



1 paveikslėlis: Atspėkite asmenis pavaizduotus paveikslėliuose

1 Veidų atpažinimas

- Veido geometrinė normalizacija
- Veido pilkumo lygmių (fotometrinė) normalizacija
- Požymių išskyrimas
- Požymių palyginimas
- Tikėtinumo santykio logaritmas
- Nuorodos
- Praktinis darbas
- Literatūra

1.1 Veido geometrinė normalizacija

Veido segmentacija apytiksliai įvertina veido akių pozicijas. Laikysime, kad vaizde buvo išskirtas tik vienas veidas, kurio akių pozicijos nusakomos skaičiais

$$\text{akys} = (x_{\text{dešinės}}, y_{\text{dešinės}}), (x_{\text{kairės}}, y_{\text{kairės}}) = (x_d, y_d), (x_k, y_k).$$

Koordinatės pateiktos įprastu būdu, t.y. koordinatinių pradžių taškas yra viršutiniame kairiajame paveikslėlio kampe. Šioje koordinatinių sistemoje $x_{\text{dešinės}} < x_{\text{kairės}}$. Atliekant veido dydžio normalizaciją pasirenkamas standartinio dydžio stačiakampis ir jame nurodomos standartinės akių pozicijos. Paprastumo dėlei, kad būtų mažiau laisvų

parametrų, naudosime tik akių koordinates standartinio dydžio vaizde, o stačiakampio dydį apibrėšime naudodami standartinių akių koordinates. Dėl simetrijos, natūralu laikyti, kad stačiakampio plotis yra dvigubai didesnis už akių vidurinio taško abscisę. Stačiakampio aukštį apibrėšime formule

$$H = (\text{akių ordinatė}) + 1.5(\text{atstumas tarp akių}).$$

Gaunamo veido rezoliuciją lemia taškelių tarp akių skaičius. Fiksuokime konkrečias standartines akių koordinates:

$$akys_{\text{standartinės}} = \{(100, 190), (230, 190)\}.$$

Šiuo atveju atstumas tarp akių yra 130 vaizdo taškelių, o normalizuoto standartinio stačiakampio dydis:

$$((100 + 230)/2 * 2, 190 + 3 * (230 - 100)/2 = (330, 385)).$$

Skaičiuojant taškeliais, prisideda po vienetą prie pločio ir ilgio ir gauname tokio dydžio standartinį stačiakampį:

Standartinis veido stačiakampis = 331x386 taškelių vaizdas.

Kaip atvaizduoti pradinį vaizdą į šį stačiakampį? Postuluosime, kad atvaizdis yra standus, t.y. transformacija bet kokį trikampį atvaizduoja į trikampį išlaikydama pradinio trikampio kampus tarp kraštinių ir viršūnių orientaciją. Matematiškai standi transformacija aprašoma lygtimi

$$u = s \cos(f)x - s \sin(f)y + a,$$

$$v = s \sin(f)x + s \cos(f)y + b.$$

Čia (x, y) yra pradinio vaizdo taškelio koordinatės, o (u, v) standartinio dydžio stačiakampio koordinatės. Parametras s apibrėžia atvaizdžio mastelį; sąlyga $s > 0$ garantuoja, kad po atvaizdžio trikampio viršūnių orientacija nesikeičia. Parametras $0 \leq f < 2\pi$ apibrėžia posūkio kampą tarp pradinės x ašies ir pasuktos u ašies; kampo reikšmė matuojama radianais prieš laikrodžio rodyklę. Parametrų a ir b pora nusako poslinkį. Taškas (a, b) yra (x, y) koordinatinių pradžios taško koordinatės naujoje (u, v) koordinatinių sistemoje.

Turint akių koordinates pradinėje ir naujojoje koordinatinių sistemoje, t.y.

$$\{(x_{\text{dešinės}}, y_{\text{dešinės}}), (x_{\text{kairės}}, y_{\text{kairės}})\} = \{(x_d, y_d), (x_k, y_k)\}$$

ir

$$\{(u_{\text{dešinės}}, v_{\text{dešinės}}), (u_{\text{kairės}}, v_{\text{kairės}})\} = \{(u_d, v_d), (u_k, v_k)\} = \{(100, 190), (230, 190)\}$$

lengvai rasime standžios transformacijos parametrus:

$$s \cos(f) = ((x_d - x_k) * (u_d - u_k) + (y_d - y_k) * (v_d - v_k)) / ((x_d - x_k)^2 + (y_d - y_k)^2),$$

$$s \sin(f) = ((x_d - x_k) * (v_d - v_k) - (y_d - y_k) * (u_d - u_k)) / ((x_d - x_k)^2 + (y_d - y_k)^2),$$

$$a = u_k - s \cos(f)x_k + s \sin(f)y_k,$$

$$b = v_k - s \sin(f)x_k - s \cos(f)y_k.$$

Naudojantis šiomis formulėmis nesunku apskaičiuoti mastelio s ir posūkio kampo f reikšmes, tačiau standžios transformacijos formulei apibrėžti pakanka žinoti $s \cos(f)$ ir $s \sin(f)$ reikšmes, kurias



2 paveikslėlis: Originali nesegmentuoto veido nuotrauka, MBGC target original 02463d634.jpg, 2272x1704 taškelių

apskaičiavome nenaudodami nei šaknies traukimo, nei arktangento operacijų; tai spartina skaičiavimus.

Atvirkštinė transformacija, t.y. originalios (x, y) koordinatės, apskaičiuojamos žinant naujas (u, v) koordinates pagal formules

$$\begin{aligned} x &= s^{-1} \cos(f)u + s^{-1} \sin(f)v + a', \\ y &= -s^{-1} \sin(f)u + s^{-1} \cos(f)v + b', \\ a' &= -s^{-1} \cos(f)a - s^{-1} \sin(f)b, \\ b' &= s^{-1} \sin(f)a - s^{-1} \cos(f)b. \end{aligned}$$

2 paveikslėlio akių pozicijos yra $\{(1138, 227), (1291, 230)\}$. Nuotraukoje yra MBGC veido duomenų bazės koordinatoriaus Kevin W. Bo-



3 paveikslėlis: Geometriškai normalizuotas standartinio dydžio segmentuotas veidas, 331x386 taškelių

wyer veidas.

3 paveikslėlyje pavaizduotas geometriškai normalizuotas Kevinio Bowyer veidas, akių pozicijos: $\{(100, 190), (230, 190)\}$.

Žinant atvirkštinės transformacijos formules, normalizuoto dydžio vaizdas $RGB(u, v)$ gaunamas pagal formulę

$$RGB(u, v) = rgb(s^{-1}\cos(f)u + s^{-1}\sin(f)v + a', -s^{-1}\sin(f)u + s^{-1}\cos(f)v + b'),$$

kur $rgb(x, y)$ yra pradinio originalaus vaizdo R, G ir B spalvų komponentės. Skaičiuojant pagal parašytas formules gaunamos nesveikaskaitės koordinatės (x, y) . Todėl atliekant praktinius skaičiavimus $rgb(x, y)$ reikšmė dažniausiai pakeičiama anaudojant sveikaskaičių koordinatę $rgb[i, j]$, $rgb[i + 1, j]$, $rgb[i, j + 1]$ ir $rgb[i + 1, j + 1]$ reikšmes, kur $i = \lfloor x \rfloor$ ir $j = \lfloor y \rfloor$ yra x ir y sveikosios dalys.

2 pav. pateiktas veido originalios nuotraukos vaizdas. Ši nuotrauka paimta iš MBGC target original kolekcijos; nuotraukoje yra MBGC duomenų rinkinio koordinatoriaus Kevin W. Bowyer veidas.

3 pav. pateiktas segmentuotas iki standartinio dydžio 331x386 veidas. Segmentacijai naudotos tokios akių pozicijos: $\{(1138, 227), (1291, 230)\}$.

Programos derinimui galite pasinaudoti konkrečiomis lygtimis, kurios gaunasi pateiktų paveikslėlių duomenims:

$$\begin{aligned} u &= 0.8493466564181399x + 0.016653856008199y - 870.336920317704 \\ v &= -0.016653856008199x + 0.849346656418140y + 16.1503971304125 \\ x &= 1.1769230769230770u - 0.023076923076923v + 1024.69230769231 \\ y &= 0.0230769230769231u + 1.176923076923077v + 1.07692307692308. \end{aligned}$$

Taip pat naudinga patikrinti, kad ištačius į pirmas dvi jūsų gautas lygtis $x = x_k, y = y_k$, turite gauti $u = u_k, v = v_k$. Analogiškai, ištačius į trečią ir ketvirtą lygtį reikšmes $u = u_d, v = v_d$, turite gauti $x = x_d, y = y_d$. Kadangi mūsų atveju gavome mažą kampo $f = -0.02$ vertę, tai reiškia, kad segmentuojamo veido akis jungianti tiesė sudaro mažą kampą su horizonto linija. $s = 0.85$ vertė artima vienetui. Tai rodo, kad normalizuoto iki standartinių akių pozicijų paveikslėlio rezoliucija mažai pasikeitė. Konkrečiau, 1-as originalaus vaizdo taškelis atitinka 0.85 segmentuoto veido taškelį ($s = 0.85$) ir 1-as segmentuoto paveikslėlio taškelis atitinka 1.18 originalaus paveikslėlio taškelį ($s^{-1} = 1.18$). Taigi mūsų atveju segmentacijos mastelis yra 1 : 1.18.

Dar viena naudinga informacija yra nejudamas transformacijos taškas. Pagal nejudamo taško apibrėžimą originalaus vaizdo plokštumos taškas (x, y) (nebūtinai sveikaskaičių reikšmių ir nebūtinai

priklausantis originaliam vaizdai) vadinamas nejudamu, jei to paties taško koordinatų reikšmės gautos segmentuotą vaizdą atitinkančioje plokštumoje, lieka nepakitusios. Kitaip tariant nejudamas taškas tenkina lygtis $x = u$ ir $y = v$. Galimi trys kokybiškai skirtingų segmentacijų atvejai.

1. Segmentuotas ir originalus vaizdas neturi nejudamo taško.

Taip bus tada ir tik tada, kai segmentacijos mastelis yra $1 : 1$, segmentacijos posūkio kampo f reikšmė yra kartotinė 2π ir segmentuotas ir originalus vaizdai **nesutampa**. Kitaip tariant šiuo atveju vaizdai yra paslinkti vienas kito atžvilgiu.

2. Visi segmentuoto ir originalaus vaizdo taškai yra nejudami.

Taip bus tada ir tik tada, kai segmentacijos mastelis yra $1 : 1$, segmentacijos posūkio kampo f reikšmė yra kartotinė 2π ir segmentuotas ir originalus vaizdas **sutampa**. Paprastai kalbant abu vaizdai šiuo atveju sutampa.

3. Segmentuotas ir originalus vaizdas turi tik vieną nejudamą tašką.

Taip bus tada ir tik tada, kai segmentacijos mastelis yra ne $1 : 1$ arba posūkio kampo reikšmė yra nekartotinė 2π (pakanka bent vienos sąlygos).

Kadangi mūsų segmentacijos pavyzdžio $s \neq 1$ (ir $f \neq 0$), segmentaciją atitinkanti transformacija turės tik vieną nejudamą tašką. Nejudamo taško koordinatės yra

$$x = u = -5695.631970260222,$$

$$y = v = 736.82156133829.$$

Nejudamo taško koordinates taip pat galite panaudoti programos derinimui. Paprasčiausiai įstatykite šias koordinates į (x, y) ir (u, v) sąryšių lygtis ir turite gauti nepakitusias koordinates.

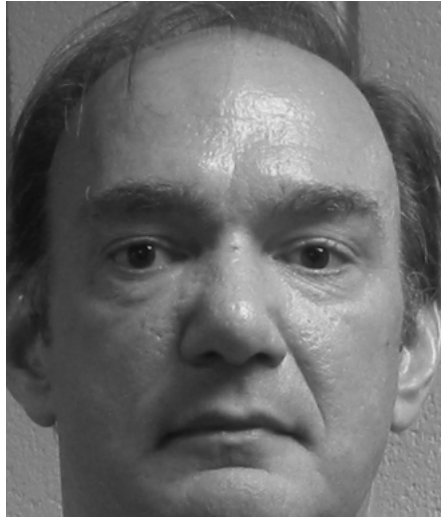
1.2 Veido pilkumo lygmenų normalizacija

Literatūroje yra daug informacijos apie veido spalvos panaudojimą jo segmentacijai. Tačiau lyginant segmentuotus veidus dažniausiai atsi sakoma spalvinės informacijos ir lyginamos veidų tekstūros pateiktos pilkumo lygmenimis. Pastebėta, kad baltaodžių ir geltonodžių veiduose dominuoja raudonoji ir žalioji komponentės, o mėlynojoje santykinai būna daug triukšmo. Todėl perėjimo prie pilkumo lygmenų formulėje

$$grey = ar + bg + cb$$

rekomenduojame paimti svorius $a = 0.5$, $b = 0.5$ ir $c = 0$. Skrupulingai optimizuojant veidų atpažinimo kokybę galima parinkti ir kitus svorius, tačiau mes fiksuosime šį paprastą variantą.

4 pav. iliustruoja segmentuotą MBGC 02463d634 veidą pateiktą pilkumo lygmenimis. Praktiškai pastebėta, kad tiesiogiai skaičiuojant veidų požymius naudojantis veido pilkumo lygmenų vaizdu gaunasi nestabilūs požymiai. Pagrindinis požymių nestabilumo šaltinis yra veido apšvietimas. Kintant apšvietimo šaltinio spektrui ir pozicijai gaunasi skirtingų pilkumo lygmenų vaizdai, o tai kenkia atpažinimo kokybei. Kad kiek eliminuoti apšvietimą, atliekama taip vadinama veido *fotometrinė* arba *pilkumo lygmenų* normalizacija. Normalizacijos pagrin-



4 paveikslėlis: Segmentuoto veido pilkumo lygmenys, naudota formulė $grey = (r + g)/2$

dinė idėja išnaudoti apšvietimo stiprio santykinai lėtą kitimą gretimuose taškuose. Praktiškai normalizacija atliekama dažnai tokiu būdu:

1. Atliekamas pilkumo lygmenų vaizdo vidurkinimas σ_1 masteliu;
2. Atliekamas pilkumo lygmenų vaizdo vidurkinimas $\sigma_2 > \sigma_1$ masteliu;
3. Normalizuotas vaizdas apibrėžiamas dviejų vidurkintų vaizdų santykiu arba skirtumu.

Kad realizuoti šią procedūrą, reikia pasirinkti vaizdo vidurkinimo būdą. Teoriškai tam tikra prasme optimalus vidurkinimas naudoja Gauso filtrą. Gauso funkcijos vidutinio kvadratinio nuokrypio dydis σ apibrėžia vidurkinimo mastelį. Gauso vidurkinimas yra invariantiškas vaizdo posūkiui, tačiau vidurkinimo procedūra yra lėta. Vidurkinimas

sparčiai atliekamas vidurkinant pasirinkto dydžio kvadrato formos slenkamuoju vidurkinimo langu, t.y. vidurkinimo filtras yra lygus 1, kai indeksai patenka į 2σ dydžio kvadratą, kurio vidurys yra taške $(0,0)$, kitais atvejais filtro reikšmė yra nulis. Tačiau vidurkinimo rezultatas panaudojant šį filtrą gana jautrus vaizdo posūkiui ir poslinkiui, nes filtras yra trūkus, t.y. jo reikšmės šuoliu pereina nuo 0 iki 1 ir atvirkščiai.

Todėl mes pasirinksimė tarpinį eksponentinį filtrą, kuris yra tolydus ir su kuriuo sparčiai atliekami skaičiavimai. Šis filtravimas detalai aprašytas kyrelyje. Kad atlikti veido fotometrinę normalizaciją, pasirinksimė du skirtingus eksponentinių filtrų mastelius, suvidurkinsimė veidą ir imant gautų vaizdų santykį arba skirtumą gauti normalizuotą vaizdą. Vidurkinimui naudosimė du tolydžiai diferencijuojamus dvimačius eksponentinius filtrus

$$\begin{aligned} f_{\sigma_1}(x) &= c_1 \left(1 + \frac{|x|}{\sigma_1}\right) e^{-\frac{|x|}{\sigma_1}} \\ f_{\sigma_2}(x) &= c_2 \left(1 + \frac{|x|}{\sigma_2}\right) e^{-\frac{|x|}{\sigma_2}}, \quad x \in (-\infty, \infty). \end{aligned}$$

Čia c_1, c_2 normuojančios konstantos, kad filtrai konstantinį signalą atvaizduotų į tą pačią konstantą, o filtrų mastelių parametrus fiksuosime taip: $\sigma_1 = 1$ ir $\sigma_2 = 2$.

5 pav. iliustruoja gautus vidurkinimo rezultatus. Normalizuotą vaizdą apibrėšime išraiška

$$\begin{aligned} u^*(x, y) &= 128 + c\delta(x, y), \\ \delta(x, y) &= (u_1(x, y) - u_2(x, y))/((u_1(x, y) + u_2(x, y))/2). \end{aligned}$$



5 paveikslėlis: Vidurkintas veidas su glodžiais eksponentiniais filtrais: $\sigma_1 = 1$ a) atveju ir $\sigma_2 = 2$ b) atveju.

Čia u_1 ir u_2 vidurkinti 5 pav. a) ir b) veidai. Adityvi konstanta 128 paslenka vaizdą iki vidutinio pilkumo lygmens, o daugiklio c reikšmė parenkama taip, kad gauto normalizuoto veido pilkumo lygmenys kistų nuo 0 iki 255. Jei vaizdas taško (x, y) aplinkoje mažai keičiasi, tai $\delta(x, y)$ reikšmė bus maža ir normalizuoto vaizdo reikšmė bus arti 128. Kad išryškinti mažas reikšmes, normalizacijai galima naudoti papildomą dėmenį $\text{sgn}(\delta(x, y))\sqrt{|\delta(x, y)|} = \delta(x, y)/\sqrt{|\delta(x, y)|}$. Pateiktoje iliustracijoje 6 pav. b) panaudota tokia išraiška:

$$u^*(x, y) = 128 + c(\delta(x, y) + \delta(x, y)/\sqrt{|\delta(x, y)|}).$$

6 pav. b) fotometrinė normalizacija labiau išryškina smulkius tekstūros pokyčius. Abi normalizacijos ženkliai sumažina dėl apšvietimo kilusius veido pilkumo lygmenų pokyčius. Nesunku pastebėti kaip sumažėjo šešėlių įtaka vaizdui, tačiau ji pilnai nėra eliminuota ir vargu ar galima



6 paveikslėlis: Normalizuotų pilkumo lygmenų veidai. a) Normalizacija atlikta panaudojant $u^*(x, y) = 128 + c\delta(x, y)$ formulę. b) Normalizacija atlikta panaudojant $u^*(x, y) = 128 + c(\delta(x, y) + \frac{\delta(x, y)}{\sqrt{|\delta(x, y)|}})$ formulę.

tikėtis idealaus veido fotometrines normalizacijos algoritmo. Kuris normalizacijos variantas pranašesnis vizualiai sunku spręsti, tai išryškėja vėlesniuose etapuose kai atpažįstami veidai ir įvertinama veidų atpažinimo kokybė.

1.3 Požymių išskyrimas

1.3.1 Dažniausiai naudojami požymiai veidams atpažinti

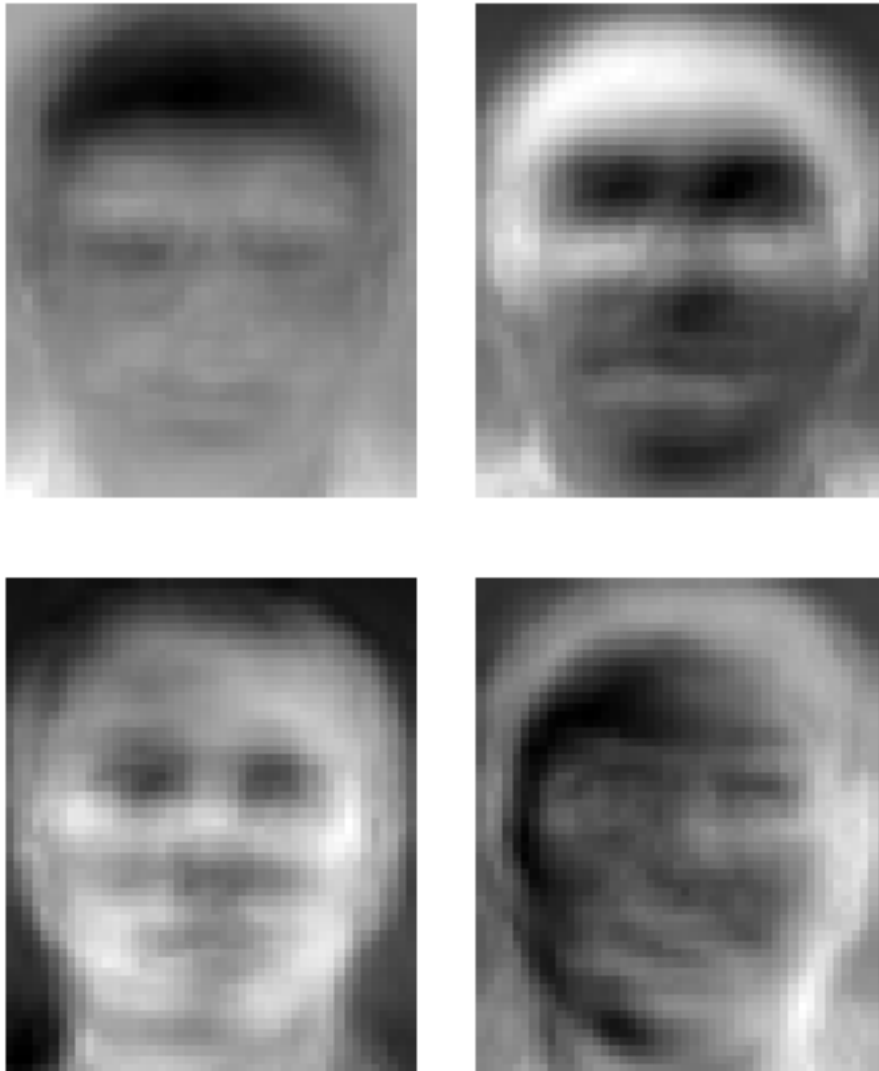
Veidų atpažinimo algoritmus galima suskirstyti į dvi stambias kategorijas: globalių (holistic) ir lokalių (local) požymių lyginimo algoritmus. 1987 metais pasirodė *tikrinio veido* (angl. *eigenface*) [12] metodas, kuris ilgą laiką buvo labai populiarus. Tikrinių veidų metodas yra pagrindinių komponentų metodo (angl. *Principal Component Ana-*

lysis, *PCA* dalinis atvejis. Šio metodo viena pirmųjų realizacijų [13] atskleidė metodikos trūkumus ir dėl prastų veido atpažinimo rezultatų yra naudojama dažniausiai kaip apatinis atramos taškas kitų algoritimų rezultatams palyginti. Klasikinis tikrinių veidų algoritmas naudoja globalius požymius. Šio metodo modifikacijos dažnai atskirų veido komponentių tikrines reikšmes ir bazines funkcijas ir tuo pačiu tolstama nuo holistinės veido reprezentacijos. Kadangi modifikacijos duoda geresnius veidų atpažinimo rezultatus, tai byloja apie tikrinių veidų ir kitų holistinių metodų ydas.

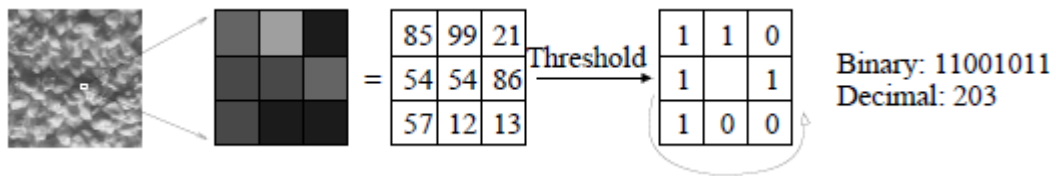
7 pav. iliustruoja kelis pirmuosius tikrinius veidus. Tikriniai veidai surandami vidurkinant turimą geometriškai normalizuotų veidų paveikslėlių kolekciją ir apskaičiavus standartinio dydžio veidų kovariacinę matricą. Toliau yra randami kovariacinės matricos tikriniai tikrines reikšmės ir tikriniai vektoriai kurie ir yra vadinami tikriniais veidais. Tikriniai veidai surūšiuojame tikrinių reikšmių mažėjimo tvarka, todėl tikriniai veidai su mažiausiais indeksais atspindi charakteringiausius (dažniausiai pasitaikančius) veidų nuokrypius nuo vidurkinio veido. 7 pav. iliustruoja pateikti keturi pirmieji tikriniai veidai, kurie buvo gauti remiantis AT&T laboratorijoje turima veidų duomenų baze. <http://blog.zabarauskas.com/eigenfaces-tutorial/> ([14]) adresu rasite Manfredo Zabarausko tikrinių veidų metodikos aprašymą ir demonstracinę programą.

1.3.2 Lokalūs binariniai vaizdai

LBP (angl. *Local Binary Patterns*) metodas buvo sukurtas tekstūroms aprašyti ir palyginti neseniai (2004 metais) LBP metodas buvo



7 paveikslėlis: AT&T laboratorijoje gautų tikrinių veidų pavyzdžiai.



8 paveikslėlis: LBP reikšmės apskaičiavimas (naudota [?] darbo iliustracija).

pritaikytas atpažinti veidams [?]. Dėl savo paprastumo ir santykinai neblogų rezultatų LBM metodas labai išpopuliarėjo. LBP reikšmė priklauso nuo pasirinkto centrinio taškelio ir jo aplinkos. 8 pav. iliustruoja LBP reikšmės apskaičiavimą viename taške. Kaip matyti iš iliustracijos, LBM reikšmė priklauso tik nuo santykinio pilkumo lygmenų didumo. Tai reiškia, kad atlikus bet kokią monotonišią vaizdo pilkumo lygmenų transformaciją gausime transformuotam vaizdui tas pačias LBP reikšmes. Ši savybė rodo LBP požymių atsparumą ir, kartu su paprastumu, yra pagrindinis šios metodikos privalumas.



9 paveikslėlis: Veidų suskaidymas į lokalias sritis, kuriose skaičiuojamos LBP reikšmių histogramos

Taikant LBP požymius veidas, pastarasis padalinamas į mažas lokalias sritis kurioms apskaičiuojamos LBP reikšmių histogramos. Skirtingų veidų paveikslėlių gautos LBM reikšmių histogramos yra lyginamos ir jų panašumas atspindi lyginamų veidų panašumą. Suskaidymas veido į lokalias sritis dar kartą atspindi tą faktą, kad globalūs požymiai nelyginamai kokybiškai aprašo ir todėl pereinama prie lokalesnių. Tačiau naudoti palyginimui pavie-

niais LBP reikšmes būtų keblu skaičiavimo laiko prasme ir iškiltų problemos nustatyti tiksliai poslinkio reikšmes, kurios garantuotų lyginamų autentiškų veidų panašumą. Lokalių sričių LBP histogramos yra kompromisinis variantas, kurias naudojant veidų palyginimas atliekamas greitai ir toliaruojami nedideli lyginamų autentiškų veidų poslinkiai. Matematinės statistikos terminais histogramos atspindi LBP reikšmių skirstinį, todėl lyginant histogramas galima naudotis matematinės statistikos metodais. Palyginimo detales galite pasižiūrėti [15] darbe.

LBP histogramos jautriai reaguoja į įvairius apšvietimo pasikeitimus, kartu ir į triukšmą. Charakteringos veido paveikslėlio zonos, kuriose yra santykinai daug triukšmo, yra kakta ir skruostai. Todėl kartais lyginant naudojami skirtingi svoriai skirtingų LBP histogramų sritims.

Toliau aprašysime požymių grupę, kuri atspari triukšmams, duoda gerus veidų palyginimo rezultatus ir yra dažnai naudojama geriausiuose veidų atpažinimo algoritmuose.

1.4 Gaboro lokalūs požymiai ir Gaboro tūtos

Gaboro požymiai gaunami filtruojant veidą įvairių parametru Gaboro filtrais. Gaboro filtras Daugmanas (<http://www.cl.cam.ac.uk/~jgd1000/>) sėkmingai pritaikė išskiriant akies rainelės binarinius požymius. Gabor filtras taiko praktiškai visos veidų atpažinimo sistemos. Apie veidų atpažinimo technikas, grįstas Gaboro požymiais, galite paskaityti ??, ??, ?? darbuose.

Gaboro filtrai yra Gauso (angl. *Gaussian*) ir kosinuso bei sinuso funkcijų sandaugos. Kadangi filtrai taikomi vaizdams, tai visos funkci-

jos yra dviejų kintamųjų x ir y . Gauso funkcijos reikšmės priklauso tik nuo taško (x, y) atstumo iki koordinatų pradžios ir turi vieną laisvą parametą σ , kuris apibrėžia funkcijos mastelį. Sinusas ir kosinus dažniausiai apjungiami į vieną kompleksinę eksponentę ir turi du laisvus parametrus: krypties vektorių ir dažnį. Matematiškai Gaboro filtras apibrėžiamas tokia išraiška:

$$G(x, y) = s(x, y) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (1)$$

kur $s(x, y)$ yra kompleksinė eksponentė, kurios realioji ir menamoji komponentės skaičiuojamos pagal formules

$$\begin{aligned} \operatorname{re} s(x, y) &= \cos(2\pi f(x \cos \alpha + y \sin \alpha)), \\ \operatorname{im} s(x, y) &= \sin(2\pi f(x \cos \alpha + y \sin \alpha)), \end{aligned}$$

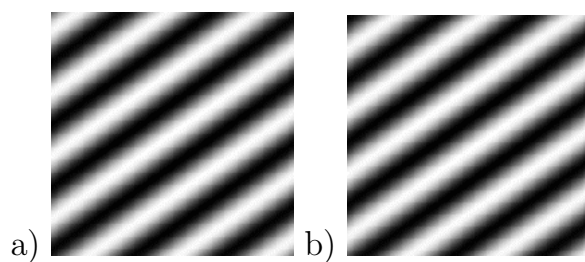
α filtro krypties, o f - dažnio parametrai. 10 paveikslėlis iliustruoja $\operatorname{re} s(x, y)$ ir $\operatorname{im} s(x, y)$ Gaboro filtro komponentes.

Jei $u(x, y)$ yra paveikslėlio pilkumo lygmenų intensyvumo funkcija, tai vaizdo Gaboro atsaku taške (x, y) vadinama sąsūka

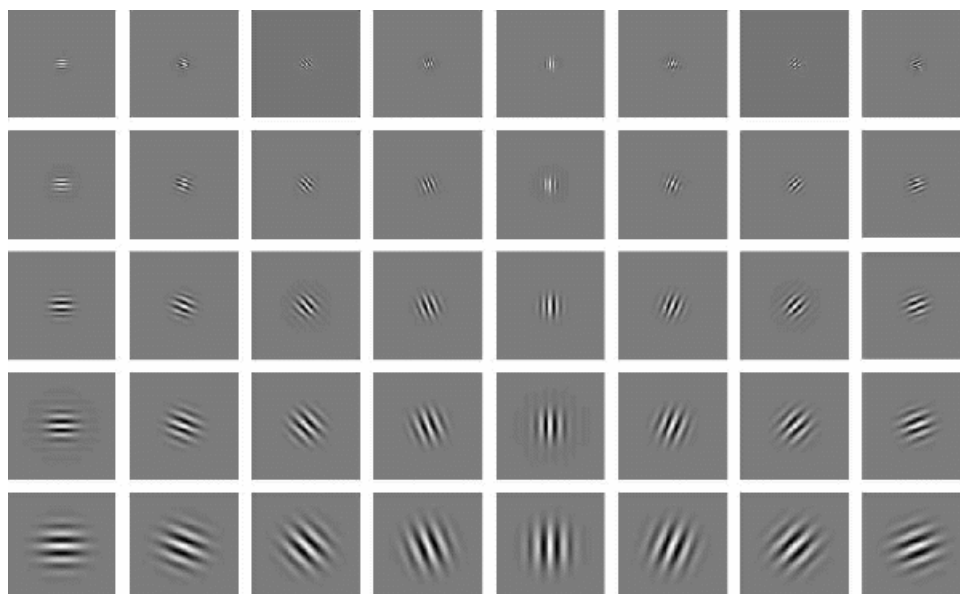
$$(u * G)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x - \xi, y - \eta) G(\xi, \eta) d\xi d\eta.$$

Daugumoje veidų atpažinimo sistemų, naudojančių Gabor filtras, naudojami aštuonių orientacijų α ir penkių mastelių σ Gabor atsakai. ?? paveikslėlis iliustruoja šių kryptių ir mastelių filtras.

Geometriškai ir fotometriškai normalizuoto veido srityje atliekama sąsūkos operacijos su Gaboro filtrų kompleksu ir kiekviename taške gaunamas 8×5 požymių rinkinys, kuris vadinamas Gaboro *tūta* (angl. *Gabor jet*). Gaboro tūta charakterizuoja veido sritį, esančią apie centrinį



10 paveikslėlis: Gaboro filtro kompleksinės eksponentės fiksuotos krypties ir dažnio realioji (a pav.) ir menamoji (b pav.) komponentė.



11 paveikslėlis: Aštuonių krypčių ir penkių mastelių Gaboro filtrų kompleksas

(tašką, kuriame eksponentinė Gauso komponentė maksimali) Gaboro filtro tašką. Veidų panašumas apskaičiuojamas lyginant kiekvieno veido gautas tūtas. Yra patikrinta praktiškai, kad Gaboro požymiai yra atsparūs apšvietimo kitimui ir mažam veido nuokrypiui atsiradusiam po geometrinės normalizacijos. Jei veidui netaikoma fotometrinė normalizacija ir lyginami veidai fotografuoti skirtingo apšvietimo

sąlygomis, tai Gaboro tūta ti dalinai atspari apšvietimo trugdžiams ir naudojant tokius požymius kiek pablogėja veidų atpažinimo rezultatai.

Pagrindinis Gaboro požymių trūkumas - jie lėtai apskaičiuojami. Todėl dažnai Gaboro požymiai įvertinami tik išretintame centrų tinklelyje, o tai menkina jų gebėjimą atskirti kiek paslinktus autentiškus veidus. Anskčiau aprašytus vaizdų vidurkinimo simetrinius eksponentinius filtrus galima apibendrinti įvedant dažnio parametą ir su gautais filtrais galima atlikti greitus skaičiavimus, bet detalai šios procedūros čia neaprašinėsime. Kad išspręsti Gaboro požymių lėto apskaičiavimo problemą, mes panaudosime greitą eksponentinį vidurkinimą ir taip Teiloro požymius, kurie taip pat greitai apskaičiuojami.

1.5 Teiloro binariniai požymiai

Apskaičiuodami veido požymius remsimės normalizuotais veido vaizdais. Lyginant vidurkintas tekstūras stengsimės įvertinti pilkumo lygmenų kitimo tendenciją pasirinkto taško aplinkoje. Vaizdo pilkumo lygmenis galime įsivaizduoti dviejų kintamųjų funkcija $u = u(x, y)$. Iš matematinės analizės gerai žinoma, kad glodžią funkciją galima gerai aproksimuoti taško aplinkoje žinant funkcijos išvestines tame taške. Nors vaizdai yra suvidurkinti, tačiau skaičiuojant aukštesnės eilės išvestines gaunamos nestabilios stipriai osciliuojančios reikšmės. Todėl naudosime tik pirmos ir antros eilės išvestines:

$$\begin{aligned} u_x(x, y) &= \frac{du}{dx}, & u_{xx}(x, y) &= \frac{d^2u}{dx^2}, \\ u_y(x, y) &= \frac{du}{dy}, & u_{yy}(x, y) &= \frac{d^2u}{dy^2}. \end{aligned}$$

Psichofiziologai išsiaiškino, kad žmogaus regos sistema remiasi skirtingomis kryptimis ir skirtingais masteliais filtruotais vaizdais. Imituodami šias regos ypatybes, pasirinksiame keturias kryptis

$$\alpha = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4},$$

ir keliais skirtingais masteliais σ vidurkinsime veido vaizdus. Fiksuotai kryptčiai gauname tokias išvestines:

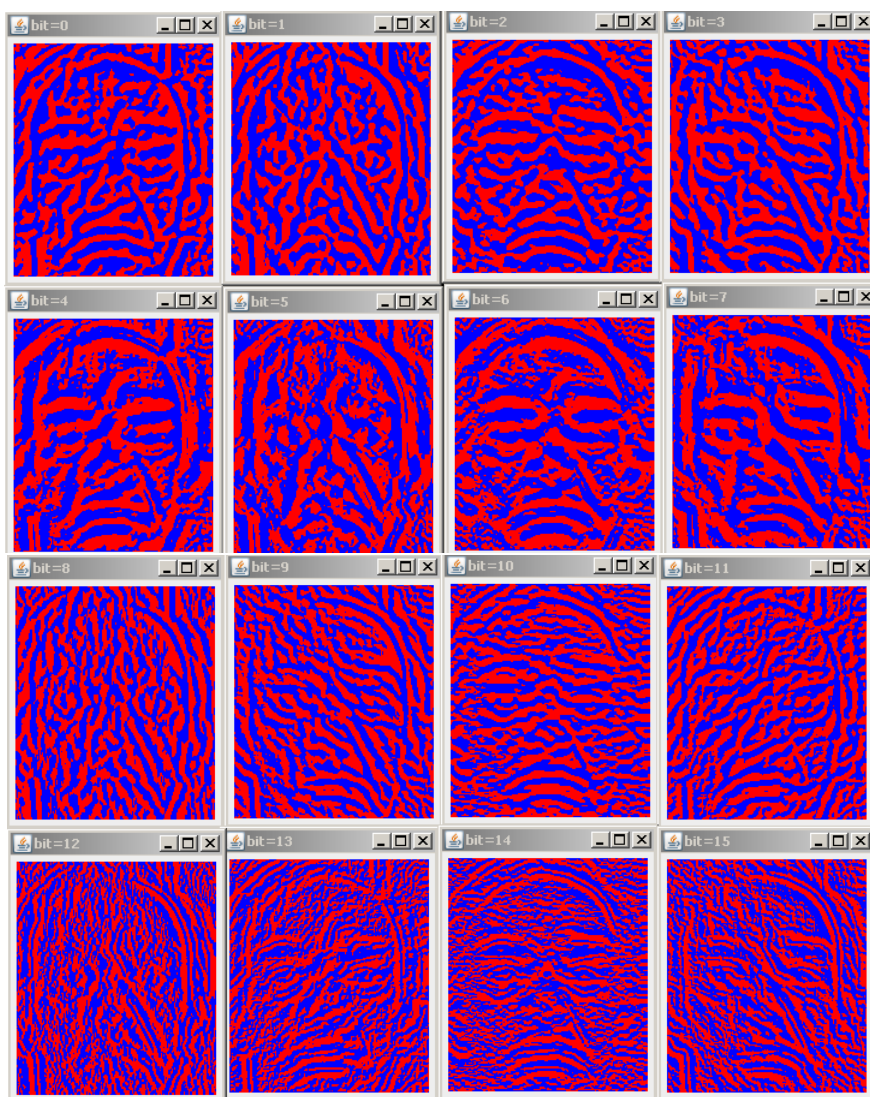
$$u'_\alpha(x, y) = u_x(x, y) \cos \alpha + u_y(x, y) \sin \alpha, \quad (2)$$

$$u''_\alpha(x, y) = u_{xx}(x, y) \cos^2 \alpha + u_{xy}(x, y) \sin 2\alpha + u_{yy}(x, y) \sin^2 \alpha. \quad (3)$$

Šiose formulėse išvestinės u_x, u_y, u_{xx}, u_{yy} yra randamos kaip aprašyta eksponentiniame vaizdų filtravimo skyrelyje. Kad taupyti požymiams saugoti reikalingus resursus ir vėliau būtų galima sparčiai atlikti dviejų veidų požymių palyginimą, saugosime tik išvestinių ženklus. Toks požymių pasirinkimas maksimaliai stabilizuoja požymio reikšmę. Tarkime, vaizdo išvestinės ženklo pasikeitimas dėl triukšmo reiškia, kad vaizdas taško aplinkoje pakeitė savo didėjimo arba mažėjimo kryptį. Kadangi mūsų vaizdai yra vidurkinami, tai toks pasikeitimas gali įvykti tik esant pakankamai dideliame triukšmo lygiui. Dar vienas tokio pasirinkimo privalumas - nenaudojami jokie adaptuoti prie duomenų slenksčiai, nes ženklas surandamas pagal vieningą "slenkstį" lygų 0.

12 pav. iliustruoja tokiu būdu gautus požymius. Raudona spalva žymi neneigiamas išvestines, o mėlyna - neigiamas. Pirmosios eilutės iš kairės į dešinę einantys binarinių požymių vaizdai atitinka tokių išvestinių ženklus:

$$u'_0(x, y), u'_{\pi/4}(x, y), u'_{\pi/2}(x, y), u'_{3\pi/4}(x, y).$$



12 paveikslėlis: Veido binariniai požymiai

Antrosios eilutės binarinių požymių vaizdai atitinka ženklus tų pačių išvestinių, skaičiuotų dviejų gretimų mastelių vidurkintų vaizdų skirtumams.

Trečiosios eilutės iš kairės į dešinę einantys binarinių požymių vaizdai atitinka tokių antrosios eilės kryptinių išvestinių ženklus:

$$u''_{0,0}(x, y), u''_{\pi/4, \pi/4}(x, y), u''_{\pi/2, \pi/2}(x, y), u''_{3\pi/4, 3\pi/4}(x, y).$$

Ketvirtosios eilutės binarinių požymių vaizdai atitinka ženklus tų pačių antrosios eilės kryptinių išvestinių, skaičiuotų dviejų gretimų mastelių vidurkintų vaizdų skirtumams.

Vaizdų vidurkinimo parametro σ konkrečių reikšmių nenurodome; jas pasirinkite laisvai vizualiai kontroliuodami gaunamų binarinių požymių vaizdus. Gaunami binariniai vaizdai turi būti ne per daug chaotiški, nes tuomet lyginamos autentiškos veidų poros vaizdai turės menką panašumą. Taip pat binariniuose vaizduose neturi būti stambių vieno ženklo sričių, nes tuomet lyginama apsišaukėlių veidų pora atsitiktai stambių sričių dėka gali turėti didelę dalį sutampančių binarinių požymių. Iš pateiktų vaizdų matyti, kad antrosios eilės kryptinių išvestinių ženklai labiau varijuoja, nei pirmosios eilės kryptinių išvestinių. Didinant kryptinės išvestinės eilę vis dažniau skiriasi gretimų išvestinių ženklai, todėl lyginant tokius binarinius vaizdus vis sunkiau gauti reikšmingą sutapimą lyginamų autentintiškų veidų porų. Pateiktus požymius vadinsime *Teiloro binariniais* požymiais, nes jie gauti skleidžiant vidurkintą vaizdą lokaliai Teiloro eilute ir binariniai požymiai koduoja skleidinio pirmųjų koeficientų ženklus. Lyginant su Gaboro požymiais, pastarųjų naudojama žymiai mažiau. Taip yra

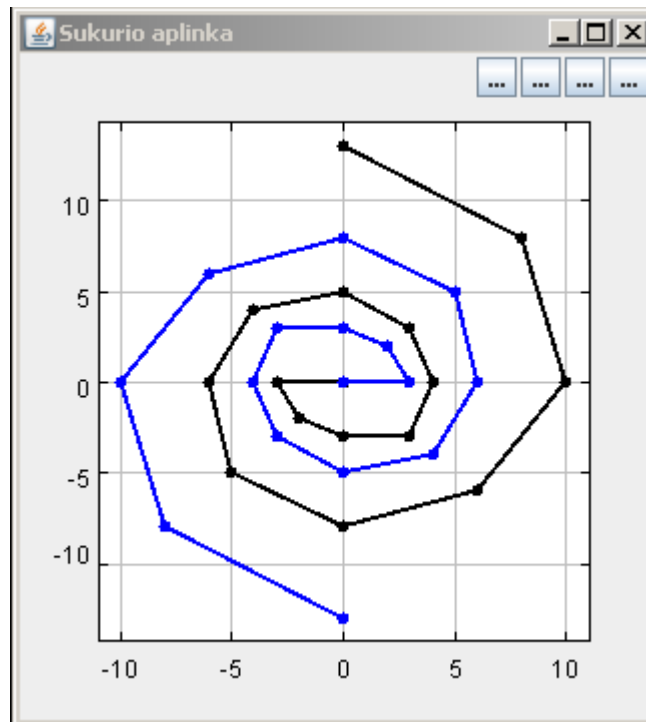
dėl to, kad Gaboro požymiai santykinai lėtai apskaičiuojami ir todėl jie dažniausiai įvertinami išretintame taškelių tinklelyje. Turint pilnas binarinių požymių matricas galima aptikti smulkius lyginamų vaizdų poslinkius, tačiau kartu tai ilgina veidų palyginimo kaštus.

1.6 Binarinių požymių grupavimas

Lyginant dviejų šablonų binarinius požymius reikia apskaičiuoti sutampančių bitų kiekį. Binarinė *xor* operacija lygina dviejų skaičių bitus: sutampančiose bitų pozicijose įrašomas 0, o nesutampančiose 1. Dabartinės architektūros kompiuteriai *xor* operacijas gali atlikti su skaičiais užimančiais nuo vieno iki aštuonių baitų. Kad pagreitin-ti skaičiavimus, naudojamos iš anksto apdorotos lentelės (angl. *look-up tables*), kuriose saugojamas skaičiaus, atitinkančio lentelės indekso reikšmę, nulinių bitų kiekis. Lentelės dydį apriboja atminties kaštai. Praktiškai įsitikinta, kad reikamos lentelėms atminties ir informacijos apdorojimų spartos optimumas pasiekiamas naudojant 16-os bitų len-teles. Tokią lentelę vadinsime *ZeroBits#*. Užrašysime keletą jos konk-rečių reikšmių:

$$\begin{aligned}
ZeroBits\#[0] &= 16, \\
ZeroBits\#[1] &= 15, \\
ZeroBits\#[2] &= 15, \\
ZeroBits\#[3] &= 14, \\
ZeroBits\#[4] &= 15, \\
&\dots \\
ZeroBits\#[0xffe] &= 1, \\
ZeroBits\#[0xfff] &= 0.
\end{aligned}$$

Kad panaudoti tokią nulinių bitų skaitliukų lentelę, paskaičiuotus binarinius požymius reikia apjungtimi grupėmis. Pirma mintis būtų paprasčiausiai fiksuoti šablono taško (x, y) koordinates ir apjungti į *short* tipo (javos terminais) skaičių šešiolika skirtingomis kryptimis ir masteliais apskaičiuotų išvestinių binarinius (ženklų) požymius. Tačiau toks pasirinkimas grupuotų gana stipriai koreliuojančius požymius. Jei mastelių arba krypčių reikšmės yra artimos, tai tikėtina, kad išvestinių ženklai sutaps; tai ir apsprendžia koreliavimą. Lyginant koreliuojančius binarinių požymių rinkinius didėja tikimybė atsitiktinai gauti didelias panašumo reikšmes, o tai didina tikimybę apsisaukėliams prisirinkti daug panašumo taškų, kas blogina atpažinimo kokybę. Pavyzdžiui, kraštutinės koreliacijos atveju, galime gauti, kad fiksuotam taškui (x, y) abiejų šablonų visi 16-a bitų yra vienodi (tarkime visos kryptinės išvestinės teigiamos) ir tokiam taške gautume maksimalų bitų panašumą. Kad išvengti tokių atvejų, siūlome viename *short* tipo



13 paveikslėlis: Binarinių požymių sūkurio rinkinys.

skaičiuje kaupti ne tos pačios pozicijos (x, y) binarinius požymius, o aplink jį esančių taškelį binarinius požymius. fig:sukury pav. iliustruoja pasirenkamus taško aplinkos taškus. Centrinio taškelio (x, y) padėtis yra dviejų sūkurių pradžios taškas. Ir mėlynai ir juodai pažymėtuose sūkuriuose yra pasirenkama po 16-a taškelį. Mėlyno ir juodo sūkurių pradžios taškas įtraukiamas į abudu 16-os bitų rinkinius. Kad sūkurių pradžios taško binarinio požymio reikšmė nesikartotų abiejuose rinkiniuose, imami dviejų skirtingų mastelių ar kryptių išvestinių ženklai. Einant vieno sūkurio gija taip pat keičiamos mastelio σ ir išvestinės krypties α reikšmės. Konkrečių mastelių ir kryptių taisyklių nenurodome; jas galite laisvai pasirinkti patys. Renkantis parametrus stenkitės, kad taško aplinka būtų ne per daug išplitusi ir 16-os bitų rinkinio bitai

būtų kuo mažiau koreliuojantys tarpusavyje.

1.7 Požymių palyginimas

Turint du požymių šablonus juos palyginti labai paprasta. Paprasčiausiai turite pasirinkti lyginimo centrų (x, y) tinklėlį ir jo taškuose įvertinti panašumą. Panašumo metriką apibrėšime formule

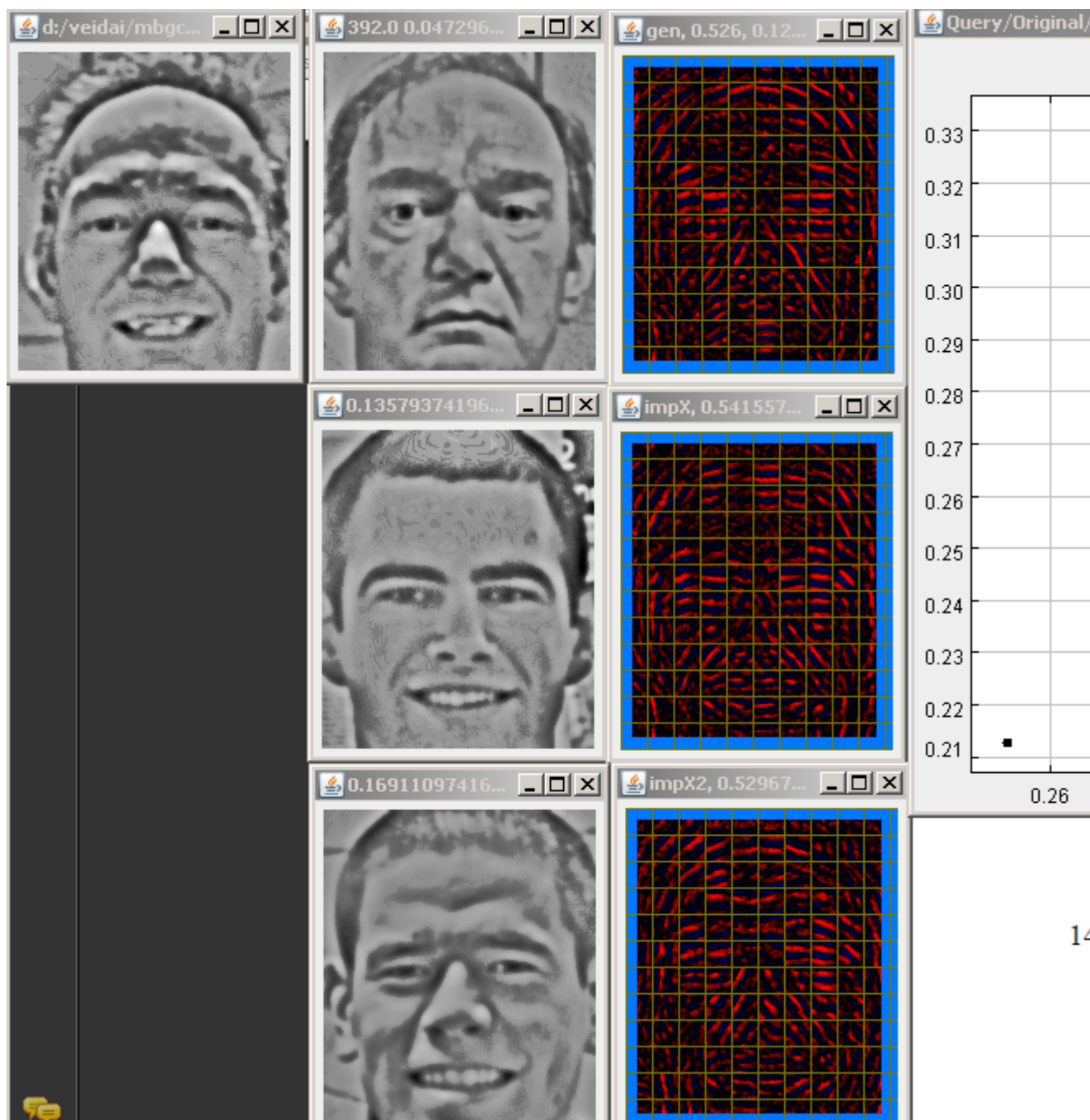
$$\rho(Target, Query) = \sum_{(x,y)} ZeroBits\#(Target_{BitųRinkinys(x,y)} XOR Query_{BitųRinkinys(x,y)}) \quad (4)$$

Kad geriau būtų interpretuoti gautą panašumo reikšmę, ją patartina normuoti. Normuojant reikia adalinti gautą panašumo reikšmę iš teoriškai galiomos maksimalios panašumo reikšmės.

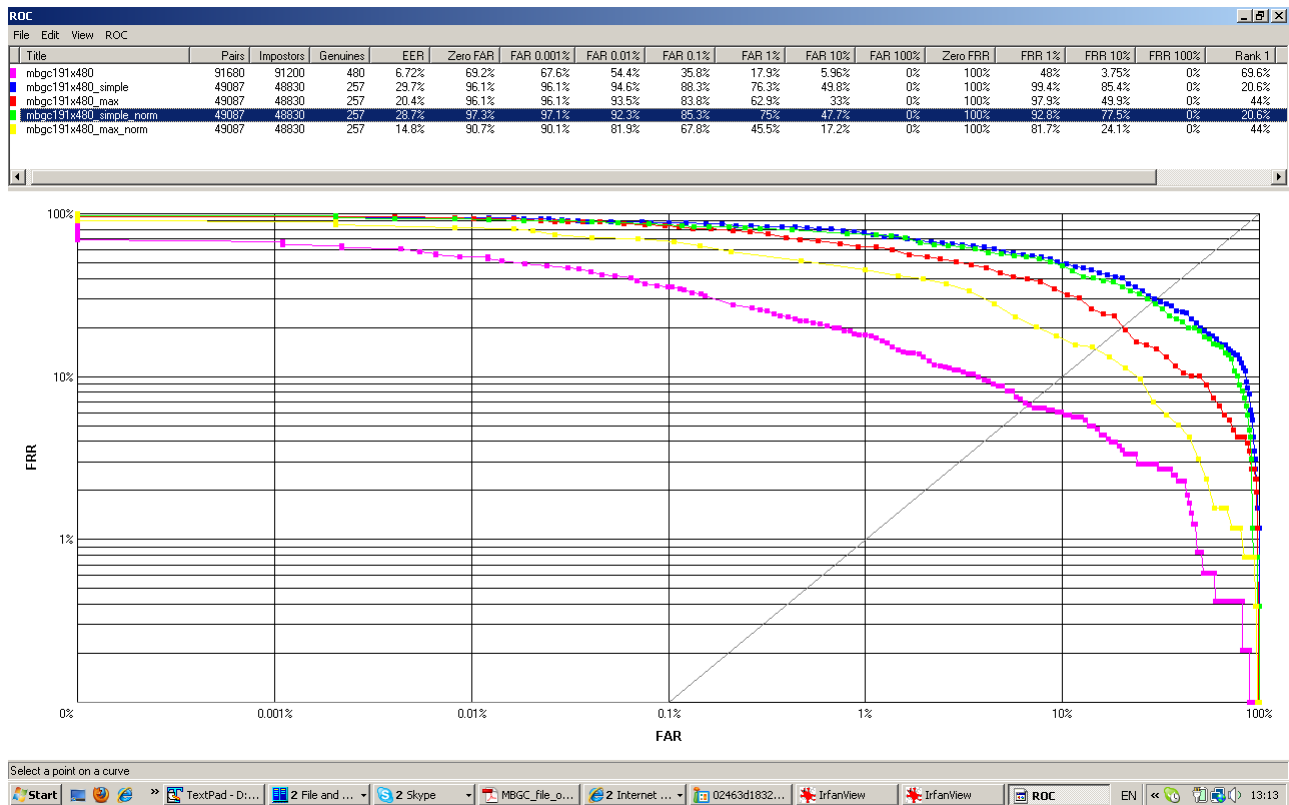
1.8 Tikėtinumo santykio logaritmas

Lyginamų porų panašumo galimų reikšmių sritis priklauso nuo lyginimo metrikos. Tai apsunkina panašumo reikšmių interpretaciją. Tarkime jei vienos poros panašumo reikšmė yra $\rho = 0.8$, o kitos - $\rho = 0.65$, galime teigti, kad santykinai pirmosios poros veidai panašūs nei antrosios poros veidai, tačiau į kiekybinį klausimą "kiek kartų labiau panašūs" atsakyti negalima. Todėl biometrikoje vis dažniau taikomas *tikėtinumo santykio* (angl. *Likelihood Ratio (LR)*) metrika, kurios reikšmės galima interpretuoti kiekybiškai. Lyginamos biometrijos poros (X, Y) panašumo tikėtinumo santykis apibrėžiamas formule

$$LR(X, Y) = \frac{P\{\text{Tikėtinumas, kad } (X, Y) \text{ biometrikos sutampa}\}}{P\{\text{Tikėtinums, kad } (X, Y) \text{ biometrikos nesutampa}\}}. \quad (5)$$



14 paveikslėlis: Panašumo metrikos derinimo aplinka.



15 paveikslėlis: MBGC 191 Target ir 480 Query rinkinio įvairios DET kreivės.

Bendraja prasme tikėtinumas yra modelio tikimybė, kai žinomi atlikto eksperimento rezultatai. Mūsų atveju "eksperimento rezultatai" yra veidai X ir Y ir jų požymiai. Skaitiklio tikėtinumas yra kokio nors modelio tikimybė gauti ("išmatuoti") X ir Y veidų požymius, darant prielaidą, kad veidai yra to paties asmens, o vardiklio tikėtinumas yra tikimybė gauti tuos pačius X ir Y požymius, darant prielaidą, kad veidai yra skirtingų asmenų. Pagal apibrėžimą LR reikšmės gali kisti nuo 0 iki ∞ . Jei $0 \leq LR = LR(X, Y) < 1$, labiau tikėtina, kad lyginami veidai X ir Y yra skirtingi. Priešingai, jei $1 < LR < \infty$, labiau

1 lentelė: Tikėtinumo santykio LR ir jo natūraliojo logaritmo LLR reikšmės

LR	LLR	Interpretacija
1000	6.9	Labai tikėtina, kad lyginami veidai sutampa
403.4	6	Labai tikėtina, kad lyginami veidai sutampa
100	4.6	Pakankamai tikėtina, kad lyginami veidai sutampa
20.1	3	Tikėtina, kad lyginami veidai sutampa
10	2.3	Labiau tikėtina, kad lyginami veidai sutampa
7.4	2	Labiau tikėtina, kad lyginami veidai sutampa
1	0	Vienodai tikėtina, kad lyginami veidai sutampa arba nesutampa
1/7.4	-2	Labiau tikėtina, kad lyginami veidai nesutampa
1/10	-2.3	Labiau tikėtina, kad lyginami veidai nesutampa
1/20.1	-3	Tikėtina, kad lyginami veidai nesutampa
1/100	-4.6	Pakankamai tikėtina, kad lyginami veidai nesutampa
1/403.4	-6	Labai tikėtina, kad lyginami veidai nesutampa
1/1000	-6.9	Labai tikėtina, kad lyginami veidai nesutampa

tikėtina, kad lyginami veidai sutampa. Kad išvengti intervalų $(0, 1)$ ir $(1, \infty)$ dydžio asimetrijos, dažnai vartotojui pateikiama LR natūraliojo logaritmo reikšmė, kuri sutrumpintai žymima $LLR(X, Y) = LLR = \log(LR(X, Y))$ (*angl. Log Likelihood Ratio, LLR*). Jei tikėtinumo santykio logaritmas teigiamas, labiau tikėtina, kad lyginamos poros veidų pavyzdžiai yra vieno asmens. Priešingu atveju, kai tikėtinumo santykio logaritmas yra neigiamas, labiau tikėtina, kad tiriamos poros veidų pavyzdžiai priklauso skirtingiems asmenims. 1 lentelėje pateiktos LR ir LLR reikšmės bei jų interpretacija. Pavyzdžiui, jei $LLR = 6.9$, tai apie

1000 kartų labiau tikėtina, kad lyginamojo ir tiriamojo veidų pavyzdžiai X ir Y priklauso vienam ir tam pačiam asmeniui nei skirtingiems; ir atvirkščiai, jei $LLR = -6.9$, tai apie 1000 kartų labiau tikėtina, kad lyginamojo ir tiriamojo veidų pavyzdžiai priklauso skirtingiems asmenims nei tam pačiam asmeniui. Tikėtinumo santykiui įvertinti naudojami visi turimi lyginamieji veidų pavyzdžiai ir vieno tiriamųjų katalogo veidų pavyzdžiai. Todėl kuo lyginamųjų daugiau, tuo LLR patikimumas didesnis.

Kad pagal (??) apskaičiuoti tikėtinumo santykį LR , reikia įvertinti skaitiklio ir vardiklio reikšmes. Tikėtinumo reikšmės priklauso nuo pasirinkto modelio. Literatūroje modeliai dažnai konstruojami remiantis Gauso skirstiniu. Gauso skirstinys yra simetrinis ir su bet kokiais modelio parametrais modeliuojamas dydis gali įgyti bet kokias reikšmes iš intervalo $(-\infty, \infty)$. Tačiau mūsų atveju modeliuojamos panašumo reikšmės ρ visuomet patenka į $[0, 1]$ uždarą intervalą. Todėl mes modelio pagrindu pasirinkome eksponentinį skirstinį. Laikome, kad kiekvieno fiksuoto tiriamojo X panašumo reikšmių tikėtinumai tenkina tokias lygtis:

$$P(\rho(X, Y) = s | \text{esant prielaidai, kad } X \text{ ir } Y \text{ yra vienodi}) = \lambda \exp(\lambda(s - a_X)) \quad (6)$$

ir

$$P(\rho(X, Y) = s | \text{esant prielaidai, kad } X \text{ ir } Y \text{ yra skirtingi}) = \lambda \exp(-\lambda(s - b_X)). \quad (7)$$

Laikome, kad ir skaitiklio ir vardiklio modelio parametras λ yra vienodas. Šį parametrą galima įvertinti remiantis panašumo reikšmių

dispersija, nes $1/\lambda^2$ yra eksponentinio skirstinio dispersija. Poslinkio parametrai $0 < a_X < b_X < 1$ parenkami individualiai kiekvienam tiriamajam X . Jei šie parametrai yra žinomi ir poros (X, Y) apskaičiuota panašumo reikšmė $\rho(X, Y) = s$, tuomet gauname tokias paprastas formules:

$$LR(X, Y) = \lambda \exp(\lambda(s - a_X)) / (\lambda \exp(-\lambda(s - b_X))) = \exp(\lambda(2s - a_X - b_X)) \quad (8)$$

ir

$$LLR(X, Y) = 2\lambda(s - \frac{a_X + b_X}{2}). \quad (9)$$

Aprašysime modelio parametrų λ , a_X ir b_X įvertinimo procedūrą. $1/\lambda^2$ parametro tikimybinė interpretacija yra veidų panašumo reikšmių dispersija skaičiuojant ją atskirai vienodiems ir skirtingiems veidams. Kadangi lyginamiems veidams turima tiksli informacija kokie veidai sutampa, o kokie skirtingi, λ parametras įvertinamas naudojant tik lyginamųjų veidus $Y_1, Y_2, \dots, Y_L$. Žymėjimų formulėse paprastumo dėlei laikysime, kad visi lyginamieji veidai $Y_1, Y_2, \dots, Y_L$ yra skirtingų asmenų. Tuomet apskaičiuojame visus galimus $L^2 - L$ skirtingų veidų panašumus:

$$\rho(Y_i, Y_j) = s_{i,j}, \quad i, j = 1, \dots, L, i \neq j.$$

Toliau kiekvienoje eilutėje i išsirenkame $K = \lfloor \sqrt{L} \rfloor$ didžiausiųjų:

$$S_{i,k}, \quad k = 1, 2, \dots, K$$

ir dispersiją $1/\lambda^2$ įvertiname pagal įprastą dispersijos įverčio formulę:

$$1/\lambda^2 = \Sigma i = 1^L (\Sigma_{k=1}^K S_{i,k}^2 - (\Sigma_{k=1}^K S_{i,k})^2 / K) / (K - 1) / L.$$

a_X ir b_X ir parametrų įvertis priklauso nuo tiriamojo X . Fiksuojame tiriamąjį veidą X ir apskaičiuojame jo panašumą į visus lyginamuosius $Y_1, Y_2 \dots Y_L$:

$$s_1, s_2, \dots, s_L \quad (s_l = \rho(X, Y_l)).$$

Laikome kad didžiausias panašumas, t.y. $s^{max} = \max_l s_l$, priklauso vieno asmens lygintų veidų porai ir jį naudojame skaitiklio tikėtinumo modelio parametrui a_X įvertinti, postuluoiant, kad $a_X = s^{max}$. Laikome, kad antroji pagal dydį veidų poros panašumo reikšmė $s^{sec} = \max_l \{s_l \text{ išskyrus } s^{max}\}$ atitinka skirtingus asmenis ir ją naudojame apibrėžiant tikėtinumo santykio vardiklio eksponentinio skirstinio modelio parametą b_X , postuluoiant, kad $b_X = s^{sec}$.

1.9 Nuorodos

Naudingos nuorodos parinktos pagal veidų išskyrimo specialistą Robert Frischholz

Nuoroda	Komentarai
Metodikos	Veidų išskyrimo populiariausių metodikų trumpi aprašymai
Straipsniai	Veidų išskyrimo rinktiniai straipsniai ir apžvalgos
Duomenys	Įvairios veidų atpažinimo ir išskyrimo algoritmų testavimo duomenys
Programinė įranga	Nuorodos į laisvai platinimas ir komercines veidų išskyrimo bei atpažinimo programas
Nuorodos	Veidų išskyrimui skirtos nuorodos
Sistemos	Veidų atpažinimo sistemos
Bendruomenė	Veidų atpažinimo bendruomenė
Kita	Veidų iliuzijos ir kita

1.10 Praktinis darbas

Realizuokite veidų atpažinimo algoritmą.

Jūsų sukurtas veidų atpažinimo algoritmas bus vertinamas pagal pateiktų algoritmui nežinomų veidų atpažinimo kokybę.

Reikalingų algoritmui atiderinti veidų pavyzdžius rasite čia. Veidų atpažinimo užduotis pasirenkama laisvai, kai atliktos privalomos užduotys. Darbas atliekamas auditorijoje pratybų metu arba savarankiškai namuose atsiskaitant auditorijoje pratyboms skirtu laiku už atskirus etapus.

1. Pilnos nuotraukos ir akių pozicijų duomenų nuskaitymas, standartinio dydžio veido "iškirpimas" (geometrinė normalizacija).

Perėjimas prie pilkumo lygmenų ir segmentuoto veido vidurkinimas dvimačiu simetriniu eksponentiniu filtru ir fotometriškai normalizuoto veido apskaičiavimas.

Atsiskaitoma pademonstruojant vidurkintą glodžiu eksponentiniu filtru veidą ir jo normalizuotą pilkumo lygmenų variantą.

Vertinama iki 1.25 balo.

2. Veido binarinių požymių apskaičiavimas.

Veidų binarinių požymių palyginimas. Palyginimui naudokite sukurtų binarinių požymių rinkinį ir paprasčiausią panašumo metriką. Palyginkite visas galimas Target ir Query veidų poras, pasirinktų Target ir Query yra nedideli (iki 20) mbgc veidų aibės poaibiai.

Atsiskaitoma parodant DET kreivę. Tema užskaitoma, jei gautos DET kreivės EER (lygios klaidos tikimybė) nedidesnė kaip 30
Vertinama 1.25 balo.

3. Veidų verifikacijos konkursas. Paruoškite veidų verifikacijos algoritmą, kuris apskaičiuotų duotų mokymo (target.list) ir testavimo (query.list) sąrašų porų panašumus.

Jūsų algoritmai bus testuojami su naujais MBGC veidų duomenimis.

Skaičiavimams turite perduoti dėstytojui algoritmus paruoštus tokiu būdu.

- Jūsų vykdomasis ”*.exe” arba java ”*.jar” failas (Matlab, Python ir kitos programavimo terpės neturi būti naudojamos) turi būti paleidžiamas komandine eilute, kurioje papildomai nurodomi du parametrai: target.list ir query.list.

Tarkime, jei jūsų programos vardas ManoVeido.jar, naudojimo pavyzdys būtų

```
java -Xmx1024m -jar ManoVeido.jar mbgcBigTarget.list  
mbgcQuery.list
```

- Jūsų programa turi kiekvieną fiksuotą Query sąrašo elementą (pvz.
F:/mbgc/data/StillChallenge/Query/Original/02463d1006.jpg
) sulygtinti su visais target.list sąrašo elementais ir gautus rezultatus įrašyti į *.roc formato failą.

Pavyzdžiui

<http://www.mif.vu.lt/atpazinimas/veidai/mbgc/Target/Original/mbgcSmallTa>
ir `mbgcQuery.list` sąrašuose yra 20 *target* ir 19 *query*
paveikslukų. Todėl į *ManoVeido.roc* failo pirmuosius
keturis baitus įrašote $20 \times 19 = 380$ sveikąjį skaičių (binarinė
forma).

Į sekančius keturis baitus įrašote 0 - target sąrašo pirmojo
failo numerį.

Sekančiuose keturiuose baituose įrašote taip pat 0 - query
sąrašo pirmąjį numerį.

Kituose keturiuose baituose įrašote 1 (genuines/autentiška
pora), nes $ID(02463d634.png) = 02463 == ID(02463d1006.jpg) = 02463$ (veido ID yra pen-
ki skaitmenys stovintys prieš failo pavadinimo "d" raidę, po
"d" nurodomas filmavimo sesijos/kadro numeris, kuris *.roc
įrašų neįtakoja).

Kiti keturi *ManoVeido.roc* failo baitai patys svarbiausi - juose
įrašote jūsų programa apskaičiuotą

similarity(02463d634.png,02463d1006.jpg)

reikšmę.

Sekančiame keturių baitų ketvertuke pasikeis target įrašo nu-
meris (jis bus lygus 1), genuine/impostors reikšmė bus 0, nes
 $ID(04201d429.png) = 04201 \neq ID(02463d1006.jpg) = 02463$
ir taip pat tikėtina, kad gausite mažesnę *similari-
ty(04201d429.png,02463d1006.jpg)* reikšmę.

Tęsiant tolimesnius skaičiavimus, atkreipkite dėmesį, kad query įrašų indeksai kinta ne po vienetą o sparčiau (tas pats gali būti ir target sąrašo elementams). Tokiu atveju į roc failą įrašote iš *.list failų nuskaitytas reikšmes.

- Kad jūsų programa sugebėtų palyginti pakankamai daug porų, jai keliami veidų poros palyginimo spartos reikalavimai.

Skaičiavimai bus atliekami su Intel Core 2 Duo, 2.1 GHz, 3 Gb Ram PC.

Programa turi palyginti bent 100 veidų porų per sekundę.

Kad pagreitinti skaičiavimus, rekomenduotina pirma apskaičiuoti požymių šablonų target ir query bazes.

Rekomenduojama ekrane atspausdinti informaciją, kad kuriami palyginimui reikalingi šablonai. Maksimalus vieno šablono sukūrimo laikas 1 sek.

- Pilnai paruošta programa vykdymui kartu su programų pradiniais kodais turi būti pateikta iki š. m. gruodžio 15 d. (imtinai).
- Vertinama 1 balo už atitinkančius reikalavimus algoritmo pateikimą.

Papildomai, priklausomai nuo algoritmu gautų rezultatų, skiriama iki penkių konkursinių egzamino balų, kurie atsikirai prisideda prie jūsų pratybų ir teorijos balų.

Literatūra

- [1] Bill Green, <http://www.pages.drexel.edu/~weg22/edge.html>
- [2] Kálmán Palágyi, Vengrija, <http://www.inf.u-szeged.hu/~palagyi/skel/skel.html>
- [3] K. Stukas, J. Janauskas, Š. Gruodis, M. Brašiškis, http://www.mif.vu.lt/bas-tys/academic/ATE/skaiciai/skaiciu_atp.htm#Praktinis
- [4] D. Rutovitz, Pattern Recognition, J. Roy. Statist. Soc., vol. 129, pp. 504-530, 1966.
- [5] Feng Zhao and Xiaou Tang, CISST02 International Conference, http://mmlab.ie.cuhk.edu.hk/2002/CISST02_Fingerprint.pdf,
(Lokali kopija http://mif.vu.lt/bas-tys/academic/ATE/pirshtai/CISST02_Fingerprint.pdf)
- [6] T. Y. Zhang, C. Y. Suen, A fast parallel algorithm for thinning digital patterns, Communications of the ACM, v.27 n.3, p.236-239, March 1984.
Realizacija Java kalba: <http://www.mif.vu.lt/atpazinimas/skaiciai/skelet/ZhangSuen.java>
- [7] Vladimiro Vapniko biografija ir svarbiausi darbai, http://en.wikipedia.org/wiki/Vladimir_Vapnik
- [8] Vapniko-Černovenkio dimensija, http://en.wikipedia.org/wiki/VC_dimension
- [9] Corinna Cortes and V. Vapnik, "Support-Vector Networks", Machine Learning, 20, 1995.

- [10] Chih-Jen Lin video paskaita apie SVM,
http://videlectures.net/mlss06tw_lin_svm/, 2008.
- [11] Klasifikavimo metodų biblioteka WEKA,
<http://www.cs.waikato.ac.nz/ml/weka/>
- [12] L. Sirovich and M. Kirby, Low-dimensional procedure for the characterization of human faces, *Journal of the Optical Society of America*, Vol. 4, Issue 3, pp. 519-524, 1987.
- [13] Matthew Turk and Alex Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience*, p. 71-86, 1991.
- [14] Manfredo Zabarauskas tikrinių veidų metodikos aprašymas ir iliustracija, <http://blog.zabarauskas.com/eigenfaces-tutorial/>
- [15] T. Ahonen, A. Hadid, M. Pietikainen, Face Description with Local Binary Patterns Application to Face Recognition, *Proc. Eighth European Conf. Computer Vision*, p. 469-481, 2004.
- [16] Jie Zou, Qiang Ji, George Nagy, A Comparative Study of Local Matching. *IEEE Transactions on image processing*. Volume 16, NO 10, p.2617-2628, 2007
- [17] Shiguang Shan, Peng Yang, Xilin Chen, and Wen Gao. AdaBoost Gabor Fisher Classifier for Face Recognition, p. 278-290, 2005
- [18] L. Wiskott, J.-M. Fellous, N. Kuiger, C. von der Malsburg, C. Face recognition by elastic bunch graph matching. *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on image processing*. Volume 19, Issue 7, p. 775-779, 1997.

- [19] Wikipedija. Bitiesinē interpoliacija.
- [20] R. Gonzalez, R. Woods, Digital Image Processing, Prentice Hall, Second ed., 2002.
- [21] M. Villegas, R. Paredes, Comparison of illumination normalization methods for face recognition. In: Third COST 275 Workshop - Biometrics on the Internet, University of Hertfordshire, pp. 27-30, 2005.
- [22] M. Villegas, R. Paredes, Illumination Invariance for Local Feature Face Recognition. 1st Spanish Workshop on Biometrics, SWB2007.
- [23] Paul Viola and Michael J. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001.
- [24] O. Jesorsky, K.J. Kirchberg, and R.W. Frishholz. Robust Face Detection Using the Hausdorff Distance. In Proc. Third International Conference on Audio- and Video - based Biometric Person Authentication, Springer, Lecture Notes in Computer Science, LNCS-209, Halmstad, Sweden, pp. 90-95, 2001.
- [25] Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky, Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. MRL Technical Report, Intel Labs, May 2002, revised Dec. 2002.
- [26] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of online learning and an application boosting. In Computa-

tional Learning Theory: Eurocolt '95, pp. 23-37. Springer-Verlang, 1995.

[27] Yarbus, Akių judesiai stebint paveikslą

[28] M.P. Dubuisson and A.K. Jain. A modified Hausdorf distance for object matching. In ICPR94, pages A:566-568, Jerusalem, Israel, 1994.