

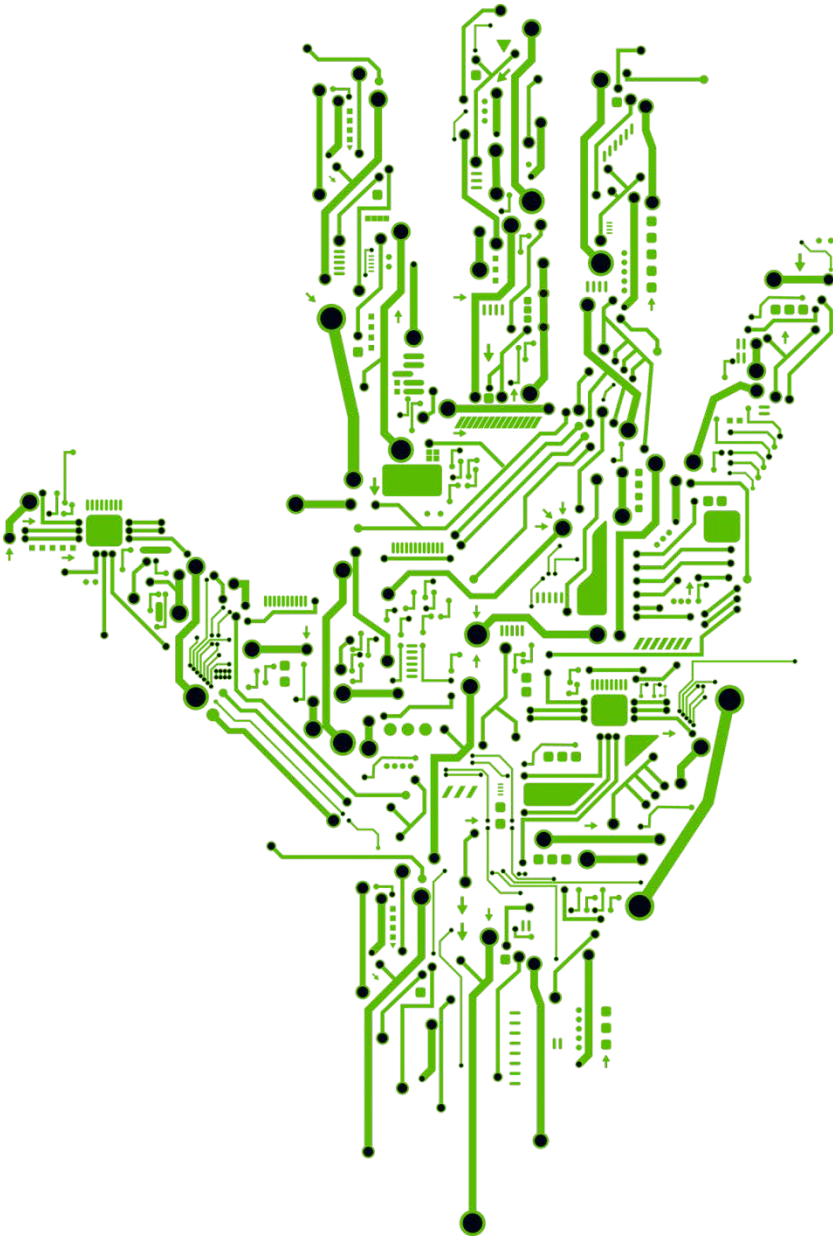
Matematički fakultet, Univerzitet u Beogradu

# ALGORITMI ZA PREPOZNAVANJE GESTIKULACIJA RUKOM

**Master rad**

Bojan Marković

Beograd, 2018



Mentor: prof. dr Miodrag Živković  
Matematički fakultet, Univerzitet u Beogradu

Članovi komisije: prof. dr Vladimir Filipović  
Matematički fakultet, Univerzitet u Beogradu

dr Aleksandar Kartelj, docent  
Matematički fakultet, Univerzitet u Beogradu

Datum odbrane: \_\_\_\_\_

## Apstrakt

Ubrzo nakon pojave računara i uređaja koji su kontrolisani od strane mikro računara, ljudi su u konstantnoj potrazi za novim načinima interakcije između njih. Gestikulacije rukom i njihovo prepoznavanje su imale i imaju svoje važno mesto u ovim interakcijama (*human-computer interaction - HCI*) zbog čovekove prirodne i lake upotrebe ruku u svakodnevnom životu. Danas, primena prepoznavanja gestikulacija ruku je široko rasprostranjena u svakodnevnom životu u kontroli pametnih televizora, dronova i drugih uređaja.

Ovaj rad započinje istorijatom i primerima obrade gestikulacija, upoznaje sa standardnim koracima i algoritmima u njihovoj obradi preko segmentacije kože, morfoloških filtera, karakteristika objekata i njihove ekstrakcije, do korišćenih algoritama za klasifikaciju gestikulacija. Predloženi algoritam je realizovan. Prilikom testiranja postigao je uspešnost od 78% za posture i 93% za gesture.

\*\*\*

Ovom prilikom želim da se zahvalim svom mentoru na savetima, dobronamernoj kritici i pomoći prilikom izbora literature, kao i članovima komisije na korisnim sugestijama. Posebnu zahvalnost dugujem prethodnom mentoru prof. Milanu Tubi pod čijim mentorstvom je urađen veći deo rada. Bez pružene moralne podrške porodice i devojke Ive, izrada ovog rada ne bi bila moguća.

Bojan Marković

Novi Sad, 2018

## Sadržaj

1. Uvod.....	6
2. Istorijat razvoja prepoznavanja gestikulacija rukom .....	8
2.1 Prepoznavanje gestikulacija kompjuterskom vizijom .....	10
2.1.1 Prepoznavanje gestikulacija korišćenjem rukavica sa markerima .....	10
2.1.2 Prepoznavanje gestikulacija rukom isključivo kompjuterskom vizijom .....	11
3. Koraci u prepoznavanju gestikulacija rukom .....	14
3.1 Predobrada.....	14
3.1.1 Segmentacija kože .....	15
3.1.2 Morfološki filteri i operacije.....	18
3.2 Ekstrakcija karakteristika.....	23
3.2.1 Pristup i tehnike zasnovane na obliku .....	24
3.2.2 Pristup i tehnike zasnovane na regionu.....	27
3.3 Klasifikacija gestikulacija .....	29
3.3.1 Mere rastojanja.....	30
3.3.2 Linerani klasifikatori.....	32
3.3.3 Nelinearni klasifikatori .....	34
4 Predloženi algoritam.....	35
4.1 Algoritam za prepoznavanje postura .....	36
4.2 Unapređenje algoritma za prepoznavanje gestura.....	42
5 Realizacija algoritma .....	46
5.1 Aplikacija .....	46
5.2 Dobijeni rezultati .....	48
5.2.1 Rezultati algoritma za prepoznavanje postura .....	48

5.2.2	Rezultati unapređenog algoritma za prepoznavanje gestura.....	51
6	Zaključak.....	55
	Lista slika .....	56
	Lista referenci.....	58

## 1. Uvod

Od nastanka računara, postoji i potreba za interakcijom između čoveka i računara (*eng. Human Computer Interaction - HCI*).

Međutim, pravim početkom interakcije u ovom smislu možemo smatrati 70-te i otkriće kompjuterskog miša i daljinskog upravljača za televizore i ostale električne uređaje.

Smatra se da kontroler koji se koristi u ovoj interakciji treba da ima sledeće osobine i ponudi [1]:

- Pouzdanost u zadavanju komandi
- Reprodukovanje bilo koje komande
- Lakoću korišćenja za čoveka na duže staze
- Kratko vreme učenja za nove korisnike
- Dugotrajnost
- Niske troškove održavanja

Zbog toga ne iznenađuje lojalnost kompjuterskoj tastaturi i mišu velikog procenta modernih korisnika, ali i činjenica da se starije generacije odupiru promenama a da ne postoje radikalno drugačija rešenja sa boljim rezultatima.

Računari i mnogi elektronski uređaji sa kojima se vrši interakcija se danas brzo menjaju sa aplikacijama razvijenim ne samo za ove tradicionalne interakcije. Nove generacije imaju mogućnost da eksperimentišu sa drugačijim kontrolerima, igranje video igara je sve popularnije a tradicionalni kontroleri za neke od njih nisu dovoljno brzi. Tradicionalni kontroleri znaju i da u određenom smislu ograničavaju korisnika i nemaju potrebne mogućnosti za fino i precizno zadavanje komandi.

Kao jedan od novih interfejsa za komunikaciju između računara i čoveka, gestikulacije rukom nude brz, prirodan i lak za učenje način interakcije. Pri tome, nije neophodno držanje uređaja u

ruci, kamere su ili široko dostupne ili već deo električnih uređaja, a pružaju mogućnosti koje nisu dostupne u tradicionalnim interfejsima.

Ovaj rad ima za cilj da demistifikuje prepoznavanje gestikulacija rukom, približi faze i algoritme u njihovom prepoznavanju, kao i da predstavi i predloži jedan od algoritama za prepoznavanje gestikulacija rukom uključujući i rezultate testiranja i njegova potencijalna unapređenja.

Tokom istraživanja i pisanja ovog rada, objavljen je rad „[Vision-Based Algorithm for Real-Time Hand Posture Recognition](#)“ [2] 2012. godine na WSEAS konferenciji, koji je sastavni deo ovog rada.

Poglavlje 2 provodi kroz početke obrada gestikulacija. U poglavlju 3 se raščlanjuju i objašnjavaju koraci i tehnike potrebne za obradu gestikulacija. Predloženi algoritam za obradu gestikulacija je iznet u poglavlju 4, dok se informacije o implementaciji predloženog algoritma i dobijenim rezultatima nalaze u poglavlju 5. Rad se zaključuje poglavljem 6.

## 2. Istorijat razvoja prepoznavanja gestikulacija rukom

Istorija prepoznavanja gestikulacija rukom započinje sa pronalaskom i razvojem interfejsa zasnovanih na rukavicama (eng. Data glove). Naučnici inspirisani znakovnim jezikom su shvatili da njegove gestikulacije mogu da se iskoriste za zadavanje prostih komandi računarima. Prve rukavice bile su opremljene različitim sensorima. Razvoj na ovom polju je omogućio kasnije prepoznavanje zasnovano samo na kompjuterskoj viziji (eng. *Computer Vision*) bez senzora ali uz pomoć posebno obojenih rukavica (različite boje dlana, prstiju).

Prve rukavice su imale žičnu vezu do računara i imale dodirne, optičke i senzore otpora na prstima ili zglobovima rukavice, uz pomoć čijih merenja su određivale da li je ruka otvorena ili ne i koji su uglovi između prstiju. Ove mere su bile vezivane za gesture i interpretirane od strane računara.

Gledajući evoluciju rukavica, mogu se razdeliti na dva tipa:

*Aktivne rukavice* – rukavice sa sensorima za merenje sa žičnom ili bezžičnom vezom. S obzirom na senzore, preciznost i mogućnosti su veće, ali dolaze sa većom cenom i korisnikova sloboda korišćenja je smanjena

*Pasivne rukavice* – rukavice bez senzora, samo sa markerima ili bojama na ključnim delovima koje služe za prepoznavanje korišćenjem kamera. Njihova cena je niska, a korisnik ima prepoznatljivi osećaj nošenja običnih rukavica.

Prvi prototipovi aktivnih rukavica bili su [3]:

*Sayre rukavica* – razvijena 1977.god. Koristila je fleksibilne cevi montirane na prstima sa izvorom svetlosti s jedne i foto ćelijom sa druge, tako da je savijanje prstiju rezultovalo promenljivom jačinom prepozate svetlosti.

*MIT LED rukavica* – razvijena početkom 1980-tih, kao deo LED sistema zasnovanog na kamerama sa ciljem da prati telo i udove za potrebe računarske grafike u realnom vremenu.



*Digital Entry Data rukavica* – dizajnirana i patentirana 1983. god. Opremljena sensorima dodira i udaljenosti, koja je mogla da prepozna 80 jedinstvenih znakova koji su bili mapirani na 96 ASCII karaktera.

Vremenom su počele da se javljaju rukavice i u komercijalne svrhe, gde je jedna od prvih Power rukavica (1989.), koju je komercijalizovao Mattel Intellivision kao kontroler za Nintendo konzolu a ubrzo postala veoma poznata i popularna među igračima.

Danas imamo veoma sofisticirane aktivne rukavice koje sensorima prepoznaju pokrete i gesture, od kojih neke dostižu i par hiljada dolara. Poznate su:

- CyberGlove II (prikazana na slici 1)
- CyberGlove III (prikazana na slici 2)
- X-IST Data Glove
- P5 Glove



Slika 1 - CyberGlove II



Slika 2 - CyberGlove III

## 2.1 Prepoznavanje gestikulacija kompjuterskom vizijom

Paralelno sa eksperimentisanjem i istraživanjem sa aktivnim rukavicama, započelo je i istraživanje prepoznavanja gestikulacija rukom korišćenjem kamera. Problemi koji su postojali u to vreme su slaba moć procesiranja i kamere slabih rezolucija, malog broja frame-ova koji su se mogli zabeležiti kao i optike koja se izražavala kroz probleme vezane za nepostojanost ili nekonzistentnost boja.

Prvi radovi u ovoj oblasti bili su zasnovani na pasivnim rukavicama sa markerima, a kasnije se nastavilo na istraživanju bez rukavica, isključivo uz pomoć kompjuterske vizije (*eng. Computer Vision*) što je preteča današnjeg najrasprostranjenijeg načina prepoznavanja gestikulacija rukom.

### 2.1.1 Prepoznavanje gestikulacija korišćenjem rukavica sa markerima

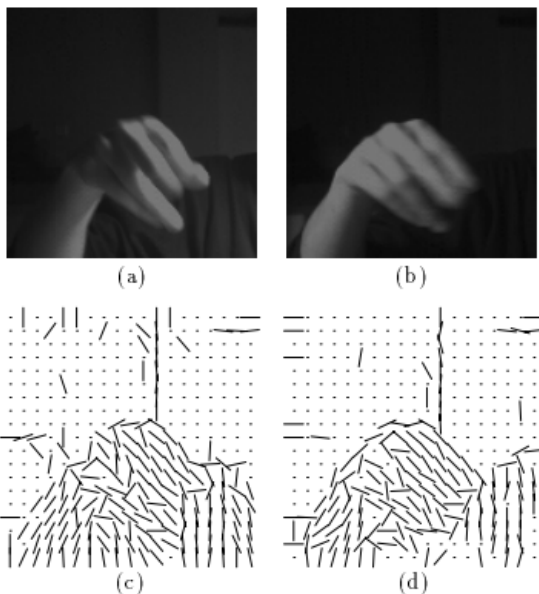
Primer tehnika prepoznavanja gestikulacija korišćenjem rukavica sa markerima na vrhovima prstiju je izložen u [4] gde su autori uz pomoć konačnih automata modelirali stanja/faze gestura, a vektorima trajektorija ih opisivali i pretraživali u statičkim tabelama vrednosti. Celo testiranje je izvedeno u prostoru sivih boja a rezultat je bio prepoznavanje 7 gestura.

Tehniku prepoznavanja korišćenjem rukavica u boji prikazuje rad [5]. Korišćen je metod drveta odlučivanja, a deset regiona šake i prstiju je bilo podeljeno na različite boje. Autori su na taj način pokušali da reše problem okluzije koja je jedan od problema koji pogađa sve metode zasnovane na prepoznavanju uz pomoć kompjuterske vizije. Okluzijom se smatra kada prsti ili šaka u odnosu na tačku kamere, delimično sakriju druge svoje delove i time dodaju posebnu složenost na samo prepoznavanje i klasifikaciju.

### 2.1.2 Prepoznavanje gestikulacija rukom isključivo kompjuterskom vizijom

Postoji veliki broj radova na ovu temu. Kao i u drugim oblastima, problem prepoznavanja gestikulacija je „napadan“ sa različitih strana, različitim tehnikama tražeći onu koja će dati najbolje rezultate. Niže su opisane specifičnosti delova nekih od radova koje daju i dobru predstavu određenih podproblema koji su bili predmet rešavanja.

Jedan od problema koji utiče na prepoznavanje gestura je promenljivo i nejednako osvetljenje koje ima uticaj na prepoznavanje boje kože, predprocesiranje slike i kasniju klasifikaciju. Naučnici Micubišijevе laboratorije za električna istraživanja u svom radu [6] koriste histograme lokalnih orijentacija slike kao vektore reprezentacije koji su se pokazali invarijantni u odnosu na promene osvetljenja. Primer orijentacionih mapa je dat na slici 3, a na ovaj način je urađeno prepoznavanje statičnih gestura na skupu od 10 gestura, koja je svaka trenirana sa po 5-10 primera.

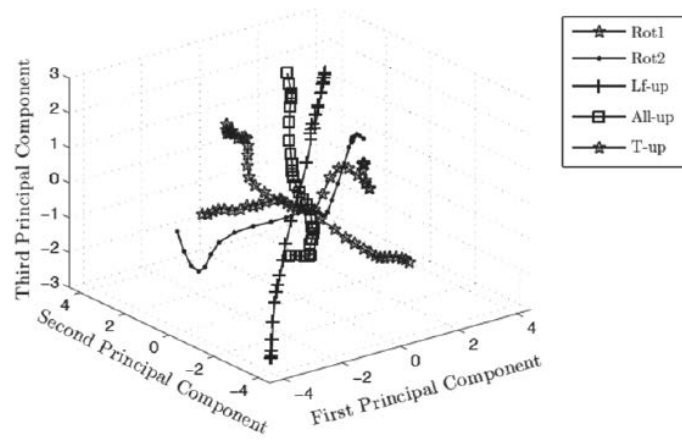


Slika 3 – Orijentacione mape

U prvom redu su prikazane slike jedne gesture pod različitim osvetljenjem, a u drugom orijentacione mape koje su jako slične

Spominjan je ranije problem okluzije koji je težak kada se u prepoznavanju koristi jedna kamera i bez mogućnosti opažanja dubine (kao što ima npr. infracrvena kamera). Grupa istraživača sa odeljenja Kompjuterski kontrolisanih mehaničkih sistema, Univerziteta u Osaki, je u svom radu [7] predložila interesantan pristup. Nakon ekstrakcije siluete sa uhvaćene slike, pronalaze se mogući grubi 3D modeli šake čija silueta odgovara uhvaćenoj. U nizu frejmova, ovoj gruboj estimaciji se koriste opažanja ograničenja vezana za moguće pokrete šake i prethodne estimacije i na taj način se eliminiše veliki skup mogućih rešenja a sve sa ciljem da se dobiju informacije u frejmovima u kojima se okluzije događa. Dodatnim izračunavanjima prostor rešenja se smanjuje. Inače, pokreti ruke se smatraju složenim slučajem s obzirom da ljudska ruka i šaka imaju veliki stepen slobode *eng. Degree of Freedom - DOF*. Autori u zaključku navode da se ovakav pristup može primeniti i na drugim „artikulisanim“ objektima tj. objektima kod kojih postoji razuman broj mogućih pokreta i njihova sledljivost.

Sve je veći broj radova koji koriste dve kamere kako bi dobili informaciju o dubini i imitirali vid koji imaju ljudi, tzv. binokularni, tj. onaj dobijen od dva izvora pri čemu se slika jednog i drugog u određenom delu preklapaju. Jedan od takvih je i [8] u kome autori za prepoznavanje gestura prate, obrađuju i koriste vrhove prstiju. Pet slika u intervalu reprezentuju gesturu. Prikupljene informacije se sumarizuju u jednu komponentu koja se koristi za prepoznavanje gesture. Transformacija se izvršava analizom osnovnih komponenti (*eng. Principal Component Analysis - PCA*). Ova ortogonalna transformacija skup moguće koreliranih vrednosti konvertuje u skup linearno nezavisnih. Posle projektovanja u podprostor dobijen posle primene tehnike, tri vrednosti su bile dovoljne da se sačuva 99% informacije. Jednostavnost je kasnije pomogla u klasifikaciji gesture. Na slici 4 je prikazano pet sekvenci nakon primenjene PCA koje predstavljaju pet gestura.

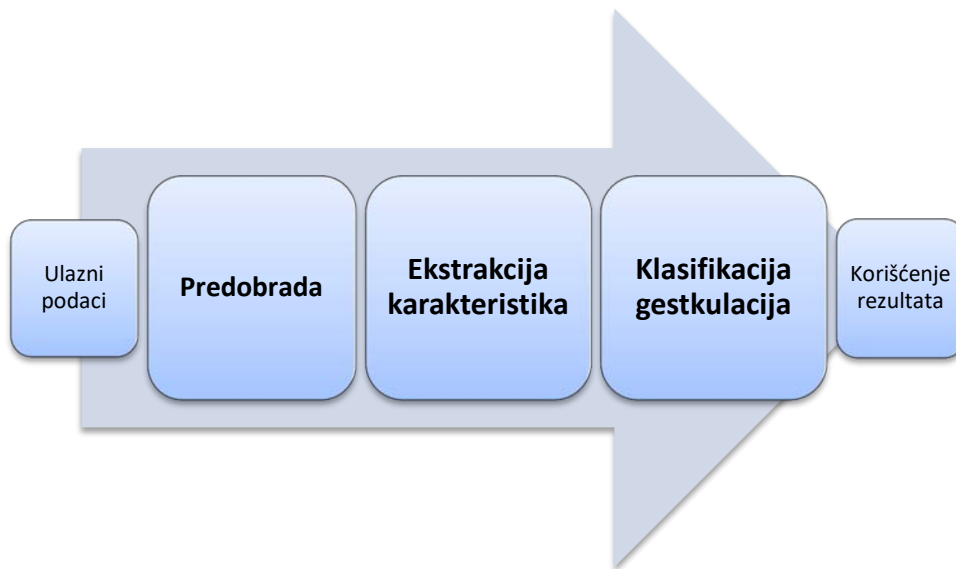


Slika 4 - Projekcija posle PCA

Pet sekvenci koje predstavljaju pet gestura

### 3. Koraci u prepoznavanju gestikulacija rukom

Nakon dobijanja ulaza najčešće u vidu video zapisa, slede koraci u kojima se vrši obrada. Posmatrajući sve radove, kao i potreba karakterističnih za kompjutersku viziju, na visokom nivou se izdvajaju predobrada, ekstrakcija karakteristika i klasifikacija gestikulacija kao standardni koraci u procesu prepoznavanja (Slika 5). Nakon prepoznate gestikulacije, rezultat može da se koristi za zadavanje komandi ili za druge namenjene aktivnosti.



Slika 5 - Standardni koraci u prepoznavanju gestikulacija

#### 3.1 Predobrada

Kompjuterska vizija pokušava da simulira vizualni sistem čoveka. U ovu svrhu se koriste kamere, koje dobijanjem slike o sceni, posledično dobijaju željene i neželjene informacije. U slučaju gestikulacija, željene informacije su one vezane za šaku, a neželjene pozadina, odelo i drugi delovi tela.

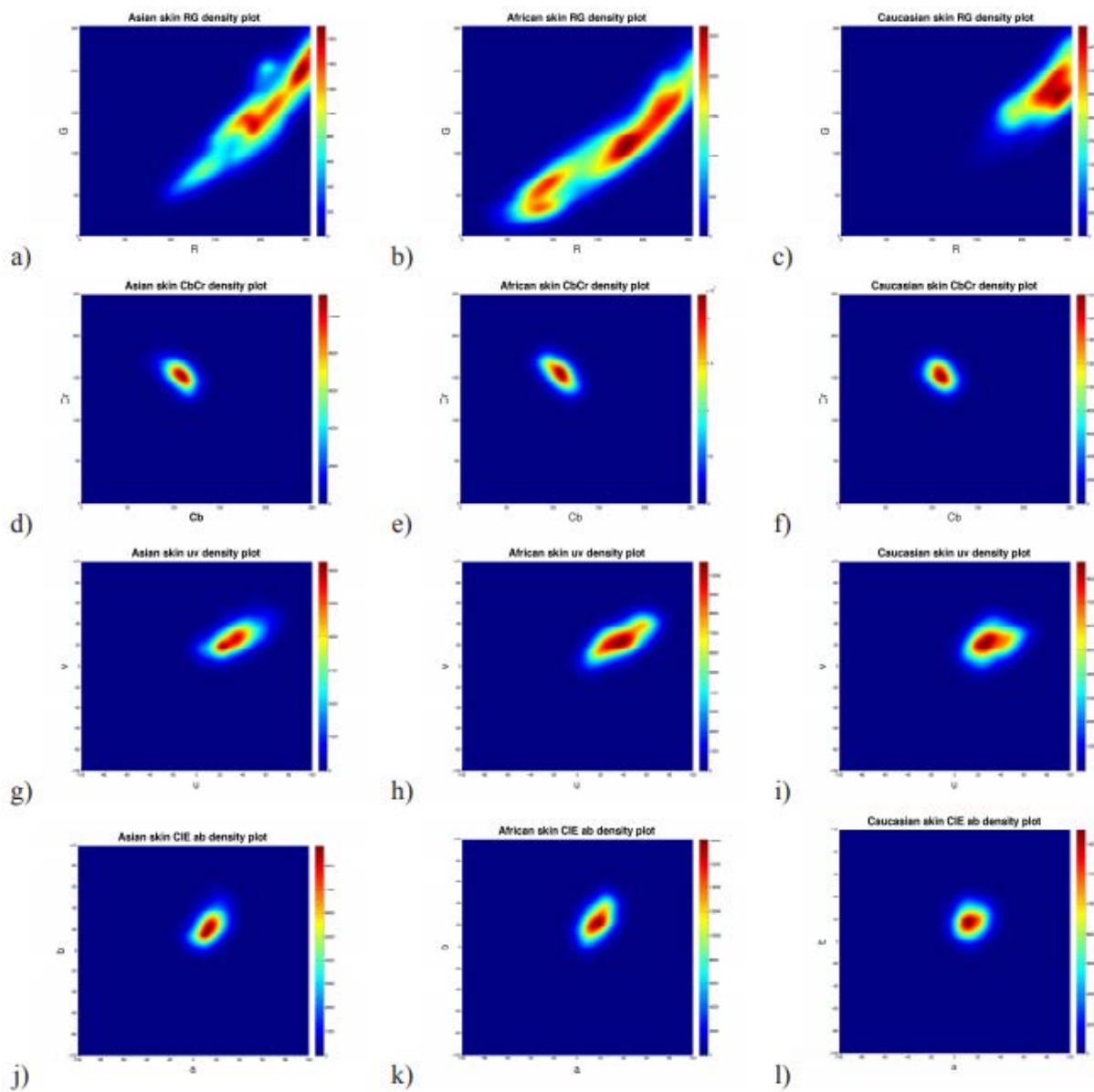
Izdvajanje šake se izvršava prepoznavanjem boje kože koje se ispostavlja kao takođe netrivialan podproblem klasifikacije zvan „segmentacija kože“. Zbog nesavršenosti ovog algoritma, nesavršenosti kamere, kao i činjenice da ista ruka ne izgleda isto u različitim okolnostima, pre svega pod različitim osvetljenjima, skoro uvek je neophodna dodatna obrada pre slanja u sledeći korak prepoznavanja. Dodatna obrada se zasniva na primeni odgovarajućih morfoloških filtera za obradu slike.

### **3.1.1 Segmentacija kože**

Segmentacija kože je proces traženja regiona ili tona kože na digitalnoj slici. U kontrolisanim uslovima nije težak problem, međutim u nekontrolisanim poput obrade slika i video zapisa u realnom vremenu, ispostavlja se da algoritmi klasifikacije moraju da budu jako robusni samim tim i složeniji.

Dodatnu težinu na nekonzistentnost boje koja dolazi od različitih kamera i varijacija osvetljenja, daje i raznolikost tona kože u zavisnosti od rase ljudi.

Boja ljudske kože se i pored svojih varijacija nalazi u ograničenom opsegu s obzirom da je sačinjena od krvi (crvena) i melanina (braon, žuta). Međutim, koji region ona zahvata u različitim prostorima boja kao i koliko su oni kompaktni pokazuje se u radu [9] čiji je deo rezultata grafički prikazan preko grafova gustine na slici 6.



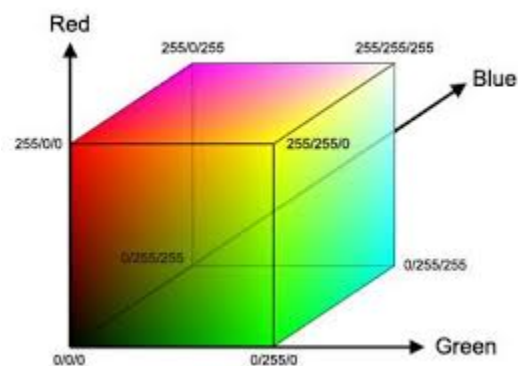
Slika 6 - Grafovi gustine boja kože rasa ljudi u različitim prostorima boja

U kolonama (Azijati, Afrikanci, Bela rasa), u redovima komponente RGB, YCbCr, YUV, CIE prostora boja



## RGB prostor boja

Ovaj prostor boja je najčešće korišćen prostor kod digitalnih slika. On je aditivnog tipa, što znači da kodira boje kao kombinaciju tri osnovne crvene (R), zelene (G) i plave (B) (Slika 7). Jednostavan je, ali razdaljine boja u njemu ne korespondiraju sa čovekovom percepcijom boja. R, G, B komponente su visoko korelirane, a osvetljenje ne postoji kao zaseban činilac u boji.



Slika 7 - RGB prostor boja

Vežano za segmentaciju kože, grafovi a), b) i c) na slici 6 pokazuju da ovaj prostor boja nije pogodan za segmentaciju, jer je region tonova boje kože jako širok. Ograničavanje klasifikacije na ovaj region donosi puno lažno pozitivnih piksela/regiona. Ovo je posledica osvetljenja koje svojom promenom, afektira sve tri komponente ovog prostora.

## YCbCr prostor boja

Ovaj prostor boja spada u drugačiju klasu tj. klasu ortogonalnih prostora boja. YCbCr se koristi u JPEG kompresiji slika i MPEG kompresiji video zapisa.

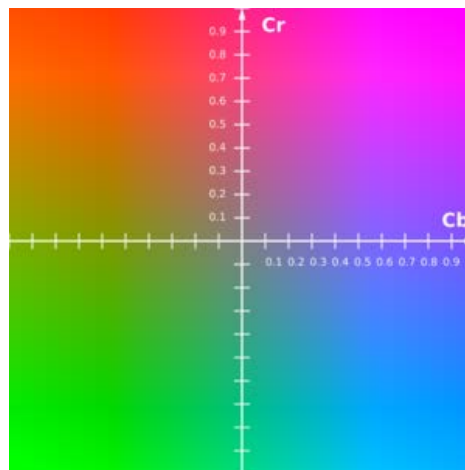
Konverzija iz RGB prostora boja u YCbCr se računa tako što je luminansa (Y) težinska suma RGB komponenti, dok se komponente hrominanse računaju oduzimanjem prethodno dobijene luminanse od plave i crvene komponente RGB prostora boja.

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$$

$$Cb = B - Y$$

$$Cr = R - Y$$

U ovom prostoru boja je izdvojena komponenta inteziteta svetlosti, dok ostale dve komponente određuju karakteristike boje (hrominansa).  $C_b C_r$  ravan sa fiksnim intenzitetom svetlosti je prikazana na slici 8. Iz tog razloga je ovaj prostor pogodan za segmentaciju kože i donosi invarijantnost po pitanju osvetljenja. Dodatno, grafovi d), e) i f) na slici 6 pokazuju kompaktnost regiona tonova kože i njihovo poklapanje između različitih rasa, što je idealno za klasifikatore. Istovetne analize se primenjuju i kod ispitivanja pogodnosti drugih prostora boja.



Slika 8 -  $C_b C_r$  ravan sa konstantim intenzitetom svetlosti  $Y=0.5$

Autori u [9] iznose da većina radova na temu detekcije kože, prijavljuje 95% tačne detekcije kože, dok se netačne detekcije kreću od 15-30%.

Algoritmi za klasifikaciju segmentacije kože mogu biti zasnovani na pikselu ili regionu. Kod algoritama zasnovanih na pikselu, vrednost boje piksela se upoređuje sa ograničanjima segmenta koji se smatra bojom kože u tom prostoru boja. Kod algoritama zasnovanih na regionu, uzimaju se vrednosti i okolnih piksela u formulama klasifikovanja, čime se izbegavaju šumovi i greške i dobijaju bolji rezultati.

### 3.1.2 Morfološki filteri i operacije

Kompjuterska vizija se oslanja na identifikovanje oblika i struktura sa slike. Kada se izoluje oblik kao binarna slika, u našem slučaju šaka korišćenjem algoritama za segmentaciju kože, oblici često imaju nesavršenosti i dolaze sa nepoželjnim šumovima i teksturama. Tada se koriste

morfološki filteri i operacije kako bi se te nesavršenosti uklonile i dobio tačniji i precizniji oblik za dalju obradu.

Morfološki filteri su širok skup nelinearnih operacija za obradu najčešće binarnih slika koji mogu da budu iskorišćeni za procesiranje slika zasnovanim na oblicima. Sve ove operacije koriste tzv. „strukturni element“ različitih veličina i oblike uz pomoć koga se ulazne slike transformišu u izlazne najčešće iste veličine. Strukturni element je matrica koja se sastoji samo od 0 i 1. 1-ice predstavljaju susedne piksele koji ulaze u obradu dok centar predstavlja piksel čiji se rezultat izračunava.

Iako mogu biti primenjeni i na slikama iz prostora sivih, morfološki filteri su po prirodi binarni. Osnovni morfološki filteri su [10]:

- Erozija
- Dilatacija
- Otvaranje
- Zatvaranje
- Istanjivanje
- Podebljavanje

### **Erozija**

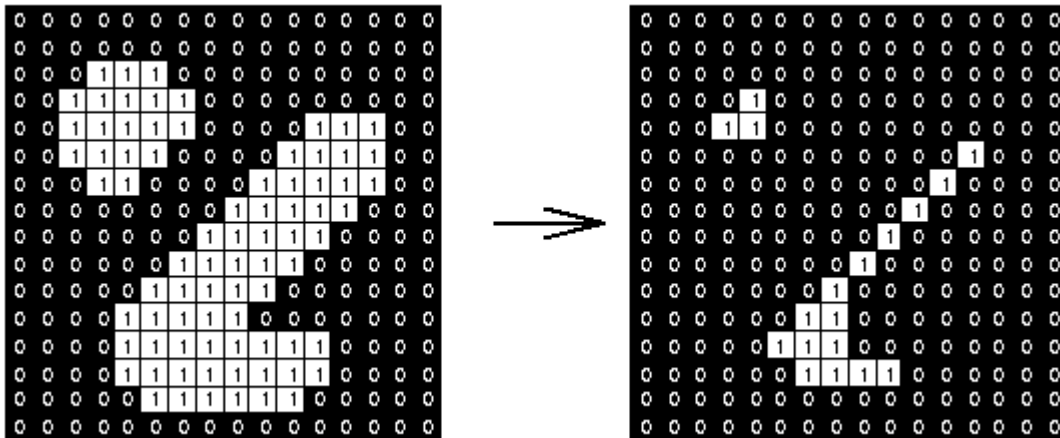
Uz dilataciju, najjednostavnija morfološka operacija. Uklanja piksele na granicama objekata. Vrednost izlaznog piksela je minimum vrednosti svih susednih piksela označenih strukturnim elementom. Znači, ukoliko je bilo koji susedni piksel 0, to je i vrednost izlaznog piksela.

Matematička definicija erozije je

$$I \ominus S = \{I | S_I \subseteq I\}$$

gde je I slika, S strukturni element a  $S_I$  S transliran sa I.

Primer izvršene erozije je dat na slici 9.



Slika 9 - Erozijska nad reprezentacijom binarne slike sa 3x3 jediničnim strukturnim elementom

### Dilatacija

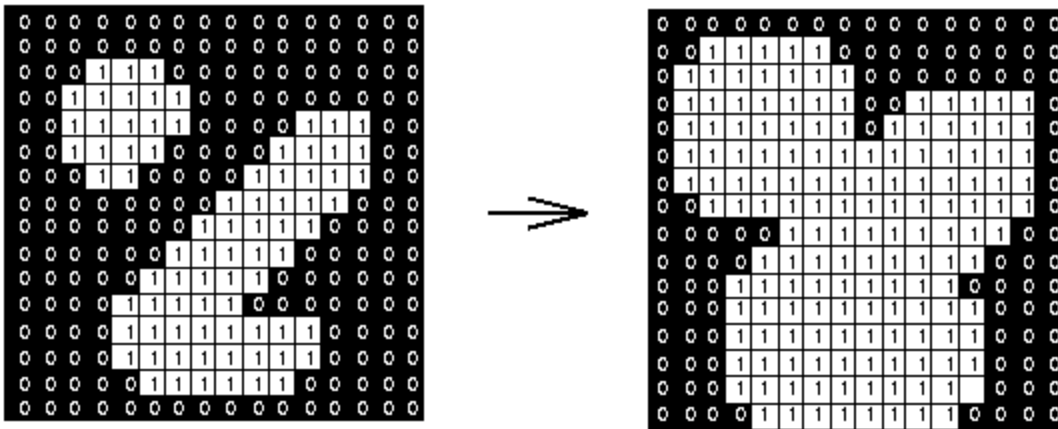
Dodaje piksele na granicama objekata. Vrednost izlaznog piksela je je maximum vrednosti svih susednih piksela označenih strukturnim elementom. Znači, ukoliko je bilo koji susedni piksel 1, to će biti i vrednost izlaznog piksela.

Matematička definicija dilatacije je

$$I \oplus S = (I^c \ominus S)^c$$

gde je  $I^c$  komplement I.

Primer izvršene dilatacije je dat na slici 10.



Slika 10 - Dilatacija nad reprezentacijom binarne slike sa 3x3 jediničnim strukturnim elementom

## Otvaranje

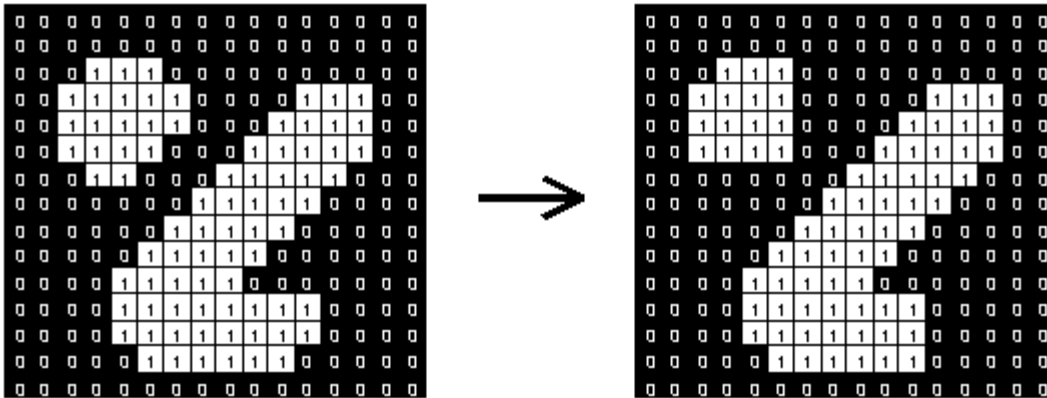
Otvaranje može da se posmatra kao da otvara rupu između objekata koji su povezani tankim „mostom“ piksela. Sastoji se od erozije za kojom se primenjuje dilatacija pri čemu se strukturni element rotira za 180 stepeni. Zbog toga se svi regioni koji prežive eroziju, dilatacijom vraćaju u originalno stanje. Koriste se strukturni elementi različitih dimenzija kako bi uklonili šumove različitih oblika.

Matematička definicija otvaranja je

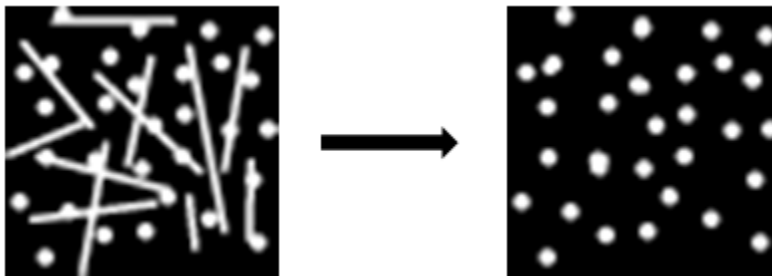
$$I \circ S = (I \ominus S) \oplus S$$

gde je  $\circ$  matematički simbol za otvaranje.

Primer izvršenog otvaranja sa jediničnim strukturnim elementom je dat na slici 11, a otvaranje sa strukturnim elementom oblika diska na slici 12.



Slika 11 - Otvaranje nad reprezentacijom binarne slike sa 3x3 jediničnim strukturnim elementom



Slika 12 - Otvaranje nad binarnom slikom sa strukturnim elementom oblika diska

### Zatvaranje

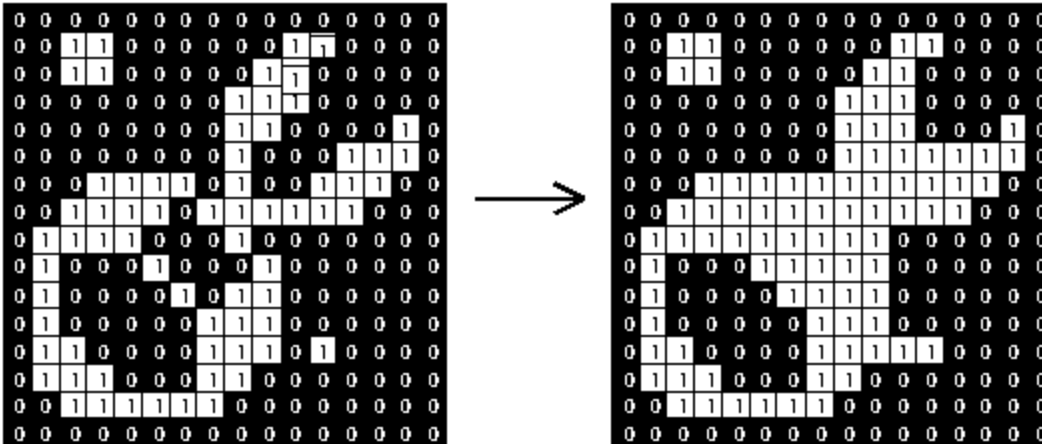
Zatvaranje popunjava rupe u regionima. Sastoji se od dilatacije za kojom se primenjuje erozija.

Matematička definicija zatvaranja je

$$I \cdot S = (I \oplus S) \ominus S$$

gde je  $\cdot$  matematički simbol za zatvaranje.

Primer izvršenog zatvaranja je dat na slici 13.



Slika 13 - Zatvaranje nad reprezentacijom binarne slike sa 3x3 jediničnim strukturnim elementom

### Istanjivanje i podebljavanje

Istanjuju i podebljavaju granice objekata i koriste drugu operaciju tzv. „hit and miss“ koja traži tačno određene strukture koristeći specifične odgovarajuće strukturne elemente koje sekvencijalno primenjuje.

## 3.2 Ekstrakcija karakteristika

Nakon predobrade, potrebno je opisati karakteristike objekata, u domenu rada objekta šake, kako bi se na osnovu ekstrahovanih karkarakteristika izvršila dalja klasifikacija tj. poređenje i prepoznavanje.

Međutim, kako bi karakteristike bile korisne za prepoznavanje, one trebaju da imaju sledeća četiri svojstva [11]:

1. Trebaju da definišu kompletan set, tako da dva objekta moraju da imaju iste karakteristike, ako i samo ako imaju isti oblik.
2. Trebaju da budu kongruentni, u smislu da slični objekti imaju slične karakteristike.

3. Poželjno je da budu invarijanti. Invarijantnost se može posmatrati iz više aspekata. Aspekata rotacije, sklaliranja i pozicije, a ukoliko se objekat posmatra i iz različitih uglova, onda i aspekata afinih i perspektivnih promena.
4. Trebaju da predstavljaju kompaktn skup, tj. da suštinu objekta prikazuju na efikasan način. Drugim rečima, da sadrže samo informacije o onome što objekat čini unikatnim i drugačijim od drugih objekata.

Ne postoje karakteristike sa svojstvima kompletnosti i kompaktnosti koje mogu da opisuju na ovaj način generičke objekte. Iz tog razloga, svojstva se pažljivo biraju u zavisnosti od primene i specifičnosti objekata koji se obrađuju.

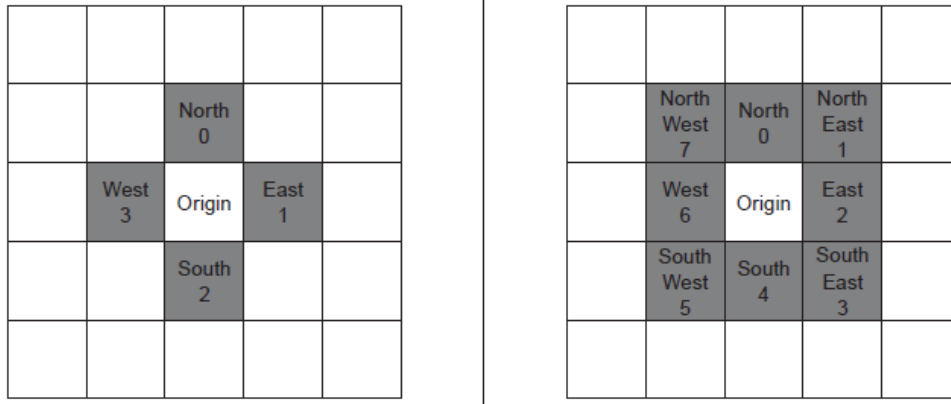
Za karakterizaciju objekata postoje dva pristupa, preko karakteristika oblika tj. konture ili preko karakteristika regiona koji objekat predstavlja.

### **3.2.1 Pristup i tehnike zasnovane na obliku**

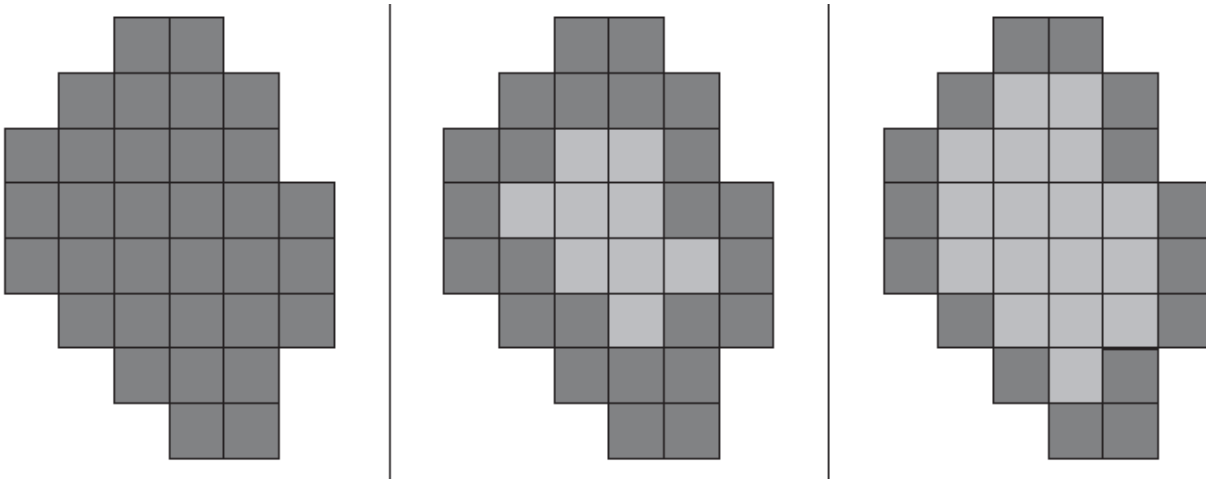
Region obično opisuje sadržaj tj. unutrašnje tačke koje su okružene sa granicom koja se naziva konturom regiona. Na formu konture se generalno posmatra kao na oblik. Tačka na konturi se definiše kao ona koja pripada regionu a sadrži bar jednu susednu tačku koja ne pripada regionu. Kontura se nalazi tako što se nađe jedna tačka na njoj a zatim u smeru kazaljke na satu ili suprotnom, nalazi sledeća.

Relacija susedstva tačaka/piksela, se opisuje pravilima povezanosti. Dva standardna načina su tzv. 4-povezanost kod koje se samo analiziraju neposredni susedi i 8-povezanost kod koje se analiziraju sve tačke/pikseli koje okružuju analiziranu tačku/piksel. Standardne oznake oba tipa povezanosti su date na slici 14. Konture mogu biti definisane na oba načina, a primer na istom regionu je prikazan na slici 15.





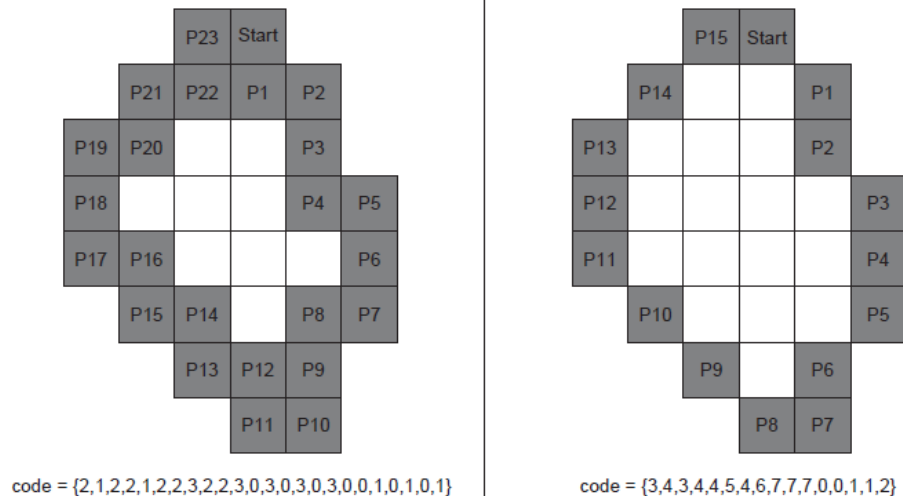
Slika 14 - 4-povezanost i 8-povezanost sa smerovima i oznakama



Slika 15 - Region, kontura 4-povezana i kontura 8-povezana

### Lanci kodova

Suština ove tehnike reprezentovanja konture je pisanje relativnih pozicija između uzastopnih tačaka/piksela na konturi u izabranoj povezanosti. Drugim rečima, skup piksela na konturi se translira u vektor konekcija između njih. Korišćenjem kodova sa slike 14, na sledećoj slici možemo videti konture sa sekvencom piksela ( $P_i$ ) i njihovim dobijenim kodovima u 4 i 8-povezanosti.



Slika 16 - Lanci kodova (levo: 4-povezanost, desno: 8-povezanost)

Lanci kodova očigledno zavise od početne tačke. Invarijantnost u odnosu na početnu tačku se može postići tako što se ceo kod posmatra kao reprezentacija celog broja i ciklično se pomera dok se ne ustanovi najmanji ceo broj. Ako je kod uzet za primer 230456, sledećim cilindričnim pomeranjem se dolazi do rezultata 230456 -> 304562 -> **045623**. Invarijantnost po pitanju rotacije se postiže predstavljanjem koda kao relativnih razlika oznaka u njemu.

Invarijantnost na skaliranje se može postići reemplovanjem granice pre kodiranja, ali ova karakteristika kao i izmene u reprezentaciji koda koji se javljaju sa šumom na slici, ograničavaju upotrebu ove tehnike čija je jednostavnost ipak najveća prednost.

### **Furijeovi deskriptori**

U pitanju je primena Furijeove teorije u opisivanju oblika. Ideja je se kontura opiše malim skupom brojeva (Furijeovih koeficijenata) koji reprezentuju frekvenciju sadržaja celokupnog oblika. Između ostalih prednosti, ovakav pristup je invarijantan na šum. Pikseli se posmatraju kao uzorci neprekidne krive, a formula Furijeovog razvoja koristi integrale neprekidnih krivi, pa se u diskretnom slučaju koristi aproksimacija numeričkim integraljenjem.

Ako se kriva predstavi kao

$$c(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos(k\omega t) + b_k \sin(k\omega t))$$

Furijeovi deskriptori  $a_k$  i  $b_k$  se dobijaju sledećim formulama

$$a_k = \frac{2}{T} \int_0^T c(t) \cos(k\omega t) dt \quad i \quad b_k = \frac{2}{T} \int_0^T c(t) \sin(k\omega t) dt$$

Furijeovi deskriptori se mogu definisati na načine da budu invarijanti na rotiranje, skaliranje i translaciju.

### 3.2.2 Pristup i tehnike zasnovane na regionu

Tehnike ovog tipa opisuju regione i mogu se podeliti na one koje koriste geometrijska svojstva regiona i one koje se koncentrišu na gustinu regiona.

#### Osnovni geometrijski deskriptori

Slede neki od osnovnih deskriptora koji se biraju i obično koriste zajedno kako bi opisali i karakterisali region.

*Površina* regiona se može izračunati sa

$$A(S) = \sum_x \sum_y I(x, y) \Delta A$$

gde je  $I(x, y) = 1$  ako je piksel unutar regiona, a 0 u suprotnom,  $\Delta A$  oblast piksela.

Ovaj deskriptor je invarijantan pri rotaciji ali se menja usled promene veličine regiona.

*Obim* se aproksimira sa

$$P(S) = \sum_i \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

gde su  $x_i$  i  $y_i$  koordinate i-tog piksela.

*Kompaktnost* koristi prethodno izračunat obim i površinu

$$C(S) = \frac{A(s)}{P^2(s)/4\pi}$$

Kompaktnost meri odnos površine regiona i površine kruga sa obimom jednakim obimu regiona. Maximalna kompaktnost je 1 i ima je krug.

Neki od ostalih deskriptora su disperzija regiona, Pointkare-ova mera (broj rupa u regionu), Ojlerov broj (razlika između broja povezanih regiona i broja rupa u njima), koveksnost/konkavnost itd.

### **Momenti**

Momenti opisuju raspored oblika, po prirodi su globalni deskriptori oblika sa istom prednosti filtriranja mogućeg šuma oblika kao i Furijeovi.

*Dvodimenzionalni Dekratov momenat* može biti od 0-tog reda pa naviše. Aproximacija formule momenta reda p i q funkcije  $I(x,y)$  je

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y) \Delta A$$

Centar oblika se naziva centar mase  $(\bar{x}, \bar{y})$  i definiše preko momenta reda 0  $m_{00}$  i dva momenta prvog reda  $m_{10}, m_{01}$  sa

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

Kako bi momenti bili invarijantni na translaciju, izvode se *centralizovani momenti*

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y) \Delta A$$

Normalizacija se vrši radi dobijanja invarijantnosti na skalabilnost i tada se izvode *normalizovani centralni momenti*

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{oo}^\gamma} \quad \text{gde je } \gamma = \frac{p+q}{2} + 1, \forall p+q \geq 2$$

Na osnovu  $\eta_{pq}$  postoji sedam tzv. *Huovih momenata* koji su dodatno invarijantni na rotaciju.

$$h_1 = \eta_{20} + \eta_{02}$$

$$h_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$h_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$h_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$h_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2)$$

$$h_6 = (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$h_7 = (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2)$$

Ovi momenti ipak nisu ortogonalni i ukoliko se želi i ovo svojstvo i potencijalna mogućnost za smanjenje skupa momenata potrebnih za precizno opisivanje, uvode se *Zenrike momenti* koji se izvode u polarnom koordinatnom sistemu.

Momenti sa svojstvima invarijantnosti su pokazali dobre rezultate u različitim primenama i oblastima kompjuterske grafike i vizije.

### 3.3 Klasifikacija gestikulacija

Nakon ekstrakcije karakteristika/deskriptora prepoznatog objekta, problem prepoznavanja gestikulacija se pretvara u pronalaženje klase kojoj vektor deskriptora pripada, tj. problem klasifikacije. S obzirom na veći broj deskriptora koji obično opisuje objekat, klasifikacija se vrši u odgovarajućem prostoru viših dimenzija. Klasifikacija, kao i tehnike koje se primenjuju, može biti linearna i nelinearna. Ukoliko se ne može prepoznati priroda podataka, najčešće se

isprobava sa linearnim klasifikatorima a ako se ispostavi da se klase ne mogu linearno razdvojiti, prelazi se na preformansno zahtevnije i složenije nelinearne klasifikatore.

Često se zbog velikog broja klasifikatora i potrebnog izračunavanja u više dimenzionom prostoru, pribegava tehnikama za smanjenje dimenzionalnosti kao što su analiza osnovnih komponenti (*eng. Principal Component Analysis - PCA*) i linarna diskriminantna analiza (*eng. Linear Discriminant Analysis - LDA*), nakon čega se tek vrši klasifikacija.

### 3.3.1 Mere rastojanja

Pripadnost neke tačke klasteru tačaka se može posmatrati kroz rastojanje te tačke od reprezentativnih tačaka klastera u više dimenzionom prostoru. Različite mere rastojanja se koriste samostalno ili u algoritmima klasifikacije i imaju značajnu ulogu u istim. Ukoliko su dva vektora u n-dimenzionom prostoru  $V_1(a_1, a_2, \dots, a_n), V_2(b_1, b_2, \dots, b_n)$ , neke od mera rastojanja su sledeća.

#### **Euklidsko rastojanje**

Direktna razdaljina dve tačke u više dimenzionom prostoru i najjednostavnija mera u upotrebi.

$$D_{Euclid} = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2}$$

#### **Menheten rastojanje**

Mera rastojanja drugačije zvana i razdaljina gradskih blokova, koja podrazumeva da je od jedne do druge tačke moguće doći samo pomerajima paralelnim sa osama višedimenzionog sistema bez dijagonalnih pomeraja.

$$D_{Manhattan} = |a_1 - b_1| + |a_2 - b_2| + \dots + |a_n - b_n|$$

Praktičan primer Euklidskog i Menheten rastojanja je dat na karti prikazanoj Slika 17.




Slika 17 - Ilustracija Euklidskog (zelenim) i Menheten (plavim) rastojanja

### Čebiševljevo rastojanje

Predstavlja maksimalno rastojanje koje bilo koji par koordinata iste dimenzije ima. Naziva se i šahovsko rastojanje, zato što je u šahu minimalni broj pomeraja kralja od jednog do drugog polja jednak Čebiševljevom rastojanju između centara jednog i drugog polja u 2D prostoru (Slika 18).

$$D_{Chebyshev} = \max(|a_1 - b_1|, |a_2 - b_2|, \dots, |a_n - b_n|)$$

	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1		1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

Slika 18 - Ilustracija Čebiševljevog rastojanja

### **Rastojanje Minkovskog**

Metrika koja u svojim specijalnim slučajevima, daje prethodna 3 rastojanja.

$$D_{Minkowski} = (|a_1 - b_1|^p + |a_2 - b_2|^p + \dots + |a_n - b_n|^p)^{1/p}$$

### **Mahalanobisovo rastojanje**

Rastojanje invarijantno na skaliranje, koje meri sličnost između skupova uzimajući u obzir korelacije u skupu.

$$D_{Mahalanobis} = (a - b)^T C^{-1} (a - b)$$

gde je matrica kovarijanse  $C = \sum_{i=1}^n \frac{(a_i - \bar{a})(b_i - \bar{b})}{n-1}$ .

### **3.3.2 Linerani klasifikatori**

Linearni klasifikatori se koriste kada je moguće skupove linearno podeliti ili pri linearnoj podeli imati grešku koja je prihvatljiva. Izložena su dva primera linearnih klasifikatora koji su detaljnije opisani u [12] i [13].

#### **Perceptron**

Perceptron je algoritam tipa nadgledanog učenja koji klasifikuje ulaz korišćenjem linearne (aktivacione) funkcije zajedno sa skupom težinskih faktora dajući izlaz u vidu klase klasifikacije. Često se posmatra kao specifičan najjednostavniji slučaj algoritama neuronskih mreža. Praktično, on definiše hiper ravan koja na optimalan način razdvaja dve klase vektora.

Tipična aktivirajuća funkcija koje se uzima je

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$



Trening se započinje sa težinskim faktorima jednakim nuli i jedan po jedan se klasifikuje tako da kada je vrednost klasifikacije

$$y = \arg \max w_y \cdot f(x)$$

tačna, težine ostaju iste a u slučaju pogrešne klasifikacije koriguju

$$w_y = w_y \pm f(x)$$

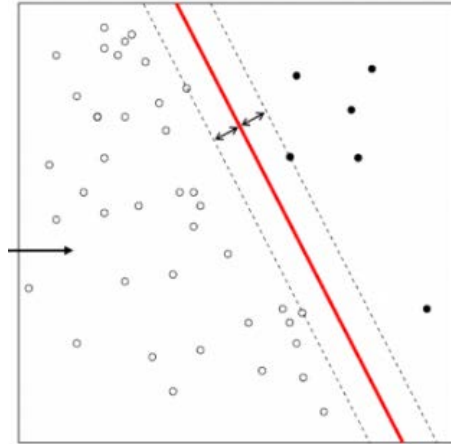
u zavisnosti od toga da li su težinski faktori tačni ili ne.

Zbog svog načina učenja, ovaj algoritam se nikada ne završava ukoliko vektori sa kojima se vrši treniranje nisu linearno razdeljivi.

### **Sistem linearnih potpornih vektora**

Sistem linearnih potpornih vektora (*eng. Support Vector Machine - SVM*) je takođe primer algoritama sa nadgledanim učenjem, sa sposobnosti da precizno i fleksibilno obrađuje vektore velikih dimenzionalnosti. Pored primena u klasifikaciji gestikulacija, između ostalih su poznate njegove primene i u klasifikaciji tekstova i klasifikaciji ručno pisanih brojeva.

Ukratko, SVM pokušava da pronađe hiper ravan sa maksimalnom mogućom marginom koja deli dva skupa različite klasifikacije trening podataka i kasnije je koristi pri klasifikovanju novih ulaznih podataka.



Slika 19 - SVM, prava sa maksimalnom marginom između dva skupa u 2D

SVM je inicijalno definisan kao algoritam za binarno odlučivanje, međutim postoje dosta njegovih modifikacija koje podržavaju više klasne klasifikacije. Postoji njegova linerna i nelinearna verzija.

### 3.3.3 Nelinearni klasifikatori

U prepoznavanju gestikulacija se predominantno koriste nelinearni klasifikatori. Obično su to modifikacije:

- Stabla odlučivanja
- K najbližih suseda
- Nelinearnih SVM
- Neuralnih mreža

Upotreba SVM-a se može videti u npr. [14], [15], a neuralnih mreža u [16].

## 4 Predloženi algoritam

Algoritam predložen u ovom radu predstavlja rezultat autorovog istraživanja i analize literature i naučnih radova iz ove oblasti, kao i praktičnog testiranja i upoređivanja. Cilj je bio kreiranje i testiranje algoritma koji može da radi u realnom vremenu obrađujući video ulaz i ima dobre rezultate, a sa druge strane ne bude isuviše zahtevan kako bi mogao da se implementira i koristi na različitim elektronskim uređajima koji su danas u upotrebi.

U literaturi se koriste različiti pojmovi i definicije vezane za gestikulacije rukom. Za ovaj rad je izabran onaj koji gestikulacije deli na statične gestikulacije - posture i dinamičke - gesture. Pod posturama se smatraju gestikulacije koje nemaju komponentu pomeranja ruke koja se obrađuje, a pod gesturama one koji obrađuju i ovu komponentu.

Za predstavljanje algoritma izabrano je predstavljanje po delovima u vidu algoritma za prepoznavanje postura izloženom u poglavlju 4.1 i njegove nadogradnje za prepoznavanje gestura izložene u poglavlju 4.2. Više je razloga za ovakav pristup:

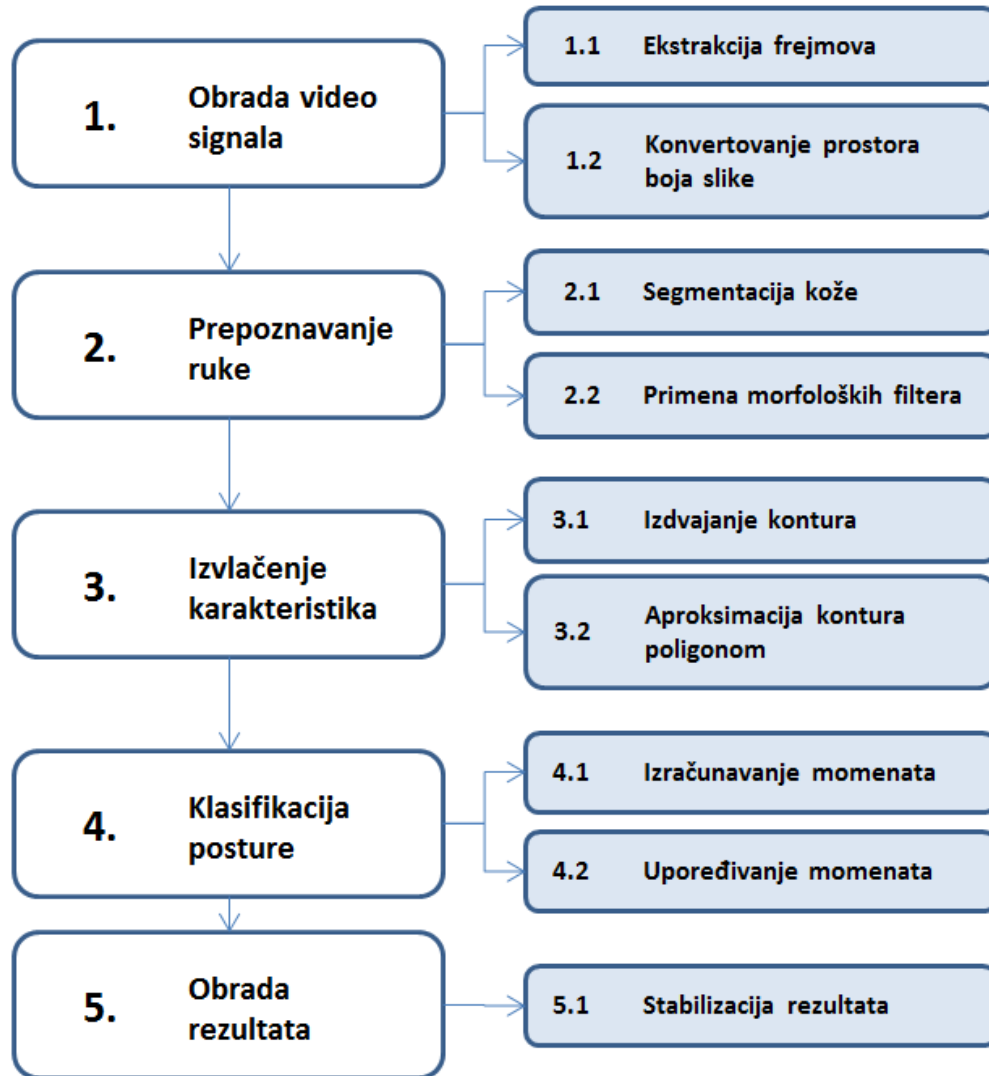
- Istraživanje i analiza za konferencijski rad [2] su bile na temu postura, nakon čega su nastavljene i za gesture
- Predloženi način obrade gestura zaista prirodno predstavlja nadogradnju na deo prepoznavanja postura
- Upotrebna vrednost je povećana, s obzirom da se u zavisnosti od slučaja upotrebe, može implementirati samo deo algoritma za posturu ili samo deo algoritma za posture uz potreban dodatak.
- Rezultati se mogu pratiti zasebno po delovima, unapređenja se mogu izvršavati zasebno, a posledično se dobijati bolji rezultat celokupnog algoritma

Algoritam pretpostavlja i podrazumeva ispunjenost dva uslova:

- Da je od dela ruke vidljiva samo šaka tj. nošenje dugih rukava pri izvođenju gestikulacija. Postoji način popularno zvan „*wrist cutting algorithm*“ kojim se i ovaj uslov može otkloniti, ali on u ovom algoritmu nije obrađen.
- Da je u slučaju više objekata sa bojom kože npr. lica, šaka objekat najveće površine. I ovaj uslov se može otkloniti obradom svih objekata umesto samo jednog i uvođenjem obavezne granice sličnosti modelima, ali u ovom algoritmu isto nije obrađeno.

#### **4.1 Algoritam za prepoznavanje postura**

Algoritam za prepoznavanje postura predstavlja osnovu predloženog algoritma i obrađuje gestikulacije dobijene putem video signala u realnom vremenu. Slika 20 prikazuje korake i podkorake algoritma, za kojom sledi opis svih podkoraka.



Slika 20 - Algoritam za prepoznavanje postura

### **Korak 1.1**

Video predstavlja niz slika u vremenu, u kontekstu videa najčešće nazivane *frejmovi*. Ovaj algoritam predlaže obradu svakog frejma pojedinačno, što nas dovodi u sferu obrade slike. Slika je predstavljena kao matrična struktura piksela, koji u zavisnosti od prostora boja imaju svoju strukturu neophodnu za predstavljanje komponenti prostora boja i njihovih vrednosti.

## **Korak 1.2**

Frejmovi najčešće dolaze u RGB prostoru boja koji nije pogodan za segmentaciju kože kako je objašnjeno u poglavlju 3.1.1. Stoga je potrebno izvršiti konverziju u  $YC_bC_r$  prostor boja formulom prikazanom u 3.1.1.

## **Korak 2.1**

Problematika segmentacija kože je objašnjena u 3.1.1. Za ovaj algoritam je izabrana segmentacija na nivou piksela. Granice u  $YC_bC_r$  prostoru boja koje se smatraju bojom kože su empiriski pronađene i približno jednake u svim radovima, uključujući i novije npr. [17] i [18].

Vrši se binarizacija slike (detekcija i segmentacija) postavljanjem piksela u belo ukoliko je njegova  $YC_bC_r$  vrednost u propisanim granicama ili u crno u suprotnom.

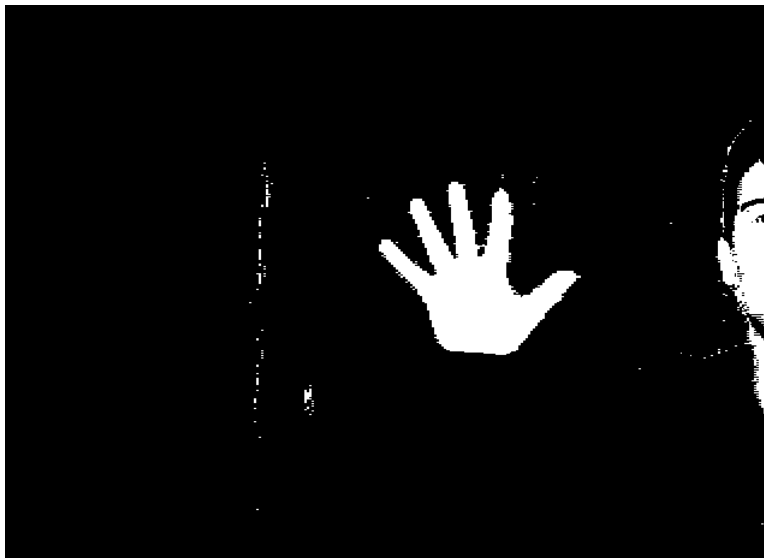
Za granice su uzete one iz [2]:

$$50 \leq Y \leq 255$$

$$130 \leq Cr \leq 185$$

$$80 \leq Cb \leq 135$$

a dobijeni rezultati su prikazani na slici 21.



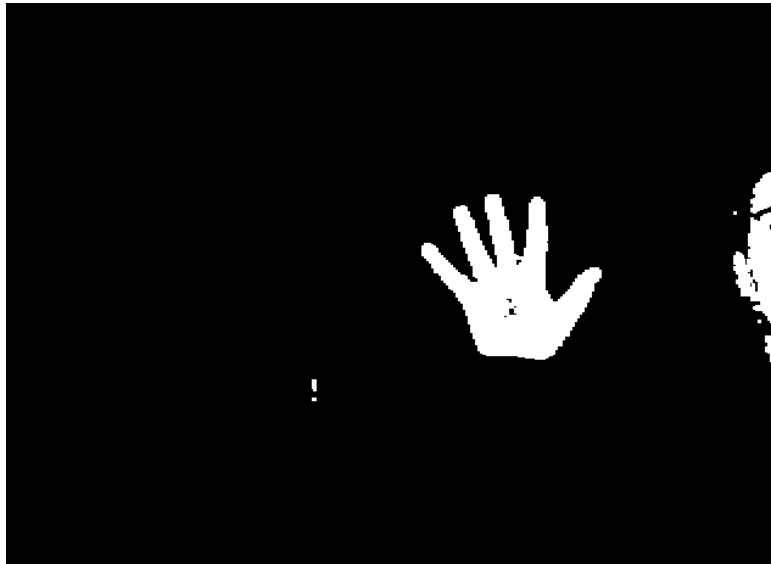
Slika 21 - Frejm nakon segmentacije kože

## **Korak 2.2**

S obzirom da se koristi segmentacija na nivou piksela, kao i činjenice da je prisutan šum pogotovu u videu koji se dobija u realnom vremenu, neophodno je na dobijenu binarnu sliku primeniti morfološke filtere i to operacije erozije a zatim i zatvaranja (objašnjeno u 3.1.2) kako bi se pre svega uklonio šum na nivou piksela i zatvorile rupe i uredile granice segmentirane ruke.

Veličina strukturnog elementa zavisi od rezolucije slike koja se dobija za ulaz, a u ovom algoritmu se za ovu operaciju preporučuje korišćenje jediničnog 3x3 strukturnog elementa.

Rezultat nakon primene navedenih morfoloških filtera je prikazan na slici 22.



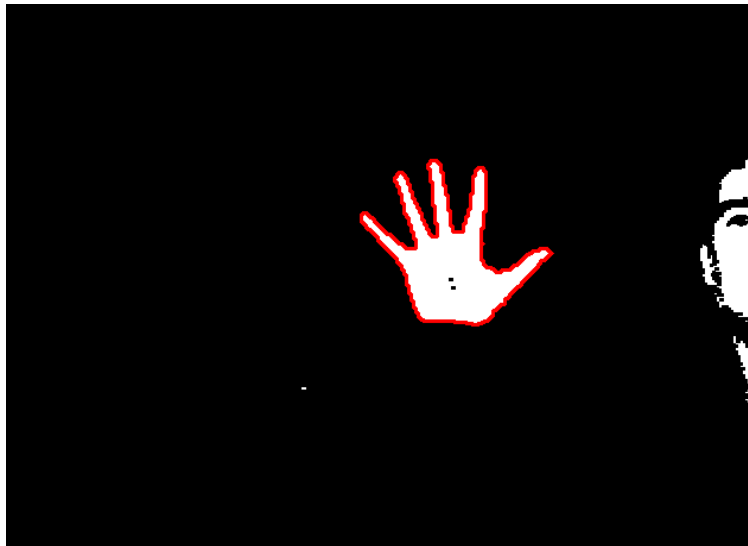
Slika 22 - Frejm nakon primene morfoloških filtera

## **Korak 3.1**

Kako bi se izvršilo ekstrahovanje kontura iz binarne slike, neophodno je izvršiti topološku strukturnu analizu. U ovu svrhu je iskorišćen poznati algoritam Suzukija i Abea [19].

Algoritam Suzukija i Abea za izlaz daje prepoznate konture na ulaznoj binarnoj slici. Šum i nepreciznost algoritma segmentacije kože, za posledicu mogu da imaju prepoznato više kontura. Za prevazilaženje ovog problema, predloženi algoritam primenjuje osnovni

geometrijski deskriptor regiona – površine (poglavlje 3.2.2) na svim vraćenim konturama i za konturu ruke uzima konturu najveće površine, koja je označena crvenim na slici 23.



Slika 23 - Frejm sa obeleženom konturom

### **Korak 3.2**

Dobijena kontura sadrži isuviše tačaka čiji broj nije pogodan za dalju obradu pa je neophodno njegovo smanjivanje, uz zadržavanje iste moći opisa dobijene konture ruke. Takođe, na ovaj način se dodatno uklanja i mogući šum pri obradi. U tu svrhu se koriste dobro poznati Daglas-Pekeroi algoritam [20]. U pitanju je rekurzivni algoritam koji vrši aproksimaciju uklanjanjem tačke ukoliko je ona na manjoj udaljenosti od zadate vrednosti od duži napravljenoj bez nje.

Daljina  $\epsilon$  zavisi od konture koja se aproksimira. U slučaju koji se obrađuje i konturama ruke, vrednost je vezana za geometrijski deskriptor – obim (poglavlje 3.2.2) i nađena je pogodna vrednost od

$$\epsilon = P(C) * 0.003$$

gde je C kontura.

Slika 24 jasno pokazuje razliku aproksimirane konture u odnosu na onu prikazanu na prethodnoj slici.





Slika 24 - Frejm nakon aproksimacije konture

#### **Korak 4.1**

Za opisivanje konture ruke, uzimajući u obzir prirodu pokreta, izabrani su Huovi momenti (poglavlje 3.2.2) zbog svoje invarijantnosti na skaliranje, tranziciju i rotaciju.

#### **Korak 4.2**

Prethodno su izračunati Huovi momenti postura čije je prepoznavanje omogućeno a što je detaljnije opisano u nastavku rada. Svaki Huov momenat tih modela postura se upoređuje sa momentom ulazne posture i za rezultat se uzima onaj model posture najbliži ulaznoj.

Zbog prirode vrednosti Huovih momenata kao i rezultata testiranja, za meru rastojanja je uzeto:

$$D(X, Y) = \sum_{i=1}^7 \left| \frac{1}{m_i^X} - \frac{1}{m_i^Y} \right|$$

gde su  $m_i^X$  i  $m_i^Y$

$$m_i^X = \text{sign}(h_i^X) \log|h_i^X|$$

$$m_i^Y = \text{sign}(h_i^Y) \log|h_i^Y|$$

a  $h_i^X$  i  $h_i^Y$  su Huovi momenti ulazne posture X i modela posture Y.

## Korak 5.1

Algoritam je kreiran za rad u realnom vremenu u kome obrađuje svaki frejm pojedinačno. Usled šuma i ostalih faktora koji utiču na rad u realnom vremenu, potrebno je uraditi stabilizaciju konačnog rezultata klasifikacije ulazne posture, a algoritam to rešava pravljenjem bafera rezultata i njegovom obradom pre vraćanja konačnog rezultata klasifikacije.

Bafer je napravljen kao FIFO struktura sa 5 mesta, u kojoj se konstantno smeštaju klasifikacije posture sa obrađenog frejma. Konačni rezultat klasifikacije posture je postura čiji je broj u buffer-u veći ili jednak 3 ili ništa u suprotnom.

*Primer:*

$$\text{Klasifikacija}_{Final}(P1, P1, P1, P3, P6) = P1$$

$$\text{Klasifikacija}_{Final}(P1, P1, P3, P6, P2) = \text{bez\_rezultata}$$

$$\text{Klasifikacija}_{Final}(P1, P3, P6, P2, P2) = \text{bez\_rezultata}$$

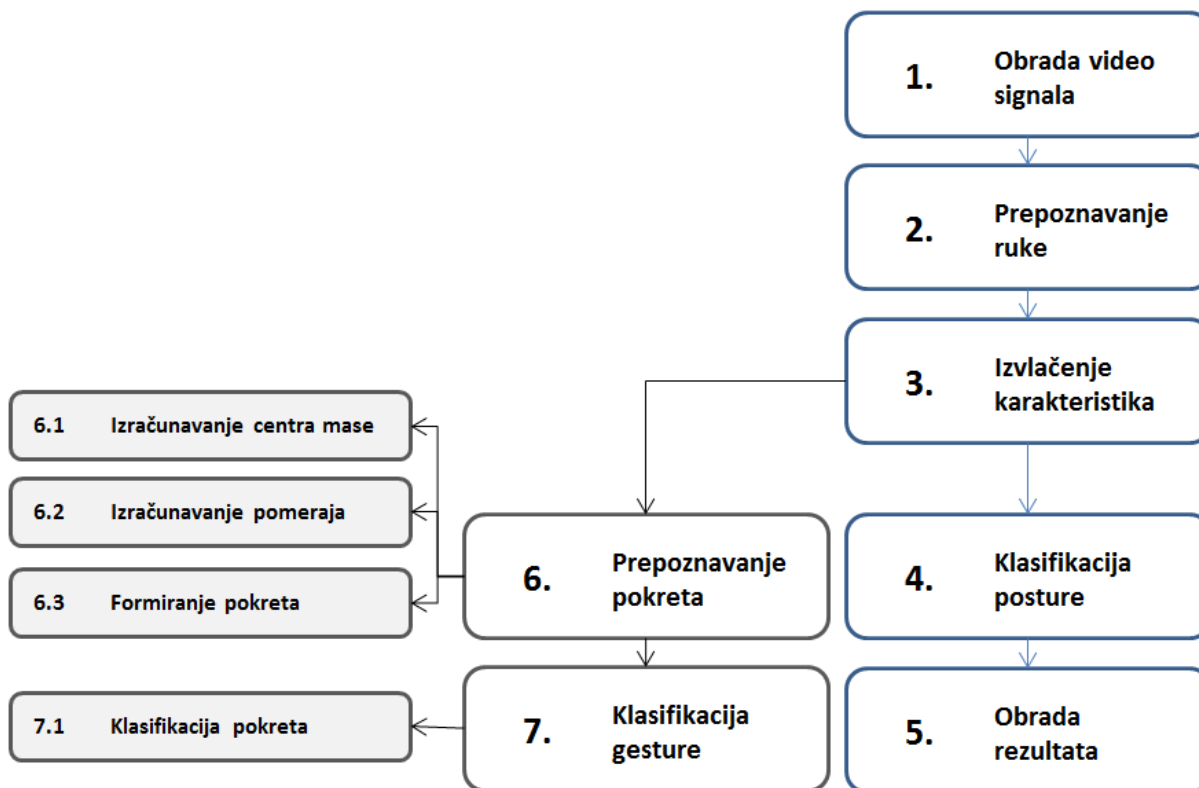
$$\text{Klasifikacija}_{Final}(P3, P6, P2, P2, P2) = P2$$

$$\text{Klasifikacija}_{Final}(P6, P2, P2, P2, P2) = P2$$

Zbog primene bafera dolazi do kašnjenja od najmanje 2 frejma pri konačnoj klasifikaciji, ali je to kašnjenje prihvatljivo malo.

## 4.2 Unapređenje algoritma za prepoznavanje gestura

U svrhu prepoznavanja gestura, algoritam je unapređen dodavanjem koraka 6 i 7 koji se paralelno izvršavaju sa koracima 4 i 5. Dodati koraci i njihovi podkoraci su prikazani na Slika 25 a zatim i detaljnije opisani.



Slika 25 - Unapređenje algoritma za prepoznavanje gestura

### **Korak 6.1**

S obzirom na korake 1, 2 i 3 algoritma nakon kojih je poznata kontura ruke, odlučeno je da se pomeraji ruke prate preko njenog centra mase  $(\bar{x}, \bar{y})$  koji je prethodno objašnjen u poglavlju 3.2.2.

Koordinate centra mase se izračunavaju formulom opisanom u poglavlju 3.2.2, a izračunavanja se vrše za svaki frejm pojedinačno.

### **Korak 6.2**

Pomeraj definišemo kao razliku pozicija centara mase ruke između dva uzastopna frejma.

Kategorizacija pomeraja se izračunava u odnosu na ugao vektora (sa početkom u prethodnom centru mase i krajem u narednom centru mase) koji se izračunava na osnovu izraza

$$Ugao\_pomeraja = \frac{180}{\pi} * \arctan\left(\frac{\bar{y}_{nared} - \bar{y}_{preth}}{\bar{x}_{nared} - \bar{x}_{preth}}\right)$$

Ugao pomeraja se kategoriše oznakom jednog od 8 smerova kome je najbliži. Smerovi i njihove oznake su sledeći

Smer koji odgovara uglu	Oznaka
0°	1
45°	2
90°	3
135°	4
180°	5
225°	6
270°	7
315°	8

### **Korak 6.3**

Pokret definišemo kao niz pomeraja, a u kontekstu nastavka algoritma, kao vektor oznaka pomeraja. Pokret započinje kada je  $D_{Euclid}(C_{nared}, C_{preth}) > \varepsilon$ , a završava se kada je  $D_{Euclid}(C_{nared}, C_{preth}) < \varepsilon$ , gde su  $C_{nared}$  i  $C_{preth}$  centri mase ruke na dva uzastopna frejma.

U toku trajanja pokreta, formira se sekvenca oznaka pomeraja koja karakteriše pokret i čija se klasifikacija vrši u sledećem koraku algoritma.

S obzirom na određeni stepen nestabilnosti konture ruke u realnom vremenu, samim tim u svakom frejmu, centri mase ruke se menjaju i u slučaju ruke koja se ne pomera. Testiranjem je utvrđeno da je odgovarajuća vrednost za ovu primenu  $\varepsilon = 10$ .

## **Korak 7.1**

Kako je pokret predstavljen nizom oznaka, gesturu možemo posmatrati kao vremensku seriju diskretnih stanja. S obzirom na slične i dokazane primene i rezultate u prepoznavanju i klasifikaciji govora, prepoznavanju rukopisa i mnogobrojnim primenama u bioinformatici, za klasifikaciju gestura izabrani su skriveni Markovljevi modeli (*eng. Hidden Markov Models - HMM*), a za učenje njihovih parametara, Baum-Velšov algoritam.

HMM su stohastičke metode za modelovanje vremenskih serija. Između ostalih primena, koriste se za računanje verovatnoće da je data serija potekla iz konkretnog HMM-a.

Baum-Velšov algoritam je algoritam nenadgledanog učenja koji se koristi za učenje pojedinačnog HMM-a na osnovu skupa opaženih serija. Koristi varijantu EM (*eng. Expectation Maximization*) algoritma kojom traži skup parametara modela, koji daju visoku verovatnoću za skup opaženih i njima sličnih serija.

Za detaljnije objašnjenje HMM-a i Baum-Velšovog algoritma početni izvori mogu biti [21] i [22].

Predloženi algoritam za svaku gesturu za koju se vrši prepoznavanje, pravi zaseban skriveni Markovljev model. Svaki se trenira skupom sekvenci oznaka pomeraja koje karakterišu pokrete koji odgovaraju gesturi. U ovom koraku algoritma, sekvenca kreirana u prethodnom koraku se propušta kroz svaki skriveni Markovljev model i klasifikuje kao gestura odgovarajućeg HMM-a koji vrati najveću verovatnoću da je potekla iz njega.

## 5 Realizacija algoritma

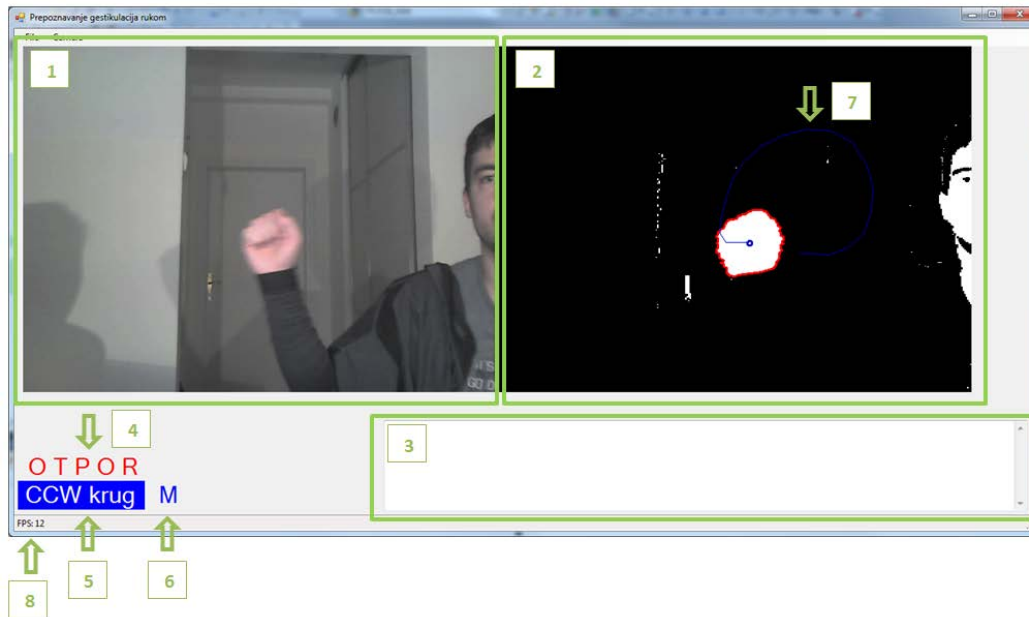
Celokupan opisani algoritam je implementiran u okviru Windows .NET aplikacije koja je korišćena za kompletno modeliranje, testiranje i prikupljanje rezultata.

Za razvojno okruženje je korišćen Microsoft Visual Studio a za kodiranje programski jezik je C#. Kao dodatak ručnom kodiranju i standardnim .NET bibliotekama, zbog efikasnosti su korišćene sledeće specifične biblioteke iz domena rada:

- **OpenCV/Emgu CV** – OpenCV (Open Source Computer Vision Library) je jedna od najpopularnijih biblioteka za računarsku viziju sa preko 14 miliona preuzimanja. API dokumentacija se nalazi na [23], a u radu su konsultovani primeri upotrebe iz [24] i [25]. Ova biblioteka ima zvanične interfejsa za C++, Python i Java-u, a Emgu CV je nezvanični tzv. wrapper, preko koga je omogućeno korišćenje ove biblioteke i za C#. API dokumentacija se može pronaći na [26].
- **Accord.NET** – popularna .NET biblioteka iz domena mašinskog učenja i naučnog izračunavanja. API dokumentacija se nalazi na [27].
- **AForge.NET** - popularna .NET biblioteka iz domena računarske vizije i veštačke inteligencije. API dokumentacija se nalazi na [28].

### 5.1 Aplikacija

Aplikacija se pokreće izvršavanjem .exe fajla nakon čega se dobija osnovni prozor prikazan na slici 26. Automatski se aktivira kamera i započinje obrada u realnom vremenu. U zavisnosti od potrebe, svojstva kamere se mogu dodatno podesiti kroz opciju „Camera“ koja se nalazi u meniju.



Slika 26 – Osnovni prozor aplikacije i njegove komponente

Komponente osnovnog prozora navedene su u narednoj tabeli.

Oznaka	Opis komponente
1	Original frejma koji se obrađuje
2	Kopija originalnog frejma na kojoj se vrši obrada i označavanje
3	Deo koji se koristi za ispis različitih potrebnih informacija (međurezultati, ručna obrada, treniranje modela i sl.). Podrazumevano ispis nije uključen i od zavisnosti od potrebe uključuje se u samom kodu aplikacije.
4	Naziv prepoznate posture
5	Naziv prepoznate gesture
6	Oznaka se pojavljuje kada je pomeraj u toku
7	Putanja centra mase u toku pomeraja
8	Broj obrađenih frame-ova u sekundi

## 5.2 Dobijeni rezultati

U svrhu testiranja algoritma, napravljen je izbor skupa postura i gestura za prepoznavanje. Modeli postura i gestura su dobijeni korišćenjem istog algoritma bez koraka klasifikacije. Na ovaj način je omogućeno da se prepoznavanje ulaznih gestikulacija vrši nad njihovim modelima dobijenim pod istim uslovima. Testiranje je izvršeno od strane autora na dva različita računara.

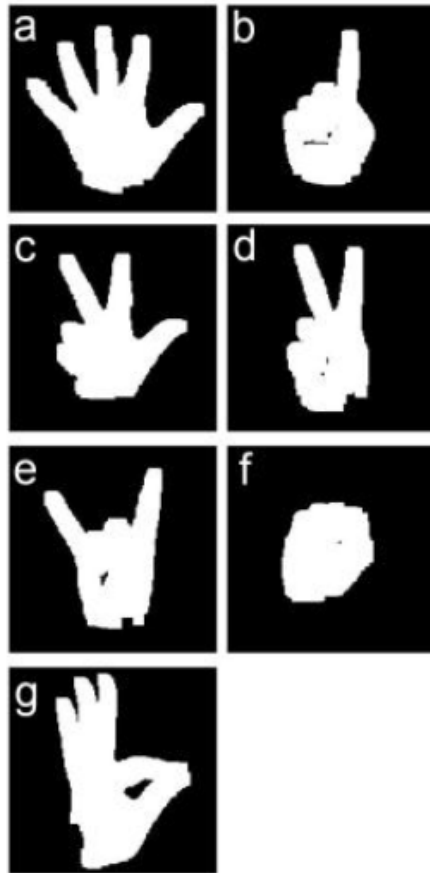
U naredna dva poglavlja izneti su detalji i rezultati testiranja osnovnog algoritma za prepoznavanje postura i njegovog unapređenja za prepoznavanje gestura.

### 5.2.1 Rezultati algoritma za prepoznavanje postura

Za prepoznavanje je izabrano sledećih 7 postura prikazanih na slici 27:

a) pet, b) prst, c) tri, d) zec, e) metal, f) otpor, g) oko.





Slika 27 - 7 modela postura nakon koraka 2.2

Nakon koraka 4.2 za ove modele postura izračunati su Huovi momenti koji će biti upotrebljavani za poređenje i klasifikaciju ulaznih postura.

Model posture	Huovi invarijantni momenti						
	h1	h2	h3	h4	h5	h6	h7
<b>Pet</b>	1.983E-1	1.776E-3	1.634E-5	5.244E-5	-1.356E-9	5.566E-7	-7.181E-10
<b>Tri</b>	2.13E-1	2.905E-3	1.408E-3	4.195E-4	1.41E-7	2.186E-5	-2.898E-7
<b>Zec</b>	2.426E-1	2.328E-2	5.417E-4	7.476E-4	4.751E-7	1.13E-4	2.349E-8
<b>Prst</b>	2.141E-1	1.252E-2	3.182E-3	1.156E-3	2.217E-6	1.293E-4	5.435E-8
<b>Otpor</b>	1.659E-1	1.545E-3	7.364E-5	7.226E-7	2.155E-12	7.468E-9	4.811E-12
<b>Oko</b>	2.284E-1	1.206E-2	2.978E-3	2.661E-4	1.853E-7	2.923E-5	-1.477E-7
<b>Metal</b>	2.351E-1	4.49E-3	3.772E-3	7.13E-4	-1.102E-6	4.571E-5	3.903E-7

Svaka postura je testirana 30 puta na različitim udaljenostima i pod različitim uglovima i dobijeni su sledeći rezultati.

Postura	Tačnih klasifikacija	Netačnih klasifikacija	Procenat uspešnih klasifikacija
<b>Pet</b>	23	7	77%
<b>Tri</b>	22	8	73%
<b>Zec</b>	27	3	90%
<b>Prst</b>	24	6	80%
<b>Otpor</b>	29	1	97%
<b>Oko</b>	17	13	57%
<b>Metal</b>	21	9	70%

Ukupan procenat uspešnog prepoznavanja se kreće oko **78%**.



Sekvence	Gestura
<p>{5,5,5,5,5,5,5,4,6,5,5,7,4,5,5,5,5,5,4,5,5,5,5,5,5}</p> <p>{5,5,5,5,5,5,5,5,5,6,5,6,3,6,4,1,5,6,4,4,6,4,5,5,5,5}</p> <p>{5,5,5,5,5,5,5,4,5,4,5,8,5,5,8,5}</p> <p>{5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,4,5}</p> <p>{5,6,5,5,5,5,5,5,5,5,5,4,5,5,5,5,1,2,5,4,5,5,5}</p> <p>{5,5,5,5,5,5,5,5,5,5,5,1,6,5,3,6,5,5,4}</p> <p>{5,5,5,5,5,5,5,5,5,5,5,5,5,5,4,6,6,5,5,5}</p> <p>{6,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,4,4,5,4,5}</p> <p>{5,5,5,5,5,5,5,5,5,5,5,5,5,4,8,4,5,8,5,2}</p> <p>{6,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,8,5,5,5,5}</p> <p>{5,5,5,5,5,5,5,5,5,5,5,5,1}</p> <p>{5,5,5,5,5,5,5,5,5,5,6,5,4,6}</p> <p>{5,6,5,5,5,5,5,5,5,5,5,5,5,5,5,7,5,5,6}</p> <p>{5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5}</p> <p>{5,5,5,5,5,5,5,5,5,5,5,4,8}</p>	<b>LEVO</b>

Sekvence	Gestura
<p>{7,7,5,7,7,7,7,7,7,7,7,7,7,7,8}</p> <p>{7,7,7,7,7,7,7,7,7,8,7,7,7,7,8,7,7,7,7,7,7,7}</p> <p>{7,6,7,7,7,7,7,7,8,7,7,7,7,7}</p> <p>{1,7,1,7,7,8,8,8,8,7,7,8,7,7,7,7,7,7,7,7,7,7}</p> <p>{8,7,7,7,7,8,7,7,7,7,7,7,7,7,7,7}</p> <p>{7,7,8,7,7,7,7,7,7,8,7,8,7,7,7}</p> <p>{7,7,8,7,7,7,7,7,7,7,7,8,8,7}</p> <p>{8,7,7,7,7,8,7,8,7,7,8,7,7,8,7,8,8,8}</p> <p>{7,1,7,8,8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7}</p> <p>{8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7}</p> <p>{7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7}</p> <p>{7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,8}</p> <p>{8,7,7,7,7,6,7,7,7,7,7}</p> <p>{7,6,6,6,6,6,6,6,6,6,6}</p> <p>{7,8,7,7,7,7,7,7,7,7,8,7,7,7,7,7,7}</p>	<b>GORE</b>

Sekvence	Gestura
<p>{4,3,3,4,3,4,4,3,3,3,3,3,3,3,3,3,3,4,3,3,3,4}</p> <p>{4,4,4,3,3,4,3,3,4,3,4,4,4,3,4}</p> <p>{4,3,3,4,3,3,4,3,3,4,3,4,3,3,4,3,4,3,3}</p> <p>{4,3,3,3,3,3,3,3,3,4,3,3,3,3,4,3,4,3,3}</p> <p>{3,4,3,4,3,3,3,3,3,3,3,3,3,4,3,4,3,3,4,4,3,3,4,3,3}</p> <p>{3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3}</p> <p>{3,4,3,3,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3}</p> <p>{4,3,3,3,4,4,4,4,3,4,3,4,2,3,4,3,3,3}</p> <p>{3,3,3,3,3,3,3,3,3,3,3,4,3,3,3,3,3,3,3,3,3,3,3,3,3}</p> <p>{3,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3}</p> <p>{3,4,3,3,3,3,3,3,3,3,3,3,3,4}</p> <p>{3,4,4,3,4,4,4,4,3,3,4,3,4}</p> <p>{4,4,3,4,3,4,4,3,4,3,3,3,3,3,4,3,3}</p> <p>{4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3}</p> <p>{4,3,4,4,4,4,3,4,3,4,4,3,3,4,4,4}</p> <p>{4,4,4,3,4,3,4,3,3,3,3,3,3,3,3,4}</p>	<b>DOLE</b>

Sekvence	Gestura
<p>{1,8,8,1,8,7,8,8,7,7,6,6,6,6,5,5,5,5,5,4,4,4,3,3,3,3,2,2,2,2,1,1,1}</p> <p>{8,2,8,1,1,1,8,1,8,8,8,8,8,7,7,8,6,6,6,6,5,5,5,5,5,5,4,4,4,4,4,3,3,3,3,2,2}</p> <p>{2,1,1,1,1,8,8,6,7,8,7,7,6,6,6,6,6,5,6,5,5,5,4,4,4,4,4,4,4,3,4,3,3,3,2,2,2,2,2}</p> <p>{2,8,2,1,1,1,1,8,8,8,8,8,7,7,7,7,6,6,6,6,6,5,5,5,5,5,5,4,4,4,4,4,4,4,3,3,3,2,2,2,2,2,2}</p> <p>{8,2,1,1,1,1,8,8,8,8,8,7,7,7,7,6,6,6,6,6,5,5,5,5,5,5,4,4,4,4,4,4,4,3,3,3,3,3,2,2,2,2,2,2,1,1}</p> <p>{1,1,1,1,1,8,8,8,8,7,7,7,6,6,6,6,6,5,5,4,5,4,4,4,4,4,4,4,3,3,3,3,2,2,2,2,1,1}</p> <p>{1,1,1,1,8,8,8,7,7,7,6,6,6,6,5,5,5,5,5,4,4,4,4,4,4,4,4,3,3,3,2,2,2,1,2,1,1,1}</p> <p>{8,2,1,8,1,1,1,8,8,8,8,7,7,7,6,7,6,6,6,6,5,5,5,5,4,4,4,4,4,4,4,4,3,3,2,2,2,1,2,1,1,1}</p> <p>{1,1,1,1,8,8,8,8,7,7,7,6,6,6,6,5,5,5,5,5,4,4,4,4,4,4,4,3,3,3,3,2,2,2,2,1,1,1,1,1}</p> <p>{2,1,8,1,8,8,1,8,8,7,7,7,7,6,6,6,6,5,5,5,5,4,5,4,4,4,4,4,4,3,3,2,2,2,2,2,1,1,1,1,1}</p> <p>{8,2,1,8,1,1,1,8,8,7,7,7,7,6,6,6,6,6,5,5,5,5,4,4,4,4,4,4,3,3,3,2,2,2,2,2,2,2,2,2,2,2,2,1}</p> <p>{1,8,1,1,1,8,8,8,7,8,7,7,7,6,6,6,5,5,5,5,4,4,4,4,4,4,4,4,3,3,3,2,2,2,2,2,1,1,1}</p> <p>{1,1,2,8,1,8,8,8,8,7,7,7,6,6,6,6,6,5,5,5,5,4,4,4,4,4,4,4,3,3,3,3,2,2,2,2}</p> <p>{2,8,1,1,1,1,8,8,7,7,7,7,6,6,6,6,6,5,5,5,4,5,4,4,4,4,4,4,3,3,3,3,2,2,2,2,2,2,2,2,1}</p> <p>{8,1,1,1,8,1,1,8,7,8,7,7,7,6,6,6,6,5,5,5,5,4,5,4,4,4,4,4,4,3,3,3,2,2,2,2,2,2,2,2,1,1,1,1}</p>	<b>CCW krug</b>

Sekvence	Gestura
{5,4,5,5,5,5,5,6,6,6,7,7,7,8,8,8,8,1,1,1,1,1,2,2,2,2,2,3,3,3,4,4,5,5,4,4,5,5}	<b>CW krug</b>
{4,4,5,5,5,5,5,5,6,6,6,7,7,7,7,8,8,8,8,8,8,1,1,1,1,2,2,2,2,2,2,3,2,3,3,4,4,4,5,4,5,5}	
{5,4,5,5,5,5,5,5,5,6,7,7,7,7,8,8,8,8,8,8,1,8,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,4,4,5,5,5}	
{5,4,5,5,5,5,5,5,5,6,6,6,7,7,7,8,8,8,8,8,8,1,1,1,1,1,2,2,2,2,2,3,3,3,4,4,4,4,4,5,5,5,5}	
{4,5,4,5,5,5,5,5,6,6,6,6,7,8,7,8,8,8,8,8,8,1,1,1,1,1,1,2,2,2,2,3,3,3,4,4,5,4,4,5,5,5}	
{3,4,5,5,5,5,5,5,5,6,6,6,7,7,7,8,8,8,8,8,8,8,1,1,1,2,2,2,2,2,2,3,3,3,4,4,4,4,5,5}	
{5,2,4,5,5,5,5,5,6,6,6,7,7,7,7,8,8,8,8,1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,4,4,5,5,5}	
{4,5,5,5,5,5,5,5,5,6,6,6,7,7,7,8,8,8,8,8,8,8,1,1,1,1,2,2,2,2,2,2,3,3,3,3,4,4,4,5,4,5}	
{3,4,6,4,5,5,5,5,6,6,6,7,7,7,7,8,7,8,8,8,8,1,1,1,1,1,2,2,2,2,2,2,3,3,3,4,4,4,5,5,5,5,5}	
{4,5,4,6,5,5,5,5,6,6,6,6,7,7,7,8,8,8,8,8,1,1,1,1,1,2,2,2,2,2,2,3,3,4,4,4,4,4,4,5,4,5}	
{5,5,5,5,5,5,6,6,6,7,7,7,7,8,8,8,8,8,1,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,4,5,5,5,4,5}	
{4,5,5,5,5,5,5,6,6,6,7,7,7,8,8,8,8,1,1,1,1,1,2,2,2,2,2,3,3,4,4,4,4,4,5,5,4,5,5,4}	
{1,5,5,5,5,5,5,5,6,6,6,7,7,7,8,8,8,8,1,1,1,1,1,2,2,2,2,3,3,3,3,3,4,4,4,4,5,4,5}	
{5,1,5,5,5,5,5,5,6,6,6,6,7,7,8,8,7,8,8,8,1,1,1,1,1,1,2,2,2,2,3,3,3,4,4,4,4,4,4,5,5,5,5}	
{5,5,5,4,5,5,5,5,6,6,6,7,7,7,8,8,8,8,1,8,1,1,1,2,2,2,2,2,3,3,3,4,4,4,4,4,5,5,5,5,5}	

Svaka gestura je testirana 30 puta i dobijeni su sledeći rezultati.

Gestura	Tačnih klasifikacija	Netačnih klasifikacija	Procenat uspešnih klasifikacija
<b>DESNO</b>	29	1	97%
<b>LEVO</b>	28	2	93%
<b>GORE</b>	30	0	100%
<b>DOLE</b>	29	1	97%
<b>CCW krug</b>	25	5	83%
<b>CW krug</b>	26	4	87%

Ukupan procenat uspešnog prepoznavanja se kreće oko **93%**.

## 6 Zaključak

Prepoznavanje gestikulacija ruku je složen problem, posebno u realnom vremenu. Predloženi algoritam je testiran u kontrolisanim uslovima osvetljenja i pozadine i dobijeni su zadovoljavajući rezultati od 78% prepoznatih postura i 93% prepoznatih gestura. Specijalno, procenat prepoznatih gestura sugerise da algoritam može biti uspešan i za prepoznavanje kompleksnijih gestura od testiranih.

Uticaj segmentacije kože se ispostavlja kao presudan na rezultate, kao i na mogućnost upotrebe algoritma u nekontrolisanim okruženjima. Početna loša i neprecizna segmentacija predstavlja ključni problem dalje obrade, kod postura u izvlačenju kontura i njihovoj klasifikaciji, a kod gestura u nestabilnosti centra mase koji utiče na prikazivanje pokreta iako se u realnosti on nije dogodio.

Dalja unapređenja predloženog algoritma mogu ići u više pravaca:

- Segmentaciji kože se može pristupiti na nivou regiona umesto na nivou piksela
- Kalibracijom korišćene kamere. Današnje kamere vrše svoje interne ne nužno tačne kalibracije boja i automatska prilagođavanja osvetljenja sa ciljem vizualne lepote a na štetu tačnosti slike i njenih boja
- Otklanjanje dva preduslova rada algoritma na načine opisane u poglavlju 4.
- Dinamičkim određivanjem parametara predloženim u ovom algoritmu u zavisnosti od okruženja i slučaja upotrebe
- Algoritam obrađuje gestikulacije jedne ruke, ali se na istovetan način može uvesti i prepoznavanje gestikulacija sa obe ruke
- Kreiranjem kompleksnijih gestikulacija kombinacijom prepoznatih postura i gestura

Na kraju, predloženi algoritam daje dobre osnove za prepoznavanje gestikulacija rukom u realnom vremenu i može predstavljati polaznu tačku za dalja istraživanja i implementacije sličnih pristupa u ovoj oblasti.

## Lista slika

Slika 1 - CyberGlove II .....	9
Slika 2 - CyberGlove III .....	10
Slika 3 – Orijentacione mape .....	11
Slika 4 - Projekcija posle PCA .....	13
Slika 5 - Standardni koraci u prepoznavanju gestikulacija.....	14
Slika 6 - Grafovi gustine boja kože rasa ljudi u različitim prostorima boja.....	16
Slika 7 - RGB prostor boja .....	17
Slika 8 - CbCr ravan sa konstantim intenzitetom svetlosti $Y=0.5$ .....	18
Slika 9 - Erozija nad reprezentacijom binarne slike sa 3x3 jediničnim strukturnim elementom .	20
Slika 10 - Dilatacija nad reprezentacijom binarne slike sa 3x3 jediničnim strukturnim elementom .....	21
Slika 11 - Otvaranje nad reprezentacijom binarne slike sa 3x3 jediničnim strukturnim elementom.....	22
Slika 12 - Otvaranje nad binarnom slikom sa strukturnim elementom oblika diska.....	22
Slika 13 - Zatvaranje nad reprezentacijom binarne slike sa 3x3 jediničnim strukturnim elementom.....	23
Slika 14 - 4-povezanost i 8-povezanost sa smerovima i oznakama .....	25
Slika 15 - Region, kontura 4-povezana i kontura 8-povezana .....	25
Slika 16 - Lanci kodova (levo: 4-povezanost, desno: 8-povezanost) .....	26
Slika 17 - Ilustracija Euklidskog (zelenim) i Menheten (plavim) rastojanja .....	31
Slika 18 - Ilustracija Čebiševljevo rastojanja .....	31
Slika 19 - SVM, prava sa maksimalnom marginom između dva skupa u 2D .....	34
Slika 20 - Algoritam za prepoznavanje postura .....	37
Slika 21 - Frejm nakon segmentacije kože .....	38
Slika 22 - Frejm nakon primene morfoloških filtera .....	39
Slika 23 - Frejm sa obeleženom konturom .....	40
Slika 24 - Frejm nakon aproksimacije konture.....	41
Slika 25 - Unapređenje algoritma za prepoznavanje gestura.....	43



Slika 26 – Osnovni prozor aplikacije i njegove komponente .....	47
Slika 27 - 7 modela postura nakon koraka 2.2.....	49
Slika 28 - 6 pokreta izabranih gestura rukom za prepoznavanje.....	51

## Lista referenci

- [1] P. Premaratne, *Human Computer Interaction Using Hand Gestures*, Springer, 2014.
- [2] B. Markovich / M. Tuba, „Vision-Based Algorithm for Real-Time Hand Posture Recognition,“ *WSEAS*, Prague, 2012.
- [3] L. Dipietro, A. M. Sabatini / P. Dario, „A Survey of Glove-Based Systems,“ *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, t. 38, br. 4, pp. 461-482, 2008.
- [4] J. Davis / M. Shah, „Recognizing Hand Gestures,“ *ECCV*, Stockholm, 1994.
- [5] Y. Iwai, K. Watanabe, Y. Yagi / M. Yachida, „Gesture recognition by using colored gloves,“ *IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems*, 1996.
- [6] M. R. William T. Freeman, „Orientation Histograms for Hand Gesture Recognition,“ *IEEE Intl. Workshop on Automatic Face and Gesture Recognition*, Zurich, 1995.
- [7] N. Shimada, Y. Shirai, Y. Kuno / J. Miura, „Hand Gesture Estimation and Model Refinement using Monocular Camera - Ambiguity Limitations by Inequality Constraints,“ *Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998.
- [8] G. Gastaldi, A. Pareschi, S. P. Sabatini, F. Solari / G. M. Bisio, „A MAN-MACHINE COMMUNICATION SYSTEM BASED ON THE VISUAL ANALYSIS OF DYNAMIC GESTURES,“ *ICIP International Conference on Image Processing*, 2005.
- [9] A. Elgammal, C. Muang / D. Hu, „Skin Detection - a short tutorial,“ *Encyclopedia of Biometrics*, Springer, 2009, pp. 1218-1224.
- [10] R. C. Gonzalez / R. E. Woods, *Digital Image Processing*, Prentice Hall, 2002.

- [11] M. S. Nixon / A. S. Aguado, *Feature Extraction and Image Processing*, Newnes, 2002.
- [12] P.-N. Tan, M. Steinbach / V. Kumar, *Introduction to Data Mining*, Pearson Education, 2006.
- [13] X. Wu / V. Kumar, *The Top Ten Algorithms in Data Mining*, CRC Press, 2009.
- [14] L. Yun / Z. Peng, „An Automatic Hand Gesture Recognition System Based on Viola-Jones Method and SVMs,“ u *Computer Science and Engineering, International Workshop on IWCSE*, Qingdao, 2009.
- [15] M. Sinith, S. G. Kamal, N. B., N. S. / K. S. a. J. P.S., „Sign Gesture Recongnition Using Support Vector Machine,“ *International Conference on Advances in Computing and Communications(ICACC)*, pp. 122-125, 2012.
- [16] M. Han, J. Chen, L. Li / Y. Chang, „Visual hand gesture recognition with convolution neural network,“ *2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 287-291, 2016.
- [17] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat / J. Jatakia, „Human Skin Detection Using RGB, HSV and YCbCr Color Models,“ *Advances in Intelligent Systems Research*, t. 137, pp. 324-332, 2017.
- [18] K. B. Shaik, P. Ganesan, V.Kalist, B.S.Sathish / J. M. Jenitha, „Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space,“ *Procedia Compter Science*, t. 57, pp. 41-48, 2015.
- [19] S. Suzuki / K. Abe, „Topological structural analysis of digitized binary images by border following,“ *Computer Vision, Graphics, and Image Processing*, t. 30, pp. 32-46, 1985.
- [20] D. Douglas / T. Peucker, „Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,“ *The Canadian Cartographer*, pp. 112-256, 1973.

- [21] L. Rabiner, „A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,“ *Proceedings of the IEEE*, pp. 257-286, 1989.
- [22] E. Frazzoli, „Intro to Hidden Markov Models: the Baum-Welch Algorithm,“ Massachusetts Institute of Technology, [Na mreži]. Available: [http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-410-principles-of-autonomy-and-decision-making-fall-2010/lecture-notes/MIT16\\_410F10\\_lec21.pdf](http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-410-principles-of-autonomy-and-decision-making-fall-2010/lecture-notes/MIT16_410F10_lec21.pdf).
- [23] „OpenCV documentation,“ [Na mreži]. Available: <https://docs.opencv.org/>.
- [24] G. Bradski / A. Kaehler, *Learning OpenCV*, O'Reilly, 2008.
- [25] D. M. E. V. G. Prateek Joshi, *OpenCV By Example*, PACKT publishing, 2016.
- [26] „EmguCV documentation,“ [Na mreži]. Available: <http://www.emgu.com/wiki/>.
- [27] „Accord.NET documentation,“ [Na mreži]. Available: [http://accord-framework.net/docs/html/R\\_Project\\_Accord\\_NET.htm](http://accord-framework.net/docs/html/R_Project_Accord_NET.htm).
- [28] „AForge.NET documentation,“ [Na mreži]. Available: <http://www.aforge.net.com/framework/docs/>.