve osnovne funkcije za dugoročnu upotrebu i održavanje realiziranog sustava identifikacije. Te funkcije uključuju dodavanje novih korisnika u bazu aktivnih korisnika, brisanje korisnika iz baze, promjenu podataka postojećih korisnika, pregled svih korisnika kao i pregled zadnjih dolazaka (odobrenih pristupa).

Grafičko sučelje krajnjem korisniku sastoji se od prozora za poruke sustava prema korisniku i prozora za pregled slike dohvaćene s kamere u stvarnom vremenu. (Sl. 3.41.) U trenutku kada dođe do pokušaja prijave korisnika (dohvaćena slika), za vrijeme izvođenja programa za identifikaciju osvježavanje trenutne slike s kamere dobiva manji prioritet za izvođenje. Tada se u prozoru za pregled slike privremeno prikazuje rezultat lokalizacije oka pristiglog korisnika (ukoliko je uspješna).

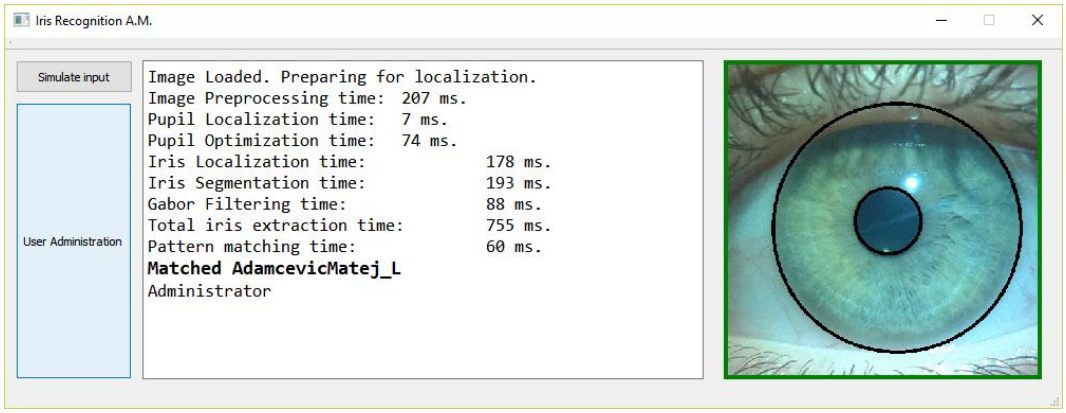


Sl. 3.41. Grafičko korisničko sučelje

Gumb „*Simulate Input*“ isprva je bio korišten u fazi razvoja programa, a može se koristiti za provjeru ispravnosti sustava za identifikaciju učitavanjem neke referentne slike.

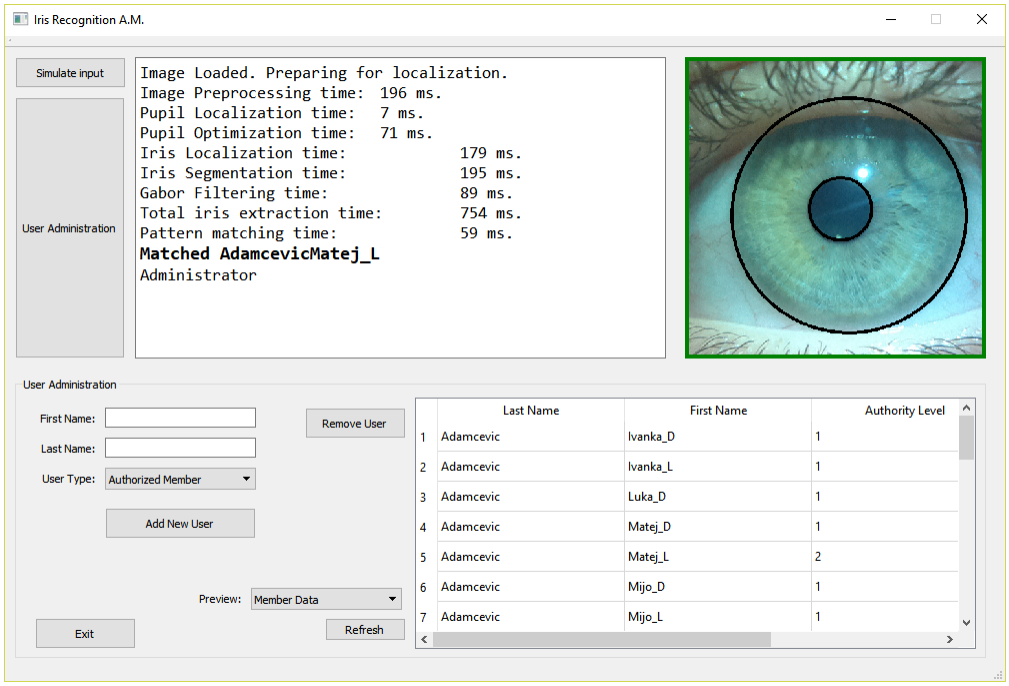
Ukoliko se prijavi korisnik s administrativnim pravima pristupa, u slobodnom se prostoru na lijevoj strani grafičkog sučelja na tri sekunde pojavljuje gumb „User Administration“. Klikom na taj gumb otvara se sučelje za upravljanje korisnicima. Za opciju dodavanja novog korisnika korištena su dva polja za unos imena i prezimena, padajući izbornik za odabir prava pristupa i gumb koji pokreće postupak dodavanja korisnika u bazu. Taj postupak sastoji se od prikupljanja

unesenih podataka, transformacije binarnog uzorka u format pogodan za zapisivanje u SQL bazu (jednodimenzionalni niz bajtova) te sastavljanja i slanja upita (engl. *query*) bazi .



Sl. 3.42. Prilikom prijave administratora, pojavljuje se gumb za pristup administrativnim funkcijama

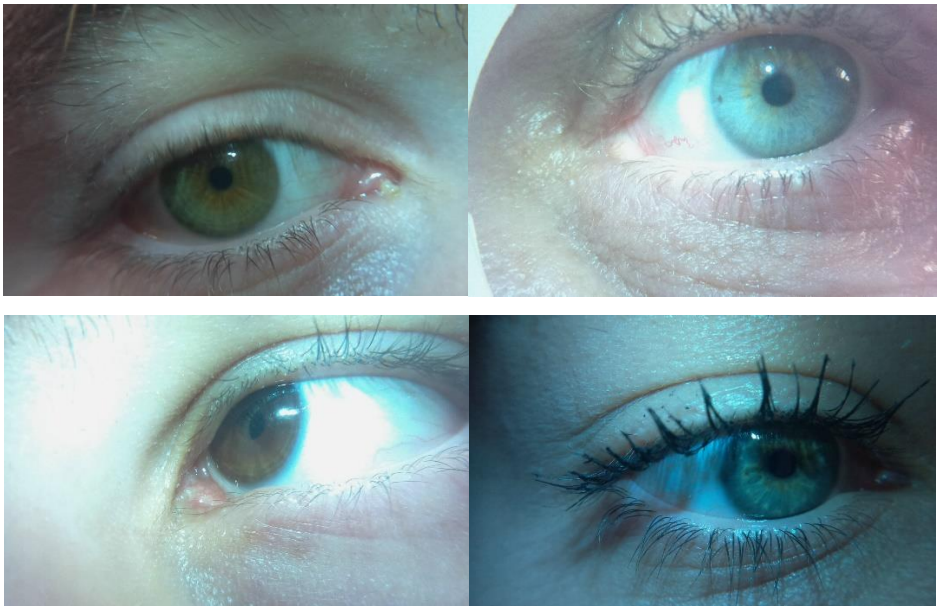
S desne strane sučelja za upravljanje korisnicima nalazi se prozor namijenjen pregledu baze. U pregled baze nije uključena druga tablica, tj. tablica s binarnim kodovima korisnika prvenstveno iz sigurnosnih razloga. Padajući izbornik nudi odabir pregleda općih informacija o korisnicima ili povijesti pristupa. Odabirom općih informacija prikazuju se podaci svih korisnika. Odabirom povijesti pristupa pojavljuje se dodatno polje za odabir broja prikazanih pristupa (zadnjih *n* odobrenih pristupa).



Sl. 3.43. Sučelje za administraciju korisnika

4. REZULTATI

Rezultati uspoređivanja binarnih kodova šarenice prikazani su u obliku histograma i kutijastog dijagrama (engl. *box plot*). Rezultati uspoređivanja kodova šarenice su na prvu ruku zadovoljavajući, no ako se u obzir uzmu rezultati koje postižu sustavi prof. Daugmana predstavljeni u [1], realizirani podsustav za generiranje i uspoređivanje kodova ima prostora za napredak. Programsko rješenje je testirano na osobnom računalu i *Raspberry Pi* mikroračunalu na ukupno 387 fotografija 26 različitih očiju od 16 različitih osoba. Lokalizacija granica zjenice i šarenice bila je uspješna za 355 fotografija. Na slici 4.1. prikazane su neke od fotografija neuspješne lokalizacije (zjenice ili šarenice). Najčešći razlozi neuspješne lokalizacije su nepovoljan korisnikov smjer gledanja i nedovoljna vidljivost rubova zjenice ili šarenice.

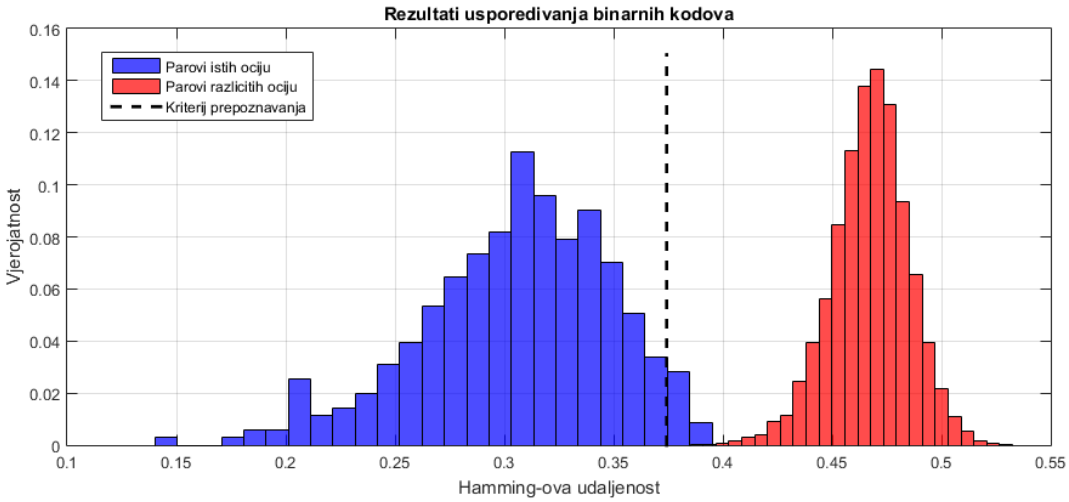


Sl. 4.1. Primjeri fotografija za koje lokalizacija nije u potpunosti uspjela

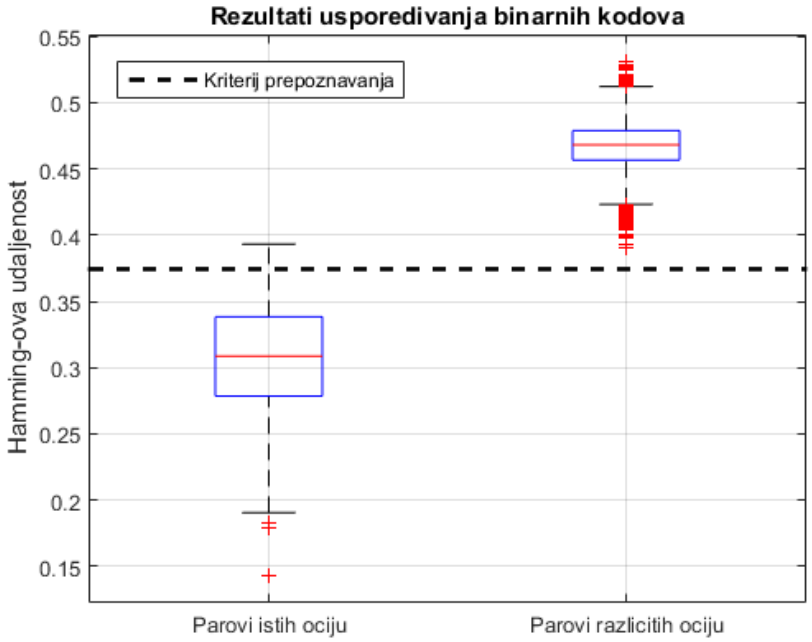
Houghova metoda za detekciju krugova na slici daje izuzetno precizne rezultate pod par uvjeta, uz one napomenute u drugom poglavlju rada. Prvi je da se postupci predobrade slike i ocjenjivanja kružnica realiziraju dovoljno specifično da točno i precizno obavljaju ono za što su namijenjeni. Drugi uvjet je da ti postupci budu dovoljno generalizirani kako bi bili primjenjivi na širokom rasponu slika s obzirom na razinu osvjetljenja i razine pigmentacije šarenice.

Uspoređeno je ukupno 355 parova istih i 8875 parova različitih očiju. Medijan Hamming-ovih udaljenosti uspoređenih parova istih očiju iznosi 0.30874, a parova različitih očiju 0.46816.

Dobivene rezultate uspoređivanja nije moguće jednoznačno odvojiti (Sl. 4.2., 4.3.) tj. rezultati uspoređivanja istih i različitih parova očiju se u nekom postotku preklapaju. Prema histogramu i kutijastom dijagramu, prag prepoznavanja nužno mora biti ispod vrijednosti 0.389 kako bi stopa lažnog prepoznavanja bila 0%. Konačni primijenjeni kriterij prepoznavanja iznosi 0.3749, što znači da se kodovi uzoraka moraju podudarati u barem 62.51% piksela. Na dostupnim slikama stopa lažnog prepoznavanja iznosi 0%, a stopa lažnog odbijanja 3.66% (odbijeno 13 od 355 uzoraka istih očiju), što je zanemarivo s obzirom na postignutu stopu lažnog prepoznavanja.

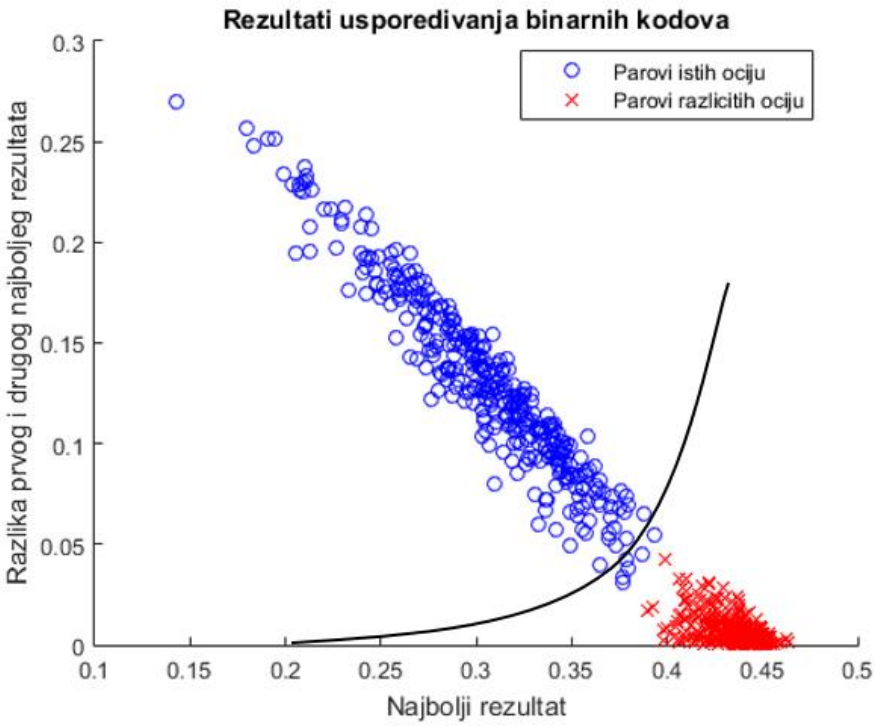


Sl. 4.2. Histogram rezultata uspoređivanja istih i različitih parova očiju



Sl. 4.3. Kutijasti dijagram rezultata uspoređivanja istih i različitih parova očiju

Prilikom pregleda rezultata uspoređivanja uočeno je kako radi nedovoljno precizne lokalizacije uzorci znaju odstupati neznatno iznad kriterija prepoznavanja. U isto vrijeme se sljedeći najbolji rezultat nalazi znatno iznad kriterija prepoznavanja – upravo zbog jedinstvenosti svakog uzorka šarenice. Pretpostavka je da, ukoliko korisnika nema u bazi, uzorak njegove šarenice neće nijednom postojećem korisniku biti znatno sličniji nego ostalima. Ako rezultat uspoređivanja uzorka postojećeg korisnika bude blago iznad praga, a sljedeće najbolje podudaranje (s uzorkom nekog od kolega) je u intervalu očekivanih vrijednosti za drukčije oko, može se s velikom vjerojatnošću reći da se radi o korisniku iz baze. U slučaju primjene ovakvog sustava odlučivanja, kriterij prepoznavanja bi bila krivulja demonstrirana na slici 4.4., gdje bi svi uzorci iznad krivulje zadovoljavali kriterij. U slučaju prikazanom na slici, stopa lažnog odbijanja iznosila bi 1.69% (odbijeno 6 od 355 uzoraka istih očiju). Ovakav pristup nije primijenjen u konačnom programskom rješenju, ali može biti od interesa za buduća istraživanja.

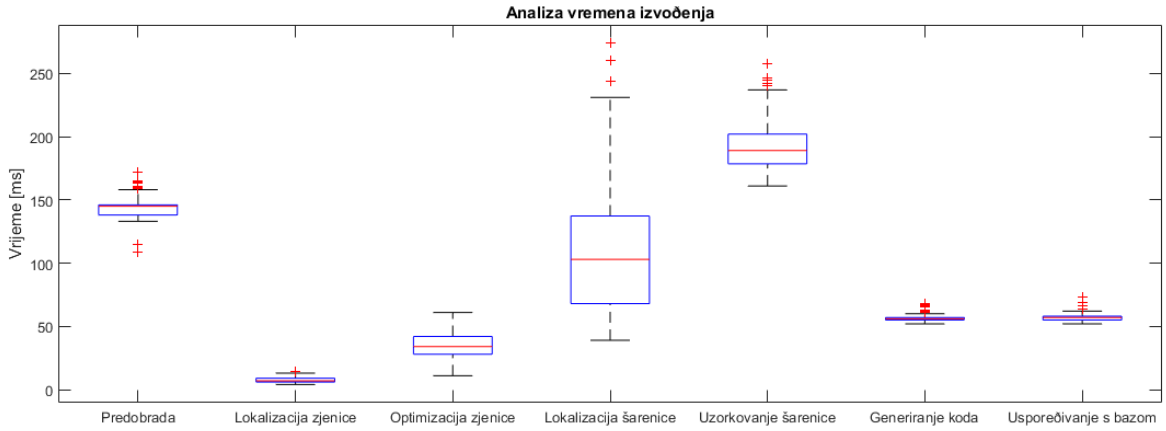


Sl. 4.4. Dvodimenzionalni pregled podataka

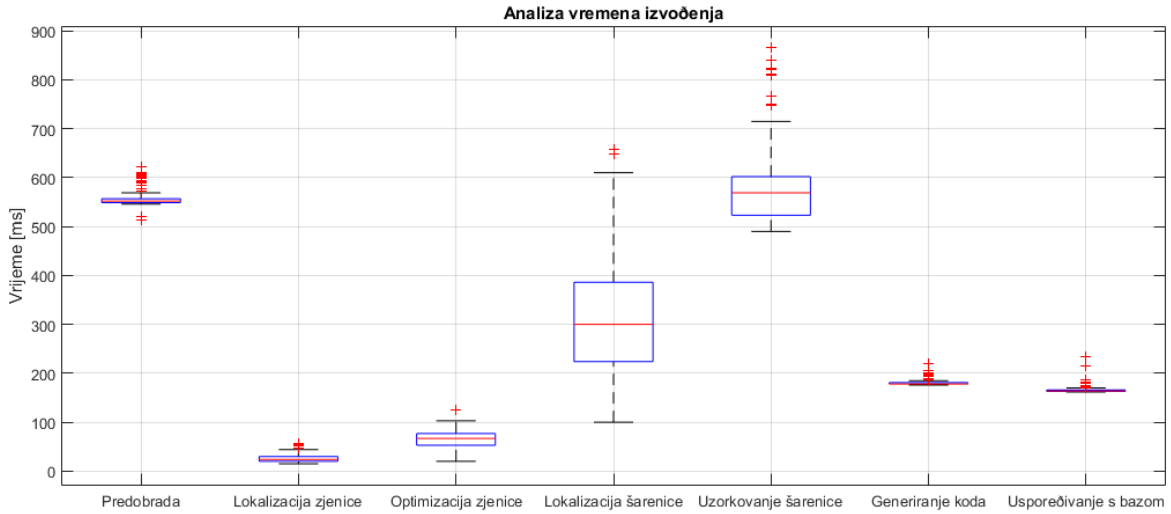
U okviru rada potrebno je predočiti vrijeme rada implementiranog algoritma identifikacije korisnika, kako na stolnom računalu tako i na ugradbenoj platformi (Raspberry Pi) na kojoj bi se ovakav sustav mogao i izvršavati u praktičnoj primjeni. Vremena izvršavanja svih većih

funkcijskih blokova su predočena kako bi se mogli identificirati računalni zahtjevni dijelovi, te dijelovi koji eventualno mogli biti predmet optimizacije prilikom implementacije sustava.

Vrijeme izvođenja je važno kako bi se kod praktične primjene postigao zadovoljavajući „user experience“ (npr. korisnici sigurno ne bi htjeli imati sustav koji provodi 5 sekundi identifikaciju, npr. za ulazak u sobu ili ured). Tako su u ovom slučaju mjerena vremena za predobradu slike, lokalizaciju i optimizaciju zjenice, lokalizaciju šarenice, uzorkovanje šarenice, generiranje binarnog koda i uspoređivanje uzorka s bazom od 26 uzoraka. Raspodjela vremena izvođenja funkcijskih blokova prikazana je kutijastim dijagramima, a statistički značajni podaci su izdvojeni u tablici 4.1.



Sl. 4.5. Vremena izvođenja funkcijskih blokova na osobnom računalu



Sl. 4.6. Vremena izvođenja funkcijskih blokova na Raspberry Pi mikroračunalu

Iz kutijastih dijagrama vremena izvođenja može se zaključiti kako je raspodjela vremena izvođenja vrlo slična – najviše vremena u oba slučaja uzimaju postupci predobrade slike i uzorkovanja šarenice, a najmanje lokalizacija i optimizacija zjenice. Vrijedi primijetiti kako je znatno manja razlika u vremenima izvođenja predobrade i uzorkovanja šarenice na *Raspberry Pi* nego na osobnom računalu. Iz toga se može zaključiti kako je predobrada relativno zahtjevan skup operacija za *Raspberry Pi* računalo. Ekstremne vrijednosti u postupku uzorkovanja šarenice javljaju se na onim slikama na kojima su kružnice koje opisuju unutarnju i vanjsku granicu šarenice izrazito ekscentrične. Takav međusobni položaj središta kružnica koje opisuju zjenicu odnosno šarenicu znači da će sa jedne strane zjenice (lijeve ili desne) udaljenost od ruba zjenice do vanjskog ruba šarenice biti veća nego ista ta udaljenost sa druge strane. Postupak linearne interpolacije traje duže ukoliko se podaci moraju dodavati (npr. red od 80 piksela svesti na duljinu 100 piksela) nego uklanjati (sa 120 na 100 piksela; detaljnije objašnjeno u potpoglavlju 3.2.5.).

Tab. 4.1. Rezultati analize vremena izvođenja pojedinih funkcijskih blokova

Osobno računalo / Raspberry Pi	Minimum	Donji kvartil	Medijan	Gornji kvartil	Maksimum	% ukupnog vremena
Predobrada	109	138	145	146	172	24.53
	513	549	552	557	623	29.77
Lokalizacija zjenice	4	6	7	9	13	1.18
	15	20	24	30	56	1.29
Optimizacija zjenice	11	28	34	42	61	5.75
	20	53	67	77	124	3.61
Lokalizacija šarenice	39	68	103	137	274	17.43
	100	224	300	386	657	16.18
Uzorkovanje šarenice	161	178	189	202	258	31.98
	490	523	569	602	865	30.69
Generiranje binarnog koda	52	55	56	57	68	9.48
	176	178	178	181	220	9.60
Uspoređivanje s bazom	52	55	57	58	73	9.64
	162	163	164	166	233	8.85

U Tab. 4.1. dani su rezultati analize vremena izvođenja pojedinih funkcijskih blokova. Medijan sveukupnog vremena izvođenja algoritma na osobnom računalu je 591 ms. Na *Raspberry Pi* mikroračunalu medijan ukupnog vremena iznosi 1854 ms. Najveća relativna razlika vremena izvođenja događa se u postupku predobrade slike za lokalizaciju zjenice. Medijan vremena izvođenja predobrade je 3.8 veći na *Raspberry*-u nego na osobnom računalu, odnosno predobrada u prosjeku traje skoro četiri puta duže na *Raspberry* računalu. Ukupno gledajući, srednje vrijeme izvođenja cijelog algoritma je 3.13 puta duže na *Raspberry* mikroračunalu u usporedbi sa osobnim računalom. Najgore vrijeme izvođenja postupka identifikacije (engl. *WCET, Worst Case Execution Time*) na osobnom računalu iznosi 919 ms, dok na ugradbenoj platformi ono iznosi 2768 ms.

5. ZAKLJUČAK

Atraktivnost automatiziranih sustava identifikacije osoba raspoznavanjem biometrijskih značajki neprestano raste. Razlog tome je što se radi o relativno mladom znanstvenom području koje puno obećava, nudi donedavno nezamislive mogućnosti, a stalno napreduje. Tehnološkim napretkom došlo je do značajnog porasta omjera kvalitete i cijene komercijalno dostupnih elektroničkih uređaja, među koje se ubraja i *Raspberry Pi* mikroračunalo razmatrano u ovom radu. Napredak tehnologije nije se očitovao samo u sklopovskom pogledu, već i u dostupnosti i kvaliteti dokumentacije i razvojnih okruženja za razvoj programskih rješenja.

Tehnike identifikacije osoba analizom uzorka šarenice doživljavaju značajan napredak u posljednjih nekoliko desetljeća, prvenstveno zahvaljujući radu pojedinaca sa američkih sveučilišta. Posljednjih godina svoje mjesto nalaze i u komercijalnoj primjeni, u projektima na državnim razinama, a i u potrošačkoj elektronici.

Realizacija kompletnog sustava od dohvaćanja fotografije, određivanja unutarnje i vanjske granice šarenice, generiranje i uspoređivanje binarnih kodova zajedno sa sustavom za upravljanje korisnicima u konačnici čini jedan vrlo opširan projekt, a spaja područja računalnog vida, ugradbenih računalnih sustava, fotografije i administracije podacima.

Dohvaćanje fotografije predstavilo se kao velik problem uporabom kamere fiksne fokalne duljine. Ostvaren sustav fotografiranja primjenjiv je za razvojne i pokazne svrhe, ali za komercijalno primjenjiv sustav bilo bi nužno implementirati mogućnost automatskog fokusiranja kamere. Time bi fotografija oka mogla biti dohvaćena iz šireg intervala udaljenosti, od minimalne do maksimalne podesive fokalne duljine.

Vrijeme određivanja granica šarenice moglo bi se skratiti implementacijom segmentacije slike po karakterističnim vrijednostima. Dodatno filtriranje segmentirane slike moglo bi jasnije odvojiti područja ključna za lokalizaciju - zjenicu, šarenicu i bjeloočnicu, umanjujući šum u podacima. Dodatno, postupci lokalizacije i optimizacije mogli bi se znatnije skratiti implementacijom višezadaćnosti u postupcima ocjenjivanja detektiranih kružnica.

Temeljna odlika sustava identifikacije baziranih na analizi uzorka šarenice je vrlo dobra odvojivost uzoraka istih i različitih očiju, u nekim primjenama i znatno bolja od one dobivene u ovom radu. Zaključak koji proizlazi iz toga je da čak i kad se uzorak šarenice kodira jednostavnim binarnim pragom, pruža dovoljno informacije za raspoznavanje od ostalih uzoraka. Kvaliteta

rezultata u smislu odvojivosti grupe istih od grupe različitih očiju može se povećati promjenom načina kodiranja uzorka šarenice. Predlaže se podjela akumulacijske matrice na manje segmente, te računanje histograma gradijenata intenziteta svjetline za svaki segment. Takvo bi kodiranje bi moglo kvalitetnije opisati uzorak šarenice, a i biti manje osjetljivo na rotaciju oka na fotografiji.

REFERENCE

[1] J. Daugman, »How Iris Recognition Works,« 2004.. [Mrežno]. Available: https://www.cl.cam.ac.uk/~jgd1000/csvt.pdf. [Pokušaj pristupa 28 Studeni 2017.].

[2] R. P. Wildes, »Iris Recognition: An Emerging Biometric Technology,« [Mrežno]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.1548&rep=rep1&type=pdf. [Pokušaj pristupa 28 Studeni 2017.].

[3] Raspberry Pi Foundation, »Raspberry Pi - Teach, Learn and Make,« [Mrežno]. Available: https://www.raspberrypi.org/. [Pokušaj pristupa 28 Studeni 2017.].

[4] Raspberry Pi Foundation, »Camera Module - Raspberry Pi Documentation,« [Mrežno]. Available: https://www.raspberrypi.org/documentation/hardware/camera/. [Pokušaj pristupa 9 Prosinac 2017].

[5] The Qt Company, »Qt | Cross-platform software development for embedded & desktop,« [Mrežno]. Available: http://doc.qt.io/. [Pokušaj pristupa 28 Studeni 2017.].

[6] OpenCV team, »OpenCV library,« [Mrežno]. Available: https://opencv.org/. [Pokušaj pristupa 28 Studeni 2017.].

[7] Hipp, Wyrick & Company, Inc., »SQLite Documentation,« [Mrežno]. Available: https://sqlite.org/docs.html. [Pokušaj pristupa 1 Prosinac 2017.].

[8] M. Bušić, B. Kuzmanović Elabjer i D. Bosnar, Seminaria Ophthalmologica, Udžbenik oftalmologije i optometrije, Osijek - Zagreb: Cerovski, 2014..

[9] »Using Telephoto Lenses,« [Mrežno]. Available: http://www.cambridgeincolour.com/tutorials/telephoto-lenses.htm. [Pokušaj pristupa 28 Studeni 2017.].

[10] M. Horner, »The Fujifilm Blog,« 26 Lipanj 2015.. [Mrežno]. Available: https://fujifilm-blog.com/2015/06/26/tutorial-understanding-depth-of-field/. [Pokušaj pristupa 28 Studeni 2017.].

[11] A. Zeng, »Andy's Iris Recognition,« [Mrežno]. Available: http://www.cs.princeton.edu/~andyz/irisrecognition. [Pokušaj pristupa 28 Studeni 2017.].

[12] OpenCV team, »Geometric Image Transformations,« [Mrežno]. Available: https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html. [Pokušaj pristupa 5 Prosinac 2017.].

[13] OpenCV team, »Eroding and Dilating,« [Mrežno]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html. [Pokušaj pristupa 1 Prosinac 2017.].

[14] »Bilateral Filtering,« [Mrežno]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html. [Pokušaj pristupa 1 Prosinac 2017.].

[15] OpenCV team, »OpenCV: Smoothing Images,« [Mrežno]. Available: https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html. [Pokušaj pristupa 1 Prosinac 2017.].

[16] OpenCV team, »Canny Edge Detector,« [Mrežno]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html. [Pokušaj pristupa 1 Prosinac 2017.].

[17] C. Boyce, A. Ross, M. Monaco, L. Hornak i X. Li, »Multispectral Iris Analysis: A Preliminary Study,« [Mrežno]. Available: http://www.cse.msu.edu/~rossarun/pubs/RossMSIris_CVPRW06.pdf. [Pokušaj pristupa 28 Studeni 2017.].

SAŽETAK

U ovom radu opisan je razvoj sustava za identifikaciju i kontrolu pristupa osoba zasnovan na analizi uzorka šarenice metodama računalnog vida. Programsko rješenje u obliku aplikacije ostvareno je koristeći Qt razvojno okruženje i openCV biblioteku za implementaciju funkcija računalnog vida. Predloženo je rješenje za dohvaćanje fotografija dodavanjem sustava leća ispred kamere. Detaljno je opisan razvoja programskog rješenja za analizu šarenice, od predobrade slike, preko određivanja unutarnje i vanjske granice šarenice do generiranja i uspoređivanja binarnog koda za potrebe identifikacije korisnika. Sličnost dva binarna koda određuje se računanjem Hammingove udaljenosti između kodova. Omogućen je sustav za kontrolu pristupa te nadzor i upravljanje korisnicima zasnovan na SQL bazi podataka i aplikaciji izrađenoj u Qt razvojnom okruženju. Programsko rješenje testirano je na osobnom računalu (Windows 10 OS) i mikroračunalu *Raspberry Pi* (*Raspbian* Linux OS) sa ukupno 355 fotografija, pri čemu je uspoređeno 355 parova istih i 8875 parova različitih očiju. Rezultati uspoređivanja binarnih kodova istih i različitih parova očiju pokazali su da je uzorak šarenice vrlo pogodno svojstvo za identifikaciju osoba. Predloženi sustav postiže stopu lažnog prepoznavanja od 0%, a stopu lažnog odbijanja od 3.66%.

Ključne riječi: identifikacija osoba, računalni vid, kontrola pristupa, biometrija, oko, zjenica, šarenica

ABSTRACT

Person Identification Using Computer Vision

This paper describes the process of developing a person identification and access control system based on iris pattern analysis, using computer vision. The application is developed using the Qt framework and the openCV computer vision library. A solution for acquiring good quality images of the eye was proposed by stacking macro lenses in front of the camera lens. The process of developing a software solution for iris pattern analysis is described in detail, from image preprocessing, inner and outer iris boundary approximation to generating and comparing binary iris codes for user identification. The similarity between two codes is determined by calculating the Hamming distance between the two codes. An access management and supervision system has been implemented using an SQL database and the Qt based application. The software has been implemented and tested on both a PC running Windows and a *Raspberry Pi* microcomputer running *Raspbian* Linux, using a set of 355 images, resulting in 355 same eye comparisons and 8875 different eye comparisons. The results of comparing identical and different pairs of eyes have shown that the pattern of the iris is a suitable basis for person identification. The proposed system achieves an ideal 0% false recognition rate and a satisfying 3.66% false rejection rate.

Keywords: person identification, computer vision, access management, biometry, eye, pupil, iris

ŽIVOTOPIS

Matej Adamčević rođen je 29.08.1993. u Osijeku. Osnovnoškolsko obrazovanje stekao je u osnovnoj školi „Mladost“ u Osijeku (2000. – 2008.), a srednjoškolsko u III. gimnaziji Osijek (2008. – 2012.). Nakon završenog srednjoškolskog obrazovanja, upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu Sveučilišta Josipa Jurja Strossmayera u Osijeku, kojeg završava 2015. godine. Iste godine upisuje diplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija na osječkom sveučilištu. Aktivno koristi engleski jezik u govoru i pismu, na razini izvornog govornika.