

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

Egidijus Kubickas

KALBANČIOJO LŪPŲ FORMOS REGISTRAVIMAS
Magistro darbas

Vadovas

doc. dr. G. Daunys

ŠIAULIAI, 2007

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

TVIRTINU

Katedros vedėjas

doc. dr. G. Daunys

2007 06

KALBANČIOJO LŪPŲ FORMOS REGISTRAVIMAS
Magistro darbas

Vadovas

doc. dr. G. Daunys

2007 06

Atliko

RM-5 gr. stud.

E. Kubickas

2007 06 08

Recenzentas

ŠU Technologijos fakulteto

asist. D. Dervinis

2007 06

ŠIAULIAI, 2007

SUMMARY

Kubickas E. Speaker lip shape registration: Master thesis of electronics engineer/research advisor Assoc. Dr. G. Daunys; Šiaulių University, Technological Faculty, Electronics Department. – Šiauliai, 2007. – 67 p.

The theme of Master Project of Electronics engineer is actual, because the lip is the one of the principal visual speech part. For better users instruction's understanding in the noisy environment, computer must rely on the visual users signs. In the theory section there is described the most important channels for communication - speech and visual signs.

There is performing lip shape registration by using OpenCv and Matlab 7.0 package in this work. The approach proposed to make and to analyze algorithm for more accurate lip registration. The main task is to investigate how to detect speaker lip shape robustly, when their location is different. There is presented research of four people lip's shape (contours). The results showed that there are few problems, related with lip registration – difficult lip extraction because of the similar skin colour, problematic lips inner contours extraction, because of the visible articulators such as teeth, tongue. Above all, that the main task was reached – there was received the robust lip registration by green colours component.

Received results can be used for the further investigations such as different expression pictures research, the neural network for lip position or letters recognizing training.

TURINYS

IŽANGA.....	8
1. VIZUALINĖ KALBA.....	9
1.1. Veido atpažinimas.....	10
1.1.1. Veido elementų ir bruožų, kurie būdingi veido atvaizdavimui, aptikimas.....	10
1.1.2. Veido atpažinimo metodai.....	11
1.1.2.1. Pagrindinių komponentų metodas (PCA).....	11
1.1.2.2. Tikrų veidų metodas (Eigenfaces).....	14
1.1.2.3. Fisher veidų metodas (Fisherfaces).....	15
1.1.2.4. Faktorių analizė (FA, factor analysis).....	16
1.2. Lūpos – svarbiausias vizualinės kalbos bruožas.....	16
1.2.1. Bendri (globaliniai) vizualiniai bruožai.....	17
1.2.2. Lokaliniai vizualiniai bruožai.....	18
2. SKAITYMAS IŠ LŪPŲ.....	20
2.1. Lūpų kontūrai.....	21
2.2. Lūpų segmentacija naudojant rankinę intervenciją.....	21
2.3. Pusiau automatinis lūpų segmentavimas.....	22
2.4. Lūpų bruožų išskyrimas.....	22
2.4.1. Lūpų bruožų išskyrimas, naudojant spalvų segmentaciją.....	24
2.4.2. Vizualinio modelio (VM) bruožų išskyrimas.....	25
2.4.3. Dinaminiai lūpų bruožai.....	27
2.4.4. Lūpų parašo vizualinė interpretacija.....	28
2.5. Kalbėtojo modeliavimas.....	30
2.5.1. Nuo teksto priklausantis testas (TD).....	31
2.5.2. Nuo teksto nepriklausantis testas (TI).....	32
3. AUDIOVIZUALINĖ KALBA.....	34
3.1. Burnos aptikimas ir žymėjimas.....	35
3.2. Audiovizualinis modelis.....	36
4. DARBE TAIKYTI METODAI.....	38
4.1. Plėtimas (dilate).....	38
4.2. Erozija.....	39
4.3. Dviejų dimensijų Gauso funkcija.....	40
4.4. Dvimatė diskretinė Furjė transformacija.....	40
4.5. OpenCV. Adaboost algoritmas.....	41

4.5.1. Haar kaskadų klasifikatoriai.....	42
4.5.2. Dirbtinis neuroninis tinklas.....	42
4.6. Praktinė dalis.....	44
IŠVADOS IR SIŪLYMAI.....	51
LITERATŪRA.....	52
PRIEDAI.....	54
1 PRIEDAS.....	54
2 PRIEDAS.....	63

LENTELIŲ SĄRAŠAS

1.1 lentelė. Veido atpažinimas.....	9
2.1 lentelė. Žmogaus identifikavimo testų tikslumas, nuo teksto priklausančio (TD) ir nuo teksto nepriklausančio (TI) modelių, naudojant formos ir intensyvumo parametrus.....	33
4.1 lentelė. Tiriamieji.....	45

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Pirmoji (PC1) ir antroji (PC2) pagrindinės komponentės.....	12
2.1 pav. Pirmi šeši pagrindiniai formos variantų modeliai, įvesti į apmokymų rinkinį iš visų subjektų ir viso žodžio sekos.....	24
2.2 pav. Atkurtos lūpos, naudojant vidutinę formą ir vidutinį profilio parametrus.....	24
2.3 pav. Lūpų išskyrimo procesas.....	25
2.4 pav. Išorinių lūpų kontūro perimetro apskaičiavimas.....	26
2.5 pav. Lūpų bruožų išskyrimo algoritmas.....	27
2.6 pav. LC-DCT komponentų lūpų parašo pavyzdžiai.....	29
2.7 pav. Apatinės DCT komponentės, atsižvelgiant į apatines ir viršutines lūpų ribas.....	30
2.8 pav. Eksperimento pavyzdys (12 kalbėtojų vaizdų pavyzdžiai, naudojami lūpų stebėjimo rezultatams).....	31
3.1 pav. Burnos aptikimo ir stebėjimo sistema.....	35
3.2 pav. Audiovizualinis keleto HMM modelis.....	37
4.1 pav. Plėtimas.....	38
4.2 pav. Eroziija.....	39
4.3 pav. Gauso filtras.....	40
4.4 pav. Realus periodiškai pratęstas vaizdas.....	41
4.5 pav. Dažniausi Haar bruožai.....	42
4.6 pav. Dirbtinio neurono schema.....	43
4.7 pav. Lūpų formos registravimo algoritmas.....	44
4.8 pav. Lūpų aptikimas.....	46
4.9 pav. Lūpų kontūro aptikimas su klaidomis.....	47
4.10 pav. Lūpos po filtravimo ir normavimo (G).....	47
4.11 pav. Lūpų registravimas pagal atskiras spalvų komponentes (RGB).....	48
4.12 pav. Lūpos po komporavimo (sulyginimo) (G).....	48
4.13 pav. Lūpų formos kontūrai (G).....	49
4.14 pav. Atrinkti lūpų kontūro taškai (G).....	50

ĮŽANGA

Darbo tikslas – sudaryti ir įvertinti algoritmą kalbančiojo lūpų formos registravimui.

Darbo uždaviniai

- Susipažinti ir aprašyti esamus veido ir lūpų aptikimo metodus.
- Išsiaiškinti ar įmanoma naudojant OpenCV tiksliai aptikti lūpų sritį.
- Sukurti algoritmą tiksliam kalbančiojo lūpų formos registravimui.
- Matlab 7.0 programiniu paketu iširti kaip tiksliai aptinkama kalbančiojo lūpų forma jai keičiantis.

Žmogaus kalba yra dvirūšė iš prigimties: audio ir vizualinė. Audio kalba perduodama per akustinius kalbėtojo signalus, vizualinė kalba perduodama per kalbėtojo lūpų, liežuvio bei kitų veido raumenų judesius. Toks dvirūšiškumas turi du aspektus: atsiradimą ir supratimą. Kalba atsiranda dėl balso stygų vibravimo bei balso trakto formos, kuris susidaro iš artikuliacinių organų, tokių kaip nosies ertmė, liežuvis, dantys ir lūpos. Naudodamas šiuos artikuliacinius organus kartu su raumenimis, kurie sudaro veido išraišką, kalbėtojas kalba. Dėl to, kad kai kurios iš šių artikuliacijų yra matomos, atsiranda neatskiriamas ryšys tarp akustinės ir vizualinės kalbos.

Kompiuterio naudojimo produktyvumas, atpažįstant kalbą, ribojamas ne dėl kompiuterio galios, tačiau dėl vartotojo sąveikos su kompiuteriu. 1950 metais, sąveika dažniausiai buvo pagrįsta jungiklių perjungimu bei perfokortomis. 1970 metais buvo sukurtas komandinės eilutės interfeisas (GUI), kurio pagalba vartotojas gali duoti instrukcijas kompiuteriui, renkant tekstą. 1980 metais sukurtas grafinis vartotojo interfeisas, nuo tada produktyvumas ženkliai padidėjo. Aišku, GUI sąveika dar labai skiriasi nuo žmonių sąveikos su vienas kitu. Žmogaus-žmogui komunikacijoje, svarbiausi informacijos kanalai yra kalbos ir vizualiniai ženklai. Šitie du kanalai dažnai papildo vienas kitą tam, kad bendravimas taptų efektyvus. Taigi, patogus vartotojui kompiuteris turėtų sugebėti ir klausytis, ir matyti vartotoją – taip užtikrindamas audiovizualinį atgalinį ryšį vartotojui.

Tam, kad būtų sukurtas toks multimodalinis žmogaus-kompiuterio interfeisas, įtraukti daugelio sričių tyrinėjimai. Pirmiausia kompiuteris turi turėti tam tikrą vizualinę gebą vartotojo veido nustatymui. Norint kuo tiksliau suprasti vartotojo sakytines instrukcijas, net ir labai triukšmingoje aplinkoje, kompiuteriui gali reikėti pasikliauti ir vizualiniais vartotojo ženklais. Pats svarbiausias vizualinis kalbos elementas – lūpos, todėl yra labai svarbu tiksliai jas ištyrinėti, jos formą ir kitimą esant įvairioms sąlygoms (apšvietimui), kad būtų galima pagerinti vizualinės kalbos atpažinimą.

1. VIZUALINĖ KALBA

1.1. Veido atpažinimas

Žmogaus veido aptikimas vaizde dažniausiai yra pirmas žingsnis sprendžiant aukštesnio lygio problemas, pavyzdžiui, veido, veido mimikos atpažinimą.

Kompiuteriniame regėjime dažniausiai sutinkamos dvi veido aptikimo modifikacijos užduotys.

Veido lokalizavimas (face detection) ir *veido poslinkio sekimas (face tracking)*. Veido lokalizavimas yra supaprastintas veido aptikimo būdas, kadangi remiamasi tuo, kad vaizde yra vienas ir tik vienas veidas. Veido poslinkio sekimo uždavinį video sraute galima suformuluoti kaip veido lokalizavimą esamame kadre, remiantis jo padėtimi prieš tai buvusiame kadre.

1.1 lentelė

Veido atpažinimas

Uždavinys	Pradiniai duomenys	Rezultatas
Veido aptikimas	Vaizdas	Veido aptikimas ir padėties nustatymas
Veido lokalizavimas	Vaizdas (ar jo fragmentas), kuriame yra vienas veidas	Veido padėtis vaizde
Veido sekimas	Esamas video srauto kadras, veido padėtis praėjusiame kadre	Veido padėtis esamame video kadre

Veido aptikimo vaizde uždavinys yra gana paprastas žmogaus akiai (1.1 lentelė). Tačiau bandant sukurti automatinę vaizdo aptikimo sistemą, susiduriama su šiomis problemomis:

- Žymiai skiriasi skirtingų žmonių veidai.
- Netgi santykinai nedidelis veido pozicijos pasikeitimas kameros atžvilgiu įtakoja nemažus veido vaizdo pasikeitimus.
- Individualūs veido bruožai (ūsai, barzda, akiniai, apgamai ir t.t.).
- Veido mimikos pakeitimas labai įtakoja veidą vaizde.
- Dalis veido gali būti nematoma (tarkime užstoja kiti daiktai).
- Filmavimo sąlygos (apšvietimas, vaizdo iškraipymai, spalvinis kameros balansas) - visa tai turi įtakos gaunamam vaizdui.

Veido atvaizdas gali būti fiksuojamas skaitmenine kamera gerai apšviestame kambaryje. Priekinis vaizdas užfiksuojamas subjektams žiūrint į kamerą, tuo tarpu vaizdas iš šono turi būti fiksuojamas, laikant kamerą maždaug dešiniame kampe nuo veido. Rinkinys yra sudaromas iš optimalių veido priekinių bei veido šoninių bruožų, šitų bruožų reikšmės yra išsaugomos skirtinguose failuose. Vektoriai pagrindiniuose veido duomenų failuose vadinami M-vektoriais.

Tam, kad būtų sumažintas bruožų vektorių reikšmių kartu su jų subjekto indeksais apskaičiavimo sudėtingumas, jie sudaromi didėjančia tvarka. Testinių bruožų reikšmės yra įterpiamos į atitinkamas vietas išrūšiuotų bruožų sąrašė į kiekvieną bruožą (stulpelį). Testinių pavyzdžių vaizdų dvi artimiausios ribos yra identifikuojamos kiekviename stulpelyje ir yra įvertinamas jų panašumas. Veido sekimui algoritmas k-artimiausiai ribai yra pateikiamas žemiau [16].

1 žingsnis. Suklasifikuoti bruožų reikšmes kartu su jų priklausomais indeksais kiekviename stulpelyje didėjančia tvarka.

2 žingsnis. Įterpti kiekvieną testinį pavyzdinį bruožo vektorių į atitinkamą vietą kiekviename stulpelyje.

3 žingsnis. Nustatyti dvi artimiausias ribas kiekviename stulpelyje ir įvertinti testinių pavyzdžių panašumus tarp abiejų arčiausių ribų, kiekvienam suteikiant atitinkamą indeksą.

4 žingsnis. Įvertinti visus testinių pavyzdžių panašumus su visais atitinkamais indeksais.

5 žingsnis. Priskirti subjekto indeksą, turint maksimumą panašumų reikšmių testiniam veidui.

Šitas pavaizduotas algoritmas užtikrina puikius veido atpažinimo rezultatus [16].

1.1.1. Veido elementų ir bruožų, kurie būdingi veido atvaizdai, aptikimas

Briaunos yra žymūs ryškumo perėjimai. Briaunos dažniausiai atitinka objekto kontūrus atvaizde. Pasinaudojant šiuo faktu, galima veido atpažinimą atlikti briaunų (kraštų) žemėlapiu (atvaizdu, kuriame pavaizduoti žymūs ryškumo perėjimai) ir elipsės formos veidu [17]. Žymūs ryškumo perėjimai taip pat atitinka *veido bruožus (facial features)* – akies, antakių, burnos, nosies kraštams.

Ryškusumas. Vaizdo sritys, atitinkančios veido bruožus (burna ir t.t.) yra tamsesnės nei juos supanti oda. Pasinaudojant šia savybe yra sukurti algoritmai, kuriais apskaičiuojama lokalių sričių ryškumo minimumai [17]. Ryškumo minimumai yra laikomi charakteringais veido bruožais.

Spalva. Nepaisant to, kad ryškumas paprastai yra pagrindinis informacijos šaltinis daugelyje mašininio regėjimo užduočių, spalva (kuri turi papildomos informacijos apie objekto atspalvį) yra galingesnis įrankis objektų atpažinimui vaizde. Kaip parodė eksperimentai [17], skirtingų žmonių odos spalva užima palyginus nedidelį spalvinį diapazoną, netgi atsižvelgiant į rases. Pagrindiniai skirtumai yra dėl skirtingo ryškumo, o ne dėl spalvos atspalvio. Tai leidžia daryti išvadą, kad žmonių odos spalvos atspalvis yra labai panašus. Ir tai galima panaudoti veido atpažinimui.

1.1.2. Veido atpažinimo metodai

OpenCV (Open Source Computer Vision) – programinių funkcijų biblioteka, kuri yra daugiausia pritaikoma realaus laiko vaizdo apdorojimui kompiuteriu (kompiuterinis regėjimas).

OpenCV bibliotekos pavyzdys - *Žmogaus-kompiuterio sąveika (Human-Computer Interaction)*, objektų indentifikavimas, segmentavimas ir atpažinimas, gesto atpažinimas, judėjimo kelio sekimas, judėjimo supratimas, mobili robotika [1].

Veido atpažinimas yra labai aktyviai tyrinėjama sritis. Esmė - kaip atpažinti veidą, esantį paveiksliuke ar video medžiagoje. Veido atpažinimas yra susijęs su veido aptikimu. Veido aptikimas yra procesas, kuomet surandamas veidas paveiksliuke ar video medžiagoje, o veido atpažinimas - tai aptikto veido palyginimas su jau žinomais veidais.

Yra nemažai algoritmų, kurie atlieka veido atpažinimą:

1. *Pagrindinių komponentių analizė (Principal Component Analysis method (PCA))*.
2. *Tiesinis diskriminanto metodas (Linear Discriminant Analysis method)*.
3. Kernelio metodai.
4. 3D veido atpažinimo metodai.
5. *Paslėpti Markovo modeliai (Hidden Markov Models (HMM))*.

1.1.2.1. Pagrindinių komponentių metodas (PCA)

Paprasčiausias ir lengviausias metodas PCA yra palaikomas OpenCV biblioteka. Tačiau jis turi ir trūkumų. Jei bus norima atpažinti veidą su skirtingu fonu, tai bus padaryti sudėtinga.

Privalumas, kad metodas greitas ir nereikalauja daug atminties. PCA atlieka dimensijos mažinimą.

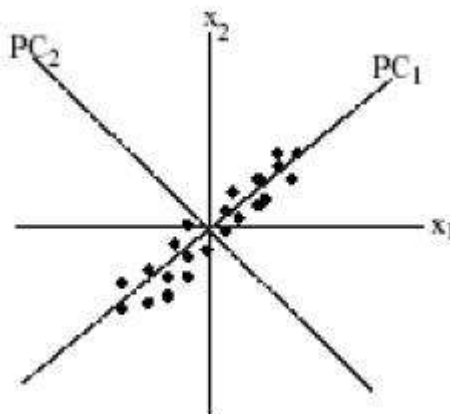
Būtina atlikti keletą pradinių apdirbimų prieš panaudojant PCA metodą veido atpažinimui [5]:

- Veido aptikimas ir išskyrimas: aptikti veidą naudojant OpenCV ir tuomet atpažinti.
- Normalizavimas.
- Maštavimas.

Norint patikrinti rezultatus, reikia pasinaudoti veidų duomenų bazėmis (*Yale Face Database, FERET database* ir t.t.)

Pagrindinių komponentių analizė (PCA) yra klasikinis statistinis metodas. Tai tiesinė transformacija plačiai naudojama duomenų analizėje kaip daugiamačių duomenų dimensijos mažinimo metodas. Metodo tikslas – rasti kryptį, kuria duomenų dispersija yra didžiausia [5]. Pagrindinė komponentės idėja – sumažinti duomenų dimensiją kiek galima geriau išlaikant duomenų dispersijas. Pagrindinės komponentės yra nekoreliuotos ir išdėstytos taip, kad pirmosios kelios turi didžiausias dispersijas. PCA padeda geriau vizualizuoti duomenis, tuo pačiu ir

palengvina duomenų suvokimą. Sukuriamas duomenų debesis ir pasukamas taip, kad matytųsi didžiausia dispersija. Didžiausią dispersiją turinti kryptis vadinama pirmąja pagrindine komponente. Ji eina per centrą ir minimizuoja kiekvieno taško kvadratinį atstumą iki šios linijos, t.y. ši linija yra kiek galima arčiau visų duomenų taškų (1.1 pav.). Antrosios pagrindinės komponentės ašis taip pat turi eiti per centrą ir ji turi būti nekoreliuota (ortogonali) pirmosios pagrindinės komponentės ašiai.



1.1 pav. Pirmoji (PC1) ir antroji (PC2) pagrindinės komponentės

Pagrindinės komponentių analizėje naudojamos sąvokos: duomenų matrica - Ξ , sudaryta iš vektorių $X^j = (x_1^j, x_2^j, \dots, x_n^j) = (x_i^j)$, $i = 1, \dots, n$, $j = 1, \dots, m$.

Matricoje Ξ stulpeliai žymi duomenų parametrus (jų iš viso yra n), o eilutės – daugiamačius vektorius (jų iš viso yra m).

Galima apskaičiuoti duomenų koreliacijos koeficientą. Koreliacija – tai dviejų reiškinių (dviejų požymių) tarpusavio sąryšis. Koreliacijos tarp k -tojo ir l -tojo vektorių-stulpelių $X_k = \{x_k^j\}$ ir $X_l = \{x_l^j\}$, $j = 1, \dots, m$, koeficientas skaičiuojamas pagal (1.1) formulę:

$$r_{kl} = \frac{\sum_{j=1}^m (x_k^j - \bar{x}_k)(x_l^j - \bar{x}_l)}{\sqrt{\sum_{j=1}^m (x_k^j - \bar{x}_k)^2 \sum_{j=1}^m (x_l^j - \bar{x}_l)^2}}, \quad (1.1)$$

čia $\bar{x}_k = \frac{1}{m} \sum_{j=1}^m x_k^j$, $\bar{x}_l = \frac{1}{m} \sum_{j=1}^m x_l^j$.

Galima suformuoti koreliacinę matricą $R = \{r_{kl}\}$.

Kovariacijos koeficientas yra skaičiuojamas pagal (1.2) formulę:

$$C_{kl} = \frac{1}{m-1} \sum_{j=1}^m (x_k^j - \bar{x}_k)(x_l^j - \bar{x}_l), \text{ kai } k \neq l, \quad (1.2)$$

Kai $k=l$, (1.2) formulė tampa dispersijos formule.

Iš kovariacijos koeficientų, gautų iš (1.2) formulės, galima suformuoti kovariacinę matricą $C = \{C_{kl}\}$. Kovariacinė matrica yra simetrinė matrica.

Iš (1.1) ir (1.2) formulių gaunama, kad koreliacijos koeficientas lygus $r_{kl} = \frac{C_{kl}}{\sqrt{C_{kk} C_{ll}}}$. Jei du

nekoreliuoti vektoriai, tai jų kovariacijos koeficientas $C_{kl} = C_{lk} = 0$, $k \neq l$.

Matricos tikras vektorius (*eigenvector*) e_k ir jį atitinkanti tikra reikšmė (*eigenvalue*) λ_k yra lygties $Ce_k = \lambda_k e_k$ sprendinys. C yra kovariacinė matrica; λ_k reikšmė randama iš charakteringos lygties $|C - \lambda_k I| = 0$. Čia I vienetinė matrica, kurios matmenys tokie pat, kaip matricos C . Ženklų $||$ apibrėžtas diskriminantas. Tikrų vektorių bus tiek, kiek kintamųjų, t.y. n , kadangi kovariacinės matricos rangas lygus n . Tikrų vektorių ir tikrų reikšmių radimas nėra trivialus uždavinys, tačiau yra sukurta nemažai šio uždavinio sprendimo metodų.

Tikras vektorius, susijęs su didžiausia nuosava reikšme, turi tokią kryptį kaip pirmoji pagrindinė komponentė. Antroji pagrindinė komponentė atitinka antrąjį tikrą vektorių ir t.t. Pagrindinių komponentių skaičius parenkamas atsižvelgiant į dimensijos skaičių projekcijoje. Yra sukurti efektyvūs algoritmai pagrindinėms komponentėms rasti. Tam gali būti taikomi ir dirbtiniai neuroniniai tinklai, taip pat savireguliuojantys neuroniniai tinklai.

Surūšiuojant tikrus vektorius e_k jų nuosavų reikšmių mažėjimo tvarka ($\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n$), galima sukurti ortogonalią bazę, kurioje pirmas nuosavas vektorius yra didžiausios duomenų dispersijos kryptis. Tikri vektoriai išreiškiami kaip pagrindinės komponentės. Tikros reikšmės nurodo visos dispersijos dalį. Tikros reikšmės λ_k - tai duomenų imties dispersija kryptimi e_k , o C_{kk} - duomenų imties dispersija kryptimi $x_k = (x_1^k, x_2^k, \dots, x_m^k)^T$.

$$1. \sum_{k=1}^n \lambda_k = \sum_{k=1}^n C_{kk};$$

$$2. \lambda_1 = \max_k \lambda_k \geq \max_k C_{kk};$$

$$3. \lambda_n = \min_k \lambda_k \geq \min_k C_{kk}$$

Pagrindinių komponentių analizėje kyla klausimas, kiek principinių komponentių pasirinkti. Yra kelios taisyklės. Viena jų: parinkti tokį mažiausią sveiką skaičių d , kad procentinis dydis nuo visos dispersijos būtų viršytas tam tikrą pasirinktą dydį. Išdėsčius tikrus vektorius jų tikrų reikšmių

mažėjimo tvarka ir imant d pirmųjų vektorių, galima apskaičiuoti, kokia paklaida bus daroma. Pagal leidžiamą paklaidą imama daugiau ar mažiau pagrindinių komponentų.

Kai kurie autoriai principinių komponentų analizėje vietoj kovariacinės matricos naudoja koreliacinę matricą. PCA gali būti naudojama ir klasterizavime. PCA metodas dažnai derinamas su kitais metodais.

1.1.2.2. Tikrų veidų metodas (eigenfaces)

Tikro veido (eigenface) reprezentacijos metodas veido atpažinimui yra pagrįstas PCA (*Pagrindinių komponentų analize*). Pagrindinė idėja yra išskaidyti veido atvaizdą į *tikrų veidų* rinkinį (t.y., mažų tipiškų bruožų vaizdų rinkinį), šitie bruožai yra pagrindiniai originalaus vaizdo komponentai [16]. Šitos *tikrų veidų* funkcijos, kaip ortogonalūs pagrindiniai vektoriai linijinės suberdvės, vadinami *face space (veido erdve)*. Veido atpažinimo strategija įtraukia naujo veido atvaizdo projektavimo į veido erdvę ir jo pozicijos palyginimą veido erdvėje su jau žinomais veidais. Pagal šitą metodą [16], mokymo rinkinyje veidų atvaizdų pavyzdžiai yra paverčiami į vektorių $M \times N \times K$, kur $M \times N$ yra atvaizdo dydis, o K mokymų pavyzdžių skaičius. Šita $M \times N$ dimensinė erdvė yra iš tikrųjų labai didelė, taigi ypač svarbu sumažinti erdvės dydį prieš bandant atpažinti veidą. PCA ir LDA (*linijinio diskriminanto analizė (linear discriminant analysis)*) yra du panašūs metodai šitos didelės dimensijos sumažinimui.

Pagrindinė PCA filosofija yra pažymėti visus $M \times N$ dimensinius veido pavyzdžius x_i $i = \{1, \dots, k\}$ į atskirą vektorių y_i , $i = \{1, \dots, n\}$ taigi šitas y_i pateikia x_i [13],

$$y_i = w^T x_i \quad (1.3)$$

čia kiekvienas x_i pavaizduoja veidą, o w yra svorio vektorius, kuris vaizduoja mastelio parinkimą.

Galutinė funkcija yra variantiškumo maksimizacija [13], t.y.

$$J = \max \sum_{i=1}^n (y_i - \bar{y})^2 \quad (1.4)$$

kur

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i. \quad (1.5)$$

Svorio vektorių w_1, \dots, w_k rinkinys atvaizduoja *tikrų* vektorių kovariacinę matricą, įvertintą iš rinkinio veidų pavyzdžių ir $k < MN$. Tai reiškia, kad yra pasirenkama tik keli *tikri* vektoriai, atitinkantys dominuojančias *tikras reikšmes* ir tokiu būdu mažinamas veido bruožų dimensionalumas. Tokiai tiksliai veido atpažinimo sistemai būtinas gana aukštas tikslumas ir, tuo

pačiu metu, proceso laikas tęstiniam veido atvaizdai turi būti trumpas. Atpažinimo sistema turi būti nekintama nei pasukus galvą, nei pakitus ryškumo intensyvumui.

1.1.2.3. Fisher veidų metodas (Fisherfaces)

Kita technika, žinoma, kaip *Fisher veidas* taip pat naudoja *tikrų veidų* linijinės projekcijos sekimą. *Fisher veido* metodas naudoja *Fisher linear discriminant function (Fisher linijinio diskriminanto funkcija)* (FLDF), kuri duoda tvirtą klasifikatorių ir sustiprina veido pavyzdžių klasifikatoriaus tikslumą. FLDF sukuria linijinę atskirtą erdvę, kuri yra veiksminga veido pavyzdžių diferenciacijai. FLDF nustato linijinių funkcijų skaičių, kurie suskirsto veido erdvę į k atskiras rūšis [2].

Naudojant *Fisher veido* būdą, mokymo veido pavyzdžiai įtraukia ne vien tik atskirą veidą, bet ir įvairias veido pavyzdžio ekspresijas. Jei i -tosios veido rūšies n_i pavyzdžiai skirtingų ekspresijų, o $m^{(i)}$ atvaizduoja bruožų vektorių iš n_i pavyzdžių i -tosios rūšies, vidurkį. Galutinė funkcija yra maksimizuoti šitą atvejį [2]:

$$J_F = \max \left\{ \frac{W^T S_B W}{W^T S_W W} \right\} \quad (1.6)$$

čia S_B yra tarp rūšies sklaidos matricos pateiktos

$$S_B = \sum_{i=1}^n n_i (m^{(i)} - m)(m^{(i)} - m)^T, \quad (1.7)$$

ir S_W yra sklaidos matricoje pateiktos

$$S_W = \sum_{i=1}^k \left[\sum_{j=1}^{n_i} (x_j^{(i)} - m^{(i)})(x_j^{(i)} - m^{(i)})^T \right] \quad (1.8)$$

1.6, 1.7, 1.8 lygtyse m atvaizduoja pavyzdinį vidutinį vektorių, n_i yra pavyzdžių skaičius i -tojoje klasėje, $x_j^{(i)}$ pateikia j -tosios veido ekspresiją i -tosios veide, o W yra projekcijos matrica.

Galutinė funkcija J_F - bandymas maksimizuoti Euklido distanciją tarp veido atvaizdų, priklausančių tam pačiam žmogui. Projekcijos kryptis, kuri maksimizuoja J_F , užleidžia projekcinės matricos W vektorių stulpelį. Galutinė J_F funkcija maksimizuojama, kai matricos W stulpelio vektoriai yra *tikri* vektoriai $S_W^{-1} S_B$. Svarbu pažymėti tai, kad pavyzdžių erdvės dimensija yra

didesnė nei pavyzdžių skaičius apmokymo rinkinyje, S_w tampa singuliarus, taigi jo inversija nebeegzistuoja. Keletas strategijų domėjosi ir bandė spręsti tokias situacijas.

Tiek PCA, tiek ir LDA išlaiko veido-erdvės globalinę struktūrą, tačiau *Fisher veido* metodas turi geresnę diskriminacinę gebą lyginant su *Tikro veido* metodu.

Kita veido atpažinimo technika, vadinama *Laplacian veido* būdu, naudoja padėties išlaikymo projekciją pažymėti veido suberdvėje [4]. Šitas metodas bando nustatyti veido suberdvę, kuri geriausiai aptinka esminę veido struktūros išlaikymo vietinę informaciją. Metodas duoda geresnį tikslumą lyginant su *Fisher veido* bei *Tikro veido* metodais.

1.1.2.4. Faktorių analizė (FA, factor analysis)

Faktorių analizė kaip ir daugelis kitų daugiavariacinių duomenų analizės metodų, remiasi hipoteze, kad tiriami kintamieji yra pagrindiniai požymiai lyginant su keliais paslėptais faktoriais. FA galime laikyti kaip PCA metodo apibendrinimą.

Veido aptikimo užduotyje FA analizės tikslas – gauti veido atvaizdo modelį (su aprėpiamu skaičiumi parametru), kurių pagalba būtų galima įvertinti testinio atvaizdo tikslumą su veido atvaizdu.

1.2. Lūpos – svarbiausias vizualinės kalbos bruožas

Iš visų galimų veido bruožų, kurie gali būti panaudojami kalbos atpažinimui, įskaitant akis, nosį, burną, tik pastaroji turi informacija, kuri yra glaudžiausiai susijusi su žmogaus kalba. Lūpos taip pat yra natūralus audio metodo (AM) papildymas. Lūpos atlieka pagrindinį vaidmenį automatinės kalbos atpažinime. Tai yra susiję su kalbos atpažinimu, naudojant vizualinį metodą (VM). Manoma [3], kad sudėtingos lūpų segmentavimo užduotys bus adresuojamos kitų, ypač vizualinės kalbos atpažinimo srityje. Galima vaizdavimo iš fotografijos sėkmė gali būti padidinama integruojant audio ir vizualinius signalus į vieną bendrą sistemą.

Gerai žinomas faktas, kad audio metodas gali būti dažnai keičiamas, jį gali įtakoti sveikata, nuotaika ir netgi vieta. Šitos nenuspėjamos variacijos reiškia, kad kuriant audio modelį, kuris tiksliai atspindi kalbėtoją, svarbiausia užduotis ir pagrindinis reikalavimas yra surinkti labai geros kokybės apmokymo duomenis iš pagrįstai ilgo (3 mėnesių) laiko periodo. Audio metodas – elgsenos duomenų apdorojimas, taigi susideda iš dinaminės informacijos; t.y. ta dinaminė informacija, kuri gali būti pastoviai pakeičiama. Apskritai vizualinis metodas (lūpos) mažiau įtakojamas nuo nenuspėjamų variacijų negu audio metodas. Taip yra dėl to, kad lūpos taip pat gali būti elgsenos duomenimis (kalbant), jos taip pat gali būti fiziologiniai duomenys. Iš esmės,

fiziologiniai duomenys yra nekeičiami. Veido bruožų išskyrimas naudojamas kalbos atpažinimui buvo klasifikuojamas į daug mažų skirtingų kategorijų. Pavyzdžiui, Kaucic [6] lūpų bruožus klasifikuoja į 3 galimas kategorijas:

1. Paremtos pikseliais;
2. Paremtos lūpų greičiu;
3. Paremtos lūpų kontūru.

Paremti pikseliais metodai maksimizuoja informacijos išsaugojimą, bet yra jautrūs šviesos variacijoms.

Paremtiems lūpų greičiu būdams svarbu lūpų judėseną. Skaičiavimo požiūriu jie yra labai brangūs bei nepritaikyti realaus laiko aplinkai.

Paskutinė kategorija – paremti lūpų kontūru metodai išskiria geometrinius dydžius iš lūpų kontūrų. Tai gali būti tiek vidiniai, tiek išoriniai lūpų kontūrų bruožai arba net ir paprasčiausiai – aukščio, pločio bei ploto dydžiai. Pagrindinis šito būdo privalumas yra tai, kad pagrindiniam procesui užtenka mažiau duomenų nei prieš tai aprašytiems metodams.

Luettin [7], kurio tyrinėjimai turėjo didžiausią indėlį kalbos atpažinimui naudojant lūpų bruožus, juos išskyrė tik į dvi kategorijas:

1. Paremtos vaizdu.
2. Paremtos modeliu.

Nepaisant atskirų autorių skirtingo lūpų bruožų kategorizavimo, lūpų bruožai išskiriami į dvi pagrindines kategorijas [3]:

1. Bendri (globaliniai);
2. Lokaliniai (specifiniai).

Praktikoje šie metodai yra neišskiriami: globaliniai bruožai apima tiek Kaucic paremtus pikseliais metodus, tiek Luettin paremtus vaizdais bruožus. Lokaliniai bruožai apima paremtus lūpų kontūru bei paremtus modeliu metodus. Lūpų greičiu paremti bruožai gali būti priskirti abiem kategorijom.

1.2.1. Bendri (globaliniai) vizualiniai bruožai

Globaliniai, holistiniai vaizdų bruožai yra pagrįsti transformacijomis su minimalios informacijos mažinimu. Duomenys yra intensyvūs. Dažnai burną sudarančių vaizdų intensyvumas yra naudojamas iškart ar po kelių išankstinių bruožų apdorojimų (pvz., sulyginimo). Šito metodo privalumas – tik nedaug duomenų yra ignoruojama, net jeigu ir nėra žinoma kokia lūpų informacija yra tikrai svarbi (pvz. išsikišimas, pozicija, dantys, liežuvis, dinamika). Šis duomenimis paremtas metodas leidžia klasifikuotojui nustatyti charakteristikų bruožus. Šito metodo trūkumai – labai daug

pasikliaujama klasifikuotoju bei jo sugebėjimu atlikti sudėtingas apibendrinimo užduotis. Dar vienas trūkumas yra tas, kad globaliniai metodai yra dažniausiai labai priklausomi nuo apšvietimo sąlygų, žiūrėjimo kampo bei orientacijos. Jie taip pat yra brangūs skaičiavimo požiūriu. Be to, skaičiavimas dažniausiai atskiria nuo realaus laiko lūpų sekimo.

Keletas globalinių/holistinių vaizdų apdorojimo būdų pavyzdžių [3]:

1. Skaitymui iš lūpų naudojamas klasikinis globalinis (paremtas pikseliais) metodas. Naudojamas langas aplink burnos regioną, intensyvumo vaizdų vektorius. Žymus duomenų sumažinimas gaunamas naudojant PCA, Linijinio diskriminanto analizę (LDA) bei taip pat 2D Greitos Furjė transformacijos (FFT) dydžio komponentą. Šių transformacijų koeficientai jungiami su Audio modeliu (AM) kalbos atpažinimui.

2. Tikrų lūpų (*eigenlips*) koncepcija (analogiškas tikrų veidų eigenfaces metodas, kuris yra pagrįstas PCA), bruožų išskyrimo metodas, kuris išskaido visą galimą vaizdo informaciją į daug mažesnius informacinius vienetus. Naudojami bruožai yra paimti tiek iš lūpų kontūrų PCA, tiek iš sucentruoto ploto aplink lūpas PCA intensyvumo. Pagrindinis PCA paremto metodo privalumas – žymiai mažesnis jautrumas iškraipymams, nuo kurių dažniausiai nukenčia globalinių metodų kokybė.

3. Palyginimo tarpusavyje metodas. Yra palyginama bangų transformacijų diskriminaciniai potencialai su Furjė transformacijomis. Naudojamas sucentruotas aplink burną esantis plotas.

Kiekvienas vaizdo pikselis yra vienos dimensijos intensyvumo signalas laiko atžvilgiu. Šitas intensyvo profilis vėliau apdorojamas Vilnelių transformacija arba Furjė transformacija [3]. Pirmasis abiejų transformacijų koeficientas yra ignoruojamas kalbos atpažinime dėl to, kad jis aprašo pikselių intensyvumo vidurkį per atitinkamą laiko atkarpą. Labai aukšti dažnio koeficientai taip pat yra eliminuojami, paliekant tik 4 ar 5 koeficientus pikselėje. 4-5 vaizdų rezultatas, kuris parodo laiko atkarpos informaciją.

1.2.2. Lokaliniai vizualiniai bruožai

Vizualinių bruožų išskyrimo metodai, kurie naudoja lokalius arba specifinius veido bruožų taškus yra sukonzentruoti į matomą kalbos artikulatorių, t.y lūpas. Rezultuojantys šito metodo tipo bruožai yra daugiausia parametruojami lūpų kontūrų apibūdinimais, kurie yra gaunami turint vertingą kalbos ar kalbėtojo informaciją [3]. Svarbiausia problema, susijusi su daug vaizdų apdorojimo užduotimis yra žymus dimensijos mažinimas. Žmogaus įsikišimas (rankinis sekimas ir segmentacija) - dažniausiai atrodo kaip sunkus darbas, tačiau efektyvus sprendimo būdas. Lokaliniai būdai bando atvaizduoti tikslią žmogaus segmentaciją. Jie naudoja algoritmus tam, kad būtų galima automatiškai segmentuoti svarbiausius apdorojimo taškus ir signalus bei išgauti

esminius bruožus. Svarbiausi šito metodo privalumai – potencialiai sugebėjimai atvaizduoti kalbėtoją žemesnėje dimensijoje nei holistiniu metodu. Kaip visada pagrindinis trūkumas yra tas, kad pasirinkti specifiniai veido bruožų taškai gali neapimti visos svarbios kalbėtojo ar kalbos informacijos (pvz., matomi dantys gali duoti papildomos informacijos, kuri nėra apibrėžta lūpų kontūro modeliu).

2. SKAITYMAS IŠ LŪPŲ

Vizualiniai bruožai išgaunami iš kalbančiojo veido vaizdų sekos ir susideda iš formos parametrų, kurie apibūdina lūpų ribų ir intensyvumo parametrus, taip pat ir pilkumos lygio pasiskirstymą burnos plote. Intensyvumo informacija yra sukurta remiantis PCA metodu, naudojant tikras erdves, kurios deformuojasi su formos modeliu [8]. Išgaunami parametrai svarbūs abiem, tiek kalbos, tiek kalbėtojo informacijai. Erdvės ir laiko kalbėtojo modelis yra sudarytas iš šitų bruožų, kombinuojant HMM su Gauso metodu [12].

Tuo tarpu aukščiausio technologijų lygio kalbėtojo atpažinimo sistemos naudoja tik akustinius kalbėtojo signalus. Kalbos atpažinimo bendruomenėje yra gerai žinoma, kad vizualinė informacija iš lūpų judesių užtikrina papildomą kalbos informaciją, kuri leidžia pagerinti kalbos atpažinimą.

Susiduriama su didele skaitymo iš lūpų problema – vizualinių kalbos bruožų modeliavimu, kurie svarbūs abiem, tiek kalbos informacijai, tiek ir kalbėtojo informacijai. Kalbėtojo atpažinimo tyrimai dažniausiai ignoravo vizualinės kalbos modalumą, kuris sudaro kalbėtojo atpažinimui diferenciacijos informaciją.

Veido atpažinimo tyrimai, iš kitos pusės, dažniausiai pagrįsti statiniais veido vaizdais bei neutralia veido ekspresija. Kad ir kaip bebūtų, veido išvaizda kalbant gali labai pakisti ir įtakoti veido ekspresiją. Nors burna ir priklauso nuo esminių pokyčių, tačiau yra labai svarbus veidų diferenciacijos šaltinis. Taigi veido ekspresijų analizės tyrinėjimai davė perspektyvių rezultatų. Šitų metodų tikslas buvo verčiau atpažinti veido ekspresijas, negu modeliuoti veido pakitimus atpažįstant žmogų. Dar vienas kombinuotas veido ir kalbėtojo atpažinimo būdas, naudojant sudėtinius signalus - veido atpažinimas, atliekamas iš statinių vaizdų su neutraliomis ekspresijomis, o kalbėtojo atpažinimas atliekamas iš akustinės pavienių signalų, išstartų kalbėtojo, analizės [8].

Deformuotas formos modelis yra naudojamas kalbant stebėti ir parametruoti lūpų ribas. Pilkumos lygio modelis paremtas PCA, deformuojasi su formos modeliu ir užtikrina burnos ploto intensyvumo informaciją. Abu, tiek formos, tiek ir intensyvumo parametrai tarnauja kaip bruožai sudaryti kalbėtojo modelį, pagrįstą kombinuotu HMM bei Gauso metodu. Preliminarūs kalbėtojo identifikacijos testai parodė gerus rezultatus, naudojant mažos duomenų bazės, sudarytos iš 12 subjektų, paremtų šitais bruožais [8].

Naujas kalbėtojo atpažinimo būdas buvo apibūdintas remiantis erdvių ir laiko burnos analize [8]. Veido bruožai išgaunami iš vaizdų sekos, kuri reprezentuoja lūpų formą ir intensyvumą. Bruožai yra mažos dimensijos ir invariantiški perdavimui bei rotacijai.

Atsižvelgiant į trumpą apmokymą bei testo trukmę, rezultatai yra daug žadantys, be to, jie parodo, kad lūpų informacija yra svarbi žmogaus identifikacijos dalis. Būtinai tolimesni tyrinėjimai metodo atlikimui su daugiau subjektų įvertinti, taip pat nagrinėti jų naudą, kombinuojant su kitais būdais, tokiais kaip kalbėtojo atpažinimas bei veido atpažinimas.

Kalbėjimas, skaitymas iš lūpų ne tik perduoda vartojimui vizualinę kalbos informaciją, tokią kaip lūpų judesiai kalbos suvokimui, bet taip pat perduoda vartojimui ir vizualinę informaciją, audio kalbos atpažinimo padidinimui ypač naudingam triukšmingoje aplinkoje.

Žmogaus skaitymo iš lūpų įgūdžiai leidžia numatyti kalbamų sakinių reikšmę iš matomos kalbėtojo artikuliacijos konfigūracijos (liežuvio, lūpų, dantų ir t.t.) bei judesių kartu su suvokimu iš konteksto. Žinios apie šias artikuliacijų pozicijas užtikrina informaciją apie akustinės kalbos signalų turinį. Derindami vizualinį turinį su leksiniu, sintaksiniu, semantiniu turiniu ir pragmatine informacija, žmonės gali išmokti suprasti sakytinę kalbą iš judesių kalbėtojo veide stebėjimo. Kalbos supratimui skaitymą iš lūpų plačiai naudoja žmonės, turintys klausos sutrikimų.

2.1. Lūpų kontūrai

Lūpų kontūrai yra specifinė lūpų kontūrų informacija, kuri naudojama kalbos ir kalbėtojo atpažinimui. Abu tiek apatinis, tiek viršutinis lūpų kontūrai yra aktyviai tyrinėjami. Luettn nuomone [7] vidinio lūpų kontūro sekimas yra žymiai sudėtingesnis nei išorinio lūpų kontūro dėl dantų. Naujausi vingiais paremti Kass metodai žymi tik išorinį kontūrą. Šnervių bei akių vietos yra taip pat naudojamos padaryti lūpų sekimą labiau tiksliau bei efektyviau, t.y. labai svarbūs faktoriai net jei ant veido yra plaukų. Dauguma lokalinių/specifinių metodų lūpų bruožų išskyrimui naudoja keletą aprašymo formų abiem tiek vidiniam, tiek išoriniam lūpų kontūrai. Be to, lūpų vidinių kontūrų išskyrimas gali būti problematiškas dėl matomų artikuliatorių, tokių kaip dantys, liežuvis, lūpos.

2.2. Lūpų segmentacija naudojant rankinę intervenciją

Nepaisant akivaizdžių skirtumų tarp globalinių ir lokalinių bruožų metodų, yra ir vienas bendras procesas, kurį abu metodai dažnai dalijasi, t.y., rankinė intervencija [3]: sudėties pritaikymas, rankinis duomenų sekimas, svarbiausio ploto apibrėžimas ir dažniausiai lūpų regionas. Teoriškai rankinis sekimas turėtų užtikrinti kokybę. Apskritai tikslumas priklauso tik nuo veido charakteristikų, bet ne nuo nevienareikšmiškų segmentacijos klaidų. Rankinis sekimas taip pat turi ir trūkumų. Praktinis trūkumas yra susijęs su rankinio sekimo laiko sąnaudomis. Tai reiškia, kad eksperimentai, kuriems duomenys yra įvedami ranka, susideda iš mažesnės duomenų bazės.

Pavyzdžiui, Chibeluschi [3] apskaičiuoja geometrinius lūpų bruožus naudojant beveik 5000 vaizdų. Luettin [7], kuris naudoja pusiau automatinę lūpų išskyrimo procedūrą, gali naudotis didesne duomenų baze. Naudojant lūpų segmentacijos procedūras pagrįstas aktyvių formų modeliais pasiūlytais Kass, Luettin apskaičiuoja lūpų bruožus naudojant beveik 27000 vaizdų.

Automatinis apskaičiavimas turi ir trūkumų – nežinomas lūpų segmentacijos tikslumas bei jo įtaka atpažinimo taškams.

2.3. Pusiau automatinis lūpų segmentavimas

Tam, kad būtų galima palyginti bendrą diskriminacinę galimybę VM ir AM bei apskaičiuoti lūpų bruožų limitus, lūpų segmentacijos klaidos turi būti prilygintos minimumui [3]. To siekiama naudojant geros raiškos lūpų duomenis. Privalumas naudojant geros raiškos lūpų duomenis prieš rankinio sekimo duomenis yra tas, kad lūpų segmentacijos tikslumas gali būti pasiekiamas didesnėje duomenų bazėje turinčioje mažai trukdžių.

Geros raiškos lūpų duomenų naudojimo trūkumas - lūpų bruožų vertingumo apskaičiavimas.

2.4. Lūpų bruožų išskyrimas

Lūpų bruožų išskyrimo algoritmas yra sudarytas iš keleto smulkesnių procesų, tokių kaip audio galutinio taško aptikimas, kontūrų braižymas, diskretinio kosinuso transformacijos (DTC) ir normalizacijos [3]. Kuomet lūpų forma yra išskiriama iš vaizdų (segmentacija), tai rinkinys iš statinių ortogonalinių kalbėtojo specifinių bruožų yra pritaikomas kiekvienam kalbėtojui. Statiniai bruožai vėliau yra apdorojami į pritaikomus dinaminis bruožus. Kontūrų bruožai yra išskiriami iš abiejų viršutinės ir apatinės lūpų.

Iškeliami du svarbūs klausimai [3]:

1. Kurie vaizdo bruožai yra svarbūs žmogaus reprezentacijai?
2. Kaip sumodeliuoti žmogų su šitais bruožais?

Bruožų išskyrimo yra panaudojami metodai iš veido atpažinimo teorijos – išskirti ir sumodeliuoti erdvės informaciją bei metodai iš kalbėtojo atpažinimo teorijos – sumodeliuoti laikinę priklausomybę iš šių bruožų.

Kadangi labiau svarbi vizualinė informacija, kylanti iš kalbėjimo, todėl limituojama burnos ploto vaizdų analizė. Bendri veido atpažinimo būdai yra paremti geometrine ir intensyvumo informacija [3]. Yra kombinuojami abu būdai, tariant, kad daug informacijos apie kalbėtojo identifikavimą yra sudaryta iš lūpų kontūrų ir pilkumos lygio diferenciacijos apie burnos plotą. Kalbėjimo metu lūpų forma keičiasi lygiai, tačiau turi ir su kalbėtoju susijusios informacijos.

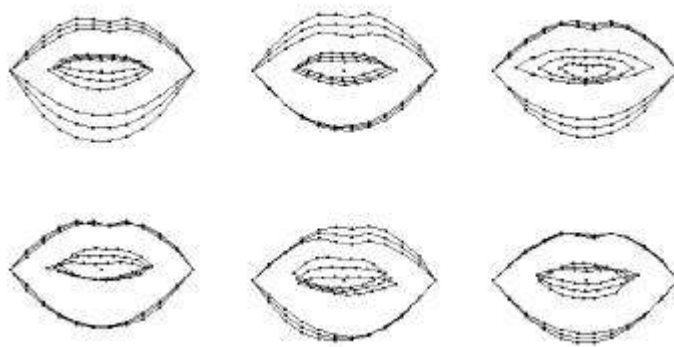
Stengiamasi išnaudoti šitą faktą sudarant kiekvienam kalbėtojui erdvės ir laiko modelį, kuris apibrėžia kalbėtojo burną ir jos laikinus pakitimus kalbant.

Naudojamas formos modelis [8], apibrėžiantis vidinius bei išorinius lūpų kontūrus, ir deformuojantis pilkumos lygio modelis, kuris apibūdina intensyvumo reikšmes aplink lūpų kontūrus. Kontūro modelis iš pradžių buvo sukurtas vaizdų paieškai ir reprezentavo tik mažą vaizdų ploto dalį. Informacija sudaryta iš kontūrų vektorių gali aprėpti ne visą kalbėtojo informaciją, t.y. pavyzdinė erdvė neaprepia viso burnos ploto, taigi informacija apie dantis, kuri gali būti labai svarbi, limituojama.

Metodai, sudaryti iš aktyvių formų modelių, yra naudojami nustatyti, stebėti ir parametruoti lūpas vaizdų sekoje. Tai yra deformuojamų kontūrų modeliai, kurie reprezentuoja objektą pažymėtų taškų rinkinį. Deformacijos pagrindiniai modeliai yra gaunami atliekant PCA pažymėtame apmokymo rinkinyje. Bet kokia forma gali būti aproksimuota prie pagrindinės formos linijinės kombinacijos ir deformacijos pirmų kelių pagrindinių modelių.

Sumodeliuoti intensyvumą aplink burnos plotą yra naudojamas pilkumos lygio modelis, kuris apibūdina intensyvumo vektorius statmenai kontūrai kiekviename modelio taške. Visų modelių taškų sąryšio vektoriai reprezentuoja kontūro pavyzdį. Analogiškai formos deformacijai PCA yra atliekamas visuose apmokymo rinkinio kontūro pavyzdžiuose tam, kad būtų sumažinta bruožų erdvė ir išgauti pagrindiniai modeliai iš kontūrų variantų, pateikiami 2.1 paveiksle [8]. Bet kuris apmokymo rinkinio kontūras gali būti aproksimuotas pagrindinio kontūro linijine kombinacija taip pat ir keli pirmi kontūrų variantų modeliai. Profilio modelis jungiasi su kontūro modeliu, taigi, reprezentuoja tuos pačius objekto bruožus.

Profilio modelis pirmiausia yra naudojamas vidinių ir išorinių lūpų kontūrų stebėjimo užtikrinimui, modeliuojant skirtingų žmonių ir skirtingų apšvietimo sąlygų intensyvumo variantus [8]. Pagrindiniai formos ir profilio modelio aspektai yra atkurti iš stebėjimo rezultatų ir pateikti kaip bruožai kalbėtojo atpažinimo sistemai. Šitie parametrai taip pat turi užtikrinti svarbią informaciją vizualinės kalbos atpažinimui (skaitymui iš lūpų). Modeliai įvertinami iš pokyčių tarp kalbėtojų ir kalbėjimo variantų. 2.2 paveikslėlyje [8] pavaizduotos atkurtos lūpos su vidutine forma ir vidutiniu profiliu (lūpų regionas pripildytas interpoliuotu pilkumos lygiu tarp profilio vektorių). Aktualų profilio modelį taip pat apibūdina burnos vidus ir išplėstas išorinis lūpų kontūras.



2.1 pav. Pirmi šeši pagrindiniai formos variantų modeliai, įvesti į apmokymų rinkinį iš visų subjektų ir viso žodžio sekos



2.2 pav. Atkurtos lūpos, naudojant vidutinę formą ir vidutinį profilio parametrus

Profilio modelis taip pat aprėpia plotą burnos viduje ir maždaug pusės lūpų storumo pločio išorines lūpų ribas, kurios čia nepavaizduotos.

2.4.1. Lūpų išskyrimas naudojant spalvų segmentaciją

Naudojami veido bruožai yra iš lūpų-geros raiškos rinkinio [3]. Svarbu yra tai, kad geros raiškos regionai gali netiksliai atspindėti tikrąjį lūpų paviršių. Segmentacijos proceso rezultatai yra virtualios lūpos, kurios tiksliai atspindi subjekto identiškumą įrašymo sesijos metu, tačiau jų beveik neįmanoma atkurti jeigu lūpų sekimas yra panaikinamas.

Mėlynos lūpos yra segmentuojamos, naudojant vientisa spalva paremtą išskyrimo algoritmą. Yra du galimi segmentacijos būdai [3]:

1. Sukurti žmogaus odos spalvos aptikimą ir nuimti visus spalvotus odos pikselius nuo vaizdo. Taip paliekami mėlyni pikseliai kurie apibūdina lūpas.
2. Kitas būdas – sukurti mėlynų lūpų spalvos modelį ir tiesiogiai išskirti mėlynus regionus. Greičio atžvilgiu 2 būdas yra labiau priimtinas.

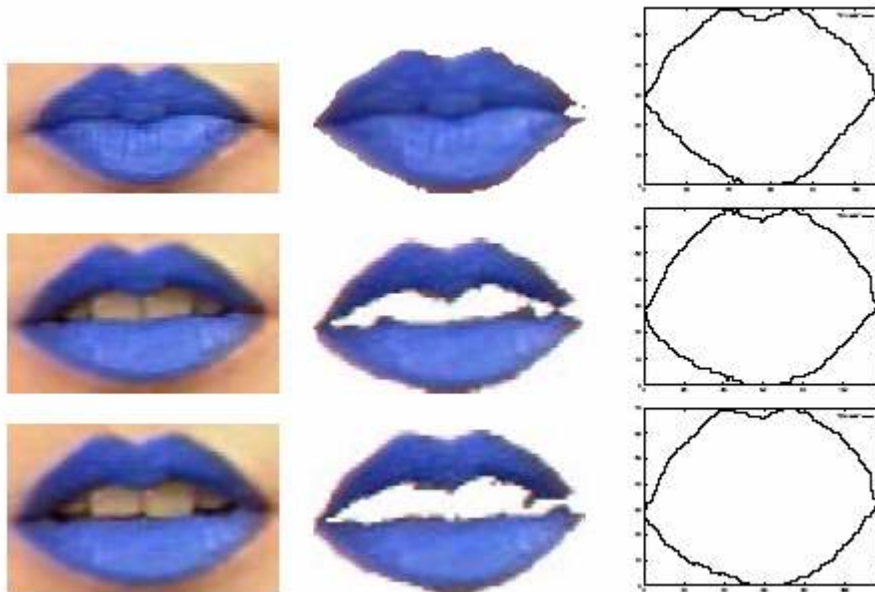
Mėlynų lūpų kiekybinio vektoriaus modelis yra sudarytas iš mažų duomenų po-rinkinių [3]. Mažos mėlynų lūpų dalys (išsaugomos kaip stačiakampiai vaizdai) yra rankiniu būdu išskiriamos iš

kalbėtojo. Apytikriai 10 lūpų dalių vaizdų yra išskiriama iš viso. Kiekvienas vaizdas susideda iš maždaug 400 pikselių (20x20). Kiekvienas pikselis sudarytas iš raudonos, žalios ir mėlynos (RGB) spalvų santykio. Visi RGB yra perkelti į atskirą failą ir sugrupuojami naudojant VQ (kiekybinio vektoriaus) metodą. Kiekvienas RGB nagrinėjamas kaip bruožų vektorius. Bruožų vektoriai tuomet yra sugrupuojami duomenų kiekio mažėjimo tvarka ir taip sukuriama mėlynų lūpų spalvos modelis. Iš viso maždaug 4000 (400x10) apmokytų bruožų vektorių (RGB) yra naudojama modelių sukūrimui. Apmokymo duomenys yra sugrupuojami tik iki 4 reikšmių bruožų vektorius, t.y. mėlynų lūpų modelis susideda iš 4 dydžio kodų rinkinio [3].

Bandomasis vaizdas yra segmentuojamas imant kiekvieną pikselį ir apskaičiuojant RGB Euklido distancija tarp pikselio ir kiekvieno kodinio žodžio modelyje. Jeigu mažiausiais distancijos matmuo yra žemiau ribos, pikselis klasifikuojamas kaip lūpų dalis.

2.4.2. Vizualinio modelio (VM) bruožų išskyrimas

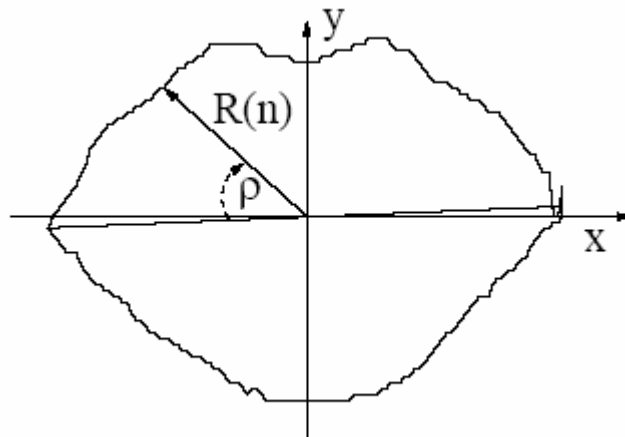
2.3 paveikslas [3] iliustruoja pirmą bruožų išskyrimo proceso etapą. Po segmentavimo stačiakampio formos ribų langelyje yra automatiškai pritaikomi lūpų ekstremumai.



2.3 pav. Lūpų išskyrimo procesas

Tuomet yra nustatoma pagrindinė įstrižainė, t.y. ta linija, kuri eina iš vieno lūpų kampo į kitą per lūpų centrą. Šitos įstrižainės vidurinis taškas yra naudojamas kaip koordinatė išorinių lūpų kontūro perimetro magnitudiniam matavimams. Tai pavaizduota 2.4 paveikslėlyje [3]. Spindulio dydis yra matuojamas kiekviename taške, kuomet pasukamas po pikselį aplink centrą ir spindulio

ašį. Šitie išmatavimai yra interpoliuojami ir sumažinami iki vienodų kampo intervalų p , aplink lūpų kontūrą (128 taškai per 360 laipsnių).



2.4 pav. Išorinių lūpų kontūro perimetro apskaičiavimas

Rezultantinis vektorius, kuris susideda iš 128 vienodų tarpų spindulių reikšmių yra žinomas kaip išorinių lūpų reikšmė (OLM). Pirmoji OLM reikšmė yra distancija iš centro į kairįjį burnos kampą. Kontūras yra fiksuojamas pagal laikrodžio rodyklę, taigi viršutinių lūpų profilis fiksuojamas prieš apatinių lūpų profilį. OLM gali būti netikslus todėl kalbėtojo specifinė informacija gali būti nevisiškai matoma. Norint pagerinti OLM, išgauti ortogonalų bruožų rinkinį bei palengvinti bruožų sumažinimą OLM yra transformuojamas diskretine kosinuso transformacija (DCT). Tai leidžia peržiūrėti esamą informaciją signale alternatyviu formatu, kuriame yra lengviau atlikti veiksmus ir analizuoti. 2.1 lygtyje parodyta DCT apskaičiavimas [3].

$$X_c(k) = \alpha(k) \sum_{n=0}^{N-1} R(n) \cos\left(\frac{\pi(2n+1)k}{2N}\right) \text{ for } k = 0, 1, \dots, (N-1)$$

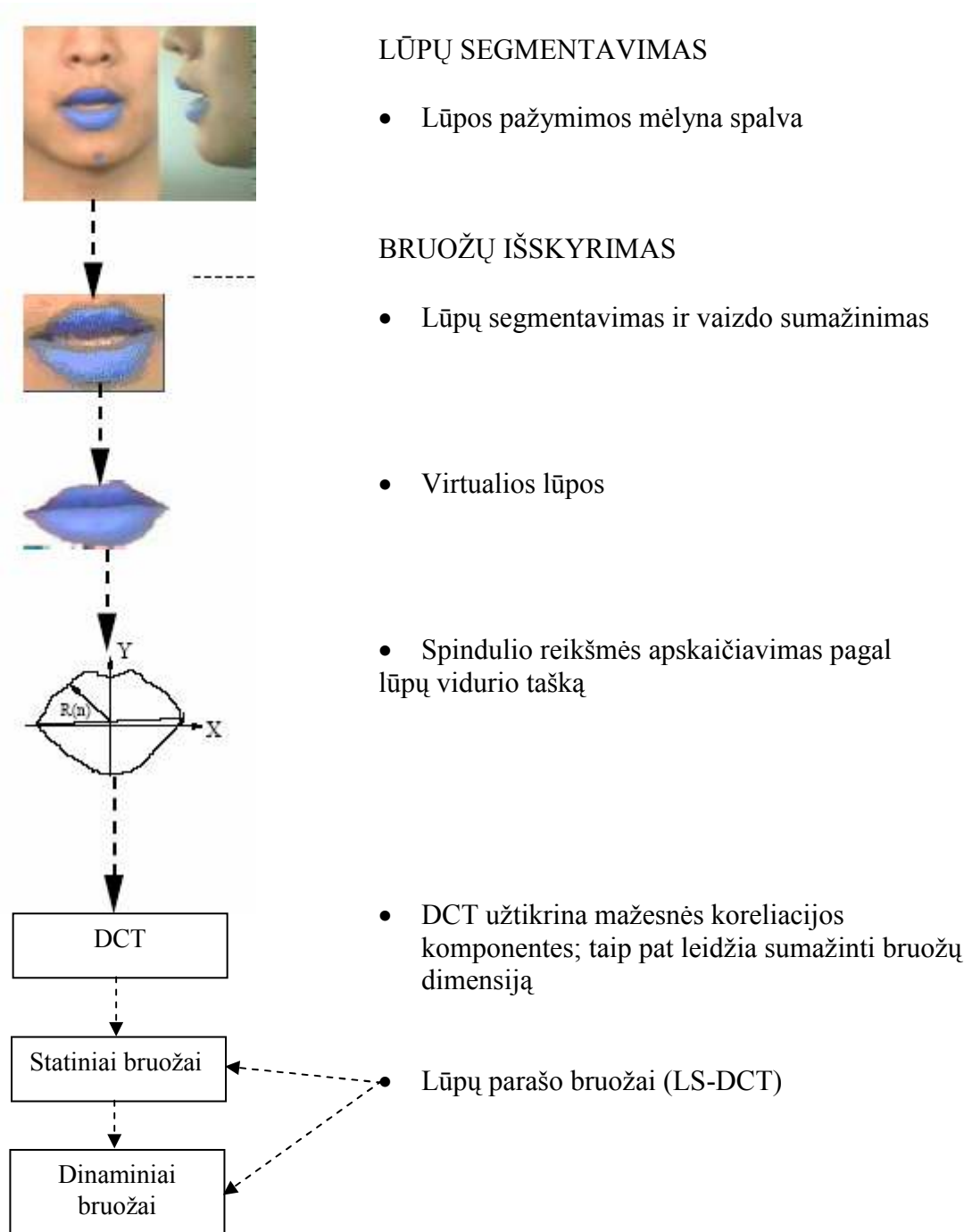
$$\text{where } \alpha(k) = \begin{cases} \sqrt{1/N} & \text{for } k = 0 \\ \sqrt{2/N} & \text{for } k = 1 \rightarrow (n-1) \end{cases} \quad (2.1)$$

čia $X_c(k)$ yra DCT koeficientai ir $R(n)$ yra diskretinis įėjimo signalas, kur N yra pavyzdžių skaičius t.y. $R(n)$ parodo interpoliuotas spindulio reikšmes.

128 OLM taškai yra transformuojami naudojant DCT į alternatyvius duomenis, kurie leidžia patikrinti visus kalbėtojo išskirtinius lūpų bruožus. Rezultantinė OLM transformacija yra nurodoma kaip lūpų parašas (LS-DCT). Vizualiniai duomenys yra fiksuojami 30 kadrų per sekundę. Kiekvienas kadras turi atskirą LS-DCT parašą.

Naudojant šiuos tris atskirus bruožų rinkinius (visą, viršutini ir apatinį) tiksliai nustatomos kalbėtojo lūpos.

Lūpų bruožų išskyrimo algoritmas yra pavaizduotas 2.5 paveikslėlyje [3].



2.5 pav. Lūpų bruožų išskyrimo algoritmas

2.4.3. Dinaminiai lūpų bruožai

Lūpos yra biometrinės savybės, kurios parodo, kad dinaminiai (judesio) bruožai yra labai svarbūs. Kadangi lūpos susideda iš dviejų dalių (kur kiekviena iš jų turi fiziologinių ir

funkcionavimo komponentų), dinaminiai bruožai, kurie turi būdingą lūpų judesį taip pat yra atskirai tyrinėjami. Tarpiniams bruožams, trumpai momentinių bruožų vektorių sekai yra analizuojami jų laiko sekos informacija [3]. Analizuojant galima naudoti paprastą diferencinę arba regresinę analizę. Audio srityje regresijos metodas turi turėti nežymų atlikimo privalumą dėl paprastos diferenciacijos. 2.2 lygtyje pateikiama pirmos eilės regresijos koeficientas $R_1(t, N_T, \Delta t)$ [3],

$$R_1(t, N_T, \Delta t) = \frac{\sum_{i=-(N_T-1)/2}^{(N_T-1)/2} i x_k(t + i \Delta t)}{\sum_{i=-(N_T-1)/2}^{(N_T-1)/2} i^2} \quad (2.2)$$

čia x_k yra momentinis bruožų komponentas ir Δt , atskaitų periodas.

Pagal 2.2 lygtį galima nustatyti, kad optimalaus lango ilgio parinkimas N_T regresijos apskaičiavimas yra balansas tarp vidurkinimo išdeformuojant, kai tuo tarpu rezoliucija išlaikoma tolygia skale su laiko įvykiais.

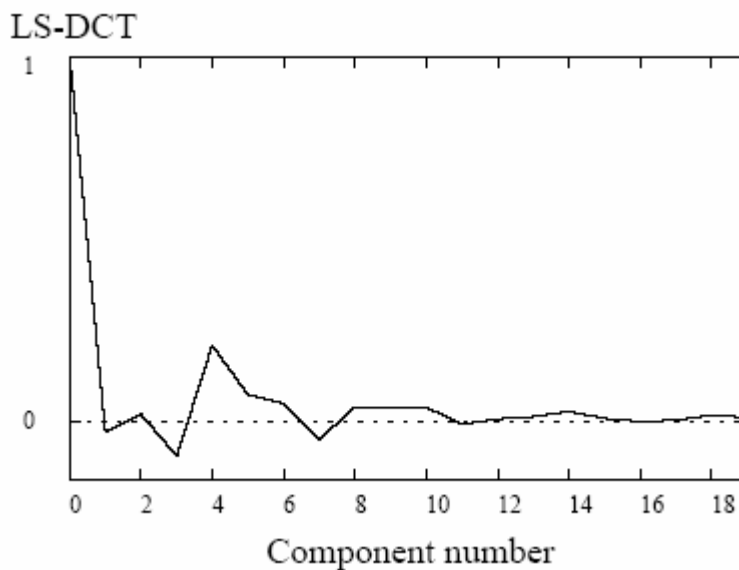
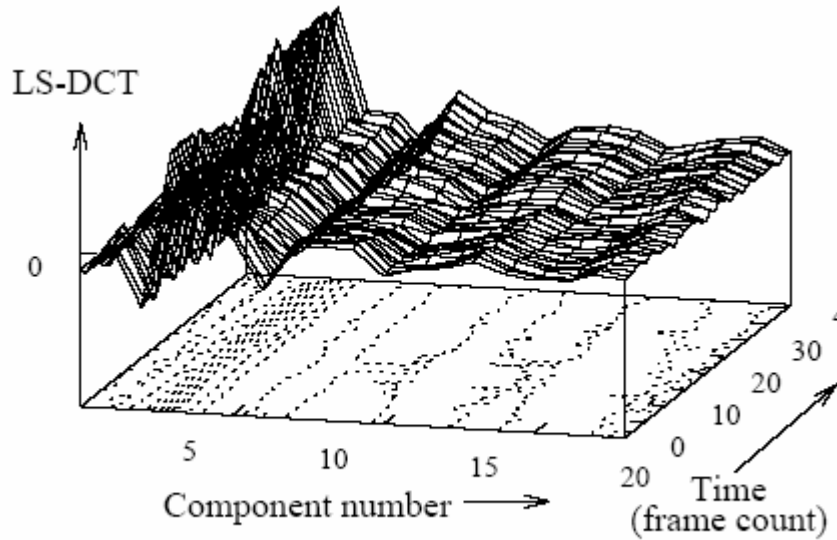
Kartais dinaminiai bruožai yra labiau tvirti nei statiniai kalbėtojo bruožai.

2.4.3. Lūpų parašo vizualinė interpretacija

2.6 pav. [3] parodyta LS-DCT eilė su pavaizduotais skaitmeninės kalbos pavyzdžiais. Pateikta apie 30 profilinių atvaizdų bei kalbos kadro. Tam, kad būtų galima įvertinti santykinės reikšmės AC komponentų, DC komponentas iš LS-DCT yra pašalinamas iš profilių viršutiniame paveiksle. Tam, kad būtų aiškiau viršutiniame paveiksle parodytas atskiras LS-DCT paimtas tik iš vieno vaizdo kadro. DC komponentas įtrauktas tik dėl palyginimo.

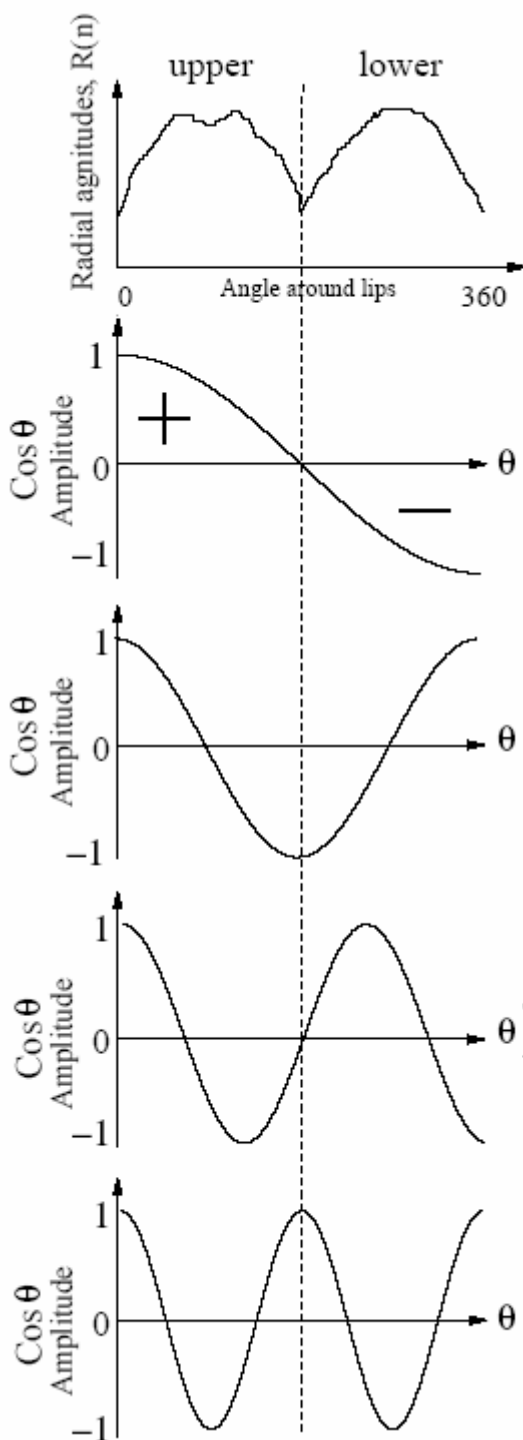
Viršutinis signalas 2.6 paveiksle pateikia OLM pavyzdį, paimtą iš atskiro vaizdo kadro. Kairė OLM pusė parodo viršutinių lūpų spindulio reikšmes, dešinė OLM pusė parodo apatinių lūpų spindulio reikšmes. X ašis parodo distanciją - kiekvieną spindulio tašką nuo pradinės pozicijos (kairiojo burnos kampo), pavaizduotą laipsniais. Y ašis parodo spindulio reikšmes $R(n)$. DCT bando skaidyti OLM signalą į kosinuso komponentų svorinių vidurkių sumą. Viršutiniai keturi signalai parodo pirmus keturis DCT dažnius. Pirmoje komponentėje pateikimas OLM signalas, kuris yra teigiamas viršutinėm lūpom ir neigiamas apatinėm lūpom. Ši savybė įrodo, kad pirmoji DCT komponentė apibūdina skirtumą tarp viršutinių ir apatinių lūpų. Antra ir ketvirta komponentės yra iš kairės į dešinę simetriškumo ir asimetriškumo indikacijos. Yra sudėtinga apibūdinti 3-ąją DCT komponentę. Iš kairės į dešinę natūrali simetrija atitinka 4 komponentę būdama didžiausia DC

sesija (tai parodoma 2.7 paveiksle [3]. Viršutinė figūra parodo DCT koeficientų seką (kur DC koeficientai pašalinti dėl aiškumo). Didžiausia komponentė yra 4 koeficientas. Apatinėje 2.7 pav. dalyje parodytas ketvirtasis koeficientas, atsižvelgiant į DC vidurki, iliustruojamas poreikis normalizuoti visus koeficientus priklausomai nuo vidurkio.



2.6 pav. LC-DCT komponentų lūpų parašo pavyzdžiai

Viršutinis - kalbėjimo momentu (DC komponentai pašalinti tam, kad būtų aiškiau); apatinis – atskiras kadras (DC komponentas įtrauktas palyginimui).



Lūpų parašas – OLM

Pirma DCT komponentė.
- skirtumas tarp apatinių ir viršutinių lūpų

Antra DCT komponentė
- apatinių/viršutinių lūpų nesimetriškumas

Trečia DCT komponentė

Ketvirta DCT komponentė
- viršutinių/apatinių lūpų simetrija

2.7 pav. Apatinės DCT komponentės, atsižvelgiant į apatines ir viršutines lūpų ribas

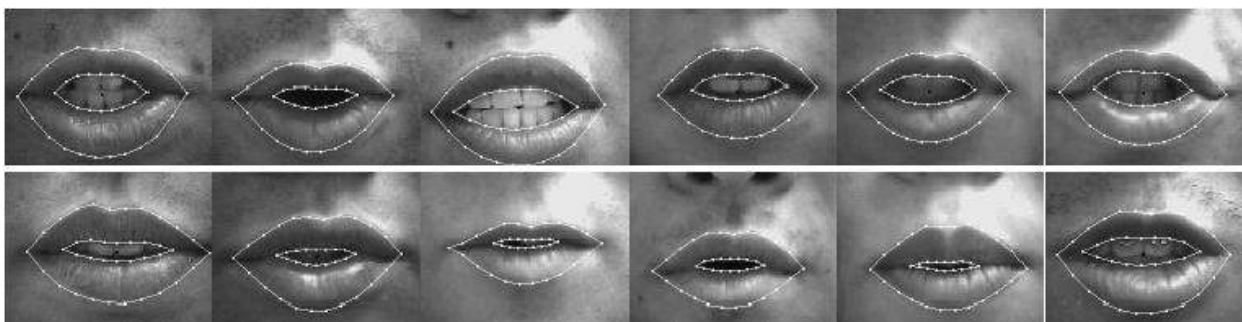
2.5. Kalbėtojo modeliavimas

Kalbėtojo modeliavimas yra sudarytas kombinuojant HMM ir Gauso metodus [8]. Kalbant naudojami išgaunami vizualiniai parametrai ir jų laikinos priklausomybės tam, kad būtų reprezentuojamas kalbantysis asmuo. Formos ir intensyvumo parametrai yra išgaunami

kiekviename išskirtiniame laiko intervale, suformuojant bruožų vektorių. Be to, skalę galima sudaryti svarbia su kalbėtoju susijusia informacija, tačiau ji nėra naudojama kaip bruožas todėl, kad neįmanoma įvertinti absoliučių skalės reikšmių iš duomenų bazės, kuri susideda tik iš burnos ploto. Formos ir intensyvumo parametrai sudaro su kalba bei kalbėtoju susijusią informaciją, kai kurie šitų intensyvumo parametrų sudarys ir apšvietimo informaciją.

Kalbėtojo atpažinimo testai gali būti išskiriami į nuo teksto priklausančias (TD) ir nuo teksto nepriklausančias (TI) užduotis. Testo kalba yra žinoma nuo teksto priklausančioms užduotims, o nuo teksto nepriklausančioms – nežinoma. Buvo atlikti eksperimentai [8], pavyzdys pateiktas 2.8 pav., su abejais TD ir TI modeliais, kur TI modelis yra apribotas žodyno dydžio. Naudojama duomenų bazė susideda iš atskirų žodžių, taigi apmokomi viso žodžio kalbėtojo modeliai. TD modeliui sudaromas vienas HMM žodžio rūšiai ir kalbėtojui. TD modelyje tariamas žodis yra žinomas, taigi tik tų pačių žodžių rūšies HMM yra naudojamas atpažinimui. TI modelyje tariamas žodis nežinomas ir nespecifinis HMM reprezentuojantis visas žodžių rūšis yra naudojamas atpažinimui.

Buvo apmokomas HMM, kuris leidžia tik uždaros grandinės metodą ir nuoseklų perėjimą tarp esamos ir sekančios būsenos. Pradinės būsenos galimybės yra lygios nuliui. Likę parametrai yra apskaičiuojami iš apmokymo rinkinio išgaunamo modelio parametrų [8]. Kiekvienas HMM inicijuojamas apmokymo vektorių linijinės segmentacijos į HMM būsenas, sekant pasikartojančių segmentų k-reikšmių grupavimo ir Viterbi lygiavimo. Modeliai toliau yra apskaičiuojami naudojant Baum-Welch procedūrą. Kiekvieno subjekto taškai yra skaičiuojami naudojant Viterbi algoritmą, o klasifikacija atliekama apskaičiuojant posteriorinės tikimybės maksimumą (MAP) bei pasirenkant subjektą su aukščiausia tikimybe. Visi eksperimentai atliekami naudojant HMM įrankių komplektą HTK V1.5.



2.8 pav. Eksperimento pavyzdys (12 kalbėtojų vaizdų pavyzdžiai, naudojami lūpų stebėjimo rezultatoms)

2.5.1. Nuo teksto priklausantis testas (TD)

TD atpažinimui naudojamas atskiras HMM kiekvienam subjektui ir kiekvienai žodžio rūšiai, iš viso 48 modeliai. Dėl mažo duomenų bazės dydžio yra naudojama tokia apmokymo procedūra [8]:

1. Apskaičiuojami vieno globalinio modelio variantai, naudojant visą apmokymo rinkinį.
2. Perskaičiuojami vidurkiai, kombinuojant reikšmes ir perdavimo galimybes nuo subjekto nepriklausantiems žodžio modeliams.
3. Perskaičiuojami vidurkiai ir kombinuojamos reikšmės nuo subjekto priklausantiems žodžio modeliams.

Visi HMM turi tas pačias, bet kurių žodžių rūšių dispersijas ir perdavimo galimybes visiems subjektams. Tik vidurkiai ir kombinuotos reikšmės yra apskaičiuojamos individualiai kiekvieno subjekto kiekvienai rūšiai.

Atpažinimas yra sudaromas iš kalbėtojo tariamo žodžio modelių. TD testo rezultatai pateikiami 2.1 lentelėje [8]. Geriausias rezultatas buvo gautas naudojant abu, tiek formos, tiek ir intensyvumo parametrus.

2.5.2. Nuo teksto nepriklausantis testas (TI)

TI atpažinimui naudojamas vienas HMM kiekvienam subjektui, reprezentuojantis visus išsireiškimus. Šito būdo tikslas yra sudaryti vieną modelį, kuris reprezentuotų skirtingas žodžių rūšis pagal skirtingus kombinuotus komponentus. Parametrų apskaičiavimas nėra toks svarbus kaip nuo teksto priklausančiame modelyje. Pateikiamas toks apibūdinimas [8]:

1. Apskaičiuojami variantai, vidurkiai ir kombinuotos reikšmės vienam globaliniam modeliui.
2. Perskaičiuojamas vidurkis, kombinuotos reikšmės ir perdavimo galimybės nuo teksto nepriklausančiam kalbėtojo modeliui.

Atpažinimas yra pagrįstas nuo teksto nepriklausančiais modeliais. 2.1 lentelė parodo nuo teksto nepriklausomo asmens identifikacijos testo rezultatus. Geriausias rezultatas taip pat buvo gautas naudojant abu tiek formos, tiek ir intensyvumo parametrus. Beje, akustiniame kalbėtojo atpažinime, TI modelio atlikimas yra žymiai blogesnis nei TD modelio. TD modelio mažas apmokymo komplektas sudaromas tik iš vieno pavyzdžio modeliui. Abiem užduotim intensyvumo parametrai buvo aukštesni nei formos parametrai. Tačiau intensyvumo parametrai gali būti klaidingi todėl, kad kai kurie intensyvumo modeliai gali skaičiuoti skirtingą ryškumą. Ryškumo įtaka turi būti tyrinėjama su didesne duomenų baze. Vienas būdas kaip padidinti ryškumo nestabilumą gali

būti apmokymo vaizdų su skirtingu apšvietimu įtraukimas arba intensyvumo modelių ir bruožų vektorių, kurie yra žinomi ryškumo apskaičiavimui, pašalinimas [8].

2.1 lentelė

Žmogaus identifikavimo testų tikslumas, nuo teksto priklausančio (TD) ir nuo teksto nepriklausančio (TI) modelių, naudojant formos ir intensyvumo parametrus

	Forma	Intensyvumas	Forma + Intensyvumas
TD	72.9 %	89.6 %	91.7 %
TI	83.3 %	95.8 %	97.9 %

3. AUDIOVIZUALINĖ KALBA

Audiovizualinės kalbos atpažinimo sistema - AVSR (audio-visual speech recognition) naudoja tiek vizualinės, tiek audio kalbos generavimo modalumus, kurie yra invariantiški akustinio triukšmo trikdžiams [9]. Nuo kalbėtojo nepriklausomo audiovizualinės nepertraukiamos kalbos atpažinimo sistema pagrįsta tvirtu vizualinių bruožų rinkiniu, išgautu iš burnos regiono tikslių aptikimų bei stebėjimų. Vizualinių ir akustinių stebėjimų seka yra integruota, naudojant *keleto paslėptų Markovo metodų (CHMM, coupled hidden Markov)* modelį. Statistinės CHMM galimybės gali modeliuoti audio ir vizualinės būsenos asinchroniškumą, be to, taip pat išlaikomos ir jų natūralios koreliacijos bėgant laikui.

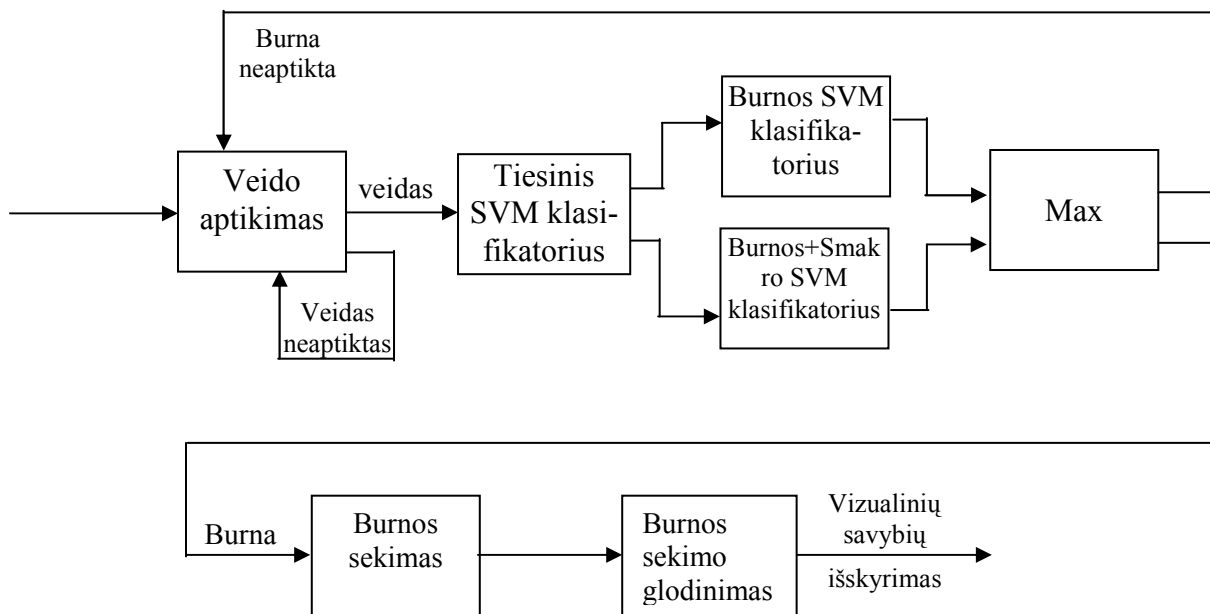
Šiuo metu galimos ASR (*audio speech recognition (audio kalbos atpažinimo)*) sėkmė apribota reliatyviai kontroliuojamos aplinkos ir gerai apibrėžia pritaikymus, tokius kaip diktavimą, balsu grįstų kontrolinių komandų mažą-vidutinį žodyną (laisvų rankų surinkimas ir t.t.). Paskutiniams metams kartu tyrinėjant atskiras akustinio triukšmo sumažinimo technikas, kombinuoti audio ir vizualinių bruožų sistemų tyrinėjimai atskleidė [9] patrauklų kalbos atpažinimo mažiau suvaržytoje aplinkoje sprendimą. Vizualiniai bruožai dažniausiai nustatomi iš burnos formos. Šitie metodai priklauso išimtinai tik nuo tikslių lūpų kontūrų aptikimo. Alternatyvus būdas yra išgauti vizualinius bruožus iš lūpų regiono transformuotos pilkumos skalės intensyvumo vaizdo.

Audiovizualinių bruožų sintezės sistemoje stebėjimo vektoriai yra išgaunami iš sujungtų audio ir vizualinių stebėjimo vektorių, sekant dimensinės redukcijos transformaciją [9]. Tuomet stebėjimo sekos rezultatai yra modeliuojami, naudojant vieną HMM. Daugiasrautis HMM parodo, kad audio ir vizualinė sekos yra sinchroniškos būsenos, tačiau leidžia audio ir vizualiniam komponentams turėti skirtingas kontribucijas visoms stebėjimo tikimybėms. Tačiau yra žinoma, kad kalbos akustiniai bruožai yra vėlinami lyginant su vizualiniais kalbos bruožais. Audiovizualinis HMM produktas yra kaip multisrauto HMM tęsinys, kuris leidžia audiovizualinės būsenos asinchroniškumą. Sprendimo sintezės sistemos nepriklausomas modelis audio ir video sekoje naudoja du HMM ir kombinuoja kiekvienos stebimos sekos tikimybę, pagrįstą kiekvieno modalumo patikimumu [9].

Nuo kalbėtojo nepriklausomos audiovizualinės nepertraukiamos kalbos atpažinimo sistema prasideda nuo burnos regiono aptikimo ir stebėjimo, sekant tvirto rinkinio išgavimu iš vizualinių burnos regiono stebėjimų. Tuomet audio stebėjimai yra kombinuojami su vizualiniais stebėjimais, naudojant CHMM.

3.1. Burnos aptikimas ir sekimas

Burnos aptikimas pavaizduotas 3.1 pav. [9]. Jis prasideda nuo vartotojo veido aptikimo, naudojant neuroninių tinklų metodą.



3.1 pav. Burnos aptikimo ir stebėjimo sistema

Burnos aptikimas yra ženkliai priklausomas nuo veido aptikimo. Naudojama SVM (*support vector machine*) klasifikatoriaus kaskada tam, kad būtų galima nustatyti burnos vietą, atskiriant ją nuo apatinės veido dalies. SVM kaskada sukurta taip, kad užtikrintų tiek realaus laiko, tiek tikslumo reikalavimus visos *audiovizualinės kalbos atpažinimo* (AVSR, *audio-visual speech recognition*) sistemos. Tam, kad būtų galima pritaikyti skalės variacijas, įvertinto diapazono multiskalinis tyrinėjamas yra tik pradedamas nuo tyrinėjamo regiono vaizdų pakartotinio pavyzdžių atrinkimo priklausomai pagal pastovų faktorių. Prieš-procesinis žingsnis normalizuoja kiekvieną testo pavyzdį. Toliau SVM filtras, kurio dydis 16x14 su linijiniu branduoliu, taikomas žemesniam veido regionui, nustatyti burnos aptikrę vietą. Du SVM klasifikatoriai su Gauso branduoliu, kurio dydis 32x28, apmokomi burnos regiono pavyzdžiais (su ir be veido plaukų), taikomi kiekvienam testo pavyzdžiui ir jo pasuktos versijos vaizdo projekcijoje. Aukščiausi burnos klasifikacijos taškai tarp visų pasuktų pavyzdžių kartu su SVM klasifikatoriais yra naudojami burnos tiksliai vietai nustatyti [9].

SVM filtrų apmokymui yra naudojami teigiami pavyzdžiai. Jie susideda iš rankiniu būdu priskirtų burnos vaizdų rinkinio bei neigiamų pavyzdžių rinkinio (veido regionas, išskyrus burną) [9]. Naudojant priskirtus taškus teigiamame rinkinyje, įtraukiant kampus, aukštesnius ir

žemesniuosius taškus, burnos centą, apmokymo rinkinio dydis padidinamas su atskirais geometriniais variantais, tokiais kaip mastelio parinkimas, horizontalus atvaizdavimas ir vaizdų projekcijos pasukimai. Tuomet prieš-procesinis žingsnis, susidedantis iš mastelio normalizacijos, histogramų sulyginimo ir ryškumo korekcijos yra taikomas abiem tiek teigiamais, tiek neigiamais apmokymo rinkinio pavyzdžiams. Gauso branduolio SVM filtrų apmokymų procedūra [9]:

1. Apmokyti SVM filtrus, naudojant teigiamus ir neigiamus apmokymo rinkinius.
2. Paleisti SVM filtrus patvirtintame rinkinyje ir padidinti abiejų tiek teigiamo rinkinio su neaptiktais burnos regionais, tiek neigiamo rinkinio su klaidingais signalais.
3. Kartoti 1 ir 2 veiksmus tol, kol burnos detektoriai pasiekia norimą rezultatą.

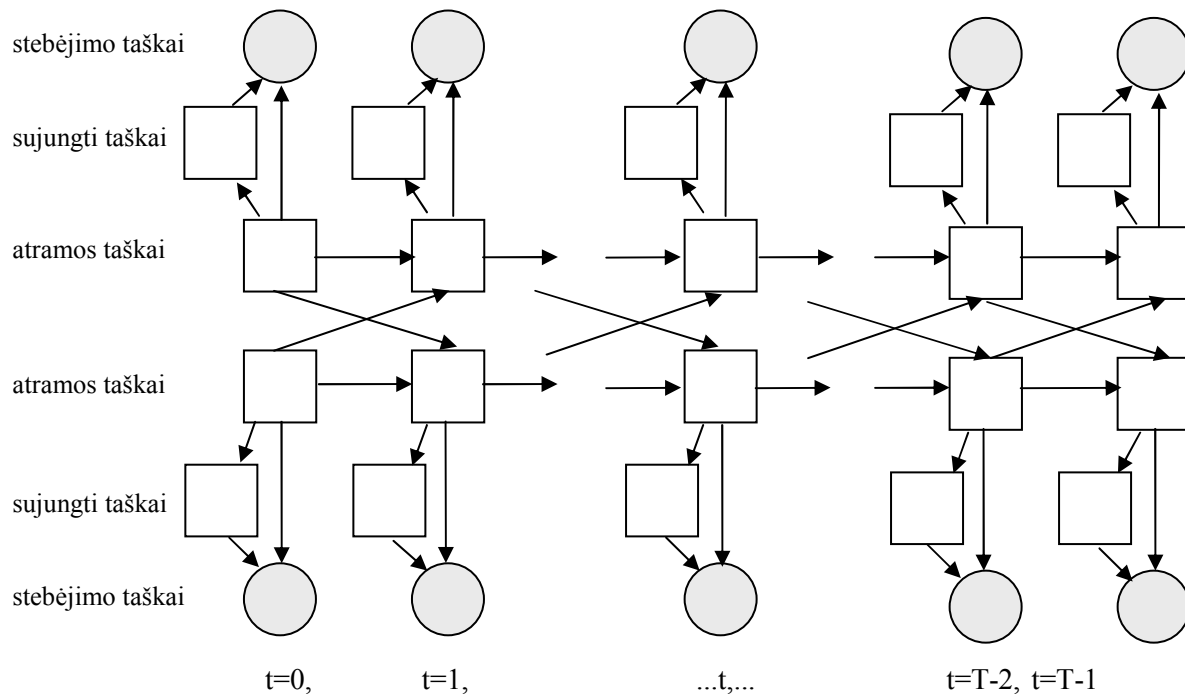
3.2. Audiovizualinis modelis

Audio ir vizualiniai stebėjimo vektoriai yra toliau integruojami naudojant CHMM. CHMM gali būti laikomas kaip paslėptų Markovo modelių kolekcija, viena kiekvienam duomenų srautui, kur paslėpti atramos taškai laikui t kiekvienam HMM, yra sąlygojami atramos taškų laikui $t - 1$ iš visų susijusių HMM. 3.2 pav. [9] iliustruoja nepertraukiamą dviejų srautų kombinuotą HMM, naudojamą audiovizualinės kalbos atpažinimo sistemoje. Kvadratai reiškia paslėptus diskretinius taškus, o apskritimai apibūdina nepertraukiamus matomus taškus. CHMM leidžia audio ir vizualinių būsenų asinchroniškumą, tuo tarpu išlaikant natūralią audio ir vizualinės sekos priklausomybę. Papildant, su dvigubu HMM audio ir video tikimybių stebėjimas yra apskaičiuojamas nepriklausomai bei žymiai mažinant parametrų erdvę. Šito modelio sudėtingumas palyginamas su modeliais, kurie reikalauja sąryšio tarp audio ir vizualinių stebėjimų. Audiovizualinės kalbos atpažinimo sistemoje CHMM modeliuoja kiekvieną vizemos-fonemos porą. Parametrų apmokymas prasideda nuo Viterbi metodu pagrįstos iniciacijos, naudojantis įvertinimo-maksimizacijos (EM) algoritmu atskiriems žodžiams. Toliau, visi CHMM parametrai yra įvertinami kartu iš nepertraukiamos audiovizualinės kalbos, naudojant EM algoritmą. Nepertraukiamos kalbos atpažinimas yra atliekamas diagramos dekoderiu, naudojant vienos rūšies Viterbi spindulio tyrinėjimus. Atpažinimo stadijoje audio ir video stebėjimo galimybės yra modifikuojamos valdyti skirtingus triukšmo lygius [9]

$$\bar{b}_t^a(i) = b_t(O_t^a | q_t^a = i)^{\alpha_a} \quad (3.1)$$

$$\bar{b}_t^v(i) = b_t(O_t^v | q_t^v = j)^{\alpha_v} \quad (3.2)$$

čia $\alpha_a + \alpha_v = 1$, $\alpha_a, \alpha_v \geq 0$ yra audio ir video srautų eksponentės, o α_a, α_v - vertės, atitinkančios specifinį akustinį SNR lygį, yra gaunamos eksperimento būdu, maksimizuojant vidutinį atpažinimo greitį [9].



3.2 pav. Audiovizualinis keletas HMM modelis

4. DARBE TAIKYTI METODAI

4.1. Plėtimas (dilate)

Plėtimas yra morfologinė transformacija, kuri apjungia dvi aibes, naudojant vektorių sumavimą iš aibių elementų (4.1 pav. [10]). Tarkime, objektas A ir struktūrinis elementas B yra dviejų dimensijų aibės. Tuomet A objekto plėtimas pagal B yra apibrėžiamas kaip aibė visų taškų c , kiekvienam $c=a+b$ [10]:

$$\text{Plėtimas: } A \oplus B = \{c | c = a + b \quad a \in A \text{ ir } b \in B\}$$

arba

$$A \oplus B = \bigcup_{x \in B} a + x$$

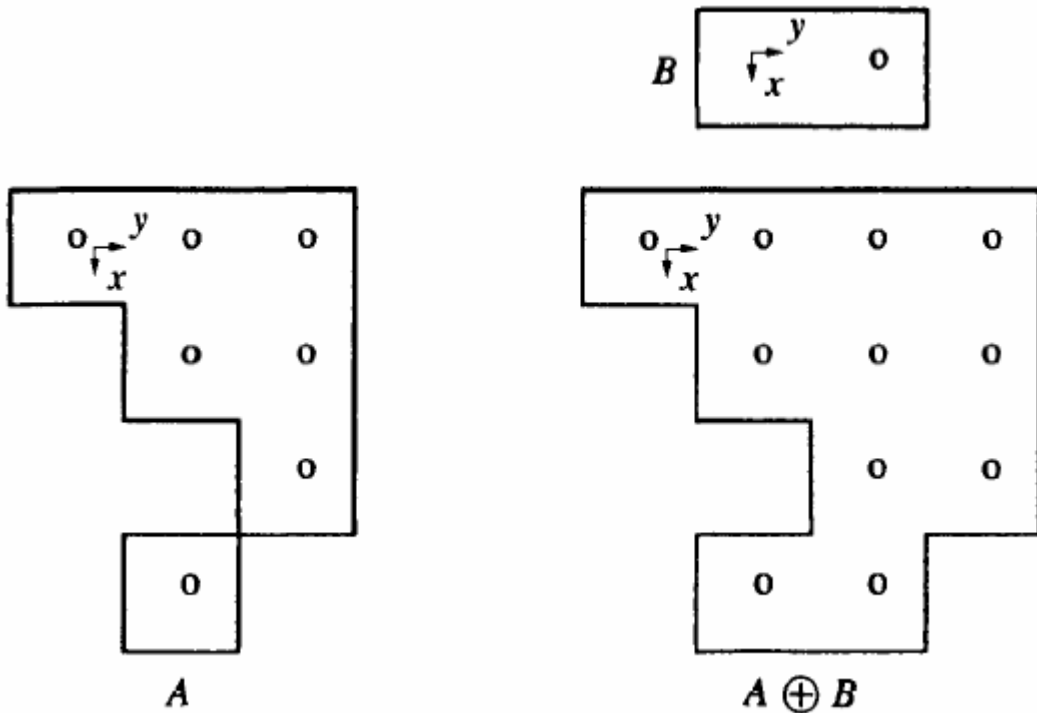
kur $A = \{a_1, a_2, \dots, a_n\}$, ir $B = \{b_1, b_2, \dots, b_N\}$

Tarkime, kad

$$A = \{(0,0), (0,1), (0,2), (1,1), (1,2), (2,2), (3,1)\}$$

$$B = \{(0,0), (0,1)\}$$

$$A \oplus B = \{(0,0), (0,1), (0,2), (1,1), (1,2), (2,2), (3,1), (0,3), (1,3), (2,3), (3,2)\}$$



4.1 pav. Plėtimas

4.2. Eroziija

Eroziija yra morfologinė transformacija, kuri apjungia dvi aibes naudojant vektorių atimtį. Tarkime, objektas A ir struktūrinis elementas B yra dviejų dimensijų aibės (4.2 pav. [10]). Tuomet A objekto eroziija pagal B yra apibrėžiamas kaip aibė visų taškų c , kur $c + b \in A$ kiekvienam $b \in B$ t.y.

$$\text{Eroziija: } A \ominus B = \{c \mid c + b \in A \text{ kiekvienam } b \in B\}$$

arba

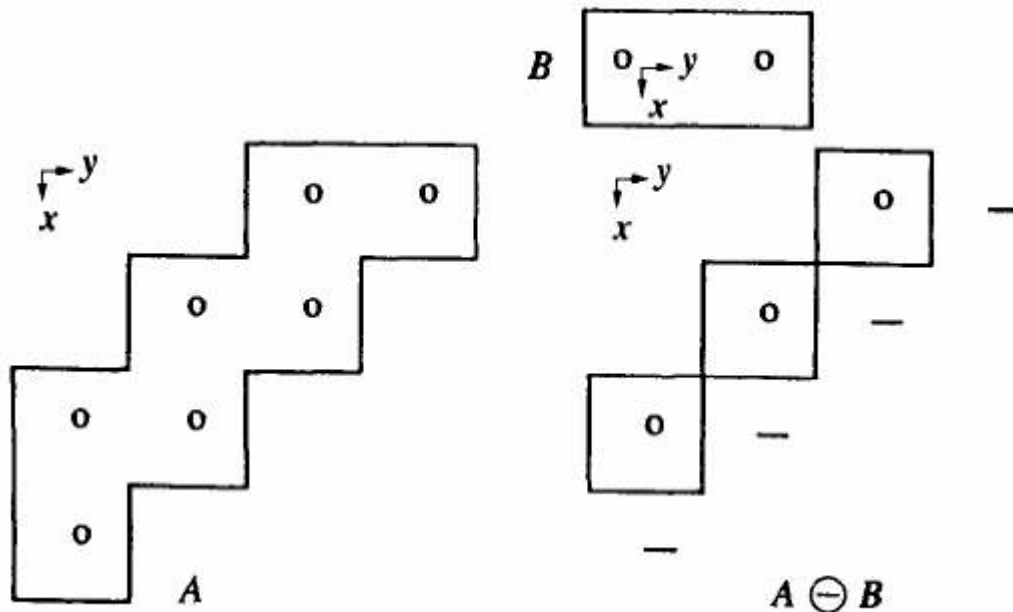
$$\text{Eroziija: } A \ominus B = \{c \mid c = a - b \text{ kiekvienam } b \in B\}$$

čia $A = \{a_1, a_2, \dots, a_n\}$, ir $B = \{b_1, b_2, \dots, b_N\}$

$$A = \{(0,2), (0,3), (1,1), (1,2), (2,0), (2,1), (3,0)\}$$

$$B = \{(0,0), (0,1)\}$$

$$A \ominus B = \{(0,2), (1,1), (2,0)\}$$



4.2 pav. Eroziija

Dažniausiai šios abi operacijos yra naudojamos kartu. Tarkime, atliekamas paveiksluko plėtimas prieš tai atlikus jo eroziiją – tai vadinama uždarymu. Galimas ir atvirkštinis variantas, po paveiksluko plėtimo atlikti jo eroziiją – tai vadinama atidarymu. Šiais būdais galima eliminuoti specifines paveiksluko detales, kurios yra mažesnės už struktūrinį elementą [11].

4.3. Dviejų dimensijų Gauso funkcija

Funkcijos išraiška [10]

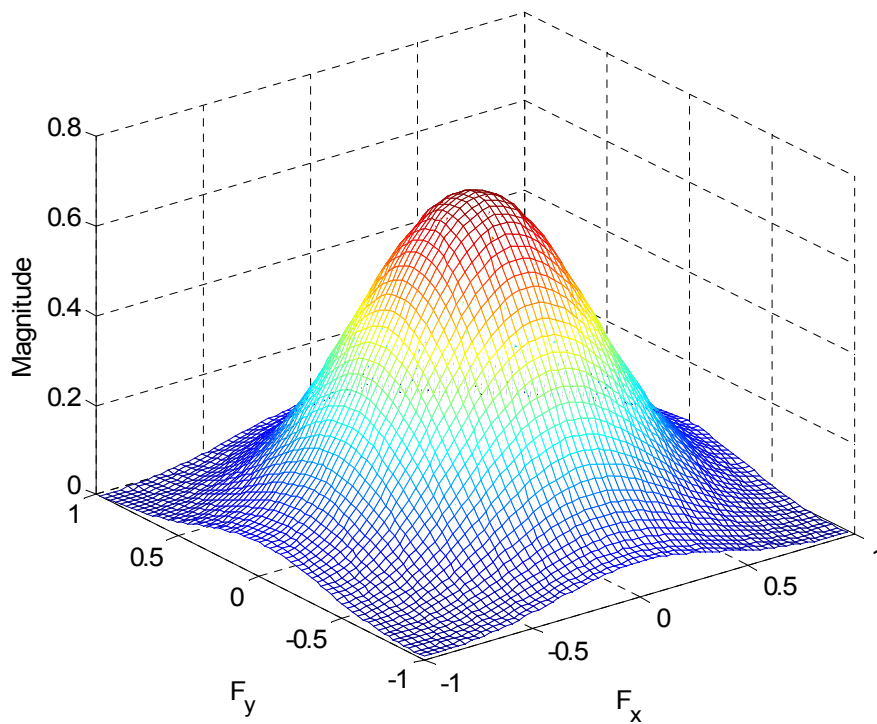
$$f(n_1, n_2) = Ae^{-\left(\left(\frac{n_1-x_0}{\sigma_x}\right)^2 + \left(\frac{n_2-y_0}{\sigma_y}\right)^2\right)} \quad (4.1)$$

čia koeficientas A amplitudė, x_0, y_0 yra centras, n_1, n_2 koordinatės.

Šiuo atveju special funkcija sukuria Gauso filtrą (4.3 pav., [10]), naudodama tokią išraišką:

$$h_g(n_1, n_2) = e^{-\frac{(n_1^2 + n_2^2)}{2\sigma^2}} \quad (4.2)$$

$$h(n_1, n_2) = \frac{h_g(n_1, n_2)}{\sum_{n_1} \sum_{n_2} h_g} \quad (4.3)$$



4.3 pav. Gauso filtras

4.4. Dvimatė diskretinė Furjė transformacija

Išreiškiame per dvimatį signalą, aprašantį diskretinį vaizdą su I_1 eilučių ir I_2 stulpelių [18]:

$$x(i_1, i_2) = x_{i_1, i_2}, \quad i_1 = \overline{0, I_1 - 1}, \quad i_2 = \overline{0, I_2 - 1} \quad (4.4)$$

Įvedame dvimatį periodinį signalą:

$$\bar{x}_{i_1, i_2} = \sum_{m_1=-\infty}^{\infty} x_{i_1 - m_1 I_1, i_2 - m_2 I_2} \quad (4.5)$$

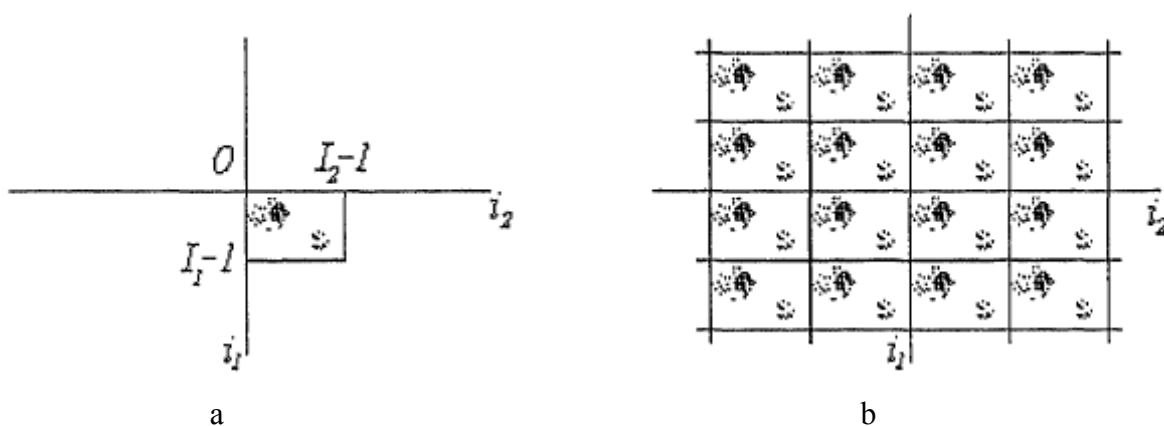
Jei signalas x_{i_1, i_2} yra tik stačiakampio viduje:

$$R_{i_1 i_2} = \{(i_1, i_2): 0 \leq i_1 \leq I_1 - 1, 0 \leq i_2 \leq I_2 - 1\} \quad (4.6)$$

Su ribomis $I_1 \times I_2$ elementų (4.4 pav., a) [18] tuomet signalas \bar{x}_{i_1, i_2} apibrėžiamas visoje plokštumoje (i_1, i_2) ir yra šioje plokštumoje stačiakampio-periodinis (4.4 pav., b) [18].

Bet kuris periodinis signalas gali būti aprašomas Furjė eilute. Dvimatis signalas Furjė eilute aprašomas taip:

$$\tilde{x}_{i_1, i_2} = \frac{1}{I_1 I_2} \sum_{k_1=0}^{I_1-1} \sum_{k_2=0}^{I_2-1} X_{k_1, k_2} \exp\left(j \frac{2\pi}{I_1} i_1 k_1 + j \frac{2\pi}{I_2} i_2 k_2\right), \quad j = \sqrt{-1} \quad (4.7)$$



4.4 pav. Realus periodiškai pratęstas vaizdas

4.5. OpenCV. Adaboost algoritmas

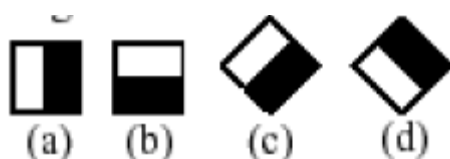
Norint aptikti žmogaus veido bruožus, tokius kaip burna, akys, nosis, reikia pirmiausia apmokyti Haar klasifikatorių kaskadas. Geriausia priemonė yra paprastas AdaBoost algoritmas ir Haar bruožų algoritmas. Tam, kad palengvintų šių algoritmų įgyvendinimą, Intel sukūrė atviro kodo biblioteką, kuri vadinama Atvira kompiuterinės regos biblioteka (Open Computer Vision Library – OpenCV) OpenCV biblioteka.

4.5.1. Haar kaskadų klasifikatoriai

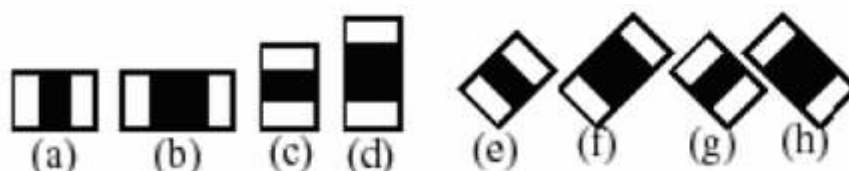
Pagrindiniai objekto aptikimo klasifikatoriai yra haar-like bruožai. Šie bruožai, dažniau nei pikselio intensyvumo reikšmes naudoja kontrasto pasikeitimo reikšmes tarp gretimų stačiakampių grupių pikselių. Kontrasto skirtumai tarp pikselių grupių naudojami tam, kad būtų galima nustatyti šviesių ir tamsių plotų santykį. Haar-like (4.5 pav. [15]) bruožai yra naudojami vaizdo aptikimui.

Haar bruožus galima lengvai skirstyti pagal pikselių grupės, kuri nagrinėjama, dydžio mažinimą ar didinimą. Tai leidžia naudoti bruožus aptinkant įvairaus dydžio objektus.

1. Briunos bruožai



2. Linijos bruožai



3. Centro bruožai



4.5 pav. Dažniausi Haar bruožai

4.5.2. Dirbtinis neuroninis tinklas

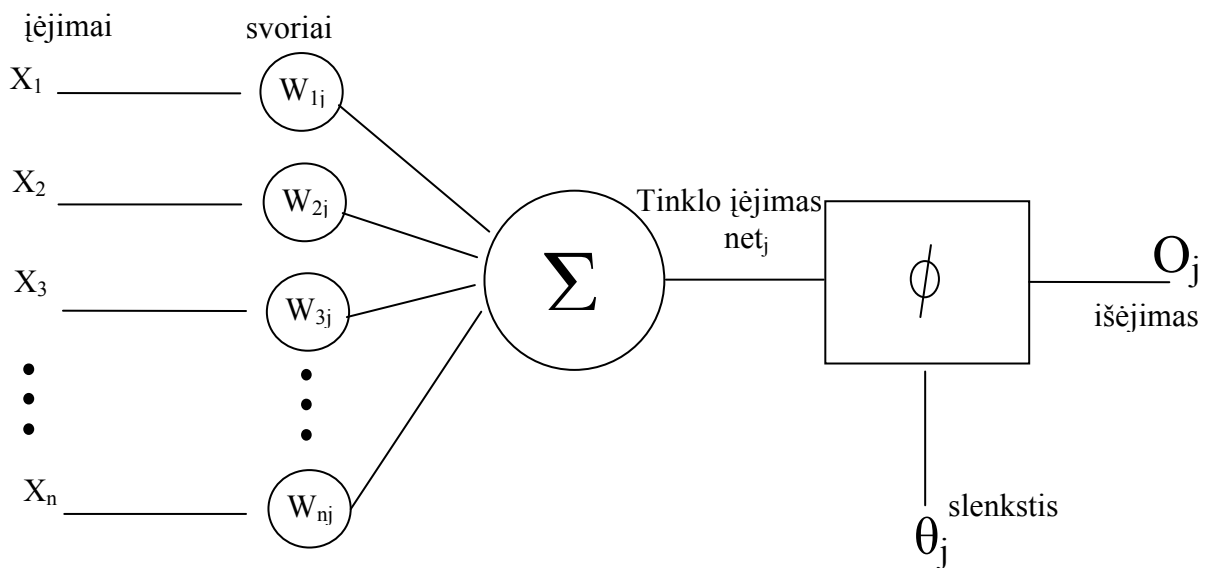
Dirbtinis neuroninis tinklas – tarpusavyje sujungtų dirbtinių neuronų grupė. Ši technologija mėgdžioja žmogaus galvos smegenų darbą – tiksliau neuronų veikimą [14].

Dirbtinis neuronas – biologinio neurono abstrakcija, pagrindinis dirbtinių neuroninių tinklų komponentas. Dirbtinis neuronas turi keletą įėjimų ($x_0, x_1, x_2, \dots, x_n$) ir vieną išėjimą (y). Išėjimo reikšmė gaunama pagal formulę [14]:

$$y = \varphi \left(\sum_{j=0}^n w_j x_j \right), \quad (4.8)$$

čia koeficientai w vadinami įėjimų svoriais, o funkcija φ – aktyvavimo funkcija.

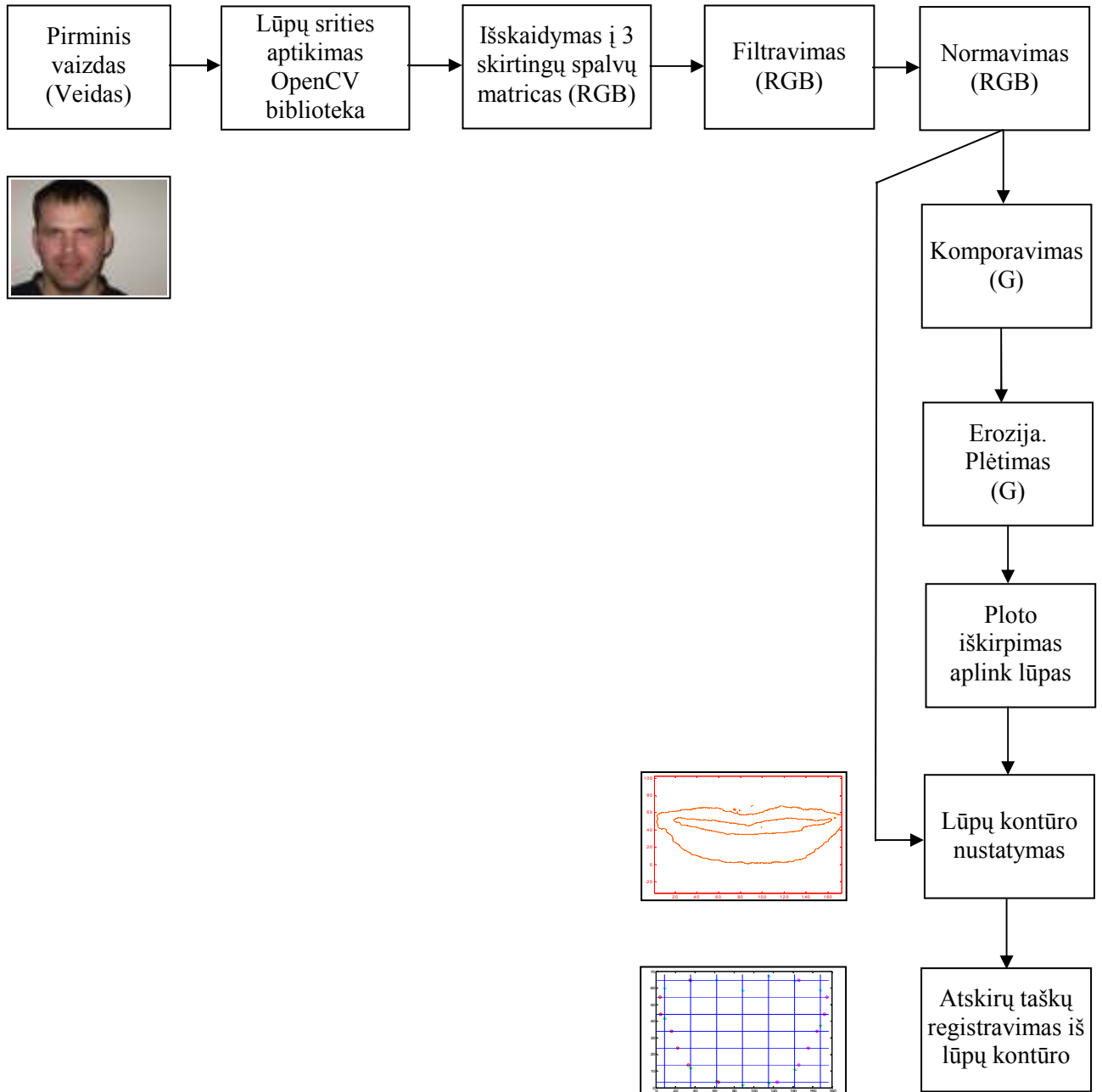
Esama įvairių rūšių neuroninių tinklų. Vieni iš jų apmokomi su mokytoju (žinant reikiamas tinklo išėjimų reikšmes), kiti – be mokytojo. Šiuos neuroninius tinklus galima panaudoti grafinių vaizdų, lūpų formos bei vizualinės kalbos atpažinimui (4.6 pav. [14]).



4.6 pav. Dirbtinio neurono schema

4.6. Praktinė dalis

Sudaromas algoritmas kalbančiojo lūpų formos registravimui (4.7 pav.).



4.7 pav. Lūpų formos registravimo algoritmas

Praktinė dalis buvo atliekama pasitelkus OpenCV ir *MathWorks, Inc* MATLAB 7.0 paketą. Sudarytas algoritmas kalbančiojo lūpų formos registravimui ir taškų iš lūpų kontūro formos atrinkimui. Algoritmo patikrinimui buvo panaudoti *.jpg formato paveiksliukai, kurių raiška 1024x768. Darbe buvo ištirti 4 tiriamieji. Kaip aptinkama lūpų forma (kontūras) esant skirtingai jų padėčiai. Tirtos 4 padėties, kurios parodytos 4.1 lentelėje. Paveiksiai po normavimo, komporavimo ir lūpų kontūrai pavaizduoti Priede Nr. 1.

4.1 lentelė

Tiriamieji

Tiriamieji	Lūpų padėtis		Tariama raidė	
	sučiauptos	nesučiauptos	a	o
Tiriamasis 1	sučiauptos	nesučiauptos	a	o
Tiriamasis 2	sučiauptos	nesučiauptos	a	o
Tiriamasis 3	sučiauptos	nesučiauptos	a	o
Tiriamasis 4	sučiauptos	nesučiauptos	a	o

1. Lūpų aptikimas AdaBoost algoritmu, naudojantis OpenCV biblioteka.

Siekiant aptikti lūpų regioną naudotasi AdaBoost algoritmu. OpenCV turi veido aptikimo apmokytus klasifikatorius. Pasinaudojant šia biblioteka buvo apmokomi klasifikatoriai lūpų atpažinimui:

1.1 Paimama daug neigiamų paveiksliukų pavyzdžių, t.y. paveiksliukų, kuriuose nėra pavaizduota lūpų. Buvo panaudoti 3286 neigiami pavyzdžiai. Taip pat būtini ir teigiami paveiksliukai, t.y., kuriuose yra pavaizduotos lūpos. Buvo panaudoti 491 teigiami pavyzdžiai.

1.2 Createsamples programa iš paveiksliukų sukuriama vektorių failas (vec-file), kurio parametras -num 491 yra teigiamų paveiksliukų skaičius. -w 20 -h 20 pavyzdžių dydis.

```
createsamples.exe -info positive/info.txt -vec data/vector.vec -num 491 -w 20 -h 20
```

1.3 Kuomet buvo sukurtas vektorių failas (vec-file), pradėtas apmokymas programa haartraining.exe.

```
haartraining.exe -data data/cascade -vec data/vector.vec -bg negative/infofile.txt -npos 491 -nneg 3286 -nstages 20 -mem 1000 -mode ALL -w 20 -h 20 -nonsym
```

čia

-data data/cascade - kelias, kur patalpinami kaskadų failai. Pasibaigus kiekvienam lygiui sukuriamas AdaBoostCARTHaarClassifier.txt failas

-vec data/vector.vec - failas, kuris buvo sukurtas naudojant createsamples.exe programą;

-bg negative/infofile.txt - neigiamų failų sąrašas;

-npos 491 - teigiamų failų skaičius;

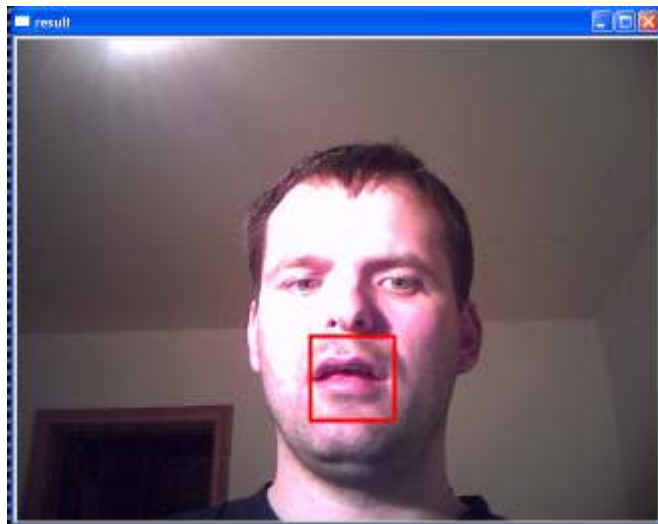
-nneg 3286 - neigiamų failų skaičius;

-nstages 20 - lygis, kuriame sustabdomas apmokymas;

-w 20 -h 20 - pavyzdžio dydis.

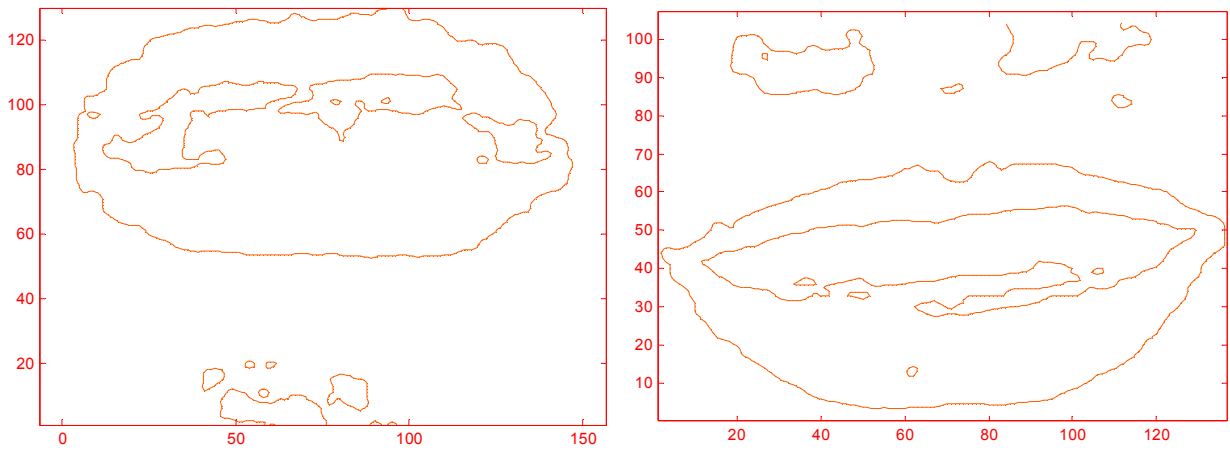
1.4 Šitie kaskadų failai konvertuojami į xml failą, kuris panaudojamas lūpų atpažinimui Haarconv.exe data kaskados.xml

Rezultatas po apmokymo pavaizduotas (4.8 pav.)



4.8 pav. Lūpų aptikimas

Šis pirmas algoritmo veiksmas atliekamas todėl, kad esant panašiai lūpų ir veido spalvai lūpų registravimas pagal RGB spalvas yra sudėtingas. Registruojama ne tik lūpos, bet ir kitos veido dalys tokios kaip smakras, nosis (4.9 pav.). Atlikus šį pirminį veiksmą atmetamas visas fonas ir didžioji veido dalis.



4.9 pav. Lūpų kontūro aptikimas su klaidomis

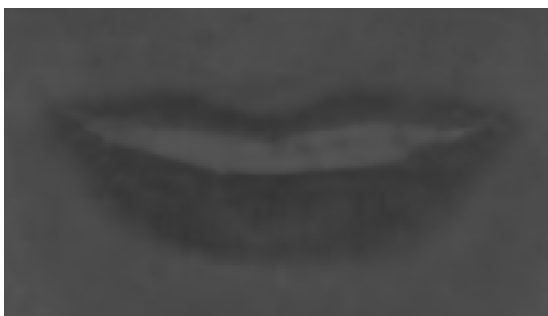
2. *MathWorks, Inc* MATLAB 7.0 paketu atliekamas paveiksluko nuskaitymas ir išskaidymas į tris spalvų komponentių (RGB) matricas `imread` funkcijos pagalba. Kiekvienas spalvoto paveiksluko ar vaizdo kadro pikselis susideda iš RGB komponentių reikšmės (pvz., 255x255x255 atitinka baltą spalvą) (2 PRIEDAS).

3. Atliekamas vaizdo filtravimas pagal atskiras spalvų komponentes (RGB). Pasirinktas dvimatis Gauso filtras.

4. Toliau atliekamas atskirų vaizdo komponentių (RGB) normavimas (4.10 pav.). Kadangi be normavimo lūpų registravimas netikslus, labai priklausantis nuo apšvietimo, krintančių šešėlių.

$$q=r+g+b \quad (4.9)$$

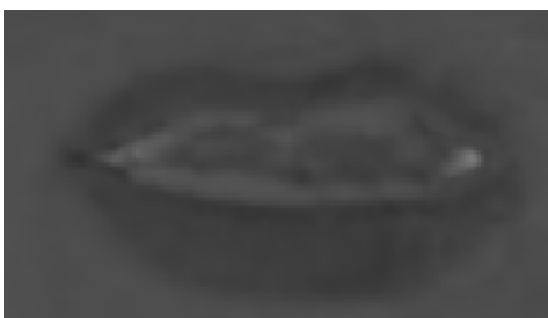
$$r_2 = r/q, \quad g_2 = g/q, \quad b_2 = b/q \quad (4.10)$$



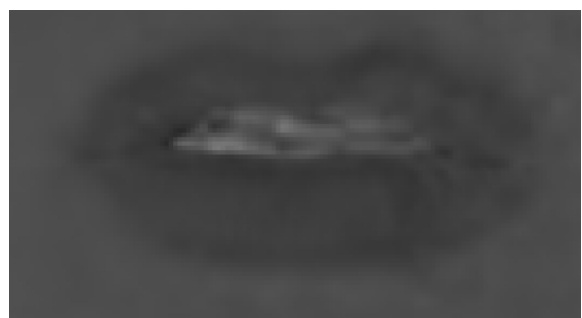
nesučiauptos



sučiauptos



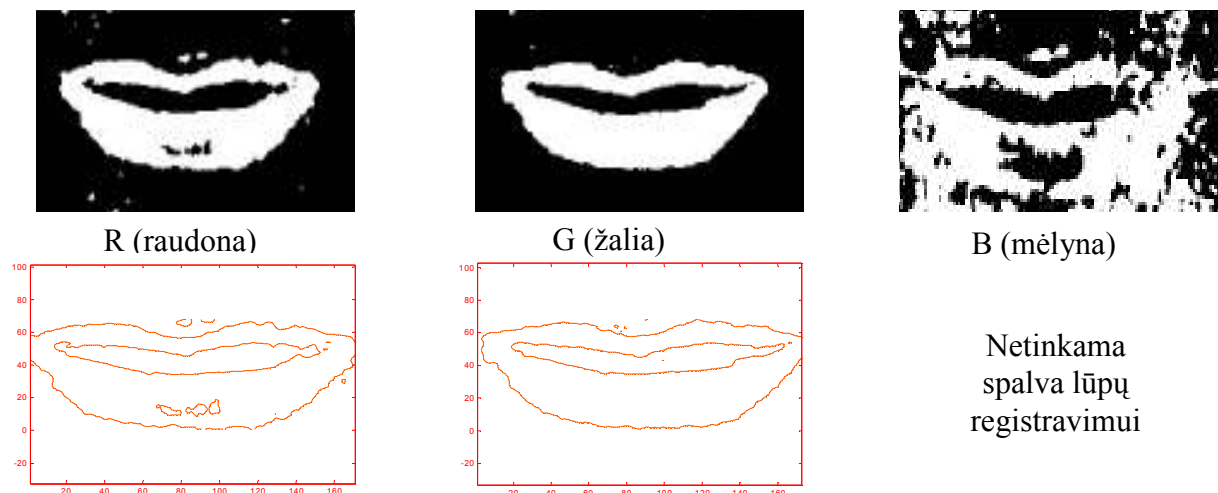
a raidė



o raidė

4.10 pav. Lūpos po filtravimo ir normavimo (G)

5. Atliekamas komporavimas (sulyginimas) pagal atskiras spalvų komponentes. Šiame etape pastebimas netikslus lūpų registravimas pagal mėlyną (B-blue) spalvą (4.11 pav.). Tolesniame nagrinėjime ši spalva atmetama. Kadangi geriausi rezultatai buvo gauti naudojant žalią G (green) spalvų komponentę, tolesni skaičiavimai atliekami naudojant būtent šią spalvą (4.12 pav.).

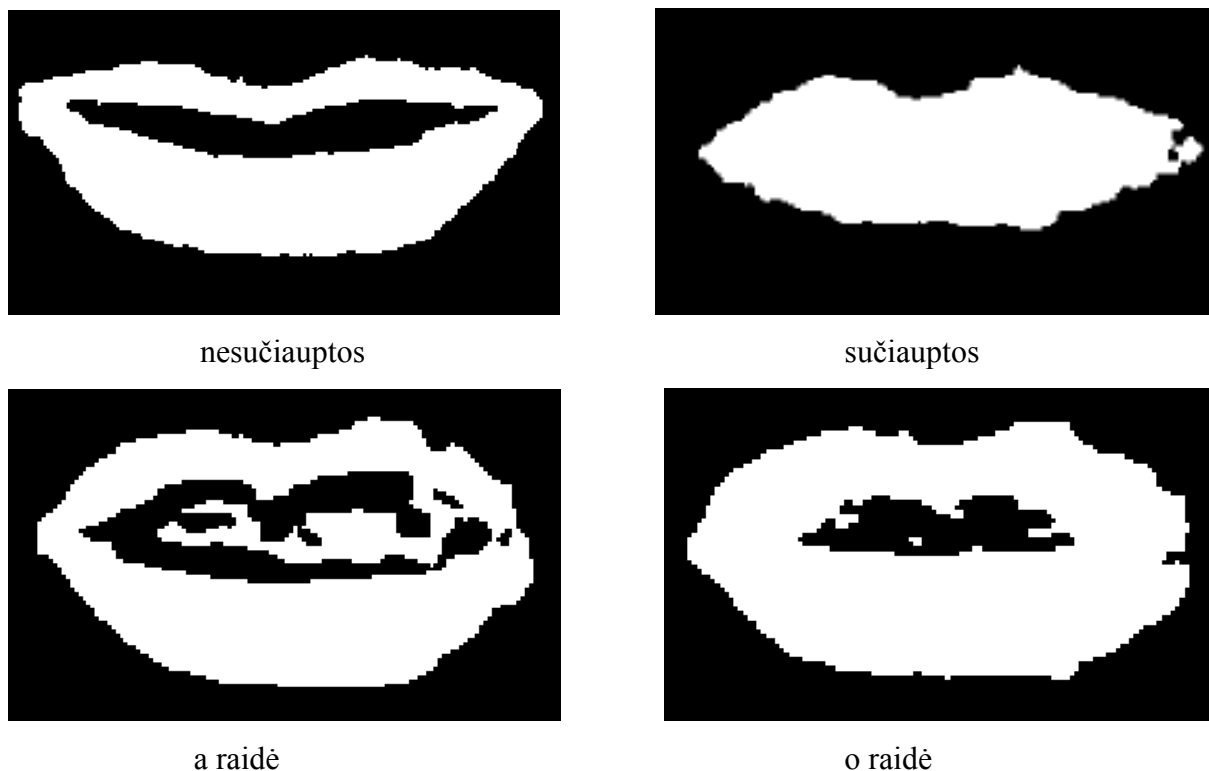


4.11 pav. Lūpų registravimas pagal atskiras spalvų komponentes (RGB)

$$g_3 = g_2 < x \quad (4.12)$$

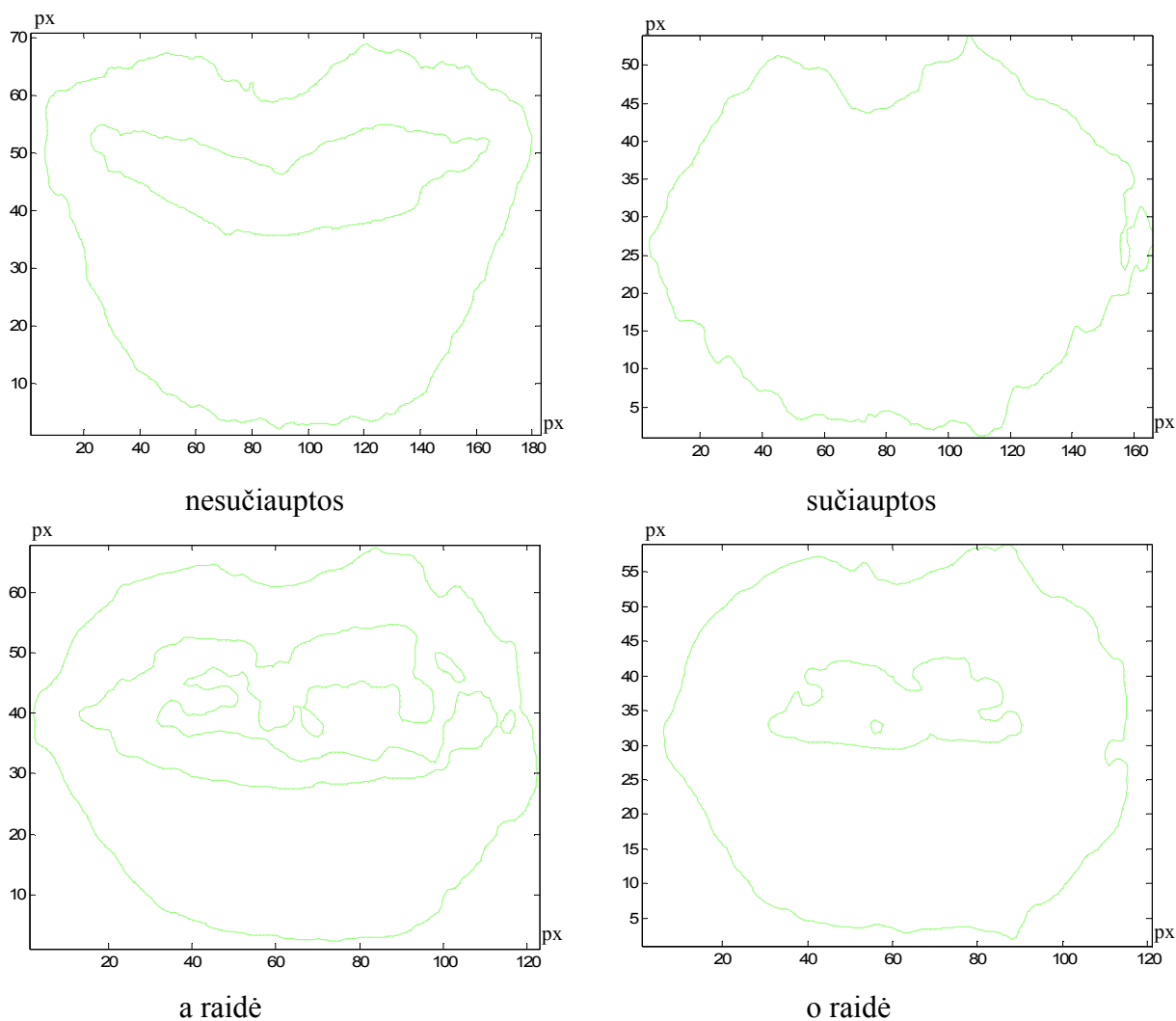
x reikšmė buvo pasirinkta 0.274. Jei g_3 mažesnis už šią reikšmę, tai traktuojama kaip lūpos.

Pavaizduota 1-jo tiriamojo lūpų registravimas.



4.12 pav. Lūpos po komporavimo (sulyginimo) (G)

6. Atliekamas plėtimas ir erozija tam, kad būtų panaikinami aplink lūpų kontūrą likę maži netikri kontūrai. Kitas žingsnis - ploto aplink lūpas iškirpimas, kuris bus naudojamas lūpų kontūro nustatymui. Surandamas lūpų kontūras (4.13 pav.). Šiuo atveju gaunamas išorinis ir vidinis lūpų kontūrai. Nuspręsta, kad tolesniems tyrinėjimams bus naudojamas tik išorinis lūpų formos kontūras, kadangi jis paprasčiau ir tiksliau surandamas. Vidinį kontūrą sunku rasti dėl vidinių burnos artikuliacijų tokių kaip liežuvis, dantys (PRIEDAS 2).



4.13 pav. Lūpų formos kontūrai (G)

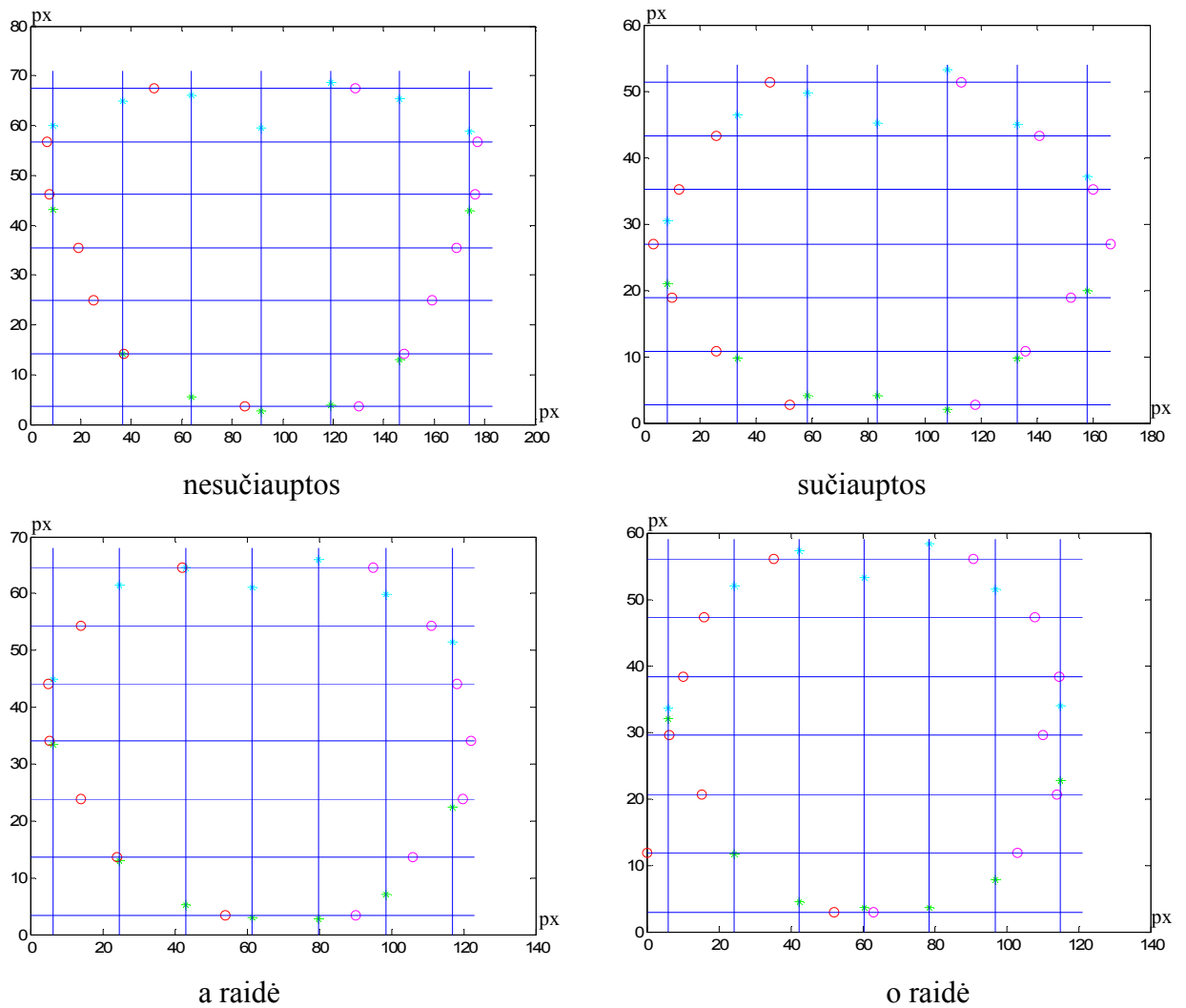
7. Kadangi gautų kontūro reikšmių yra pakankamai daug ir tuo pačiu norint atmesti vidinį kontūrą, nuspręsta išrinkti tik keliolika lūpų kontūro reikšmių tam tikrose vietose. Tam tikslui sudarytas simetriškas tinklelis.

$$tx=(0.05:0.15:0.95)*size(g5,2)$$

$$ty=(0.05:0.15:0.95)*size(g5,1)$$

čia g5 lūpų formos kontūro dydis.

Surandami labiausiai nutolę tinklelio linijų ir kontūro susikirtimo taškai, kuriuos bus galima panaudoti tolesniam tyrinėjimui (PRIEDAS 2). Pvz., nustatyti ar lūpos užčiauptos ar ne, kokia raidė tariama ir t.t (4.14 pav.). Tam panaudoti galima dirbtinius neuroninius tinklus.



4.14 pav. Atrinkti lūpų kontūro taškai (G)

IŠVADOS IR SIŪLYMAI

Norint tiksliai aptikti lūpų sritį veide OpenCV AdaBoost algoritmu, reikia naudoti didelę teigiamų ir neigiamų paveiksliukų duomenų bazę. Tačiau dėl to labai pailgėja apmokymo procesas.

Tiriamajame darbe Matlab 7.0 paketu gauta, kad lūpų formos registravimas sukurtu algoritmu neįmanomas naudojant mėlynos spalvos komponentę.

Tiksliausias kalbančiojo lūpų formos registravimas gautas naudojant žalios spalvos komponentę (G).

Yra sudėtinga išskirti lūpas iš veido dėl panašios veido ir lūpų spalvos.

Problematisškas lūpų vidinių kontūrų išskyrimas dėl matomų artikuliatorių, tokių kaip liežuvis, dantys.

Lūpų formos atpažinimo tikslumas kinta keičiantis apšvietimui (šešėliai).

Norint praktiškai panaudoti šį algoritmą reikia kalbančiojo lūpų formą registruoti video kadru sekoje.

Tolesniame tyrinėjime gauti duomenys gali būti panaudoti neuroninių tinklų apmokymui, kad būtų galima įvertinti, kiek tiksliai įmanoma atpažinti lūpų padėtį, kalbą.

LITERATŪRA

1. Adolf F. How-to build a cascade of boosted classifiers based on Haar-like features. 2003, birželis. [žiūrēta 2007-04-12]. Prieiga per internetu: <<http://portal.acm.org/citation.cfm?id=1127389.1127416&coll=GUIDE&dl=GUIDE&CFID=15151515&CFTOKEN=6184618>>.
2. Belhumeur P. N., Hespanha J. P., Kriegman D. J., Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Tran. on Pattern Analysis and Machine Intelligence. 1997, nr.19(7). [žiūrēta 2006-12-10]. Prieiga per internetu: <<http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=13123ieeexplore.ieee.org/iel1/34/13123/00598228.pdf?arnumber=598228>>.
3. Brand J. D. Visual speech for speaker recognition and robust face detection. University of Wales SWANSEA, 2001, gegužė. [žiūrēta 2006-12-03]. Prieiga per internetu: <<http://citeseer.ist.psu.edu/brand01visual.html>>.
4. He X., Yan S., Hu Y., Niyogi P., Zhang H. J. Face Recognition on Pattern Analysis and Machine Using Laplacianfaces. IEEE Tran. Intelligence. Washington, 2005, nr. 27(3). [žiūrēta 2006-12-19]. Prieiga per internetu: <<http://portal.acm.org/citation.cfm?id=1042399&jmp=cit&coll=GUIDE&dl=GUIDE&CFID=7646709&CFTOKEN=83845909>>.
5. Yeung K. Y., Ruzzo W. L. An empirical study on Principal Component Analysis for clustering gene expression data. Technical Report UW-CSE-01-04-02, University of Washington. 2001. [žiūrēta 2007-01-19]. Prieiga per internetu: <<http://www.cs.washington.edu/homes/ruzzo/papers/pca-bioinf.pdf-pca1.pdf>>.
6. Kaucic R., Dalton B., Blake A. Real-time lip tracking. 1996. [žiūrēta 2007-04-17]. Prieiga per internetu: <<http://citeseer.ist.psu.edu/kaucic96realtime.html>>.
7. Luetin J. Visual speech and speaker recognition. 1997. [žiūrēta 2007-01-26]. Prieiga per internetu: <<http://citeseer.ist.psu.edu/75889.html>>.
8. Luetin J., Thacker A. N., Beet W. S. Speaker identification by lipreading. University of Sheffield. 1996. [žiūrēta 2007-01-09]. Prieiga per internetu: <<http://www.asel.udel.edu/icslp/cdrom/vol1/954/a954.pdf>>.
9. Luhong L., Xiaoxing L., Yibao Z., Xiaobo P., Nefian V. A. Speaker independent audio-visual continuous speech recognition. IEEE International Conference on Multimedia and Expo. Santa Clara, 2002, rugpjūtis [žiūrēta 2007-03-08]. Prieiga per internetu: <http://www.anefian.com/nefian02_speaker.pdf>.

10. Ray Liu K. J. Pattern recognitions and image preprocessing. University of Maryland. 2002. [žiūrēta 2007-01-12]. Prieiga per internetą: <<http://www.ethbib.ethz.ch/etext/sg/098.html>>.
11. Russ J. C. The image processing handbook. 1998, liepa. [žiūrēta 2007-04-12]. Prieiga per internetą: <<http://lib.mexmat.ru/books/10535>>.
12. Tsuban C. Audiovisual speech processing. Lip reading and lip synchronization //IEEE signal processing magazine. 2001, nr.1. [žiūrēta 2007-03-22]. Prieiga per internetą: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=911195>.
13. Turk M., Pentland A. Eigenfaces for recognition. The Journal of Cognitive Neuroscience. 1991, nr.3(1). [žiūrēta 2006-12-19]. Prieiga per internetą: <<http://cs.gmu.edu/~kosecka/cs803/Eigenfaces.pdf>>.
14. Vikipedija, laisvoji enciklopedija. [žiūrēta 2007-05-11]. Prieiga per internetą: <http://lt.wikipedia.org/wiki/Dirbtinis_neuroninis_tinklas>.
15. Wilson P.I., Fernandez J. Facial feature detection using haar classifiers. Texas A&M University, Corpus Christi, 2006, balandis. [žiūrēta 2007-01-25]. Prieiga per internetą: <<http://portal.acm.org/citation.cfm?id=1127416>>.
16. Zakaria A., Bouguet J. Y., Mersereau M. R.. Learning a face model for tracking and recognition. 2002. [žiūrēta 2007-02-12]. Prieiga per internetą: <<http://www.zackvision.com/papers/icassp2002.pdf>>.
17. Вежневцев В., Дегтярева А. Обнаружение и локализация лица на изображении. 2003. [žiūrēta 2007-03-17]. Prieiga per internetą: <<http://cgm.graphicon.ru/content/view/40/60/>> .
18. Грузман И.С., Киричук В.С. Цифровая обработка изображений в информационных системах. 2002. [žiūrēta 2007-04-15]. Prieiga per internetą: <<http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=30766&isYear=2004&count=358&page=8&ResultStart=200>>.

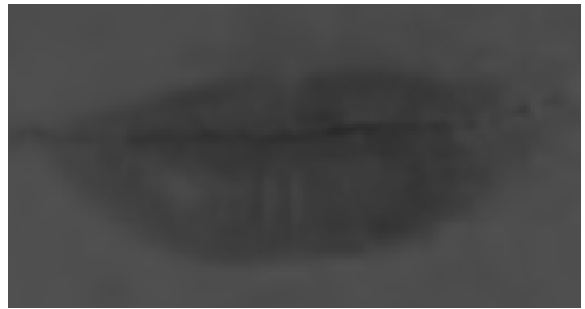
PRIEDAI

1 PRIEDAS

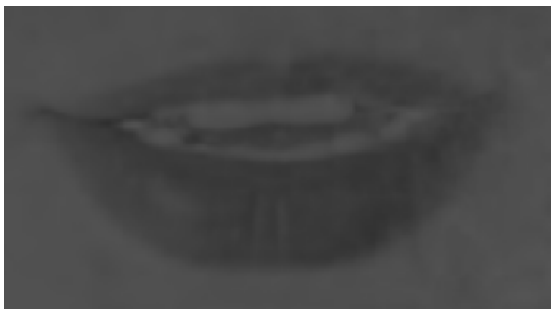
LŪPU FORMOS REGISTRAVIMAS (TIRIAMASIS 2)



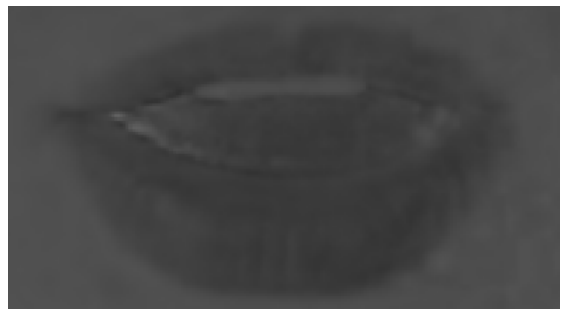
nesučiauptos



sučiauptos



a raidė

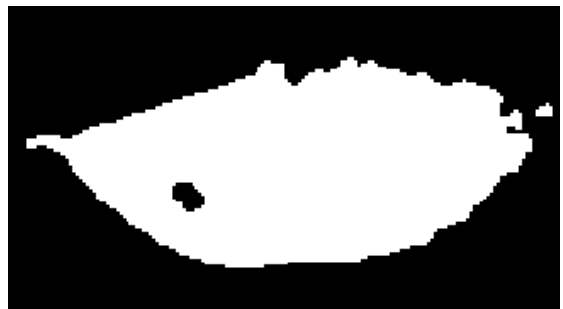


o raidė

P1.1 pav. Lūpos po filtravimo ir normavimo (G)



nesučiauptos



sučiauptos

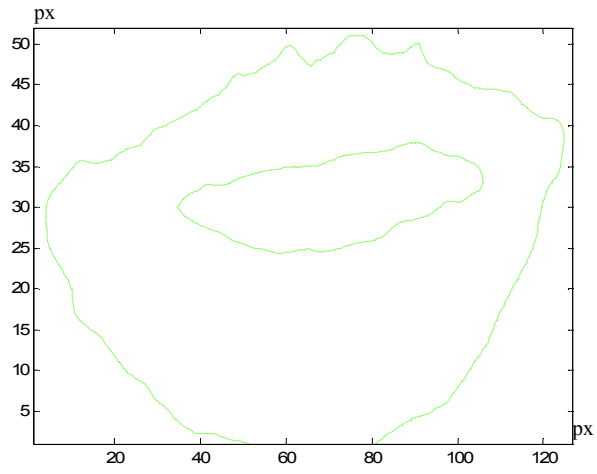


a raidė

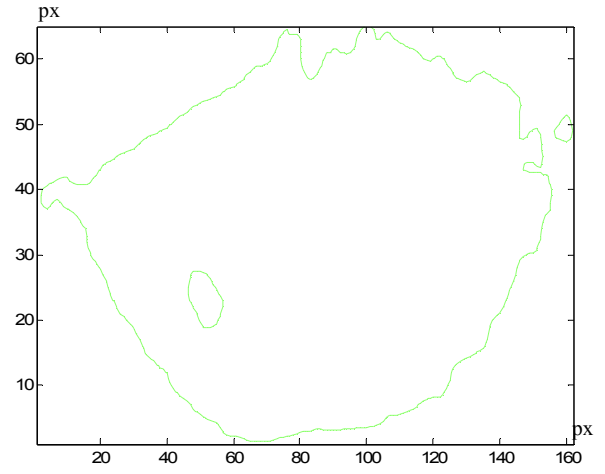


o raidė

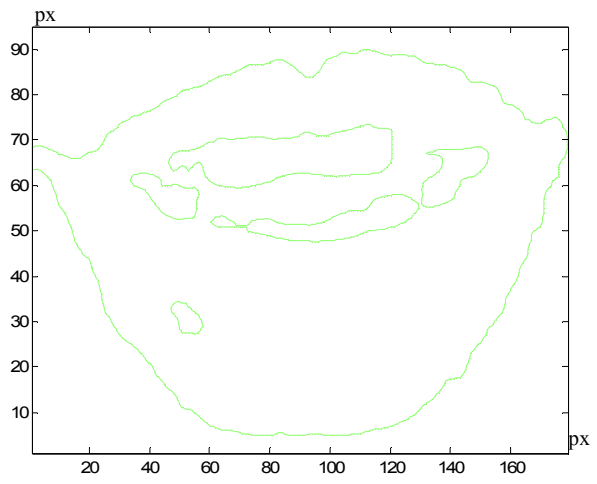
P1.2 pav. Lūpos po komporavimo (sulyginimo) (G)



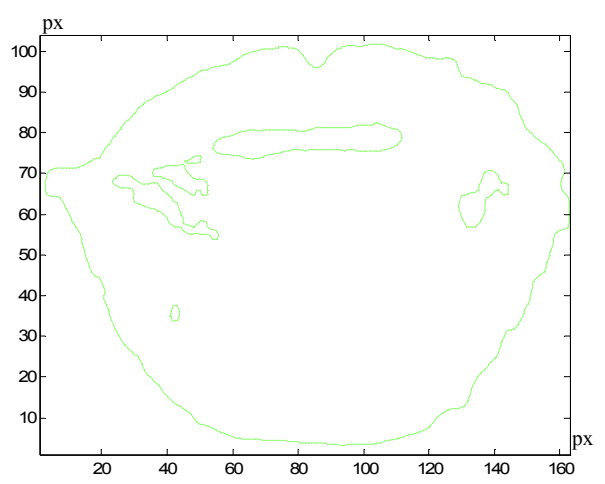
nesučiauptos



sučiauptos

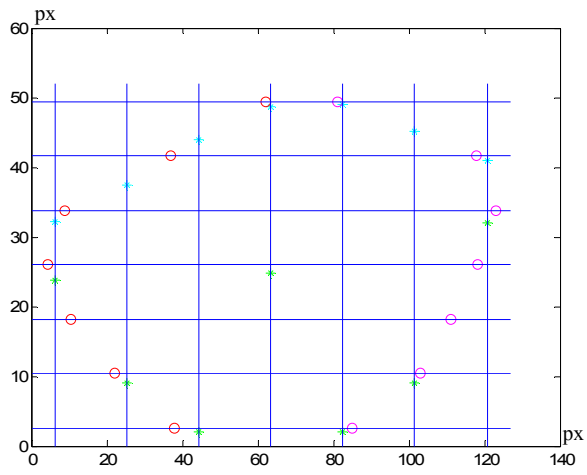


a raidė

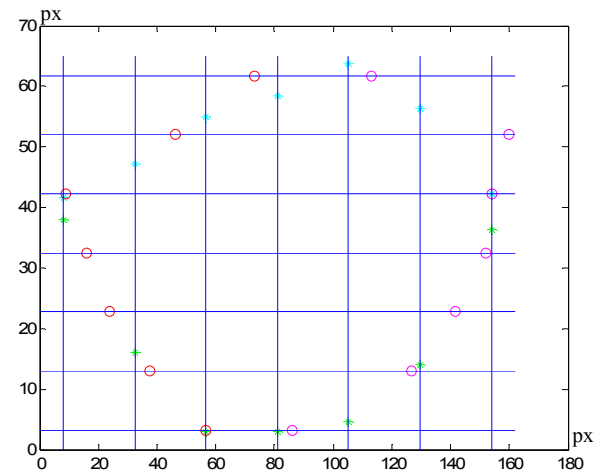


o raidė

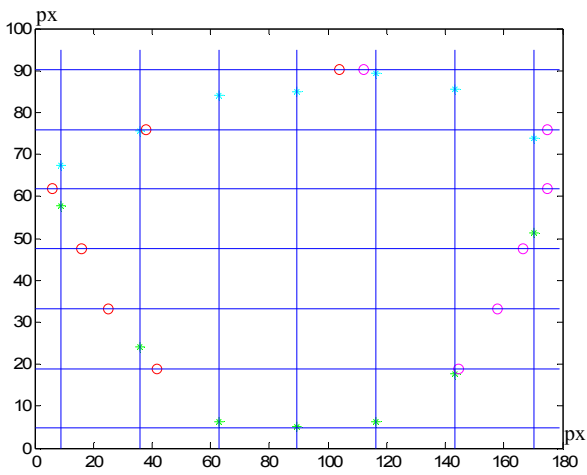
P1.3 pav. Lūpų formos kontūrai (G)



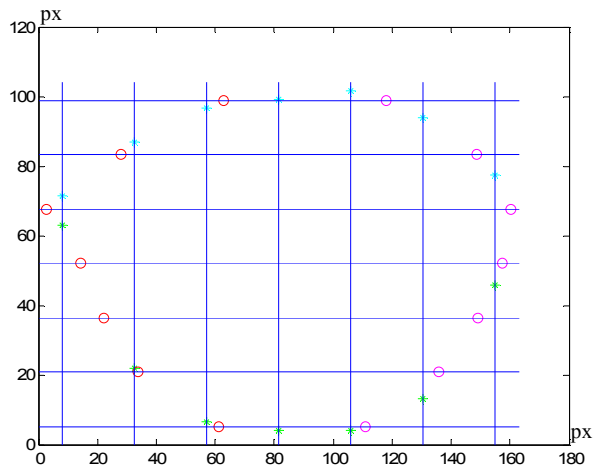
nesučiauptos



sučiauptos



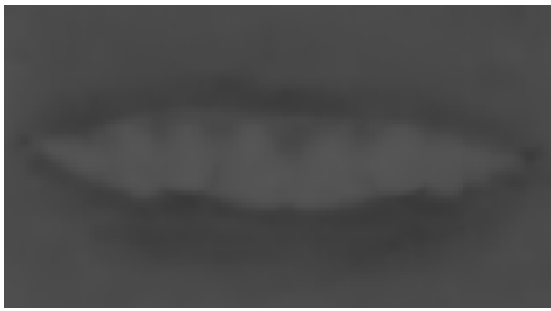
a raidė



o raidė

P1.4 pav. Atrinkti lūpų kontūro taškai (G)

LŪPU FORMOS REGISTRĀVIMAS (TIRIAMĀSIS 3)



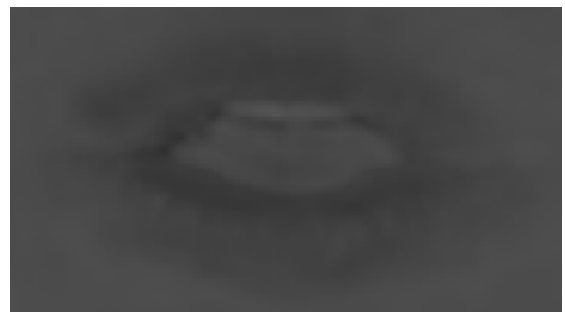
nesučiauptos



sučiauptos



a raidē



o raidē

P1.5 pav. Lūpos po filtravimo ir normavimo (G)



nesučiauptos



sučiauptos

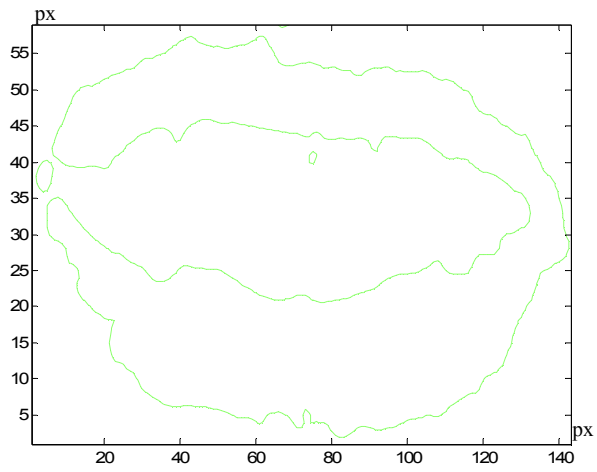


a raidē

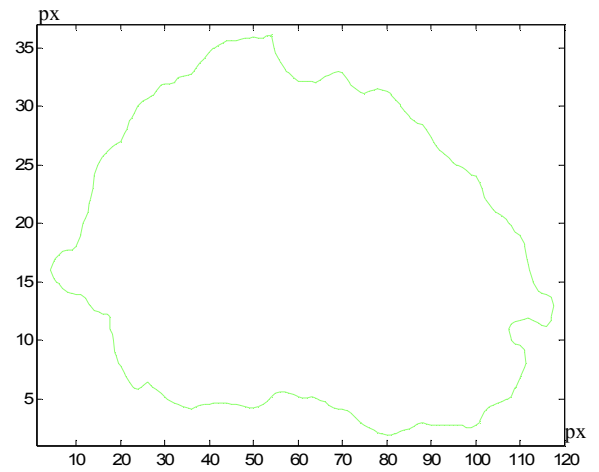


o raidē

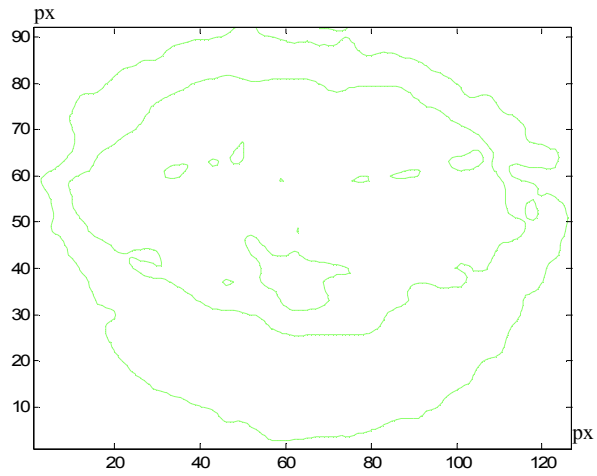
P1.6 pav. Lūpos po komporavimo (sulyginimo) (G)



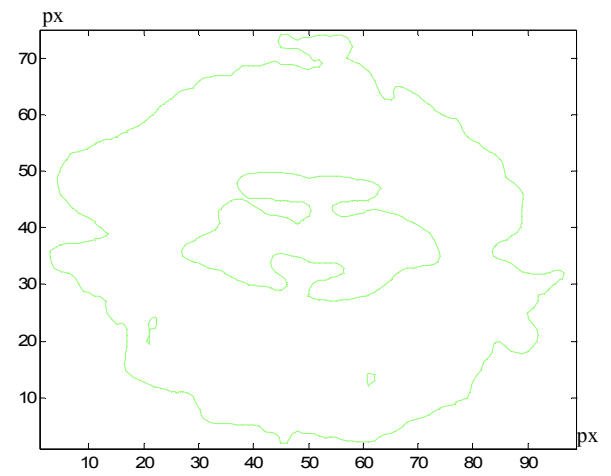
nesučiauptos



sučiauptos

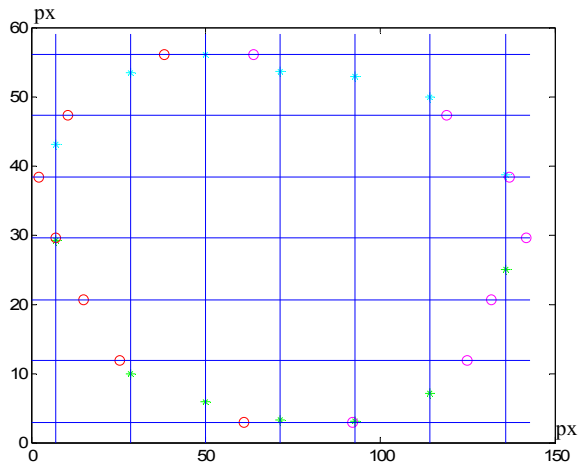


a raidė

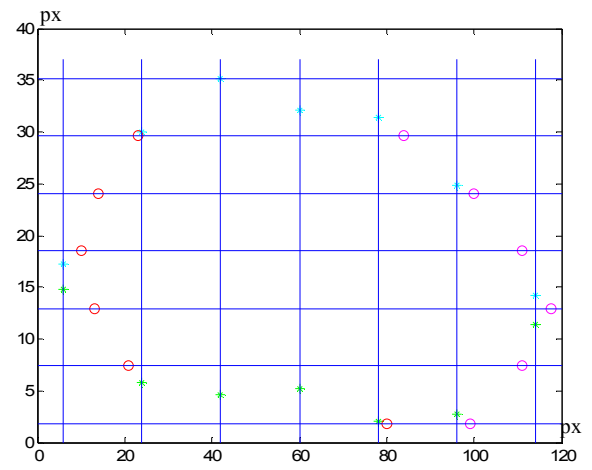


o raidė

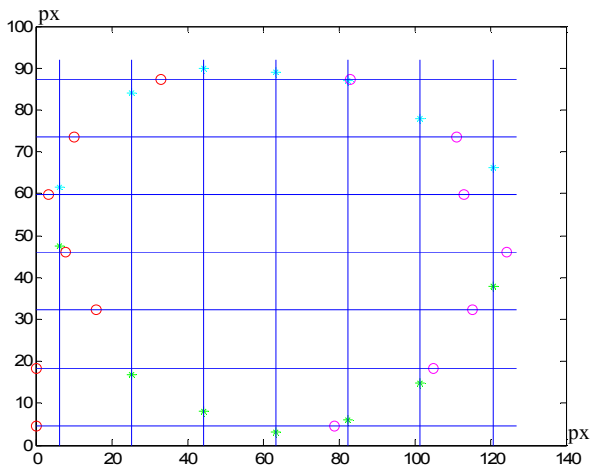
P1.7 pav. Lūpų formos kontūrai (G)



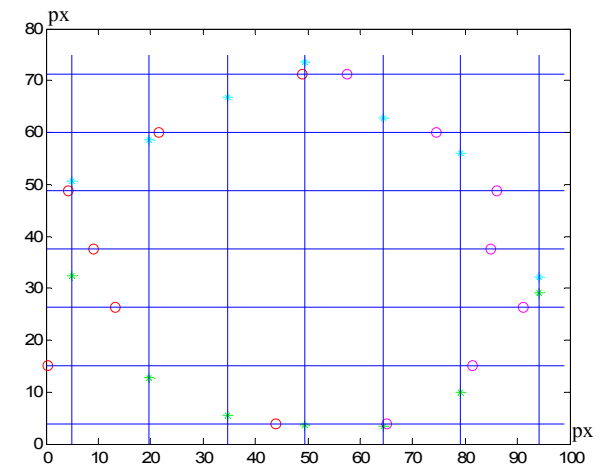
nesučiauptos



sučiauptos



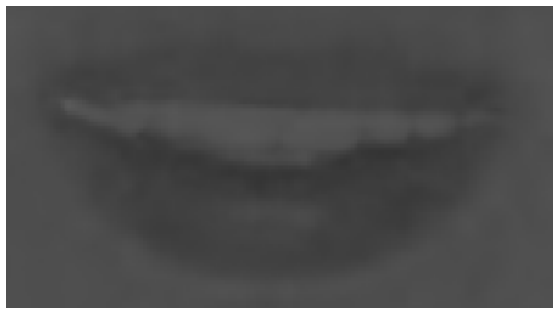
a raidė



o raidė

P1.8 pav. Atrinkti lūpų kontūro taškai (G)

LŪPU FORMOS REGISTRĀVIMAS (TIRIAMĀSIS 4)



nesučiauptos



sučiauptos



a raidē



o raidē

P1.9 pav. Lūpos po filtravimo ir normavimo (G)



nesučiauptos



sučiauptos

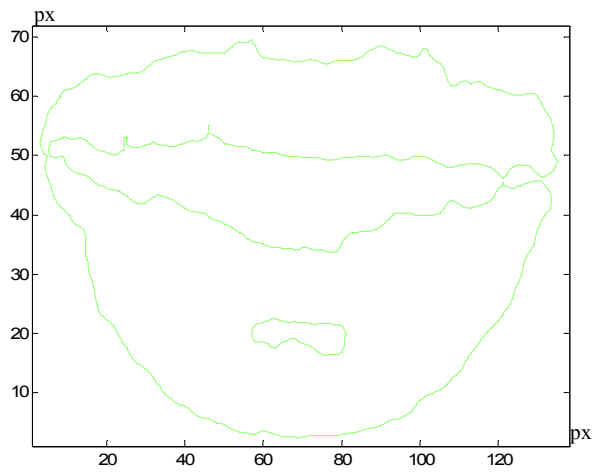


a raidē

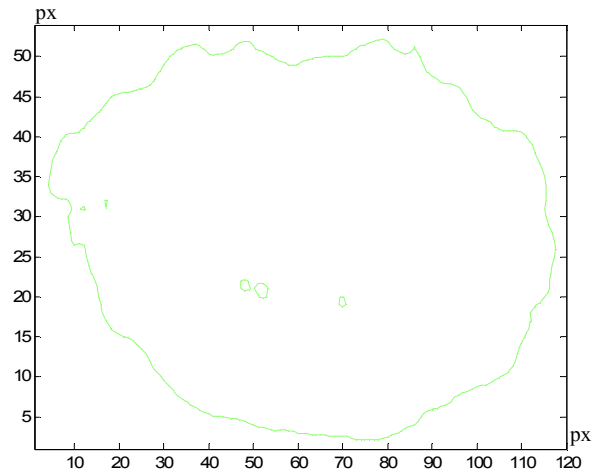


o raidē

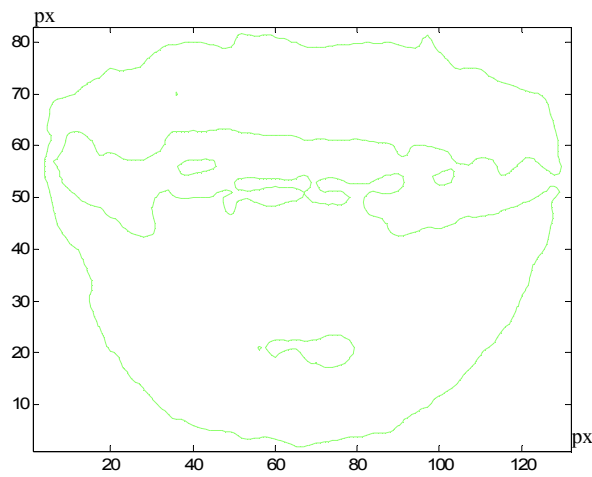
P1.10 pav. Lūpos po komporavimo (sulyginimo) (G)



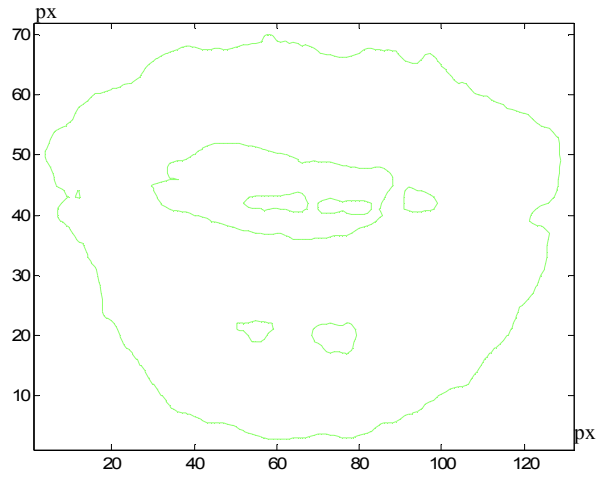
nesučiauptos



sučiauptos

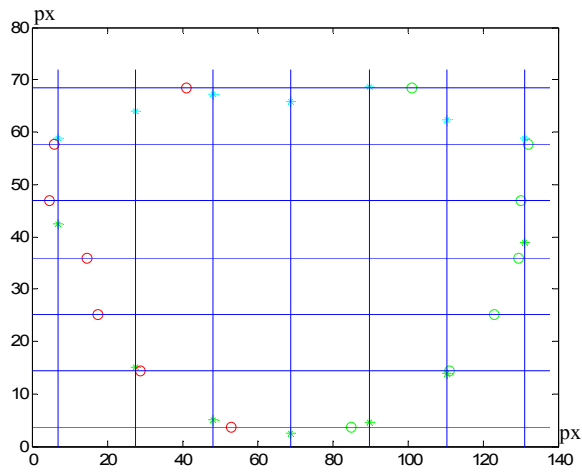


a raidė

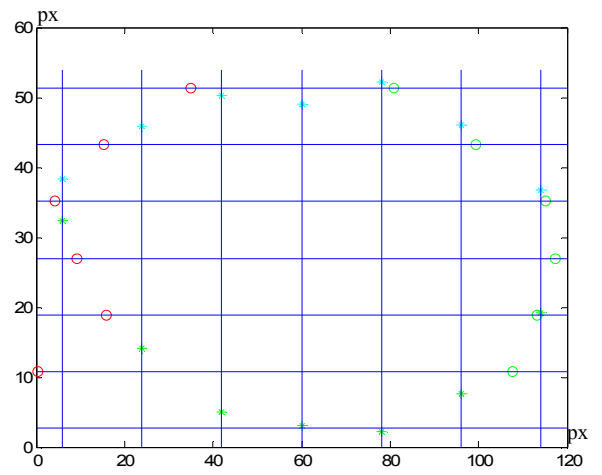


o raidė

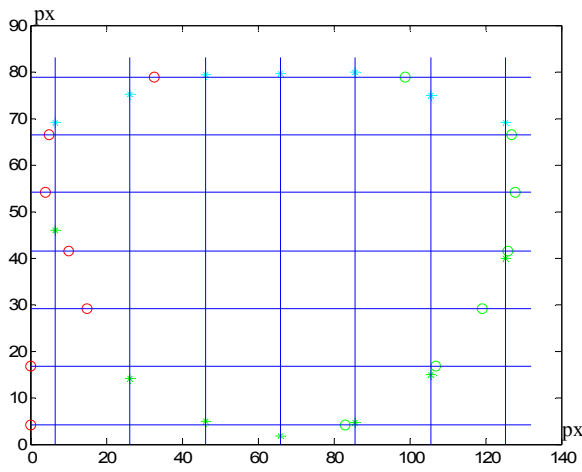
P1.11 pav. Lūpų formos kontūrai (G)



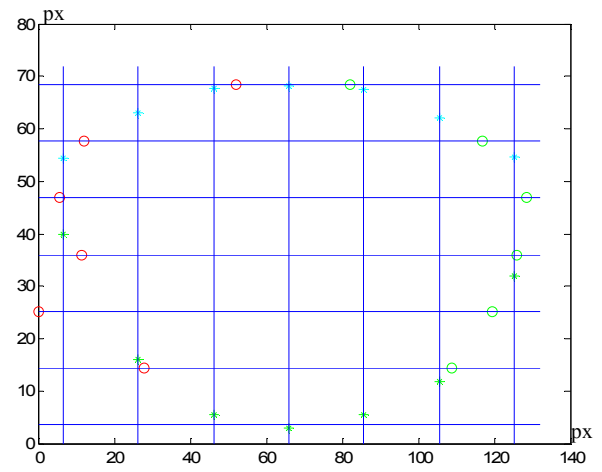
nesučiauptos



sučiauptos



a raidė



o raidė

P1.12 pav. Atrinkti lūpų kontūro taškai (G)

MATHWORKS, INC MATLAB 7.0 PAKETU PARAŠYTA PROGRAMA LŪPŲ
REGISTRAVIMUI

```

% Nuskaitomas paveikslukas;
a=imread('eg1.jpg');
%Kuris yra išskaidomas į skirtingas 3 spalvų RGB matricas;
r=a(:,:,1);
g=a(:,:,2);
b=a(:,:,3);
figure, imshow(a)
figure, imshow(r)
figure, imshow(g)
% Sukuriamas 2-jų dimensijų vektorius;
[f1,f2] = freqspace(21,'meshgrid');
% Sukuriama vienetinė matrica;
Hd = ones(21);
rr = sqrt(f1.^2 + f2.^2);
Hd(rr>0.5) = 0;
colormap(jet(64));
mesh(f1,f2,Hd);
% Sukuriamas 2-jų dimensijų filtras;
win = fspecial('gaussian', 21, 1);
win = win ./ max(win(:));
mesh(win);
% Sukuriamas 2-jų dimensijų juostinis filtras;
h = fwind2(Hd,win);
freqz2(h)
% Paveiksluko pagal rgb filtravimas 2-jų dimensijų FIR filtru;
% Sąsūka;
r=filter2(h,r);
g=filter2(h,g);
b=filter2(h,b);
figure, imshow(r/256)
% Atliekamas normavimas;
q=double(r)+double(g)+double(b);

```

```

r2=double(r)./q;
g2=double(g)./q;
b2=double(b)./q;
figure(13);
imshow(r2);
figure(14);
imshow(g2);
figure(15);
imshow(b2);
% Atliekamas komporavimas (sulyginimas)(G);
g3=(g2<0.274);
figure(16)
imshow(g3*255);
% Atliekamas paveiksluko (G) komponentės erozija ir plėtimas;
% Uždarymas
s1=strel('rectangle',[4,38])
l=imerode(g3,s1);
s2=strel('rectangle',[7,42])
l=imdilate(l,s2);

box=regionprops(double(l), 'BoundingBox');
box2=box.BoundingBox;
if box2(1)-2>0
    box2(1)=box2(1)-2
end
if box2(2)-2>0
    box2(2)=box2(2)-2
end
if box2(3)+2<size(g3,2)
    box2(3)=box2(3)+2
end
if box2(4)+2<size(g3,1)
    box2(4)=box2(4)+2
end
end

```



```

% Lūpu srities iškarpimas;
g5=imcrop(g2,[box2(1), box2(2), box2(3), box2(4)]);
figure(21),imshow(g5);
g5=flipud(g5);
figure(22), imshow(g5)
figure(23);
[C,h] = contour(g5, [0.274 0.274]);
size(g5)
% Kontūro reikšmių radimas (Y);
tx=(0.05:0.15:0.95)*size(g5,2)
clear Y Ya Yv
for j=1:length(tx)
    i=find((abs(C(1,:)-tx(j)))<1)
    diff(i)
    ribos=find(diff(i)>1)
    for k=1:length(ribos)+1
        if k==1
            s3=i(1:ribos(1)),
            [delta, indeks]=min(abs(C(1,s3)-tx(j)))
            Y(j,k)=C(2,s3(1)+indeks-1)
        elseif k==(length(ribos)+1)
            s3=i(ribos(k-1)+1:ribos(k))
            [delta, indeks]=min(abs(C(1,s3)-tx(j)))
            Y(j,k)=C(2,s3(1)+indeks-1)
        else
            s3=i(ribos(k-1)+1:end)
            [delta, indeks]=min(abs(C(1,s3)-tx(j)))
            Y(j,k)=C(2,s3(1)+indeks-1)
        end
    end
end
Ya(j)=min(Y(j, 1:k));
Yv(j)=max(Y(j, 1:k));
end
Ya
Yv

```

```

% Kontūro reikšmių radimas (X);
ty=(0.05:0.15:0.95)*size(g5,1)
clear X Xa Xv
for j=1:length(ty)
    try
        i=find((abs(C(2,:)-ty(j)))<1)
        diff(i)
        ribos=find(diff(i)>1)
        for k=1:length(ribos)+1
            if k==1
                s3=i(1:ribos(1)),
                [delta, indeks]=min(abs(C(2,s3)-ty(j)))
                X(j,k)=C(1,s3(1)+indeks-1)
            elseif k==(length(ribos)+1)
                s3=i(ribos(k-1)+1:ribos(k))
                [delta, indeks]=min(abs(C(2,s3)-ty(j)))
                X(j,k)=C(1,s3(1)+indeks-1)
            else
                s3=i(ribos(k-1)+1:end)
                [delta, indeks]=min(abs(C(2,s3)-ty(j)))
                X(j,k)=C(1,s3(1)+indeks-1)
            end
        end
        end
        Xk(j)=min(X(j, 1:k));
        Xd(j)=max(X(j, 1:k));
    catch
        Xk(j)=NaN;
        Xd(j)=NaN;
    end
end
ty
Xk
Xd
tx
Ya

```

```
Yv
figure
plot(tx, Ya, 'g*')
hold on
plot(tx, Yv, 'c*')
plot(Xk, ty, 'ro')
plot(Xd, ty, 'mo')
% Tinklelis;
for j=1:length(tx)
    line([tx(j) tx(j)], [0 size(g5,1)])
    line([0 size(g5,2)], [ty(j) ty(j)])
end
```