

1. Parašų atpažinimas

- Parašo binarizacija
- Parašo išlyginimas
- Požymių išskyrimas
- Požymių palyginimas
- Nuorodos
- Praktinis darbas

1.1. Parašo binarizacija

1 paveikslėlis iliustruoja skenuotą parašą "0_0". [ltTEParasuDB1](#) adresu patalpintas [ltTEParasu-DB1.zip](#) archyvas, kuriame yra 16-os asmenų 113-a skenuotų parašų pavyzdžių. Skaičius prieš brūkšnį failo varde žymi asmens ID, po brūkšnio - parašo sesijos numeris.

Patalpinti archyve parašų paveikslėliai yra apdoroti taip, kad gryna mėlyna spalva (spalvos kodas 0xff0000ff) žymi parašo linijos taškus, o likusių spalvų taškai yra fono taškai. Trumpumo dėlei kartais vietoje 0xff0000ff rašysime 1. 2 paveikslėlis iliustruoja parašo "0_0" vaizdą, kuriam atliktas

Home Page

Title Page



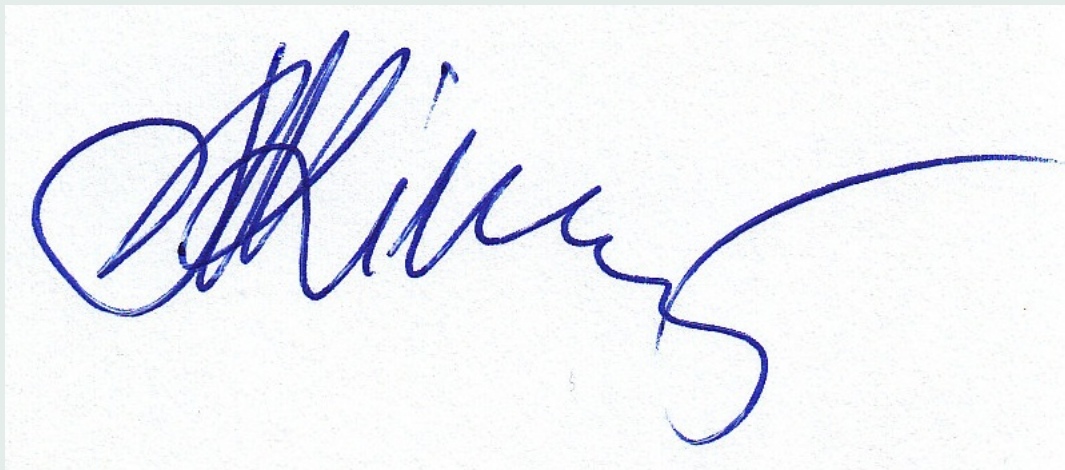
Page 2 of 26

Go Back

Full Screen

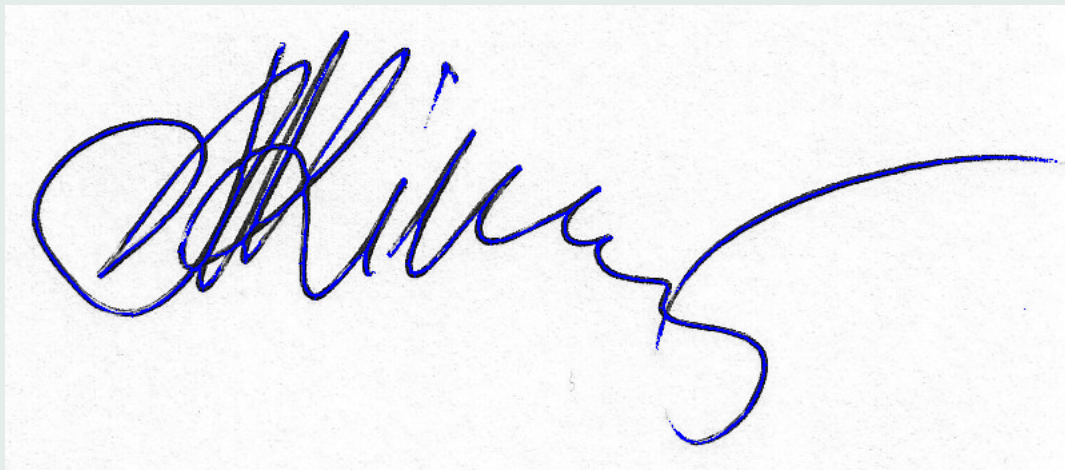
Close

Quit



1 pav.: Originalus skenuoto parašo vaizdas, Lietuvos teismo ekspertizės DB1 0_0 parašas, 803x350 taškelių

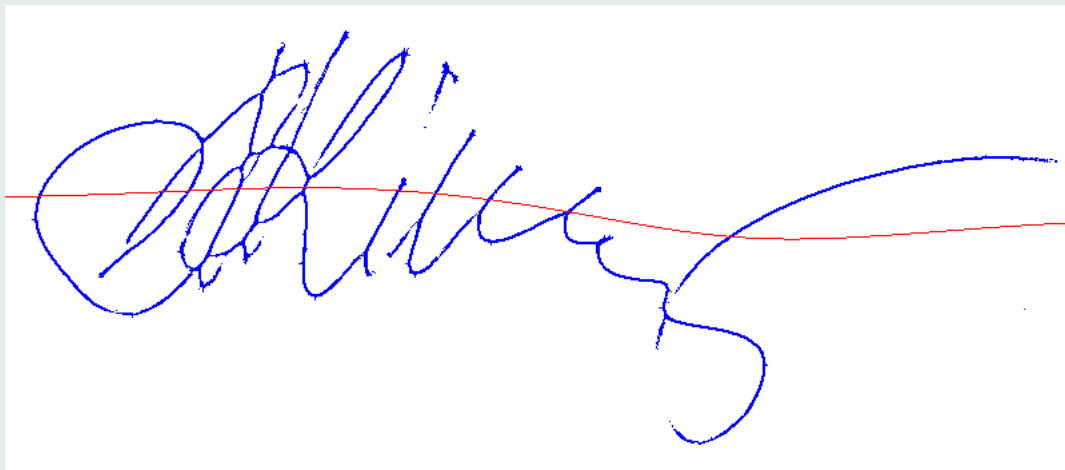
pirminis apdorojimas. Praktikos užduotims bus pateikiami parašų vaizdai, kuriems yra atliktas pirminis vaizdų apdorojimas, kurio rezultate parašo linijos taškams priskiriamas spalvos kodas 0xff0000ff. Todėl parašas binarizuojamas paliekant grynos mėlynos spalvos taškelius ir priskiriant baltos spalvos kodą (0xffffffff=-1) likusiems taškeliams. 3 paveikslėlis iliustruoja parašo "0_0" binarizuotą vaizdą.



2 pav.: Apdoroto parašo vaizdas, grynų mėlynos spalvos kodas 0xff0000ff žymi parašo linijų taškelius.

1.2. Parašo išlyginimas

To paties asmens parašas gali būti skirtingai nukreiptas pasirašymo metu. Papildomi dominuojančios parašo krypties iškraipymai atsiranda skenavimo metu. Todėl rekomenduoja, prieš juos lyginant, išlyginti parašus taip, kad dominuotų horizontali pasirašymo kryptis. Kad tai pasiekti, galima įvertinti vidutinę parašo taškelių ordinatę fiksuotai horizontaliai (x) pozicijai. Tuo tikslu siūloma įsivesti darbinį $u0v[x]$, $u1v[x]$, $x = 0, 1, \dots, \text{width} - 1$ masyvus, kur width yra parašo paveikslėlio plotis. Masyve $u0v[x]$ suskaičiuojama kiek fiksuotam x yra paveikslėlio $u[x][y]$ reikšmių, kuriose spalvos kodas



3 pav.: Binarizuotas parašo vaizdas. Raudona kreivė žymi adaptyvų parašo linijų taškelių vidurkį vertikalia kryptimi.

lygus parašo linijos kodui (grynai mėlynai spalvai). Analogiškai, masyve $u1v[x]$ suskaičiuojama kokia fiksuotam x yra paveikslėlio $u[x][y]$ parašo linijų taškų ordinačių suma. Tokiu būdu gauname formules

$$u0v[x] = \sum_{(x,y) \in \Omega} 1$$

ir

$$u1v[x] = \sum_{(x,y) \in \Omega} y,$$

kur

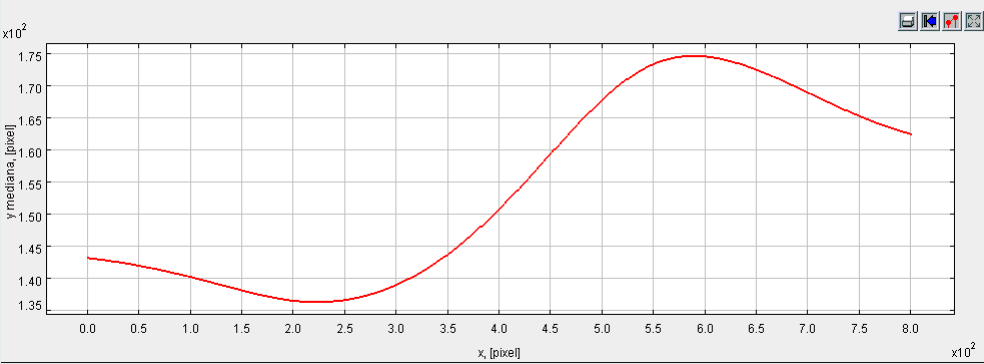
$$\Omega = \{(x, y) : u[x][y] = 0x\text{ff}0000\text{ff}\}.$$

Raidė v panaudota, pažymėti, kad vidurkinimas atliekamas vertikalia kryptimi. Kadagi $u0v$ ir $u1v$ kinta diskrečiai, sumų reikšmės atžvilgiu x gali kisti gana nereguliariai. Kad tai kompensuoti ir išvengti dalybos iš nulio problemos apskaičiuojant vidutines ordinate rekomenduojama pritaikyti masyvams $u0v[x]$ ir $u1v[x]$ eksponentinį filtravimą, kuris aprašytas <http://uosis.mif.vu.lt/bastys/academic/A-TE/biometrika/Vidur.pdf>. Eksponentinio vidurkinimo java kodas patalpintas [čia](#), rekomenduotina naudoti vidurkinimo metodą *vidur* su eksponentinio filtro pločio reikšmė $\sigma = 100$. Vidurkinimo rezultatai bus patalpinti GlodusExpo objekto *re* masyve. Vidurkintus su filtro parametru σ masyvus žymėsime $u0v_\sigma[x]$ ir $u1v_\sigma[x]$ simboliais.

Ieškomas paveikslėlio stulpelio x poslinkis randamas apskaičiuojant vidutinę parašo vidurkintą ordinatę, t.y.

$$\text{shift}V_\sigma[x] = \frac{u1v_\sigma[x]}{u0v_\sigma[x]}.$$

4 paveikslėlis iliustruoja vertikalų poslinkių $\text{shift}V_\sigma[x]$ reikšmes atidėtas tiesiogiai parašo paveikslėlyje. Jei parašo paveikslėlis būtų triviali horizontali linija, tai vidurkinta raudona linija sutaptų su horizontalia linija. 5 iliustruoja $\text{shift}V_{100}[x]$ kreivę gautą 1 paveikslėliui. Toliau žymėjimų paprastumo dėlei dažnai nenaudosime parametro σ , t.y. sutrumpintai rašysime $u1v_\sigma = u1v$, $\text{shift}V_\sigma = \text{shift}V$ ir



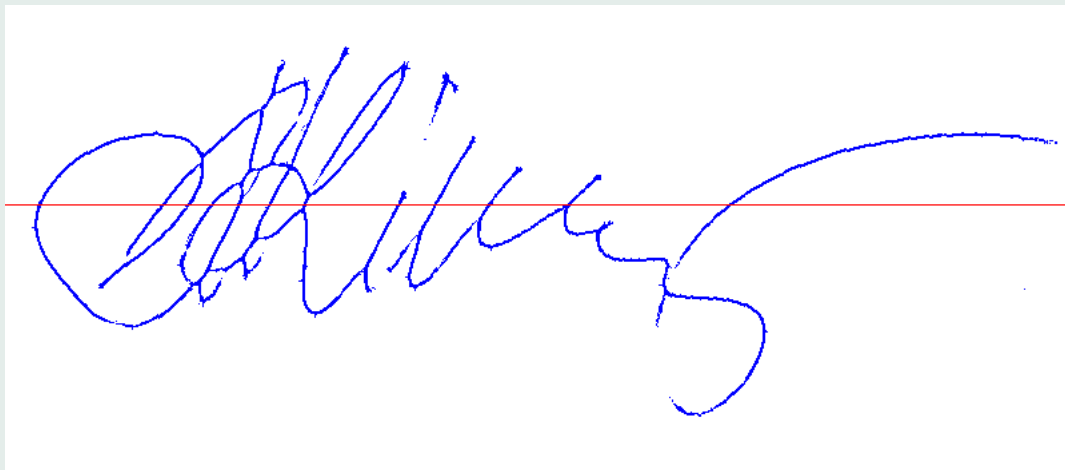
4 pav.: Adaptyvus poslinkis y kryptimi ($\text{shiftV}_\sigma[x]$) reikšmės

t.t. Turint $\text{shiftV}[x]$ reikšmes nesunku išlyginti parašo paveikslėlį atliekant paveikslėlio poslinkius y (vertikalia) kryptimi, imant poslinkių reikšmes proporcingas vidutiniai vidurkintai parašo ordinatei. Kad po poslinkio nepakistų vidutinė parašo pozicija, rekomenduojama surasti $\text{shiftV}[x]$ vidurkį $\overline{\text{shiftV}}$, kuris apskaičiuojamas formule

$$\overline{\text{shiftV}} = \frac{\sum_{x=0}^{\text{width}-1} \text{shiftV}[x]}{\text{width}}.$$

Turint konstantos $\overline{\text{shiftV}}$ reikšmę, išlygintas vertikalia (y) kryptimi vaizdas gaunamas taip:

$$u'[x][y] = u[x][y + \text{shift}[x]],$$



5 pav.: Išlygintas vaizdas y (vertikalia) kryptimi

kur vaizdo stulpelio x poslinkis $\text{shift}[x]$ apskaičiuojamas formule

$$\text{shift}[x] = \text{round}(\text{shiftV}[x] - \overline{\text{shiftV}}).$$

Atliekant vaizdo stulpelių poslinkį rekomenduojama nekeisti pradinio parašo vaizdo matmenis periodiškai perkeltant išeinančius iš vaizdelio ribų taškelius. 5 paveikslėlis iliustruoja gautą parašo vaizdą išlyginus paslenkant jį vertikalia kryptimi. Lygiagreti horizontui raudona kreivė gaunasi dėl to, kad pagal stulpelių poslinkių formules gauname lygybę

$$u'[x][\overline{\text{shiftV}}] = u[x][\text{shiftV}[x]].$$



6 pav.: Krypties iliustracija

7 ir 8 paveikslėliai iliustruoja tolimesnį galimą vaizdo išlyginimą horizontalia kryptimi. Į šio išlyginimo detales nesigilinsime, nes jis dažnai parašų palyginimą pablogina, nes skirtingų asmenų parašai per daug suvienodėje, nes skirtinguose parašuose atsiranda daug panašaus statmenumo tiesės atkarpų.

Home Page

Title Page



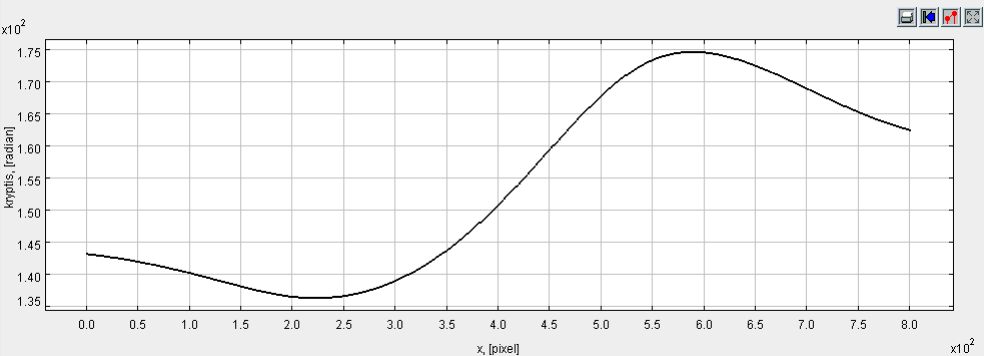
Page 9 of 26

Go Back

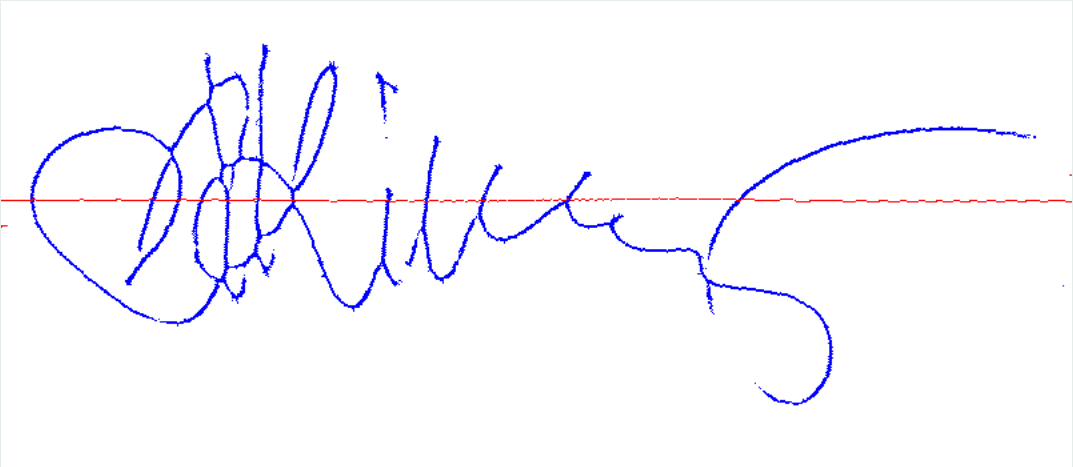
Full Screen

Close

Quit



7 pav.: Adaptyvus krypties vidurkis



8 pav.: Išlygintas vaizdas vertikalia (y) ir horizontalia (x) kryptimis

1.3. Požymių išskyrimas

Atlikus parašo išlyginimą, galima pereiti prie požymių skaičiavimų. Biometrikoje ir kitose automatinio objektų atpažinimo srityse atpažinimui, kaip taisyklė, pateikiama ne pradinė informacija (pvz. veidų, parašų skaitmeninės nuotraukos, balso garso įrašo failas), o objektų požymiai, kurie apskaičiuojami iš pradinių duomenų ir labai koncentruotai atspindi esmines objekto savybes. Dalinai išskyrus pirštų atspaudus, kitose biometrikos rūšyse intuityviai nėra aišku kokius požymius geriausiai naudoti objekto arašymui, kad jie derintų dvi vienas kitai priešingas savybes:

1. požymių būtų kuo mažesnis skaičius ir kad jie būtų patogūs jų greitam palyginimui;
2. požymiai kuo detaliau atspindėtų pradinę atpažinimui pateikiamą objekto reprezentaciją, idealiu atveju iš požymių turėtų būti galima gana tiksliai rekonstruoti pradinį objektą.

Šia prasme objektų požymių išskyrimas ir jų gebėjimas apytiksliai aprašyti pradinę informaciją pri-
mena nuostolingąją duomenų kompresiją, kur siekiama kuo mažesniu bitų skaičiumi apytiksliai at-
vaizduoti pradinį objektą.

Parašų atveju skiriamos dinaminių ir statinių parašų verifikavimo sistemos. Dinaminių parašų at-
veju naudojami specialūs dinamiškai sekantys parašo kontūrus įrenginiai. Statiniais parašais laikomi
skenuoti rastiniai parašų vaizdai. Statinio parašo atveju turima mažiau informacijos apie pasirašy-

mo linijų formavimo dinamiką. Todėl šiuo parašus sunkiau verifikuoti, t.y. kompiuterio algoritmu patvirtinti ar tvirtinimas apie asmens tapatybę pateiktas verifikavimui parašas atitinka jo parašų pavyzdžius, dėl kurių autentiškumo nedvejojama. Tačiau rastinių parašų verifikavimas turi privalumą, kad jo duomenų įskaitmeninimui tinka standartinė skenerio įranga, kurios dažnei nereikia specialiai pirkti, nes skeneriai turi ir daugybę kitų taikymų, todėl jie dažnai būna įsigijami kompiuterinės technikos komplektuose. Kadangi mūsų turima parašų duomenų bazė sudaryta iš rastinių vaizdų, mes nagrinėsime statinių parašų palyginimo algoritmus. Įsisavintą techniką bus galima pritaikyti palyginimui kompiuteriu bet kokių objektų, kurių rastiniuose vaizduose dominuoja plonos kreivės. Mes remsimės [[1]] darbu, kuriame pateikta statinių parašų atpažinimo algoritmų apžvalga. Įvairių parašų atpažinimo metodų lygios klaidos tikimybė svyruoja nuo 0.3 iki 0.02. Deja nėra nusistovėjusios viešai prieinamos parašų duomenų bazės, todėl algoritmai dažnai įvertinami su autorių turimomis vidinėmis duomenų bazėmis ir todėl jų rezultatus sunku sulyginti. Kartu reikia pažymėti, kad pagal darbe [[1]] surinktus duomenis dinaminių parašų verifikavimo kokybė neženkiai lenkia statinių parašų atpažinimo kokybę matuojamą lygios klaidos tikimybe. Lygios klaidos tikimybė yra gaunama pritaikius konkretų algoritmą konkrečiai parašų duomenų bazei ir algoritmu gautiems parašų porų panašumams surandamas toks sprendimo priėmimo slenkstis, kad dalis klaidingai pripažintų autentiškais ir dalis klaidingai atmestų autentiškų parašų būtų vienoda. Dažnai, kad pagerinti verifikacijos kokybę, vienas

tiriamasis parašas lyginamas su verikuojamą asmens parašų aibe, bet mes apsiribosime "grynu" atveju, kai priimti sprendimą apie parašų autentiškumą reikia remiantis tik viena lyginamų parašų pora.

Viena parašų lyginimo metodika [[1]] siūlo požymiais naudoti parašo taškų projekcija pagal kokią nors pasirinktą liniją. Mes apibendrinsime šią požymių išskyrimo idėją, įvesdami projektavimo mastelį s ir tris projektavimo tiesių normalių kryptis: $\alpha_k, k = 0, 1, 2$. Fiksuokime skenuoto išlyginto parašo taškelį $[i, j]$, kuriame $u'[i][j] = 0x f f 0 0 0 0 f f$. Tuomet šį parašo tašką projektuosime masteliu $s = 1/\sqrt{2}$ trimis normalių kryptimis

$$\alpha_0 = -\frac{\pi}{12}, \quad \alpha_1 = \frac{\pi}{12}, \quad \alpha_2 = \frac{\pi}{4},$$

pagal formulę

$$\text{proj}_k(i, j) = s(i + (j - \overline{shiftV}) \tan \alpha_k), \quad k = 0, 1, 2.$$

Perbėgę visus parašo taškelius, sudarome tris skaitiklius

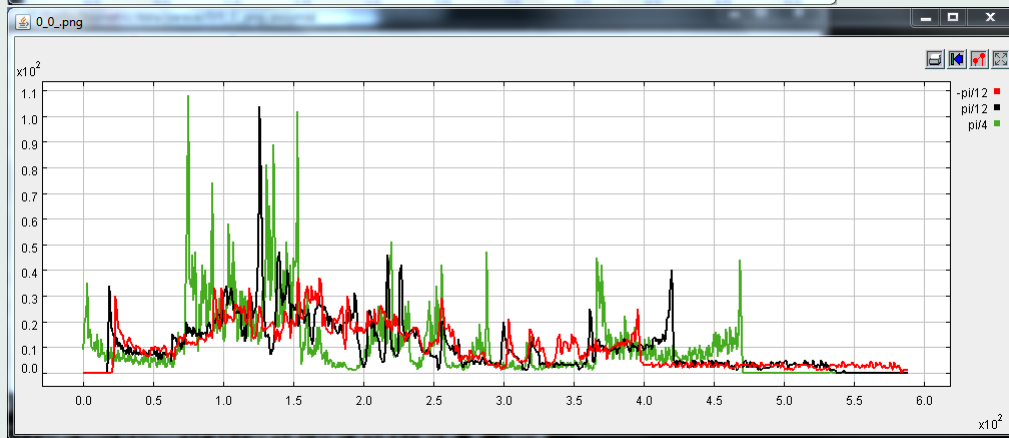
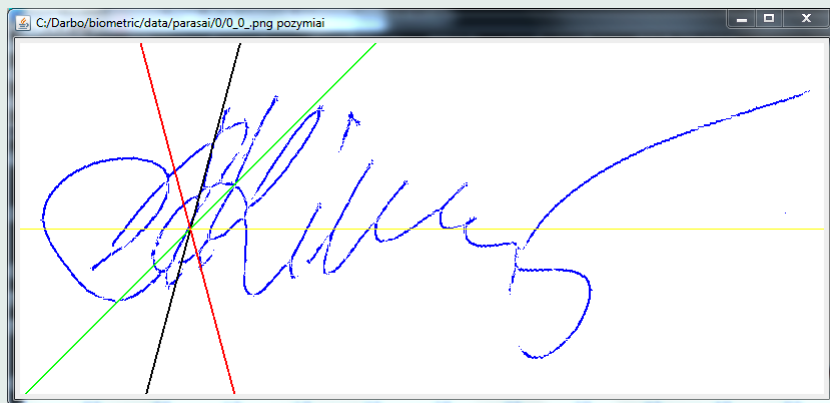
$$\#u_k[x] = \|\Omega_k[x]\|,$$

kur

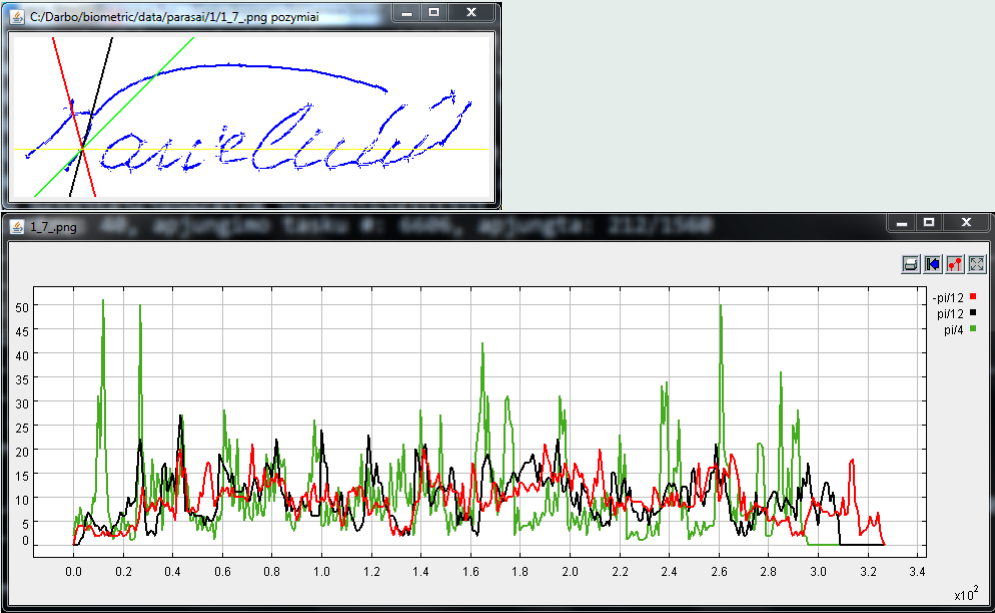
$$\Omega_k[x] = \{(i, j) : u'[i][j] = 0x f f 0 0 0 0 f f \text{ ir } \text{round}(\text{proj}_k(i, j)) = x\}$$

ir $\|\Omega\|$ žymi aibės Ω elementų kiekį.

9 pav. ir 9 pav. iliustruoja skaitiklių reikšmes dviem parašų paveikslėliams.



9 pav.: 0_0 parašų požymiai. Raudona kreivė žymi $\#u_0[x]$, juoda - $\#u_1[x]$, ir žalia - $\#u_2[x]$ skaitiklius. Viršutiniame paveikslėlyje pavaizduotos trys linijos, kurių taškai projektuojasi į argumentą $x^* = \arg \max_x \#u_1$, kuriame juodos kreivės skaitiklio reikšmė $\#u_1^*(x)$ yra maksimali.



10 pav.: 1_7 parašų požymiai. Raudona kreivė žymi skaitiklį $\#u_0[x]$, juoda - $\#u_1[x]$, ir žalia - $\#u_2[x]$. Viršutiniame paveikslėlyje pavaizduotos trys linijos, kurių taškai projektuojasi į tą patį argumentą $x^* = \arg \max_x \#u_1$, kuriame parašo 1_7 juodos kreivės skaitiklio reikšmė $\#u_1(x^*)$ yra maksimali.

1.4. Požymių palyginimas

1.5. Požymių palyginimas

11 ir 12 pav. iliustruoja keturių parašų požymius. Vizualiai galima įvertinti, kad 11 pav. dviejų parašų požymiai yra tarpusavyje panašesni, nei lyginant juos su 12 pav. pateiktais požymiais. To ir reikėtų tikėtis, nes 11 ir 12 atskirai pav. viduje iliustruoja autentiškų parašų požymius, tačiau skirtinguose pav. yra pavyzdžiai skirtingų asmenų parašų požymių. Mūsų tikslas sukonstruoti parašų panašumo metriką, kuri paverčia šias intuicijas konkrečių skaičių kalba.

Tegul $\mathbf{u} = \{(\#u_0[x], \#u_1[x], \#u_2[x])\}_{x=0}^X$ ir $\mathbf{v} = \{(\#v_0[y], \#v_1[y], \#v_2[y])\}_{y=0}^X$ yra dviejų parašų požymiai. Konstruosime panašumo metriką $\rho(\mathbf{u}, \mathbf{v})$, kuri pasižymėtų tokiomis savybėmis:

$$\rho(\mathbf{u}, \mathbf{v}) = \rho(\mathbf{v}, \mathbf{u}); \tag{1}$$

$$0 \leq \rho(\mathbf{u}, \mathbf{v}) \leq 1; \tag{2}$$

$$\rho(\mathbf{u}, \mathbf{v}) = 0, \tag{3}$$

jei $\mathbf{u}$ arba $\mathbf{v}$ parašas yra tuščias (t.y. visuose taškuose x visi požymių skaitliukai $\#u_k[x]$ arba $\#v_k[y]$

Home Page

Title Page

◀ ▶

◀ ▶

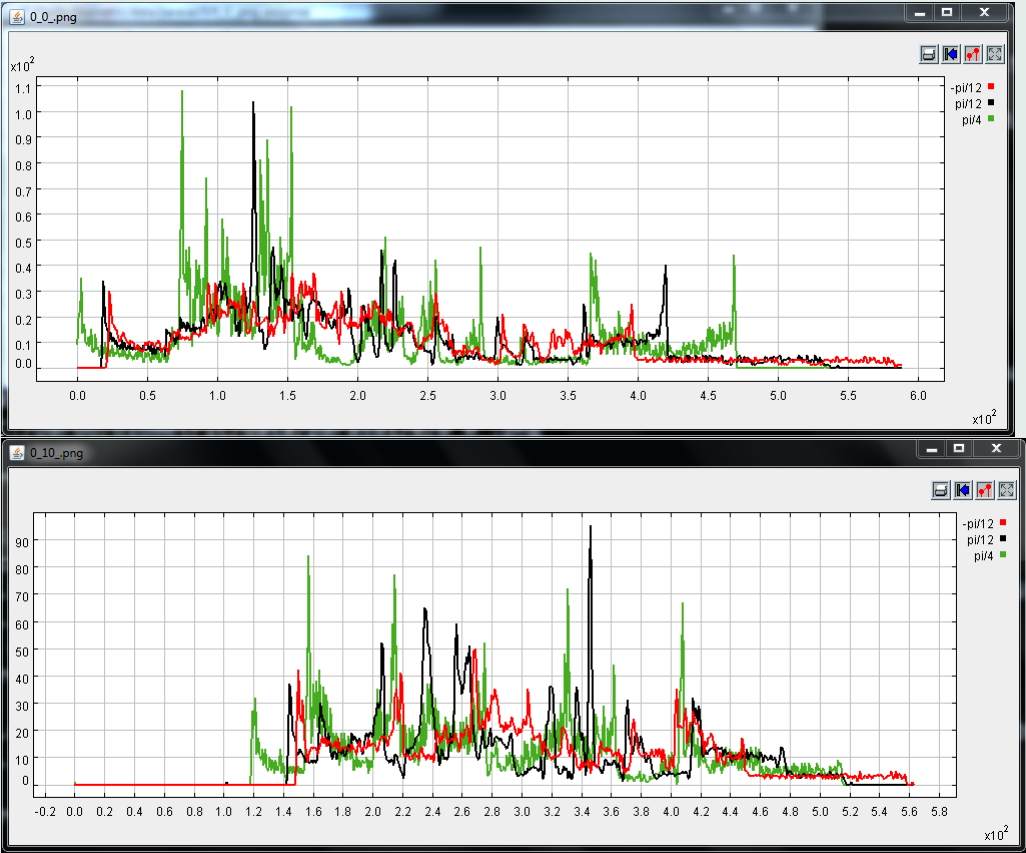
Page 16 of 26

Go Back

Full Screen

Close

Quit



11 pav.: 0_0 ir 0_10 autentiškų parašų požymiai

Home Page

Title Page

◀ ▶

◀ ▶

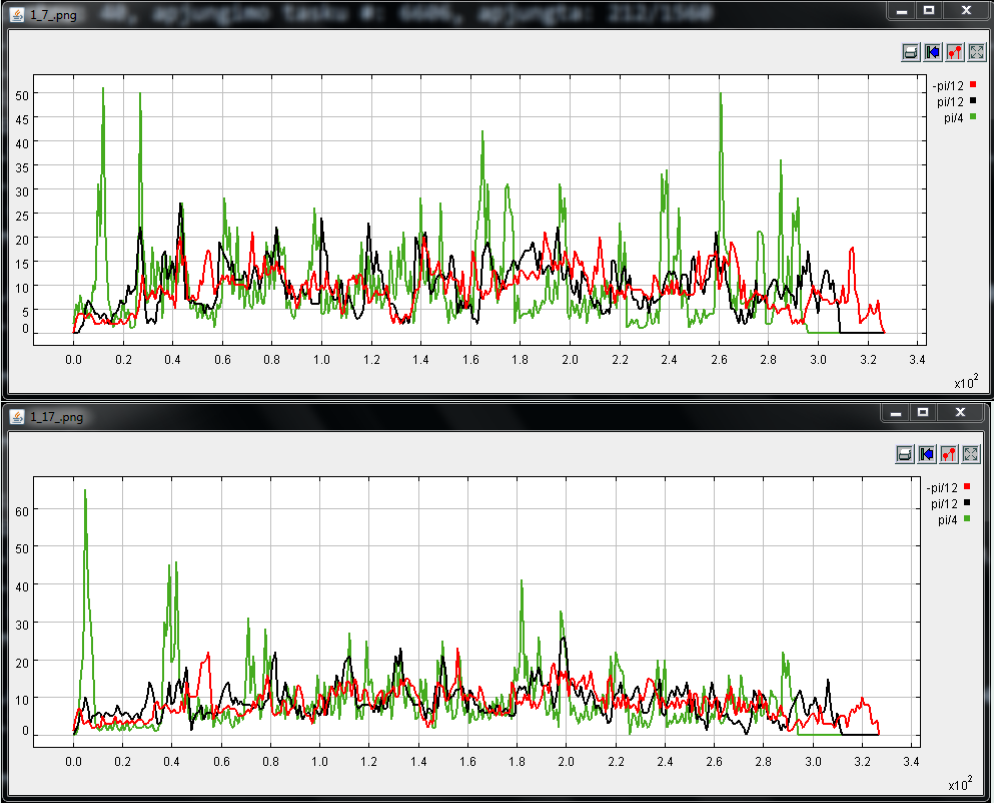
Page 17 of 26

Go Back

Full Screen

Close

Quit



12 pav.: 1_7 ir 1_17 autentiškų parašų požymiai

lygūs nuliui);

$$\rho(\mathbf{u}, \mathbf{u}) = 1, \tag{4}$$

jei parašas $\mathbf{u}$ nėra tuščias. Pagrindinė problema, kurią reikia išspręsti lyginant parašus yra ta, kad neprasminga lyginti skaitliukus $\#u_k[x]$ ir $\#v_k[y]$ tik sutampantiems argumentams $x = y$. Taip yra dėl to, kad, kaip taisyklė, skirtingi parašai turės skirtingas argumentų x ir y kitimo sritis ir, kas yra svarbiausia, autentiški parašai turės individualias skaitliukų ekstremumų pozicijas ir todėl reikia rasti būdą jas išlyginti. Tuo tikslu rekomenduojama naudoti mastelio kraipymo techniką, kuri angliškai vadinama DTW (*Dinamic Time Warping*). Tegul $\|\#u_k\|$ žymi bendrą $\#u_k$ skaitliuko reikšmių sumą, t.y.

$$\|\#u_k\| = \sum_{x=0}^X \#u_k[x].$$

Toliau mums bus patogų laikyti, kad visų skaitliukų argumento reikšmių aibė šiek tiek praplėsta, kad visiems parašams $\#u_k[0] = 0$. Įveskime kaupiamąją dviejų parašų $\mathbf{u}$ ir $\mathbf{v}$ panašumo matricą $\rho[x][y]$, kurios elementų skaičiavimui taikysime tokias nuoseklaus skaičiavimo taisykles:

1. $\rho[0][y] = 0, y = 0, 1, \dots, Y$, ir $\rho[x][0] = 0, x = 0, 1, \dots, X$;
2. $\rho[x][y] = \max\{\max\{\rho[x-1][y], \rho[x][y-1]\}, \rho[x-1][y-1] + \sum_{k=0}^2 w_k * \min\{\#u_k[x], \#v_k[y]\}\},$

$x = 1, \dots, X, y = 1, \dots, Y$, kur savoriai

$$w_k = \frac{2}{3 \max\{1, \|\#u_k\| + \|\#v_k\|\}}$$

parinkti taip, kad idealiu atveju, kai $\mathbf{u} = \mathbf{v}$, $\max_{x,y} \rho[x][y] = \rho[X][Y] = 1$.

Pagal apibrėžimą $\mathbf{u}$ ir $\mathbf{v}$ parašų požymių panašumu laikysime skaičių

$$\rho(\mathbf{u}, \mathbf{v}) = \rho[X][Y]. \tag{5}$$

Nesunku įsitikinti, kad taip apibrėžtas panašumas tenkina visus jam keliamas sąlygas (1)-(4).

Pailiustruosime parašų požymių panašumo metriką konkrečiu atveju. Paprastumo dėlei laikysime, kad skaitliukai $\#u_1, \#u_2, \#v_1$ ir $\#v_2$ yra trivialūs, t.y. $\#u_1 \equiv 0, \#u_2 \equiv 0, \#v_1 \equiv 0$ ir $\#v_2 \equiv 0$, o

$$\#u_0 = \{0, 2, 2, 0, 3, 0, 1, 1\}$$

ir

$$\#v_0 = \{0, 3, 1, 2, 4, 0, 1\}$$

Tuomet normuojantis svoris $w_0 = \frac{2}{9+11} = 0.1$ (šiuo atveju 3 vardiklyje nerašėme, nes mūsų dirbtiniame pavyzdyje antrieji ir tretieji skaitliukai yra trivialūs, ko realiems duomenims negali būti, nes visuomet

$\|u_0\| = \|u_1\| = \|u_2\|$). Apskaičiuojame kaupiamąją panašumų ρ lentelę

$\rho =$

1	0	0.2	0.4	0.4	0.7	0.7	0.7	0.8
0	0	0.2	0.4	0.4	0.7	0.7	0.7	0.7
4	0	0.2	0.4	0.4	0.7	0.7	0.7	0.7
2	0	0.2	0.4	0.4	0.5	0.5	0.5	0.5
1	0	0.2	0.3	0.3	0.3	0.3	0.4	0.4
3	0	0.2	0.2	0.2	0.3	0.3	0.3	0.3
0	0	0	0	0	0	0	0	0
	0	2	2	0	3	0	1	1

Todėl $\rho(\mathbf{u}, \mathbf{v}) = \rho[7][6] = 0.8$. Kaupiamoji panašumų lentelė naudinga rasti argumentų išlyginimo taisyklę. Tuo tikslu pradėdame kelionę nuo viršutinio dešiniojo letelės langelio, kuriame $\rho[7][6] = 0.8$ ir leidžiamės įstrižai žemyn/kairiau, jei einamojo langelio reikšmė didesnė už kairesniojo ir vienu žemiau esančio langelio reikšmės arba keturiuose gretimuose langeliuose, kurių viršutinis dešinysis langelis sutampa su einamuoju langeliu. Jei taip nėra, tai pasirenkame kairesnįjį langelį, jei jame įrašyta reikšmė sutampa su einamajame langelyje esančia reikšme, priešingu atveju einamąjį langelį paslenkame vienu langeliu žemiau. Tokiu būdu žingsnis po žingsnio nukeliausime nuo viršutinio dešiniojo langelio iki apatinio kairiojo langelio. Apačioje pateikta lentelė iliustruoja mūsų pavyzdžio

$\rho =$

1	0	0.2	0.4	0.4	0.7	0.7	0.7	0.8
0	0	0.2	0.4	0.4	0.7	0.7	0.7	0.7
4	0	0.2	0.4	0.4	0.7	0.7	0.7	0.7
2	0	0.2	0.4	0.4	0.5	0.5	0.5	0.5
1	0	0.2	0.3	0.3	0.3	0.3	0.4	0.4
3	0	0.2	0.2	0.2	0.3	0.3	0.3	0.3
0	0	0	0	0	0	0	0	0
	0	2	2	0	3	0	1	1

1 lentelė: Požymių atitikmenų trajektorija sudaryta iš paryškinto šrifto langelių

langelių kelio trajektoriją, kur trajektorijos langelių reikšmės yra paryškintos riebesniu šriftu. Iš trajektorijos gauname tokius $\mathbf{u}[x]$ ir $\mathbf{v}[y]$ požymių argumentų atitikimus:

$$0 < - > 1; 1 < - > 1; 2 < - > 3; 4 < - > 4; 6 < - > 5; 7 < - > 6.$$

Skaičiuodami kaupiamąjį panašumą, mes pradėjome nuo apatinio kairiojo langelio ir užbaigėme viršutiniu dešiniuoju. Toks pasirinkimas yra natūralus, nes logiška laikyti, kad vieno parašo pats pirmasis indeksas atitinka pirmąjį lyginamo parašo požymių indeksą ir, analogiškai, yra laikoma, kad

$\rho^* =$

1	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1
4	0	0.2	0.2	0.2	0.3	0.3	0.3	0.3
2	0	0.2	0.4	0.4	0.5	0.5	0.5	0.5
1	0	0.2	0.4	0.4	0.7	0.7	0.7	0.7
3	0	0.2	0.4	0.4	0.7	0.7	0.8	0.8
0	0	0.2	0.4	0.4	0.7	0.7	0.8	0.8
	0	2	2	0	3	0	1	1

2 lentelė: Jungtinis panašumas ir požymių atitikmenų jungtinė trajektorija

atitinka galiniai parašų požymių indeksai. Praktiniai eksperimentai rodo, kad naudinga apibrėžti reliatyvų panašumą formule

$$\rho'(\mathbf{u}, \mathbf{v}) = 2\rho(\mathbf{u}, \mathbf{v}) - \rho^*(\mathbf{u}, \mathbf{v}), \tag{6}$$

kur jungtinis panašumas $\rho^*(\mathbf{u}, \mathbf{v})$ gaunamas pradedant skaičiuoti kaupiamąjį panašumą nuo viršutinio kairiojo langelio: Todėl $\rho^*(\mathbf{u}, \mathbf{v}) = \rho^*[X][0] = 0.8$ ir $\rho'(\mathbf{u}, \mathbf{v}) = 2\rho(\mathbf{u}, \mathbf{v}) - \rho^*(\mathbf{u}, \mathbf{v}) = 20.8 - 0.8 = 0.8$.

13 iliustruoja ρ ir ρ' metrių kokybę, taikant jas verifikuoti 113 parašų paimtų iš Lietuvos Teismo

Ekspertizės DB1 parašų bazės. Lyginant dvi DET kreives galima konstatuoti, kad santykinė metrika ρ' šiai duomenų bazei geriau pasiteisina.

1.6. Nuorodos

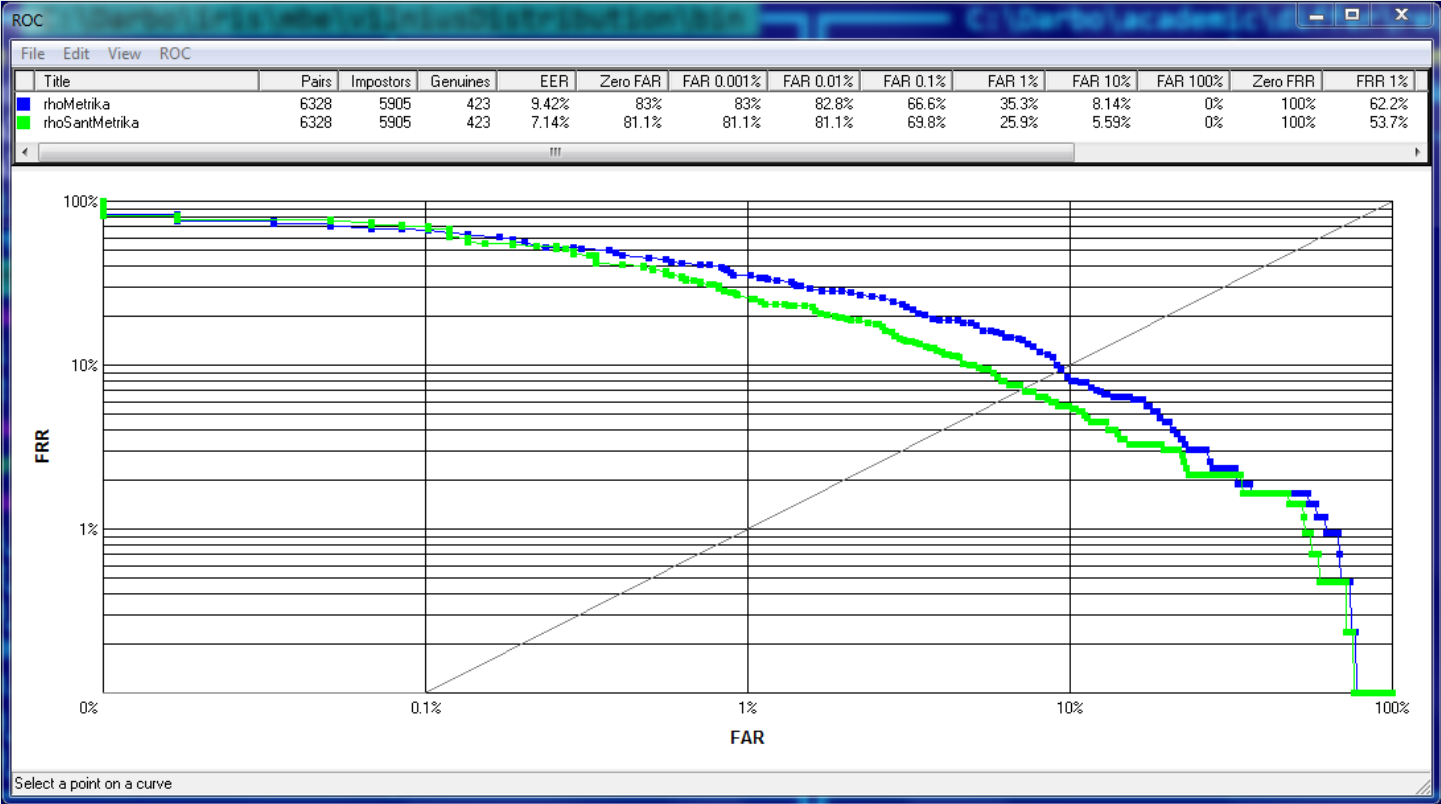
Literatūra

[1] [Off-Line Signature Recognition Systems](#), V. A. Bharadi, H. B. Kekre

1.7. Praktinis darbas

*Realizuokite parašų atpažinimo algoritmą. Bus vertinimas kiekvienas jūsų kuriamo parašų atpažinimo algoritmo etapas, algoritmo kūrimo etapai, jų vertinimo balai ir terminai pateikiami žemiau. Papildomai galutinis parašų atpažinimo algoritmas bus vertinamas pagal pateiktų algoritmui nežinomų skenuotų parašų **atpažinimo kokybę**. Kokybė vertinama pagal algoritmo DET kreivę, palyginant jūsų algoritmu gautą DET kreivę su kitų autorių (jūsų kolegų) gautomis DET kreivėmis.*

Reikalingų algoritmui atiderinti parašų pavyzdžius rasite [čia](#). Parašų atpažinimo užduotis yra privaloma. Darbas atliekamas auditorijoje pratybų metu arba savarankiškai namuose atsiskaitant auditorijoje pratyboms skirtu laiku už atskirus etapus. Išimtiniais atvejais (pvz. esate ilgalaikėje studijų komandiruotėje), galima atsiskaityti *skaipu*. Programavimo kalba neribojama, tačiau programuojan-



13 pav.: Parašų verifikavimo DET kreivės gautos Lietuvos teismo ekspertizės DB1 parašų duomenims. Mėlyna kreivė žymi ρ metrika gautą DET kreivę, žalia santykinė ρ' metrika.

tys Java kalba turės privalumą, nes semestro eigoje bus pateikti java kodai užduočių, kurioms terminas baigėsi, ir pateiktus kodus bus galima naudoti realizuojant tolimesnius algoritmo etapus.

1. Skenuoto parašo binarizacija.

Atlikite nuskaityto skenuoto parašo binarizaciją 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 0.5 balo, iki 09.13 (visuomet imtinai).

2. Parašo išlyginimas.

Išlyginkite skenuoto parašo vaizdą vertikalia kryptimi. Išlyginimui reikalingoms kreivėms gauti panaudokite eksponentinį filtrą, kurio java kodą rasite čia, rekomenduotina filtro $\sigma = 100$.

Atsiskaitoma pademonstruojant vidurkintą poslinkių kreivę ir išlygintą parašo vaizdą.

Vertinama iki 0.5 balo, iki 09.27 .

3. Požymių apskaičiavimas.

Apskaičiuokite parašo požymius pasirinktam masteliui ir trims parašo taškų projektavimo kryptims. Požymiams skaičiuoti naudokite išlygintą parašo vaizdą.

Atsiskaitoma pademonstruojant tris projekcijų skaitliukų kreives, kurioms pavaizduoti galima išorines priemones, pvz. Ekselio programą.

Vertinama iki 0.5 balo, iki 10.04 .

4. Požymių palyginimas.

Realizuokite dviejų parašų poros palyginimą pagal (5) ir (6) panašumo metrikas. Gautas panašumo reikšmės įrašykite *.roc formatu. Kadangi šis formatas reikalauja neneigiamų sveikaskaičių panašumo reikšmių, gautas ρ metrikos panašumo reikšmės dauginkite iš 1000 ir apvalinkite iki artimiausio neneigiamo sveikąjo skaičiaus. Požymiams skaičiuoti naudokite išlygintą parašo vaizdą.

Atsiskaitoma pademonstruojant dvi *.roc kreives analogiškas 13 kreivėms. Reikalaujama, kad bent vienos kreivės EER būtų mažesnis už 20%.

Vertinama iki 1 balo, iki 10.18 . Algoritmai, kurių $EER < 10$

5. Užduočių, kurioms galutinai baigėsi terminas, Java kodai