

Home Page

Title Page

◀◀

▶▶

◀

▶

Page 1 of 48

Go Back

Full Screen

Close

Quit

1. Rainelės atpažinimas

1. Įvadas
2. Privalumai
3. Trūkumai
4. Praktiniai taikymai
5. Rainelės atpažinimo schema
6. Rainelės vaizdo pirminis apdorojimas
7. Akies rainelės duomenų skenavimas
8. Vyzdžio lokalizavimas
9. Vyzdžio ir rainelės kraštų aproksimavimas dviems apskritimais
10. Binarinių požymių išskyrimas
11. Rainelės šablonas
12. Šablonų palyginimas
13. Praktinė užduotis

Įvadas

Rainelė yra žiedo formos matoma akies dalis tarp vyzdžio ir stiklakūnio (žiūr. 3 pav). Rainelės vaizdai pasižymi turtinga tekstūra. Oftalmologas (akių ligų specialistas) Frank Burch 1936 m.

suformulavo mintį apie akių rainelių panaudojimą asmens identifikavimui. Pereito amžiaus 8-ajame dešimtmetyje akies rainelės panaudojimo identifikacijai idėja pasirodė James Bond seriale, tačiau praktškai ji dar nebuvo realizuota. 1987 m. du oftalmologai Aran Safir ir Leonard Flom užpatentavo šią idėją ir 1989 m. jie paprašė John Daugman (Harvardo universiteto lektoriaus) sukurti tinkantį kompiuteriui akių rainelių atpažinimo algoritmą. Daugmano pradinis algoritmas aprašytas [29] straipsnyje. Šiuolaikinės komercinės rainelių atpažinimo sistemos naudoja įvairias originalaus Daugman algoritmo modifikacijas, tačiau principinė atpažinimo schema nesikeičia. Teises į patentus išpirko Iridian Technologies [30] kompanija ir sėkmingai vysto asmenų identifikavimo pagal akies rainelę biznį.

Privalumai

Akies rainelės panaudojimas biometrikoje turi daug gerų savybių.

- Rainelė gerai apsaugota nuo išorinio poveikio labai skaidria membrana. Palyginimui pirštų atspaudai nuolat kinta nuo ranka laikomos cigaretės sukeltų mikro-nudeginimų, įsipjovimų peiliu, mechaninio poveikio dirbant rankomis.
- Rainelė iš esmės yra plokščia ir jos geometrija keičiama tik dviem nepriklausomais raumenimis reguliuojančiais vyzdžio skersmenį. Tai sąlygoja rainelės pastovumą lyginant pavyzdžiui su trimačio veido projekcijomis.

Home Page

Title Page



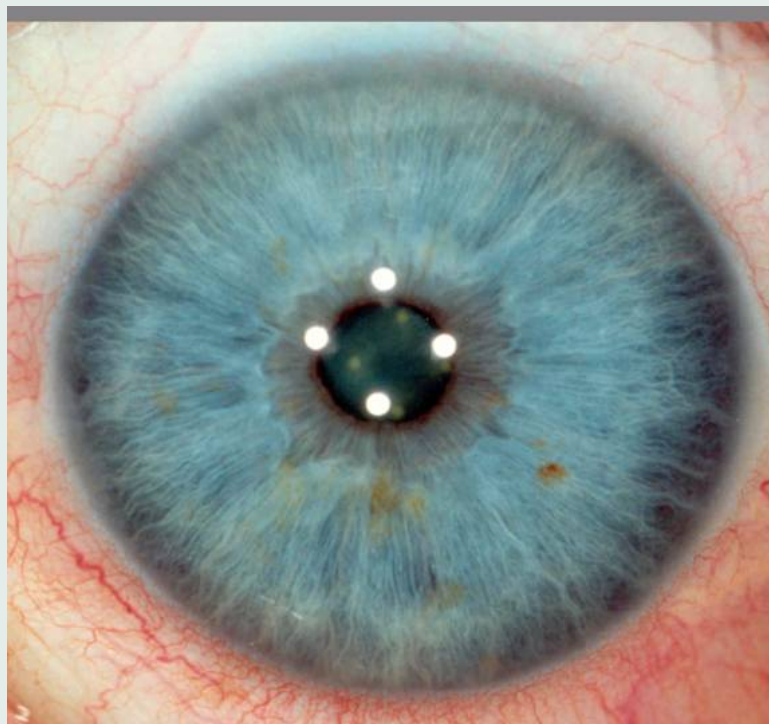
Page 3 of 48

Go Back

Full Screen

Close

Quit



1 paveikslėlis: Žmogaus akies rainelės spalvota nuotrauka

Home Page

Title Page

◀

▶

◀

▶

Page 4 of 48

Go Back

Full Screen

Close

Quit

- Rainelės turi turtingą unikalią tekstūrą, kuri, panašiai kaip ir pirštų atspaudai, atsitiktinai suformuojama embriono vystymosi stadijoje. Net genetiškai identiški dvyniai turi nepriklausomas rainelių tekstūras. Palyginimui pasaulyje yra apie 10 milijonų genetiškai identiškų dvynių, kurių DNR negali būti panaudota jų identifikacijai.
- Rainelės skenavimo procedūra analogiška paprastam fotografavimui atliekamam nuo 10 cm. iki kelių metrų atstumu. Tai žymiai maloniau, nei pavyzdžiui liesti pirštų atspaudų skenavimo mechanizmą, kurį, gal būt, ką tik lietė kitas jums nežinomos higienos kultūros asmuo.
- Yra publikacijų teigiančių, kad apie 200 taškelių akies rainelės skersmens vaizdas yra žymiai informatyvesnis nei pirštų atspaudų vaizdas.
- Šiuo metu komercinis John Daugman IrisCode algoritmas pasižymi unikaliu FAR (*angl. False Acceptance Rate*) rodikliu geresniu nei 10^{-11} . Atlikus daugiau kaip 200 milijardų palyginimų naudojant realius Jungtinių Arabų Emiratų imigracijos kontrolės sistemoje sukauptus duomenis [32] nebuvo nei vieno false acceptance atvejo.
- Nors ir egzistuoja medicininės ir techninės procedūros dalinai keičiančios rainės spalvas ir geometriją, tekstūra iš esmės lieka stabili dešimtmečiais. Yra pavyzdžių sėkmingos rainelių identifikacijos 30 metų laikotarpiu.

Home Page

Title Page

◀◀

▶▶

◀

▶

Page 5 of 48

Go Back

Full Screen

Close

Quit

Trūkumai

- Rainelių skenavimas yra santykinai jauna technologija ir negali lygintis su analogiškais pirštų atspaudų duomenų saugyklomis.
- Identifikuoti rainelę labai keblu didesniu nei keli metrai atstumu ir tuo atveju, kai identifikuojamas asmuo nekooperuoja su sistema nukreipdamas galvą nuo kameros.
- Kaip ir kitos biometrinės sistemos rainelių biometrinė identifikacijos sistema jautri blogos kokybės vaizdams.
- Kaip ir kitų biometrinių identifikavimo sistemų atveju, prieš rainelių duomenų bazių kaupimą pasisako kovos už asmenų teises organizacijos.
- Skirtingai nuo pirštų atspaudų, kol kas nemokama išgauti rainelių vaizdų kopijas paliktas nusikaltimų vietose.

Praktiniai taikymai

Tarptautinė civilių skrydžių organizacija [33] akies raineles, kartu su pirštų atspaudais ir veidais, pripažino ir standartizavo ateities pasų panaudojime. Olandijos Schiphol aerouoste naudojama imigracijos tarnyboje nuo 2001 metų [34]. Jungtinių Arabų Emiratų visuose 17 oro, sausumos ir jūros pasienio

kontrolės punktuose naudojama akies rainelės atpažinimo procedūra taip pat nuo 2001 metų [35]. [36] Anglijoje funkcionuoja IRIS Iris Recognition Immigration System. Keletas Kanados aerouostų taip pat naudoja rainelės atpažinimą skrydžių saugumo kontrolės tikslais.

Rainelės atpažinimo schema

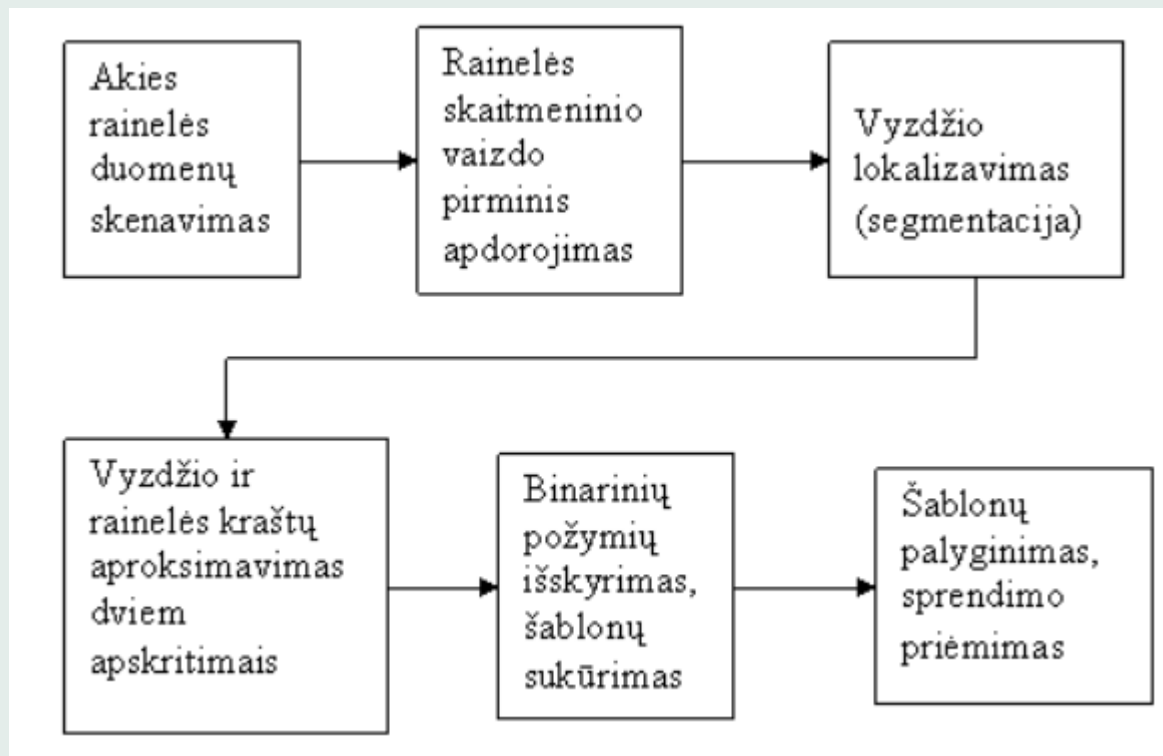
Bendra rainelės atpažinimo schema nesiskiria nuo įprastinės vaizdų atpažinimo schemas. 2 paveikslėlis iliustruoja tipinius akies rainelės atpažinimo etapus.

Trumpai aptarsime kiekvieną rainelės atpažinimo etapą atskirai.

Akies rainelės duomenų skenavimas

Akies rainelės duomenų skenavimas įskaitmenina rainelės analoginį (realų) vaizdą. Duomenų atpažinimui pateikimą skaitmenine forma sąlygoja šiuolaikinių kompiuterių architektūra. Kadangi spalvinė rainelių informacija mažiau atspari laikui ir medicininėms

Priemonėms, spalvinė rainelių informacija šiuo metu komercinėse atpažinimo sistemose nenaudojama. 3 pav. pateiktas vienas nespaltvotos rainelės nuotraukos pavyzdys. Atkreipkite dėmesį į ryškia baltą dėmę vyzdžio srityje (23 val. kryptimi). Tokios dėmės kaip taisyklė matomos skaitmeninėse rainelių nuotraukose ir gaunasi dėl specifinio apšvietimo šaltinio išryškinančio infraraudonąsias elektromagnetines vaizdo bangas. Kitas skenavimo būdo sąlygotas defektas yra taip vadinamas interlacing [38], kuris pasireiškia lyginių ir nelyginių vaizdo eilučių pasislinkimu, kuris atsiranda dėl skenavimo



2 paveikslėlis: Žmogaus akies raielės atpažinimo etapai

pirma lyginių, o po to nelyginių vaizdo eilučių.

Rainelės vaizdo pirminis apdorojimas

Pirminis rainelės vaizdo apdorojimas, kaip ir kitais vaizdų atpažinimo atvejais, yra mažiausiai standartizuotas. Mes naudodami pirminį vaizdo apdorojimą spręsimė tris problemas.

1. Eliminuosime vaizdo kokybės pablogėjimą sąlygotą lyginių/nelyginių eilučių skenavimo ypatumu (vadinamu interlacing efektu).
2. Pašalinsime apšvietimo šaltinio sąlygotus ryškius atšvaitus.
3. Apskaičiuosime kaupiamąjį vaizdą (angl. *Integral image*), kuris bus naudingas lokalizuojant vyzdį vaizde.

Egzistuoja daug ir gana sudėtingų interlacing efekto sukeltų vaizdo iškraipymo problemos sprendimų. Šie vaizdo iškraipymai atsiranda dėl to, kad dalis skenerių pirma skenuoja lygines $(0, 2, \dots)$, o po to nelygines $(1, 3, \dots)$ eilutes. Dėl to atsiranda skenavimo laikų skirtumai ir akis gana dažnai spėja kiek pakeisti gretimų eilučių skenavimo pozicijas. Vienas būdų sumažinti šio efekto sąlygotus vaizdų iškraipymus yra eliminuoti kas antrą (pavyzdžiui nelyginę) vaizdo eilutę. Kadangi mums svarbu išsaugoti apskritiminę rainelės simetriją mes eliminuosime ne tik nelygines vaizdo eilutes bet ir nelyginius vaizdo stulpelius. Kadangi rainelės nuotraukos kokybės būna aukštos

Home Page

Title Page



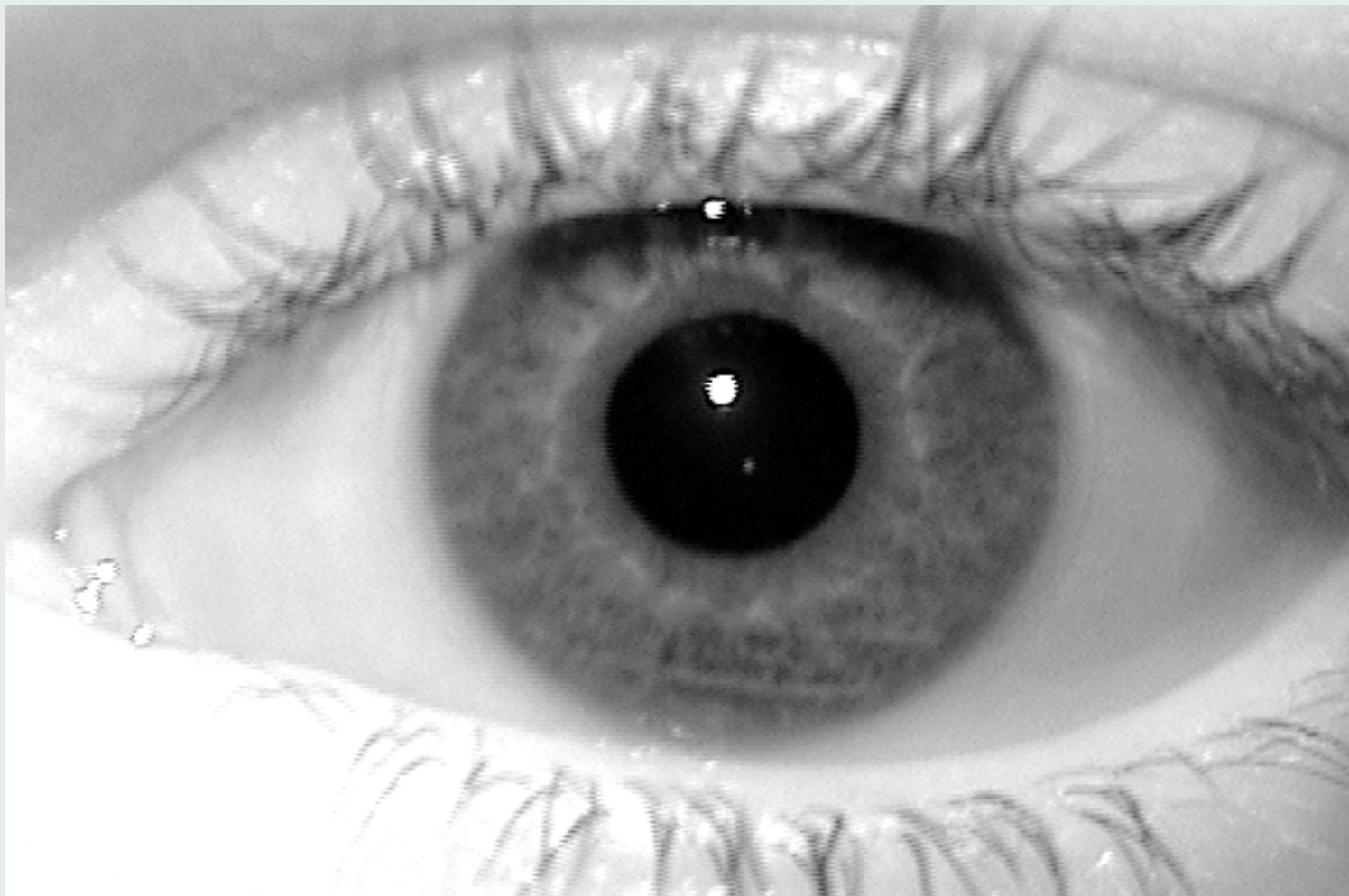
Page 9 of 48

Go Back

Full Screen

Close

Quit



3 paveikslėlis: NIST ICE 2005 rainelių duomenų bazės 242116.tiff failo rainelės vaizdas.

Home Page

Title Page



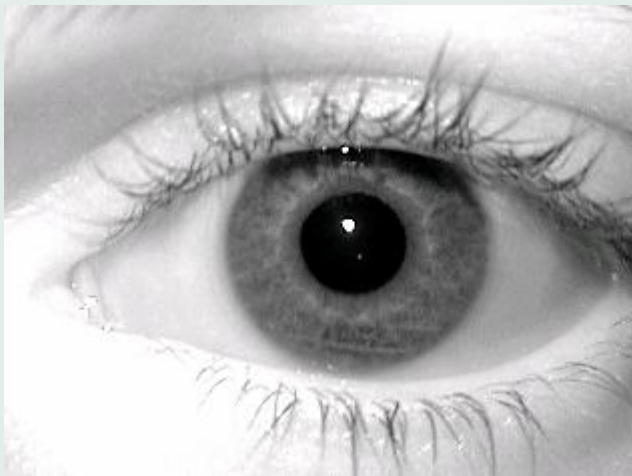
Page 10 of 48

Go Back

Full Screen

Close

Quit



4 paveikslėlis: 242116 vaizdas, kuriame eliminuotas lyginių/nelyginių eilučių triukšmas

rezoliucijos, sumažintas du kartus imčių skaičius horizontalia ir vertikalia kryptimis išsaugos gana detalią informaciją apie rainelės tekstūrą ir tuo pačiu apie 4 kartus paspartins vėlesnius skaičiavimus atliekamus su vaizdo taškelių matrica. Atlikus šią parastą deinterlacing procedūrą gausime 4 paveikslėlyje pavaizduotą rainelės vaizdą.

Pradiniam vaizdelyje interlacing iškraipymai lengvai pastebimi balto atšvaito bei atskirų blakstienų srityje. 4 pav. vaizdo kokybė šia prasme yra geresnė. Sumažintos rezoliucijos vaizde rainelės tekstūra išlieka pakankamai aiški, kad remiantis ja būtų galima identifikuoti skirtingas

Home Page

Title Page

◀◀

▶▶

◀

▶

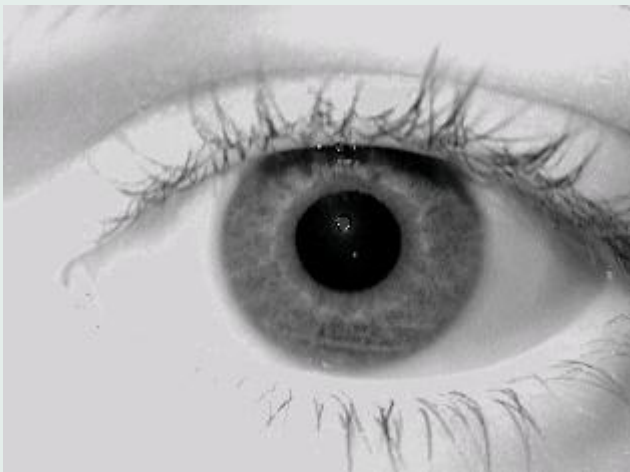
Page 11 of 48

Go Back

Full Screen

Close

Quit



5 paveikslėlis: Rainelės vaizdas pašalinus didelio intensyvumo sritis (apšvietimo atšvaitus).

rainelės.

Baltus apšvietimo atšvaitus pašalinsime tokiu būdu. Pirma pasirinksime šviesumo slenkstį $trshld$ (pvz. $trshld = 220$). Tuomet surandame jungią taškų (x, y) sritį vaizde, kurioms vaizdo reikšmės $img(x, y) > trshld$. Fiksuojame kokią nors tokių taškų sritį ir užpildome jos reikšmes srities krašto vaizdo reikšmių vidurkiu. Atlikę šias procedūras gauname 4 pav. rainelės vaizdą.

Kaupiamasis vaizdas $IImg(x, y)$ (angl. *Integral Image*) gaunamas susumuojant visas $img(i, j)$ reikšmes, kur (i, j) koordinatės tenkina sąlygas $i \leq x$ ir $j \leq y$. Šio vaizdo nevizualizuosime, nes

kaupiamasis vaizdas skirtas ne vizualizacijai, o yra pagalbinė priemonė greitų vaizdo vidurkių bet kokiame stačiakampyje apskaičiavimui kuo mes vėliau ir pasinaudosime. Plačiau apie kaupiamąjį vaizdą galite paskaityti [39] šaltinyje.

Vyždžio lokalizavimas

Objekto lokalizacija yra sudedamoji daugelio vaizdų atpažinimo algoritmų dalis. Lokalizuojant kokį nors objektą reikia surasti sritį ar sritis vaizde kuriose yra mus dominantis objektas nekreipiant dėmesio į individualias objekto detales pagal kurias vėliau yra identifikuojamas konkretus objektas. Java programavimo terminais lokalizuojant reikia atpažinti Java klasės tipą, o atpažinimas ekvivalentus kokio nors klasės konkretaus egzemplioriaus identifikavimui. Rainelės atveju patogiau pradžioje lokalizuoti ne visą rainelę, o tik sritį vaizde, kuri atitinka vyzdį. Užpildžius baltus atšvaitus, vyzdys pasižymi mažo intensyvumo $img(x, y)$ reikšmėmis. Tačiau blakstienos, ypač merginų, kurios specialiai ryškina jas juodu tušu, taip pat pasižymi mažomis intensyvumo reikšmėmis. Kadangi blakstienos yra nedidelio storio vidurkintame vaizde blakstienas atitinkančios reikšmės vidurkinsis su gretimomis didesnėmis veido spalvos reikšmėmis, o vyždžio vidurio taškai išliks tokie pat tamsūs. Kad greitai atlikti vidurkinimo operaciją patogiu naudoti kaupiamąjį vaizdą. Stačiakampyje, kurio viršūnės taškuose $(a - 1, b - 1)$ ir (c, d) , $(a < c, b < d)$ esančių vaizdo reikšmių $img(x, y)$ suma yra lygi

$$IImg(c, d) - IImg(a, d) - IImg(c, b) + IImg(a, b).$$

Vadinasi šio stačiakampio vidutinė vaizdo reikšmė bus randama keletu sudėties/atimties bei daugybos/dalybos veiksmiais:

$$\overline{img}_{(c,d) \times (a-1,b-1)} = \frac{IImg(c,d) - IImg(a,d) - IImg(c,b) + IImg(a,b)}{(c-a)(d-b)}.$$

Kad vidurkintas vaizdas būtų labiau lokalizuotas mes vidurkiname du vidurkintus vaizdus, kurių stačiakampiai lokalizuoti arba horizontalia arba vertikalia kryptimi. Galutinė vidurkinto vaizdo reikšmė taške (x, y) apskaičiuojama formule

$$\overline{img}(x, y) = 0.5 * (\overline{img}_{(x+1,y+h) \times (x-1-1,y-h-1)} + \overline{img}_{(x+h,y+1) \times (x-h-1,y-1-1)}).$$

Čia h yra vidurkinimo stačiakampio didžiosios kraštinės pusė.

6 pav. iliustruoja gaunamą vidurkinimo rezultatą ($h = 10$). Alternatyvus būdas gauti vidurkintą raišelės vaizdą yra pritaikyti simetrinį eksponentinį filtrą aprašytą vaizdų vidurkinimo skyriuje.

Detektuojant vyzdį vaizde yra ieškomas globalus vidurkinto vaizdo minimumas

$$\arg\text{Min}(\overline{img}(x, y)) = (x_{\min}, y_{\min})$$

ir surandama jungi taškų sritis, kurios elementai (x, y) tenkina sąlygą

$$\overline{img}(x, y) \geq \overline{img}(x_{\min}, y_{\min}) + \delta.$$

Home Page

Title Page

◀◀

▶▶

◀

▶

Page 14 of 48

Go Back

Full Screen

Close

Quit



6 paveikslėlis: 20x20 kvadratuose vidurkintas rainelės vaizdas

Slenksčio reikšmė δ parenkama eksperimentiniu būdu ir 256 pilkumo lygių vaizde tipinė reikšmės yra $\delta = 8, 16$ arba 32 .

7 pav. iliustruoja gaunamo vyzdžio lokalizacijos rezultatą. Rausvais taškais pažymėtas išskirtos jungios vyzdžio srities kraštas. Šiuo atveju delta reikšmė buvo 32 .

Suradus akies vyzdžio sritį randamas šios srities taškų koordinatų aritmetinis vidurkis. 7 pav. taškas su aritmetinio vidurkio koordinatėmis pažymėtas mažu raudonu kryžiu. Taip pat apytiksliai įvertinamas vyzdžio spindulys, kurio reikšmė randama apskaičiuojant atstumų vidurkį tarp išorinio

Home Page

Title Page

◀◀

▶▶

◀

▶

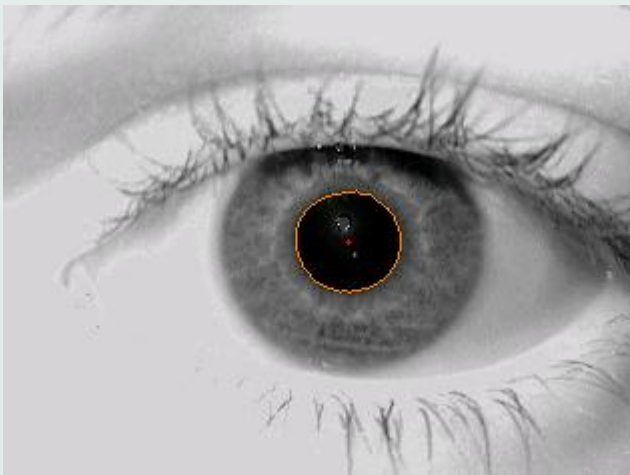
Page 15 of 48

Go Back

Full Screen

Close

Quit



7 paveikslėlis: Lokalizuoto vyzdžio vaizdas. Rausvi taškai žymi vyzdžio srities kraštinius taškus, raudonas - vyzdžio srities taškų koordinatų aritmetinį vidurkį

vyzdžio kontūro taškų ir rasto vyzdžio centro. 7 pav. pavaizduotos akies atveju gaunama vyzdžio spindulio reikšmė yra apytiksliai lygi 28 taškeliams.

Vyždžio ir rainelės kraštų aproksimavimas dviems apskritimais

Sekačiame etape yra patikslinamas vyzdžio centras bei jo kraštą aproksimuojantis apskritimas ir surandamas analogiškas apskritimas rainelės išoriniam kraštui. Kad išryškinti kraštus dažnai naudojamas diferencijavimas. Vieno kintamojo funkcijos pirmosios eilės išvestinė greito reikšmių kilimo

sirtyje turės lokaly maksimumą, o greito kritimo srityje minimumą. Antrosios eilės išvestinė ir maksimumo ir minimumo taškų aplinkose kirs nulinę reikšmę. Dažniausiai tokie taškai ir yra naudojami detektuoti kraštams. Mūsų atveju patogų naudoti jau detektuotą vyzdžio centrą nustatyti kryptčiai kuria yra skaičiuojama antrosios eilės išvestinė. Tiksliau antrosios eilės išvestinė yra aproksimuojama antrosios eilės skirtumais spindulio išvesto iš vyzdžio centro kryptimi. Tokiu būdu vaizdo $img(x, y)$ antrosios eilės diferencialas spindulio kryptimi randamas pagal formulę

$$\frac{d^2 img(x, y)}{dr^2} = \frac{img(x + dr \cos(\phi), y + dr \sin(\phi)) - 2img(x, y) + img(x - dr \cos(\phi), y - dr \sin(\phi))}{(dr)^2}.$$

Čia (r, ϕ) yra taško (x, y) polinės koordinatės randamos iš lygčių

$$x = x_o + r \cos(\phi),$$

$$y = y_o + r \sin(\phi),$$

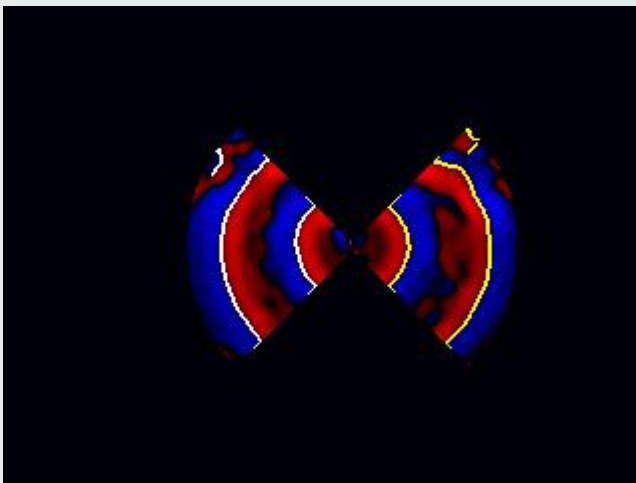
(x_o, y_o) detektuoto vyzdžio centro koordinatės, dr diferencialo žingsnis, kurio praktinė reikšmė gali būti apie 5 taškelius. Įstačius konkrečias reikšmes daugeliu atvejų gausime nesveikaskaites vaizdo koordinatės. Todėl tenka vaizdo reikšmes bitiesiškai interpoliuoti panaudojant 4-ias vaizdo reikšmes artimiausiose sveikaskaitėse koordinatėse (plačiau žiūr. [40]). Kad neskaičiuoti be reikalo stipriai nutolusiuose nuo vyzdžio centro vaizdo taškuose diferencialų galima apsiriboti maksimalaus rainelės spindulio skrituliu. Kadangi dažnai į rainelės sritį patenka apatinis ir viršutinis akies vokas taip pat

tikslingas apsiriboti tik 45 laipsnių sektoriais išeinančiais iš vyzdžio centro, kurių simetrijos ašis yra lygiagreči horizonto linijai. 8 pav. iliustruoja apskaičiuotas nurodytoje srityje antrosios eilės diferencialo reikšmes. Kuo spalva raudonesnė tuo didesnė diferencialo teigiama reikšmė pažymėtame taške, kuo spalva mėlynėsnė tuo reikšmė didesnė neigiama, reikšmės artimos nuliui arba kurios neskaiciuotos nuspaldintos juodai. Balta spalva žymi perėjimo iš mėlynos spalvos į raudoną taškus (t.y. antrosios eilės diferencialo spindulio kryptimi nulinio kirtimo taškus, o geltona analogiškus taškus kai spalva žiūrint iš kairės į dešinę iš raudonos pereina į mėlyną. Kad greitai ir tiksliai įvertinti apskritimų ribojančių vyzdžio ir rainelės kraštus sudaroma nulinio kirtimo taškų nuotolių iki vyzdžio centro histograma. Histograma skaičiuojama atskirai kiekvienam jungiam nulinio kirtimų kontūrai. Gautas histogramas taip pat galima kiek pavidurkinti, kad jų grafikai būtų tolydesni.

9 pav. iliustruoja gaunamas histogramas. Dvi beveik susiliejančios histogramos turi maksimumus atitinkančius argumento reikšmes 55 ir 56. Šios histogramos atitinka kairinį ir dešinįjį vyzdžio kontūrą, o maksimumai atitinka vyzdį aproksimuojančio apskritimo spindulį. Prisiminkime, kad pradinė įvertinta apskritimo reikšmė buvo lygi 28. Beveik dvigubas spindulio padidėjimas gavosi todėl, kad pereidami prie polinių koordinačių mes pasmulkinome du kartus skaičiuojamų taškų mastelį kad gauti tikslesnes vyzdžio ir rainelės kraštus ribojančių apskritimų reikšmes. Kitų dviejų reikšmingų histogramų grafikų maksimumai yra dešiniau ir jų reikšmės yra 130 ir 136. Šios dvi histogramos

[Home Page](#)[Title Page](#)[<<](#)[>>](#)[<](#)[>](#)

Page 18 of 48

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

8 paveikslėlis: Antrosios eilės diferencialai apskaičiuoti riboto apskritimo ilgio 45 laipsnių sektorių, išeinančių iš vyzdžio centro, srityse

atitinka rainelės dešinio ir kairiojo išorinio krašto kontūrus, o jų maksimumai rainelės kraštus aproksimuojančių apskritimų spindulius. Šiuo atveju gautų spindulių reikšmės skiriasi gana daug, tiksliau $136130 = 6$ taškeliais. To priežastis yra ta, kad vyzdžio centras netiksliai sutampa su rainelę ribojančio apskritimo centru. Todėl papildomai apjungiamo atskirai taškus sudarančius du kontūrus (kairinį ir dešinį), kurie riboja vyzdį, bei taškai analogiška rainelės kraštų kontūrų taškų aibė. Šioms dvejoms taškų aibėms surandami du apskritimai minimizuojantys atstumus kontūrų taškų iki apskritimų. Į

šių apskritimų apskaičiavimo detales nesigilinsime, pažymėsime tik kad gaunamų apskritimų centrai ir spinduliai ir yra galutinis segmentacijos/lokalizacijos rezultatas.

10 pav. iliustruoja galutinį rainelės lokalizacijos algoritmo darbo rezultatą. Žaliasis apskritimas ir jo centro taškas aproksimuoja akies vyzdį ribojantį kontūrą, kuris sutampa su rainelės vidiniu kontūru, o geltonai nuspalvintas apskritimas ir centro taškas žymi rainelės išorinio krašto kontūrą aproksimuojantį apskritimą. Originalus Daugman ir kiti algoritmai papildomai dar detektuoja apatinį ir viršutinį voką aproksimuojančius apskritimus. Kadangi šiems apskritimams išskirti netinka histograma sudaryta iš kontūro taškų atstumų iki vyzdžio centro, šiuo atveju reikėtų taikyti apskritiminę Hough transformaciją [41]. Hough transformacijos sudėtingumas yra žymiai sudėtingesnis, nes ieškomas ne vienas parametras apskritimo spindulys, bet dar papildomi du centro taško koordinatės, todėl mes savo algoritme jo nenaudojame. Būsiu dėkingas, jei kas pasiūlysite greitus ir patikimus algoritmus akių vokų kraštams surasti. Čia aprašytas algoritmas yra realizuotas Java kalba ir vienam 3 pav. pateiktos rezoliucijos vaizde lokalizuoja rainelę vidutiniškai per 0.25 sek. (2.8 GHZ PC).

Binarinių požymių išskyrimas

Atlikus segmentaciją atsiranda galimybė išskirti požymius. Apskaičiuodami požymius remsimės Daugman tradicija, t.y. rainelės požymiai bus binarinių skaičių, t.y. 0/1, matrica. Tokia binarinių požymių patogi palyginimui nes galima apjungti 0/1 grupėmis, pvz. po 16 pozicijų ir lyginimo metu

Home Page

Title Page

◀ ▶

◀ ▶

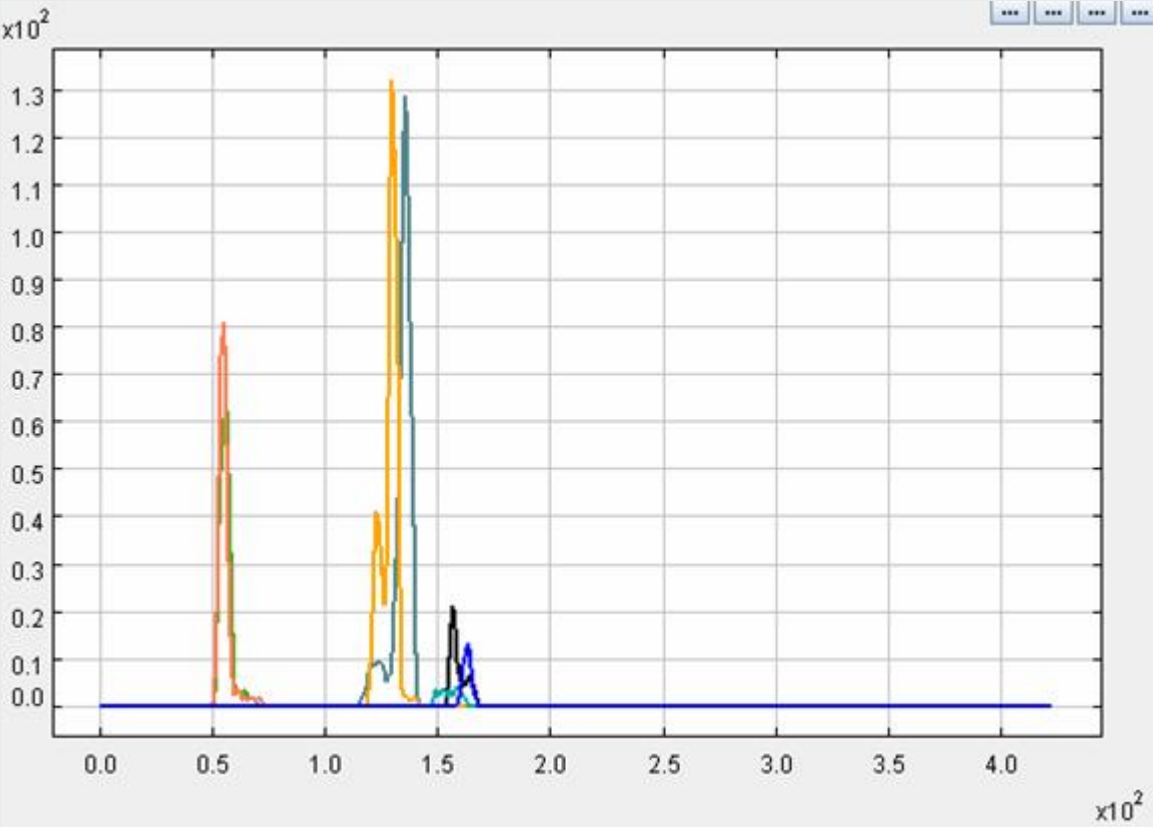
Page 20 of 48

Go Back

Full Screen

Close

Quit



9 paveikslėlis: Jungių kontūrų taškų atstumų iki vyzdžio centro histogramos

Home Page

Title Page

◀

▶

◀

▶

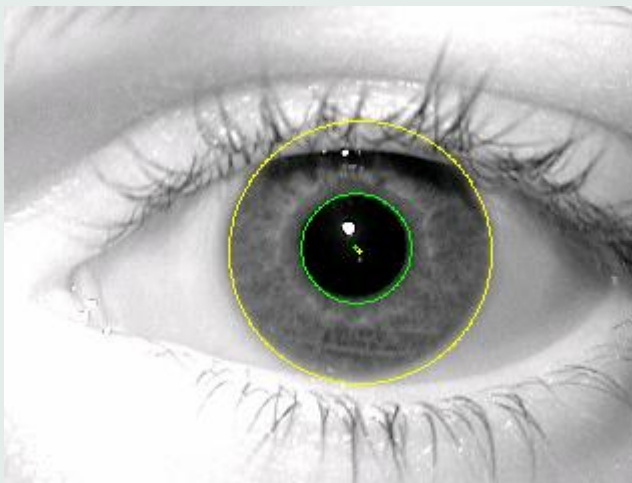
Page 21 of 48

Go Back

Full Screen

Close

Quit



10 paveikslėlis: Vyzdį ir akies rainelės išorinę kontūrą aproksimuojantys apskritimai

operuoti su tomis grupėmis, kas žymiai pagreitina skaičiavimus. Originaliame Daugmas darbe yra naudojami Gaboro filtrai [42]. Mes naudosime paprastesnius filtrus, kurių esmė išlieka tapati filtrų svoriai parenkami taip, kad jie trivialius signalus, pvz. konstantą atvaizduotų į nulį ir konkreti požymio reikšmė parenkama pagal filtravimo rezultato ženklą.

Pradžioje reikia nuspręsti kokį signalą filtruosime. Atlikę rainelės segmentaciją gauname du apskritimus, kurių mažasis aproksimuoja kontūrą ribojantį vyzdį ir vidinį rainelės kraštą, o didysis aproksimuoja išorinį rainelės kontūrą. Pradžioje tarkime paprastumo dėlei, kad šių dviejų apskritimų

centrai sutampa. Tuomet išskirta rainelė sudaro žiedą, ribojamą R_{pupil} ir R_{iris} spinduliais. Daugman schemeje požymiams išskirti yra pereinama prie vaizdo polinių koordinačių sistemos, kurios centras sutampa su išskirtų apskritimų centru, o reikšmės sudaro stačiakampę taškelių matricą, gaunamą keičiant polinio kampo ir spindulio reikšmes. Jei (x_o, y_o) pažymėti rasto rainelės centro koordinatės, tai gausime tokias formules:

$$\begin{aligned} img_{n,m} &= img(x_o + r_n \cos(\phi_m), y_o + r_n \sin(\phi_m)) \\ \phi_m &= \frac{2\pi m}{M}, \quad m = 0, 1, \dots, M-1, \\ r_n &= R_{pupil} + \frac{(R_{iris} - R_{pupil}) * (n + 0.5)}{N}, \quad n = 0, 1, \dots, N-1. \end{aligned}$$

Praktiškai taikant šias formules tenka spręsti vaizdo reikšmės $img(x, y)$ apskaičiavimo problemą, nes nuskaitytas pilkumo lygmenų rainelės vaizdas yra žinomas tik sveikaskaitėse koordinatėse, t.y. taškuose $(x, y) = (i, j)$, kur i ir j yra sveikieji skaičiai. Mūsų atveju $x = x_o + r_n \cos(\phi_m)$ ir $y = y_o + r_n \sin(\phi_m)$ yra realūs skaičiai kurie tik retais atvejais bus sveikieji. Ši problema sprendžiama naudojant bitiesinę interpoliaciją, kurios esmė yra tokia.

1. Taškui (x, y) surandami artimiausi 4-i taškai su sveikaskaitėmis koordinatėmis (žiūr. 11 pav.).

Naudojant Java programavimo operatorius (i, j) taško koordinatės bus randamos taip:

```
int i = (int) x;
```

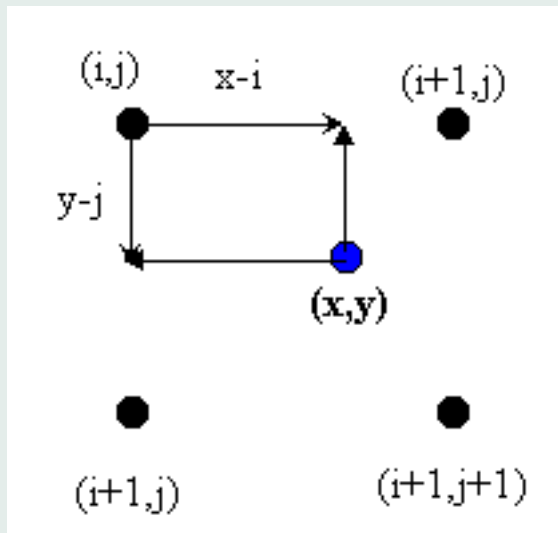
```
int j = (int) y;
```

2. Vaizdo reikšmė taške (x, y) apskaičiuojama formule

$$\begin{aligned} img(x, y) = & img[i, j] * (1 - dx) * (1 - dy) + img[i + 1, j] * dx * (1 - dy) + \\ & img[i, j + 1] * (1 - dx) * dy + img[i + 1, j + 1] * dx * dy. \end{aligned} \quad (1)$$

Čia $dx = x - i$, $dy = y - j$. Nesunku pastebėti, kad fiksuotam x arba y išraiška $img(x, y)$ yra tiesinė likusio (y arba x) kintamojo atžvilgiu. Jei x ir y yra sveikaskaitės, t.y. $x = i$ ir $y = j$, tai formule apskaičiuota reikšmė $img(x, y)$ sutampa su vaizdo reikšme $img[i, j]$. Dėl šių dviejų savybių formulė ir yra vadinama *bitiesine interpoliacija*.

Modifikuosime rainelės reikšmių apskaičiavimo išraiškas, kad jos tiktų bendru atveju kai mažojo ir didesniojo apskritimo centrai yra skirtingi. Formulų modifikacijos idėja yra gana paprasta formulėse reikia pakeisti fiksuotą centrą (x_o, y_o) kintamu centru $(x_o(r), y_o(r))$ taip, kad kintant spinduliui r nuo minimalios reikšmės $r = R_{pupil}$ iki maksimalios R_{iris} centro koordinatės išraiška $(x_o(r), y_o(r))$ kistų tiesiškai nuo (x_{opupil}, y_{opupil}) (vyzdžio centro taškas) iki (x_{oiris}, y_{oiris}) (rainelės centro koordinatės).



11 paveikslėlis: Taško (x, y) artimiausios sveikaskaitės koordinatės naudojamos bitiesinei interpoliacijai

Atlikus šią modifikaciją gauname tokias formules:

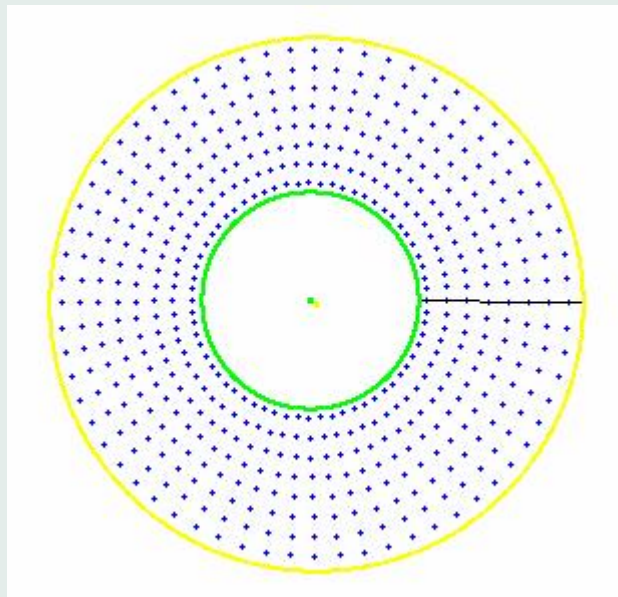
$$iris_{n,m} = img(x_o(r_n) + r_n \cos(\phi_m), y_o(r_n) + r_n \sin(\phi_m)),$$

$$\phi_m = 2\pi n/M, \quad m = 0, 1, \dots, M-1,$$

$$r_n = R_{pupil} + (R_{iris} - R_{pupil})(n + 0.5)/N, \quad n = 0, 1, \dots, N-1,$$

$$x_o(r) = x_{opupil} + (r - R_{pupil})/(R_{iris} - R_{pupil})(x_{oiris} - x_{opupil}),$$

$$y_o(r) = y_{opupil} + (r - R_{pupil})/(R_{iris} - R_{pupil})(y_{oiris} - y_{opupil}).$$



12 paveikslėlis: Rainelės $iris(n, m)$ gardelės taškai, kuriuose, taikant bitiesinę interpoliaciją, apskaičiuojamos rainelės pilkumo lygmenų reikšmės

12 pav. mėlynais taškeliais pavaizduoti taškai $(x_{n,m}, y_{n,m})$, kuriuose taikant bitiesinę interpoliaciją apskaičiuojamos $iris_{n,m}$ reikšmės

Sekantis 13 paveikslėlis iliustruoja gautų bitiesine interpoliacija pilkumo lygmenų taškelių matricą. Kaip ir anksčiau, iliustracijai gauti naudojamas NIST ICE 2005 [37] rainelių duomenų bazės 242116.tiff



13 paveikslėlis: Rainelės pilkumo lygmenų 64×512 taškelių vaizdas

failo rainelės vaizdas, 12 pav. atveju $N = 8$, $M = 64$, 13 pav. $N = 64$, $M = 512$, juoda 12 pav. linija žymi atkarpą ties kuria buvo perkirtas rainelės žiedas pavaizduojant jį 13 pav. pilkumo lygmenų taškeliais, apatiniai 13 pav. taškai atitinka rainelės vidinio apskritimo taškus.

Turint stačiakampę rainelės pilkumo lygmenų matricą galima pradėti išskyrinėti binarinius požymius. Požymiams skaičiuoti pasirenkamas koks nors filtras, juo filtruojamas rainelės vaizdas ir pagal filtravimo rezultato ženklo reikšmę priskiriama galutinė binarinio požymio reikšmė. Daugmanas filtravimui naudojo dvimatį Gaboro filtrą [42]. Dvimačiai Gaboro filtrai horizontalia ir vertikalia kryptimi turi po du laisvus parametrus, kurių konkrečios reikšmės literatūroje nenurodomos, todėl praktiškai neįmanoma atkartoti autorių gaunamų rainelių atpažinimo rezultatų gaunamų taikant šį filtravimą. Todėl mes naudosime paprastesnius vienmačius filtrus ir nurodysime konkrečias filtrų koeficientų reikšmes. Kadangi binariniai požymiai gaunami remiantis filtravimo rezultato ženklu, aišku, kad filtro koeficientų suma turi būti lygi nuliui. Jei pavyzdžiui vidutinė filtro koeficientų reikšmė būtų teigiama, tai filtruojant rainelės vaizdo reikšmes gautume dominuojančią teigiamą

filtravimo rezultato reikšmę, nes visos rainelės reikšmės yra neneigiamos ir kinta intervale $[0, 255]$. Paprasčiausias vienmatis netrivialus filtras, kurio koeficientų suma lygi nuliui, yra

$$f_m = \begin{cases} -1, & m = 0, \\ 1, & m = 1, \\ 0, & m \neq 0, 1. \end{cases}$$

Pritaikius šitą filtrą vienmačiam signalui u_n gausime tokį filtravimo rezultatą:

$$(u * f)_m = u_m - u_{m-1},$$

kuris yra signalo u išvestinės aproksimacija. Atlikus požymio binarizaciją, pozicijoje m rašytume 1, jei u_m didesnis už u_{m-1} , ir 0 priešingu atveju, kai u_m mažesnis arba lygus u_{m-1} . Praktiškai įsitikinta, kad daugeliu atvejų geriau analizuoti ne funkciją arba jos išvestinę, bet jos antrosios eilės išvestinę. Antrosios eilės išvestinės skirtuminis analogas apibrėžiamas filtru

$$f_m = \begin{cases} 1, & m = 1, \\ -2, & m = 0, \\ 1, & m = -1, \\ 0, & m \neq -1, 0, 1. \end{cases}$$

Pritaikius šį filtrą vienmačiam signalui gauname tokį rezultatą:

$$(u * f)_m = u_{m+1} - 2u_m + u_{m-1}.$$

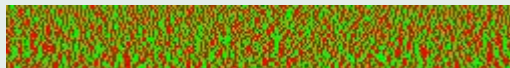
Šiuo atveju binarinis požymis taške m būtų apibrėžiamas išraiška

$$bin_m = \begin{cases} 1, & \text{kai } \Delta u_m = u_{m+1} - 2u_m + u_{m-1} \geq 0, \\ 0, & \text{kai } \Delta u_m < 0. \end{cases}$$

Rainelės atveju turime dvimatį signalą $iris(n, m)$. Skaičiuoti signalo pokyčius spindulio kryptimi gana rizikinga, nes rainelės vidinis ir ypač išorinis kontūras neturi aiškiai apibrėžtos ribos. Todėl spindulio (indekso n) reikšmė r_n yra fiksuojama ir skaičiuojami pokyčiai kampo (indekso m) kitimo atžvilgiu. Tokiu būdu gauname binarinių požymių matricą

$$bin_{n,m} = \begin{cases} 1, & \text{kai } iris_{n,m+1} - 2iris_{n,m} + iris_{n,m-1} \geq 0, \\ -1, & \text{kai } iris_{n,m+1} - 2iris_{n,m} + iris_{n,m-1} < 0, \\ n = 0, 1, \dots, N-1, m = 0, 1, \dots, M-1. \end{cases}$$

Taip skaičiuojami požymiai yra gana nestabilūs atsitiktiniam triukšmui. 14 pav. pavaizduota 32×256 binarinių požymių matrica $bin_{n,m}$ apskaičiuota 242116.tiff paveikslui imant 32 fiksuotas r_n ($N = 32$) ir 256 ϕ_m ($M = 256$) reikšmės. Žali taškeliai žymi reikšmes $bin_{n,m} = 0$, o raudoni taškus $bin_{n,m} = 1$.



14 paveikslėlis: 242116.tiff failo rainelės binariniai požymiai skaičiuojami pagal rainelės antrosios eilės išvestinės skirtuminio analogo skaičiuojamo kampo kryptimi

Iš paveikslėlio matyti, kad dviejų autentiškų rainelių binarinių požymių matricos gali stipriai skirtis dėl smulkaus mastelio.

Kad spręsti šią problemą atliksime skaičiuojamų požymių vidurkinimą spindulio ir kampo kryptimi. Paprasčiausiu atveju gaunamas vidurkinimo rezultatas bus ekvivalentus filtravimui su tokiu dvimačiu filtru:

$$f_{n,m} = \begin{bmatrix} 0.5 & 0.5 & -1 & -1 & 0.5 & 0.5 \\ 1 & 1 & -2 & -2_{0,0} & 1 & 1 \\ 0.5 & 0.5 & -1 & -1 & 0.5 & 0.5 \end{bmatrix},$$

$$(u * f)_{n,m} = \sum_{j=-1}^1 \sum_{i=-3}^2 u_{n-j,m-i} f_{j,i}.$$

Čia $2_{0,0}$ žymi filtro $f_{n,m}$ centrinių koeficientų atitinkantį indeksų reikšmes $n = 0$ ir $m = 0$. Taikant šias formules laikoma, kad $u_{n-j,m-i}$, kai $m > i$ arba $m \geq M + i$, yra gaunamos periodu M pratęsiant reikšmes, t.y. $u_{n-j,m-i} = u_{n-j,(M+m-i)\%M}$. Toks periodinis pratęsimas yra natūralus, nes pradinis

Home Page

Title Page

◀

▶

◀

▶

Page 29 of 48

Go Back

Full Screen

Close

Quit



15 paveikslėlis: Binarinių požymių matrica gaunama filtruojant antrosios eilės išvestinę atitinkančių skirtuminiu filtru vidurkintu 2 taškeliais kampo ir minimaliu trikampių vidurkinimu spindulio kryptimi. rainelės signalas yra periodinis kampo kryptimi. Kadangi spindulio kryptimi signalas nėra periodinis, tai pateiktose dvimačio filtravimo formulėse laikoma, kad $u_{n-j,m-i} = 0$, kai $n < j$ arba $n \geq N + j$. Atlikus minimalų vidurkinimą spindulio ir kampo kryptimis gausime 15 pav. pavaizduotą binarinių požymių matricą. Ši požymių matrica kiek panaši į 14 pav. požymių matricą tik apskaičiuotą kitu dvigubai smulkesniu masteliu.

Daugman darbe taip pat nurodyta, kad galutinė požymių matrica formuojama naudojant kelių mastelių Gaboro filtrus. Todėl mes apibendrinsime vidurkinimo formules tuo atveju, kai imama KF vienodais svoriais vidurkinimų kampo ir $(2 * KR + 1)$ trikampių lango svoriais spindulio kryptimi. Tokį vidurkinimą atitiks filtras

$$f_{n,m} = \begin{cases} 1 - \frac{n}{KR+1}, & n \leq KR, \quad m = -KF, \dots, -1, \text{ arba } KF, \dots, 2KF - 1, \\ -2(1 - \frac{n}{KR+1}), & n \leq KR, \quad m = 0, \dots, KF - 1, \\ 0, & \text{likusiais atvejais.} \end{cases}$$

Home Page

Title Page

◀

▶

◀

▶

Page 30 of 48

Go Back

Full Screen

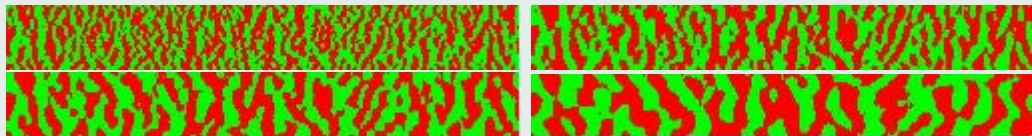
Close

Quit

15 pav. atitinka vidurkinimo parametrų reikšmės $KF = 2$ ir $KR = 1$.

Rainelės šablonas

Šablonas (angl. *template*) yra formuojamas iš požymių. Šablono pradžioje gali būti antraštė (angl. *header*), kurioje įrašoma specifinė informacija, pavyzdžiui pradinio paveiksluko katalogas ir vardas, objekto unikalūs identifikatoriai. Šablono antraštėje patalpinama informacija nenaudojama skaičiuojant dviejų šablonų panašumą. Likusi rainelės šablono dalis sudaroma iš binarinių požymių. Tarkime fiksuoto mastelio požymių matricai sudaryti naudosime $32 r_n$ ($N = 32$) ir $256 \phi_m$ ($M = 256$) reikšmes, bei keturis skirtingus mastelius/vidurkinimus atitinkančius parametrus $KF = 2$, $KR = 1$; $KF = 3$, $KR = 2$; $KF = 4$, $KR = 2$ ir $KF = 6$, $KR = 3$. 16 pav. iliustruoja 242116.tiff failo rainelės gaunamą šabloną. Vaizdumo dėlei 4-ių skirtingų mastelių binarinių požymių matricos atskirtos kelių taškelių baltais tarpais. Sudarant vieningą $4 \times 32 \times 256$ bitų matricą nebūtina išdėstyti paveikslėlyje nurodyta tvarka. Pavyzdžiui gali būti patogų vieną baitą sudaryti iš dviejų ketvertukų, gaunamų sudedant greta keturių skirtingų mastelių binarinių požymių bitus. Objektus atspindintys šablonai kaip taisyklė užima žymiai mažiau vietos nei pradinė informacija. Pavyzdžiui mūsų pradinis rainelės paveikslėlis pateikiamas $640 \times 480 \times 256$ pilkumo lygmenų taškeliais, kas atitinka $640 \times 480 \times 8 = 2457600$ bitus informacijos, o tos pačios rainelės šablonas užima tik $32 \times 256 \times 4 = 32768$ bitus, t.y. pradinė informacija suspaudžiama lygiai 75 kartus. Daugmano algoritmas naudoja dar glaudesnius šablonus,



16 paveikslėlis: Rainelės šablonas sudarytas iš 4 - ių mastelių 32×256 binarinių požymių matricių.

kurių dydis yra 2048 bitai binarinių požymių plus tokio paties dydžio kaukės (angl. *mask*), t.y. vieno rainelės šablono dydis yra tik 4096 bitai.

Šablonų palyginimas

Beliko išsiaiškinti paskutinį rainelių atpažinimo/verifikacijos etapą, t.y. šablonų palyginimą. Verifikacijos atveju atpažinimui pateikiami du rainelių šablonai ir reikia priimti sprendimą ar tie šablonai atitinka vieną ir tą pačią rainelę ar tai yra šablonai gauti iš skirtingų rainelių. Kaip taisyklė pirmasis šablonas yra išsirenkamas iš anksto įrašytos rainelių duomenų bazės pagal asmens prisistatymą, pvz. klaviatūroje surinktą vardą/pavardę ar kokį nors asmenį identifikuojantį kodą, -o antrasis šablonas sudaromas operatyviai nuskenavus asmens rainelę ir atlikus pilną rainelės lokalizavimo bei požymių išskyrimo procedūrą. Taigi verifikacijos metu reikia įvertinti dviejų šablonų panašumą. Šablonų panašumas įvertinamas lyginant atskirus šablonų bitus. Jei dviejų šablonų atitinkamose pozicijose bitai sutampa (t.y. $0 = 0$ arba $1 = 1$), tai panašumo metrika padidinama vienetu. Paprastumo dėlei laikysime, kad šablone yra įrašyta $K \times N \times M = L$ bitų bin_l , $l = 0, 1, \dots, L - 1$, kur K yra mastelių

skaičius, N skirtingų spindulių skaičius, M kampų skaičius¹. Tuomet dviejų šablonų panašumas (angl. *similarity*) gali būti įvertintas taip

$$\text{sim}(bin^1, bin^2) = \frac{\sum_{l=0}^{L-1} \text{sim}[bin_l^1, bin_l^2]}{L},$$

kur $\text{sim}[a, b] = 1$, kai $a = b$ ir $\text{sim}[a, b] = 0$, kai $a \neq b$. Kitaip tariant panašumas yra dviejuose šablonuose sutapusių bitų dalis. Deja tokia paprasta panašumo samprata duoda nelabai aukštą rainelių verifikacijos kokybę. Taip atsitinka dėl to, kad dvi skirtingos tos pačios rainelės nuotraukos gali būti kiek pasisukusios kampu ϕ_0 ir prielaida, kad abiejų rainelių kampų reikšmės vieningoje koordinatinių sistemoje sutampa, t.y. $\phi_m^1 = \phi_m^2$, yra neteisinga ir turėtų būti patikslinta pvz. $\phi_m^1 = \phi_m^2 + \phi_0$ formule. Tačiau praktiškai posūkio ϕ_0 reikšmė būna nežinoma, tuo labiau, kad bendru atveju dėl rainelių nestandžių deformacijų ir lokalizacijos paklaidų kiekvienam m gali būti kiek kita posūkio ϕ_0 reikšmė. Todėl praktiškai yra fiksuojamas ieškomo posūkio ϕ_0 tikslumas $\delta(Rot)$, pasirenkamas galimų posūkių skaičius $NRot$ (pvz. Daugman naudoja $NRot = 3$) ir ieškomas maksimalus panašumas

$$\text{Sim}(bin^1, bin^2) = \max_{rot \in \{-NRot, \dots, NRot\}} \text{sim}(bin^1, Rot(bin^2, rot)) = \frac{\sum_{l=0}^{L-1} \text{sim}[bin_l^1, Rot(bin_l^2, rot^*)]}{L}.$$

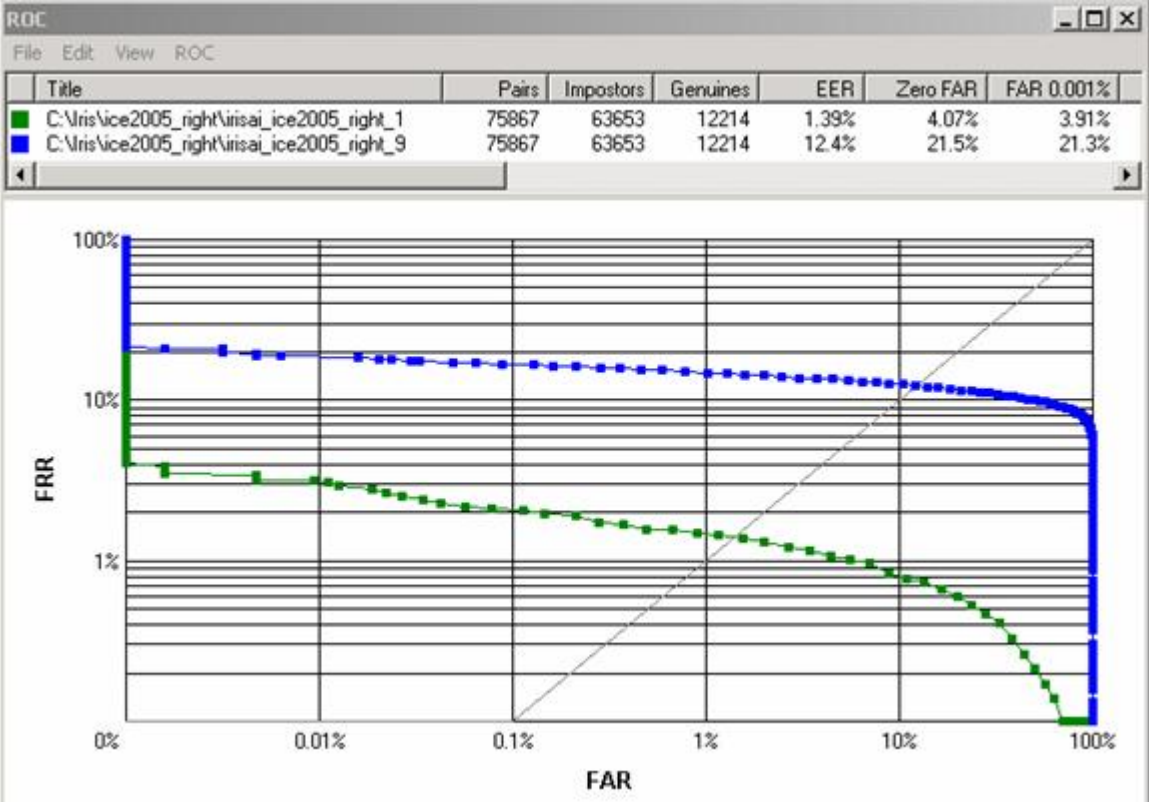
Kaip taisyklė $Rot(bin_l, rot) = bin_{l+rot}$, tačiau jei $(l+rot \geq L)$ arba l ir $l+rot$ yra skirtingų spindulių eilutėse, indeksas $l + rot$ paslenkamas kartotiniu eilutės periodu M taip, kad l ir $l + rot + mM$ būtų

¹ Terminijos paprastumo dėlei laikome, kad rainelės išorinio ir vidinio apskritimų centrai yra vienodi ir tuomet N atitinka naudojamų fiksuotų spindulių skaičių, o M - pilno kampo 2π padalinimų skaičių

vienoje eilutėje. Pavyzdžiui, jei $M = 256$ ir formuojant šabloną naudojama 16 pav. iliustruojama skirtingų mastelių išdėtymo tvarka, tai

$$Rot(bin_{254}^2, 7) = bin_5^2.$$

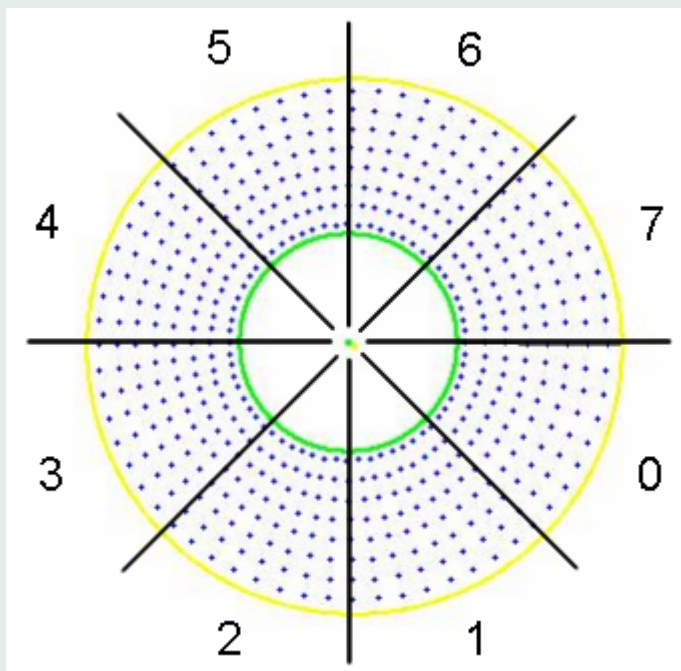
Kad palyginti skirtingų panašumo metrikų kokybę ir parinkti optimalius požymių išskyrimo ir palyginimo algoritmų parametrus naudosime JAV nacionalinio standartizacijos instituto ICE 2005 rainelių duomenų bazę [37]. Rainelės lokalizavimo mes neoptimizuosime, taip pat fiksuosime 4 požymių mastelius, spindulių skaičių $N = 32$ ir kampų $M = 256$. 17 pav. iliustruoja dviejų skirtingų rainelių palyginimo algoritmų kokybę. Mėlyna spalva pažymėta verifikacijos kokybės kreivė DET [43] gaunama tiesiogiai lyginant du šablonus, t.y. atveju $NRot = 0$, o kita, žaliai nuspelvinta kreivė, gauta atveju $NRot = 4$, kas atitinka $1 + 2 * 4 = 9$ skirtingus rainelės poslinkius (postūmius šablonų atveju). Kadangi žalia DET kreivė visuose taškuose žemiau mėlynos DET kreivės, tai vienareikšmiai galime teigti, kad šių duomenų bazei antrasis šablonų palyginimo variantas tinka geriau. Atkreipkite dėmesį, kad skaičiuojant DET kreivę panašumo įverčiai buvo skaičiuojami ne visose galimose apsisaukėlių porose. Taip daroma gana dažnai, nes didelių duomenų bazių atveju galima sudaryti labai daug apsisaukėlių porų ir taupant skaičiavimo laiką galima skaičiuoti panašumo įverčius ne visoms, o pavyzdžiui tik kas 16 ar 64 apsisaukėlių porai. Autentiškų porų palyginus būna mažiau, todėl nesunku joms visoms apskaičiuoti panašumo įverčius.



17 paveikslėlis: NIST ICE 2005 Right Iris duomenų DET kreivės gaunamos naudojant nesukiojamus šablonus (mėlyna kreivė) ir sukiojamus 9 pozicijose (žalia kreivė)

Pasiūlysime dar keletą šablonų palyginimo modifikacijų, kurios gerina verifikacijos kokybę. Lyginamų šablonų poslinkius galima atlikti ne tik kampo kryptimi bet ir spindulio kryptimi. Idealaus rainelių lokalizacijos atveju tokie poslinkiai neturi prasmės, tačiau nedideles vyzdį ir rainelę ribojančių apskritimų spindulių suradimo paklaidas galima ištaisyti šablonų palyginimo metu slenkant šablonus keliais taškeliais spindulio kryptimi. Turint K mastelių ir M vieno mastelio skirtingų kampų atliekant vieno taškelio poslinkį spindulio kryptimi reikia bin_l reikšmę pakeisti bin_{l+K*M} reikšme. Gana kebli problema yra lyginti nestandžiai deformuotas raineles. Nedaug pakėlus/nuleidus, pasukus/pakreipus galvą tos pačios akies skenuotas rainelės vaizdas išsitampo ir dviejų apskritimų modelis nepilnai sugeba kompensuoti atsiradusius iškraipymus. Šiai problemai spręsti siūlome suskaidyti rainelės gardelės taškus atskirais sektoriais. 18 pav. pavaizduotas gardelės taškų grupavimas 8 sektoriais. Laikysime, kad vieno sektoriaus ribose deformacijos yra pakankamai mažos ir lygindami dviejų rainelių požymius ieškosime šablonų optimalių poslinkių kampo ir spindulio kryptimis.

Tarkime $KRot$ yra teigiamų poslinkių skaičius kampo kryptimi, o $KShift$ teigiamų poslinkių skaičius spindulio kryptimi. Atliekant poslinkį spindulio kryptimi susiduriame su krašto problema, nes šiuo atveju periodinis reikšmių pratęsimas būtų nelogiškas. Sprendžiant šią problemą galima apsiriboti mažesniu lyginamų taškelių skaičiumi, pavyzdžiui $KMx(N - 2KShift)$ parenkant juos taip, kad visi teigiami ir neigiami poslinkiai tilptų $KMxN$ binarinių požymių matricoje. Kitas sprendimas



18 paveikslėlis: Rainelės gardelės taškų suskaidymas į 8 sektorius.

taškeliams, kurių indeksai išeina iš $KM \times N$ matricos ribų priskirti su tikimybe 0.5 reikšmę 0 arba 1 ir lyginti pilnas $KM \times N$ taškių matricas. Kadangi pagal filtravimo ženklą parenkamų binarinių požymių reikšmės gana tiksliai su vienoda tikimybe 0.5 yra lygios 0 arba 1, tai toks santykinai nedidelio iš ribų išeinančių taškių pakeitimas atsitiktiniais 0 arba 1 nežymiai keičia palyginimo rezultatą ir

sprendžia gana sudėtingą problemą kokią atlikti normavimą lyginant skirtingų dydžių binarinių požymių matricas.

Tarkime suskaidėme gardelę į S sektorių. Tokiu atveju maksimizuodami atžvilgiu $KShift$ ir $KRot$ poslinkių kiekvieno lyginamo sektoriaus binarinių požymių sutapimo procentą turėsime

$$((2KShift + 1)(2KRot + 1))^S$$

laisvės laipsnių. Didėjant S šis skaičius greitai auga ir todėl apsišaukėlių poros gali atsitiktinai įgyti dideles panašumo reikšmes. Šiai problemai spręsti siūlome tokią dinaminio programavimu paremtą procedūrą:

```
int[][]score = new int[−KShift : KShift][−KRot : KRot]
```

```
for sector := 0 to S − 1 step 1 do
```

```
    for k := −KShift to KShift step 1 do
```

```
        for r := −KRot to KRot step 1 do
```

```
            score[k][r] += sim(shift(bin1, k), rot(bin2, r))sector +
```

```
            max(score[k][r], score[k − 1][r], score[k + 1][r], score[k][r − 1], score[k][r + 1])
```

```
        od
```

```
    od
```

od

od

Realizuojant $\max(score \dots)$ skaičiavimą reikia pasirūpinti, kad indeksų ribos neišeitų iš rėžių. Dviejų šablonų panašumo reikšmė įvertinama dvimačio $score$ masyvo maksimumu. Taip skaičiuojamas panašumas mažų autentiškų rainelių deformacijų atveju surinks didelį panašumo taškų skaičių, o apsišaukėlių atveju panašumo reikšmė turi mažiau šansų surinkti daug panašumo taškų, nes dabar laisvės laipsnių skaičius mažesnis nei

$$(2KShift + 1)(2KRot + 1)5^{S-1}.$$

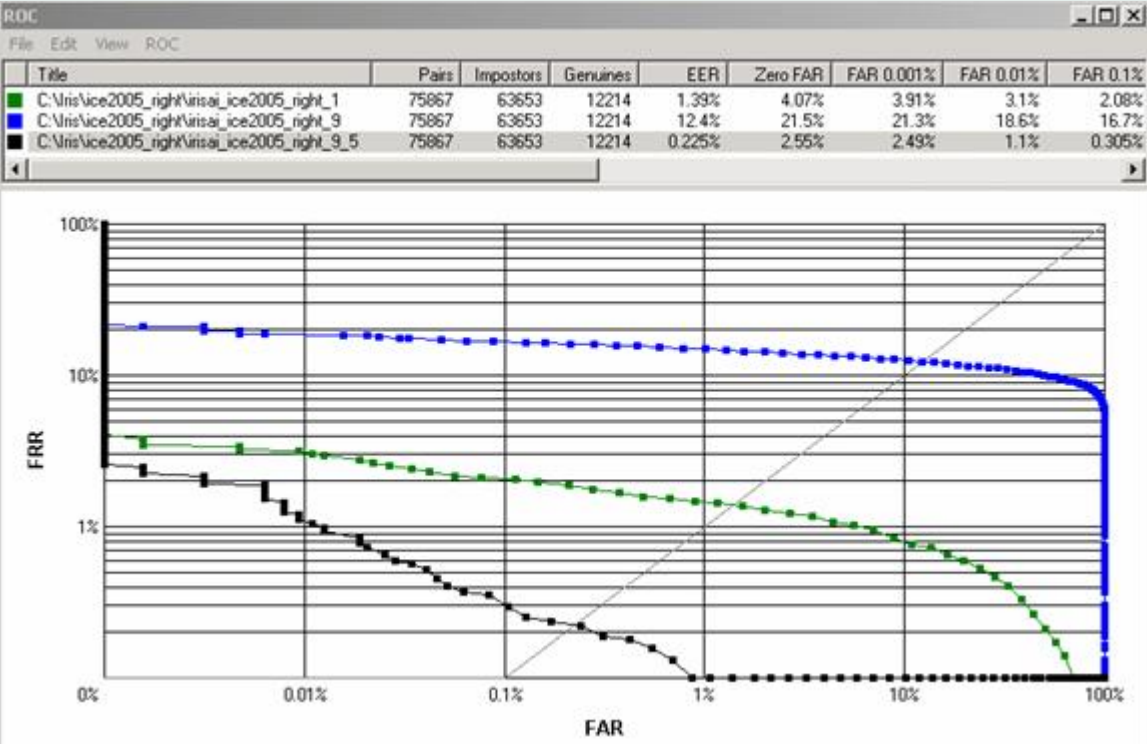
19 pav. iliustruoja verifikacijos DET kreivės pagerėjimą gautą aprašytu algoritmu, kurio parametrai yra $S = 16$, $N = 32$, $M = 256$, $KShift = 2$, $KRot = 4$.

Praktinė užduotis

Realizuokite aprašytą rainelių požymių apskaičiavimo, šablonų sukūrimo ir jų palyginimo algoritmą.

Rainelių atpažinimo užduotis pasirenkama laisvai, kai atliktos privalomos užduotys. Darbas atliekamas auditorijoje pratybų metu arba savarankiškai namuose atsiskaitant auditorijoje pratyboms skirtu laiku už atskirus etapus.

Rainelių vaizdų pavyzdžius rasite <http://www.mif.vu.lt/atpazinimas/iris/ice2005/Spring2004iris/>



19 paveikslėlis: 17 pav. DET kreivės papildytos juoda verifikacijos DET kreive, kuri gauta palyginimo algoritmu naudojančiu 16 sektorių, 9 poslinkius kampo kryptimi, 5 poslinkius sindulio kryptimi ir dinaminio programavimo panašumo taškų skaičiavimo algoritmą.

kataloge. Rainelių segmentacijos informacija patalpinta [Ice2005_right.info](#) ir [Ice2005_left.info](#) failuose.

1. Apskaičiuokite ir pavaizduokite ištiesintos rainelės pilkumo lygmenis analogiškai **13** paveikslėliui.

Raskite ir pavaizduokite rainelės binarinių požymių šabloną analogiškai **16** paveikslėliui.

Vertinama iki **1.25** balo.

2. Realizuokite rainelių šablonų palyginimo variantą, kuris naudoja rainelių posūkius. Palyginimo rezultatus įrašykite "*.roc" formato faile (žiūr. [\[43\]](#)). NIST ICE 2005 right rainelių šablonų bazės verifikacijos lygios klaidos tikimybė (EER) turi būti mažesnė už 5 %.

Vertinama iki **1.25** balo.

3. Rainelių verifikacijos konkursas. Paruoškite rainelių verifikacijos algoritmą, kuris apskaičiuotų duotų rainelių ir jų segmentacijos duomenų (raineles.info) raineles.roc failą. 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 nurodomas vienas parametras: [raineles.list](#).

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

Home Page

Title Page

◀

▶

◀

▶

Page 42 of 48

Go Back

Full Screen

Close

Quit

java -Xmx1024m -jar ManoRainele.jar raineles.info

- Jūsų programa turi kiekvieną fiksuotą raineles.info sąrašo elementą palyginti su raineles.info sąrašo elementais įrašytais didesnio numerio eilutėse. Gautus rezultatus įrašyti į *.roc formato failą. **raineles.info** pavyzdyje yra 1425 įrašų. Vadinasi paskaičiuoto pavyzdžio *.roc faile turės būti $1425 * 1424 / 2$ palyginimų įrašų.

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

Skaiciavimai bus atliekami su Intel Core 2 Duo, 2.1 GHz, 3 Gb Ram PC. Programa turi palyginti bent 100 rainelių porų per sekundę.

- 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.

Home Page

Title Page

◀

▶

◀

▶

Page 43 of 48

Go Back

Full Screen

Close

Quit

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://videolectures.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 Zabarauskio 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. Bitiesinė 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.](#)

[Home Page](#)[Title Page](#)

◀

▶

◀

▶

Page 46 of 48

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

- [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 Computational 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.
- [29] J. Daugman, High confidence visual recognition of persons by a test of statistical independence. IEEE Transactions on Pattern Analysis and Machine Intelligence

ce, vol. 15(11), pp. 1148-1161, 1993.// Skenuotą straipsnio kopiją autorius patalpino <http://www.cl.cam.ac.uk/jgd1000/PAMI93.pdf> adresu.

[30] Iridian Technologies kompanijos tinklalapis <http://www.iridiantech.com>

[31] Iris recognition, http://en.wikipedia.org/wiki/Iris_recognition, 2010.

[32] John Daugman, [Results from 200 billion iris cross-comparisons](#)

[33] International Civil Aviation Organization, <http://www.icao.int>

[34] Amsterdam Schiphol Airport, http://en.wikipedia.org/wiki/Schiphol_Airport

[35] United Arab Emirates, http://en.wikipedia.org/wiki/United_Arab_Emirates

[36] IRIS - Iris recognition immigration system, <http://www.ind.homeoffice.gov.uk/applying/iris/>

[37] Iris challenge evaluation, <http://iris.nist.gov/ice/>

[38] Interlacing, <http://neuron2.net/LVG/interlacing.html>

[39] [Feature detection using Integral Image](#)

[40] Bilinear Interpolation, http://en.wikipedia.org/wiki/Bilinear_interpolation

Home Page

Title Page



Page 48 of 48

Go Back

Full Screen

Close

Quit

[41] Circular Hough Transform, <http://ipml.ee.duth.gr/papamark/GHT/Help/CHT.htm>

[42] Gabor filter, http://en.wikipedia.org/wiki/Gabor_filter

[43] Atpažinimo sistemos kokybės įvertinimas ir jo vizualizacija (DET arba ROC kreivė),
<http://uosis.mif.vu.lt/bastys/academic/ATE/ROC/ROC.htm>