

ŠIAULIŲ UNIVERSITETAS  
TECHNOLOGIJOS FAKULTETAS  
ELEKTRONIKOS INŽINERIJOS KATEDRA

Mindaugas Bertašius

GESTŲ ATPAŽINIMAS  
Magistro darbas

**Vadovas**

doc. dr. G. Daunys

ŠIAULIAI, 2005

ŠIAULIŲ UNIVERSITETAS  
TECHNOLOGIJOS FAKULTETAS  
ELEKTRONIKOS INŽINERIJOS KATEDRA

**TVIRTINU**  
**KATEDROS VEDĖJAS**

doc. dr. G. Daunys

2005 0..

**GESTŲ ATPAŽINIMAS**  
Magistro darbas

**Recenzentas**

ŠU Technologijos fakulteto  
(.....) prof. dr. V. Lauruška  
2005 0..

**Vadovas**

(.....)  
2005 0..

doc.dr.G. Daunys

Atliko

(.....)  
2005 0.. 0..

RM3 gr. stud.  
M. Bertašius

ŠIAULIAI, 2005

Bertašius M. Gesture recognition: Master thesis of electronics engineer/research advisor Assoc. Prof. Dr. G. Daunys; Šiauliai University, Technological Faculty, Electronics Engineering Department. – Šiauliai, 2005. – 70p.

## **SUMMARY**

The theme of Master project of Electronics engineer is actual for human-computer interaction (HCI). Hand gestures provides an attractive alternative to cumbersome interface devices for human-computer interaction (HCI). In particular, visual interpretation of hand gestures can help in achieving the ease and naturalness desired for HCI.

Working with computers has become an integral feature of our society. The keyboard and mouse are currently the main interfaces between man and computer. In other areas where 3D information is required, such as computer games, robotics and design, other mechanical devices such as roller-balls, joysticks and data-gloves are used. Humans communicate mainly by vision and sound, therefore, a man-machine interface would be more intuitive if it made greater use of vision and audio recognition. Another advantage is that the user not only can communicate from a distance, but need have no physical contact with the computer. However, unlike audio commands, a visual system would be preferable in noisy environments or in situations where sound would cause a disturbance. The visual system chosen was the recognition of hand gestures. There are many explorations of objects recognition in images and in video stream. Yearly improvements of computer technology lets researchers model and build more and more complex systems of recognition but it still remains a lot of unsolved problems and it is hard to decide witch system is the best. Our aim is to create a program for hand gesture recognition based on pattern recognition algorithms and Eigen vectors.

# TURINYS

<b>IŽANGA</b> .....	<b>3</b>
<b>1. TIRIAMOJO OBJEKTO – RANKOS - SAVYBĖS</b> .....	<b>5</b>
1.1. RANKOS FORMOS MODELIAVIMAS.....	5
1.2. RANKOS KINEMATINĖ STRUKTŪRA.....	6
<b>2. RANKOS PARAMETRŲ PERDAVIMAS Į KOMPIUTERĮ</b> .....	<b>9</b>
2.1. NAUDOJAMOS TECHNINĖS PRIEMONĖS.....	9
2.2. SKAITMENINIS ATVAIZDAS GAUNAMAS KAMERA.....	10
<b>3. INFORMACIJOS APIE RANKOS GESTĄ APDOROJIMAS</b> .....	<b>11</b>
3.1. PRADINIS INFORMACIJOS APDIRBIMAS – „PREPROCESSING“.....	11
3.2. RANKOS GESTUI BŪDINGŲ SAVYBIŲ IŠRINKIMAS.....	12
3.3. KLASIFIKAVIMO METODAI.....	12
<b>4. PRAKTINIS TYRIMAS</b> .....	<b>25</b>
4.1. PRADINŲ SĄLYGŲ NUSTATYMAS.....	25
4.2. TYRIMO EIGA.....	25
<b>IŠVADOS IR SIŪLYMAI</b> .....	<b>40</b>
<b>LITERATŪRA</b> .....	<b>41</b>
<b>PRIEDAI</b> .....	<b>43</b>

Paveikslų sąrašas:

- 1.1 pav. Rankos geometriniai modeliai.
- 1.2 pav. Rankos skeletas.
- 2.1 pav. Spalvų kodavimo sistemos.
- 3.1 pav. Rankos lokalizavimas.
- 3.2 pav. Euklido atstumas.
- 3.3 pav. Markovo tinklas.
- 3.4 pav. Modelių sudarymas.
- 3.5 pav. Modelių pavyzdžiai.
- 4.1 pav. Bendras algoritmas.
- 4.2 pav. Naudojamas programos algoritmas.
- 4.3 pav. Kadras su rankos gestu.
- 4.4 pav. Pagrindo kadras be rankos.
- 4.5 pav. Skirtumas tarp pagrindo ir rankos su gestu atvaizdų.
- 4.6 pav. RGB sluoksniai.
- 4.7 pav. Sudėtos RGB kaukės.
- 4.8 pav. Atvaizdas su užpildytomis mažomis skylutėmis.
- 4.9 pav. Originalus atvaizdas su pritaikyta kauke.
- 4.10 pav. Atvaizdas paverstas į HSV spalvų sistemą.
- 4.11 pav. Rankos kaukės.
- 4.12 pav. Veiksmai su kaukėmis.
- 4.13 pav. Galutinis rezultatas.
- 4.14 pav. Raiščio kaukės.
- 4.15 pav. Veiksmai su raiščio kaukėmis.
- 4.16 pav. Išsirinktas atpažinimui kadras.
- 4.17 pav. Pagal kalibravimo reikšmės išskirta ranka su raiščiu.
- 4.18 pav. Raiščio išskyrimas.
- 4.19 pav. Atvaizdo paruošimas.
- 4.20 pav. Naudojamos gestų reikšmės ir bendra gestų erdvė.
- 4.21 pav. Duomenų bazė.
- 4.22 pav. 10 geriausiai rankos erdvę apibūdinančių tikrinių vektorių -  $u_k$ .
- 4.23 pav. „A“ klasės atvaizdų svorių atstumai nuo kitų klasių.
- 4.24 pav. Atvaizdų svorių atstumai nuo bendros rankos erdvės.

## IŽANGA

Vienas iš pačių pagrindinių ir būtiniausių žmonių poreikių yra **bendravimas**. Vienas kito supratimas – būtina normalaus gyvenimo sąlyga. Aklieji bendrauja Brailio raštu, kurtieji - rankų gestų kalba. Psichiškai ir fiziškai neįgalieji šiandien įvairiose pasaulio šalyse naudoja simbolių kalbą. Negalią turinčių žmonių gyvenime vis didesnę vaidmenį vaidina kompiuteriai ir specialioji programinė įranga. Jie padeda lavinti įvairius pojūčius ir moko geriau orientotis aplinkoje, ugdo negalią turinčiųjų savarankiškumą. Jei kompiuteris suprastų simbolių kalbą, jų gyvenimas, bendravimo kokybė galėtų iš esmės pasikeisti.

Kitas aspektas yra įvairių įrenginių valdymas. Įsivaizduokite kompiuterį, kuris išsijungia kai jūs jam pamojuojate ranka atsisveikindami, ar televizorių, kuriame kanalus į priekį ir atgal perjunginėjate rodydami nykščiu į viršų ir į apačią, kambarį, kuriame šviesą įjungiate, išjungiate pirštų spragtelėjimu. Įsivaizduokite pasaulį, kuriame paprastos, kasdieniškos operacijos būtų valdomos be jokių mygtukų, svirčių, valdymo pultų. Viską galima būtų valdyti paprasčiausiu rankos mostelėjimu [1].

Rankos gestų analizė - sunkus uždavinys, kurį galime išskirti į kelias sritis: judesių analizė, tekstūros, struktūros atpažinimas, mašinų mokymas ir netgi psicholingvistinių studijos.

Pagal įvairius panaudojimo būdus gestai gali būti suskirstyti į kelias kategorijas: valdymo, kontrolės, manipuliavimo ir bendravimo.

Šiuo metu pagrindinės valdymo, navigacijos priemonės yra pelė, klaviatūra, vairalazdė. Tačiau jos yra dirbtinės ir nepatogios, nėra tiesioginio valdymo. Blogai, todėl, kad šiomis priemonėmis sunku atkurti visus laisvės laipsnius, būdingus tikrovei ir didesnių matmenų erdvėse. Būtų labai gerai, jei kompiuterių valdyme sąsaja su vartotoju būtų natūrali. Ji turėtų būti paprasta, intuityvi, nereikalaujanti specialų žinių, panaši į kasdieninį, įprastą bendravimą tarp žmonių. Tokiu galėtų tapti rankų gestų panaudojimo metodas.

Rega padeda žmonėms suvokti supantį pasaulį. Kompiuterių „regėjimas“ įvairiais sensoriais siekia kopijuoti šitą žmonių sugebėjimą, tai daroma analizuojant vaizdus. Regėjimo suteikimas kompiuteriams yra sunkus uždavinys, nes mes gyvename trimačiame pasaulyje ir kai kompiuteriai analizuoja šitą informaciją, dažniausiai ją verčia į dviejų matmenų erdvę, tuo būdu prarandant didelę dalį informacijos. Atliekama daug tyrimų šia tema, tačiau nėra rastas geriausias sprendimas ir

sukurtas galutinis produktas, leidžiantis panaudoti ranką, kaip priemonę kompiuterio ir žmogaus bendravime.

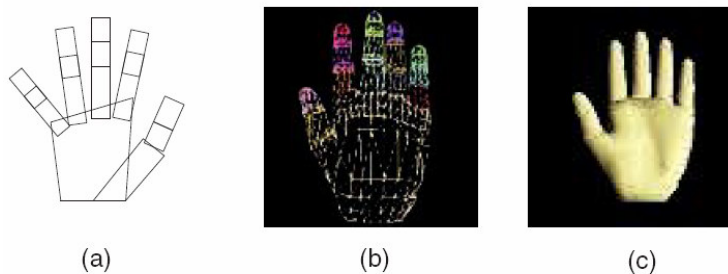
Šiame darbe sieksime išanalizuoti rankos panaudojimo, kaip natūralios priemonės žmogaus ir kompiuterio sąveikoje, metodus ir sukurti programą rankos gestų atpažinimui.

# 1. TIRIAMOJO OBJEKTO – RANKOS - SAVYBĖS

## 1.1. Rankos formos modeliavimas

Delnas ir pirštai - kinematinio požiūriu sistema, sudaryta iš tarpusavyje sujungtų atskirų dalių, taigi jų judėjimas yra apribotas sąnarių paslankumu. Taip pat nemažai suvaržymų, apribojimų yra pačiame rankos judesyje, todėl tokių sistemų modeliavimas iš vienos pusės sudėtingas. Dažniausiai rankos modeliavimą galima suskirstyti į kelias sritis: formos ieškojimas, kinematika, dinamika ir semantika.

Rankos formos modeliavimas gali būti padalintas į kelias grupes: geometrinis modelis, fizinis modelis ir statistinis modelis. Geometrinis modelis tinka trimačių vaizdų generavimui ir rankos animacijos kūrimui. Taip pat šį modelį galime panaudoti tiriant rankos judesius, taikant analizės – sintezės metodą. Fizinis ir statistinis modeliai tiria rankos deformavimąsi. Tik fizinis - tiria tikslų formos deformavimąsi atvaizde, o statistinis - charakterizuoja rankos deformavimąsi pagal mokymo rinkinių analizę. Kreivėmis pagrįstas geometrinio paviršiaus modelis, tai visuma geometrinių paviršių, aproksimuotų įvairiomis kreivėmis. Šis modelis realistiškas, bet modeliuojant reikia labai daug parametrų. Paprastesnis būdas - ranką ar kūno dalį aproksimuoti įvairiomis geometrinėmis figūromis, pavyzdžiui, cilindrais. Šio metodo privalumas: reikia mažiau pradinių parametrų, gaunamas geras vaizdinis rezultatas. Kitaip nei parametriniai, laisvos formos rankos modeliai yra formuojami trimačių taškų rinkiniais. Daugiakampių tinklelis, kurį sudaro trimačiai taškai, aproksimuoja rankos formą, tai gerai tinka skaičiavimams. Skaičiavimo efektyvumui galima panaudoti „cardboard“ metodą. Kiekviena iš dalių tokiame modelyje yra dvimatė plokštuma, jų tarpusavio sujungimo kampai galima derinti 1.1 pav.



1.1 pav. Rankos geometriniai modeliai:  
a) „cardboard“ modelis; b) kontūrinių rėmų modelis c) daugiakampių tinklelių modelis.

Fiziniai rankų modeliai apibrėžia rankos formos pakitimą veikiant įvairioms jėgoms. Modelio judėjimas yra aprašomas Niutono dinamika. Rankos formos išlaikymui naudojamos

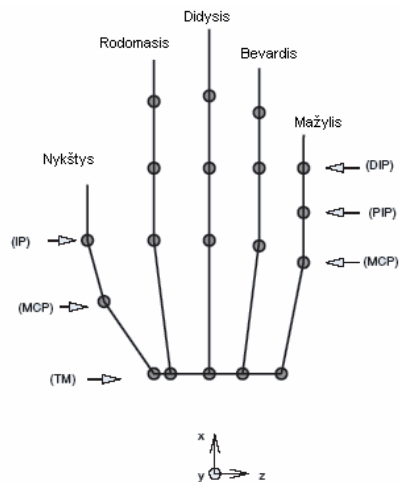


vidinės jėgos, o išorinės jėgos pritaiko modelį prie atvaizdo duomenų. Tokio modelio pavyzdys - simpleksinis tinklelinis modelis.

Statistinis rankos formos modelis per mokymo pavyzdžius apmokomas, kaip kinta formos. Mokymo pavyzdžiais gali būti dvimačiai atvaizdai, arba tam tikra atvaizdų grupė. Vidutinė forma ir būsenų variacija randama panaudojant principinių elementų analizę (Principal Component Analysis - PCA).

## 1.2. Rankos kinematinė struktūra

1.2 pav. pavaizduotas rankos formos skeletas. Kiekvienas pirštas turi tris sąnarius. Tuo remiantis galima sukurti vieną iš labiausiai paplitusių modelių, kai visa žmogaus ranka aprašoma 27 laisvės laipsniais.



1.2 pav. Rankos skeletas.

Kiekvienas pirštas modeliuojamas kaip kinematinė grandinė, kurios pradžia yra delno pabaiga, o piršto galiukas - grandinės pabaiga.

Atvirkštinis kinematikos uždavinys - apskaičiuoti sujungimo kampus, analizuojant pirštų judėjimą. Daugiausiai naudojami gradientais – nuokrypiais paremti metodai. Kitas būdas spręsti šitą užduotį - genetinis algoritmas. Natūralus rankos judesys yra suvaržytas tam tikrų dėsnų. Vieni iš jų - statiniai dėsniai, piršto judėjimas apribotas rankos anatomicinės struktūros. Pirštas gali judėti tam tikru kampu, sąlyginai nuo nulio iki devyniasdešimt laipsnių. Kitų rūšių apribojimais apibūdina koreliaciją tarp skirtingų sujungimų, tuo sumažindami rankos judėjimo erdvės matmenis.

Dinamikos modeliavimas. Sudėtingo rankos judesio fiksavimui ir ištisinių gestų atpažinimui, yra modeliuojama rankos judesio dinamika ir semantika. Naudojamas Kalmano

filtravimas ir išplėstinės Kalmano filtravimo technikos. Tokie filtravimai tinka sekimo (angl. k. – „tracking“) uždaviniams. Tačiau jie remiasi sąlyginai mažo judesio analizavimu, kai atstumai didesni, rezultatai tampa nepatikimi. Paprastus judesius galime modeliuoti baigtinių reikšmių metodu, tačiau šitas metodas netinka analizuojant sudėtingus gestus. Taisyklėmis paremtais algoritmais galima modeliuoti komplikuočius gestus. Norint sukurti taisykles reikia įvesti daug euristikos (negatyvi euristika, iš graikų kalbos „heurisko“ – atrasti, taisyklių, nurodančių, kokių veiksmų reikia vengti realizuojant programą, ir pozityvią euristiką, nurodančią, kaip programa turi būti realizuojama kuriant hipotezes). Daug panašumų yra tarp ženklų kalbos ir šnekos, tuo remiantis dinamiškai modeliuoti galime panaudoti HMM – paslėptuosius Markovo modelius. HMM apibendrinimui gali būti panaudoti dinaminiai Bajeso tinklai. Šitie būdai yra esminiai mašinių mokymo metodai, naudojamys mokymą ir treniravimą. Dinamikos ir semantikos analizė nėra aiškiai išskirta šiuose metoduose, tačiau yra jų sudedamoji dalis. Mokymo rezultatai priklauso nuo parinktų mokymo duomenų ir mokymo būdų. Viena iš pagrindinių problemų yra tiesioginė rezultatų priklausomybė nuo mokymo duomenų. Tačiau mokymo duomenų parinkimas lyginant su visu procesu yra tik maža problema. Šiuo metu dinamikos analizė traukia nemažai HCI tyrinėtojų, kompiuterinės grafikos, ir psichologijos specialistų.

Rankos judėjimas. Žmogaus rankos judesį galime padalinti į rankos judesį ir pirštų judesius, kuriuos galime aprašyti kaip:

$$J=[JG,JL] \quad (1.1)$$

čia JG – globalus rankos judesys;

JL – lokalūs judesiai (pirštų).

Globalus rankos judesys taip pat gali būti aprašytas taip:

$$J=[S,V] \quad (1.2)$$

čia S – rankos pasukimas;

V – rankos pavertimas.

Dar vienas svarbus uždavinys - patikimai nustatyti rankos judėjimą atvaizdų sekoje. Lokalūs rankos judesiai gali būti charakterizuojami sujungimo kampais. Vienas iš galimų būdų analizuoti judesį, tai rankos formos pakitimo fiksavimas atvaizduose. Kitas būdas, turint trimatį rankos modelį, ieškoti padėčių atitinkančių vaizdą užfiksuotame kadre, nufilmuotame viena kamera. Panaudojus kelias kameras galima būtų atskirti okliuzijas, tokias padėtis, kai vienas paviršius

užstoja kitą. Trimačio modelio panaudojimas palengvina paviršiaus struktūros iškilimų ir įdubimų, kitaip dar vadinamų - gylio radimą atvaizde [\[2\]](#).

## 2. RANKOS PARAMETRŲ PERDAVIMAS Į KOMPIUTERĮ

### 2.1. Naudojamos techninės priemonės

Mechaninis atpažinimas - gestų ir ženklų kalbos atpažinimas panaudojant įvairius mechaninius prietaisus. Daug techninių priemonių yra naudojama kūno padėties nustatymui. Dažniausiai tai daroma pagal vaizdo atpažinimą - taikant kameras, judančias šviesas, arba pritaikant tam tikrus specialius prietaisus, tokius kaip pirštines, padėties sekėjus (angl.k. – „trackers“), taip pat taikant kombinuotus kelių tipų variantus. Mechaninio atpažinimo priemonės dažniausiai brangiai kainuoja, tai mažina jų panaudojimo sritį. Egzistuoja specialūs rankos formos skaneriai taikomi apsaugos sistemose, jie atlieka biometrinius rankos matavimus. Randamas rankos plotas, geometrija, pirštų atspaudai, odos spalva. Tiesioginių atstumų gavimui galima panaudoti Cyber Scanner, MRI („Magnetic Resonance Imaging“) techniką ir kitus skaitmeninius erdvės keitiklius.

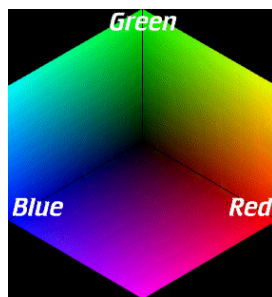
Esama projektų, kuriuose virtualios realybės valdymui ir gestų analizei naudojamos sensorinės pirštines. Tokio prietaiso naudojimas gana nepatogus, nes prie pirštinių prijungti laidai judėjimą daro nepatogų ir sunkų. Tokiomis pirštinėmis, pavyzdžiui, daktaras galėtų operuoti pacientą per atstumą ar valdyti jautrią aparatūrą. Galima šį metodą panaudoti judesių treniravimui. Tačiau paprastam kasdieniniam vartojimui toks prietaisas netinka dėl savo brangumo ir judesių suvaržymo[8].

Didelė žmogaus smegenų dalis skirta regėjimui - vienam iš penkių mūsų aplinkos suvokimo jutimui. Žmonės sprendžia dauguma su regėjimu susijusių problemų beveik be pastangų, kartais to net nesuvokdami. Perduoti šį sugebėjimą mašinai sudėtinga problema. Be abejonės matančias mašinas padaryti įmanoma, pavyzdžiui, gamyklose jos kiekvieną dieną vizualiniu būdu patikrina milijonų lempučių kaitinimo siūlus. Sukurtos automatinės atpažinimo sistemos, kurios gali identifikuoti žmogų pagal akies rainelės atvaizdą ir veikia sistemos, kuriose mašina kelyje yra valdoma kompiuterio, panaudojant tik vaizdo kamerą. Realus vaizdo atvaizdas gaunamas kamera panašus į žmogaus akies perduodamą vaizdą į smegenis, tai trimačio pasaulio dviejų matmenų atvaizdų projekcijos. Toks informacijos praradimas žmogaus smegenims nesudaro jokių sunkumų, kurios vientisai, pastoviai atnaujina pasaulio modelį. Sugebėjimas spręsti apie gylį iš matomų objektų vaizdų skirtumų teikia smegenims vienus iš pagrindinių duomenų apie realų pasaulį. Vis dėlto netgi kai stereoskopinė, iš dviejų akių, informacija neprieinama, žmogaus smegenys gali interpretuoti duomenis ir tiksliai nustatyti objekto padėtį ir orientaciją. Tai įmanoma milžiniškų žinių dėka, kaip smegenys suvokia trimatį pasaulį, bendrų taisyklių, formų, objektų struktūros žinojimas ir kokias projekcijas objektai sukuria tinklainėje [3].

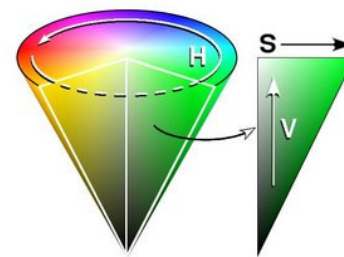
Nekontaktiniems metodams priklauso vieno ar kelių kamerų panaudojimas. Panaudojus kamerų teikimą vaizdinę informaciją galima kurti vaizdo analizės algoritmais pagrįstus metodus. Tokiems metodams nereikia tiesioginio kontakto tarp prietaiso ir žmogaus, jie yra vieni iš pigiausių.

## 2.2. Skaitmeninis atvaizdas gaunamas kamera

Panaudojant kamerą į kompiuterį ar kitą informacijos apdorojimo priemonę gauname skaitmeninį vaizdą, kurio kokybė priklauso nuo naudojamų prietaisų ir aplinkos sąlygų (pvz.: šviesumas). Skaitmeninis atvaizdas - tai pastovaus skaičiaus eilučių ir stulpelių iš pikselių sudarytas masyvas. Pikselis – mažiausias atvaizdo elementas. Kuo daugiau pikselių atvaizde tuo smulkesnis vaizdas, daugiau informacijos, didesnė rezoliucija. Kuo didesnė rezoliucija, tuo daugiau informacijos reikia apdoroti kompiuteriui ir kartais tikslinga rezoliuciją sumažinti, atsižvelgiant į prarandamos informacijos kiekį. Spalvų sistemos, erdvės standartiniais kodais aprašo kaip atrodo vaizdas. Vėliau panaudojant spalvų sistemos reikšmes galima jį atkurti. Naudojamos RGB, HSV ir kitos spalvų sistemos. RGB spalvų schema dažniausiai naudojama kompiuteriuose. R - atitinka raudoną spalvą, G – žalią ir B – mėlyną. Vieno pikselio reikšmės gali kisti nuo 0 iki 255, kiekviename iš trijų RGB sluoksnių. RGB spalvų sistema galima aprašyti 16,7 mln. spalvų. RGB suteikia galimybę atvaizduoti platų spalvų spektrą. Tačiau kalbant apie spalvas, taip kaip jas mato žmogus, artimesnė būtų HSV spalvų sistema. Kartais spalvas, kurias žmogus mato kaip labai artimas, RGB spalvų sistema apibūdina su dideliu skirtumu. HSV sistema: H atitiktų atspalvį; S – išotinimą; V – kiekį arba intensyvumą. Atspalvis apibūdina raudoną, žalią, purpurinę spalvą, išotinimas – kaip spalva nutolusi nuo to paties intensyvumo pilkos spalvos, o kiekis apibūdina šviesumą.



RGB spalvų schema



HSV spalvų schema

2.1 pav. Spalvų kodavimo sistemos.

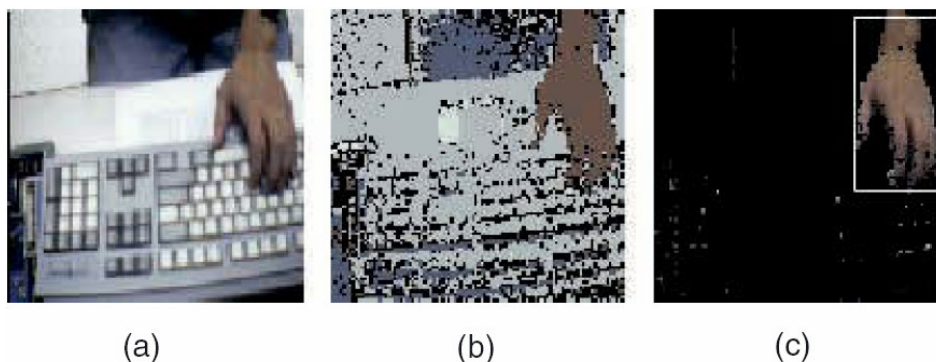
Priklausomai nuo pasirinktos spalvų kodavimo sistemos, parenkami atvaizdo apdirbimo metodai.

### 3. INFORMACIJOS APIE RANKOS GESTĄ APDOROJIMAS

#### 3.1. Pradinis informacijos apdirbimas – „preprocessing“

Rankos išskyrimas iš supančios aplinkos sudėtingas uždavinys, reikalaujantis ne mažiau pastangų negu pats atpažinimas.

Rankos lokalizavimas. Lokalizacija - tai rankos regionų radimas atvaizdų sekoje. Dažnai gana efektyviai tam tikslui panaudojama odos spalva. Spalvos - pagrindas spalvinei segmentacijai. Pagal spalvos skyrimo būdus galima išskirti du metodus: neparametrinis („nonparametric“) ir parametrinis („parametric“) 3.1 pav.



3.1 pav. Rankos lokalizavimas a) įėjimo atvaizdas; b) segmentacijos rezultatas; c) išskirta ranka.

3.1 pav. pavaizduotas atvaizdo segmentavimas pagal rankos odos spalvą, išskiriami spalvą atitinkantys pikseliai ir grupuojami į artimus padėtimi regionus, didžiausias gaunamas regionas – ranka. Vienas iš neparametrinių būdų – histogramų panaudojimas atpažinime. Kadangi histogramose yra kiekybiškai įvertinama spalvų erdvė, tai rezultatai tiesiogiai priklauso nuo to, kokį vertinimo tikslumą pasirinksi. Kitas būdas panaudoti SOM – „self organising maps“, „neprižiūrimą mokytojo“ - klusterizavimo algoritmą spalvų pasiskirstymo aproksimavimui. Visi paminėti neparametriniai metodai veikia gerai, kai gerai parinkti kvantavimo, kiekybinio įvertinimo parametrai ir yra pakankamai pradinių duomenų. Parametriniai metodai modeliuoja spalvų tankumą parametrinėmis formomis, pavyzdžiui pagal Gauso pasiskirstymo dėsnius, ar Gauso mišinių modelius. Tikimybės maksimizavimo būdai leidžia parinkti modelį, atitinkantį stebėjimo duomenis. Šituose modeliuose be spalvos, galima įtraukti ir rankos formos, judesio parametrus.

### 3.2. Rankos gestui būdingų savybių išrinkimas

Savybių gesto atvaizde (kadre, paveikslėlyje) išrinkimas. Modelio įvertinimui reikia žinoti jo savybes. Rankos geometrinės savybes charakterizuoja taškai, linijos, kontūrai ir siluetai. Dažnai kaip parametras naudojami pirštų galiukai, pagal jų išsidėstymą galima atpažinti nemažai gestų. Taip pat naudojami spalviniai žymekliai rankos orientacijai nustatyti. Rankos pozicija randama, orientacijai pritaikant trimačius cilindrus. Pirštų galiukų padėčiai rasti pritaikomos linijos [2].

Informacijos surinkimas yra tik pirmas žingsnis. Kitas žingsnis - atpažinti gestą ar ženklą, kuris buvo užfiksuotas. Žymiai sunkiau tai padaryti nenutrūkstančiame informacijos sraute, realiu laiku.

Vienas iš pagrindinių sunkumų, kuriant automatinę gestų atpažinimo sistemą yra laikina, kintama dinaminių gestų struktūra. Skiriasi net vienos gesto reikšmės galimos padėtytys. Kai žmonės rodo tą patį gestą, gestai gali sutapti tik dalinai, net vienas žmogus tą patį gestą parodo skirtingai, neidentiška. Taip pat sunkumų kelia aplinka, jos įtaka atpažinimui labai didelė (apšvietimas, šešėliai).

### 3.3. Klasifikavimo metodai

Tikriausiai paprasčiausias vizualiniu modeliu paremtas mašinų regėjimas yra pavyzdžių gretinimas. Jei žinomas objektas ar savybės, kuriuos reikia rasti atvaizde, sukuriama pavyzdys, turintis objekto savybes. Po to pavyzdys yra gretinamas, taikomas įėjimo atvaizdai visose padėtyse. Modelio ir šablonu atitikimas, bet kurioje padėtyje gali būti įvertintas. Galime nustatyti (su tam tikra paklaida) objekto ar savybių padėtį. Toks daug laiko trunkantis modelio sudarymas gali būti taikomas esant suvaržytoms, apribotoms aplinkos ir įėjimo padėtims, situacijoms, pavyzdžiui toks modelis labai tinka pramoninei veiklai. Modeliui pritaikant mašinas, realiu laiku pasiekiamas didelis pavydžių suliginimų skaičius [3].

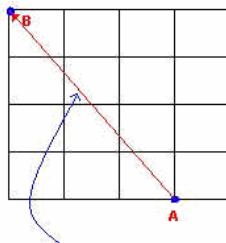
Įvertinti gautų rezultatų atitikimą, panašumą tarp savybių naudojami įvairūs atstumai. Dažnai tam yra naudojamas Euklido atstumas:

$$d_e = \sqrt{\sum_{i=1}^n (x_i - i_i)^2} \quad (3.1)$$

čia  $d_e$  - atstumas;

x ir i - kintamieji nusakantys padėtį.

Euklido atstumu galima laikyti trumpiausią atstumą tarp dviejų taškų, iš esmės tai Pitagoro lygybė dviejų matmenų erdvėje 3.2 pav.



3.2 pav. Euklido atstumas: Pitagoro taisyklės pritaikymas, trumpiausias atstumas tarp dviejų taškų.

Kontūrų modeliai. Reikšmingas Kass darbas naudojant „Snake“ metodą arba dar kitaip vadinamą „Aktyvių kontūrų“ modelį, kuris naudoja pusiau automatinį metodą atvaizdo segmentavimui. Tai atliekama panaudojant energiją minimizuojančias kreives. Tyrimo pradžioje vartotojas apibrėžia ranka dominančią sritį atvaizde. Tada elastingas kontūras pritaikomas ribos taškams ir kreivė iteratyviai tobulinama, kol vidinė energija, apibūdinama kreivumu, minimizuojama pagal veikiančias išorines jėgas. Jėgos pridedamos palei atvaizdo kraštus. Daug tyrinėtojų parodė kaip „Aktyvūs judantys kontūrai“ panaudojami objektui atvaizde rasti ir sekti.

Zhou ir Pycock segmentavo dviejų matmenų atvaizdų elementus naudodami statistinius modelius panašiai kaip „Snake“ metodas. Iš segmentuotų elementų kuriami pritaikomi modeliai. Interpretacijos procese, panaudojant Bajeso atstumo matavimų reikšmes, optimizuojami modeliai ir duomenų sutapimai. Lobregt ir Viergever išplėtė šį modelį, pateikdami sprendimus nepageidaujamos deformacijos problemoms. Šios problemos tai objekto sutraukimas, sumažinimas („shrinking“) ir vertikalus klasterizavimas. Yra daug metodų, kurių pagrindas yra Kass pasiūlytas „Snake“ būdas. Skirtumai tik naudojamuose suvaržymuose, kurie leidžia padidinti informacijos kiekį apie rankos formą, tai - „a priori“ duomenys. Tuo pasiekiamas geresnis proceso efektyvumas.

Terzopolous ir Vasilescu išplėtė „Snake“ metodą įtraukdami išsipūtimo jėgą, kuri leidžia pašalinti vartotojo daromą rankinį pradinio kontūro nustatymą ir tuo pačiu pašalinti vietinio minimumo konvergavimą. Išsipūtimo jėga, tai jėga, kuri „Snake“ modelyje, tarsi pučiamas balionas išplečia kontūrus iki objekto ribų atvaizde. Terzopolous ir Vasilescu šį modelį formulavo kaip baigtinių elementų tinklą ir vėliau išplėtė jį iki plono, lėkšto splaino. Savo metodo privalumus jie pademonstravo sėkmingame duomenų atkūrimo ir paviršiaus atvaizdavime. Bowden taip pat dirbo panašiais metodais paremtus darbus. Keletas tyrinėtojų pasiūlė aktyvaus kontūro modelių kūrimo metodus su patobulintomis „B-Spline“ variacijomis. Schnabel ir Arridge tyrė problemas susijusias su dideliu išlinkimu aktyviuose kontūro modeliuose, siūlydami lyginimo techniką „isophoto“



išlinkimų lyginimui. Jie ieškojo konkrečių pritaikymų tokiam metodui, pavyzdžiui - didelio kiekio kreivių išlinkimų segmentavimas medicininiuose smegenų atvaizduose. Blake ir Isard sujungė daugelį savo idėjų straipsnyje „Aktyvūs Kontūrai“, jame apimamas taikomųjų programų sudarymas ir objekto padėties sekimas įvairiomis kontūrų „B-spline“ aproksimacijomis. Buvo įrodyta, kad šitie dviejų matmenų modeliai gali būti panaudojami atkurti trijų matmenų paviršiams, tam reikia duomenų, kurie randami „Snake“ modeliu. Kiekviename atvaizde randami kontūrai ir iš jų kuriamas trimatis modelis. Tipinis tokių sistemų pritaikymas: panaudojamas vieno pasirinkto atvaizdo galutinis modelis, gauta informacija priskiriama pradiniais duomenims - tuo išvengiama vartotojo įsikišimo, ranka daromų pradinių kontūro nustatymų. Ivins ir Porrill pristatė „Aktyvių regionų modelių“, Kass sukūrė „Aktyvių kontūro modelių“ pritaikymą. Jame tam tikros spalvos sritys atvaizde panaudojamos rasti ir sekti kontūrus. Chiou pasiūlė neuronų tinklų metodą. Jis pagrįstas stochastiniu aktyvių kontūrų modeliu (NNS-SNAKE), kuris sujungia neuronų tinklų klasifikatorių, sistematinėms žinioms kurti ir aktyvių kontūrų modelį, automatizuotam kontūrų padėties radimui. Skaičiavimuose panaudojamos energijos funkcijos stochastiškai nustatyti labiausiai tikėtiną kontūrą.

Statistiniai deformacijos modeliai. Taško pasiskirstymo dėsnis („Point Distribution Model,“) arba PDM gauna „Smart Snake“ pavadinimą pagal akivaizdų panašumą į elastinio „Snake“ modelį „Aktyvių kontūrų modelį“. Pagrindinis skirtumas tai, kad „Snake“ modelis išlaiko formos informaciją elastingumo ir standumo sudėtiniuose taškuose, o PDM naudoja statistinį modelį leistinų deformacijų apibūdinimui. Tuo sumažinamas skaičiavimų skaičius ir tada lengviau modeliuoti modelio deformaciją. Po to kai Cootes pasiūlė šitą modelį buvo atliekami tolesni PDM pritaikymo tyrimai. PDM (matematinis modelis) arba „Aktyvios formos modelis“, tai statistinis modelis, kuris kuriamas iš tinkamai parinkto mokymo rinkinio. PDM objektas yra sužymėtų taškų rinkinys, kartu su jų vidutinėmis reikšmėmis ir mažu būsenų rinkiniu, kurie apibūdina kaip gali kisti objekto forma. Šitos būsenų variacijos „Principinių komponentų“ analize gaunamos iš mokymo rinkinio ir reprezentuoja didžiausius tikrinius vektorius iš kovariacijos matricos. „Aktyvios Formos modelis“ naudoja tiesinį PDM formulavimą itaratyvos paieškos metodikoje, kuri gali greitai rasti modeliuojamas struktūras triukšmingame, netvarkingame atvaizde. Turk ir Pentland pristato metodą, kurį naudojant ištraukiamas tik toks tikrinių vektorių skaičius, kuris lygus mokymo pavyzdžių skaičiui, o ne rinkinio matmenims, panašiu būdu kaip Cootes. Bowden buvo įrodyta, kad PDM leidžia sumažinti dimensijas ir galima panaudoti paprastą klasifikavimą pagal statinę formą. Šitie autoriai aprašo metodus kaip panaudoti matmenų sumažinimo būdą, pagal svorių variacijas vidurkiuose, formos deformacijos klasifikavimui. Jie demonstruoja kaip statiniai rankos gestai gali būti atpažįstami realiu laiku panaudojant PDM. Lantis, Taylor ir Cootes pratęsė savo pradinių darbų nuo kontūro modelių iki formos analizės ir pilkos spalvos atvaizdų panaudojimo technikų. Jie

naudojo kombinuotą PDM, kuris pagal formą ir pilkos spalvos žemėlapius gali rasti ir identifikuoti žmonių veidus. Turk ir Pentland naudojo principinių komponentų analizę, kuri apibūdina veido atvaizdus bazinių funkcijų rinkiniu arba dar kitaip vadinamais „tikriniais vektoriais“. Teisingi pokyčių modeliai yra nustatomi iš mokymo rinkinio ir turbūt dėl to yra labiau tinkami negu „fizinis modelis“, tačiau tikriniai veidai neįtraukia formos pakylėms ir nelabai tinka kai smarkiai kinta veido padėtis ir mimika. Toks modelis gali būti lengvai pritaikytas atvaizdui naudojant koreliacija paremtus metodus. Magee ir Bole pristatė „Vektoriaus pasiskirstymo modelius“. Jų pagalba taškai, nurodantys sujungtą kontūrą yra paverčiami į vektorių ir suvedami į jungtinį duomenų rinkinį – galutinį mokymo vektorių. Pagal galutinį mokymo vektorių atliekami PDM skaičiavimai. Šie autoriai diskutavo apie „Kanonines analizės“ naudą, tai procedūra panaši į PDM skaičiavimą. Joje suformuojamos dvi kovariacijos matricos. Viena iš jų - „Intra“, apibūdina vidinės klasės variacijas, o kita - „Inter“, išorinės klasės variacijas. Po apibendrintos tikrinės sistemos sudarymo, randama nauja tikrinė erdvė. Nors ši erdvė nebūtinai turi būti optimizuota matmenų sumažinimui, ji naudinga duomenų klasifikavimui pagal vidinės klasės variacijų pirmus komponentus. Swets ir Weng pristatė savo sukurtą techniką, kuri pavadinta – kombinuota tikrine kanonine transformacija. Ji apjungia kanoninę su principinių komponentų analizę. Privalumas - gauname sumažintą naudingų duomenų kiekį ir pagerintą klasifikavimą. Kanoninė analizė atliekama po to kai duomenys suprojektuojami ant žemesnės tikrinės erdvės, kuri gauta principinių komponentų analize. Pradinį PDM (Aktyvios formos modelį) išplėsti iki trijų matmenų pasiūlė Hill. Ferryman naudojo „Principinių komponentų analizės“ modelį naudodamas nelanksčius trijų matmenų modelius. Tyrėja modeliavo skirtingų formų mašinių deformacijas. Vėliau tai panaudojo judančiam eismui rasti ir sekti. Šis procesas labai panašus į PDM. Tačiau vietoj to, kad modeliuoti objektą taškais, kuriančiais objekto ribas, pasirinkti tokie specialūs taškai kaip kampai, modelis sukuriamas pagal žinomus tų taškų tarpusavio ryšius. O’Toole pristatė savo darbą – „Trimačiai veidų modeliai“. Jo metodo pagrindas - vidutinis veidas su svoriais, kurie panaudojami veidą deformuoti. Veidai atkuriami kaip trimačiai paviršiai iš šešiasdešimt penkių vyrų ir šešiasdešimt penkių moterų galvų atvaizdų rinkinio. Principinių komponentų analizė sukuria kompaktišką modelį. Jie parodė, kad pradinė tikrinių veidų duomenų rinkinio kitimų moda suteikia galimybę stebėti variacijas tarp vyro galvos ir moters galvos.

Netiesiniai PDM. Tiesinis PDM formuluojamas darant prielaidą, kad panašios objekto formos duos išėjime panašius vektorius. Šiuo atveju daroma palanki prielaida, kad mokymo rinkinys sugeneruos klasterį tam tikroje formos erdvėje. Nagrinėjant sudėtingesnius modelius, daroma prielaida, kad mokymo rinkinys gali sugeneruoti daugialypius atskirus klasterius formos erdvėje. Tokiomis aplinkybėmis tiesinis PDM nebetiks, kadangi netiesiniai mokymo rinkiniai išduos didelių matmenų formas, kurios modeliuojamos tiesiniais PDM sukurs nepatikimus modelius. Kai žinoma, kad modelyje yra sukamasis netiesiškumas, jį galima pašalinti atvaizduojant

ant alternatyvios tiesinės erdvės. Heap ir Hogg pasiūlė naudoti „log polar“ žymėjimą, atvaizdavimą netiesiškumo pašalinimui iš atvaizdo. Tai leidžia netiesinio pobūdžio mokymo rinkinį suprojektuoti ant tiesinės erdvės, ten panaudojant „Principinių komponentų analizę“ rasti ieškomas deformacijas. Po to šis modelis suprojektuojamas atgal į originalią erdvę. Šis požiūris naudingas tokiems pritaikymams, kai egzistuoja pagrindinis vienas netiesiškumo požymis ir jis matomas originalaus objekto paviršiaus taškų keliuose. Šis metodas negali būti naudojamas kai egzistuoja aukštas netiesiškumas, sugeneruojamas iš kitų šaltinių. Aukštesnės eilės nelijiniškumas dažnai yra neteisingo mokymo rinkinio duomenų parinkimo rezultatas. Rūpestingai parenkant paviršiaus taškus ranka galima rasti beveik optimalius duomenis, toks parinkimas minimizuoja mokymo rinkinio netiesiškumą. Tačiau visiems, išskyrus pačius paprasčiausius atvejus, toks sprendimas nepriimtinas. Tiriant formą dažnai naudojamos pusiau automatinės procedūros, kai vartotojas savarankiškai pagreitina pavyzdžių sužymėjimą. Pilnai automatinės procedūros naudojamos retai, kadangi iškyla problemų dėl teisingo pavyzdžių išrinkimo. Baumberg ir Hogg darbas sprendžia aukštos eilės netiesiškumo problemas panaudojant „B-Spline“ metodą. Paviršiaus taškai šiam metodui randami PDM būdu. Išlinkimų sprendimas pagal „B-Spline“ metodą sumažina netiesiškumą. Kotcheff ir Taylor pasiūlė netiesiškumą pašalinti automatiškai parenkant paviršiaus taškus. To galima pasiekti analizuojant tiesiškumo dydį mokymo rinkinyje pagal PDM. Kuo didesnės eilės netiesiškumas gaunamas sudarinėjant mokymo rinkinį, tuo didesnių skaičiavimų kiekį reikia atlikti panaudojant PDM, tam, kad išanalizuotumėme visas deformacijas. Procedūra buvo pademonstruota naudojant mažą testinę formą ir specialiu būdu sužymint paviršiaus taškus, tai buvo daroma atsižvelgiant į mokymo rinkinio dydį (jis gautas analizuojant principines padėtis ir pagal tai kaip modelis deformuojasi tose padėtyse, pvz.: tikrinės kovariacijos matricos reikšmės). Toks metodas formuluojamas kaip minimizavimo problema, sprendimui panaudojant genetinį algoritmą. Šis būdas gerai veikia, bet reikia atlikti daug skaičiavimų. Tačiau analizuojant labai sudėtingus trimačius objektus šis metodas tiesiog nepamainomas. Bendresnis tokios problemos sprendimas būtų naudoti netiesinius vaizdavimus. Kaip tiesinis „Principinių komponentų“ analizės būdas yra naudojamas tiesiniams PDM modeliams, taip netiesinis „Principinių komponentų“ analizės būdas gali būti naudojamas moduluoti netiesinius PDM modelius. Sozou pirmas pasiūlė polinominės regresijos būdą aprašyti aukštos eilės polinomams, mokymo rinkiniams su netiesine ašimi. Nors šis būdas ir kompensuoja mokymo rinkinio išlinkimus, jis neadekvačiai kompensuoja aukštesnio lygio netiesiškumus. Tai akivaizdžiai matoma mažesnėse pakitimų būsenose su aukštais svyravimais, okliuzijomis. Be to, turi būti pasirinkta polinomo eilė ir pritaikymo procesas ilgai užtrunka. Vėliau Sozou pasiūlė modeliuoti mokymo rinkinio netiesiškumą panaudojant atgalinio sklaidimo neuronų tinklus, kurie atlieka netiesinę „Principinių komponentų“ analizę. Šitas metodas gerai vykdo savo užduotį, tačiau neuronų tinklo architektūra kiekvienai taikomajai programai yra

specifinė. Mokymo trukmė ir optimizavimo laikas ilgai trunka. Visų šitų metodų pagrindinė užduotis tiksliai sumodeliuoti netiesiškumą, tik su tokiu pat paprastumu ir trukme kaip tiesiniuose modeliuose. Keletas tyrinėtojų pasiūlė alternatyvas, kurios naudoja netiesines aproksimacijas. Įvertindami netiesiškumą jie naudoja daug mažų tiesinių modelių kombinacijų. Šie metodai parodė savo efektyvumą modeliuojant sudėtingus netiesinius modelius, kurių savybės yra labai aukštos eilės dimencijų erdvėse. Pagrindinis šių požiūrių principas - suskaldyti išlinkimus į paprastas tiesines smulkias dalis, taip apibūdinant netiesiškumą, o ne modeliuojant jį patį. Tai būdas panašus į polinominį paviršiaus atvaizdavimą. Vienalytis kreivių paviršius gali būti išsprendžiamas jį suskaldžius į mažus tiesinius kelius. Kompiuterių grafikoje ši technika atliekama per trumpą laiką. Žinoma yra ryšys tarp matomo tikslumo ir skaičiavimo greičio (naudojamas minimalus polinomu skaičius norimo vaizdo radimui). Su panašia problema susiduriama ir netiesiniuose PDM modelių skaičiavimuose. Ten turime rasti minimalų tiesinių kelių skaičių, tam, kad teisingai, tiksliai atvaizduoti modelį. Bregler ir Omohundro pasiūlė modeliuoti netiesinius žmogaus lūpų duomenis panaudojant „Formos erdvės suvaržymo erdvę“ („Shape Space Constraint Surface). Analizuojant klasterius, erdvės paviršius suskirstomas į tiesinius kelius ir tokie paviršiaus suvaržymai pritaikomi modeliui. Vis dėlto lūpų paviršiui dimencijų skaičius yra palyginus mažas. Cootes ir Taylor pasiūlė modeliuoti netiesinius duomenis naudojant Gauso mišinių modelį, kuris pritaikomas pagal „Laukimo maksimizavimo“ techniką. Daugialypiai Gauso klasteriai pritaikomi mokymo rinkiniui. Tai suteikia patikimesnį modelį, kadangi suvaržymai pritaikomi prie kiekvienos formos erdvės kelio dalelės. Modeliuojama kiekvieno Gauso mišinio dydžiu ir padėtimi. Abi šios sprendimų technikos tampa netinkamos kai matmenų skaičius ir mokymo rinkinio dydis yra dideli. Vis dėlto projektuojant mokymo rinkinį į tiesinį poerdvį, tuo sumažinant netiesiškumo sudėtingumą, galima žymiai palengvinti mokymo rinkinio statistines ir tikimybinės analizes trukmes. Toks projektavimas remiasi „Principinės analizės“ modelio dimensijų sumažinimu, išlaikant svarbią informaciją – mokymo rinkinio formą.

Padėties sekimas. Analizuojant modelio pritaikymo ir optimizavimo metodų problemas sprendžiami duomenų išsimėtymo, netolygumo klausimai. Hill pasiūlė naudoti genetinius algoritmus rasti diskretiems formos pokyčiams. Cootes siūlė vartoti genetinius algoritmus pradinio objekto atvaizde ir PDM modelio pradinių būtinų reikšmių radimui. Buvo iširta padėties sekimui naudotinių genetinių algoritmų pranašumai ir trūkumai lyginant su suskaldytais netiesiniais modeliais. Tačiau, šito tyrimo formulavime didelę reikšmę turi paties genetinio algoritmo struktūra. Blake pabrėžė privalumus naudojant žemų parametrų apibūdinimus deformuojamiems modeliams „B-Spline“ terminais. Šiuo metodu deformuojamas modelis laikomas tiesine bazinių pavyzdžių kombinacija ir modelio būseną apibūdinama tų pavyzdžių koeficientų vektoriais. Šis metodas natūraliai atveda prie Kalmano filtro, kuriame modelis sudaromas pagal detalią vietinę ribų paiešką.

Pasiūlyti žingsniai pagal Kalmano filtrą panaudojami atnaujinti modelį. Ivinns ir Porrill pasiūlė panašų būdą, tik kaip alternatyvą Kalmano filtrui naudojo mažiausių – kvadratų aproksimavimą. Egzistuoja daug metodų objekto padėties sekimui, vienas iš naujesnių – „Kondensavimas„. Blake ir Isard pristatė „Stochastinį kodensavimo tankumo sklidimo“ algoritmą. Jame objekto kontūro padėtis tikimybiškai randama laiko tėkmėje panaudojant objekto dinamines savybes konkrečiu laiko momentu. Objektai nevaizduojami vienu parametrizavimu, bet vietoj to naudojama tikimybinė tankumo funkcija, kuri išreiškia visus galimus modelio parametrizavimo būdus. Iš šito pasiskirstymo kiekvienai iteracijai generuojama daug hipotezių ir jas tikrinant pagal palaikomą informaciją kiekvienam atvaizdui. Sutirštėjimas leidžia susekti objektus, kurie juda trūkinėdami, netolygiai, sudėtingose triukšmingose aplinkose [3].

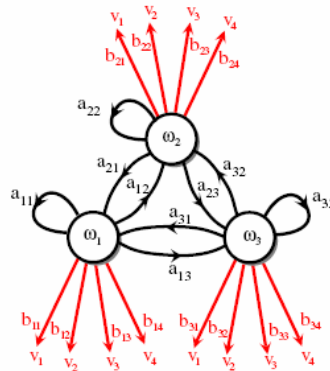
Kai susiduriame su situacija, kuri priklauso nuo laiko, tai yra sudaryta iš procesų, kurie vyksta laiko plotmėje, tai yra turime būsenas laiko momentu  $t$ , kurios įtakojamos būsenų laiko momentu  $t-1$ . Paslėptieji Markovo modeliai labai naudingi spręsti tokias problemas, pavyzdžiui kalbos ir gestų atpažinime. Paslėptieji Markovo modeliai turi tam tikrą skaičių parametrų, kurių reikšmės nustatomos, kad kuo geriau aprašytų mokymo struktūras žinomai kategorijai. Vėliau bandomasis šablonas yra kvalifikuojamas pagal didžiausią posteriorinę (vėlesnę) tikimybę.

Paslėptųjų Markovo modelių panaudojimas. Tai statistinis modelis plačiai naudojamas kalbos atpažinime. Kadangi laikinis gestų atpažinimas ir kalbos atpažinimas yra panašūs, pastaraisiais metais paslėptieji Markovo modeliai naudojami atpažinti žmogaus judesį. Jis naudojamas modeliuoti būsenų kitimą kai kuriuose dinaminuose modeliuose. Yra nemažai paslėptųjų Markovo modelių variacijų. Kai kuriuose darbuose gestai yra modeliuojami multidimenciniais Markovo modeliais, kurie naudoja daugiau negu vieną stebėjimo simbolį, būseną vienu metu. Šis požiūris leidžia modeliuoti skirtingų kelių gestus ir teikia būdus sujungti sudėtinus metodus, padidinti atpažinimo dažnį, apimtį. Kadangi išėjimo tikimybė kiekvienai HMM būsenai yra unikali, tai toks modelis gali apdoroti tik stacionarius procesus ir nėra adekvatus daugialypių gestų modeliavimui. Dalinai stebimas Markovo sprendimų procesas buvo pasiūlytas laikinajam sulyginimui. Standartinis HMM buvo išplėstas iki tiek, kad apimtų ir globalias parametrų variacijas išėjimo tikimybėse, tam, kad valdytume parametrizuotus judėjimus, tokius, kaip dirigavimas muzikoje.

Kai Markovo sąlyga yra pažeidžiama, įprastinis HMM modelis žlunga. HMM netinka modeliuoti būsenoms, kurios susideda iš smulkesnių, kitų dalių. Suporuotų HMM technika naudojama modeliuoti ir mokyti modelius, kai du procesai vienas kitą įtakoja įvairiais laipsniais, ir turi įvairias būsenų struktūras.

Markovo modeliai. Bendrais atvejais tinklai, tokie kaip – 3.3 pav. yra baigtinių būsenų. Kai tokie tinklai turi perėjimo iš vienos būsenos į kitą tikimybes jie vadinami Markovo. Jie yra grynai

priežastiniai – tikimybės priklauso nuo prieš tai buvusios būsenos. Markovo modelis vadinamas ergodiniu jei kiekviena iš būsenų turi ne nulinę tikimybę. Jei yra pasiekiami galutinė ar norima padėtis  $w_0$ , tai iš jos nebegali būti perėjimų į kitas būsenas.



3.3 pav. Markovo tinklas: trys paslėpti vienetai ir perėjimai tarp jų pavaizduoti juoda spalva, matomos būsenos ir jų galimas kitimas pavaizduoti raudona spalva.

Bajeso sprendimo teorija yra viena iš pagrindinių statistinių teorijų sprendžiant struktūros atpažinimo problemą. Šis metodas paremtas suderinimų matavimu tarp įvairių klasifikavimo sprendimų, panaudojant tikimybes ir išlaidas, kurios gaunamos tokiais sprendimais. Daroma prielaida, kad problema apibūdinama tikimybiname lygyje ir visos svarbios tikimybių reikšmės yra žinomos.

Optimalūs Bajeso klasifikatoriai. Sakykim, kad  $\pi_c$  priorinė tikimybė priklausyti klasei c. Jei žinome X pasiskirstymus, posterioriniu šansų santykis būti klasėje 2, atsižvelgiant į klasę 1, lygus [4]:

$$\frac{P(Y = -1 | X)}{P(Y = +1 | X)} = \frac{f_2(X)}{f_1(X)} \times \frac{\pi_2}{\pi_1} \quad (3.1)$$

čia P - tikimybė;

f(X) – pasiskirstymai;

$\pi_c$  - priorinė tikimybė;

$$Y = \begin{cases} +1, & \text{jei priklauso klasei 1} \\ -1, & \text{jei priklauso klasei 2} \end{cases}$$

$$X \sim \begin{cases} \text{Jei } X \text{ priklauso klasei 1 :} \\ \text{pasiskirstymas } f_1 \text{ su vidurkiu } \mu_1 \text{ ir nuokrypiu } E[(X - \mu_1)(X - \mu_1)' | Y = +1] = \sum_1 \\ \text{Jei } X \text{ priklauso klasei 2 :} \\ \text{pasiskirstymas } f_2 \text{ su vidurkiu } \mu_2 \text{ ir nuokrypiu } E[(X - \mu_2)(X - \mu_2)' | Y = -1] = \sum_2 \end{cases}$$

Jei mes žinome tarpusavyje priklausomų klasių tikimybių pasiskirstymo tankumų parametrinę formą, mes galime suvesti mokymo užduotį iš pasiskirstymo ieškojimo į parametrų radimą (kurį išreiškia vektorius  $\Theta_j$  kiekvienai  $w_j$  kategorijai),  $\Theta$  - modelis iš pilno rinkinio  $a_{ij}$  ir sekos  $w^T$ , o gautus rezultatus panaudoti klasifikacijai. Didžiausios tikimybės metodas siekia rasti parametro reikšmę, kuri geriausiai palaikoma mokymo duomenų, pavyzdžiui maksimizuoja tikimybę išėjime gauti iš tikrųjų stebėtus pavyzdžius (praktikoje paprastumui dažniausiai naudojamas logaritminis panašumas). Bajeso metodo įvertinimuose parametrais laikomos atsitiktinės reikšmės turinčios a priori (ankstesnę) tankumą; mokymo duomenys paverčia tai į posteriori tankumą. Rekursinis Bajeso metodas atnaujinama Bajeso parametrus inkrementuodamas, pav.: taip yra atrenkamas kiekvienas mokymo taškas. Bajeso sprendimo metodai yra labiau priimtini, maksimalaus panašumo metodai yra lengviau pritaikomi praktikoje ir didelioje mokymo sekoje duoda tokius pat tikslus klasifikatorius.

Pakankamą statistinę  $s$ , galima laikyti funkcija, kurios užtenka apibūdinti modelį  $\Theta$ . Kai mes žinome ganėtinai statistikos duomenų modeliams, mums tereikia apskaičiuoti reikšmes, reikalingas sukurti klasifikatoriui – nereikalingi kiti duomenys ar funkcijos. Laukimo – maksimizavimo metodas yra iteratyvi schema maksimizuojanti modelio parametrus, net kai trūksta dalies duomenų. Kiekviena iteracija susideda iš dviejų dalių: laukimo - E (angl.k. expectation), kuri reikalauja sumažinti trūkstamų duomenų vaidmenį turimam modeliui ir maksimizavimas - M (angl.k. maximization), kurioje parenkamos naujo modelio optimalios reikšmės. Apskritai paėmus Laukimo - Maksimizavimo metodas beveik reikalauja duomenų pagerinimo, o ne optimizavimo, kiekvienai iteracijai, kai apdorojami dideli modelių kiekiai. Bajeso taisyklių tinklas kūrėjui leidžia apspręsti funkcinės priklausomybes pagal sujungimų topologiją tarp modelio kintamųjų. Kai bet kuris kintamųjų posistemis yra suvaržomas žinimų reikšmių, pagal Bajeso išvadų skaičiavimą kiekvienam tinklo mazgui randama jo tikimybė. Parametrai apibūdinantys sąlygines tikimybes gali būti nustatomi eksperto. Paslėptieji Markovo modeliai susideda iš mazgų, kurie priklauso paslėptajam sluoksniui, sujungtais kaip į vieną vienetą saitais, kurie apibūdina sąlygines perėjimų tarp būsenų tikimybes. Kiekvienas paslėptas sluoksnis susijęs su tikimybių rinkiniu, išskiriantis tam tikras matomas būsenas. Paslėptieji Markovo modeliai gali būti labai naudingi modeliuojant sekas, ypač susijusias kontekstu, pavyzdžiui kalbos fonemas. Visos perėjimo tikimybės gali būti išmokytos iš pavyzdžių sekų, pagal „Priekini- atgalinį“ arba „Baum-Welch“ algoritmus. Klasifikacija susideda iš vienintelio modelio radimo, kuris panašiausiai sumodeliavo stebėjimo seką [10].

Tikrinių vektorių metodas. Šis metodas informatiniu požiūriu remiasi svarbiausios informacijos ištraukimu iš atvaizdo, tada ją reikia kaip galima greičiau užkoduoti ir palyginti kodą

su duomenų bazėje turimais pavyzdžiais. Paprasčiausias būdas informacijos ištraukimui iš atvaizdo, rasti pokyčių svyravimus vaizdų sekoje ir panaudoti šitą informaciją kodavimui ir palyginimui.

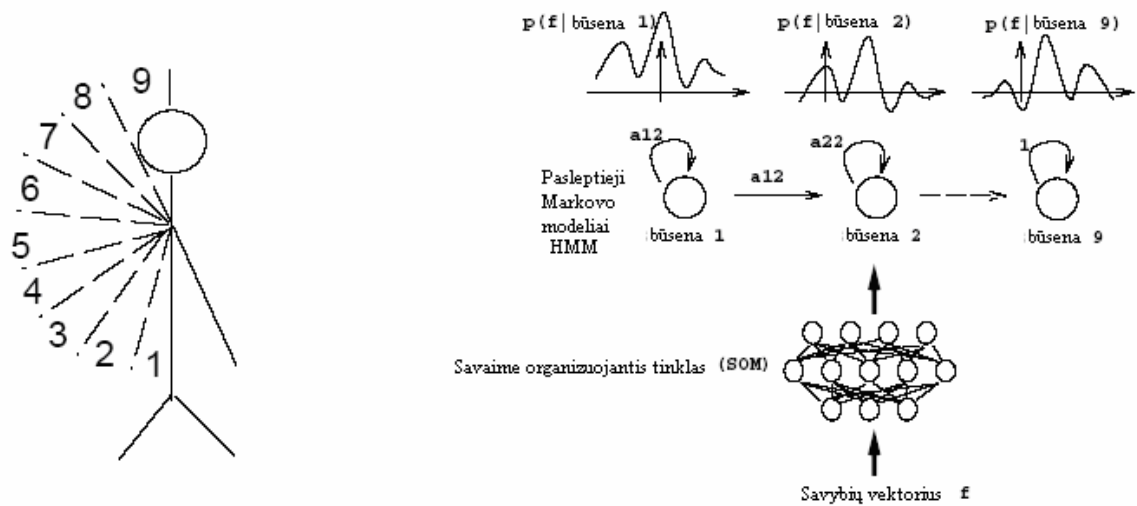
Matematinio požiūriu reikia rasti principinius komponentus atvaizdų pasiskirstyme, arba kovariacijos matricos tikrinius vektorius atvaizdų sekoje, su sąlyga, kad atvaizdas tai daugiadimensinis taškas (vektorius). Po to tikriniai vektoriai surūšiuojami. Šitie tikriniai vektoriai gali būti laikomi savybių rinkiniu, kartu jie charakterizuoja variacijas tarp atvaizdų. Kiekviena atvaizdo vieta labiau ar mažiau atitinka kiekvieną tikrinį vektorių. Kiekvienas atvaizde esantis objektas gali būti aprašomas tiesine tikrinių vektorių kombinacija. Kiekvienas objektas gali būti aproksimuotas naudojant tik „geriausią“ tikrinį vektorių – tokį, kurio tikrinės reikšmės didžiausios, ir kurios tuo pačiu apibūdina didžiausius objekto atvaizdų sekos pokyčius. Geriausi tikriniai objektai yra visų galimų atvaizdų  $M$  - dimenciniame poerdvyje – „objekto erdvėje“. Panašias problemas nagrinėjo mokslininkai Sirovich ir Kirby, nagrinėdami principinių komponentų efektyvumo analizę, atvaizduojant veidus. Pradėję nuo originalių veidų atvaizdų aibės, jie paskaičiavo geriausią koordinačių sistemą vaizdo suspaudimui, kur kiekviena koordinatė iš tikrųjų yra atvaizdas, kurį jie pavadino tikriniu atvaizdu. Jie įrodinėjo, kad iš principo, bet kuris veidų rinkinys gali būti apytiksliai atkurtas, panaudojant įrašytą į atmintį mažą svorių rinkinį kiekvienam veidui ir mažą standartinių atvaizdų (tikrinių atvaizdų) rinkinį. Svoriai, apibūdinantys kiekvieną veidą, randami projektuojant veido atvaizdą ant kiekvieno tikrinio atvaizdo [6].

Dinaminis modelių lyginimas. Pradiniai metodo duomenys - spalvotų atvaizdų seka. Išėjime gaunamas rezultatas – savybių vektorių seka su tolydžiais, ištisiniais parametrais. Tiksliau šis metodas naudoja Kohonen savybių žemėlapi - (SOM) įėjimo erdvės klasterizavimui, po to kiekviena kodų knyga iš diskretaus alfabeto verčiama į simbolį. Toks simbolis klasifikuojamas panaudojant diskretinį HMM modelį.

Hibridinis SOM (self organising map) / diskretinis HMM modelis. Kai turime įėjimo parametrų seką, sekantis žingsnis susijęs su savybių vektoriaus kvantavimu į simbolių seką. Daugiadimensinės erdvės savybių topologijai išsaugoti panaudojamas SOM metodas. Erdvė išskirstoma į dviejų matmenų savybių vektorius. Dėl nuoseklios gesto prigimties toks topologiją išsaugantis metodas gali būti patobulintas sudarant trajektorijas, kur užregistruojami geriausiai tinkantys SOM metodo neuronai. SOM klasteriai - tai neturintys pavadinimo mokymo savybių vektoriai, kurie yra vieni šalia kitų savybių erdvėje. Kodavimo knygos vektorius, kuris jautriausias tikrajam mokymo vektoriui, ir laiko erdvėje arčiausiai išsidėstę vektoriai yra suderinami, sukuriant gerai subalansuotą seką, sudarytą iš svorių reikšmių, jų įėjimo tankumo funkcijos atžvilgiu. Mokymo stadijoje svorių koregavimas atliekamas pagal Euklido atstumus tarp tikrojo keturiolikos matmenų įėjimo vektoriaus ir sujungiamo svorių vektoriaus, pagal pasirinktą mokymo greitį. Visų junginių apėmimui mokymo procesą reikia pradėti dideliu spinduliu. Šito reikia, kad gautume



rezultatai nebūtų įtakojami mažos mokymo vektorių sekų apimties. SOM metode sukurtas sąlyginis gestų žodynas yra dar suskirstomas į tarpinius gestus.



3.4 pav. Modelių sudarymas: iš kairės į dešinę: mojavimo dešine ranka gestas yra sudalinamas į devynis tarpinius gestus, kurių kiekvienas užima 20 laipsnių dvimatės paveikslėlio erdvės. SOM/HMM sistema parodo kaip kiekviena HMM būsena atsako už vieną tarpinį gestą.

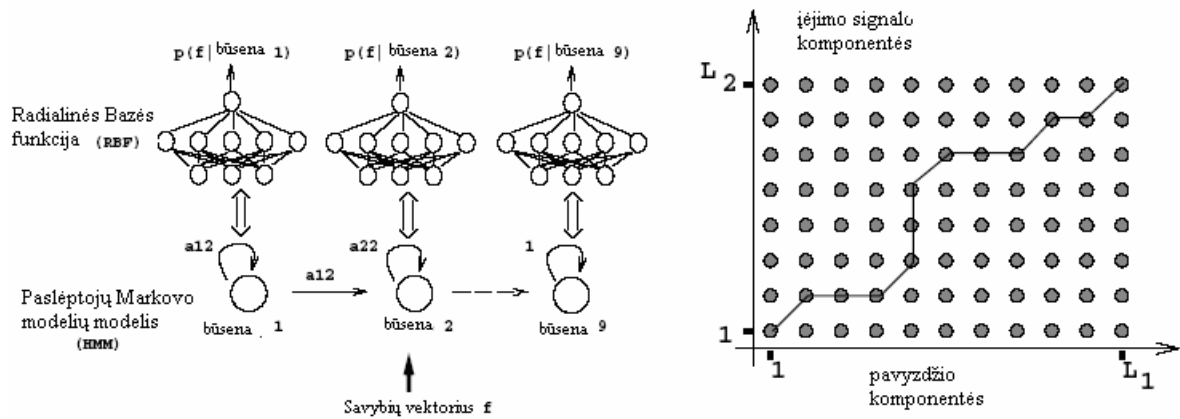
Klasių atskyrimui suteikiame pavadinimą kiekvienam iš SOM klasterių. Pavadinimai priskiriami pagal gaunamus junginius, naudojant tarpinių gestų žodyną – posistemę. Tai daroma naudojant sistemos įėjime žinomus mokymo rankos gestų pavyzdžius (3.4 pav.).

Kiekvienam gestui, kurį norime atpažinti, yra sukuriamas vienas, einantis iš kairės į dešinę HMM modelis, su tiek būsenų, kiek jame yra tarpinių gestų. Mokymo etape, norint gauti mokymo simbolių sekų modelį atitinkantį tam tikrą gestą, HMM parametrai optimizuojami. Atpažinimo stadijoje duota seka yra lyginama su kiekvienu HMM. Gestas, kuris geriausiai atitinka stebimų judesių seką yra pasirenkamas kaip atpažintas.

Diskretinio HMM modelio parametrų įvertinimui panaudojamas Baum-Welch perskaičiavimo metodas, kuris pagrįstas maksimalaus panašumo kriterijumi.

Statistinis klasifikavimas. Šiam metodui kaip ir dinaminiam modelių palyginimui, reikalinga spalvotų atvaizdų įėjimo seka, o išėjime gaunama – savybių vektorių seka su tolydžiais, ištiniais parametrais. Spindulinės bazės funkcijos (Radial Basis Function network -RBF) tinklas tiesiogiai naudojamas skaičiuoti galimoms HMM būsenų stebėjimų tikimybėms.

Hibridinis pastovus HMM/RBF. HMM būsenų tikimybėms nustatyti naudojami RBF - Spindulinės bazės funkcijos (Radial Basis Function network) tinklai (3.5 pav.).



3.5 pav. Modelių pavyzdžiai: iš kairės į dešinę: HMM/RBF hibridinė architektūra, DTW (dynamic time wrapping) pavyzdys, dviejų modelių, kurių ilgiai  $L_1$  ir  $L_2$ , laikų išlyginimas ir normalizavimas.

RBF tinkle išėjimo neuronų skaičius apsprendžiamas tarpinių gestų skaičiumi ir yra lygus HMM būsenų skaičiui. Bazinėmis funkcijomis pasirenkame Gauso tikimybės pasiskirstymo funkcijas, kurių kiekviena turi vidurkio vektorių ir skirtingas kovariacijos matricas. Duomenų rinkinys sudarytas iš įėjimo vektoriaus „ $v$ “, ir iš binarinių tikslų vektorių „ $t$ “, kurių  $j$ -tasis elementas pagal tarpinių gestų rinkinį prilygintas vienetui.

Gauso funkcijas verčiant į atitinkamus svorius reikia įvertinti poslinkių parametrus ir normalizavimo faktorius, taip išėjime gaunama tam tikra tikimybė. Pirmame išėjimo mazge:

$$y_l(\mathbf{x}) = \frac{\sum_{j=0}^{N_G} w_{jl} \exp \{t(\mathbf{x} - \mu_j) \Sigma_j^{-1} (\mathbf{x} - \mu_j)\}}{\sum_{u=1}^{N_O} \sum_{j=0}^{N_G} w_{ju} \exp \{t(\mathbf{x} - \mu_j) \Sigma_j^{-1} (\mathbf{x} - \mu_j)\}} \quad (3.2)$$

- čia  $y$  - išėjimo funkcija;
- $\Sigma$  - kovariacijos matrica;
- $\omega$  - svoriai;
- $x$  - įėjimas;
- $\mu$  - vidurkio vektorius;

Postūmis su papildoma bazės funkcija, kurios aktyvavimas nustatomas vienetui, yra įtraukiamas į nematomą lygmenį. Tikslėsniam mokymo duomenų pasiskirstymo nustatymui, bazės funkcijai tinkantys parametrai yra apskaičiuojami naudojant iteracijų  $k$ -reikšmių klasterizavimo algoritmą. Klasterių skaičius atitinka bazės funkcijų skaičių. Po vidurkio vektoriaus „ $\mu$ “ ir kovariacijos matricos „ $\Sigma$ “ parametrų nustatymo, kiekvienai Gauso funkcijai jie išlaikomi pastovūs, o antrojo lygmens svoriai yra nustatomi pagal gradientų - nuolydžių kitimų metodą.

RBF tinklai ir atitinkantys HMM nėra mokomi kartu: pirmiausiai apmokomi RBF tinklai po to HMM, vėliau taikome Baum-Welch perskaičiavimo algoritmą HMM parametrų nustatymui.

Dinaminis modelių lyginimą ir statistinį klasifikavimą galime prilyginti statistiniam uždaviniui, kuriame gestų judesiai atitinka klases  $C_j | j = 1..C$  ir parametrai charakterizuojantys gestą  $X_N = \{x_1 \dots x_N\}$ , toks uždavinys statistiniu požiūriu - maksimalaus „posteriori“ (MAP) kriterijaus radimas. Pagal Bajeso taisyklę, toks uždavinys gali būti suformuluotas, kaip klasių, kuriose „posteriori“  $P(C_j | X_N)$  tikimybė maksimali, paieška. Tokiam uždaviniui spręsti yra sukuriama klasė tikimybių modelių, kurie aprašo vartotojo judėjimą ir pakitimus: paslėptieji Markovo modeliai.

Neuronų tinklų metodas. Naudojama dinaminio programavimo technikos. Įėjimo savybių vektorių seka yra lyginama su prieš tai apibrėžtais pavyzdžiais, naudojant „Dynamic time wrapping“ algoritmą. Ši sistema tai pavyzdžių lyginimo sistema, tačiau naudojami ir dinaminio programavimo modeliai, pavyzdžiui, jeigu reikia rasti minimalius atstumus tarp įėjimo sekų ir prieš tai apibrėžtų sekų.

Šablonų lyginimas naudojant DTW metodą. Laikant gestą savybių vektoriumi, galime sugalvoti atitinkančią šablonų seką kiekvienam norimam atpažinti judesiui, po to rasti minimalų atstumą tarp naujos įėjimo sekos ir turimų šabloninių.

Klasifikavime įėjimo signalas priskiriamas prie tam tikros klasės, pagal atstumų funkciją. Naudojant tokį metodą iškyla keletas problemų: kiek reikės klasių ir kaip jas sugalvoti, pasirinkti; ar galime skaičiuoti atstumus tarp dviejų klasių, kurių ilgiai nelygūs.

Sprendžiant pirmą klausimą, paprasčiausiai apskaičiuojame reikšmių sekos ilgį, ir naudojame vidurkių sekos vektorių. Nustatome vieną šabloną kiekvienai judesių sekai.

Antrajam klausimui spręsti galima panaudoti DTW algoritmą. Šis algoritmas leidžia sulyginti du įvairių ilgių signalus (3.5 pav.) [14].

## 4. PRAKTINIS TYRIMAS

### 4.1. Pradinų sąlygų nustatymas

Gestas - kūno, ypač rankų, judesys jausmams, kalbai paryškinti (ar nebyliams susižinoti), mostas („Dabartinės lietuvių kalbos žodynas“ Lietuvių kalbos institutas, 2000). Daugiausiai informacijos neša rankos gestai, būtent juos ir analizuosime.

Programa kuriama su Matlab programų paketu. Ji patogi modeliuoti ir spręsti tokio pobūdžio problemas. Visus skaičiavimus ir veiksmus galima atlikti naudojant vien šią programą. Galimi įvairių modulių sudarymai ir vėlesni jų apjungimai. Patogi, valdoma sąsaja su vartotoju, funkcijos pritaikytos duomenų analizei.

Nusprendėme dirbti su statiniu vaizdu.

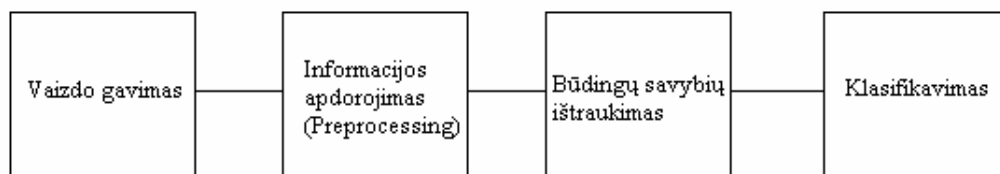
Buvo pasirinkti 10 rankos gestų.

Filmavimas vykdomas skaitmenine vaizdo kamera. Funkcijų pagalba analizuojami atskiri kadrai.

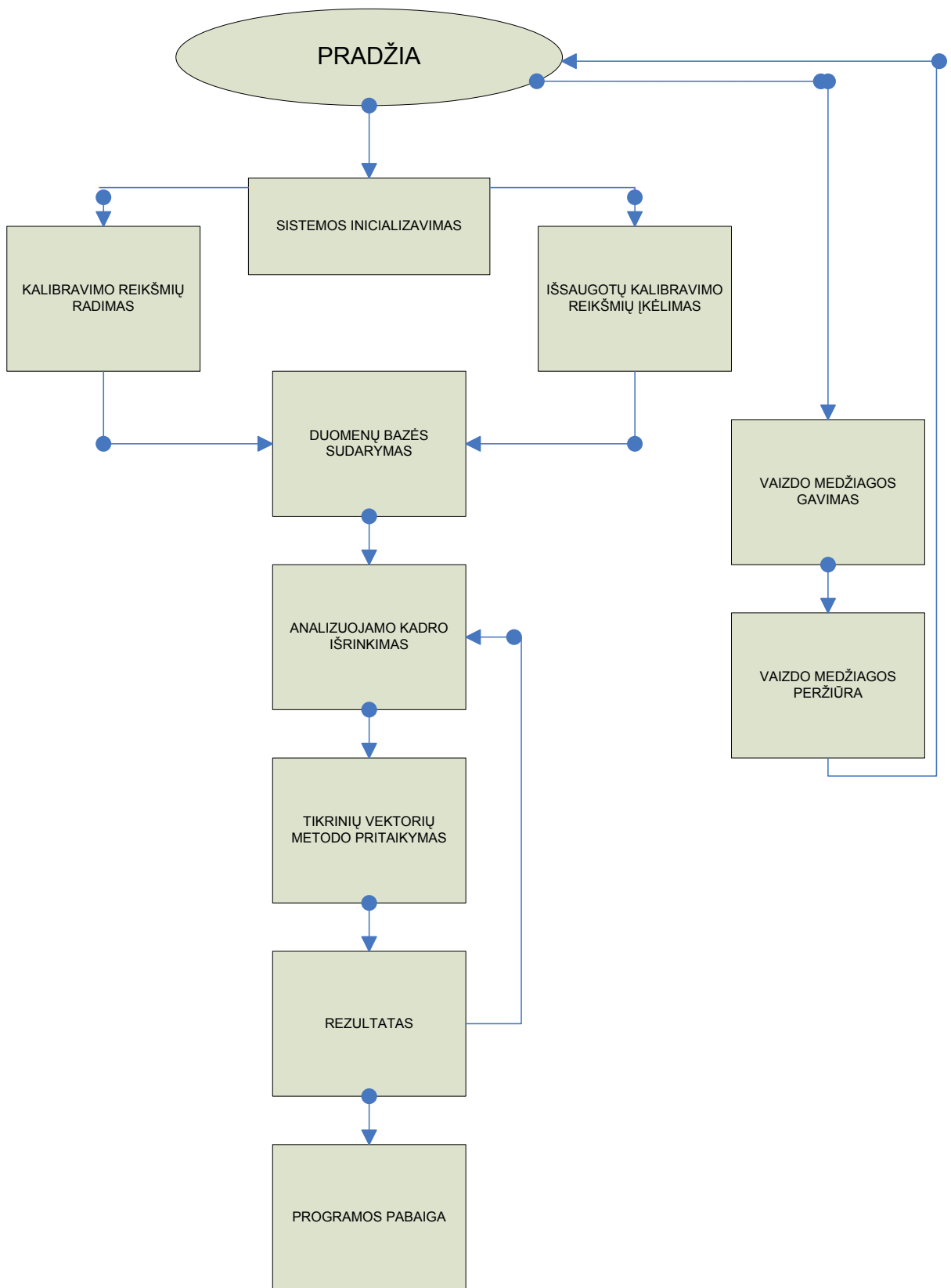
### 4.2. Tyrimo eiga

Pats bendriausias algoritmas, būdingas tokioms atpažinimo sistemoms – 4.1 pav., susideda iš atvaizdo gavimo, informacijos apdorojimo, būdingų savybių ištraukimo ir klasifikavimo.

1. Vaizdo gavimas arba pradinių parametrų apie rankos gestą ištraukimas.
2. Gautos informacijos apdorojimas - perteklinės informacijos, triukšmo pašalinimas iš atvaizdo neprarandant reikalingų duomenų.
3. Būdingų savybių išrinkimas – duomenų reikalingų palyginimui, klasifikavimui surinkimas.
4. Klasifikavimas - pagal išrinktus duomenis, savybes priskyrimas klasei pasirinktu metodu.



4.1 pav. Bendras algoritmas.

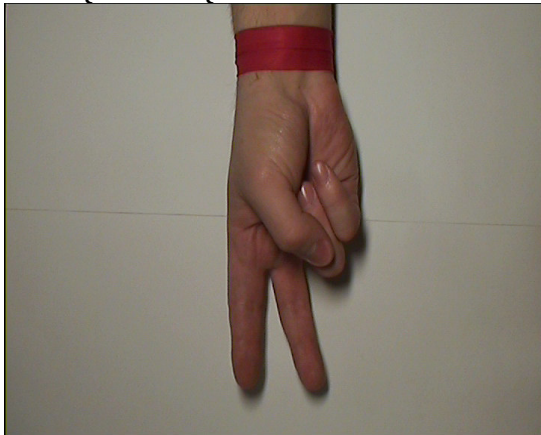


4.2 pav. Naudojamas programos algoritmas.

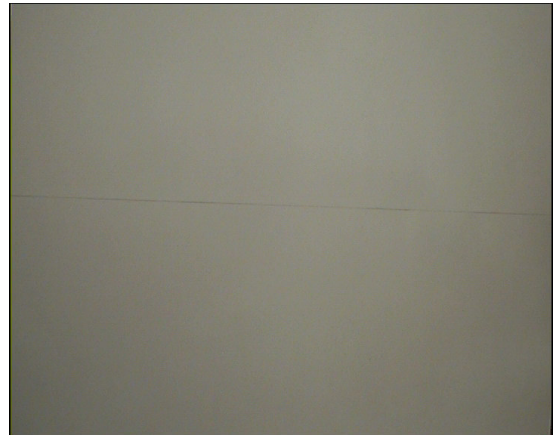
Vaizdui gauti pasirinkta skaitmeninė kamera. Ja filmuojame, programoje nustatomos gestų trukmės, atvaizdus. Panaudoję „Matlab“ aplinkai sukurtą video įrašymo programą, gauname video medžiagą su kuria dirbame.

Iš gautos medžiagos rankiniu būdu pasirenkame rankos gesto kadra. Prieš pradėdant klasifikuoti rankos gestų reikšmes reikia sistemą sukalibruoti arba įkelti ankstesnio eksperimento kalibravimo metu išsaugotas reikšmes. Kalibravimas vykdomas atitinkamu Matlab programai sukurtu moduliui.

**Pradinių reikšmių kalibravimas:**

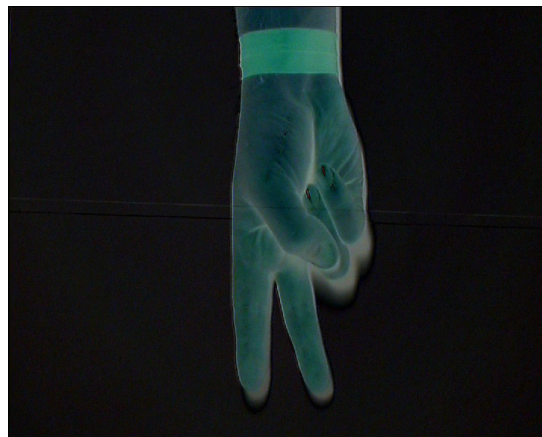


4.3 pav. Kadras su rankos gestu



4.4 pav. Pagrindo kadras be rankos

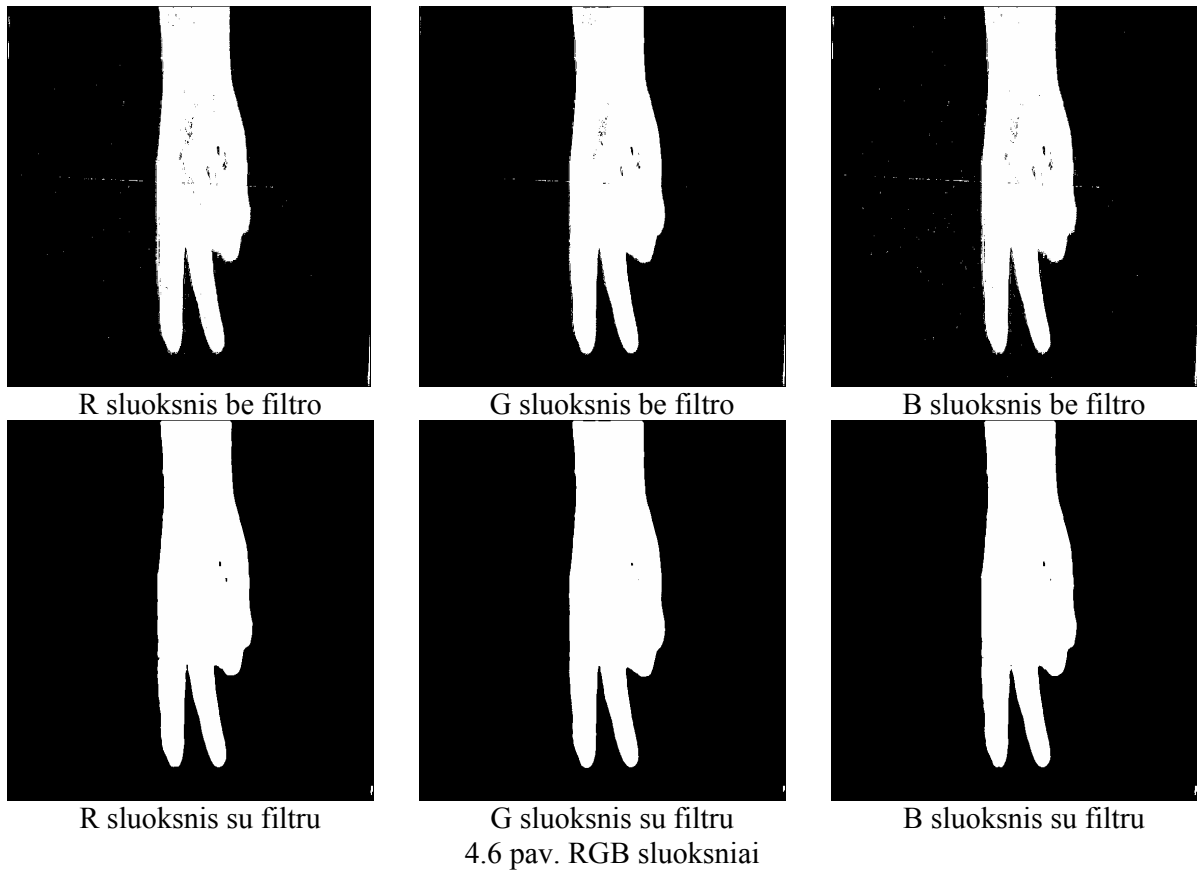
Naudojame pagrindo atskyrimo techniką, tai metodas, kai reikalingas objektas išskiriamas iš kadro atimant iš jo kadra be objekto. Tam išrenkame du kadrus ir panaudodami matematinės funkcijas randame kadru skirtumą.



4.5 pav. Skirtumas tarp pagrindo ir rankos su gestu atvaizdų.

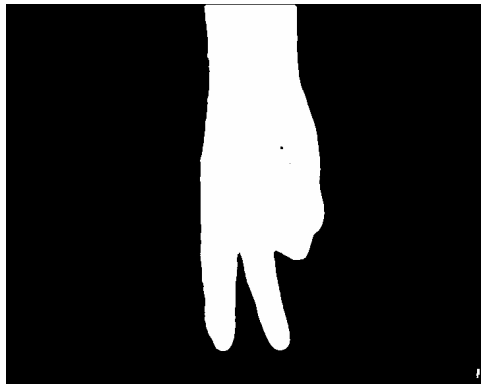
Panaudojame RGB spalvų sistemos savybes, ieškome rankos gesto padėties kiekviename iš RGB sluoksnių, tai yra iš pradžių R, po to G, po to B. Tam, kad atvaizde liktu tik rankos gestas be triukšmų, kiekvienas sluoksnis verčiamas į juodai baltą, su tam tikra slenkstine riba. Ši riba ir

nustato kiek triukšmo liks atvaizde. Slenkstinės ribos skaičiavimas vykdomas panaudojant automatinį Otsu metodą (Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.). Jei matoma, kad atvaizde liko daug triukšmo, ar priešingas atvejis – prarasta dalis naudingos rankos gesto informacijos, slenkstinė riba sumažinama arba padidinama rankiniu būdu. Rezultatai matomi 4.5 pav. be filtro. Smulkių dalelių pašalinimui naudojamas filtras.

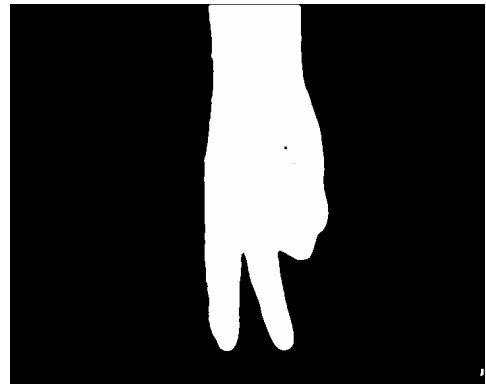


4.6 pav. RGB sluoksniai

Eksperimentais nustatyta, kad geriausiai smulkias daleles pašalina median filtras. Jis filtruoja atvaizdą artimiausių kaimynų principu. Panaudojus šį, o ne kitus bandytus filtras gaunami geriausi rezultatai 4.5 pav. - su filtru.



Sudėtos R+G+B kaukės

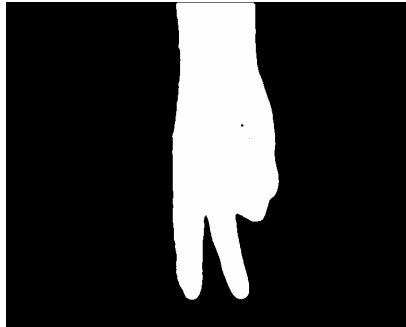


Sudėtos R+G+B kaukės po morfologinių operacijų

4.7 pav. Sudėtos RGB kaukės

Atlikus šias operacijas gaunamos rankos gesto padėties kaukės kiekviename iš RGB sluoksnių. Galutiniam rezultatui sudedame visas gautas R, G, B kaukes į vieną ir panaudojame „`bw= bwmorph(bw,'open')`“ morfologinę funkciją (Performs morphological opening (erosion followed by dilation)), tuo dar labiau išvalydami atvaizdą.

Matlab funkcija užpildome, dėl panaudotų operacijų ir rankos netolygaus šviesos atspindėjimo, atsiradusias mažas skylutes, plotelius, kurie priklauso rankai 4.7 pav.



4.8 pav. Atvaizdas su užpildytomis mažomis skylutėmis.

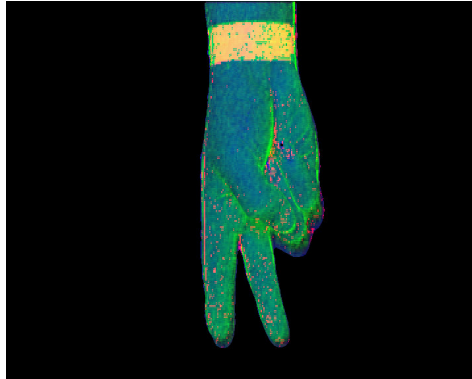
Tada gautą galutinę kaukę pritaikome originaliam kadrai su ranka.



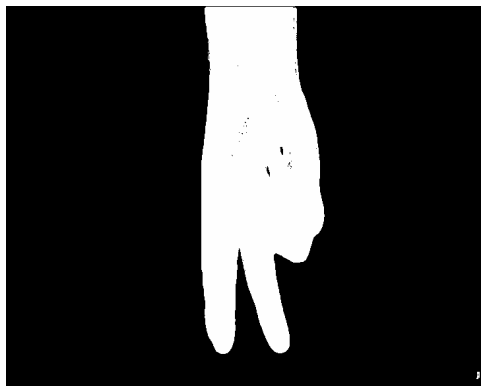
4.9 pav. Originalus atvaizdas su pritaikyta kauke.



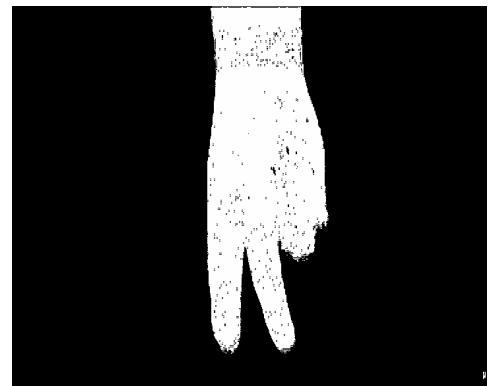
Iš gauto atvaizdo specialios funkcijos pagalba išrenkame rankai priklausančius plotelio pikselius. Randame minimalias ir maksimalias rankos odos reikšmes RGB ir HSV spalvų sistemose.



4.10 pav. Atvaizdas paverstas į HSV spalvų sistemą  
Pagal gautas reikšmes sudarome kaukes originalui 4.10.



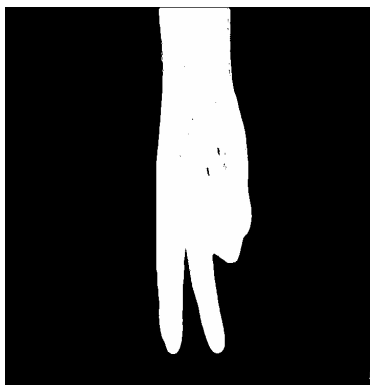
RGB kaukė



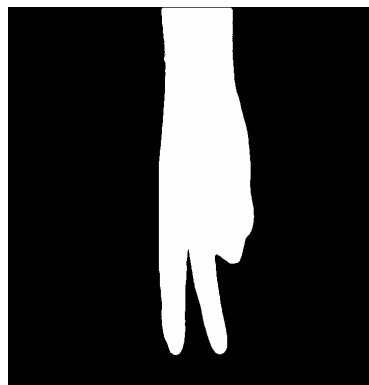
HSV kaukė

4.11 pav. Rankos kaukės

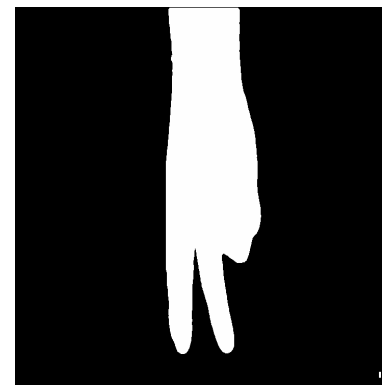
Gautos RGB ir HSV kaukes sudedamos, užpildomos atsiradusios skylutės ir atliekamos morfologinės operacijos.



RGB+HSV



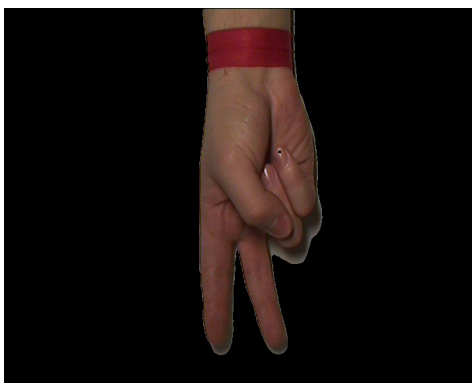
RGB+HSV užpildytos skylės



Po morfologinių operacijų..

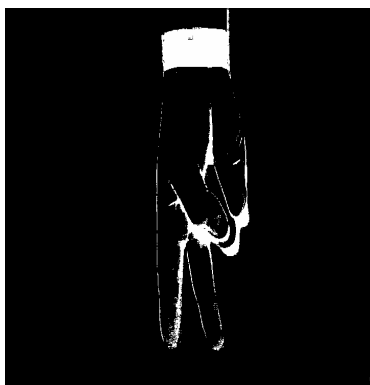
Pritaikome gautą kaukę originalui:

4.12 pav. Veiksmai su kaukėmis

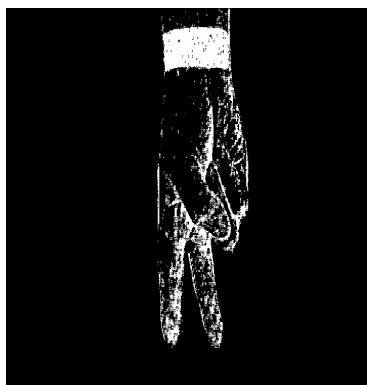


4.13 pav. Galutinis rezultatas

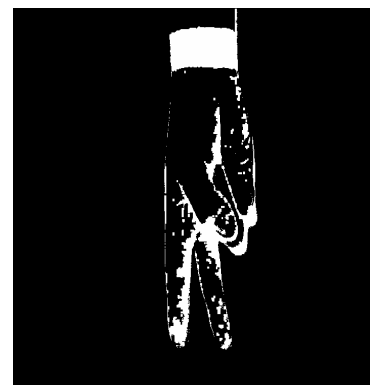
Palyginus su RGB sluoksnių filtravimo metodu gautu rankos atvaizdu, matome, kad ryškesni rankos kontūrai, mažiau šešėlių. Analogiškas operacijas atliekame rankos raiščio išskyrimui, jis reikalingas atskirti rankos delną su pirštais nuo likusios rankos 4.13pav., 4.14 pav.



RGB kaukė

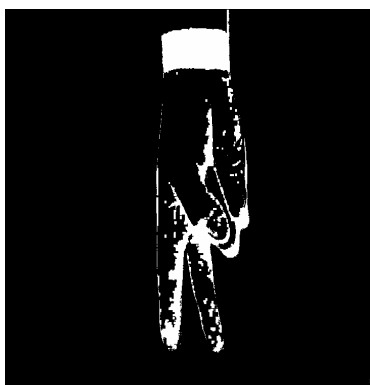


HSV kaukė

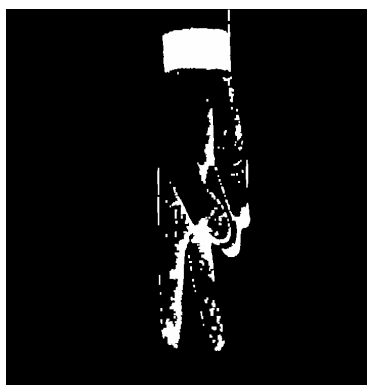


RGB+HSV kaukė

4.14 pav. Raiščio kaukės



Kaukė užpildytomis skylėmis



Kaukė po morfologiniu operacijų



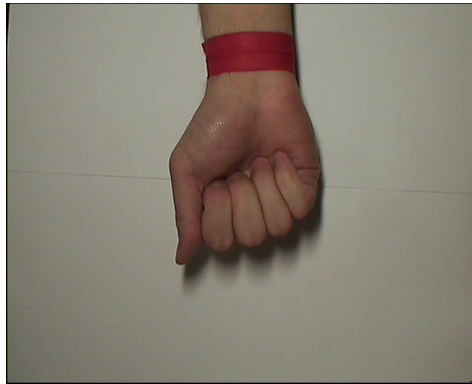
Kaukė pritaikyta originaliam atvaizdui

4.15 pav. Veiksmai su raiščio kaukėmis

Kaip matome ranka turi daug raudonos spalvos pikselių. Jie bus pašalinti maksimalaus raiščio regiono paieškos būdu.

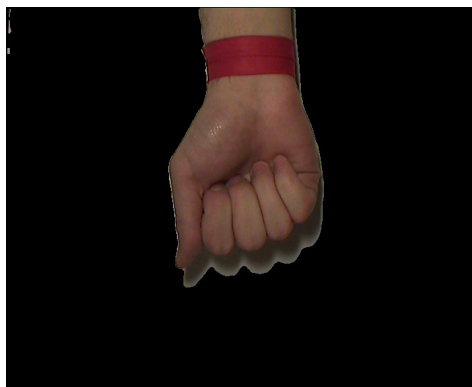
Kalibravimo operacijos duoda rankos RGB sluoksnių slenkstines vertes, rankos minimalias ir maksimalias RGB, HSV reikšmes, taip pat turime raiščio reikšmes, kurios bus reikalingos atskirti rankos delną ir pirštus nuo likusios rankos dalies. Kalibravimo reikšmes galime išsaugoti ir naudoti, kol nepasikeičia pradinės atvaizdo gavimo sąlygos.

Klasifikavimui pritaikome kalibravimo reikšmes pasirinktam rankos gesto kadrai 4.15 pav.



4.16 pav. Išsirinktas atpažinimui kadras

Pagal ankstesnes reikšmes gauname tokį atvaizdą 4.16 pav. 4.17 pav., 4.18 pav.



4.17 pav. Pagal kalibravimo reikšmes išskirta ranka su raiščiu

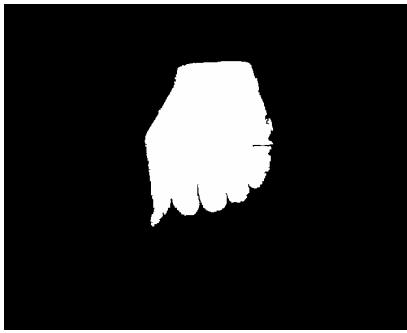


Rastos raiščio galimos vietos



Maksimalaus dydžio raiščio regionas

4.18 pav. Raiščio išskyrimas



Rankos kaukė, kai atimta raiščio kaukė.



Iš viso atvaizdo, pagal ribojama stačiakampį, išskirta ranka.



Iš RGB spalvų rankos gestas paverčiamas į pilką spalvą, to reikia tolesniems skaičiavimams.

4.19 pav. Atvaizdo paruošimas

Iš RGB spalvų sistemos atvaizdas paverčiamas į atitinkamo intensyvumo pilkos spalvos atvaizdą. Paveikslėlio dydis sumažinamas iki 180x180 pikselių, tai padaroma su visais apdorotais atvaizdais. Tuo pasiekiamas spartesnis duomenų apdorojimo greitis ir išnaudojama mažiau kompiuterio resursų. Tokiu būdu apdorotas rankos gesto atvaizdas talpinamas į duomenų bazę priskiriant atitinkamai klasei pagal reikšmę. Iš apdirbtų atvaizdų sudaroma rankos gestų duomenų bazė. Apdorotas atvaizdas pagal reikšmę talpinamas į atskiras klases. Atpažinimui išrinktas kadras apdirbamas tokiu pat būdu, kaip ir duomenų bazės sudaryme.



A



B



C



Č



E



Ė



F



I



L



V

4.20 pav. Naudojamos gestų reikšmės.

Klasifikavimui naudojamas tikrinių vektorių metodas.

Atpažinimas tikrinių vektorių metodu. Kadangi objektų atvaizdai gali būti apytiksliai atkurti iš mažo rinkinio svorių sumų, tai veiksmingas būdas atpažinti objektus būtų sukurti sistemą, kuri laiko sekoje surinktų pagrindines savybes iš objektų atvaizdų rinkinių ir juos atpažinti lyginant savybių svorius. Tokiam metodui reikia tokių veiksmų:

1. Surinkti pradinį objektų atvaizdų rinkinį (mokymo rinkinį);
2. Rasti mokymo rinkinio tikrinius vektorius, išsaugant vektorius M turinčius aukščiausias reikšmes. Šitie M vektoriai apibūdina objekto erdvę.
3. Rasti atitinkamą pasiskirstymą M - dimencinėje svorių erdvėje, kiekvienai objekto klasei, projektuojant objekto atvaizdus į „objektų erdvę“.

Inicializavus sistemą, laikomasi tokių žingsnių:

1. Suskaičiuoti įėjimo objekto atvaizdo svorių rinkinį ir M tikrinius objektus projektuojant įėjimo atvaizdą ant kiekvieno iš tikrinių vektorių.
2. Nuspręsti ar įėjimas reikalingas objektas, tai matoma matematiškai patikrinus ar pakankamai arti atvaizdas yra „objekto erdvės“.
3. Jei tai žinomas objektas, charakterizuoti svorių struktūrą, kaip žinomo objekto ar nežinomo.

Sakykim rankos atvaizdas  $I(x,g)$ , yra dviejų  $N \times N$  matmenų ir jį sudaro pilkos spalvos pikselių reikšmės. Atvaizdas, taip pat gali būti laikomas  $N^2$  matmenų vektoriumi. Taigi, jei atvaizdas yra  $180 \times 180$  dydžio, jis tampa 32400 matmenų vektoriumi. Atvaizdų aibė tokiu būdu tampa labai didele erdve.

Rankų atvaizdai nebus bet kaip išdėstyti šioje erdvėje ir gali būti aprašyti palyginus mažų matmenų poerdviu. Esminė principinių komponentų analizės mintis (kitai Korhunen – Loeve skleidinį) – rasti vektorius, kurie geriausiai apibūdina rankų atvaizdų pasiskirstymą visoje atvaizdų erdvėje. Šitie vektoriai aprašo rankos atvaizdų poerdvį, kurį vadinsime „rankų erdve“. Kiekvienas vektorius yra  $N^2$  ilgio, jis apibūdina  $N \times N$  atvaizdą ir yra originalaus rankos atvaizdo tiesinė kombinacija. Kadangi šitie vektoriai atitinkantys rankos atvaizdus yra kovariacijos matricos tikriniai vektoriai mes vadinsime juos „tikrinėmis rankomis“.

Sakykim, kad rankos atvaizdų mokymo seka yra :  $G_1, G_2, G_3, \dots, G_M$ . Tai yra atvaizdai sudėti į duomenų bazę. Vidutinė rinkinio ranka apibūdinama:

$$\Psi = \frac{1}{M} \sum_{n=1}^M G_n; \quad (4.1)$$

čia  $\Psi$  - vidutinės objekto atvaizdo reikšmės;

M- atvaizdų skaičius;

G - mokymo atvaizdo reikšmės.

Kiekvienas objektas skiriasi nuo vidutinio vektoriaus tokia reikšme:

$$\Phi_i = G_i - \Psi; \quad (4.2)$$

čia  $\Phi$  - skirtumas, tarp objekto reikšmių ir vidutinių objekto reikšmių;

$\Psi$  - vidutinės objekto reikšmės;

Turint vektorių rinkinį galima atlikti principinę elementų analizę. Reikia rasti M orto normalių vektorių  $u_k$ , kurie aprašo duomenų pasiskirstymą. K – tasis vektorius yra pasirenkamas toks:

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (u_k^T \Phi_n)^2 \quad (4.3)$$

čia  $\lambda_k$  - tikrinės reikšmės;

M – atvaizdų skaičius;

$u_k$  - tikrinis vektorius;

$\Phi$  - skirtumas, tarp objekto reikšmių ir vidutinių objekto reikšmių.

Jis turi būti maksimalus, atsižvelgiant tokią sąlygą:

$$u_l^T u_k = \delta_{lk} \begin{cases} 1, & \text{jei } l = k \\ 0 \end{cases} \quad (4.4)$$

čia  $u_k$  - tikrinis vektorius;

$u_l$  - atvaizdo tikrinis vektorius;

Vektoriai  $u_k$  ir skaliariai  $\lambda_k$  yra tikriniai vektoriai ir tikrinės reikšmės, atitinkamai kovariacijos matrica:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad (4.5)$$

čia C – kovariacijos matrica;

$\Phi$  - skirtumas, tarp objekto reikšmių ir vidutinių objekto reikšmių;

M – atvaizdų skaičius;

Matrica  $A = [\Phi_1 \Phi_2 \dots \Phi_M]$ . Matrica C susideda  $N^2 \times N^2$  reikšmių ir rasti  $N^2$  tikrinius vektorius ir tikrines reikšmes įprasto dydžio atvaizdams yra sunkiai išsprendžiamas uždavinys.

Jei duomenų taškų skaičius atvaizdo erdvėje yra mažesnis negu erdvės matmuo ( $M < N^2$ ), tada bus tik  $M - 1$ , o ne  $N^2$  reikšmingų tikrinių vektorių (likusiems tikriniams vektoriams bus suteiktos nulinės tikrinės reikšmės). Laimei mes galime išspręsti  $N^2$  – matmenų vektorius, pirma radę  $M \times M$  matricos tikrinius vektorius, pavyzdžiui išsprendę  $16 \times 16$  matricą, negu sprendę  $16384 \times 16384$  matricą – ir paėmę atitinkamą rankos atvaizdų  $\Phi_i$  kombinaciją. Laikome matricos  $A^T A$  tikrinius vektorius  $v_i$ :

$$A^T A v_i = \mu_i v_i \quad (4.6)$$

čia  $A$  – matrica;  
 $v_i$  - tikrinis vektorius;  
 $\mu_i$  - koeficientas.

dauginame abi puses iš  $A$ :

$$A A^T v_i = \mu_i A v_i \quad (4.7)$$

Iš paskutinės formulės matome, kad  $A v_i$  yra lygūs  $C = A A^T$  tikriniams vektoriams. Sudarome matricą  $M \times M$  lygią  $L = A^T A$ , kur  $L_{mn} = \Phi_m^T \Phi_n$  ir randame matricos  $L$  tikrinius vektorius  $v_i$ . Šitie vektoriai lemia  $M$  mokymo rinkinio tiesines kombinacijas, kurios formuoja tikrinius vektorius  $u_i$ . Tokiu būdu yra sumažinamas skaičiavimų skaičius, nuo pikselių atvaizde ( $N^2$ ) eilės iki atvaizdų mokymo sekoje ( $M$ ) eilės. Taigi realybėje kai mokymo atvaizdų rinkinys bus palyginus mažas ( $M \ll N^2$ ), skaičiavimų kiekis apčiuopiamai sumažėja. Šitos susietos tikrinės reikšmės leidžia mums išrūšiuoti tikrinius vektorius pagal jų naudingumą tarp atvaizdų charakterizuojant pokyčius.

Naujas rankos atvaizdas transformuojamas į tikrinius komponentus (projektuojamas į „rankos erdvę“) paprasta operacija:

$$w_k = u_k^T (G - \Psi) \quad k=1, \dots, M \quad (4.8)$$

čia  $w_k$  - tikrinis komponentas;  
 $M$  – pasirinktas skaičius, kurio užtenka apibūdinti rankos erdvei.

Ši formulė aprašo rinkinio taškas po taško atvaizdų sudauginimus ir sumavimus. Svariai iš vektoriaus  $\Omega^T$  apibūdina kiekvienos tikrinės rankos indėlių į bazinį rankos atvaizdų rinkinį:

$$\Omega^T = [w_1, w_2, \dots, w_M] \quad (4.9)$$

čia  $\Omega$  - vektorius;

$w_k$  - tikrinis komponentas.

Šis vektorius vėliau gali būti panaudojamas standartiniame struktūros atpažinimo algoritme, rasti kuriai iš prieš tai apibudintų klasių priklauso nagrinėjamas atvaizdas. Vienas iš paprasčiausių metodų rasti, kuri klasė geriausiai apibūdina įėjimo atvaizdą, tai rasti rankos klasę  $k$ , kuri minimizuoja Euklido atstumą:

$$\varepsilon_k^2 = \|(\Omega - \Omega_k)\|^2 \quad (4.10)$$

čia  $\varepsilon$  - Euklido atstumas;

$\Omega$  - vektorius.

$\Omega_k$  vektorius apibūdinantis  $k$ -tąją rankos klasę. Rankos klasės  $\Omega_i$  randamos vidurkinant tikrinės rankos rezultatus panaudojant mažą rankos atvaizdų kiekį kiekvienam objektui. Ranka priskiriama klasei  $k$ , kai minimalus  $\varepsilon_k$  yra žemiau nustatytos ribos  $\Theta_\varepsilon$ . Kitu atveju ranka klasifikuojama, kaip nežinoma.

Kadangi svorių vektorius kūrimas yra tas pats kaip originalaus rankos atvaizdo projektavimas į žemo lygio matmenų rankos erdvę, dauguma atvaizdų projektuosis ant duoto struktūros vektorius. Tai ne problema sistemai, tačiau, kadangi atstumas  $\varepsilon$  tarp atvaizdo ir rankos erdvės yra paprasčiausiai atstumas kvadratu tarp reikšmių – sukoreguoto įėjimo atvaizdo

$\Phi = G - \Psi$  ir  $\Phi_f = \sum_{i=1}^{M'} w_i u_i$  projekcijos į rankos erdvę:  $\varepsilon^2 = \|\Phi - \Phi_f\|^2$ . Taigi yra keturi galimi

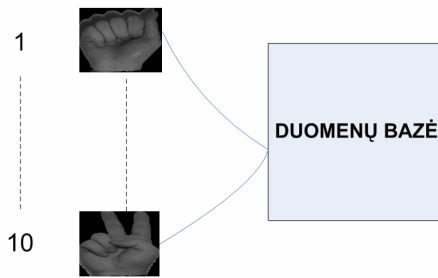
atvejai įėjimo atvaizdui ir jo struktūros vektoriui:

1. šalia rankos erdvės ir šalia rankos klasės, tai reiškia, kad objektas žinomas ir atpažintas;
2. šalia rankos erdvės, bet ne šalia žinomos rankos klasės, tai reiškia, kad objektas nėra žinomas, bet tai ranka;
3. toli nuo rankos erdvės ir šalia rankos klasės reiškia, kad objektas ne ranka;
4. toli nuo rankos erdvės ir toli nuo žinomų klasių, objektas ne ranka.

Atpažinimo tikrinių vektorių metodu pritaikymas:

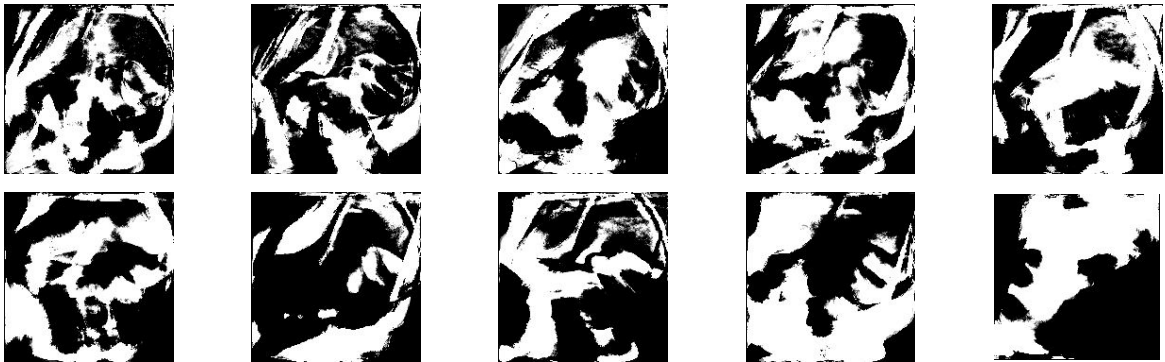
1. Panaudoję anksčiau aprašytus metodus randame rankos padėtį ir sukuriame pradinį rankų atvaizdų rinkinį (mokymo rinkinį). Pagal reikšmę atvaizdai sugrupuojami į dešimt klasių.





4.21 pav. Duomenų bazė.

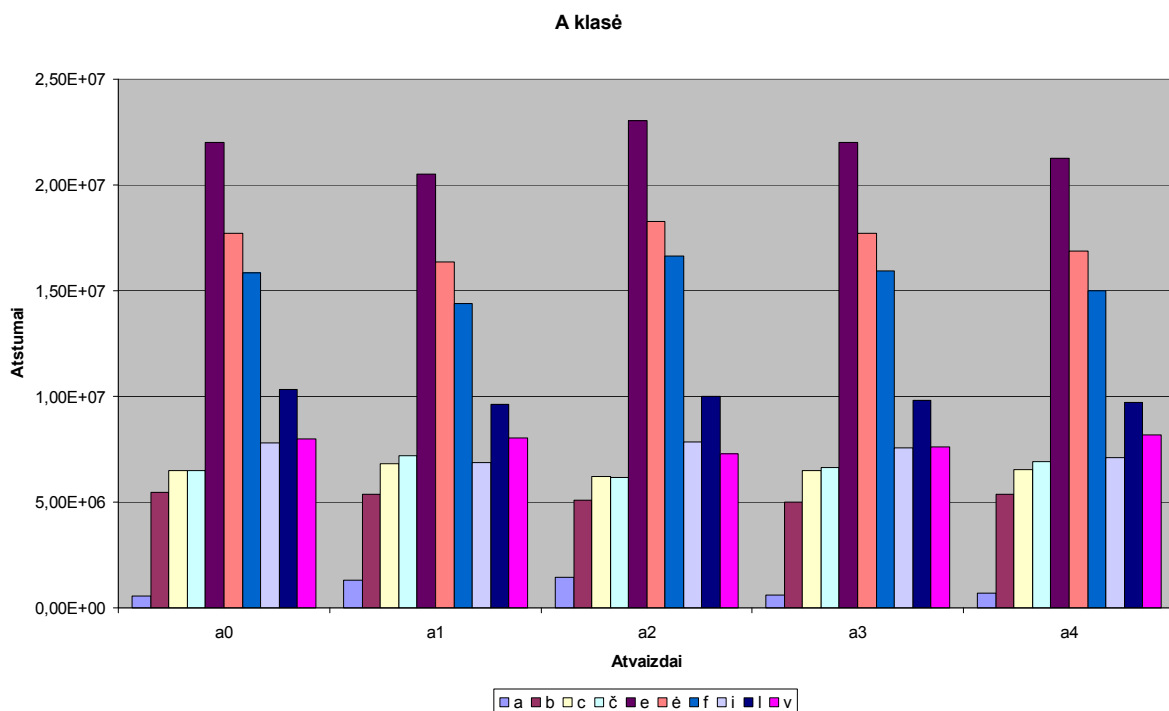
2. Randame mokymo rinkinio tikrinius vektorius, ir išsaugome tikrinius vektorius  $M'$ , kurie turi aukščiausias reikšmes. Pasirinkti  $M'$  vektoriai geriausiai apibūdina rankos erdvę.



4.22 pav. 10 geriausiai rankos erdvę apibūdinančių tikrinių vektorių -  $u_k$ .

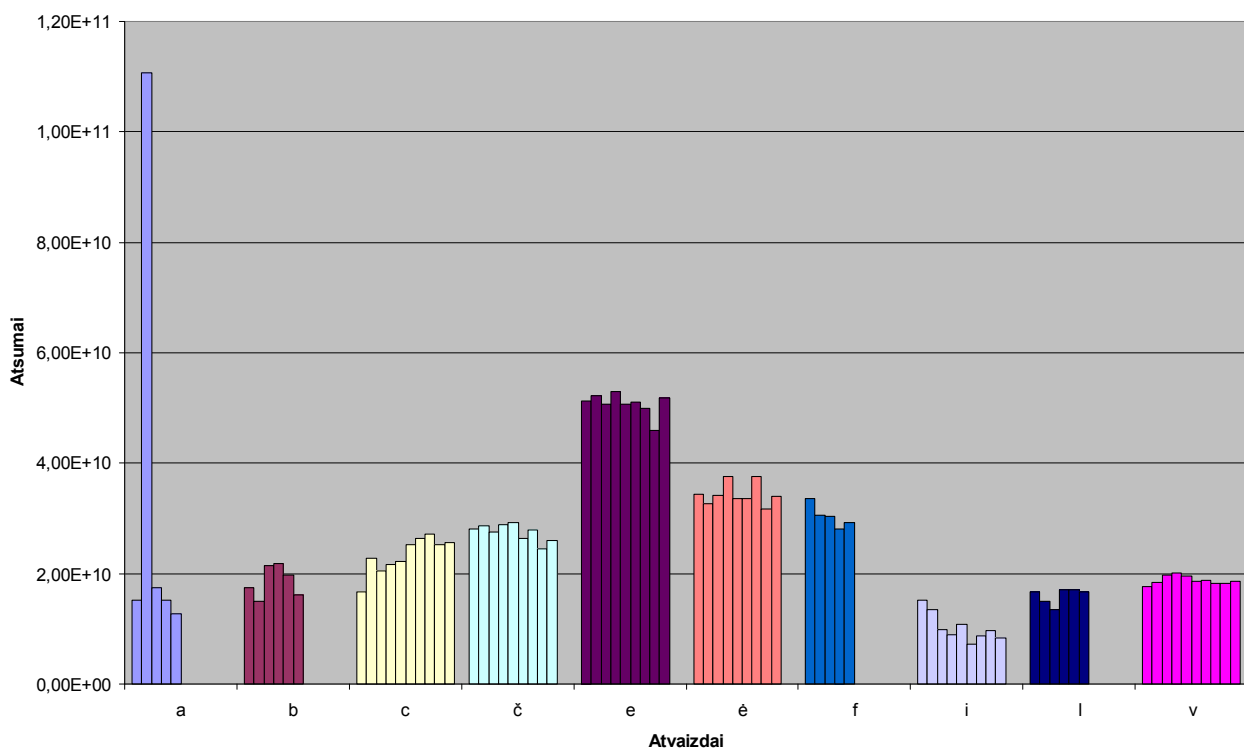
3. Kiekvienai rankos gesto reikšmės klasei randame atitinkamą pasiskirstymą  $M'$  - dimencinėje svorių erdvėje.
4. Suskaičiuojame įėjimo rankos atvaizdo svorių rinkinį, tai padaroma projektuojant įėjimo atvaizdą ant kiekvieno iš tikrinių vektorių.
5. Charakterizuojame įėjimo atvaizdo svorių struktūrą, pagal Euklido minimalų atstumą, priskirdami jį vienai iš dešimties klasių. Buvo panaudoti visi atvaizdai, iš kurių sudaryta duomenų bazė. Visų klasių atvaizdų rezultatai skaitiniu pavidalu - priede nr.1, grafiniu pavidalu – priede nr.2. Pagal gautus minimalius atstumus tarp svorių, teisingai savo klasei buvo priskirti visi naudoti septyniasdešimt aštuoni atvaizdai.

Gautų atstumų pavyzdys - A klasės atvaizdų charakterizavimas - 4.23 pav. Kaip matome minimalus, bet kurio atvaizdo iš klasės A atstumas yra būtent iki savo klasės.



4.23 pav. „A“ klasės atvaizdų svorių atstumai nuo kitų klasių.

Buvo paskaičiuoti visų atvaizdų atstumai nuo bendros rankos erdvės, grafiniai rezultatai matomi 4.24 pav. iš jų galime daryti išvadą, kad įėjimo atvaizdo svorių skirtumas turi būti didesnis už  $1,20E+12$ , tam, kad jis būtų priskirtas ne rankai, o neatpažintam įėjimo objektui.



4.24 pav. Atvaizdų svorių atstumai nuo bendros rankos erdvės.

## IŠVADOS IR SIŪLYMAI

Buvo ištirtos įvairios kompiuterio regėjimo ir atpažinimo technikos ir metodai. Taip pat objekto iš atvaizdo išskyrimo metodai. Panaudojant pagrindo pašalinimo techniką, atskirtas rankos vaizdas nuo pagrindo. Pagal RGB sluoksnių slenkstinių reikšmių vertes pašalinti smulkūs objektai ir triukšmai. Panaudojus rankos odos spalvos RGB ir HSV reikšmes tiksliau išskirti rankos kontūrai atvaizde ir sumažintas šešėlių kiekis. Gautas rezultatas – išskirta rankos padėtis. Sukurta dešimties rankos gestų reikšmių, septyniasdešimt aštuonių atvaizdų bazė. Kaip rodo tyrimas, panaudojus tikrinių vektorių metodą, galima tiksliai rasti rankos poziciją. Taigi įėjimo atvaizdą galima tiksliai priskirti vienai iš aprašytų klasių. Iš gautų rezultatų darome išvadą, kad tikrinių vektorių metodą galima sėkmingai pritaikyti rankos pozicijos radimui. Vienos klasės atvaizdo svorių atstumas (pavyzdžiui A klasės) iki jam priklausančios klasės yra beveik apie 5 kartus mažesnis lyginant su svorių atstumais iki kitų klasių. Didelę įtaką turi įėjimo atvaizdai, jei jie gerai sutvarkyti, išvalyti nuo bereikalingos informacijos, galima tikėtis sėkmingų klasifikavimo rezultatų. Tolimesniame darbe galima būtų panaudoti ir kitus aprašytus klasifikavimo metodus, tokius kaip palėptieji Markovo ar neuronų tinklai. Panaudojus rankos ir raiščio centroidus galima būtų masteliuoti rankos atvaizdą, o juos sujungus į liniją, galima būtų sukurti automatino pasukimo techniką. Kadangi naudojama sukurta programa Matlab aplinkai, galima būtų ją perrašyti C++ programavimo kalbai, tuo pagreitinant programos veikimą ir pritaikyti ją realaus laiko atpažinimui.

## LITERATŪRA

- 1 – Gupta D. *Computer Gesture Recognition*. Caltech Undergraduate Research Journal [interaktyvus]. 2001. [žiūrėta 2005-01-10]. Prieiga per Internetą: [www.curj.caltech.edu/archives/vol1/1002026](http://www.curj.caltech.edu/archives/vol1/1002026).
- 2 – Ying Wu. *Hand modeling, analysis and recognition For Vision-Based Human Computer Interaction*/ Ying Wu, Thomas S. Huang. 2001. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.
- 3 - Richard Bowden. *Learning Non-linear Models of Shape and Motion*. 1999. Department of Systems Engineering, Brunel University. [žiūrėta 2005-02-10]. Prieiga per internetą: [www.ee.surrey.ac.uk/Personal/R.Bowden/publications/thesis/index.html](http://www.ee.surrey.ac.uk/Personal/R.Bowden/publications/thesis/index.html).
- 4 - James H. Stock. *A Note on Linear Classifier*. Department of Economics Harvard University, June 16, 2003. [žiūrėta 2004-12-10]. Prieiga per internetą: [http://ksghome.harvard.edu/~jstock/wrights/lin\\_classifiers.pdf](http://ksghome.harvard.edu/~jstock/wrights/lin_classifiers.pdf)
- 5 - Pridmore T. *Histogram Equalisation*. University of Nottingham, School of Computer Science and IT, 2005. [žiūrėta 2004-12-10]. Prieiga per internetą: <http://www.cs.nott.ac.uk/~tpp/G5A1VI/histogram.pdf>
- 6 - Turk M. *Eigenfaces for recognition* / Turk M., Pentland A.. Journal of Cognitive Neuroscience, 3(1), 1991.
- 7 - Vezhnevets V. *A Survey on Pixel-Based Skin Color Detection Techniques* / Vezhnevets V., Sazonov V., Andreeva A., Graphics and Media Laboratory, Faculty of Computational Mathematics and Cybernetics, Moscow State University. [žiūrėta 2005-02-10]. Prieiga per internetą: <http://graphics.cs.msu.su/en/publications/text/gc2003vsa.pdf>.
- 8 - Vladimir I. Pavlovic. *Visual Interpretation of Hand Gesture for Human-Computer Interaction* / Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 19, NO. 7, JULY 1997
- 9 - Chapman M. *The Karhunen Loeve Transformation for Gappy Data*. Colorado State University Department of Mathematics, december 9, 1999. [žiūrėta 2005-02-10]. Prieiga per internetą: <http://www.math.colostate.edu/~chapman/stuff/paper.pdf>.
- 10 - Richard O. Duda. *Pattern Classification. Secon Edition* / Richard O. Duda, Peter E. Hart, Davis G. Stork. October, 2000. ISBN: 0-471-05669-3.
- 11 - Lockton R. *Real-time gesture recognition using deterministic boosting* // Lockton R, Andrew W. Fitzgibbon. Robotics Research Group in the Department of Engineering Science, University of Oxford. [žiūrėta 2004-10-10]. Prieiga per internetą: <http://www.robots.ox.ac.uk/~vgg/publications/papers/lockton02.pdf>.
- 12 - Linda G. Shapiro. *Computer Vision: 1st edition* / Linda G. Shapiro, George C. Stockman. Prentice Hall, January 23, 2001. ISBN: 0130307963.

13 - David A. Forsyth. *Computer Vision - A Modern Approach, 1st edition* / David A. Forsyth, Jean Ponce. Prentice Hall, August 14, 2002. ISBN: 0130851981 (Prentice Hall 2002).pdf

14 – Corradini A. *Implementation And Comparison Of Three Architectures For Gesture Recognition* / Corradini A., Horst-Michael Gross. Technical University of Ilmenau, Department of Neuroinformatics Ilmenau, Germany. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

15 - Matlab 6.5 versijos žinynas ir dokumentacija.

16 – Bendraukime lietuvių gestų kalba [CD-Rom].

## PRIEDAI

### 1 PRIEDAS

#### ATSTUMŲ REIKŠMĖS SKAITINIŲ PAVIDALU

Klasės Nr.	a0	a1	a2	a3	a4		
1,00	5,51E+05	1,33E+06	1,44E+06	5,93E+05	7,03E+05		
2,00	5,45E+06	5,38E+06	5,09E+06	5,02E+06	5,37E+06		
3,00	6,48E+06	6,82E+06	6,22E+06	6,50E+06	6,52E+06		
4,00	6,49E+06	7,20E+06	6,19E+06	6,66E+06	6,92E+06		
5,00	2,20E+07	2,05E+07	2,30E+07	2,20E+07	2,13E+07		
6,00	1,77E+07	1,64E+07	1,83E+07	1,77E+07	1,69E+07		
7,00	1,58E+07	1,44E+07	1,67E+07	1,60E+07	1,50E+07		
8,00	7,83E+06	6,86E+06	7,84E+06	7,57E+06	7,10E+06		
9,00	1,03E+07	9,61E+06	1,00E+07	9,83E+06	9,71E+06		
10,00	7,99E+06	8,05E+06	7,30E+06	7,62E+06	8,18E+06		
Klasės Nr.	b0	b1	b2	b3	b4	b5	
1,00	4,71E+06	4,90E+06	5,69E+06	5,81E+06	5,36E+06	5,10E+06	
2,00	9,22E+05	1,38E+06	1,13E+06	1,22E+06	4,69E+05	1,02E+06	
3,00	7,25E+06	6,98E+06	7,09E+06	7,04E+06	6,91E+06	6,99E+06	
4,00	6,90E+06	7,17E+06	6,27E+06	6,27E+06	6,45E+06	7,03E+06	
5,00	2,30E+07	2,22E+07	2,43E+07	2,44E+07	2,37E+07	2,25E+07	
6,00	1,83E+07	1,75E+07	1,97E+07	1,98E+07	1,92E+07	1,79E+07	
7,00	1,68E+07	1,60E+07	1,81E+07	1,83E+07	1,76E+07	1,64E+07	
8,00	7,41E+06	6,41E+06	8,30E+06	8,30E+06	7,76E+06	6,76E+06	
9,00	9,03E+06	7,94E+06	9,34E+06	9,21E+06	8,86E+06	8,10E+06	
10,00	6,95E+06	7,35E+06	6,97E+06	7,05E+06	7,33E+06	7,35E+06	
Klasės Nr.	c0	c1	c2	c3	c4	c5	c6
1,00	7,23E+06	6,85E+06	6,67E+06	6,66E+06	6,66E+06	7,02E+06	7,06E+06
2,00	7,43E+06	7,33E+06	7,03E+06	6,97E+06	6,96E+06	7,53E+06	7,61E+06
3,00	3,75E+06	7,87E+05	1,66E+06	1,24E+06	1,05E+06	8,09E+05	1,34E+06
4,00	7,01E+06	4,58E+06	5,35E+06	5,02E+06	4,83E+06	4,05E+06	3,51E+06
5,00	2,31E+07	2,49E+07	2,43E+07	2,46E+07	2,48E+07	2,57E+07	2,61E+07
6,00	1,73E+07	1,96E+07	1,88E+07	1,92E+07	1,94E+07	2,04E+07	2,09E+07
7,00	1,60E+07	1,80E+07	1,72E+07	1,76E+07	1,78E+07	1,88E+07	1,92E+07
8,00	5,34E+06	7,64E+06	6,70E+06	7,07E+06	7,23E+06	8,50E+06	9,01E+06
9,00	5,64E+06	8,32E+06	7,41E+06	7,74E+06	7,89E+06	9,22E+06	9,83E+06
10,00	1,04E+07	1,07E+07	1,03E+07	1,04E+07	1,04E+07	1,10E+07	1,11E+07
Klasės Nr.	c7	c8	c9				
1,00	6,73E+06	5,91E+06	6,34E+06				
2,00	7,48E+06	6,77E+06	7,14E+06				
3,00	2,33E+06	2,38E+06	2,08E+06				
4,00	2,75E+06	3,00E+06	2,63E+06				
5,00	2,62E+07	2,55E+07	2,57E+07				
6,00	2,12E+07	2,06E+07	2,07E+07				
7,00	1,95E+07	1,88E+07	1,89E+07				
8,00	9,52E+06	9,01E+06	9,07E+06				
9,00	1,06E+07	1,04E+07	1,03E+07				
10,00	1,10E+07	1,02E+07	1,05E+07				

Klasés Nr.	cc0	cc1	cc2	cc3	cc4	cc5	cc6
1,00	6,80E+06	7,16E+06	6,62E+06	7,03E+06	7,26E+06	6,56E+06	6,91E+06
2,00	6,50E+06	6,90E+06	6,69E+06	6,89E+06	7,37E+06	6,83E+06	7,10E+06
3,00	4,16E+06	3,85E+06	4,19E+06	4,02E+06	4,58E+06	4,18E+06	4,38E+06
4,00	6,31E+05	7,63E+05	3,88E+05	6,51E+05	9,82E+05	8,45E+05	9,13E+05
5,00	2,64E+07	2,66E+07	2,62E+07	2,66E+07	2,68E+07	2,57E+07	2,62E+07
6,00	2,17E+07	2,19E+07	2,16E+07	2,20E+07	2,22E+07	2,13E+07	2,17E+07
7,00	1,97E+07	1,99E+07	1,95E+07	2,00E+07	2,01E+07	1,91E+07	1,96E+07
8,00	9,97E+06	1,01E+07	1,01E+07	1,02E+07	1,08E+07	1,01E+07	1,04E+07
9,00	1,08E+07	1,08E+07	1,11E+07	1,10E+07	1,18E+07	1,13E+07	1,16E+07
10,00	9,50E+06	1,00E+07	9,78E+06	1,00E+07	1,04E+07	9,96E+06	1,02E+07
Klasés Nr.	cc7	cc8					
1,00	5,83E+06	6,19E+06					
2,00	6,08E+06	5,76E+06					
3,00	3,79E+06	3,61E+06					
4,00	1,06E+06	1,42E+06					
5,00	2,51E+07	2,58E+07					
6,00	2,05E+07	2,09E+07					
7,00	1,85E+07	1,91E+07					
8,00	9,23E+06	8,99E+06					
9,00	1,06E+07	9,86E+06					
10,00	8,98E+06	8,64E+06					
Klasés Nr.	e0	e1	e2	e3	e4	e5	e6
1,00	2,20E+07	2,23E+07	2,18E+07	2,26E+07	2,17E+07	2,19E+07	2,14E+07
2,00	2,35E+07	2,39E+07	2,33E+07	2,42E+07	2,34E+07	2,35E+07	2,31E+07
3,00	2,53E+07	2,56E+07	2,50E+07	2,58E+07	2,50E+07	2,52E+07	2,47E+07
4,00	2,63E+07	2,66E+07	2,61E+07	2,68E+07	2,61E+07	2,62E+07	2,58E+07
5,00	6,34E+05	6,57E+05	4,92E+05	9,89E+05	3,76E+05	5,45E+05	5,04E+05
6,00	1,18E+07	1,21E+07	1,16E+07	1,24E+07	1,15E+07	1,18E+07	1,13E+07
7,00	1,02E+07	1,04E+07	9,81E+06	1,06E+07	9,78E+06	9,98E+06	9,51E+06
8,00	2,03E+07	2,07E+07	2,00E+07	2,10E+07	2,01E+07	2,03E+07	1,99E+07
9,00	2,30E+07	2,34E+07	2,27E+07	2,37E+07	2,28E+07	2,32E+07	2,27E+07
10,00	2,34E+07	2,38E+07	2,32E+07	2,40E+07	2,32E+07	2,33E+07	2,29E+07
Klasés Nr.	e7	e8					
1,00	2,01E+07	2,21E+07					
2,00	2,17E+07	2,37E+07					
3,00	2,33E+07	2,54E+07					
4,00	2,44E+07	2,65E+07					
5,00	2,00E+06	5,41E+05					
6,00	9,83E+06	1,16E+07					
7,00	7,90E+06	1,00E+07					
8,00	1,84E+07	2,04E+07					
9,00	2,12E+07	2,31E+07					
10,00	2,17E+07	2,36E+07					

Klasés Nr.	ee0	ee1	ee2	ee3	ee4	ee5	ee7
1,00	1,73E+07	1,70E+07	1,73E+07	1,83E+07	1,72E+07	1,72E+07	1,82E+07
2,00	1,87E+07	1,81E+07	1,88E+07	1,97E+07	1,83E+07	1,86E+07	1,97E+07
3,00	1,98E+07	1,92E+07	1,97E+07	2,08E+07	1,95E+07	1,96E+07	2,08E+07
4,00	2,15E+07	2,09E+07	2,14E+07	2,24E+07	2,12E+07	2,13E+07	2,24E+07
5,00	1,13E+07	1,16E+07	1,17E+07	1,08E+07	1,13E+07	1,21E+07	1,08E+07
6,00	5,98E+05	1,17E+06	6,30E+05	1,26E+06	1,11E+06	7,91E+05	1,32E+06
7,00	6,00E+06	5,71E+06	5,83E+06	6,07E+06	5,72E+06	6,14E+06	6,11E+06
8,00	1,40E+07	1,32E+07	1,39E+07	1,50E+07	1,35E+07	1,37E+07	1,50E+07
9,00	1,68E+07	1,60E+07	1,68E+07	1,78E+07	1,62E+07	1,65E+07	1,78E+07
10,00	1,83E+07	1,79E+07	1,84E+07	1,93E+07	1,81E+07	1,82E+07	1,93E+07
Klasés Nr.	ee8	ee9					
1,00	1,67E+07	1,73E+07					
2,00	1,81E+07	1,88E+07					
3,00	1,90E+07	1,96E+07					
4,00	2,07E+07	2,14E+07					
5,00	1,25E+07	1,19E+07					
6,00	1,28E+06	7,22E+05					
7,00	6,01E+06	5,81E+06					
8,00	1,30E+07	1,37E+07					
9,00	1,58E+07	1,65E+07					
10,00	1,77E+07	1,84E+07					
Klasés Nr.	f0	f1	f2	f3	f4		
1,00	1,63E+07	1,56E+07	1,56E+07	1,51E+07	1,54E+07		
2,00	1,80E+07	1,72E+07	1,72E+07	1,66E+07	1,70E+07		
3,00	1,91E+07	1,82E+07	1,81E+07	1,74E+07	1,78E+07		
4,00	2,04E+07	1,95E+07	1,94E+07	1,88E+07	1,92E+07		
5,00	8,14E+06	9,68E+06	9,72E+06	1,09E+07	1,06E+07		
6,00	6,29E+06	6,15E+06	6,26E+06	5,52E+06	5,46E+06		
7,00	1,78E+06	3,80E+05	4,26E+05	1,24E+06	9,56E+05		
8,00	1,41E+07	1,32E+07	1,31E+07	1,23E+07	1,26E+07		
9,00	1,73E+07	1,65E+07	1,64E+07	1,56E+07	1,59E+07		
10,00	1,80E+07	1,73E+07	1,72E+07	1,66E+07	1,69E+07		



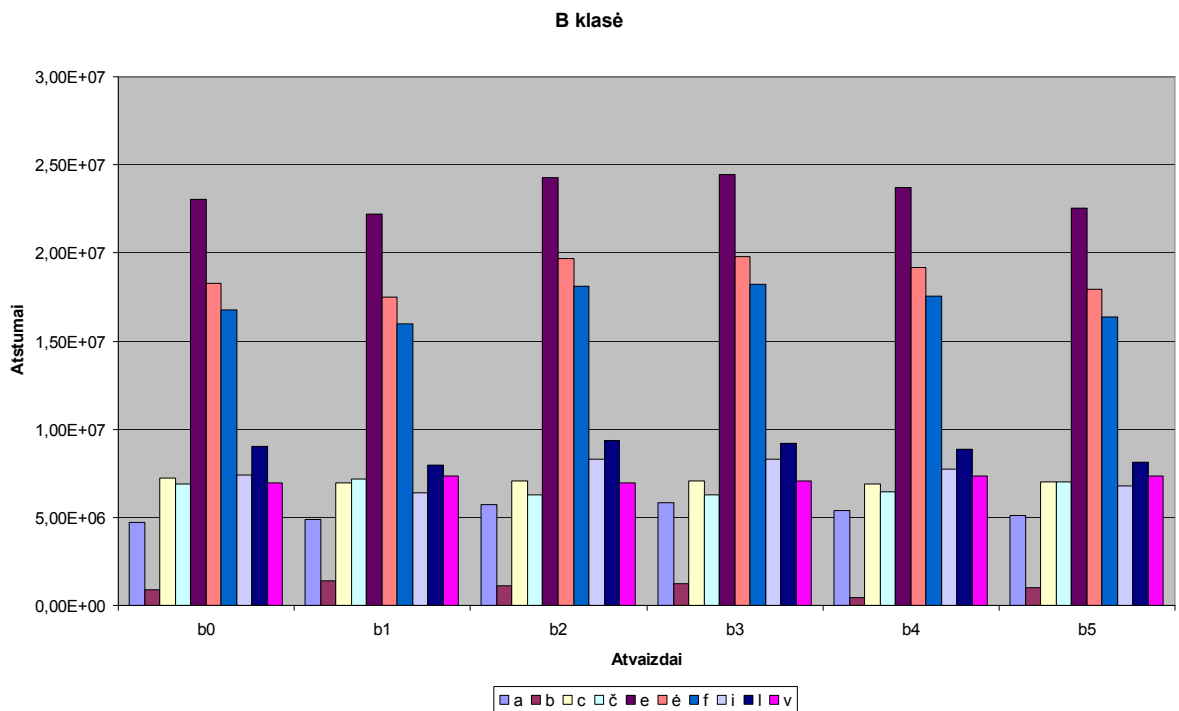
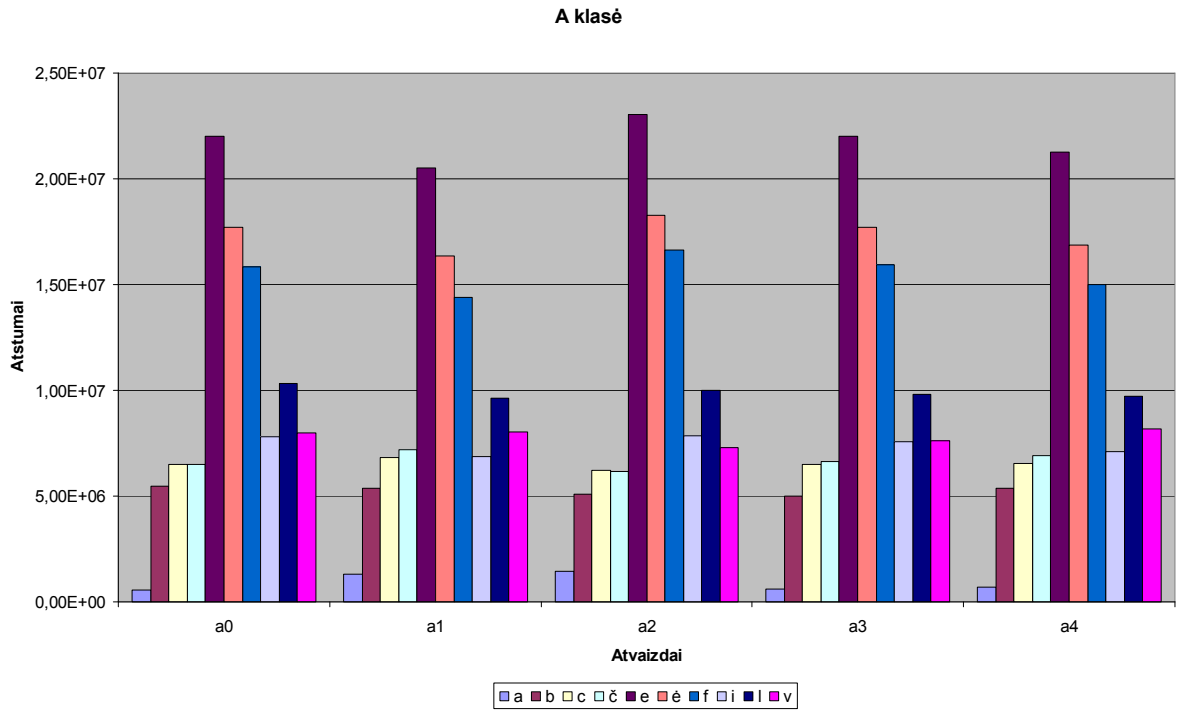
Klasés Nr.	i0	i1	i2	i3	i4	i5	i6
1,00	5,50E+06	6,18E+06	8,33E+06	9,52E+06	1,06E+07	8,47E+06	7,91E+06
2,00	4,70E+06	5,86E+06	8,43E+06	9,80E+06	1,11E+07	8,68E+06	8,00E+06
3,00	5,63E+06	5,82E+06	7,95E+06	9,78E+06	1,16E+07	9,21E+06	7,99E+06
4,00	6,89E+06	7,76E+06	1,04E+07	1,21E+07	1,37E+07	1,14E+07	1,03E+07
5,00	2,27E+07	2,21E+07	2,06E+07	1,91E+07	1,75E+07	1,90E+07	2,02E+07
6,00	1,70E+07	1,63E+07	1,41E+07	1,23E+07	1,02E+07	1,26E+07	1,40E+07
7,00	1,59E+07	1,52E+07	1,34E+07	1,18E+07	9,94E+06	1,20E+07	1,32E+07
8,00	4,18E+06	2,96E+06	1,58E+06	2,98E+06	4,75E+06	1,71E+06	9,03E+05
9,00	6,35E+06	5,54E+06	5,04E+06	6,01E+06	7,55E+06	5,78E+06	5,35E+06
10,00	7,56E+06	8,65E+06	1,08E+07	1,16E+07	1,21E+07	1,06E+07	1,04E+07
Klasés Nr.	i7	i8					
1,00	7,20E+06	6,01E+06					
2,00	7,15E+06	5,87E+06					
3,00	7,33E+06	7,36E+06					
4,00	9,46E+06	8,86E+06					
5,00	2,07E+07	2,04E+07					
6,00	1,48E+07	1,47E+07					
7,00	1,39E+07	1,36E+07					
8,00	1,37E+06	2,46E+06					
9,00	5,49E+06	6,29E+06					
10,00	9,64E+06	8,14E+06					
Klasés Nr.	i0	i1	i2	i3	i4	i5	i6
1,00	5,50E+06	6,18E+06	8,33E+06	9,52E+06	1,06E+07	8,47E+06	7,91E+06
2,00	4,70E+06	5,86E+06	8,43E+06	9,80E+06	1,11E+07	8,68E+06	8,00E+06
3,00	5,63E+06	5,82E+06	7,95E+06	9,78E+06	1,16E+07	9,21E+06	7,99E+06
4,00	6,89E+06	7,76E+06	1,04E+07	1,21E+07	1,37E+07	1,14E+07	1,03E+07
5,00	2,27E+07	2,21E+07	2,06E+07	1,91E+07	1,75E+07	1,90E+07	2,02E+07
6,00	1,70E+07	1,63E+07	1,41E+07	1,23E+07	1,02E+07	1,26E+07	1,40E+07
7,00	1,59E+07	1,52E+07	1,34E+07	1,18E+07	9,94E+06	1,20E+07	1,32E+07
8,00	4,18E+06	2,96E+06	1,58E+06	2,98E+06	4,75E+06	1,71E+06	9,03E+05
9,00	6,35E+06	5,54E+06	5,04E+06	6,01E+06	7,55E+06	5,78E+06	5,35E+06
10,00	7,56E+06	8,65E+06	1,08E+07	1,16E+07	1,21E+07	1,06E+07	1,04E+07
Klasés Nr.	i7	i8					
1,00	7,20E+06	6,01E+06					
2,00	7,15E+06	5,87E+06					
3,00	7,33E+06	7,36E+06					
4,00	9,46E+06	8,86E+06					
5,00	2,07E+07	2,04E+07					
6,00	1,48E+07	1,47E+07					
7,00	1,39E+07	1,36E+07					
8,00	1,37E+06	2,46E+06					
9,00	5,49E+06	6,29E+06					
10,00	9,64E+06	8,14E+06					

Klasés Nr.	l0	l1	l3	l4	l5	l6	
1,00	1,04E+07	9,77E+06	1,00E+07	9,78E+06	9,56E+06	1,00E+07	
2,00	9,20E+06	8,66E+06	8,93E+06	8,48E+06	8,42E+06	8,96E+06	
3,00	9,01E+06	8,66E+06	9,35E+06	8,50E+06	8,24E+06	8,63E+06	
4,00	1,12E+07	1,10E+07	1,17E+07	1,08E+07	1,04E+07	1,09E+07	
5,00	2,32E+07	2,25E+07	2,17E+07	2,33E+07	2,34E+07	2,32E+07	
6,00	1,71E+07	1,62E+07	1,56E+07	1,72E+07	1,72E+07	1,70E+07	
7,00	1,67E+07	1,59E+07	1,53E+07	1,68E+07	1,68E+07	1,66E+07	
8,00	5,77E+06	5,03E+06	4,95E+06	5,55E+06	5,47E+06	5,57E+06	
9,00	8,69E+05	1,18E+06	1,61E+06	8,56E+05	1,54E+06	8,14E+05	
10,00	1,11E+07	1,12E+07	1,13E+07	1,11E+07	1,02E+07	1,09E+07	
Klasés Nr.	v0	v1	v2	v3	v4	v5	v6
1,00	7,46E+06	7,81E+06	8,27E+06	8,52E+06	8,50E+06	8,54E+06	8,50E+06
2,00	7,20E+06	7,47E+06	7,86E+06	7,99E+06	7,82E+06	7,85E+06	7,72E+06
3,00	9,75E+06	1,01E+07	1,09E+07	1,10E+07	1,12E+07	1,11E+07	1,11E+07
4,00	9,23E+06	9,51E+06	1,00E+07	1,02E+07	1,03E+07	1,05E+07	1,04E+07
5,00	2,31E+07	2,33E+07	2,36E+07	2,38E+07	2,35E+07	2,32E+07	2,32E+07
6,00	1,80E+07	1,83E+07	1,88E+07	1,89E+07	1,87E+07	1,83E+07	1,84E+07
7,00	1,69E+07	1,72E+07	1,76E+07	1,78E+07	1,75E+07	1,72E+07	1,72E+07
8,00	9,06E+06	9,45E+06	1,03E+07	1,04E+07	1,04E+07	9,91E+06	9,95E+06
9,00	1,01E+07	1,04E+07	1,15E+07	1,14E+07	1,17E+07	1,10E+07	1,11E+07
10,00	2,02E+06	1,85E+06	1,07E+06	1,36E+06	8,65E+05	1,25E+06	9,05E+05
Klasés Nr.	v7	v8	v9				
1,00	7,32E+06	7,53E+06	7,10E+06				
2,00	6,44E+06	6,55E+06	5,90E+06				
3,00	1,03E+07	1,04E+07	9,83E+06				
4,00	9,50E+06	9,61E+06	9,02E+06				
5,00	2,30E+07	2,31E+07	2,33E+07				
6,00	1,83E+07	1,84E+07	1,86E+07				
7,00	1,70E+07	1,71E+07	1,71E+07				
8,00	9,54E+06	9,63E+06	9,29E+06				
9,00	1,12E+07	1,12E+07	1,09E+07				
10,00	2,24E+06	2,21E+06	2,86E+06				

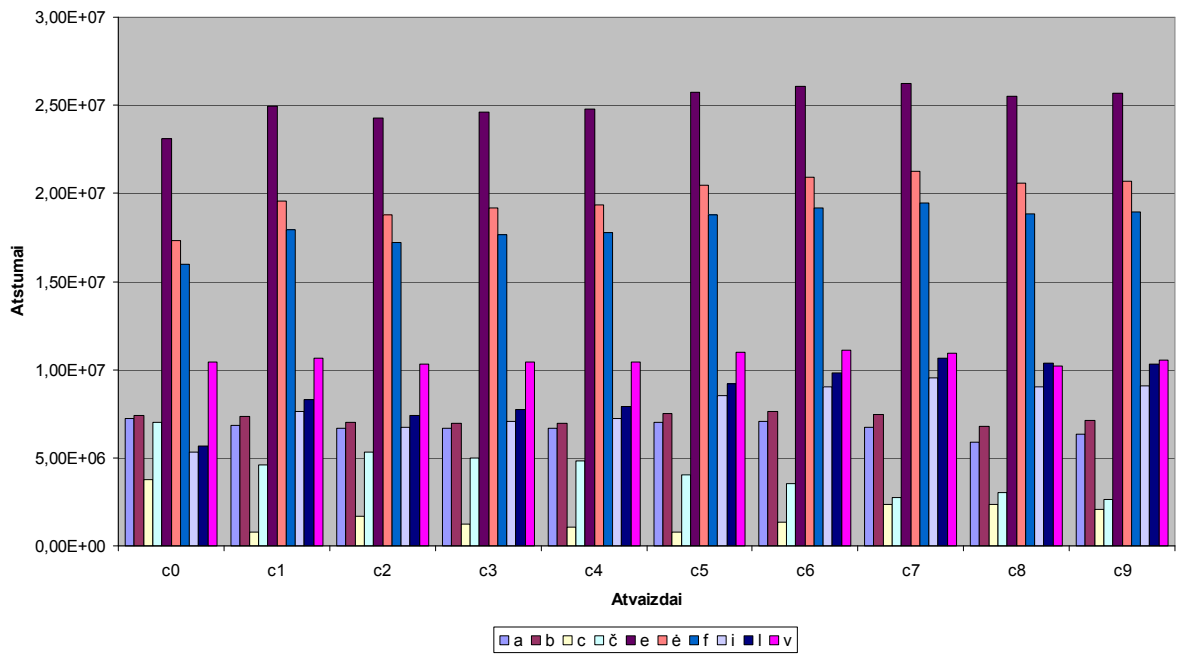
VISŲ ATVAIZDŲ ATSTUMAI NUO RANKOS ERDVĖS									
<b>a0</b>	<b>a1</b>	<b>a2</b>	<b>a3</b>	<b>a4</b>					
1,51E+10	1,11E+14	1,75E+10	1,52E+10	1,28E+10					
<b>b0</b>	<b>b1</b>	<b>b2</b>	<b>b3</b>	<b>b4</b>	<b>b5</b>				
1,74E+10	1,50E+10	2,14E+10	2,19E+10	1,98E+10	1,61E+10				
<b>c0</b>	<b>c1</b>	<b>c2</b>	<b>c3</b>	<b>c4</b>	<b>c5</b>	<b>c6</b>	<b>c7</b>	<b>c8</b>	<b>c9</b>
1,68E+10	2,27E+10	2,05E+10	2,17E+10	2,22E+10	2,52E+10	2,65E+10	2,72E+10	2,52E+10	2,56E+10
<b>č0</b>	<b>č1</b>	<b>č2</b>	<b>č3</b>	<b>č4</b>	<b>č5</b>	<b>č6</b>	<b>č7</b>	<b>č8</b>	
2,81E+10	2,87E+10	2,75E+10	2,89E+10	2,93E+10	2,63E+10	2,78E+10	2,45E+10	2,61E+10	
<b>e0</b>	<b>e1</b>	<b>e2</b>	<b>e3</b>	<b>e4</b>	<b>e5</b>	<b>e6</b>	<b>e7</b>	<b>e8</b>	
5,13E+10	5,23E+10	5,06E+10	5,30E+10	5,07E+10	5,11E+10	4,98E+10	4,59E+10	5,19E+10	
<b>ė0</b>	<b>ė1</b>	<b>ė2</b>	<b>ė3</b>	<b>ė4</b>	<b>ė5</b>	<b>ė7</b>	<b>ė8</b>	<b>ė9</b>	
3,44E+10	3,27E+10	3,42E+10	3,76E+10	3,36E+10	3,35E+10	3,75E+10	3,18E+10	3,40E+10	
<b>f0</b>	<b>f1</b>	<b>f2</b>	<b>f3</b>	<b>f4</b>					
3,36E+10	3,05E+10	3,03E+10	2,81E+10	2,92E+10					
<b>i0</b>	<b>i1</b>	<b>i2</b>	<b>i3</b>	<b>i4</b>	<b>i5</b>	<b>i6</b>	<b>i7</b>	<b>i8</b>	
1,52E+10	1,34E+10	9,80E+09	8,85E+09	1,07E+10	7,27E+09	8,75E+09	9,60E+09	8,30E+09	
<b>l0</b>	<b>l1</b>	<b>l3</b>	<b>l4</b>	<b>l5</b>	<b>l6</b>				
1,68E+10	1,50E+10	1,34E+10	1,70E+10	1,71E+10	1,67E+10				
<b>v0</b>	<b>v1</b>	<b>v2</b>	<b>v3</b>	<b>v4</b>	<b>v5</b>	<b>v6</b>	<b>v7</b>	<b>v8</b>	<b>v9</b>
1,77E+10	1,85E+10	1,97E+10	2,01E+10	1,96E+10	1,85E+10	1,87E+10	1,81E+10	1,83E+10	1,86E+10

## 2 PRIEDAS

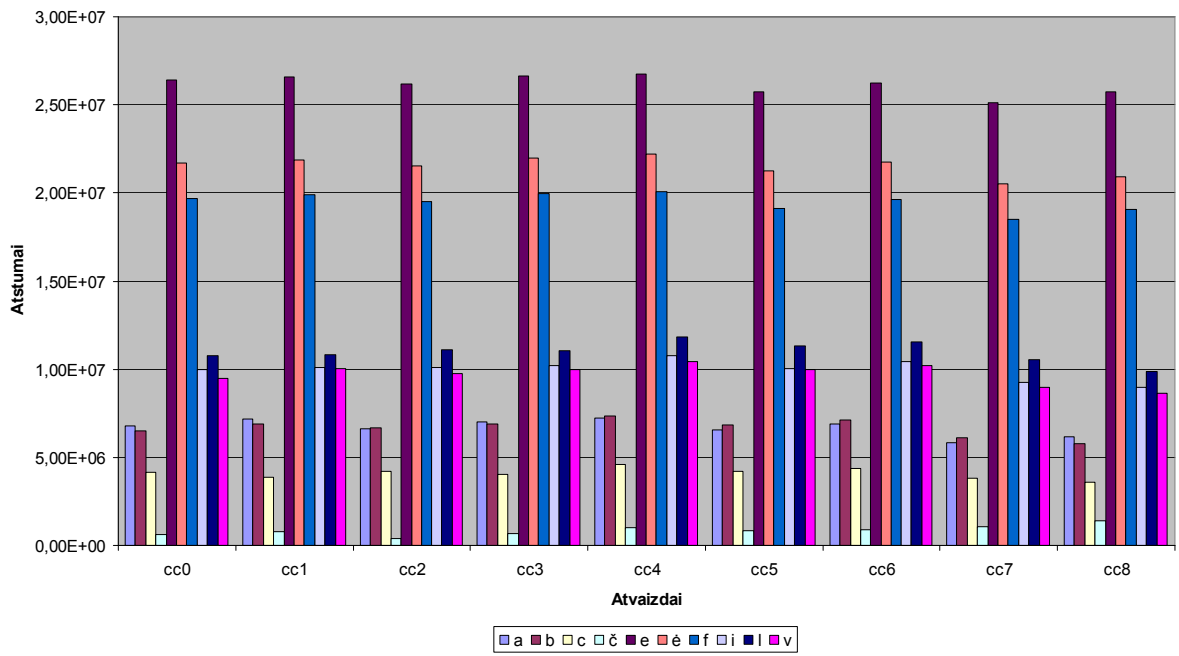
### KLASIŲ ATSTUMAI GRAFINIU PAVIDALU



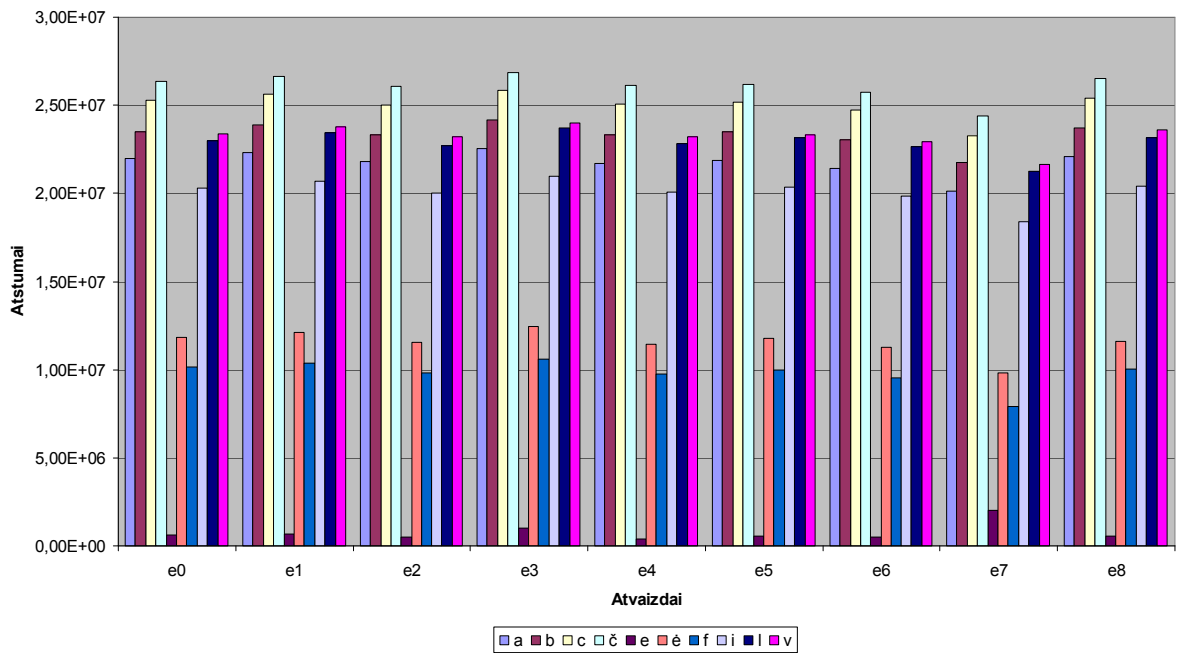
C klasė



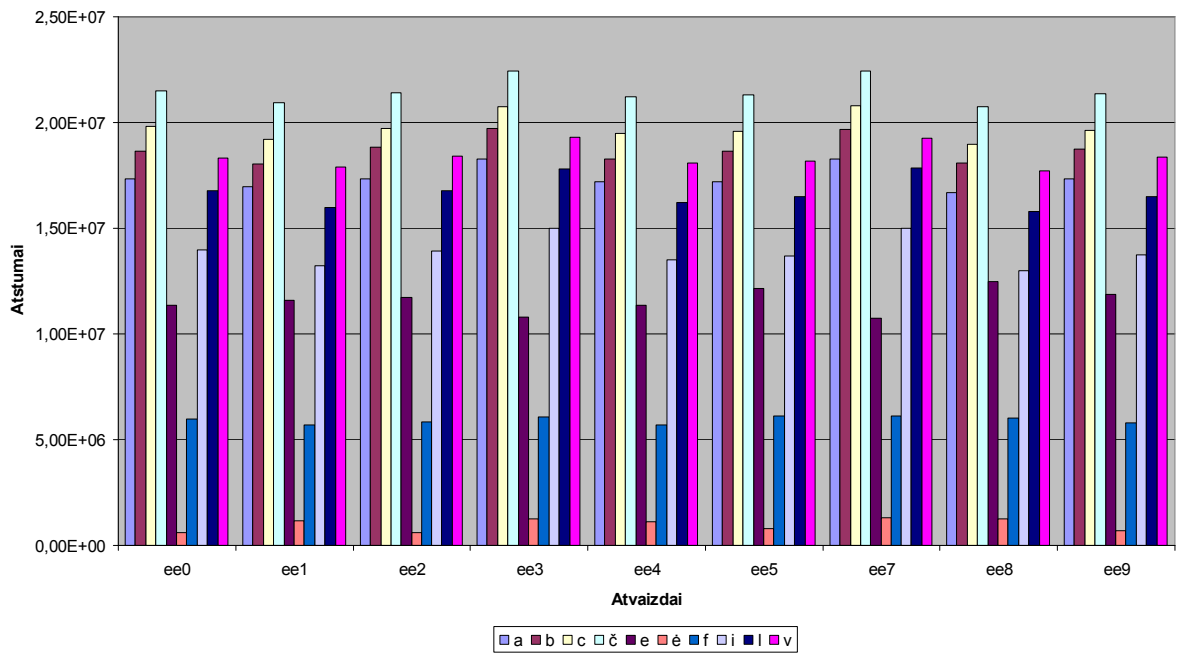
Č klasė



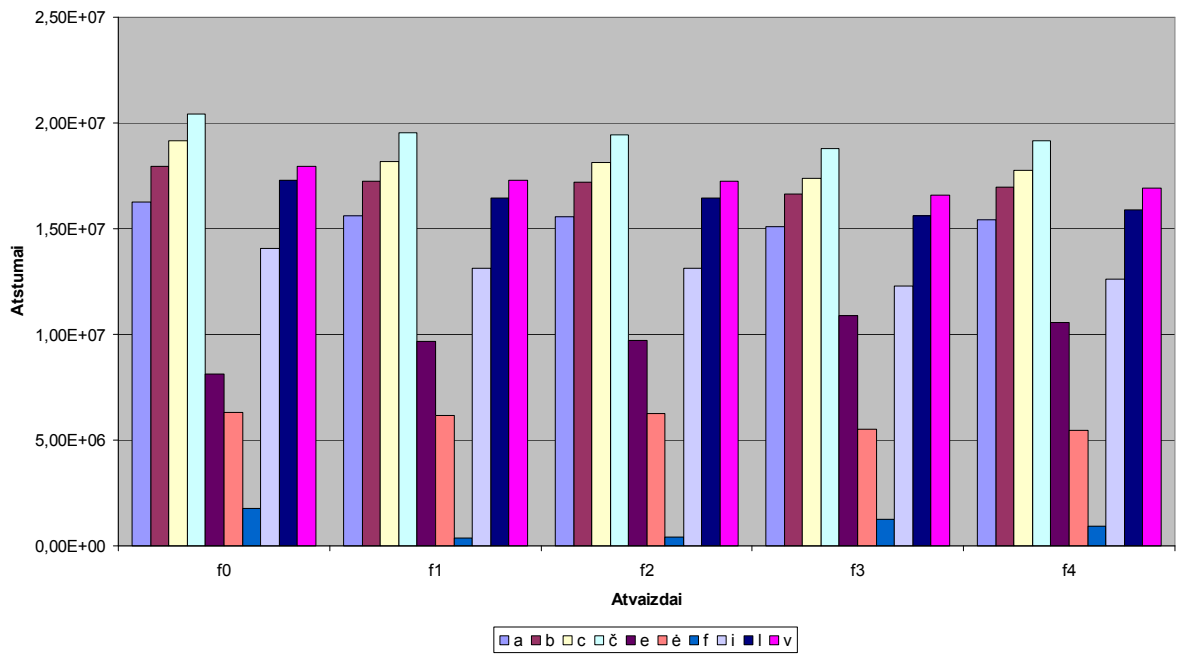
### E klasė



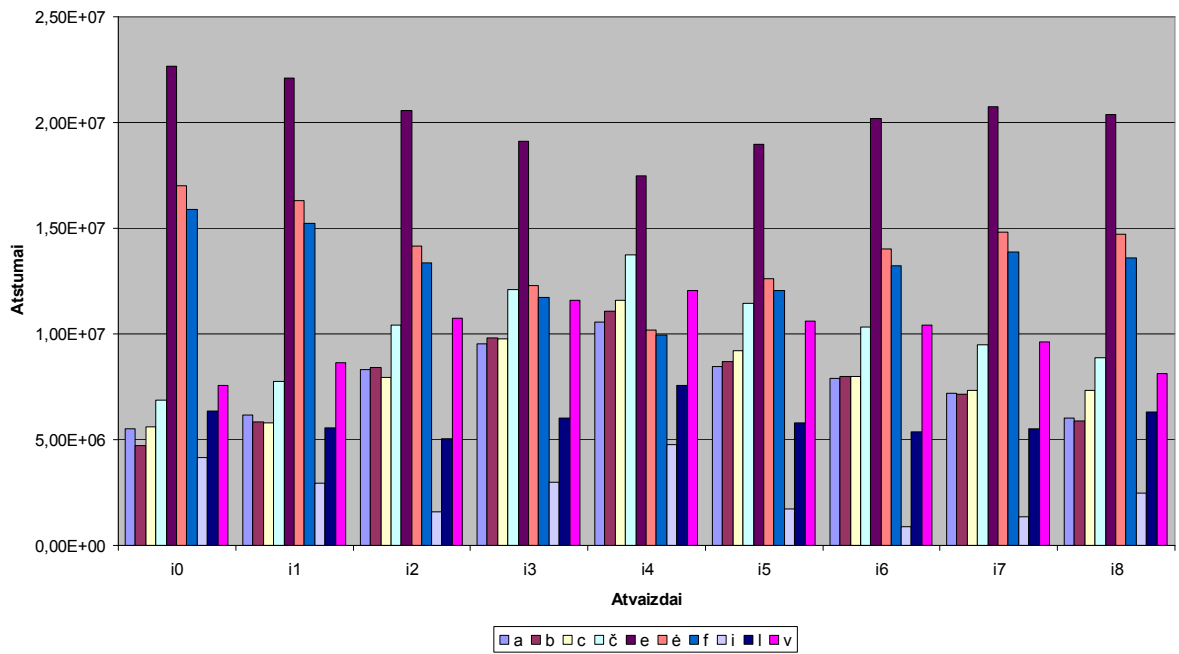
### É klasė



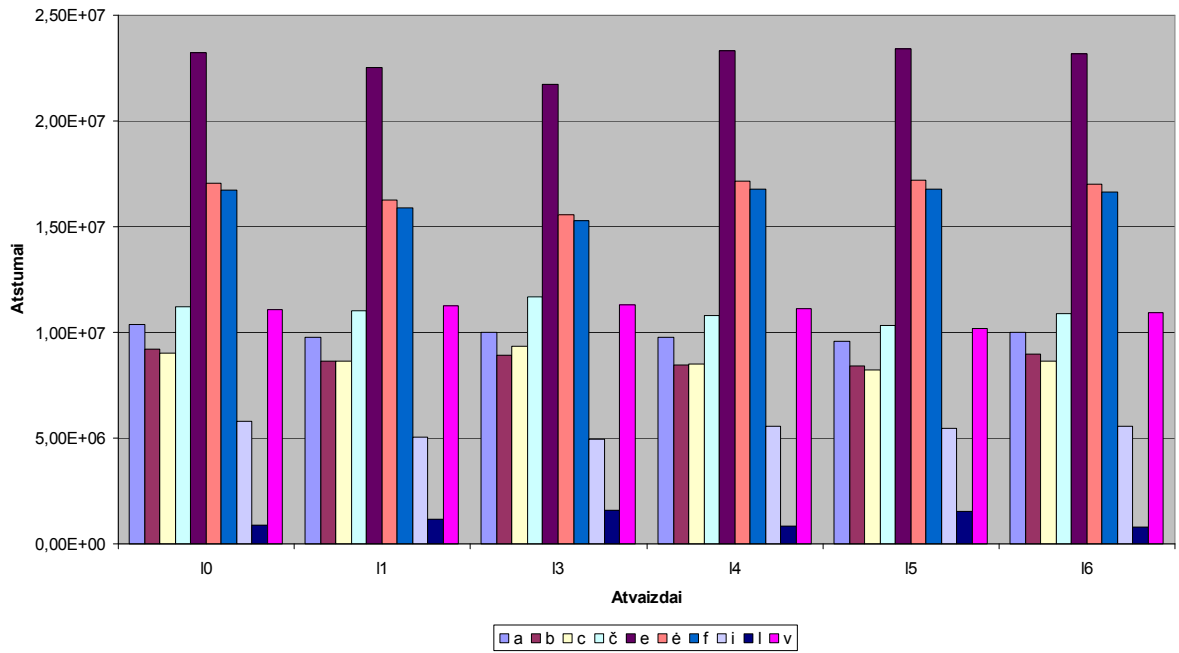
### F klasė



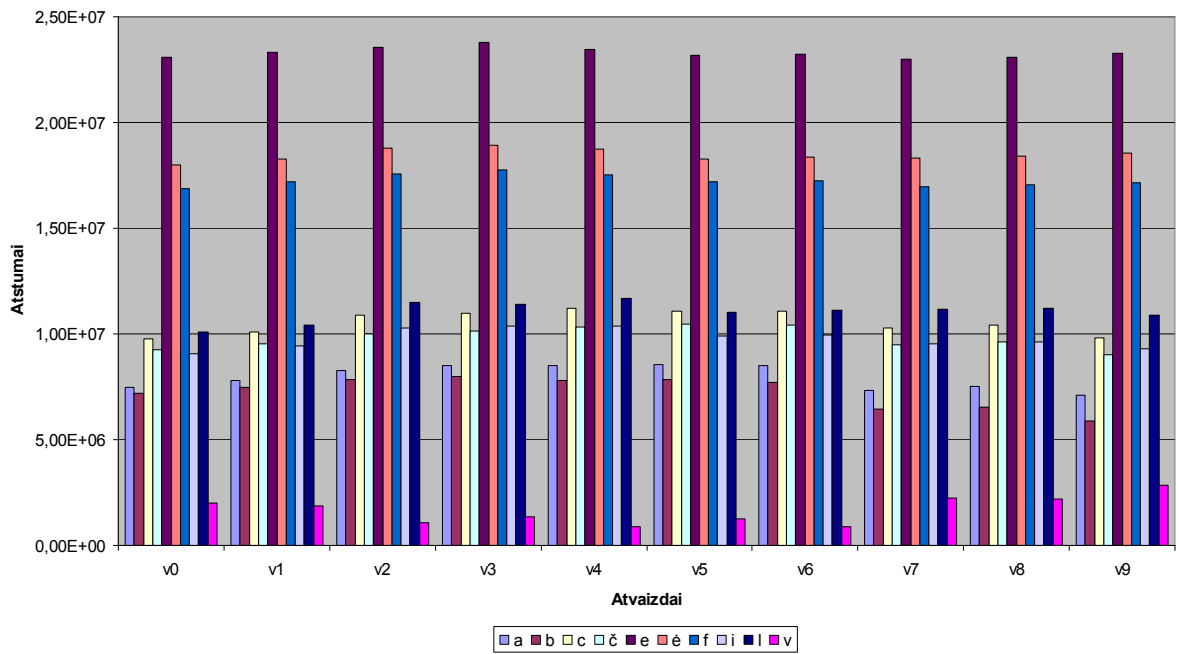
### I klasė



### L klasė



### V klasė





### 3. PRIEDAS

#### „MATLAB“ PROGRAMOS KODAS

Programa susideda iš kelių modulių.

**Pagrindinis modulis:**

```
clear;
clc;
pasirinkimas=12;
chos=0;
pranesimas='Iveskite rinkinio nr.: kiekvienas rinkinys apibudina klase.';

while chos~=pasirinkimas,
    chos=menu('Programa','Kalibravimas','Duomenu bazes sudarymas','Kadro issirinkimas','Rankos
atpazinimas','Duomenu bazes naikinimas','Duomenu bazes info','Video medziagos irasymas','Video
perziurejimas','Informacija','Duomenu uzkrovimas','Kalib. duomenu issaugojimas','Isejimas');
    %-----
    if chos==1,
        clc;

[levelred,levelgreen,levelblue,hsv_colorlimits_sk,rgb_colorlimits_sk,failopav,keliovardas,hsv_colorlimits,rgb_
colorlimits]= kalib_ranka();%

    end
    %-----
    if chos==2,
        clc;

[galas]=apdoroj1(levelred,levelgreen,levelblue,hsv_colorlimits_sk,rgb_colorlimits_sk,failopav,keliovardas,hsv_
_colorlimits,rgb_colorlimits);

        if exist('galas')
            if (exist('rankos_database.dat')==2)
                load('rankos_database.dat','-mat');
                rankos_nr=rankos_nr+1;
                data{rankos_nr,1}=galas(:);
                prompt={strcat(pranesimas,'Klases nr. turi buti pozityvus integer tipo skaicius <=
,num2str(max_class))};
                title='Klases pavadinimas';
                lines=1;
                def={'1'};
                answer=inputdlg(prompt,title,lines,def);
                zparameter=double(str2num(char(answer)));
                if size(zparameter,1)~=0
                    class_number=zparameter(1);
                    if
(class_number<=0)||((class_number>max_class)||((floor(class_number)~=class_number)||(~isa(class_number
,'double'))||((any(any(imag(class_number))))
                    warndlg(strcat('Klases numeris turi buti pozityvus integer tipo skaicius <=
,num2str(max_class))',' Warning '))
                    else
                        if class_number==max_class;
                            max_class=class_number+1;
                        end
                        data{rankos_nr,2}=class_number;
                        save('rankos_database.dat','data','rankos_nr','max_class','-append');
                        msgbox(strcat('Duomenu baze egzituoja: atvaizdas sekmingai pridetas prie klases
numeriu',num2str(class_number)),'Duomenu bazes rezultatas','help');
                        close all;
                        clear('galas')
                    end
                else
```

```

                warndlg(strcat('Klases numeris turi buti pozityvus "integer" tipo skaicius <=
',num2str(max_class)), ' Warning ')
            end
        else
            rankos_nr=1;
            max_class=1;
            data{rankos_nr,1}=galas(:);
            prompt={strcat(pranesimas,'Klases numeris turi buti pozityvus "integer" tipo skaicius <=
',num2str(max_class))};
            title='Klases numeris';
            lines=1;
            def={'1'};
            answer=inputdlg(prompt,title,lines,def);
            zparameter=double(str2num(char(answer)));
            if size(zparameter,1)~=0
                class_number=zparameter(1);
                if
                    (class_number<=0)||((class_number>max_class)||((floor(class_number)~=class_number)||(~isa(class_number
                    ,'double'))||((any(any(imag(class_number))))
                    warndlg(strcat('Klases numeris turi buti pozityvus "integer" tipo skaicius <=
',num2str(max_class)), ' Warning ')
                else
                    max_class=2;
                    data{rankos_nr,2}=class_number;
                    save('rankos_database.dat','data','rankos_nr','max_class');
                    msgbox(strcat('Duomenu baze buvo tuscia. Duomenu baze sukurta. Atvaizdas sekmingai
                    pridetas prie klases numeriu',num2str(class_number)), 'Duomenu bazes rezultatas','help');
                    close all;
                    clear('galas')
                end
            else
                warndlg(strcat('Klases numeris turi buti pozityvus "integer" tipo skaicius <=
',num2str(max_class)), ' Warning ')
            end

            end
        else
            errordlg('Nera atvaizdo.','File Error');
        end
    end
    %-----
    if chos==3,
        clc;
        [galas]=apdoroj1(levelred,levelgreen,levelblue,hsv_colorlimits_sk,rgb_colorlimits_sk,failopav,keliovardas,hsv
        _colorlimits,rgb_colorlimits);
        if exist('galas')
            figure
            imshow(galas)
        else
            warndlg('Input image must be selected.',' Warning ')
        end
    end
    %-----
    if chos==4,
        clc;
        close all;
        if exist('galas')
            kintamasis=double(galas(:));
            if (exist('rankos_database.dat')==2)
                load('rankos_database.dat','-mat');
                matrica=zeros(size(data{1,1},1),rankos_nr);
                for ii=1:rankos_nr
                    matrica(:,ii)=double(data{ii,1});

```

```

end
suma=sum(matrica,2);
vidurkis=suma/rankos_nr;
for ii=1:rankos_nr
    matrica(:,ii)=matrica(:,ii)-vidurkis;
end
matrica=matrica/sqrt(rankos_nr);
% iki dabar matrica "matrica" yra is teorijos matrica A
matrc=matrica*matrica; %78x78 kovariacijos matrica
% matrica "matrc" yra matrica L is teorijos

%"sumazintos" matricos A*A eigen reiksmes ir eigen vektoriai
[V,D]=eig(matrc);
% sekantis skaiciavimas atliekamas rasti
% originalios matricos A*A' eigen vektoriams
Vtrue=matrica*V;
Dtrue=diag(D); %

% eigen reiksmes isrusiuojamos pagal eile ir tik M'
% paimama. Ivedame, kad M' lygus klasiu skaiciui
% (max_class-1)
[Dtrue,papr]=sort(Dtrue);
Dtrue=flipud(Dtrue);
papr=flipud(papr);
Vtrue(:,1:rankos_nr)=Vtrue(:,papr);

Vtrue=Vtrue(:,1:max_class-1);
Dtrue=Dtrue(1:max_class-1);
% normalizuoto iejimo (mean-adjusted). I.e. iejimo
% atvaizdas projektuojamas i "rankos-space"
reiks=Vtrue*(kintamasis-vidurkis);
reiks_database=zeros(max_class-1,max_class-1);
nr_elementu_klase=zeros(max_class-1,1);
for ii=1:rankos_nr
    kintamasis_database=double(data{ii,1});
    klasiu_database=data{ii,2};
    reiks_kintamasis=Vtrue*(kintamasis_database-vidurkis);
    reiks_database(:,klasiu_database)=reiks_database(:,klasiu_database)+reiks_kintamasis;
    nr_elementu_klase(klasiu_database)=nr_elementu_klase(klasiu_database)+1;
end
for ii=1:(max_class-1)
    reiks_database_vidurkiai(:,ii)=reiks_database(:,ii)/nr_elementu_klase(ii);
end
% reiks_database_vidurkiai yra is suvidurkintu eigenrankos komponentu, kurie yra is atvaizdu
% paduodamu i duomenu baze. Kiekviena klase turi savo suvidurkinta "eigenrankos" objekta.
% Randame arciausia iejimo eigenrankos komponentu vektoriui
atstumo_reiks=zeros(max_class-1,1);
for ii=1:(max_class-1)
    atstumo_reiks(ii)=norm(reiks-reiks_database_vidurkiai(:,ii));
end

[minimali_reiks,pozicijos_minimali_reiks]=min(atstumo_reiks);

% ivertiname atstuma tarp normalizuotos (mean-normalized)
% iejimo rankos is "space-rankos" tam, kad nuspresti ar
% iejimo atvaizdas yra ranka.
poz=zeros(size(data{1,1},1),1);
for ii=1:(max_class-1)
    poz=poz+reiks(ii)*Vtrue(:,ii);
end
rankos_atstumo_erdve=norm((kintamasis-vidurkis)-poz);

```

```

pranesimas1='Matlab komandiniame lange galime pamatyti rezultatus. Programa ka tik
suskaiciavo minimalu atstuma tarp rankos ir klasiu erdviu. ';
pranesimas2=' Galima pataisyti ribu reiksmes rezultato teisingumui ivertinti. Jei abi gautos
reiksmes zemesnes negu riba, reiskia, kad ';
pranesimas3=' rankos atvaizdas buvo teisingai palygintas. Jei atstumas tarp ranku erdviu zemiau
nustatytos ribos, o minimalus atstumas nuo klases auksciau ribos ';
pranesimas4=' reiskia, kad iejimas yra nezinomas.';

msgbox(strcat(pranesimas1,pranesimas2,pranesimas3,pranesimas4),'Rezultatas','help');

disp('Arciausios klases numeris: ');
disp(pozicijos_minimali_reiks);
disp('atstumas lygus');
disp(minimali_reiks);
disp('Rankos erdves atstumas: ');
disp(rankos_atstumo_erdve);

else
    warndlg('Negalimas atvaizdo padorojimas. Duomenu baze tuscia.',' Warning ')
end
else
    warndlg('Iejimo atvaizdas turi buti pasirinktas.',' Warning ')
end
end
%-----
if chos==5,
    clc;
    close all;
    if (exist('rankos_database.dat')==2)
        button = questdlg('Ar tikrai norite pasalinti duomenu baze?');
        if strcmp(button,'Yes')
            delete('rankos_database.dat');
            msgbox('Duomenu baze sekmingai pasalinta.','Duomenu baze pasalinta','help');
        end
    else
        warndlg('Duomenu baze tuscia.',' Warning ')
    end
end
%-----
if chos==6,
    if (exist('rankos_database.dat')==2)
        load('rankos_database.dat','-mat');
        msgbox(strcat('Duomenu baze sudaro: ',num2str(rankos_nr),' paveikslelis(iu). Joje yra
',num2str(max_class-1),' klase(iu). Iejimo atvaizdai turi buti vienodo dydzio.','Duomenu bazes
rezultatas','help');
    else
        msgbox('Duomenu baze tuscia.','Duomenu bazes rezultatas','help');
    end
end
%-----
if chos==7,
    clc;
    VideoAcquire;
end
%-----
if chos==8,
    clc;
    prompt = {'Iveskite ribini kadra',};
    kadrai1 = inputdlg(prompt);

```

```

index1 = 1: str2double (kadrai);
%fileinfo = aviinfo('avi')
[namefile2,pathname2]=uigetfile('*.*','Pasirinkite video medziaga');
if namefile2~=0
    mov = aviread(strcat(pathname2,namefile2),index1);
else
    warnldg('Turite pasirinkti video medziaga',' Warning ')
end
disp(' ')
disp('rodomas filmas')
prompt = {'Iveskite perziurejimo greiti',};
greitis = inputdlg(prompt);
index1 = 1: (str2double(greitis));
movie(mov,1,index1);
disp('filmo pabaiga')
end
%-----
if chos==9,
    clc;
    close all;
    helpwin rankos_atpazinimas;
end
%-----
if chos==10,

load('matlab1','levelred','levelgreen','levelblue','hsv_colorlimits_sk','rgb_colorlimits_sk','failopav','keliovardas','
hsv_colorlimits','rgb_colorlimits'); %nereikia kai daroma nuo pradziu,kintamieji uzkraunami
end
%-----
if chos==11,

save('matlab1','levelred','levelgreen','levelblue','hsv_colorlimits_sk','rgb_colorlimits_sk','failopav','keliovardas','
hsv_colorlimits','rgb_colorlimits'); %nereikia kai daroma nuo pradziu,kintamieji uzkraunami
end
%-----
end
clc;
clear;
close all;

```

### **Kalibravimo modulis:**

```

function
[levelred,levelgreen,levelblue,hsv_colorlimits_sk,rgb_colorlimits_sk,failopav,keliovardas,hsv_colorlimits,rgb_
colorlimits]=kalib_ranka();
close all
echo off
prompt = {'Iveskite adorojama kadru skaiciaus ribini kadra',};
kadrai = inputdlg(prompt);
index = 1: str2double (kadrai);
%fileinfo = aviinfo('E:\matlab\apdirbama medziaga\vs\captureASAF.avi')
[namefile,pathname]=uigetfile('*.*','Pasirinkite video medziaga');
if namefile~=0
    mov = aviread(strcat(pathname,namefile),index);
else
    warnldg('Turite pasirinkti video medziaga',' Warning ')
end
prompt = {'Issirinkite pav. - iveskite kadro skaiciu.',};
answer = inputdlg(prompt);
pav1 = aviread(strcat(pathname,namefile), str2double(answer));
pav = im2double(pav1.cdata);
imwrite(pav,'pav.bmp');

```

```

%pagrindo isrinkimas
[failopav,keliovardas]=uigetfile('*.*','Pasirinkite pagrindo medziaga');
    if failopav~=0
        backgnd = aviread(strcat(keliovardas,failopav),(1));
    else
        warndlg('Turite pasirinkti video medziaga',' Warning ')
    end
backgnd_d = im2double(backgnd.cdata);
imwrite(backgnd_d,'backgnd.bmp');
%*****%
figure (1)
image (pav);
title ('pagrindas')
I1 = imabsdiff(pav,backgnd_d);
figure (2)
image(I1)
title ('skirtumas')
levelblue=graythresh(I1(:, :,3))
levelgreen=graythresh(I1(:, :,2))
levelred=graythresh(I1(:, :, 1))
% gaunama raudona kauke
i=1;
while i==1
prompt = {'Ivekite red leveli'};
title1 = 'Suteikiamos leveliu reiksmes';
lines = 1;
level1=num2str(levelred);
def = {level1};
answer = inputdlg(prompt,title1,lines,def);
levelred=str2double(answer{1});
bw_red = im2bw(I1(:, :,3),levelred);
bw_red = medfilt2(bw_red,[7 7]);
figure (3)
image (bw_red);
title ('bw red')
pause
button = questdlg('Ar gaunamas ryskus rankos plotas?','Rezultatai','Taip','Ne','Ne');
switch (button)
    case {'Ne'}
        i=1;
    otherwise
        i=2;
end
end
% gaunama zalia kauke
i=1;
while i==1
prompt = {'Ivekite green leveli'};
title1 = 'Suteikiamos leveliu reiksmes';
lines = 1;
level1=num2str(levelgreen);
def = {level1};
answer = inputdlg(prompt,title1,lines,def);
levelgreen=str2double(answer{1});
bw_green = im2bw(I1(:, :,2),levelgreen);
bw_green = medfilt2(bw_green,[7 7]);
figure (3)
image (bw_green);
title ('bw green')
pause
button = questdlg('Ar gaunamas ryskus rankos plotas?','Rezultatai','Taip','Ne','Ne');
switch (button)
    case {'Ne'}

```

```

        i=1;
    otherwise
        i=2;
end
end
% gaunama melyna kauke
i=1;
while i==1
prompt = {'Ivekite blue leveli'};
title1 = 'Suteikiamos leveliu reiksmes';
lines = 1;
level1=num2str(levelblue);
def = {level1};
answer = inputdlg(prompt,title1,lines,def);
levelblue=str2double(answer{1});
bw_blue = im2bw(I1(:, :, 3),levelblue);
bw_blue = medfilt2(bw_blue,[7 7]);
figure (3)
image (bw_blue);
title ('bw blue')
pause
button = questdlg('Ar gaunamas ryskus rankos plotas?','Rezultatai','Taip','Ne','Ne');
switch (button)
    case {'Ne'}
        i=1;
    otherwise
        i=2;
end
end
%*****sudedamos kaukes*****%
tmp=imadd(bw_blue, bw_red);
bw = im2bw(tmp, 0.9);
bw = imadd(bw,bw_green);
bw = im2bw(bw, 0.9);
figure (5)
image (bw);
title ('Ka gaunam bw')
bw= bwmorph(bw,'open');
figure (6)
image (bw);
bw_d = im2double(bw);
bw_d(:, :, 2) = bw_d(:, :, 1);
bw_d(:, :, 3) = bw_d(:, :, 1);
bw_d= imfill(bw_d,'holes'); %uzpildomos skyles
pic3_d = immultiply(bw_d,pav);
figure
image (pic3_d);
title('isskirtas')
%*****iskyrimas pagal pasirinktos rankos ploto reiksmes *****
%*****
pic_d=im2double(pic3_d);
pav = im2double(pic3_d);
%Isskiriame ploteli rankos rgb reiksmes randamos
plotas = imcrop;
% imwrite(plotas,'plotas.jpg')
figure(2)
image (plotas);
title ('issirinktas plotelis')
[rows, cols, depth] = size(plotas);
R = plotas (:, :, 1);
G = plotas (:, :, 2);
B = plotas (:, :, 3);
%R matrica

```

```

MinofR= min(min(R));           %randame vektoriu kuri sudaro R keikvieno stulpelio (ar eilutes) minimumai
MaxofR= max(max(R));
%G matrica
MinofG = min(min(G));         %randame vektoriu kuri sudaro g keikvieno stulpelio (ar eilutes) minimumai
MaxofG = max(max(G));
%B matrica
MinofB = min(min(B));         %randame vektoriu kuri sudaro B keikvieno stulpelio (ar eilutes) minimumai
MaxofB = max(max(B));
%Isskiriame ploteli rankos hsv reiksmes randamos
HSVpav = im2double(plotas);
HSVplotas = rgb2hsv(HSVpav);
[rows1, cols1, depth1] = size(HSVplotas);
H = HSVplotas ( :,:,1);
S = HSVplotas ( :,:,2);
V = HSVplotas ( :,:,3);
%H matrica
MinofH= min(min(H));         %randame vektoriu kuri sudaro H keikvieno stulpelio (ar eilutes) minimumai
MaxofH= max(max(H));
%S matrica
MinofS = min(min(S));         %randame vektoriu kuri sudaro S keikvieno stulpelio (ar eilutes) minimumai
MaxofS = max(max(S));
%V matrica
MinofV =min(min(V)) ;        %randame vektoriu kuri sudaro V keikvieno stulpelio (ar eilutes) minimumai
MaxofV = max(max(V));
%Maskuojam pav pagal plotelio reiksmes
min1 = 1;
max1 = 2;

hsv_colorlimits = [MinofH MinofS MinofV;...
                  MaxofH MaxofS MaxofV];
rgb_colorlimits = [MinofR MinofG MinofB;...
                  MaxofR MaxofG MaxofB];

picsize = size(pav);
%pastebejimas: iejimas turi buti double tipo

% Maskuojam pagal RGB reiksmes
pic = abs(pic3_d-0.0001);

red = im2bw(pic(:,:,1), rgb_colorlimits(min1,1));
redup = im2bw(pic(:,:,1), rgb_colorlimits(max1,1));
redup=~(redup);
red = immultiply(red,redup);

green = im2bw(pic(:,:,2), rgb_colorlimits(min1,2));
greenup = im2bw(pic(:,:,2), rgb_colorlimits(max1,2));
greenup=~(greenup);
green = immultiply(green,greenup);

blue = im2bw(pic(:,:,3), rgb_colorlimits(min1,3));
blueup = im2bw(pic(:,:,3), rgb_colorlimits(max1,3));
blueup=~(blueup);
blue = immultiply(blue,blueup);

gluf1a = immultiply(red,green);
gluf1=immultiply(gluf1a,blue);
% Maskuojam pagal HSV reiksmes
pic = rgb2hsv(pic3_d)-0.0001;
imwrite (pic,'rgbtohsv.bmp');

red = im2bw(pic(:,:,1), hsv_colorlimits(min1,1));
redup = im2bw(pic(:,:,1), hsv_colorlimits(max1,1));

```



```

redup=~(redup);
red = immultiply(red,redup);

green = im2bw(pic(:,:,2), hsv_colorlimits(min1,2));
greenup = im2bw(pic(:,:,2), hsv_colorlimits(max1,2));
greenup=~(greenup);
green = immultiply(green,greenup);

blue = im2bw(pic(:,:,3), hsv_colorlimits(min1,3));
blueup = im2bw(pic(:,:,3), hsv_colorlimits(max1,3));
blueup=~(blueup);
blue = immultiply(blue,blueup);

gluf2a = immultiply (red,green);
gluf2 = immultiply(gluf2a,blue);

%Susedam kaukes BITWISE gluf1 gluf2.
gluf = imadd(gluf1, gluf2);
gluf = im2bw(gluf, 0.9);
gluf= imfill(gluf,'holes');
figure(3)
image(gluf);
title ('sudeti filtrai')
imwrite (gluf,'gluf.bmp');
gluf2= bwmorph(gluf,'open');
imwrite (gluf2,'gluf2.bmp');
r111=pav(:,:,1);
r222=pav(:,:,2);
r333=pav(:,:,3);
galutinis1=(double(gluf2).*r111);
galutinis2=(double(gluf2).*r222);
galutinis3=(double(gluf2).*r333);

galas (,:,1)=galutinis1;
galas (,:,2)=galutinis2;
galas (,:,3)=galutinis3;

figure
image (galas);
title('galutinis su bwmorph')
imwrite (galas,'galas.bmp');
galutinis11=(double(gluf).*r111);
galutinis21=(double(gluf).*r222);
galutinis31=(double(gluf).*r333);

galas1 (,:,1)=galutinis11;
galas1 (,:,2)=galutinis21;
galas1 (,:,3)=galutinis31;
figure
image (galas1);
title('galutinis be morph')
imwrite (galas1,'galas1.bmp');
pause
[hsv_colorlimits_sk,rgb_colorlimits_sk]=kalib_raist(pic3_d);

```

### **Atvaizdo apdorojimo modulis:**

Function [galas]=apdoroj1(levelred,levelgreen,levelblue,hsv\_colorlimits\_sk,rgb\_colorlimits\_sk,

```

failopav,keliovardas,hsv_colorlimits,rgb_colorlimits);
close all
echo off
prompt = {'Iveskite adorojamu kadru skaiciaus ribini kadra',};
kadrai = inputdlg(prompt);
index = 1: str2double (kadrai);
%fileinfo = aviinfo('E:\matlab\apdirbama medziaga\vs\lcaptureASAF.avi')
[namefile,pathname]=uigetfile('*.*','Pasirinkite video medziaga');
    if namefile~=0
        mov = aviread(strcat(pathname,namefile),index);
    else
        warndlg('Turite pasirinkti video medziaga',' Warning ')
    end
prompt = {'Issirinkite pav. - iveskite kadro skaiciu:.'};
answer = inputdlg(prompt);
pav1 = aviread(strcat(pathname,namefile), str2double(answer));
pav = im2double(pav1.cdata);
%imwrite(pav,'pav.bmp');
%pagrindo isrinkimas
    if failopav~=0
        backgnd = aviread(strcat(keliovardas,failopav),(1));
    else
        warndlg('Nera pagrindo medziagos',' Warning ')
    end
backgnd_d = im2double(backgnd.cdata);
%imwrite(backgnd_d,'backgnd.bmp');
%*****%
figure (1)
image (pav);
title ('pav.')
l1 = imabsdiff(pav,backgnd_d);
figure (2)
image(l1)
title ('skirtumas')
%raudona kauke gaunama
bw_red = im2bw(l1(:,3),levelred);
bw_red = medfilt2(bw_red,[7 7]);
%zalia kauke gaunama
bw_green = im2bw(l1(:,2),levelgreen);
bw_green = medfilt2(bw_green,[7 7]);
%melyna kauke gaunama
bw_blue = im2bw(l1(:,3),levelblue);
bw_blue = medfilt2(bw_blue,[7 7]);
%*****sudedamos kaukes*****%
tmp=imadd(bw_blue, bw_red);
bw = im2bw(tmp, 0.9);
bw = imadd(bw,bw_green);
bw = im2bw(bw, 0.9);
% figure (5)
% image (bw);
% title ('Ka gaunam bw')
bw= bwmorph(bw,'open');
bw_d = im2double(bw);
bw_d(:,2) = bw_d(:,1);
bw_d(:,3) = bw_d(:,1);
bw_d= imfill(bw_d,'holes'); %uzpildomos skyles
pic3_d = immultiply(bw_d,pav);
%*****%
pav=pic3_d;
%juostos radimas pagal hsv ir rgb
picsize = size(pav);
%pastebejimas: iejimas turi buti double tipo
min=1;

```

```

max=2;
% Maskuojam juosta pagal RGB reiksmes
pic = abs(pic3_d-0.0001);

red = im2bw(pic(:,:,1), rgb_colorlimits_sk(min,1));
redup = im2bw(pic(:,:,1), rgb_colorlimits_sk(max,1));
redup=~(redup);
red = immultiply(red,redup);

green = im2bw(pic(:,:,2), rgb_colorlimits_sk(min,2));
greenup = im2bw(pic(:,:,2), rgb_colorlimits_sk(max,2));
greenup=~(greenup);
green = immultiply(green,greenup);

blue = im2bw(pic(:,:,3), rgb_colorlimits_sk(min,3));
blueup = im2bw(pic(:,:,3), rgb_colorlimits_sk(max,3));
blueup=~(blueup);
blue = immultiply(blue,blueup);

gluf1a = immultiply(red,green);
gluf1=immultiply(gluf1a,blue);
% Maskuojam pagal HSV reiksmes
pic = rgb2hsv(pic3_d)-0.0001;
% imwrite (pic,'rgbtohsv.bmp');

red = im2bw(pic(:,:,1), hsv_colorlimits_sk(min,1));
redup = im2bw(pic(:,:,1), hsv_colorlimits_sk(max,1));
redup=~(redup);
red = immultiply(red,redup);

green = im2bw(pic(:,:,2), hsv_colorlimits_sk(min,2));
greenup = im2bw(pic(:,:,2), hsv_colorlimits_sk(max,2));
greenup=~(greenup);
green = immultiply(green,greenup);

blue = im2bw(pic(:,:,3), hsv_colorlimits_sk(min,3));
blueup = im2bw(pic(:,:,3), hsv_colorlimits_sk(max,3));
blueup=~(blueup);
blue = immultiply(blue,blueup);

gluf2a = immultiply (red,green);
gluf2 = immultiply(gluf2a,blue);

%Susedam kaukes BITWISE gluf1 gluf2.
gluf = imadd(gluf1, gluf2);
gluf = im2bw(gluf, 0.9);
gluf= bwmorph(gluf,'open');
gluf= imfill(gluf,'holes');
%randame didziausia regiona
[label,num] = bwlabel(gluf,8);
    if num==1
        didz = gluf;
    end
regiondata = regionprops(label,'Area');
picArea = regiondata(num).Area;
while num~=0
    if picArea < regiondata(num).Area
        picArea=regiondata(num).Area
    end
    num=num-1;
end

```

```

% maxarea = max([regiondata.Area]); %
Ind = find([regiondata.Area]==picArea); %
didz = ismember(label,Ind); %
imshow(didz);
%pause
didz = ~ismember(label,Ind); %galutiniam rezultatui juostos kauke
%padidinam juostos kauke, kad panaikinti seselius is krastu
% i=im2bw(i);
% [label,num]=bwlabel(i,8);
% num
label=double(~didz);
stats=regionprops(label,'all');
stats.Orientation %sio kampo reikes atsukimui
figure
imshow(label)
title ('origin')
hold on
x=stats.Centroid(1)
y=stats.Centroid(2)
plot(x,y,'r*');
hold off
[rows,cols,depth]=size(label);
eil=cols/2
stulp=rows/2
skirt1=round(eil-stats.Centroid(1))
skirt2=round(stulp-stats.Centroid(2))
B = circshift(label, [skirt2, skirt1]);
figure
imshow(B)
title ('perstumtas')
hold on
x=stats.Centroid(1);
y=stats.Centroid(2);
plot(x,y,'r*');
B = imrotate(B,-stats.Orientation,'crop');
figure
hold on
imshow(B)
title ('pasuktas');
B=double(B);
stats1=regionprops(B,'all');
stats1.Orientation;
x=stats1.Centroid(1);
y=stats1.Centroid(2);
% pll=(stats1.MajorAxisLength/2+stats1.MajorAxisLength*0.10+x);
% pll1=abs(stats1.MajorAxisLength/2+stats1.MajorAxisLength*0.10-x);
% lin=line([x:pll],x,'y',y,'LineStyle','-');
% lin1=line([pll1:x],x,'y',y,'LineStyle','-');
hold off
B1= circshift(B, [0, 40]);
B2= circshift(B, [0, -40]);
B=imadd(B,B1);
B=imadd(B,B2);
B = im2bw(B, 0.9);
B = imrotate(B,stats.Orientation,'crop');
skirt1=round(eil-stats.Centroid(1));
skirt2=round(stulp-stats.Centroid(2));
B = circshift(B, [-skirt2, -skirt1]);
figure
imshow (B)
% whos B;
didz=~double(B);
% Maskuojam ranka pagal RGB reiksmes

```

```

pic = abs(pic3_d-0.0001);

red = im2bw(pic(:,:,1), rgb_colorlimits(min,1));
redup = im2bw(pic(:,:,1), rgb_colorlimits(max,1));
redup=~(redup);
red = immultiply(red,redup);

green = im2bw(pic(:,:,2), rgb_colorlimits(min,2));
greenup = im2bw(pic(:,:,2), rgb_colorlimits(max,2));
greenup=~(greenup);
green = immultiply(green,greenup);

blue = im2bw(pic(:,:,3), rgb_colorlimits(min,3));
blueup = im2bw(pic(:,:,3), rgb_colorlimits(max,3));
blueup=~(blueup);
blue = immultiply(blue,blueup);
gluf1a = immultiply(red,green);
gluf1=immultiply(gluf1a,blue);
% Maskuojam pagal HSV reiksmes
pic = rgb2hsv(pic3_d)-0.0001;
% imwrite (pic,'rgbtoshv.bmp');
red = im2bw(pic(:,:,1), hsv_colorlimits(min,1));
redup = im2bw(pic(:,:,1), hsv_colorlimits(max,1));
redup=~(redup);
red = immultiply(red,redup);
green = im2bw(pic(:,:,2), hsv_colorlimits(min,2));
greenup = im2bw(pic(:,:,2), hsv_colorlimits(max,2));
greenup=~(greenup);
green = immultiply(green,greenup);
blue = im2bw(pic(:,:,3), hsv_colorlimits(min,3));
blueup = im2bw(pic(:,:,3), hsv_colorlimits(max,3));
blueup=~(blueup);
blue = immultiply(blue,blueup);
gluf2a = immultiply (red,green);
gluf2 = immultiply(gluf2a,blue);
% Susedam kaukes BITWISE gluf1 gluf2.
gluf = imadd(gluf1, gluf2);
gluf = im2bw(gluf, 0.9);
gluf= bwmorph(gluf,'open');
gluf= imfill(gluf,'holes');
% randame didziausia regiona
[label,num] = bwlabel(gluf,8);
    if num==1
        maksimalus = gluf;
    end
regiondata = regionprops(label,'Area');
picArea = regiondata(num).Area;
while num~=0
    if picArea < regiondata(num).Area
        picArea=regiondata(num).Area;
    end
num=num-1;
end%
Ind = find([regiondata.Area]==picArea); %
maksimalus = ismember(label,Ind); % galutiniam rezultatui rankos kauke
filtras=immultiply(maksimalus,didz);
[label,num] = bwlabel(filtras,8);
regiondata= regionprops(label,'BoundingBox','Area');
% jei daugiau nei vienas objektas paveikslelyje
picArea =0;
while num~=0
    if picArea < regiondata(num).Area

```

```

        picArea=regiondata(num).Area;
        numeris=num;
    end
num=num-1;
end
picBoundingBox=zeros(size(regiondata(numeris).BoundingBox));
picBoundingBox=regiondata(numeris).BoundingBox;
%*****88
%sumazinimas - resize
width = picBoundingBox(3);
box_x1 = floor(picBoundingBox(1));
if(~box_x1)
    box_x1=1;
    width = width -1;
end
height = picBoundingBox(4);
box_y1 = floor(picBoundingBox(2));
if (~box_y1)
    box_y1=1;
    height = height -1;
end
r=pav(:,:,1);
g=pav(:,:,2);
b=pav(:,:,3);
r = immultiply(r(box_y1:box_y1+height , box_x1:box_x1+width,1),filtras(box_y1:box_y1+height ,
box_x1:box_x1+width));
g = immultiply(g(box_y1:box_y1+height , box_x1:box_x1+width,1),filtras(box_y1:box_y1+height ,
box_x1:box_x1+width));
b = immultiply(b(box_y1:box_y1+height , box_x1:box_x1+width,1),filtras(box_y1:box_y1+height ,
box_x1:box_x1+width));
%imshow(filtras)
%grazinimas rgb spalvu
% pav(:,:,1)=immultiply(pav(:,:,1),filtras);
% pav(:,:,2)=immultiply(pav(:,:,2),filtras);
% pav(:,:,3)=immultiply(pav(:,:,3),filtras);
[eilutes,stulp]=size(r);
img=zeros(eilutes,stulp,3);
img(:,:,1)=r;
img(:,:,2)=g;
img(:,:,3)=b;
figure
imshow(img)
%pause
grey=uint8(.2989 * double(pav1.cdata(:,:,1)) + .5870 * double(pav1.cdata(:,:,2)) + .1140 *
double(pav1.cdata(:,:,3))); % calculate the monochrome luminance (i.e. greyscale) by
% combining the RGB values according to the
% NTSC standard, which applies coefficients
% related to the eye's sensitivity to RGB colors.
grey= immultiply(grey(box_y1:box_y1+height , box_x1:box_x1+width,1),filtras(box_y1:box_y1+height ,
box_x1:box_x1+width));
grey= IMRESIZE(grey,[180 180]);
galas=grey;
pause (3)
close all

```

### Video gavimo modulis:

```

function [video]=VideoAcquire();
imaqhwinfo % info apie galimus prietaisus
imaqreset %resetuoja hardware
imaqhwinfo
%dev_info = imaqhwinfo('winvideo',1) %info apie pasirinkta hardware
%celldisp(dev_info.SupportedFormats) %palaikomi formatai

```

```

prompt = {'Iveskite irasomos medziagos pavadinima su .avi galune',};
    atsakymas = inputdlg(prompt);
    atsakymas=char(atsakymas);
aviobj = avifile(atsakymas,'Compression','none');
%sukurti faila video loginimui, out = avifile('outputAVI1.avi','Compression','none');
%reikia ji uzdaryti kad butu galima naudotis ne matlabo viduje
aviobj % imanomi parametrai
aviobj.fps=5 %kiek freimu per sekunde
%aviobj.Quality =100; %galima padaryti kad logintu mazesnios apimties
%failus, jei failas nekompresintas neturi itakos
vid = videoinput('winvideo',1); %video objekto sukurimas
%get(vid,'Source'); %video saltiniai,chaneliai galima adapteriui,
%galima rankiniu budu parinkti: vid.SelectedSourceName = 'CH1';
%get(vid) %parodo visus nustatymus video objektui
%get(getselectedsource(vid)) %apie filmavimo buda
vid.LoggingMode = 'disk&memory'; %&memory
vid.DiskLogger = aviobj;
% veikia=isrunning(vid)
% irasineja=islogging(vid)
isrunning(vid)
triggerconfig(vid, 'immediate')
kartojimas = 2;
set(vid, 'TriggerRepeat', kartojimas)
framesPerTrigger = 5;
set(vid, 'FramesPerTrigger', framesPerTrigger) %FramesAcquired = FramesPerTrigger * (TriggerRepeat + 1)
vid.ReturnedColorSpace %kokia naudojama palete RGB ar YUV, galima nustatyti kad grazintu pilka:
set(vid, 'ReturnedColorSpace', 'grayscale')
%get(vid, 'VideoFormat') %koku formatu irasines NTSC RGB
%get(vid, 'VideoResolution') %kokia rezoliucija 640x460
%vid.NumberOfBands
%%vid.FramesPerTrigger =200; %freimu daznis kart FramesPerTrigger = laikas kuri turetu filmuoti
%galima kad kas kelinta kadra imtu su: FrameGrabInterval
triggerconfig(vid) %kokie trigerio nustatymai parodo
pause
get(vid)
start(vid)
isrunning(vid)
islogging(vid)
%frame = getsnapshot(vid);
Preview(vid)
%frame = getsnapshot(vid);
%size(frame) %patikrinimas koko dydzio freimas irasytas
%class(frame) %kokia klase irasinejama pav.: unit8, galima nustayti kad iskarto double tipo irasinetu
%trigger(vid) %paleidzia iraso, pagal uzduotas salygas
wait(vid,1000) %
%%getdata (vid,100);
vid.DiskLoggerFrameCount %kiek freimu irase i diska
%imaqmontage (vid.UserData)
%close(vid.DiskLogger); %uzdaro loginima i diska
closepreview(vid)
%[data time] = getdata(vid,50); %Bring the acquired data into the workspace
%elapsed_time = time(50) - time(1) %praejes laikas
aviobj=close (aviobj);
delete(vid)
clear vid
imaqfind %patikrinimas ar neliko video irasyimo objektu, ar viskas isvalyta
delete(imaqfind)

```