

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Digitalna obrada slike - Projekt

**IDENTIFIKACIJA OSOBA NA TEMELJU
ŠARENICE OKA POMOĆU SLOGA
GABOROVIH FILTATRA**

Toni Benussi

Darko Jurić

Ivan Kušt

Zagreb, svibanj 2011.

Sadržaj

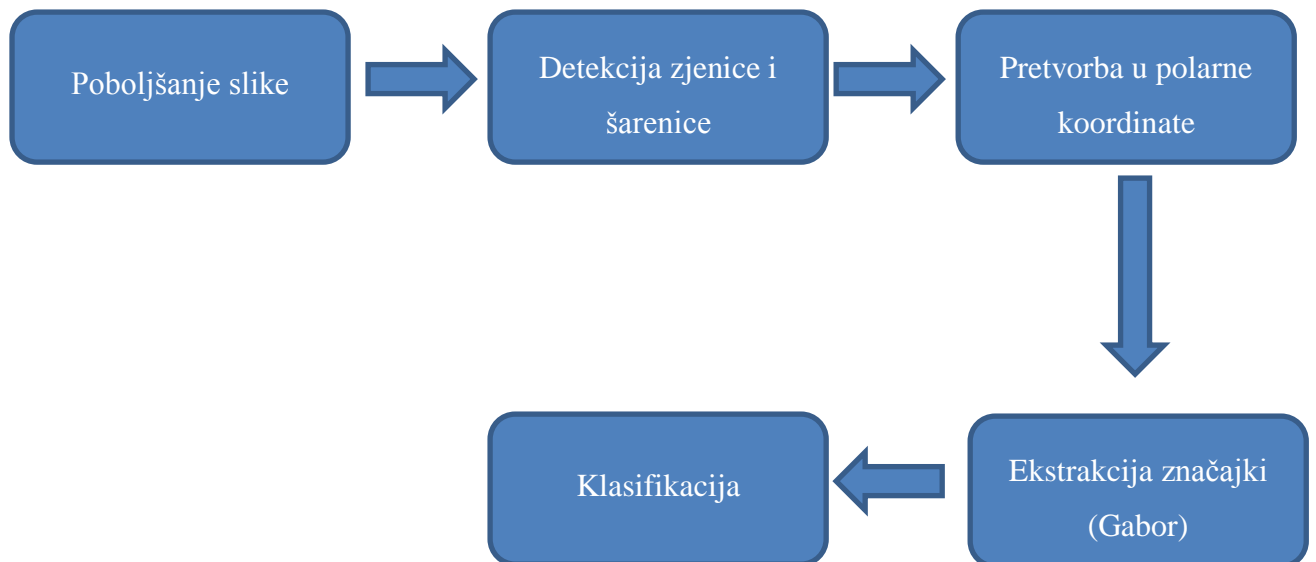
Sadržaj	1
Uvod	3
Korištene metode – teorija	4
1 Gaborov filter	4
1.1 Uvod	4
1.2 Analiza Gaborovog filtra u frekvencijskoj domeni	4
1.2.1 Ovisnost o frekvenciji	5
1.2.2 Ovisnost o veličini ovojnice	5
1.2.3 Ovisnost o orijentaciji	6
1.3 Slog Gaborovih filtara	6
2 KNN algoritam	8
Opis postupka	9
3 Detekcija zjenice	9
4 Detekcija šarenice	11
5 Odmatanje šarenice	13
6 Izlučivanje značajki	15
1.1 Ekstraktor – standardni slog Gaborovih filtara	16
1.2 Ekstraktor – hijerarhijski Gaborov slog	17
1.2.1 Odabir točaka	17
1.2.2 Ekstrakcija značajki	18
1.2.3 Optimiranje filtra	19
1.3 Značajka	20
1.4 Klasifikacija	20
1.5 Rezultati	21
7 Zaključak	23

8	Literatura.....	24
---	-----------------	----

Uvod

U postupcima za identifikaciju soba na temelju šarenice koriste se različite metode ekstrakcije značajki i klasifikacije. U ovom radu isprobane su različite ekstrakcije značajki koje se temelje na slogu Gaborovih filtara. Važno je napomenuti da se standardne metode temeljene na Gaborovim filtrima koje se koriste za identifikaciju šarenice u ovom radu uspoređuju sa metodama koje se primjenjuju na druge probleme konkretno na raspoznavanje rukom pisanih znakova.

Postupak identifikacije je sljedeći. Najprije se vrši poboljšanje slike zatim detekcija zjenice, a poto detekcija šarenice. Takva slika se prevodi u polarne koordinate. Ona se pomoću ekstraktora značajki temeljenog na slogu Gaborovih filtara prevodi u vektor značajki. Niz takvih vektora uvodi se u klasifikator (kNN). Kako je rečeno isprobano je više takvih ekstraktora značajki te su uspoređeni njihovi rezultati.



Korištene metode – teorija

Ovo poglavlje iznosi samo teorijsku podlogu o korištenim metodama. Rezultati kao i postupci vezani uz ovaj rad nalaze se u drugoj cjelini.

1 Gaborov filter

1.1 Uvod

Motivacija za korištenje Gaborovih filtara najviše je potaknuta biološkim aspektom, jer su u vizualnom korteksu mozga živih bića pronađene stanice čiji je odziv na pobudu sličan odzivu Gaborovih filtara. Pozitivnim reagiranjem na određene značajke u slici kao što su rubovi i oštri kutevi te time i osjetljivošću na određene frekvencije bili su ključni pokretači za daljnje istraživanje njegovih mogućnosti.

Gaborov filter je definiran kao produkt Gaussove funkcije i kompleksnog sinusnog signala. Kako Gaussova funkcija množi sinusni signal, ograničava njegovo djelovanje. Gaborov filter je kompleksan filter. Za dvodimenzionalnu inačicu filtra prikazan je njegov odziv izrazom 20.

$$g(\theta; \lambda; \sigma; x; y) = Ae^{\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)} e^{j\frac{\pi}{\lambda}(x\cos\theta+y\sin\theta)} \quad (1)$$

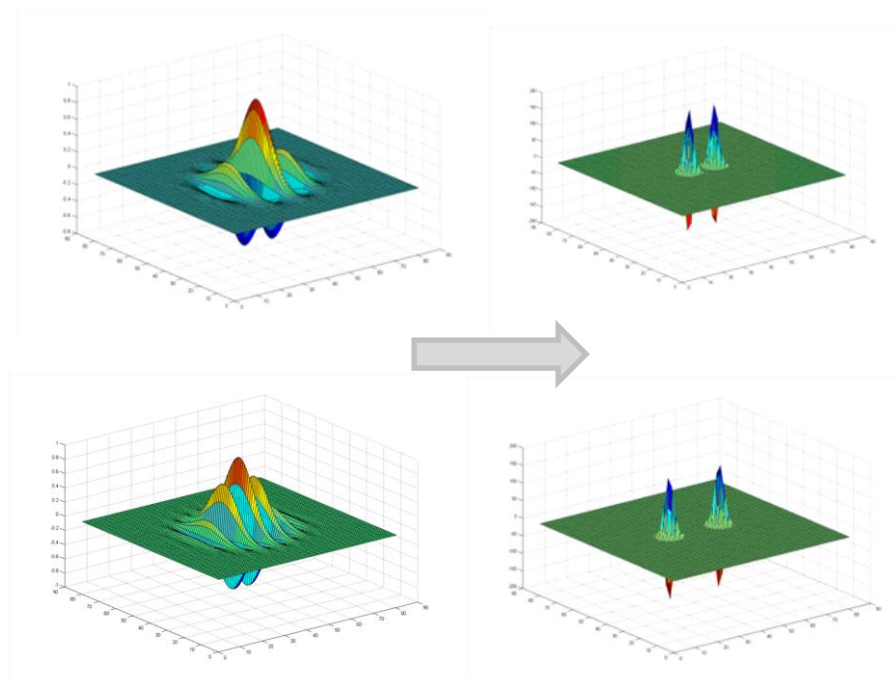
Parametri su: konstanta A, orijentacija filtra Θ , valna duljina λ , veličina Gaussove ovojnice σ , te ulazne koordinate x i y. Faze nema odnosno njezina je vrijednost jednaka 0.

1.2 Analiza Gaborovog filtra u frekvencijskoj domeni

Karakteristike Gaborovih filtara kao i ovisnost o parametrima je lakše razumjeti ako se izvrši Fourierova transformacija filtra te se takav promatra. Bit će prikazana ovisnost o promijeni frekvencije, veličini filtra te njegovoj orijentaciji. Kroz ova tri elementa moći ćemo pokriti cijelu frekvencijsku domenu. Iako je filter kompleksan zbog nemogućnosti prikaza oba dijela istovremeno prikazani su samo realni dijelovi filtra. Imaginarni dijelovi su samo pomaknuti u fazi.

1.2.1 Ovisnost o frekvenciji

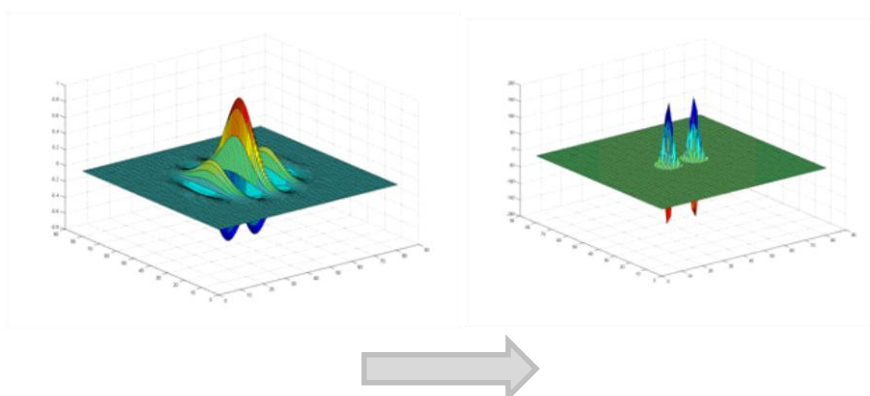
Mijenjanjem frekvencije postavljamo filter na odgovarajuće mjesto u frekvencijskoj domeni. Smanjivanjem frekvencije sinusni dijelovi filtra se približavaju dok njezinim povećanjem ti isti dijelovi se razmiču kako je i prikazano slikom.

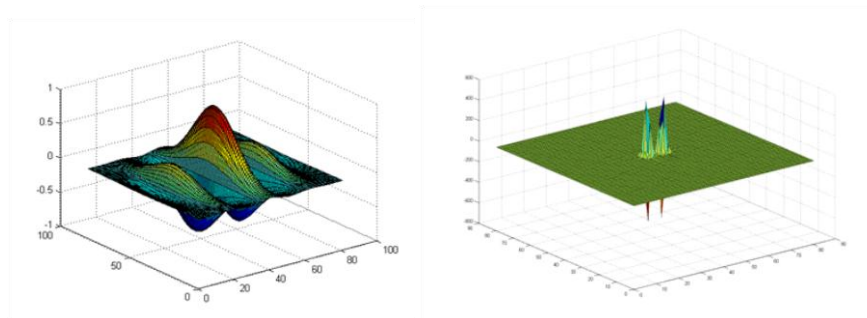


Slika 1. Ovisnost filtra o povećanju frekvencije.

1.2.2 Ovisnost o veličini ovojnice

Mijenjanjem veličine filtra dobivamo obrnuto proporcionalno mijenjanje veličine filtra u frekvencijskoj domeni.

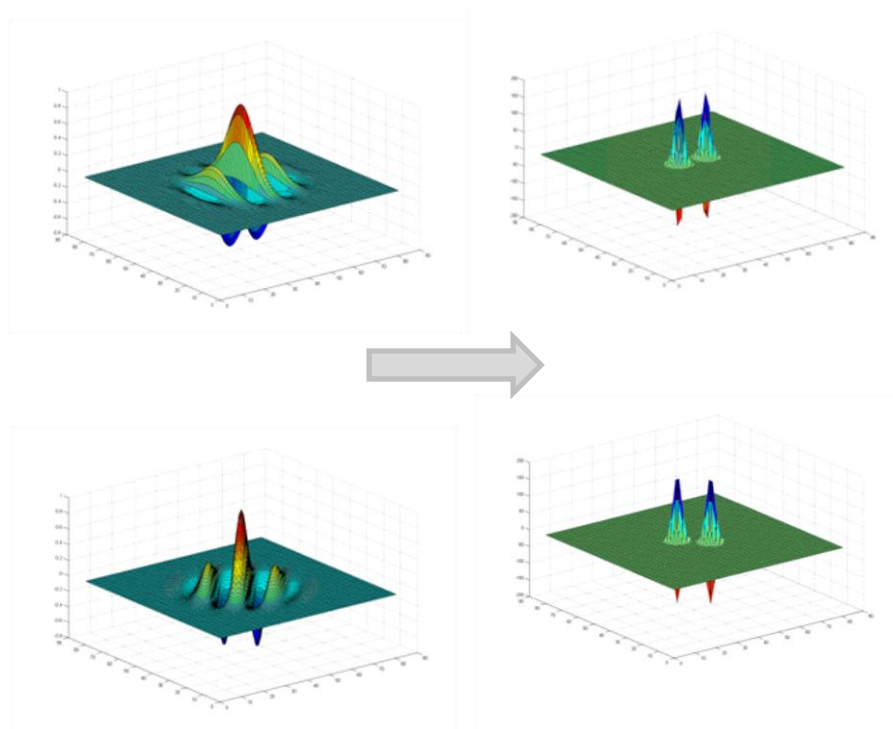




Slika 2. Ovisnost filtra o povećanju veličine Gaussove ovojnice

1.2.3 Ovisnost o orijentaciji

Promjena orijentacije utiče na obje domene jednako. Rotacijom filtra u prostornoj domeni zakrećemo filter i u frekvencijskoj domeni.



Slika 3. Ovisnost filtra o orijentaciji.

1.3 Slog Gaborovih filtara

Kako bi se izlučilo što više značajki objekta na slici česta je uporaba filtara uz mijenjanje njegove veličine i frekvencije. Da bi se lakše kreirao set filtara (*eng. Gabor bank*) često se uvodi oblik koji u sebi se sadrži parametar valne duljine.

$$g(x; y; \omega; \sigma; r; \theta) = \frac{1}{\sigma\sqrt{\pi}r} e^{-\frac{1}{2}\left[\left(\frac{R_1}{\sigma}\right)^2 + \left(\frac{R_2}{r\sigma}\right)^2\right]} e^{i\omega R_1}$$

gdje su R_1 i R_2 :

(2)

$$R_1 = x\cos\theta + y\sin\theta,$$

$$R_2 = -x\sin\theta + y\cos\theta,$$

Vrijednosti parametara σ (skaliranje, širina Gaussove ovojnice) se generiraju eksponencijalno.

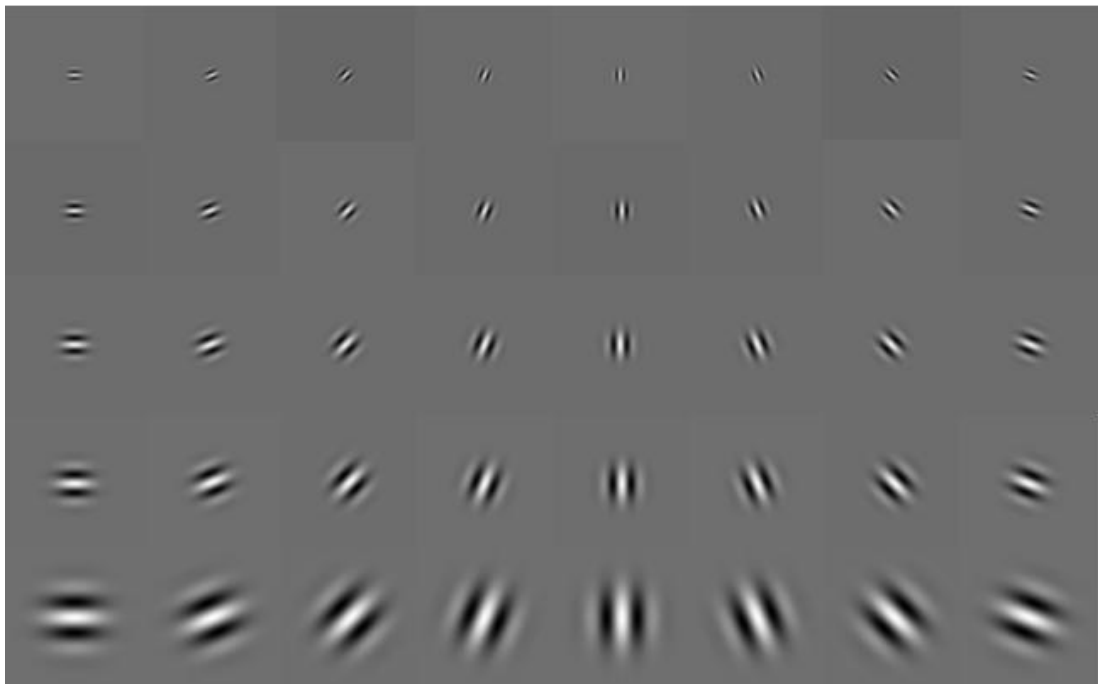
Jedan takav izraz prikazan je dolje.

$$\sigma_i = A * 2^{\frac{(-i+3/2)}{2}}, i \in \{0,1,2,\dots,N\}, A = \text{const.} \quad (3)$$

Vrijednosti parametra θ (kuta rotacije) dobivaju se tako da se jedinična kružnica podijeli na N jednakih dijelova:

$$\theta_j = j \frac{\pi}{N}, j \in \{0,1,2,\dots,N\} \quad (4)$$

Slikom dolje prikazano je 40 realnih dijelova Gaborovih filtara (5 skaliranja i 8 orijentacija uz konstantu frekvenciju) koji su generirani pomoću izraza (21).



Slika 4. Gaborov slog filtara (realni dio)

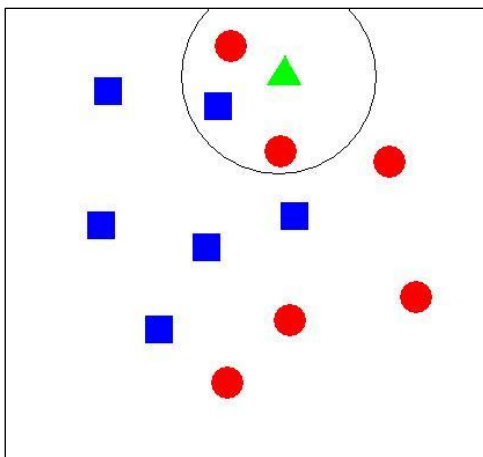
2 KNN algoritam

Algoritam k najbližih susjeda se temelji se na klasifikaciji uzoraka na temelju njima najbližih uzoraka iz skupa za učenje u n-dimenzionalnom vektorskom prostoru značajki. Ideja ovog algoritma je vrlo jednostavna i može se opisati u sljedećih nekoliko koraka:

- 1) Potrebno je zapamtiti pozicije u prostoru značajki za sve uzorke iz skupa za učenje kojima je poznata klasifikacija
- 2) Kad se dovede novi uzorak na ulaz klasifikatora pronade se k najbližih uzorka (iz skupa za učenje) tom uzorku nepoznate klasifikacije, pri čemu se udaljenost uzoraka računa kao euklidska udaljenost u vektorskom prostoru
- 3) Klasifikacija uzorka se obavlja sistemom glasanja, tako da se uzorak klasificira u razred koji se najčešće pojavljuje među susjedima tog uzorka (uzima se neparni broj susjeda k da se smanji vjerojatnost da broj glasova za 2 razreda bude jednak)

Klasifikator temeljen na ovom algoritmu, bez obzira na svoju jednostavnost, vrlo je robustan i daje konzistentne rezultate. Dokazano je da pogreška u klasifikaciji KNN algoritmom, ako broj uzoraka teži u beskonačnost, nije veća od dvostruke minimalne pogreške koju je moguće postići s obzirom na razdiobu kojoj pripadaju uzorci koje klasificiramo.

Na slici 10 vidimo primjer rada algoritma k najbližih susjeda u 2-dimenzionalnom prostoru značajki kojem su uzorci jedne klase označeni plavim kvadratima, a uzorci druge klase crvenim krugovima, a novi uzorak zelenim trokutom. Ako uzmemo $k=3$ onda su tri najbliža susjedna tri označena uzorka, te sistemom glasanja klasificiramo novi uzorak u klasu predstavljenu crvenim krugovima.



Slika 5. Primjer klasifikacije pomoću KNN algoritma

Opis postupka

3 Detekcija zjenice

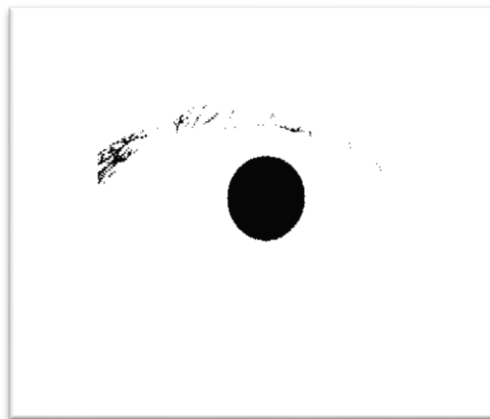
Prvi korak u lokalizaciji oka na slici je detekcija kružnice koja aproksimira rub zjenice.

Prije svega potrebno je infracrvenu snimku oka pretvoriti u sliku u tonovima sive boje. Nakon toga provodi se binarizacija slike. Iz vrijednosti svjetline piksela slike računa se prag za binarizaciju po sljedećoj formuli:

$$\text{threshold} = 0.54 \frac{\sum_N f(x, y)}{N} \quad (5)$$

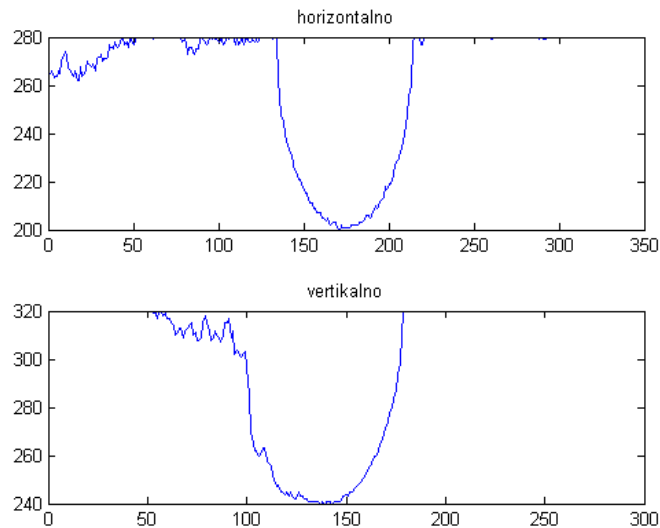
gdje je N broj piksela slike.

Suma vrijednosti svjetline svih piksela dijeli se s brojem piksela u slici čime se dobiva prosječna vrijednost svjetline piksela. Ta vrijednost se množi s faktorom 0.54. Binarizacija slike se provodi tako da se sve vrijednosti svjetline piksela koje su manje od zadanog praga postave u 0 (crna boja), a sve koje su više u 1 (bijela boja). Rezultat provođenja ovog postupka možemo vidjeti na sljedećoj slici:



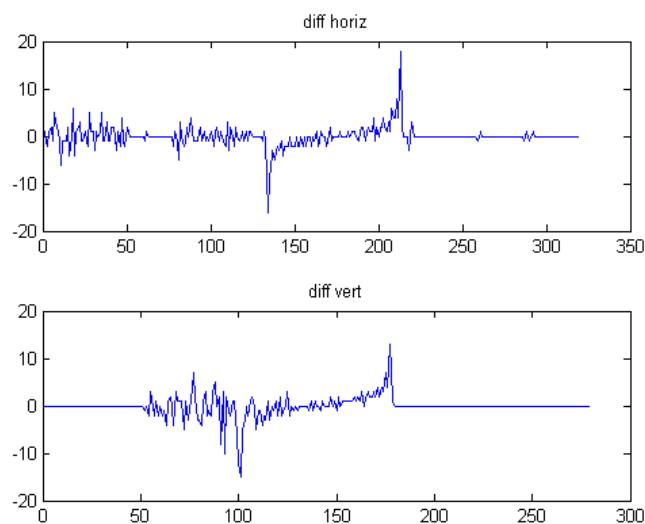
Slika 6. Slika nakon provedene binarizacije

Na dobivenoj slici možemo uočiti krug koji predstavlja zjenicu oka uz nešto malo šuma od trepavica. Nakon binarizacije slike radimo projekciju po redcima i po stupcima slike (tj. na vertikalnu i horizontalnu os). Projekcija se računa tako da se zbroje sve vrijednosti piksela u nekom retku ili stupcu, i tako za svaki redak / stupac koji se projicira na os. Kao rezultat dobiti ćemo dva vektora:



Slika 7. Projekcije na horizontalnu i vertikalnu os

Iz projekcija možemo lako vidjeti da imaju „rupu“ na mjestu koje odgovara položaju zjenice na horizontalnom, odnosno vertikalnom presjeku slike kroz središte zjenice. Da bi odredili centar i radijus kružnice koja aproksimira zjenicu, potrebno je odrediti rubove dobivenih „rupa“ na projekcijama. Jednom kada odredimo te rubove, možemo odrediti kvadrat u koji je upisana kružnica koja aproksimira zjenicu. Središte kružnice je tada težište kvadrata, a radijus je jednak polovini stranice kvadrata.



Slika 8. Funkcije razlike projekcija na horizontalnu i vertikalnu os

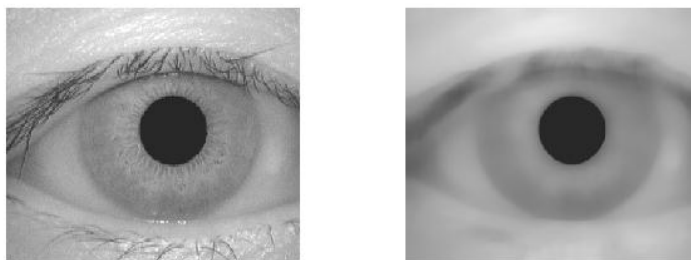
Kako bi mogli detektirati rubove „rupe“ u projekcijama, računamo njihove funkcije razlike. Pod funkcijom razlike misli se na razliku susjednih elemenata vektora projekcije. Na gornjoj slici možemo vidjeti te funkcije. Lako je uočiti da je lijevi rub „rupe“ u projekciji predstavljen minimumom dobivene funkcije, a desni maksimumom. Nakon što odredimo rubove, x

koordinatu centra zjenice možemo odrediti kao aritmetičku sredinu vrijednosti za koje se postiže minimum i maksimum u horizontalnoj projekciji. Analogno, y koordinatu centra zjenice možemo odrediti kao aritmetičku sredinu vrijednosti za koje se postiže minimum i maksimum u vertikalnoj projekciji. Za radijus uzmemo veću od polovica duljine intervala između minimuma i maksimuma iz svakog od presjeka.

4 Detekcija šarenice

Detekcija šarenice u implementiranom algoritmu oslanja se na rezultate dobivene detekcijom zjenice i na zasebni postupak predobrade slike čiji opis slijedi u nastavku.

Za detekciju šarenice korišten je postupak za koji je ključno detektirati rubove šarenice, te se predobrada slike upravo fokusira na to da se smanji mogući šum kod detekcije rubova i poveća kontrast na slici kako bi sami rubovi šarenice postali jasniji. Glavni izvori šuma kod detekcije granica šarenice mogu biti sama tekstura unutar šarenice i prisustvo trepavica koje dijelom zaklanjaju šarenicu. Karakteristika oba izvora šuma je da je riječ o vrlo tankim izduženim regijama slike, te se zbog toga za njihovo uklanjanje vrlo dobrim pokazao median filtar s velikom prozorom. Rezultati takvog filtriranja mogu se vidjeti na slici Slika 9 .

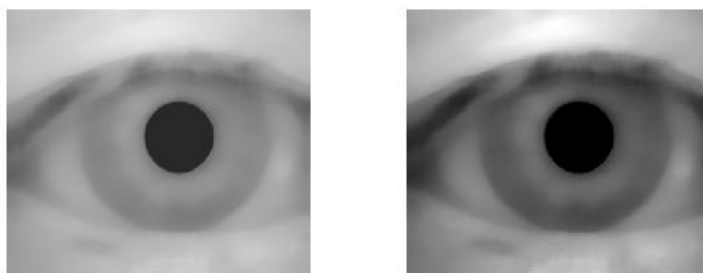


Slika 9. Primjena median filtra na sliku oka

Nakon primjene median filtra za povećanje kontrasta u dijelu raspona intenziteta sivih razina na slici koristi se postupak ekvivalentan gamma korekciji prema formuli:

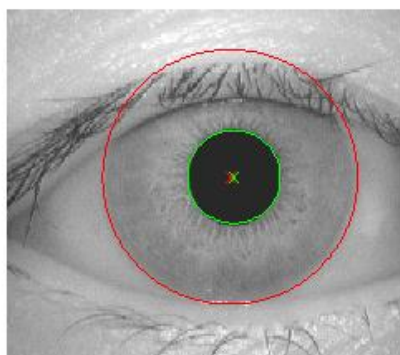
$$s = cr^\gamma \quad (6)$$

Takvom se operacijom i nakon toga skaliranjem na originalni raspon vrijednosti dobiva rezultat prikazan na slici Slika 10, a glavna prednost slike dobivene gamma korekcijom je značajno bolji kontrast između šarenice i bjeloočnice.



Slika 10. Gamma korekcija s faktorom 1.8 na sliku dobivenu median filtriranjem

Sljedeći korak u detekciji šarenice je detekcija rubova na slici za što je korišten Cannyjev detektor rubova na slici dobivenoj postupkom obrade koji smo upravo opisali. Znanje iz domene primjene nam govori da se sigurno zjenica koju smo prethodno detektirali nalazi unutar šarenice i da su središta tih dviju kružnicu vrlo blizu jedno drugom, ali se nužno ne poklapaju u potpunosti. Unatoč tome pozicija zjenice je vrlo dobar početni rezultat za traženje šarenice. Sam postupak detekcije šarenice je vrlo jednostavan. Na slici dobivenoj Cannyjevim detektorom rubova traže se prvo lijevo i prvo desno sjecište horizontalne linije kroz središte zjenice sa detektiranim rubovima koji se nalaze na udaljenosti od središta zjenice barem 1.5 puta većoj od radiusa zjenice. Ovaj zadnji uvjet je dodan da bi se izbjegao pronalazak sjecišta sa rubovima koji mogu eventualno biti blizu unutarnjeg ruba šarenice ako median filter nije uspio potpuno ukloniti teksturu unutar šarenice. Iz dobivenih sjecišta se računaju središte šarenice kao srednja vrijednost koordinata tih sjecišta i radius šarenice kao polovina udaljenosti tih sjecišta. Na slici 3 je prikazan rezultat detekcije šarenice gdje zelena kružnica predstavlja ranije detektiranu zjenicu, a crvena kružnica šarenicu.



Slika 11. Primjer detektirane šarenice

Zamjerka ovakvom načinu detekcije šarenice mogla bi biti već spomenuta činjenica da se središta zjenice i šarenice ne poklapaju, a ovaj algoritam podrazumijeva da se ta središta

nalaze na istoj horizontalnoj liniji. Međutim na većini slika gornji i donji dio šarenice je djelomično zaklonjen kapkom oka, te je neupotrebljiv za pouzdanu identifikaciju na temelju šarenice, te se u postupku klasifikacije ne koristi, pa eventualna manja pogreška u procjeni vertikalne komponente položaja šarenice nije značajna za sam postupak identifikacije.

5 Odmatanje šarenice

Detektirana šarenica ima oblik prstena, koji nije uvijek jednake širina, a i pomak središta zjenice u odnosu na središte šarenice otežava korištenje šarenice u klasifikaciju u svom izvornom obliku. Zbog toga se kod identifikacije osoba na temelju šarenice u pravilu koristi postupak odmatanje šarenice (*eng. unwrapping*) kako bi se šarenica pretvorila u oblik trake standardiziranih dimenzija koja se koristi kao ulaz u klasifikator. Za potrebe ovog rada korišten je postupak opisan u radu [5].

Cilj postupka je prikaz šarenice na način da se za opis šarenice umjesto koordinata standardnog koordinatnog sustava koriste polarne koordinate. Postupak počinje određivanjem brojem točaka koje će se koristiti za prikaz šarenice, tj. na koliko dijelova ćemo podijeliti šarenicu po kutu i po radijusu. U konkretnoj implementaciji podjela se vrši na 512 segmenata po kutu tako da je $d\theta = 2\pi/512$, a radijalno na ukupno 64 segmenta ($N=64$). Postupak pretvorbe šarenice u oblik trake računa za svaki sljedeći kutni segment veličine $d\theta$ četiri točke:

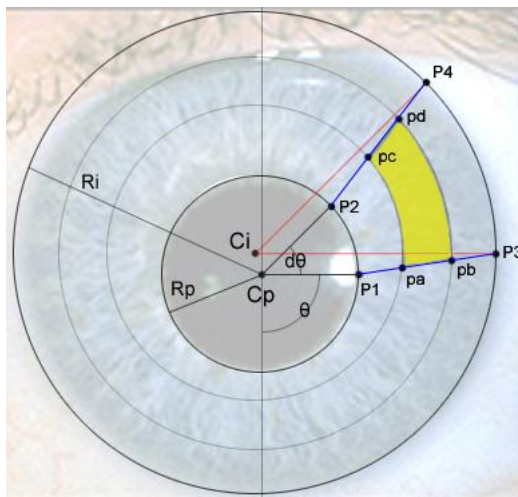
$$\begin{aligned} P_1 &= C_p + R_p(\cos(\theta) - \sin(\theta)) \\ P_2 &= C_p + R_p(\cos(\theta + d\theta) - \sin(\theta + d\theta)) \\ P_3 &= C_i + R_i(\cos(\theta) - \sin(\theta)) \\ P_4 &= C_i + R_i(\cos(\theta + d\theta) - \sin(\theta + d\theta)) \end{aligned} \quad (7)$$

gdje C_p i R_p predstavljaju središte i radius zjenice, a C_i i R_i su središte i vanjski radijus šarenice. Nakon toga se za svaki (k -ti) segment po radijusu računaju sljedeće točke:

$$\begin{aligned} P_a &= P_1(1 - \frac{k}{N}) + P_3 \frac{k}{N} \\ P_b &= P_1(1 - \frac{k+1}{N}) + P_3 \frac{k+1}{N} \\ P_c &= P_2(1 - \frac{k}{N}) + P_4 \frac{k}{N} \\ P_d &= P_2(1 - \frac{k+1}{N}) + P_4 \frac{k+1}{N} \end{aligned} \quad (8)$$

Svaki dio šarenice omeđen točkama P_a, P_b, P_c, P_d u svakoj iteraciji služi za određivanje

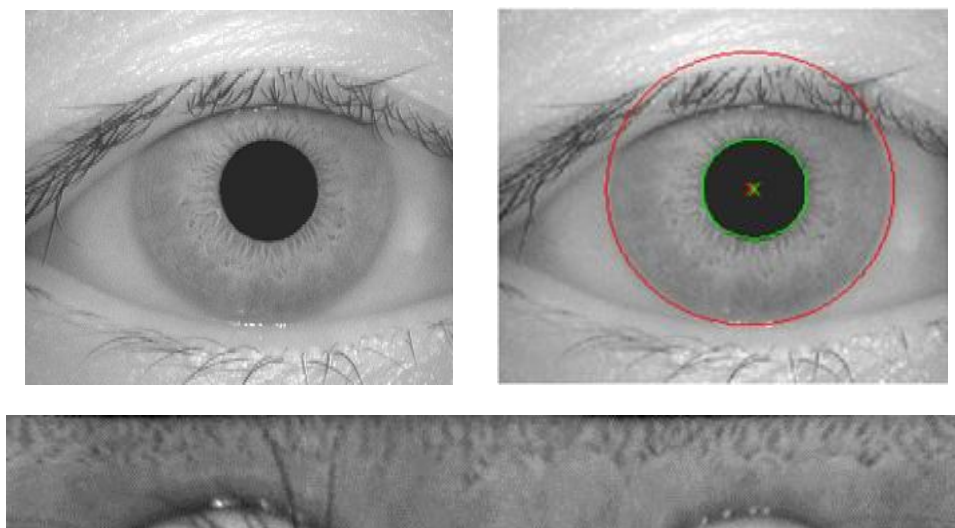
jednog piksela odmotane šarenice tako da se vrijednost tog piksela računa kao srednja vrijednost svih piksela u segmentu slike omeđenom sa te 4 točke. Grafički prikaz tog postupka prikazan je na primjeru na slici Slika 12.



Slika 12. Primjer algoritma odmotovanja šarenice

Rezultati takvog postupka za sve oblike i veličine šarenice daju traku istih dimenzija, što je puno lakše koristiti kasnije prilikom izrade klasifikatora. Druga velika prednost ovakvog rastezanja ili stezanja šarenice na istu veličinu je ta da je takvo ponašanje u skladu s prirodnim načinom funkcioniranja oka. Budući da zbog različitih uvjeta osvjetljenja zjenica može biti različite veličine, to uvjetuje i promjenu veličine šarenice, a prilikom promjene otvora zjenice šarenice se steže ili rasteže na način koji se može vrlo dobro aproksimirati linearnim rastezanjem, što se upravo i koristi prilikom uzorkovanja šarenice po radijusu. Također ovakav postupak je invarijantan i na samu veličinu slike tako da se u načelu mogu uspoređivati šarenice dobivene uređajima sa različitim rezolucijama snimanja.

Prije korištenja šarenice u postupku klasifikacije na već razmotanoj šarenici koristi se isti postupak povećavanja kontrasta gamma korekcijom koji je opisan u poglavlju o detekcije šarenice kako bi sama tekstura šarenice na kojoj se identifikacija osoba temelji bila bolje izražena na donjoj slici.

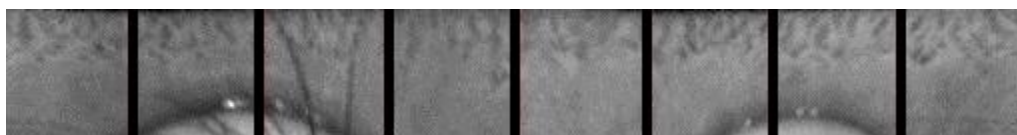


Slika 13. Originalna slika, detektirana šarenica i njena odmotana šarenica

6 Izlučivanje značajki

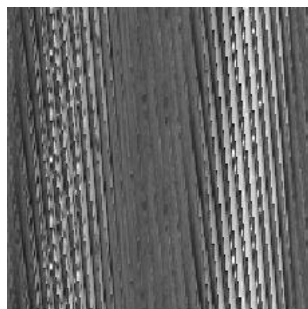
Nakon metoda poboljšavanja slike oka, slika se prikazuje u polarnim koordinatama kako je navedeno u prošlom poglavlju. Dobivena slika se obrađuje na dva načina.

Za standardni slog Gaborovih filtara (slijedeće poglavlje) dobivena slika se dijeli na $\text{širinaSlike}/\text{visinaSlike}$ dijelova te svaki takav segment obrađuje se zasebno slogom Gaborovim filtrima. Važno je da takav segment ima jednaku visinu i širinu inače će se ostatak odbaciti. Značajke koje su dobivene iz svake takvog segmenta slažu se u jedinstveni vektor značajki.



Slika 14. Primjer razmotane slike podijeljene na pravokutne segmente.

Za hijerarhijski slog je više pogodna jedinstvena slika pravokutnog oblika. Zbog toga se pravokutna slika proširuje do onog broja točaka kod kojeg se može podijeliti na jednaku širinu i visinu. Takva slika ulazi dalje u hijerarhijski ekstraktor.



Slika 15. Primjer razmotane slike pretvorene u pravokutnu sliku

Slijedi opis više ekstraktora značajki kao i njihovih vektora koje generiraju.

1.1 Ekstraktor – standardni slog Gaborovih filtara

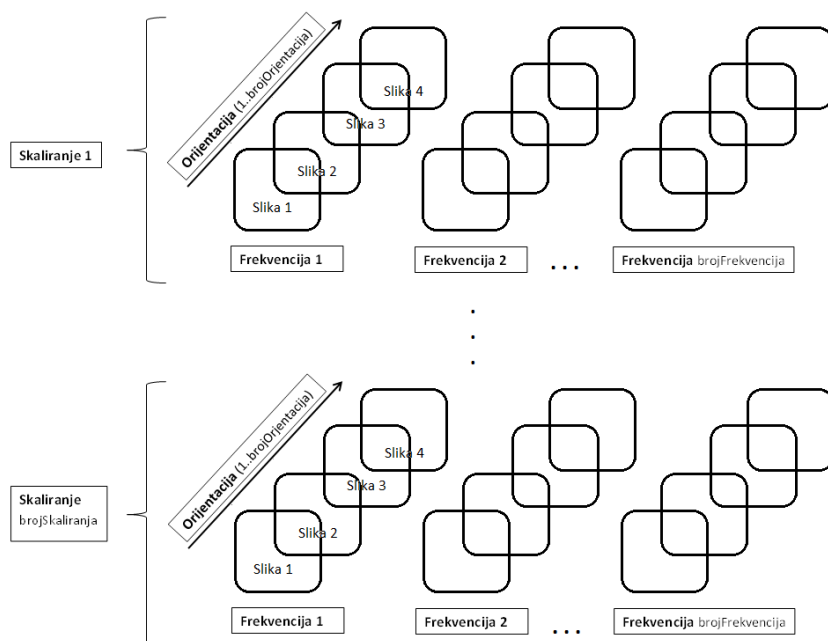
Kako je rečeno Gaborov filtar sastoji se od nekoliko parametara. Ovdje su korišteni sljedeći:

$$Orjentacije (\theta) = 0^\circ, 45^\circ, 90^\circ, 135^\circ$$

$$Frekvencije (\omega) = 2, 4, 8, 16, 32 \text{ Hz}$$

$$Skaliranje (\sigma) = 4$$

Slog se tako upotrebljava da za svako skaliranje za svaku frekvenciju i orijentaciju se vrši konvolucija sa i -tim ($i \in [1..brojSegmenata]$) segmentom slike. Kao postoji 5 različitih frekvencija te 4 orijentacije dio vektora značajki duljine je 20 točaka dok cijeli iznosi $20 * 8 = 160$ točaka.



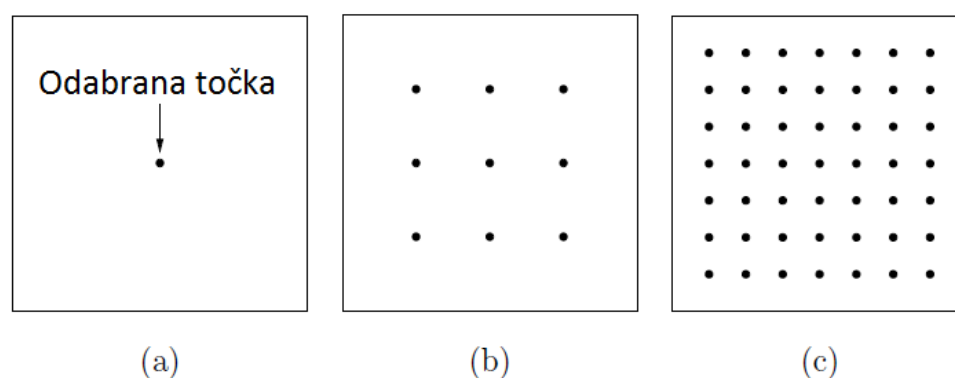
Slika 16. Prikaz ekstrakcije značajki pomoću sloga filtara

1.2 Ekstraktor – hijerarhijski Gaborov slog

1.2.1 Odabir točaka

Na ulaznoj slici se najprije određuje centar filtra. Nakon toga traže se točke jednako udaljene od pronađenog centra. Ova procedura je rekurzivna što znači da se može dobiti proizvoljan broj točaka na slici. Svako daljnje nalaženje susjednih točaka ujedno predstavlja jednu razinu. Neka broj razina iznosi 3. Time broj točaka na pojedinoj razini iznosi: 1, 3 i 49.

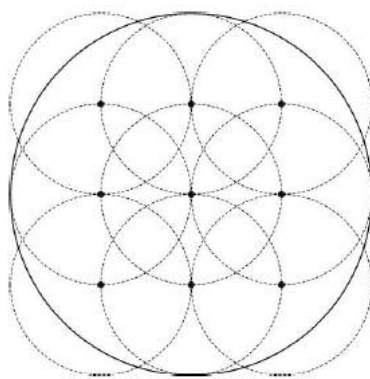
Ovaj postupak ilustriran je donjom slikom.



Slika 17. Postupak odabira točaka pojedinih razina.

Važno je za uočiti da se povećanjem broja točaka, točka koja je dobila nove susjede također ostaje u novoj razini. Ako se upotrebljava veći broj skaliranja i orijentacija tada ova metoda generira prevelik vektor značajki. Da bi ga se smanjilo broj točaka iduće razine odabire se tako da samo zadovoljava uniformnu raspoređenost (kvadrat nekoga broja) i veću količinu točaka nego prethodna razina. Time svojstvo da točka prethodne razine ostaje u sljedećoj razini ne mora vrijediti.

Svi filtri neke razine imaju zajedničku širinu Gaussove ovojnice. Svaka sljedeća razina smanjuje ovu vrijednost tako da se zadovolji da filtri prekrivaju površinu koja je određena udaljenošću između susjednih točaka, a ujedno s vrlo malim udjelom ulaze u takve površine njihovih susjeda. Prikaz sa 9 točaka i filtrima koji imaju centar u njima dan je donjom slikom.



Slika 18. Prikaz prekrivanja filtara neke razine

1.2.2 Ekstrakcija značajki

Najprije se odredi broj razina i broj točaka u svakoj od njih. Također za svaku razinu je određeno skaliranje. Za svaku točku je pridijeljen skup filtara koji je određen izabranim orijentacijama. Svaki filter koji je određen skaliranjem, točkom i orijentacijom određuje se optimalna frekvencija (vidjeti sljedeće poglavlje).

Korišteno su sljedeći parametri:

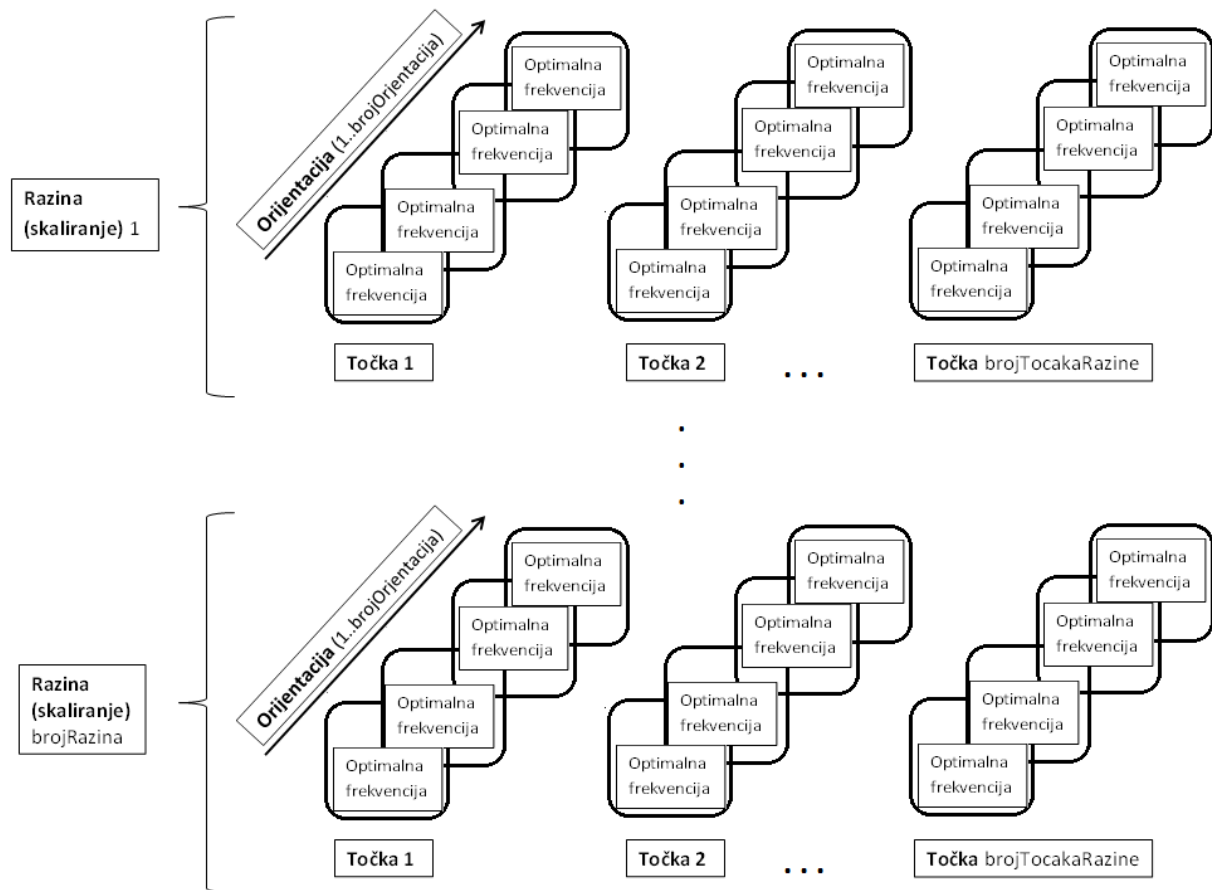
$$\text{Orijentacije } (\theta) = 0^\circ, 45^\circ, 90^\circ, 135^\circ$$

$$\text{Frekvencije } (\omega) = 2, 8, 16 \text{ Hz}$$

$$\text{Skaliranje} = 8, 4, 2$$

$$\text{Broj točaka po razini} = 1, 4, 16$$

Takav postupak je prikazan donjom slikom.



Slika 19. Prikaz ekstrakcije značajki pomoću sloga filtara

1.2.3 Optimiranje filtra

Parametar skaliranja i orijentacije već smo odabrali. Jedini slobodan parametar koji je preostao je frekvencija. Nju (ili bilo koji drugi slobodan parametar) bismo mogli odabrati tako da izvršimo klasifikaciju sa jednom mogućom vrijednosti, zatim s drugom te ponavljamo postupak sa svim frekvencijama koje smo odabrali koje su moguće.

Negativna strana ovoga postupka što ovisi o klasifikatoru te što se broj kombinacija frekvencija povećava eksponencijalno sa brojem Gaborovih filtara.

Predlaže se uporaba kriterijske funkcije linearne diskriminantne analize. Kako je veličina vektora značajki puno veća od broja uzoraka vektori značajki nužni za izračunavanje dobre funkcije ne mogu se izračunati, te stoga preostaje samo prva metoda.

1.3 Značajka

Računanje značajke koja se dobiva od jedne slike kovoluirane sa jednim Gaborovim filtrom zajedničko je svim ekstraktorima značajki. Računanje značajke provodi se prema:

$$V = \frac{1}{N} \left(\sum_N |f(x, y) - m| \right) \quad (9)$$

gdje N označava broj slika, a m srednja vrijednost pojedine slike. Prosječna apsolutna devijacija (*eng. average absolute deviation*) je mjera koja je slična varijanci, ali se prema radu [3] pokazala boljom. Kako je prije navedeno tako dobivene značajke slažu se u 1D vektor značajki.

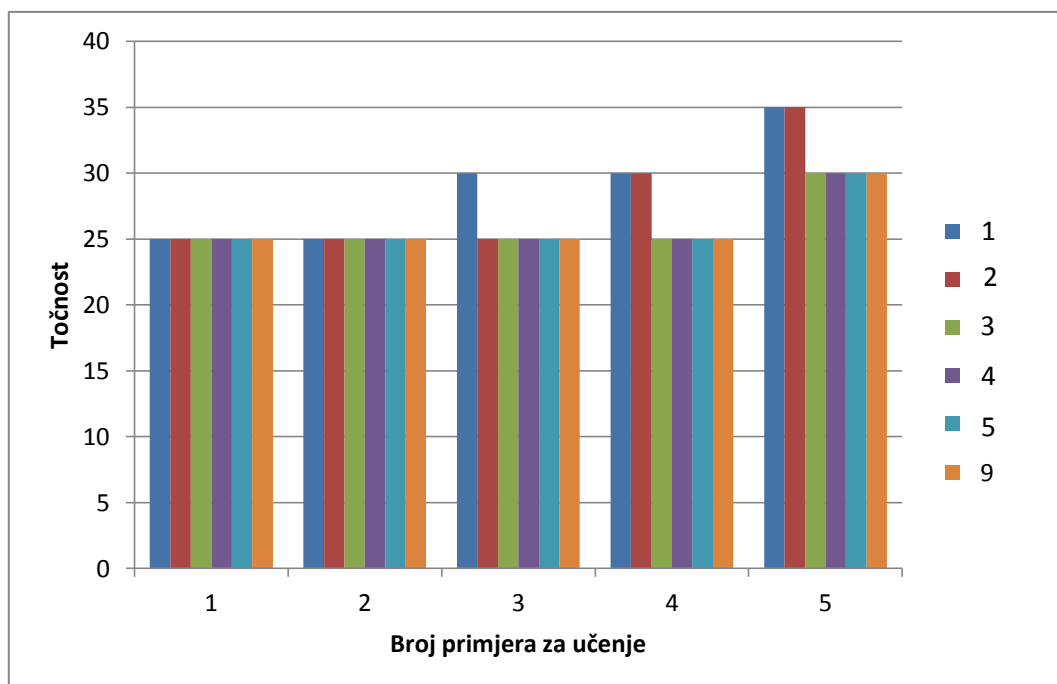
1.4 Klasifikacija

Skup osoba na kojem je vršena klasifikacija je ograničen na 10. Broj slika svake osobe je 7. Dakle cijeli skup za učenje i testiranje sadrži 70 slika. Učenje se obavljalo na 1..5 slika, a testiranje na 2 slike. Slike su slijedno uzimane iz skupa.

Razlog odabira kNN klasifikatora je ta što radovi [3], [4], [5] koriste klasifikaciju takvom metodom uz rezultate točnosti identifikacije >95%. Drugi razlog jest premalen skup uzoraka u odnosu na dimenzionalnost vektora gdje se u takvim slučajevima navedeni klasifikator pokazao vrlo dobrim. Rezultati klasifikacije slijede u nastavku.

1.5 Rezultati

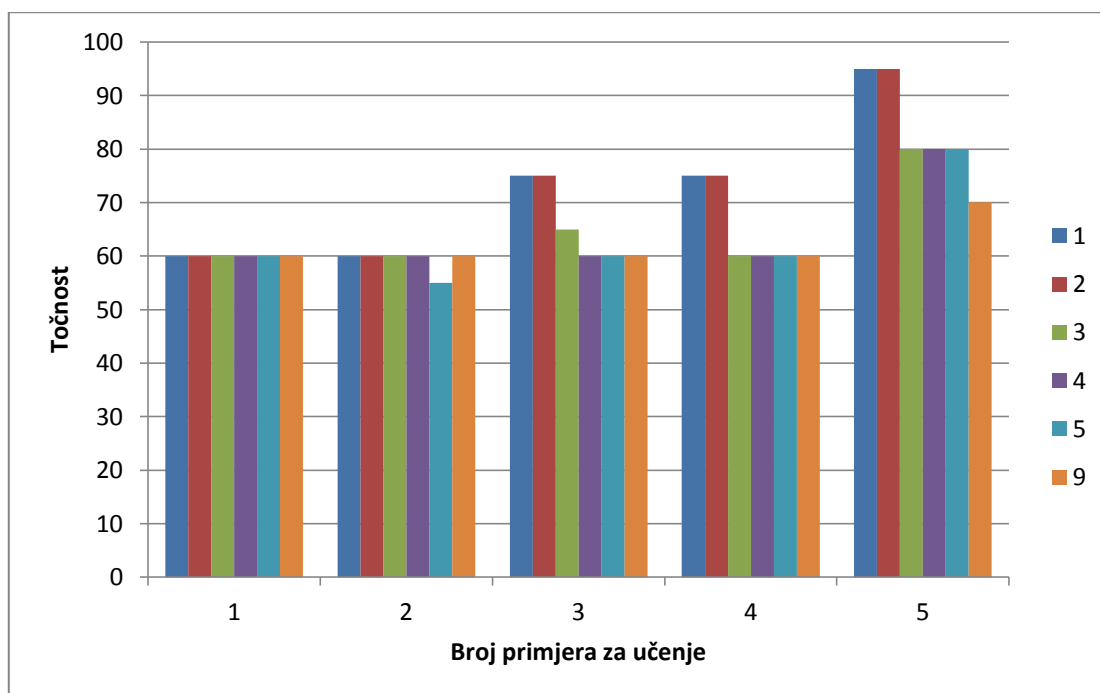
Donjim grafom su prikazani rezultati koji su dobiveni hijerarhijskim ekstraktorom značajki.



Slika 20. Ovisnost klasifikacije o parametru k (parametar kNN klasifikatora) – hijerarhijski ekstraktor

Vrlo slabi rezultati klasifikacije upućuju na neoptimiranost parametara metode (budući da su frekvencije i skaliranja heuristički zadana) ili njezinu neprimjerenost za ovaj problem.

Drugi graf prikazuje rezultate koji su dobiveni standardnim ekstraktorom značajki.



Slika 21. Ovisnost klasifikacije o parametru k (parametar kNN klasifikatora) – standardni ekstraktor

Najbolji rezultat dobiva se učenjem na najvećem skupu i iznosi 95%. Iz slike se može uočiti da je najbolji parametar klasifikatora 1 ili 2 što može značiti „neizgladenost granica klasa“, a time visoku varijancu.

7 Zaključak

U ovom projektu je implementiran sustav za raspoznavanje šarenica oka. Sustav koristi slog Gaborovih filtara za generiranje značajki slike. Evaluiran je standardni ekstraktor značajki koji se pokazao vrlo robusnim na šum što pokazuje i točnost klasifikacije od 95%. Također evaluirana je i metoda ekstrakcije koja se koristi za problem rukom pisanih znakova. Kako je vrlo teško i dugotrajno ručno optimirati veći broj filtara rezultati te metode su dosta lošiji u usporedbi sa rezultatima dobivenih standardnom metodom. Ukupna uspješnost postupka je i više nego zadovoljavajuća.

8 Literatura

Korišteni radovi:

- [1] J. Sung, S.Y. Bang, S. Choi, *A Bayesian network classifier and hierarchical Gabor features for handwritten numeral recognition*, *Pattern recognition letters*, 2006.,
<http://linkinghub.elsevier.com/retrieve/pii/S0167865505001935>
- [2] T.C. Bau, *Using Two-Dimensional Gabor Filters for Handwritten Digit Recognition*,
http://www.ics.uci.edu/~dramanan/teaching/ics273a_winter08/projects/tbau_gabor_handwritten_digit_recognition.pdf
- [3] Ma, L., Wang, Y., Tan, T., *Iris recognition based on multichannel Gabor filtering*,
Proc. Fifth Asian Conf. Computer Vision, 2006.,
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.3967&rep=rep1&type=pdf>
- [4] Zewail, R., Seil, A., Hamdy, N., Saeb, M., *Iris Identification Based on Log-Gabor Filtering*,
Proceedings of the IEEE Midwest Symposium on Circuits, Systems, & Computers, 2003.,
http://www.magdysaeb.net/images/IRISPAPER_v3.pdf
- [5] Carr D., Khabashesku D., Lipinski B, Robichaux P., *Iris recognition*
<http://cnx.org/content/col10256/1.1/>, 22.04.2011.

Pomoćna literatura:

- [6] Petkov, Grigorescu; *2-D Gabor function - interactive visualization*,
<http://www.cs.rug.nl/~imaging/simplecell.html>
- [7] *Web enabled image processing operators*, <http://matlabserver.cs.rug.nl/>
- [8] *Gabor Wavelets*, <http://www.ansatt.hig.no/erikh/papers/scia99/node6.html>
- [9] *Gabor Filter* - <http://www.neuroinformatik.ruhr-uni-bochum.de/VDM/research/computerVision/imageProcessing/wavelets/gabor/gaborFilter.html>
- [10] Kepenekci B.; *Face Recognition Using Gabor Wavelet Transform*, 2001.,
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.6259&rep=rep1&type=pdf>
- [11] *Web aplikacija* - <http://matlabserver.cs.rug.nl/cgi-bin/matweb.exe>
- [12] *Gabor Wavelet transformation*, <http://www.neuroinformatik.ruhr-uni-bochum.de/VDM/research/computerVision/imageProcessing/wavelets/gabor/contents.html>

- [13] Welling, Max; *Fisher Linear Discriminant Analysis.*, Department of Computer Science, University of Toronto, 2001.
- [14] Wikipedia, *k-nearest neighbor algorithm*, <http://en.wikipedia.org/wiki/KNN>
- [15] S. Ribarić, *Predavanja iz kolegija Raspoznavanje uzoraka*, FER, 2010.
- [16] S. Lončarić, *Predavanja iz kolegija Digitalna obrada i analiza slike*, FER, 2011.