

Eiropas Reģionālās attīstības fonds

Prioritāte: 2.1. Zinātne un inovācijas

Pasākums: 2.1.1.1. Zinātne, pētniecība un attīstība

Aktivitāte: 2.1.1.1. Atbalsts zinātnei un pētniecībai

Projekta nosaukums: „Multimodālas biometrijas tehnoloģija drošai un ērtai personu
autentifikācijai” (BiTe)

Līguma noslēgšanas datums: 10.12.2010.g.

Projekta sākuma datums: 01.01.2011.g.

Projekta beigu datums: 31.12.2013.g.

Vienošanās Nr.2010/0285/2DP/2.1.1.1.0/APIA/VIAA/098

Eiropas Reģionālās attīstības fonda finansējuma saņēmējs: Elektronikas un
datorzinātņu institūts (EDI)

ZINĀTNISKĀ PĒTĪJUMA PROGRESU APLIECINOŠĀ DOKUMENTĀCIJA

Pārskata numurs Nr.3. par periodu no 01.07.2011.g. līdz 31.10.2011.g.

Projekta zinātniskais vadītājs: Modris Greitāns, Dr.sc.comp., vad. pētnieks
Pētījuma projekta izpildītāju saraksts: Mihails Broitmans, Dr.sc.comp., vad. pētnieks
Rihards Fuksis, pētnieks
Mihails Pudžs, pētnieks
Rinalds Ruskuls, asistents
Oļegs Ņikišins, asistents
Zanda Seržāne, asistente
Artūrs Kadiķis, programmēšanas inženieris
Dāvis Barkāns, elektronikas inženieris
Teodors Eglītis, informātikas/elektronikas tehniķis

ELEKTRONIKAS UN DATORZINĀTŅU INSTITŪTS
Rīga, Latvija
2011. gads

Anotācija	3
Ievads	3
Rezultātu kopsavilkums.....	3
Aktivitāte: Lietišķie pētījumi	5
Sejas un plaukstu attēlu iegūšanas metožu izpēte.....	5
Biometrisku datu apstrādes algoritmu darbības izvērtēšana.....	13
Biometrisku datu kriptēšanas, glabāšanas un lietojuma apakšsistēmas izveide	14
Atsevišķu sistēmas komponentu un bloku izstrāde un izpēte	18
Secinājumi	19
Atsauces	20
Pielikumi.....	21

Anotācija

BiTe ir ERAF līdzfinansēts projekts zinātnei un pētniecībai. Projekta mērķis ir droša, ērta un plaši pielietojama personu identifikācijas risinājuma izveide. Projekts ietver sevī sekojošas aktivitātes: Lietišķos pētījumus biometrijas parametru ieguvē, apstrādes metožu implementēšanā ierobežotu resursu sistēmās, atsevišķu sistēmas komponentu un bloku izstrādē; Eksperimentālos pētījumus tehnoloģiju demonstratora izstrādē; Rūpniecisko tiesību aizsargāšana, patenta pieteikums; Projekta publicitāte. Šajā dokumentā dots pārskats par projekta otrajā periodā (01.04.2011.-30.06.2011) veiktajiem pētniecības darbiem un šobrīd sasniegtiem rezultātiem.

Projektu atbalsta Eiropas Reģionālās attīstības fonds, līguma Nr. 2010/0285/2DP/2.1.1.1.0/APIA/VIAA/098

Ievads

Identitātes nozagšana ir viens no modernajiem noziegumiem, kas „zaglim” dod iespēju radīt personām finansiālus zaudējumus. Katru gadu identitātes nozagšana pasaulē rada aptuveni 221 miljardu dolāru zaudējumus. Lielu interesi par biometrijas izmantošanu personu identifikācijā izrāda banku sektors. Bankas atzīst, ka šādas sistēmas ieviešana dotu iespēju aizstāt (vai kombinēt) uz parakstu vai PIN kodu ievadi balstītu norēķinu karšu autorizāciju ar kartes īpašnieka biometrisku atpazīšanu. Radot ērtu, lētu un drošu personu identifikācijas risinājumu, ir plašas iespējas to ieviest ikdienas dzīvē – bankās, tirdzniecības vietās, objektu piekļuves un lietošanas kontrolē (ieskaitot autotransportu) u.c.

BiTe pētniecības grupas darbs ir saistīts ar augstāk minēto problēmu risinājumu. Šajā projekta pārskata posmā ir veikti darbi un sasniegti rezultāti sekojošos aktivitātes „Lietišķie (rūpnieciskie) pētījumi” pētījumos:

- Sejas un plaukstu attēlu iegūšanas metožu izpēte;
- Biometrisku datu apstrādes algoritmu darbības izvērtēšana;
- Biometrisku datu kriptēšanas, glabāšanas un lietojama apakšsistēmas izveide;
- Atsevišķu sistēmas komponentu un bloku izstrāde un izpēte.

Rezultātu kopsavilkums

Projekta trešajā periodā (01.07.2011.-31.10.2011.) veikto pētniecības uzdevumu un rezultātu kopsavilkums ir sekojošs:

- Sejas un plaukstu attēlu iegūšanas metožu izpēte – iepriekšējos pārskata periodos tika izpētīts kā izveidot augstas jutības infrasarkanās un redzamās gaismas attēlu iegūšanas sistēmu. Tika apgūtas arī prasmes sarežģīto sistēmu izveidē uz FPGA un izveidota 12-bitu izšķirtspējas attēlu iegūšanas sistēma. Vēlāk izstrādājām iespiedplati sejas atpazīšanas algoritmu testēšanai un izveidojām FPGA bāzētu attēlu ieguves sistēmu. Paralēli tika strādāts pie plaukstu detektēšanas, kur pirmais

uzdevums bija pareizi izdalīt plaukstu parametrus, tāpēc izveidojām pagaidu sistēmas blokshēmu (attēlu apstrādes bloks, pazīmju atlases bloks un klasifikators), kur kā pazīmes tika izvēlēti plaukstu kontūra fragmenti, kuru aprakstīšanai lietojām kontūra elementus (*edges*). Šajā periodā tika projektēts 2D filtrs (konvolūcijas veikšanai FPGA platformā), tika iegūta un attēlota bilde reālā laikā un apvienojot iepriekš minēto, tika veikta bildes filtrēšana reālā laikā. Tika strādāts arī pie plaukstu detektēšanas, kur plaukstu detektēšanai attēlā tika izdalītas plaukstu specifiskās pazīmes (pirkstiem līdzīgas struktūras).

- Biometrisku datu apstrādes algoritmu darbības izvērtēšana – Iepriekš tika realizēta 1. versija 2D sejas atpazīšanas algoritmam, kas sastāvēja no vairākiem posmiem (sejas detektēšana, lokalizācija un atpazīšana). Šis algoritms vēlāk tika implementēts iegultā sistēmā, kur uzlabojām funkcionālo bloku precizitāti. Tika apskatīts arī sejas ādas detektēšanas algoritms un uzsākta virtuālā instrumenta izveide LabView. Šajā instrumentā izveidota sistēma, kas nosūta un saņem datus, apstrādā tos un grafiski attēlo. Pēdējā periodā centrālais virziens bija iegultas sistēmas izveidošana un algoritmiskās bāzes uzlabošana. Tika veidots automātisks sejas atpazīšanas algoritms signālprocesoram TMS320C6416 un uzsākts darbs pie kameras moduļa izstrādes šim procesoram.
- Biometrisku datu kriptēšanas, glabāšanas un lietojuma apakšsistēmas izveide – Iepriekšējos pārskata periodos: izanalizēti vairāki realizācijas algoritmi *biohash* funkcijas skaitļošanai; apskatītas vairākas matemātiskās metodes; izveidots algoritms, kas ar biokodu un papildus informāciju iegūst vidēji tādu pašu EER kā salīdzinot nešifrētus datus; izstrādāta programmatūra, kas ar pietiekami mazu aprēķinu kļūdu salīdzina nešifrētus un šifrētus datus uz Java viedkartes un izstrādāts protokols, kas ļauj izveidot drošu sakaru kanālu starp viedkarti un autentifikācijas iekārtu. Šajā periodā tika izpētītas ROI automātiskās noteikšanas metodes un sastādīts ROI noteikšanas algoritms. Tika piedāvāti dažādi plaukstu un rokas pirkstu ģeometrisku parametru mērījumi. Ir arī apskatīti vairāki biometrijas datu šifrēšanas algoritmi, kā arī daži kļūdu koriģēšanas algoritmi. Sagatavotas nepieciešamās operācijas ar paplašinātiem galīgu lauku polinomiem $GF(2^m)$.
- Atsevišķu sistēmas komponentu un bloku izstrāde un izpēte – Sākumā tika veidota nepieciešamo atsevišķo funkcionālo bloku elektriskās principiālās shēmas *Altium Designer* vidē un uzsākta šo bloku iespiecplašu projektēšana, komponentu izvietošana, galveno principiālo bloku trasēšana. Vēlāk tika uzsākta iespiecplates izstrāde šajā vidē, un tika izveidota principiālā viedkaršu komunikācijas maketa shēma. Tika uzsākta arī kameras moduļa iespiecplates otrās versijas montāža (veikta attēlu sensora montāža un testēšana). Šajā periodā ir izveidota programma plaukstu reģiona automātiskai detektēšanai un izveidota programma priekš MSP430 F2013 mikrokontroliera I2C kopnes interfeiss ar Aptina attēlu sensoru. Ir veikti eksperimenti ar Aptina attēlu sensoru

Šie pētījumi tiek veikti, lai nākamā gada otrajā ceturksnī uzsākamās „Eksperimentālās izstrādes” aktivitātes ietvaros varētu izveidot pilnvērtīgu projektā izstrādājamās tehnoloģijas demonstratoru un metodisko aprakstu.

Turpmāk dokumentā ir detalizēti aprakstītas katras uzdevumu grupas pētniecības darbības un rezultāti.

Aktivitāte: Lietišķie pētījumi

Sejas un plaukstas attēlu iegūšanas metožu izpēte

Laika posmā no jūlija līdz oktobra beigām tika veiktas šādās aktivitātes:

- 2D Filtra projektēšana (konvolūcijas veikšanai FPGA platformā)
- Bildes iegūšana un attēlošana reālā laikā (darbs ar RAM pie 125MHz)
- Bildes filtrēšana reālā laikā (apvieno iepriekš 1. un 2. minētos)
- Attēlu sensora iespaidplates montāža un attēlu iegūšanas sistēmas pilnveidošana izmantojot FPGA

Lai to sasniegtu tika pētīti veidi kā implementēt 2D Filtrus FPGA, kā arī optimizēt attēla ierakstīšanu un izvadīšanu.

2D Filtra projektēšana

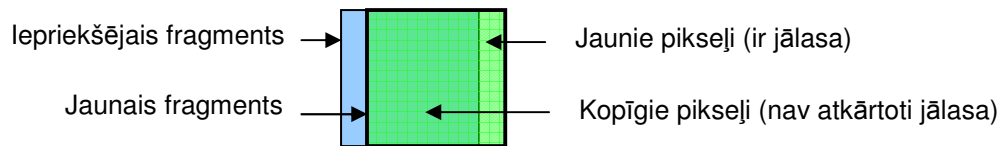
2D filtrēšana, ko jāimplementē FPGA ir konvolūcija starp masku un tekošo ieejas attēlu. Lai veiktu šādu operāciju ir nepieciešams reizināt un summēt maskas pikselus ar ieejas attēla fragmentu pikseliem, un saglabāt iegūtos datus. Ir vairāki veidi kā to implementēt FPGA.

Standarta gadījumos filtrēšana tiek veikta sekojoši:

- Attēls tiek saglabāts atmiņā;
- Tiek lasīti nepieciešamie dati, jeb attēla fragments, kam virsū uzklāj filtra masku, piemēram, bildes augšējais stūris (9 x 9 pikseli);
- Tiek veikta nolasītā fragmenta sareizināšana ar filtra masku, un iegūts viens filtrētā attēla pikselis;
- Pēdējās divas darbības izpilda visiem iespējamiem vienāda izmēra attēla fragmentiem.

Šādā gadījumā filtra veiktspēju nosaka atmiņas maksimālais lasīšanas ātrums, protams, to ietekmē arī izmantotais FPGA.

Šo metodi ir iespējams optimizēt. Tā kā attēla fragmenti, kas tiek reizināti ar filtra masku, daļēji pārklājas (skat. att.1.), tad katram nākamā attēla fragmentam nav jālasa no atmiņas visi tā pikseli, jo daļa no vajadzīgajiem pikseliem pieder arī iepriekšējam fragmentam, un atrodas FPGA atmiņā (ir jau nolasīti). Tā, piemēram, att.1 var redzēt, ka, ja maskas izmērs ir 9x9, tad nākamā fragmenta 8x9 pikseli nav jālasa, ir jālasa tikai 1x9 pikseli.

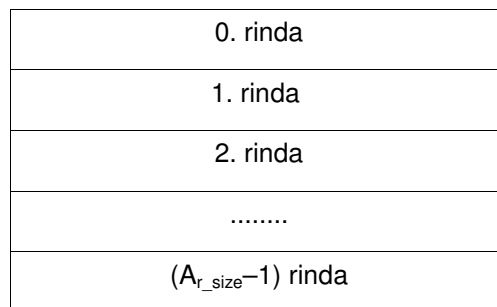


Att. 1. Attēla fragmentu pārkļāšanās

Ātrās filtrācija pamati

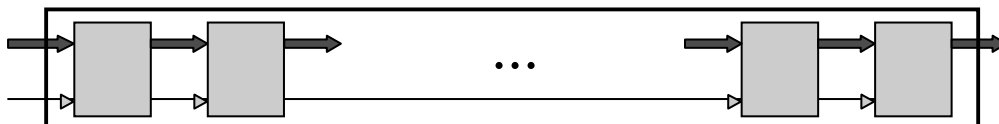
Šīs metodes pamatā ir tāda FPGA resursu organizācija, ar kuras palīdzību panāk, ka katra izejas pikseļa iegūšanai ieejā ir nepieciešams nolasīt tikai vienu pikseli. Tas, savukārt, ļauj veikt konvolūciju ar tikpat lielu ātrumu ar kādu nāk pikseli no attēlu sensora, jeb filtrēt attēlu reālā laikā bez ārējās RAM starpniecības.

Šādai pieejai FPGA iekšējā atmiņā (RAM bitos un reģistros) ir nepieciešams glabāt veselas attēla rindas, nevis attēla fragmentus. Atmiņas daudzumu izvēlas pēc filtra maskas izmēra, mūsu gadījumā tā ir 9x9 ($A_{r_size}=9$, turpmāk aprēķinos tiek pieņemts, ka filtra maska ir kvadrātveida, tāpēc arī malas apraksta tikai viens lielums); pēc izmantotā attēla rindas garuma – mūsu gadījumā tas ir 640 ($R_{size}=640$). Kopā FPGA RAM bitos ir jāglabā $A_{r_size}^2 + (A_{r_size} - 1) \cdot (R_{size} - A_{r_size})$ pikseli. Atmiņa ir organizēta pa rindām (skat. att.2.).



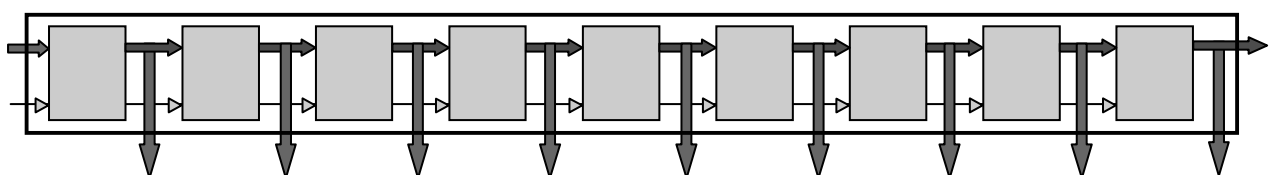
Att. 2. Filtra atmiņas organizācija

Rindas tiek dalītas aktīvajās un pasīvajās, katra no tām ir organizēta kā pikseļu FIFO (**F**irst **I**n **F**irst **O**ut). Katra pasīvā rinda (skat. att.3.) glabā pikselus, nodrošinot piekļuvi tikai pie viena pikseļa – rakstīšanai, un pie viena – lasīšanai. Pasīvās rindas ir realizētas ar FPGA RAM bitiem kā bīdes reģistrs, kur dati ir pieejami tikai bīdes reģistra izejā.



Att. 3. Filtra pasīvās rindas organizācija

Aktīvās rindas (skat. att.4.) no pasīvām atšķiras ar to, ka tās nodrošina arī vienlaicīgu piekļuvi pie visiem tajās saglabātiem datiem, tāpēc tās tiek realizētas ar reģistru palīdzību.



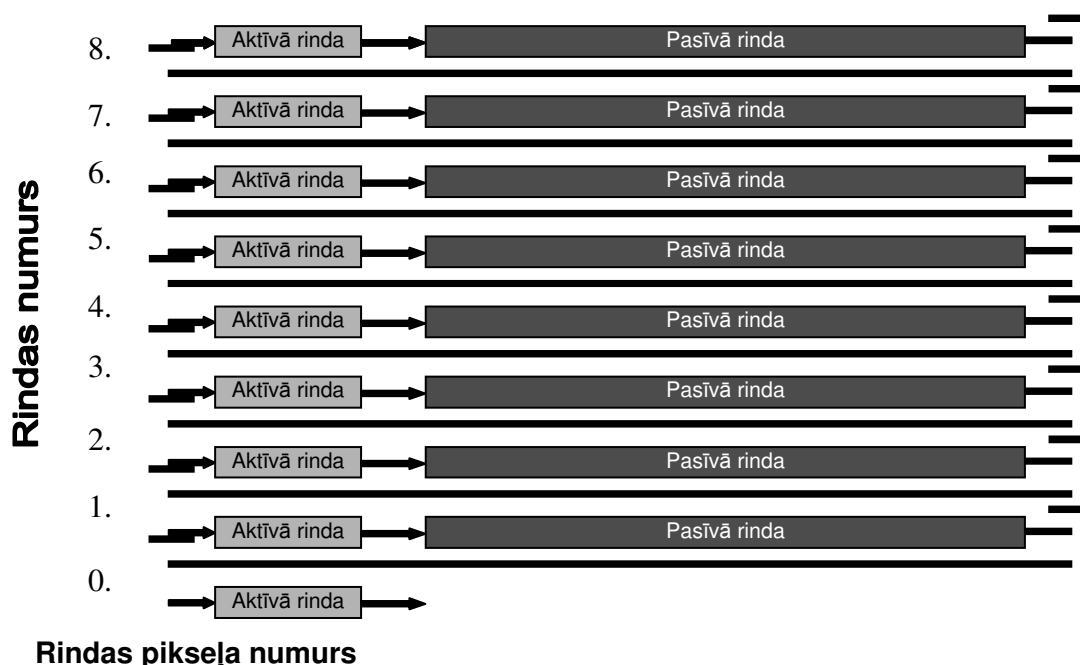
Att. 4. Filtra aktīvās rindas organizācija

Aktīvo rindu garums ir atkarīgs no maskas garuma ($A_{r_size}=9$), pasīvo rindu garums tiek noteikts pēc formulas: $P_{r_size} = R_{size} - A_{r_size}$, kur

R_{size} – kopējais pikseļu daudzums vienā attēla rindā (mūsu gadījumā ir 640),

A_{r_size} – aktīvās rindas garums (jeb 9).

Izveidotās rindas (aktīvās un pasīvās) tiek pamīšus saslēgtas virknē:



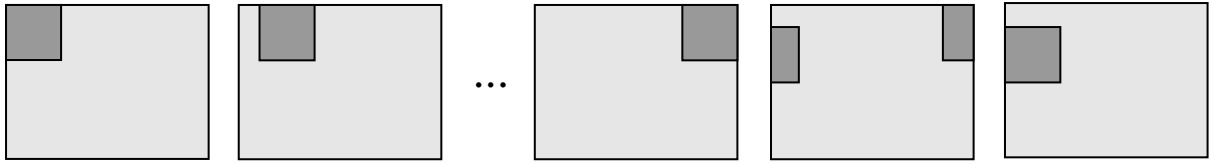
Att.5. Filtra atmiņas organizācija

Katrs pāris (aktīvā rinda + pasīvā rinda) glabā sevī vienu attēla rindu, jeb 640 pikseļus, 9 – aktīvajā rindā un 631 – pasīvajā. Aktīvā filtra daļa (aktīvo rindu kopa) tiek izmantota attēla fragmentu iegūšanai un to reizināšanai ar filtra masku, pasīvā daļa – tikai pikseļu glabāšanai. Bīdot šādā sistēmā attēlu sensora uzdotus datus pēc $(A_{r_size}-1) \cdot R_{size} + A_{r_size}$ operācijām aktīvajās rindās parādīsies bildes pirmais filtrējamais apgabals (skat. Att. 6)



Att.6. Filtra attēla logs

Bīdot pikseļus tālāk, attēla logs, kas ir reprezentēts ar filtra aktīvo daļu, bīdīsies pa labi ar t. s. *rollover effect* – t. i. katru reizi izejot no ekrāna labajā pusē, tas atkal parādīsies kreisajā pusē zemāks par 1 pikseli (skat. Att. 7)



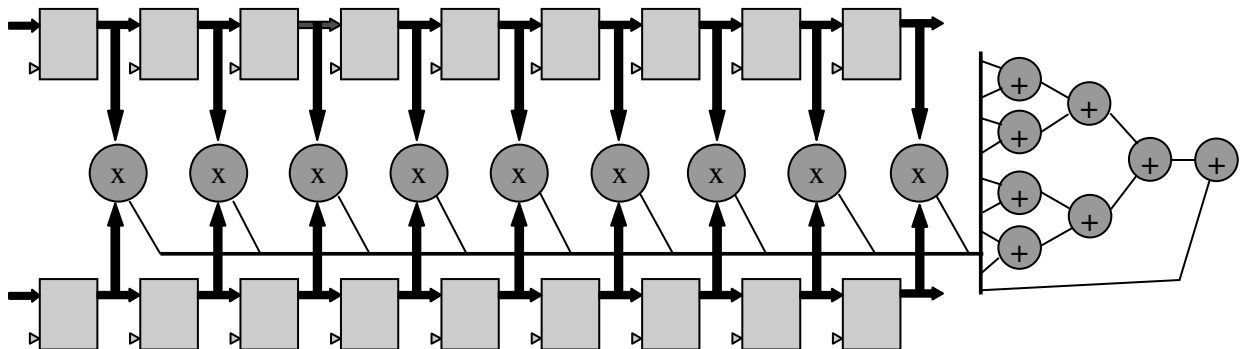
Att. 7. Filtra loga kustība par attēla pikseļiem vienai attēla rindai

Pikseļu koordinātes logā sakrīt ar aktīvās rindas numuru (uzdod pikseļa Y koordināti; ass vērsta uz leju), un aktīvas rindas pikseļa numuru (uzdod pikseļa X koordināti). Šo secību ir jāievēro, lai pareizi reizinātu filtra masku ar attēla fragmentu.

Lai veiktu maskas un attēla fragmenta reizināšanu tika pielietoti FPGA iebūvētie reizinātāji, kas tika piesaistīti pie aktīvām attēla un maskas rindām. Lai izveidotais filtrs būtu universāls, tas ir, tam varētu dinamiski mainīt masku, tika izveidots papildus bloks, kas atbild par maskas pikseļu glabāšanu. Šis bloks sastāv no virknē slēgtām aktīvām rindām, kas veido FIFO ar iespēju vienlaicīgi piekļūt pie maskas pikseļiem. Šādā masīvā maskas pikseļus ieraksta virknē, piemēram, no datora FTDI.

Nākamajā attēlā ir redzama aktīvās rindas detalizētāka reprezentācija, tā sastāv no:

1. Datu bīdes reģistra
2. Maskas bīdes reģistra
3. Reizinātājiem
4. Saskaitītājiem



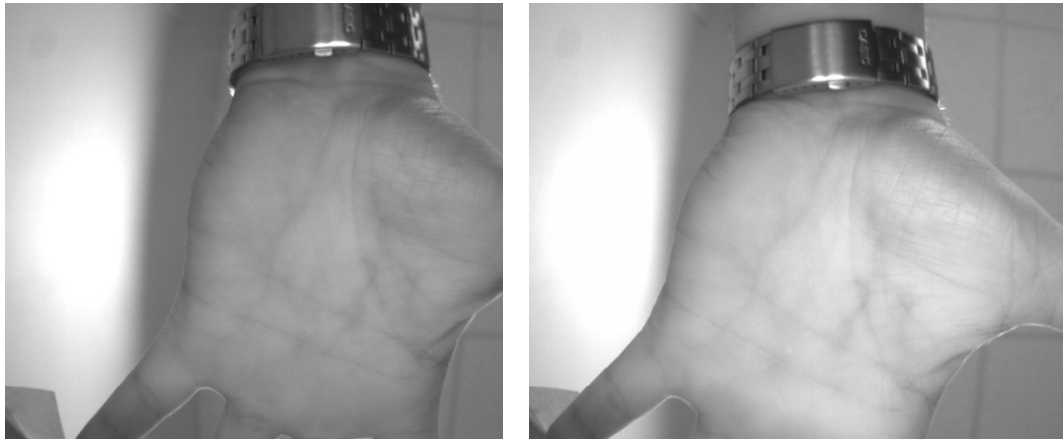
Att. 8. Attēla fragmenta rindas un maskas rindas reizinātāja organizācija

Praktiskā realizācija un testēšana FPGA

Izveidotais 2D filtrs ar aktīvās maskas izmēru 9x9 aizņem

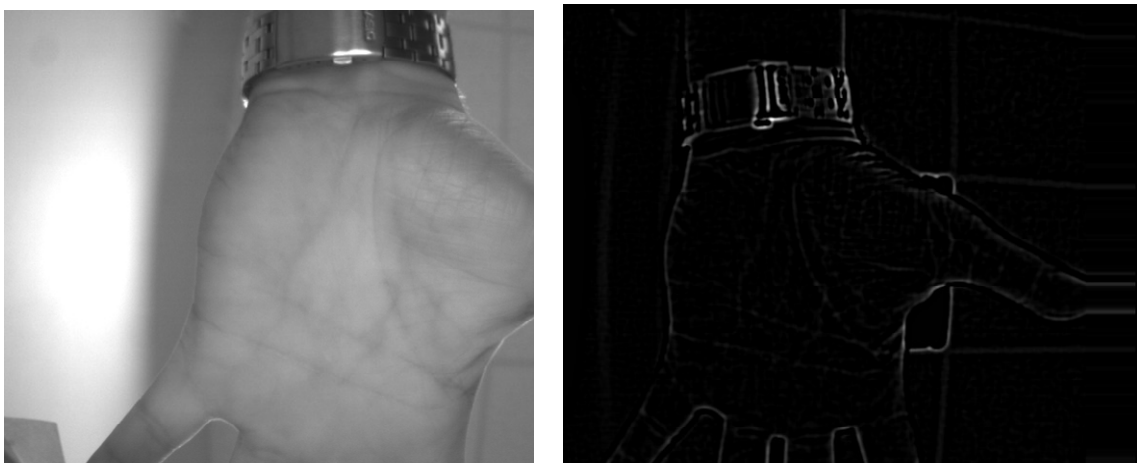
1. 4% no loģisko elementu daudzuma
2. 2% no RAM bitiem
3. 30% no reizinātājiem

Filtra pārbaudei tika veikti dažādi testi, pirmais no tiem bija maskas uzdošana pirms kompilācijas – nozīmē to, ka masku nav iespējams dinamiski mainīt. Ja vēlas savādāku masku jāpārkompilē FPGA projekts. Pirmā testa maska bija vienības maska, tas ir, filtrētais attēls ir vienāds ar oriģinālo:



Att. 9. Attēla filtrācija ar vienības masku

Nākošie testi tika veikti ar LoG (Laplancian of Gaussian) masku:



Att.10. Attēla filtrācija ar LoG masku

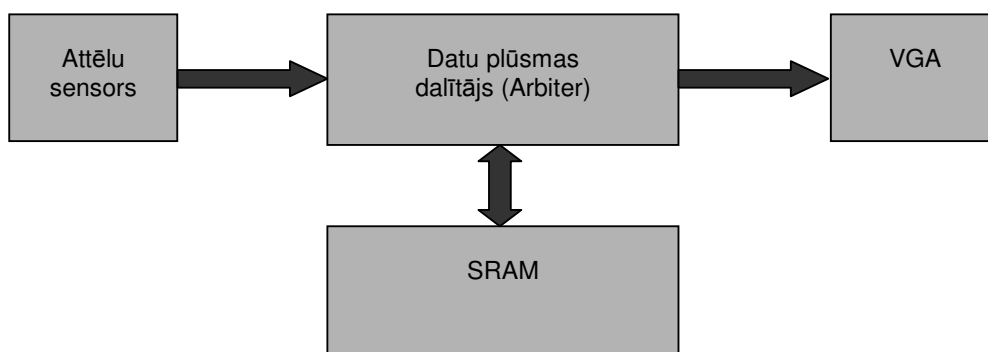
Redzams, ka ieejas attēls (pa kreisi) tiek nofiltrēts ar uzdoto masku. Attēlu ir iespējams uzņemt apstrādātu vai oriģinālu, tāpēc bildes nav identiskas – oriģinālās un apstrādātās ar uzdoto masku.

Bildes apstrādāšana reālā laikā

Līdz šim tika izstrādāta sistēma, kas nodrošina bildes ieguvī nospiežot pogu vai arī ar speciāli izveidotu datora programmatūru. Nākošais solis bija izveidot sistēmu, kas spētu attēlot bildi reālā laikā uz LCD monitora, ar izšķirtspēju 640x480. Izmantotais attēlu sensors un VGA kontrolieris strādā ar 25 MHz takts frekvenci.

Lai attēls tiktu gan ierakstīts, gan attēlots ir nepieciešams izveidot plūsmas dalītāju, kas pilda šādas funkcijas:

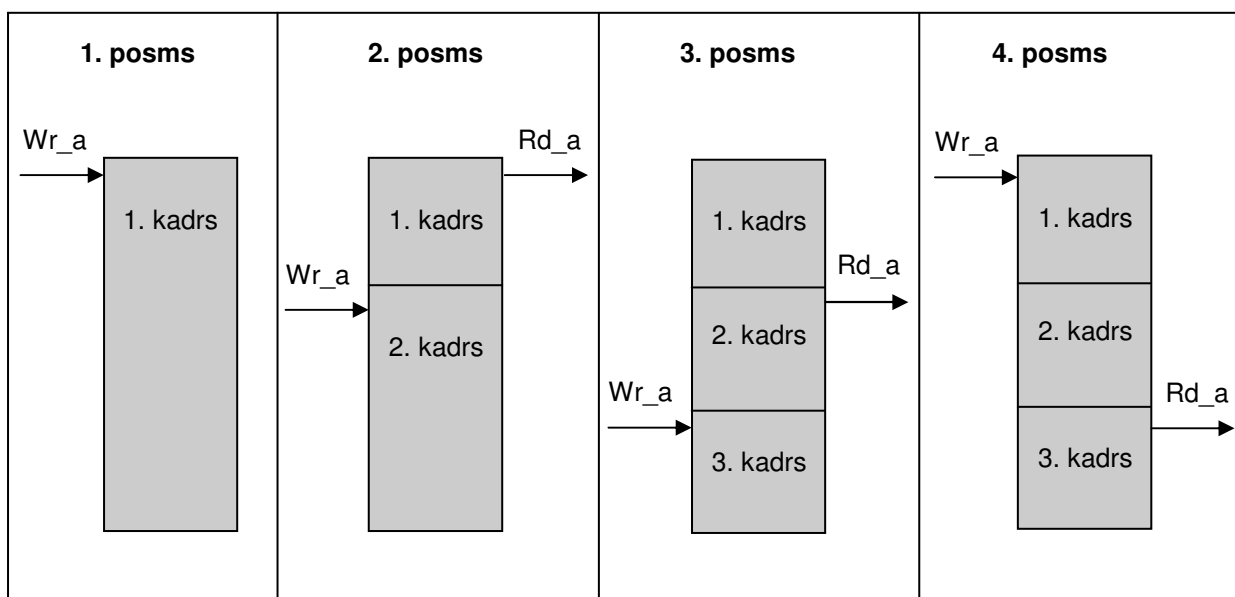
1. Sadala ierakstīšanas un nolasīšanas plūsmu starp attēlu sensoru, VGA un SRAM kontrolieri (skat. Att. 11)
2. Veic buferēšanu – saglabā SRAM atmiņā vairākus kadrus.



Att. 11. Datu plūsmas virzieni

Datu plūsmas dalītāja veicamās operācijas:

1. Sistēmas inicializācijas brīdī SRAM atmiņā tiek ierakstīts pirmais kads (lasīšana nenotiek);
2. Pēc pirmā kads ierakstīšanas, rakstīšanas adrese ($Wr_a \leq 2$. kads) tiek palielināta par 1 kads izmēru, tiek nolasīts pirmais kads ($Rd_a \leq 1$. kads);
3. Pēc katra kads ierakstīšanas, rakstīšanas adrese tiek palielināta par 1 kads vai nullēta, sasniedzot 4. kads lasīšana notiek no pēdējā pilnībā ierakstītā kads;
4. Cikliski tiek atkārtota trešā operācija.



Att.12. Kadru pārslēgšanās algoritms (2.-4. posms cikliski atkārtojas)

Attēlu sensora iespaidplates montāža un attēlu iegūšanas sistēmas pilnveidošana izmantojot FPGA

Ir daļēji veikta elektronisko komponentu montāža sejas atpazīšanas kameras moduļa iespaidplatei (savienotāji, oscilators, R un C elementi, attēlu sensors). Sakarā ar nevienmērīgu savienojuma vietu alvojumu, attēlu sensoru tomēr neizdevās uzmontēt. Iemesls: attēlu sensora – Aptina MT9V024, virsmas stiprinājuma veids ir BGA (**B**all **G**rid **A**rray). Atkārtoti veicot lodēšanu, attēlu sensoram iestājās neatgriezeniskas izmaiņas – tas

vairs nedarbojas. Pārējo komponentu lodēšana tika apturēta līdz jauna attēlu sensora iegādei.

Uz FPGA bāzēta attēla ieguves sistēmas optimizācija. (sk. Pielikums Nr. I)

Tiek apskatīti efektīvu kodu izveides algoritmi, kas aprakstīti Pong P. Chu „RTL hardware design using VHDL”, tas palīdzēja uzlabot:

1. VGA kontrolieri;
2. Attēlu sensora kontrolieri;
3. ASRAM kontrolieri.

Pateicoties efektīva koda izveidei tika atrisināta laika aiztures problēma bez .sdc faila (laika aizturu ierobežošanas fails) esošajiem sistēmas nosacījumiem.

Izveidotais VGA kontrolieris pienākošos datus (16 biti) sadala pa 8 biti un attēlo uz monitora. Šāda pieeja ļauj izmantot ASRAM efektīvāk, jo vienā adresē var ierakstīt 16 bitus. Pirmajā dizainā dati atmiņā tika ierakstīti pa 8 biti uz vienu adresi, tādējādi atstājot neizmantotus vēl 8 bitus. Tagad ir iespēja vienā adresē glabāt divas pikseļu vērtības, kur katru pikseli var aprakstīt ar 8 bitiem – robeža, ko atļauj DE2- 115 attīstības rīka VGA.

Izveidotais attēlu sensora kontrolieris formē datu paketes ierakstīšanai atmiņā. Pienākošie dati no attēlu sensora ir 10 biti, bet reāli tiek izmantoti 8 nozīmīgākie. Kontrolieris divu taktu laikā saformē vienu datu paketi, kura sastāv no 2x8 bitiem, respektīvi – viena pakete = divi pikseli. Tālāk pakete(16 biti) tiek nosūtīta uz atmiņu.

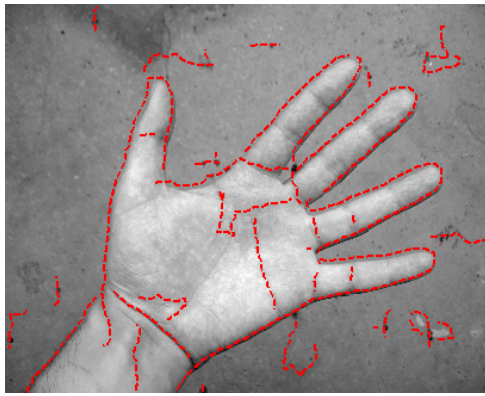
Turpmākie plāni:

1. Implementēt kompleksi salāgotu filtru (CMF) FPGA
2. Testi ar izveidoto filtru
3. Papildus funkcionalitātes, lai FPGA varētu kontrolēt no datora
 - 3.1. gaismas diožu kontrole
 - 3.2. oriģinālu un filtrētu attēlu ierakstīšana, izvadīšana un nosūtīšana uz datora
 - 3.3. mehāniskā filtra kontrole
4. Datora programmnodrošinājums priekš FPGA
5. Iespiedplates komplektācijas pabeigšana

Plaukstas detektēšana

Lai attēlā detektētu plaukstu, tad ar attēlu apstrādes palīdzību ir jāizdala plaukstai specifiskās pazīmes. Par pazīmēm tika izvēlēti plaukstas pirkstiem līdzīgas struktūras — to atrašanās vietas, pagrieziens leņķi. Katru plaukstas pirkstu var aprakstīt ar divu līniju, kas ir tuvu paralēlām un atrodas viena no otras noteiktajā attālumā, palīdzību.

Šādu struktūru izdalīšanai attēlā nav nepieciešams izmantot sarežģītus attēlu apstrādes paņēmienus — attēlā izdala kontūrus (piem. ar Kanī detektoru vai komplekso salāgotu filtru, jeb CMF) un izdalītās līnijas vektorizē, t. i. saglabā tos kā atsevišķus nogriežņus. Katram šādam nogriežnim saglabā 4 koordinātes: x_{start} , y_{start} , x_{end} , y_{end} . Vidēji attēlā izdalās ap 100 šādu nogriežņu.



a) CMF un segmentācijas rezultāts

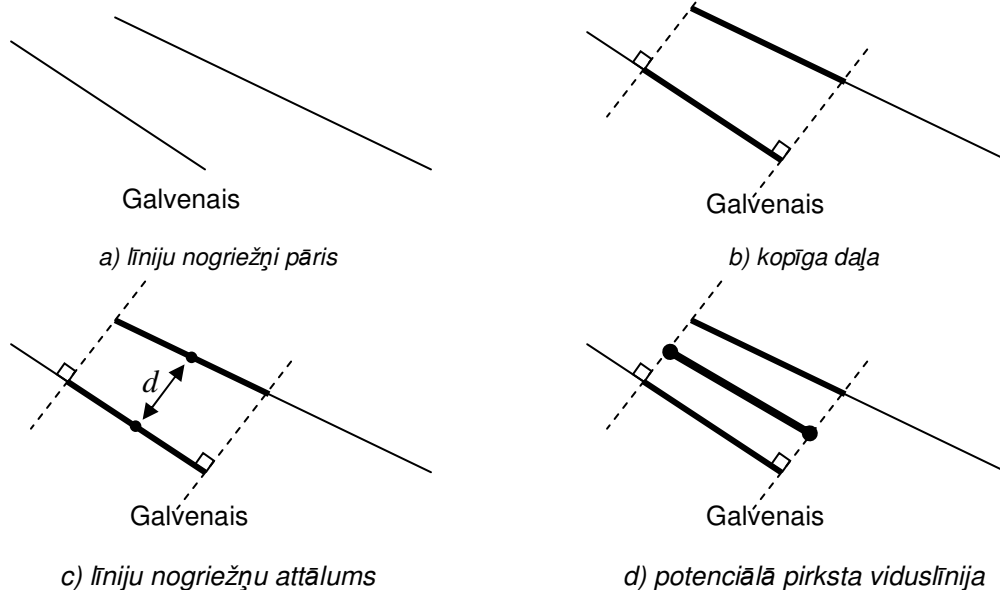


b) Līniju nogriežņu vektorizācijas rezultāts

Att. 13. Kontūra izdalīšana un vektorizēšana

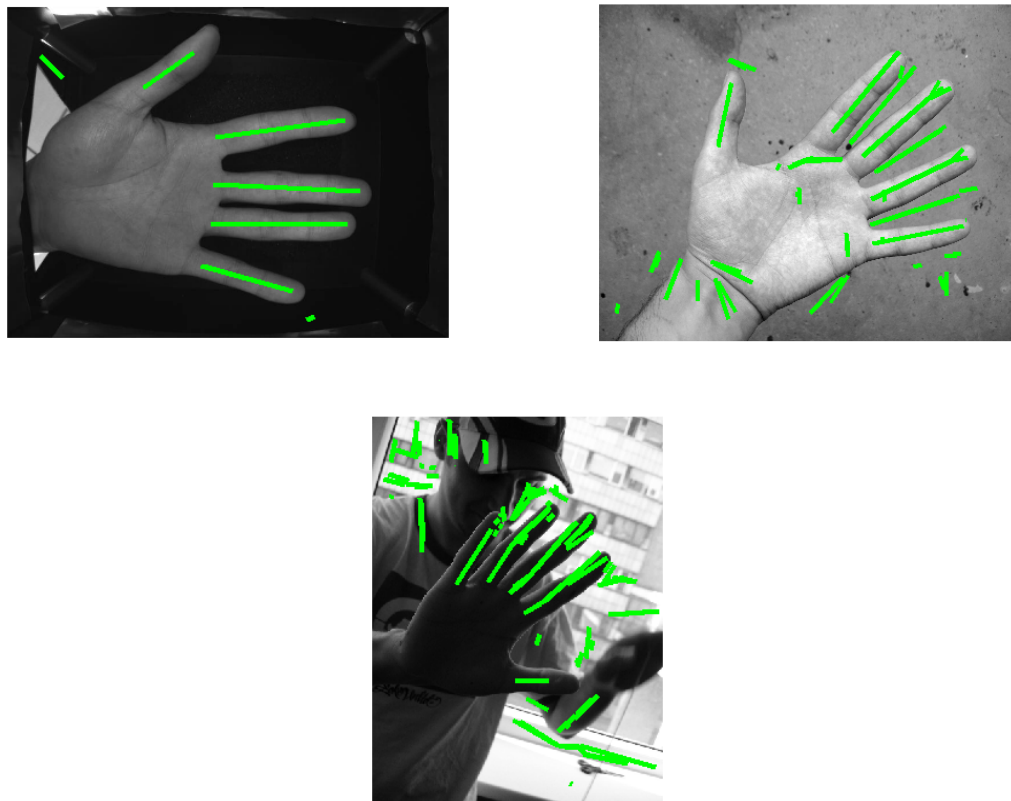
Strādājot ar vektorizētiem līniju nogriežņiem, paralēlo līniju meklēšana reducējās uz sekojošu algoritmu:

1. Izvēlās nogriežņu pāri (pavisam kopā šādu pāru sanāk ap 100·100): viens nogrieznis tiek paņemts par galveno, pret kuru tiks veiktas darbības ar otru nogriezni (att. 14a);
2. Ja nogriežņu orientācijas leņķi atšķiras vairāk par noteiktu $\Delta\varphi$, tad šo pāri tālāk neapskatās (šī operācija reducē apstrādājamo pāru skaitu desmitos-simtos reižu);
3. Vienu nogriezni projicē uz otras (uz galvenās) un izrēķina kopīgo daļu, jeb abiem nogriežņiem atstāj tikai tās daļas, kuras abas vienlaicīgi krustotu pirmajam nogriežnim perpendikulāri novilkta taisnes (att. 14b);
4. Nogriežņu attālumu nosaka kā attālumu starp atrasto daļu viduspunktiem (att. 14c). Ja šis attālums nav noteiktajā vērtību diapazonā $< d_{\max}$ un $> d_{\min}$, tad šo nogriežņu pāri vairāk neapskata;
5. Kopīgo daļu vidū šiem nogriežņiem uzzīmē potenciālo pirkstu (att. 14d), kurš tiks novērtēts tālāk, klasifikatorā.



Att. 14. Pirkstiem līdzīgu kontūru izdalīšana

Veicot testus ar izveidotu Matlab programmu tika konstatēts, ka algoritms ir spējīgs detektēt plaukstas pirkstiem līdzīgas struktūras pat ļoti zemas kvalitātes attēlos (ar raibu fonu, piem. att. 15). Taču turpmākie algoritma darbības uzlabojumi, kā arī optimizācija ir nepieciešami.



Att. 15. Plaukstas pirkstu detektēšanas rezultāti

Turpmākie plāni:

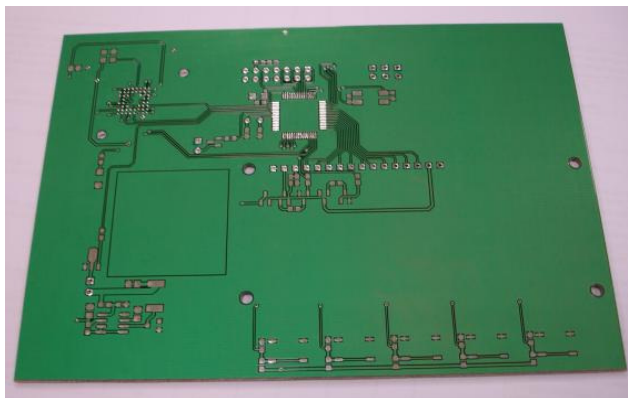
1. Citu plaukstai raksturīgo, pazīmju meklēšana un izdalīšana; klasifikators
2. Esošo algoritmu optimizācija un implementācija FPGA

Biometrisku datu apstrādes algoritmu darbības izvērtēšana

Centrālais virziens, kas tiek attīstīts projekta ietvaros sejas atpazīšanas jomā ir iegultas sistēmas izveidošana un algoritmiskās bāzes uzlabošana. Tika turpināts arī darbs pie virtuālā instrumenta pilnveidošanas un tika apskatīti jaunākie sejas atpazīšanas algoritmi.

Uz doto brīdi tiek veidots automātisks sejas atpazīšanas algoritms firmas Texas Instruments signālprocesoram TMS320C6416. Tas ir pirmais solis iegultas sejas atpazīšanas sistēmas izveidošanas virzienā, bet ieejas attēls tiek saņemts no datora caur RTDX (Real Time Data Exchange) interfeisu. Šis fakts padara sistēmu atkarīgu no datora. Tāpēc tiek uzsākts darbs pie kameras moduļa izstrādes priekš TMS320C6416 DSK platformas. Šis darbs tiek veikts sadarbībā ar citiem projekta dalībniekiem.

Centrālais kameras moduļa elements ir attēlu sensors MT9V024. Modulis ietvers arī sekojošus blokus: mikrokontrolieris; LCD displejs; jaudīgs LED elements un tā barošanas ķēde. Mikrokontrolieris paredzēts kameras moduļa bloku kontrolei. LCD displeju tiek plānots izmantot atpazīšanas rezultāta izvadīšanai un citas svarīgas informācijas par sistēmas darbību attēlošanai. LED elements strādās IR diapazonā, kas ļaus notestēt sejas atpazīšanas algoritmus uz bildēm infrasarkanajā diapazonā, kā arī iespējams mazinās sistēmas jutību pret mainīgo apkārtējo apgaismojumu. Šī virziena ietvaros ir izstrādāta iespiedplate, kura attēlota zemāk.



Att. 16. Kameras moduļa iespiedplate.

Turpmākie plāni:

- Kameras moduļa pabeigšana, t.s., visu elementu lodēšana un testēšana
- Programmatūras uzrakstīšana mikrokontrolierim (komunikācijas nodrošināšanai starp moduļa blokiem) un signālprocesoram (attēla iegūšanai no kameras)
- Sejas atpazīšanas sistēmas precizitātes testēšana IR diapazonā
- Virtuālā instrumenta pilnveidošana, papildināšana, uzlabošana

Biometrisko datu kriptēšanas, glabāšanas un lietojama apakšsistēmas izveide

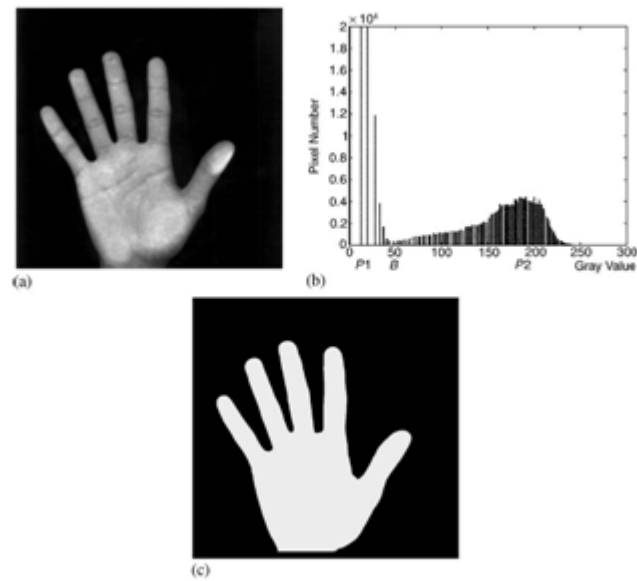
Atskaides periodā tika izpētītas ROI automātiskās noteikšanas metodes. Sastādīts ROI noteikšanas algoritms, kā arī piedāvāti dažādi plaukstu un rokas pirkstu ģeometriskos parametru mērījumi, kas varētu tikt izmantoti personas identificēšanai.

ROI noteikšanai visefektīvākais šķiet algoritms, kas secīgi izpilda sekojošus soļus:

1.solis. Plaukstu attēla iegūšana ar kādu no paņēmieniem, kas aprakstīti iepriekšējā atskaitē.

2.solis. Plaukstu attēla histogrammas izveide un sekojoša attēla binarizācija. Attēla iegūšanas iekārtas (skenera vai kameras) trokšņi var tikt samazināti attēla turpmākās apstrādes laikā, piemēram, ar mediānu filtru.

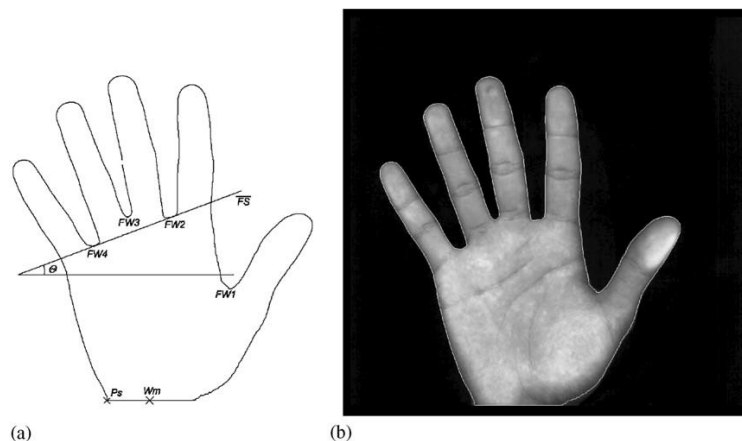
Binarizācija



Att. 17. Binarizācijas process

Attēla binarizācijai, tiek noteikta sliekšņa intensitāte B . Tad visi pikseli, kuru intensitāte $I(x,y) > B$ tiek pārkrāsoti baltā krāsā, bet pikseli, kuru intensitāte $I(x,y) < B$ tiek pārkrāsoti melnā krāsā. Binarizācijas sliekšnis B tiek noteikts kā lokāls minimums starp diviem lokāliem maksimumiem (P_1 un P_2) histogrammā (Att.17b). Lai paaugstinātu iegūtā attēla kvalitāti, ir vēlams attīrīt to no iespējamiem «atkritumiem», šim nolūkam izmantojams speciāls algoritms, kas nodzēš (pārkrāso baltus) nelielus melno pikseļu grupējumus. Apstrādes rezultātā iegūstam melnbaltu plaukstas attēlu (Att.17c).

3.solis. Lai noteiktu robežpikseļu kopumu, var izmantot speciālu trasēšanas algoritmu, kurš nosaka saistīto pikseļu koordinātes, piemēram, balto, kuriem kaut viens no līdzās esošiem pikseliem ir melns. Pašu zemāko pikseli, pirmo no kreisās puses, P_s (Att.18a) var izmantot kā apgaitas starta punktu. Trasēšanas algoritma darbības piemērs parādīts Att.18b. No pikseliem, kas atrodas uz robežas, formējas t.s. robežpikseļu vektors (border pixel vector - BPV).



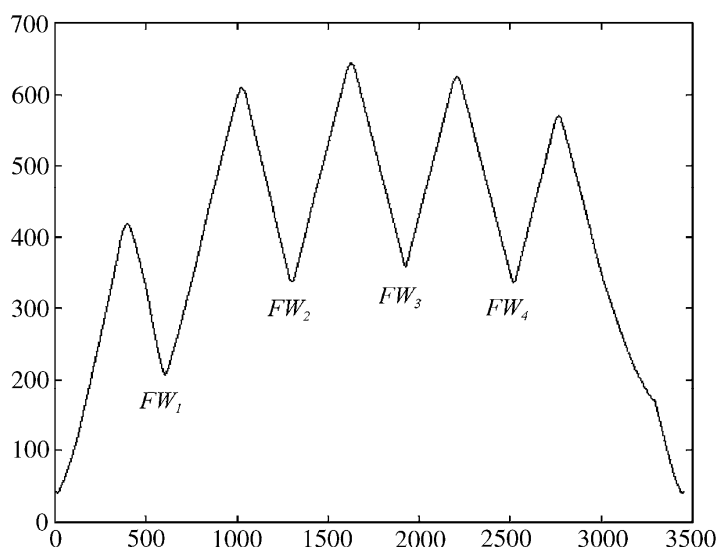
Att.18. Trasēšanas algoritma darbības princips

4.solis. Šī soļa uzdevums: noteikt pikseļu koordinātes, kas atrodas starp otro un trešo pirkstu, FW2 (šeit par pirmo tiek uzskatīts lielais pirksts) un starp ceturto un piekto pirkstu, FW4 (Att.2a). Tieši šie punkti tiks tālāk izmantoti ROI izveidei. Tika izpētītas dažādas pikseļu koordinātu FW2 un FW4, kā arī ROI formas un izmēra noteikšanas metodes [1, 2]. Rezultātā tika izvēlēta metode, kas balstās uz Eiklīda distances (Euclidean distance), starp atbalsta punktu un BPV pikseliem, izmantošanu. Viens no šīs metodes īpaši svarīgiem momentiem ir atbalsta punkta (AP) izvēle. Tika izanalizēti dažādi punkti (plaukstar centrā, plaukstar pamata galējie punkti un citi). Salīdzināšanas rezultātā, optimālākais izrādījās punkts W_m (Att.18a), kas atrodas plaukstar attēla apakšējās robežas vidū. Tā priekšrocības pirmkārt, to viegli noteikt, otrkārt, tas atrodas, gandrīz maksimālajā attālumā no punktiem FW2 un FW4, kas nodrošina mazāku mērījumu kļūdu. Izmantojot izvēlēto AP (W_m), var noteikt Eiklīda distanci $D_E(i)$ no W_m līdz katram i -tajam BPV pikselim:

$$D_E(i) = \sqrt{(X_{wm} - X_{BPV(i)})^2 + (Y_{wm} - Y_{BPV(i)})^2}$$

kur X_{wm}, Y_{wm} - AP koordinātes, $X_{BPV(i)}, Y_{BPV(i)}$ - BPV i -tā pikseļa koordinātes.

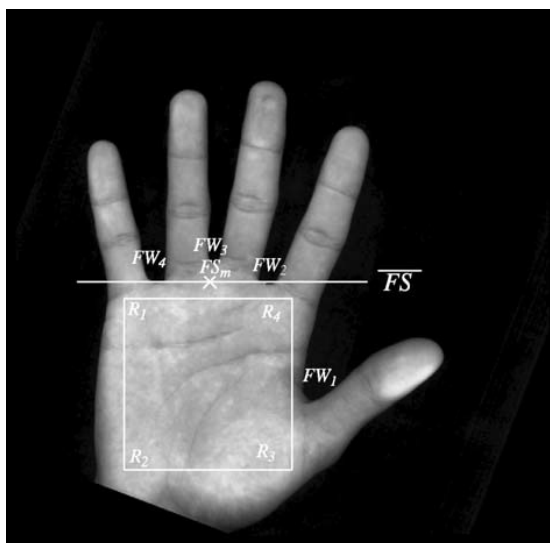
No šādā veidā aprēķinātajiem $D_E(i)$ var izveidot Eiklīda distanču sadalījuma diagrammu Att.19.



Att. 19. Eiklīda distanču sadalījuma diagramma

Diagrammas lokālie minimumi (FW1, FW2, FW3, FW4) atbilst plaukstar punktiem starp pirkstiem. Pikseļu koordinātes FW2 un FW4 ir meklētie dati, kas tiek izmantoti, lai izveidotu kvadrātveida zonu. Starp diviem pikseliem (punktiem) FW2 un FW4 tiek novilkta taisne. Viduspunktu uz šīs taisnes, starp FW2 un FW4, apzīmēsim FS_m , (Att.20). Tālāk no punkta FW3 novelkam perpendikulu pret līniju FS. Uz perpendikula, no perpendikula krustpunkta ar FS, uz plaukstar pusi, nomērām attālumu, vienādu ar attālumu starp FW3 un līniju FS (Att.20). Rezultātā iegūtais punkts ir pirmais ROI punkts. Zinot vismaz viena ROI punkta koordinātes, var piedāvāt vairākus ROI formu variantus, kā arī to izmērus. Var noteikt vismaz

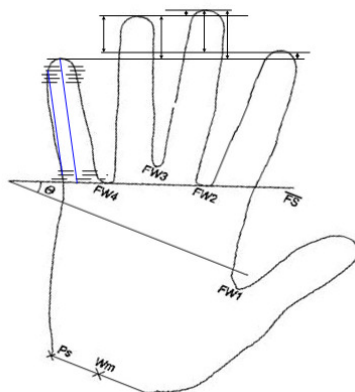
trīs ROI formas: kvadrātveida, taisnstūrveida vai apaļu (pēc plaukstā ievilktais līnijas). ROI izmēru var noteikt ar fiksētām vērtībām, piemēram, kvadrāts, ar izmēru 300x300 pikseļu, vai ar relatīvām, kuras tiek noteiktas no citiem pirkstu un/vai plaukstas izmēriem.



Att. 20. RIO punkta noteikšana

Šajā pētījumu etapā tiks apskatīta kvadrātveida ROI forma. Pieņemtas fiksēta garuma kvadrāta malas – 256 pikseļi.

Personas identificēšanai var tikt izmantoti arī dažādi pirkstu un plaukstas ģeometriskie izmēri. Tādi, kā pirkstu platums un garums, plaukstas platums un citi (Att.21). Jāatzīmē, ka ROI noteikšana ir tikai starpposms, pēc kura sekos plaukstas individuālo raksturlielumu noteikšana, piemēram, vēnu un/vai grumbu tekstūra un citi.



Att. 21. Personu identificēšanā izmantojami plaukstas ģeometriskie izmēri

ROI un citu ģeometrisku raksturlielumu noteikšanai izstrādātos algoritmus būs nepieciešams optimizēt arī pēc to izpildes laika, lai nodrošinātu identificēšanu reālā laikā.

Vēl šajā progresa periodā ir apskatīti vairāki biometrijas datu šifrēšanas algoritmi (fuzzy sketch, fuzzy vault, fuzzy extractor), kā arī daži kļūdu korigēšanas algoritmi (Heminga kods, Hadamarda kods, Rīda-Solomona kods). Ir sagatavotas nepieciešamās operācijas ar paplašinātiem galīgu lauku polinomiem $GF(2^m)$.

Lai pārbaudītu izveidotos algoritmus, tika izveidots lietotāju interfeiss, kurā sākotnēji bija iespējams atzīmēt svarīgos punktus, lai pēc tam varētu izgriezt ROI, izmantojot to informāciju. Pēc tam programmatūra tika papildināta ar kodu, kas bez ārējas palīdzības atrod visus nepieciešamos punktus, lai aprēķinātu ROI.

Tika izveidots lietotāju interfeiss, kas komunicē ar FPGA, lai būtu iespējams saņemt uzņemtos attēlus. Interfeiss satur arī nepieciešamo funkcionalitāti biometrijas paraugu datubāzes izveidei.

Turpmākie darbi:

- Visefektīvākā RIO formas un izmēra noteikšana;
- Pirkstu un plaukstu ģeometrisku izmēru, kas ir pietiekami unikāli (katrs atsevišķi vai to kombinācijas) katrai personai noteikšana
- Rīda-Solomona koda implementēšana, lai varētu pārbaudīt iespējamās uzlabojumus, kas tiktu iegūti izmantojot kļūdu korigējošos algoritmus.

Atsevišķu sistēmas komponentu un bloku izstrāde un izpēte

Šī ceturkšņa ietvaros svarīgākie veiktie darbi:

- Raksta sagatavošana;
- Izveidota programma plaukstu reģiona automātiskai detektēšanai;
- Uzsākts darbs pie patenta pieteikuma sagatavošanas;
- Izveidota programma priekš MSP430 F2013 mikrokontroliera I2C kopnes interfeiss ar Aptina attēlu sensoru;
- Eksperimenti ar Aptina attēlu sensoru.

Šajā laika periodā ir sagatavots zinātniskais raksts par projektā veiktajām aktivitātēm, kas saistītas ar biometrijas datu iegūšanu, apstrādi un šifrēšanu. Raksts „Biohashing and Fusion of Palmprint and Palm Blood Vein Biometric Data” iesniegts un apstiprināts prezentēšanai starptautiskā IEEE organizētā konferencē par plaukstu biometrisku parametru izmantošanu personu atpazīšanā (ICHB2011), Honkongā. Raksts pievienots pielikumā Nr II.

Lai veiktu automātisku plaukstu reģiona detektēšanu tika izveidota programma Matlab vidē, kas veic sekojošas darbības:

- aprēķina attēla histogrammu,
- izvēlas sliekšņa vērtību atkarībā no iegūtās histogrammas funkcijas,
- iegūst bināru attēlu un detektē plaukstu kontūru,
- no plaukstu centrālā punkta līdz atrastajiem kontūra punktiem tiek aprēķināts Eiklīda attālums un iegūta distances funkcija,

- apstrādājot distances funkciju tiek atrasti pirkstu galapunkti, kā arī pirkstu starpas,
- pēc atrastajiem punktiem tiek aprēķināts interesējošais reģions,
- tiek izdalīts atrastais reģions, kā arī saglabāts leņķis kādā šis reģions tika atrasts.

Programmas testi veikti izmantojot publiski pieejamo plaukstas attēlu datubāzi. Pielikumā Nr.III prezentācija, kurā ar attēliem parādīta izstrādātā algoritma darbība.

Izveidota programma, kas nodrošina MSP430 F2013 mikrokontroliera saziņu ar Aptina attēlu sensoru caur I²C datu kopni. Programma paredzēta attēlu sensora reģistru konfigurēšanai un uzstādījumu mainīšanai. Izmantojot šo programmu, veikti vairāki eksperimenti mainot attēlu sensora uzstādījumus. Pārbaudīts HDR (High Dinamic Range – Augsta dinamiskā diapazona režīms).

Nākošā perioda darbi:

- kameras moduļa testēšana, attēlu sensora reģistru konfigurēšana;
- plaukstas fiksējošā statīva izveide;
- personu plaukstas asinsvadu un rievojuma attēlu datubāzes izveidošana;
- plaukstas detektēšanai un apstrādei nepieciešamā algoritma izveide.

Secinājumi

Projekta „BiTe” pētnieciskā darbība „Lietišķo pētījumu” aktivitātē sekmīgi turpinās arī trešajā periodā. Noris izstrādājamās tehnoloģijas atsevišķu moduļu, algoritmu un elektrisko principiālo shēmu izpēte un izveide.

Attēlu sensora iespaidplates projektēšanai un attēlu iegūšanas sistēmai izmantojot FPGA tiek plānots implementēt kompleksi salāgoto filtru (CMF). Lai pārbaudītu sistēmas darbību, tiks veikti dažādi testi un izveidotas papildus funkcionalitātes, lai FPGA varētu kontrolēt no datora (t.s., gaismas diožu kontrole; oriģinālu un filtrētu attēlu ierakstīšana, izvadīšana un nosūtīšana uz datora; mehāniskā filtra kontrole).

Plaukstas detektēšanā turpmāk nepieciešams veikt darbus sekojošos virzienos: nepieciešams meklēt un izdalīt visas plaukstai raksturīgās pazīmes un noteikt klasifikatoru, bet esošos algoritmus jāoptimizē un jāimplementē FPGA.

Sejas atpazīšanas sistēmai tiek plānots pabeigt kameras moduli, kur nepieciešams pielodēt un notestēt visus moduļa elementus. Nepieciešams uzrakstīt atbilstošu programmatūru priekš mikrokontroliera (kas nodrošinātu komunikāciju starp moduļa blokiem) un signālprocesora (attēla iegūšanai no kameras). Papildus tiek plānots notestēt sejas atpazīšanas sistēmas precizitāti IR diapazonā.

Izanalizējot biometriskas informācijas kriptēšanas un salīdzināšanas metodes jānosaka visefektīvākās RIO formas un izmērus, nepieciešams noteikt katras personas unikālos plaukstas un pirkstu ģeometriskos izmērus (katru atsevišķi vai savstarpēji kombinējot) un jāimplementē Rīda-Solomona kods, lai varētu pārbaudīt iespējamās uzlabojumus, kas tiktu iegūti izmantojot kļūdu koriģējošos algoritmus.

Atsevišķu sistēmas komponentu un bloku izstrādei un izpētei nepieciešama plaukstas moduļa testēšana un attēlu sensora reģistru konfigurēšana. Plānots izveidot arī fiksējošu plaukstas statīvu, lai varētu sākt uzņemt plaukstas asinsvadu un rievojumu attēlus datubāzei. Tiks turpināts darbs pie plaukstas detektēšanas un apstrādes algoritma.

Projekta pētnieciskie rezultāti tiek apspriesti izpildītāju iknedēļas sanāksmēs. Šīs sanāksmes tiek veidotas, lai projekta izpildītāji kopīgi izdiskutētu par pētījumiem un problēmām, kas jārisina, lai apakšaktivitātes veiksmīgi attīstītos. Trešajā progresa pārskata periodā notika sanāksmes, kurās projekta izpildītāji sniedza dažādas prezentācijas: „Plaukstas ģeometrijas svarīgākie parametri, to aprēķināšanas metodes”, „Plaukstas kontūra detektēšana”, „Plaukstu ģeometriskie raksturojumi”.

Šajā laika periodā ir sagatavots zinātniskais raksts par projektā veiktajām aktivitātēm, kas saistītas ar biometrijas datu iegūšanu, apstrādi un šifrēšanu. Raksts „Biohashing and Fusion of Palmprint and Palm Blood Vein Biometric Data” iesniegts un apstiprināts prezentēšanai starptautiskā IEEE organizētā konferencē par plaukstas biometrisku parametru izmantošanu personu atpazīšanā (ICHB2011), Honkongā.

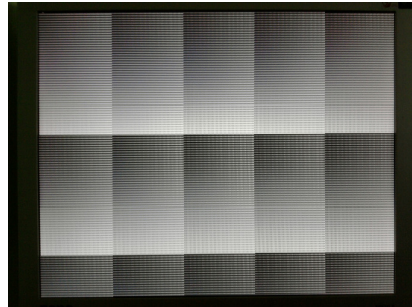
Ir uzsākts darbs pie patenta pieteikuma sagatavošanas.

Ir noteikti nākamā perioda pētniecisko darbu uzdevumi un sasniedzamie rezultāti. Par projekta pētnieciskiem sasniegumiem tiek plānots informēt arī piedaloties ar referātiem un publikācijās konferencēs un semināros.

Atsauces

1. Adams Kong, David Zhang, Mohamed Kamel. A survey of palmprint recognition - Pattern Recognition 42(2009)1408--1418
2. Feng Yue, Wangmeng Zuo, David Zhang, Kuanquan Wang. Orientation selection using modified FCM for competitive code-based palmprint recognition - Pattern Recognition 42 (2009) 2841--2849

Attēli, iegūti ar pattern ģeneratoru, kurš ar katru takti palielina vērtību (skaitītājs). Ģenerators saskaita līdz $640 \times 480 = 307\,200$ vērtībai un apstājas. VGA kontrolieris (8 biti) attēlo skaitītāja datus (16 biti).



Iegūtais attēls attēlo visas 307 200 vērtības. Intensitātes līmenis atbilst skaitītājam, kura 16 bitu dati tiek sadalīti divās paketēs pa 8 biti.



Attēls reprezentē datus no vienas adreses atmiņā. Iegūtais attēls sastāv no divu krāsu vertikālēm, kuras atkārtojas viena pēc otras. Tātad – VGA kontroliera uzdevums izpildīts – VGA attēlo pienākošos 16 bitus.



Pattern ģeneratorā tika ieviestas izmaiņas, datiem tika piešķirta konstanta vērtība:

Dati (15 down to 8) $\leq$ X„3A” - heksadecimālajā kodā

Dati (7 down to 0) $\leq$ X„3A” - heksadecimālajā kodā

Biohashing and Fusion of Palmprint and Palm Vein Biometric Data

Rihards Fuksis, Arturs Kadikis, and Modris Greitans

Institute of Electronics and Computer Science

14 Dzerbenes Str., Riga, LV1006, Latvia

Email: {Rihards.Fuksis; Arturs.Kadikis; Modris.Greitans}@edi.lv

Telephone: (371) 67558238

Fax: (371) 67555337

<http://www.edi.lv>

Abstract—This paper combines the results of our previous experiments concerning acquisition of the images of the human palm in infrared and visible light, the extraction of features from images as well as our current results on biometric data hashing with the advanced biohashing algorithm. We first describe the properties of the complex 2D matched filtering for feature extraction from images, followed by biometric vector construction techniques and raw biometric data comparison. We address the problem of securing biometric data for multimodal biometric systems, by analyzing the biohashing algorithm and proposing our enhancements. Results of experiments that include raw biometric data comparison, biohashing and advanced biohashing biocode comparisons are presented at the end of the paper.

I. INTRODUCTION

Multimodal biometric systems use two or more biometric parameters (e.g., fingerprint, face, iris, etc.) to increase the overall system's performance. However, along with accuracy it is also necessary to ensure a convenient enrollment procedure of the person. Therefore, it is necessary to select biometric parameters that are unique and easy to present. A number of different research activities on biometric parameter fusion have been presented in recent years, such as palm geometry and palmprint fusion [1], [2], palm print and face [3], finger vein and finger-dorsa texture fusion [4], multispectral hand biometrics [5] and palmprint and palm vein image fusion applied at the image level [6].

In this paper, we use two palm biometric features - palmprints and palm veins. The most significant problems associated with biometric systems are:

- Efficient acquisition of biometric parameters - the main biometric features should not be distorted; the acquisition system must recognize false features (image);
- Effective extraction of unique information from the acquired biometric data - the acquired images usually contain a lot of unwanted data or data that give no information for biometric matching;
- Biometric data security - how to generate a unique code from the biometric data so that real biometric data is not revealed;
- Biometric data comparison - how to compare two encrypted biometric vectors.

By using the palmprint and palm vein images, we have tried to address all of the four above-mentioned issues. Firstly, it is easy to present a palm to a biometric system. Also, it is hard to falsify a palmprint and palm vein pattern if the image of palm is taken in two different wavelengths within hundred of milliseconds. Secondly, we extract the predefined features from the images by using a complex 2D matched filter and construct the biometric data vector, thus reducing the amount of data. Thirdly, we use the extracted data to create a data vector and apply several enhancements of the biohashing algorithm to obtain a biocode. Fourthly, we introduce a biocode comparison method. All of these steps will be discussed further in this paper.

II. IMAGE ACQUISITION

It is important to acquire images of the palm so that the significant features that are used for person recognition are clearly visible. For palmprints, these are skin wrinkles, for palm veins - visible veins. The capturing of the palm images is performed by using only one image sensor, but dual spectrum illumination. Palm vein images are captured in the near infrared (NIR) spectrum, but images of the palmprint structure are captured in the visible light spectrum. Most of the popular methods of palm vein imaging are based on the infrared (IR) light absorption properties of the blood. Hemoglobin absorbs light of 760nm and 850nm depending on oxygen concentration; illuminating a palm with IR light, veins appear darker in the image than the rest of the palm because part of the IR light is absorbed in places where there are veins, and the rest is reflected from the surrounding tissue. Images taken in visible and IR light are shown in Fig. 1.

III. BIOMETRIC FEATURE EXTRACTION FROM IMAGES

After the images of the palmprint and the palm veins are captured, they must be processed in order to extract the chosen features (wrinkles from palmprint image and veins from the palm vein image). Conventional image processing methods such as histogram equalization or global thresholding [7] are not acceptable due to the irregular intensity and noisy background of the images. A cross section of the captured palm vein and palm print images (Fig.2) reveals that the

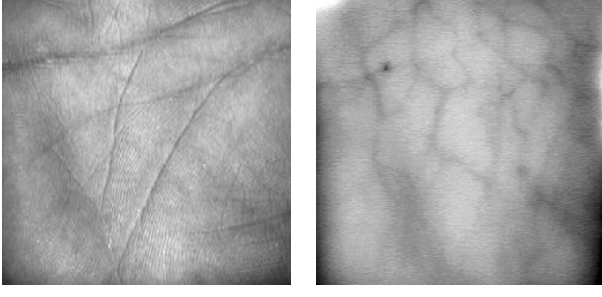


Fig. 1. One person's palm print (left) and palm vein (right) images

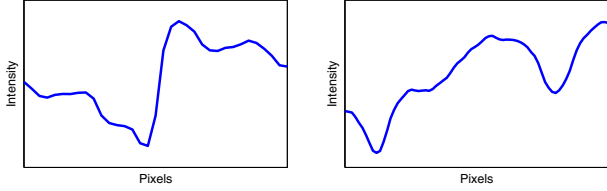


Fig. 2. Cross section of a palm print ridge (left) and two palm blood vessels (right)

corresponding features are different, as a result of which two different methods have to be used to extract the desired features.

A. Palmprint Feature Extraction

Various representations have been proposed for palmprint recognition. Those include, representation by lines [8], points [9], [10], Fourier spectrum [11], morphological representation [12], texture [13], wavelets [14] and complex wavelets [15], Gabor filters [16], fusion code [17], ordinal code [18] and representation with so called "Laplacianpalm" [6]. As shown in the Fig. 2 (left), the palm print ridge has a sharp intensity change. This sharp intensity change can be detected using the first derivatives which are implemented using the magnitude of the gradient. For a function $g[x, y]$, the gradient of g at coordinates $[x, y]$ is defined as the two-dimensional column vector [7]

$$\nabla g \equiv \text{grad}(g) = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} \quad (1)$$

Derivatives of discrete functions in a point $[x_0, y_0]$ are calculated as the difference of nearest pixel values, for example:

$$\left. \frac{\partial g[x, y]}{\partial x} \right|_{[x_0, y_0]} = g[x_0 + d, y_0] - g[x_0 - d, y_0] \quad (2)$$

where d is the half of the distance between taken neighbourhood pixels.

Typically the image is corrupted with noise and the direct application of (2) can lead to an incorrect result. Therefore, before the derivatives are calculated, the image $f[x, y]$ is smoothed using a Gaussian filter to reduce rapid intensity spikes that are initiated by noise.

$$g[x, y] = f[x, y] \otimes e^{-\frac{x^2 + y^2}{\sigma^2}} \quad (3)$$

where σ specifies the smoothing rate, and $\otimes$ is the convolution operator.

Function's gradient vector $\text{grad}(g)$ points at the greatest rate of intensity change of g , which corresponds to the cross section of the skin ridge. To facilitate the following stage of vector set construction and acquire the resulting matrix of vectors F_1 , the gradient vector is rotated by 90° . Since the method used to extract the blood vessels is expressed in the complex form, for convenience, we rewrite (2) using the complex notation as:

$$\underline{E}_1[x_0, y_0] = (g[x_0, y_0 - d] - g[x_0, y_0 + d]) + j(g[x_0 + d, y_0] - g[x_0 - d, y_0]) \quad (4)$$

B. Palm Vein Feature Extraction

In our previous papers [19], [20], we have introduced a method for line-like object extraction that is based on the matched filtering (MF) approach [21] and is called Complex 2D Matched Filtering (CMF) [19]. CMF performs faster than MF because only two convolution operations with the CMF kernel are required. CMF also extracts the selected features from the images and provides additional information about the direction of the extracted features of the image. The nature of CMF can be explained by looking at the principles of Matched Filtering with the Gaussian 2D function $G(x, y)$ [21].

$$G(x, y) = \begin{cases} -\exp\left(-\frac{y^2}{\sigma^2}\right), & |x| \leq \frac{D}{2} \\ 0, & |x| > \frac{D}{2} \end{cases} \quad (5)$$

where D is the length of the filter kernel in direction x . In order to detect blood vessels, the filter mask must be scaled and rotated in different directions. Further in the paper the rotated and scaled Gaussian 2D kernel is referred to as:

$$G[x, y; \phi, c] \equiv G\left(\frac{x \cos \phi - y \sin \phi}{c}, \frac{x \sin \phi + y \cos \phi}{c}\right) \quad (6)$$

where c is the scaling factor and ϕ is the rotation angle. The more rotation angles are used the more accurate is the extraction of blood vessels. However, this involves image convolution with the rotated Gaussian kernels and, therefore, is computationally inefficient for embedded systems. Eight differently rotated masks for MF are shown in Fig.3.



Fig. 3. Eight differently rotated MF masks for vein extraction

To simplify the computational complexity of the MF approach, we use the CMF approach. This image processing method is slightly different from the MF approach. First of all, the filtering is done with one complex mask, instead of many rotated masks. This method obtains additional information about the analyzed feature orientation, in our case about the blood vessels and skin wrinkles. The output of the filtering procedure is a matrix of vectors that is of the same size as the input image. Vectors represent the correlation with the

previously defined mask representing the objects that have to be found. Instead of consecutive filtering with several differently oriented MF masks, CMF filters image only with one complex mask, which incorporates all the angles and scales. The kernel of the complex matched filter is defined by the following expression:

$$\underline{M}[x, y] = \sum_{n=1}^N \sum_{l=0}^{L-1} \exp(j2\phi_l) G[x, y; \phi_l, c_n], \quad (7)$$

where N – total number of used scales,
 L – total number of used angles,
 $\phi_l = \frac{l}{L} \cdot \pi$.

The image is filtered with the CMF kernel:

$$\underline{C}[x, y] = f[x, y] \otimes \underline{M}[x, y]. \quad (8)$$

Additional operation of the angle decrement is performed to $\underline{C}$ to acquire the final CMF result:

$$\underline{F}_2[x_0, y_0] = |\underline{C}[x_0, y_0]| \exp\left(j \frac{\text{Arg} \underline{C}[x_0, y_0]}{2}\right). \quad (9)$$

The magnitude of the vectors represents the congruence between the filter mask and the object in the particular pixel of the image. The angle of the vector shows the orientation in which the maximum of $\underline{C}$ is found. This information is crucial in the segmentation and recognition stage.

C. Feature Vector Construction

Some of the information obtained in previous stages can be discarded, because most of it represents undesired regions of the palm where palmprint and vein features are not present. The goal is to extract most significant vectors that represent the blood vessels or ridges of the skin. The following procedure is repeated several times until R most significant vectors are found: the most significant vector from the processed image is extracted, and the region around each vector found is excluded from further processing to avoid multiple vector extraction from the same region.

D. Vector Set Comparison

After the vector set A of the most significant features is acquired, the recognition process can begin. First, vector set comparison without any encryption (using raw biometric data) is performed. Vector set A is compared with the database vector sets B_n to find the best match. To compare two sets of vectors, an approach similar to [20] can be used: each vector $v_p(A)$ from the first set A is compared with each vector $v_q(B)$ from the second set B . The similarity of two vectors is evaluated by positive $s_{p,q}$ and is a product of three terms:

$$s_{p,q} = \text{magnitudes}_{p,q} \cdot \text{angles}_{p,q} \cdot \text{distance}_{p,q} \quad (10)$$

These three parts evaluate the positions of two vectors - distance:

$$\text{distance} = \exp\left(-\frac{d_{\parallel}^2}{\sigma_{\parallel}^2}\right) \cdot \exp\left(-\frac{d_{\perp}^2}{\sigma_{\perp}^2}\right), \quad (11)$$

the angular difference - angles:

$$\text{angles} = |\cos \angle(v_p(A) \cdot v_q(B))|, \quad (12)$$

and the significance of these vectors - magnitudes:

$$\text{magnitudes} = |v_p(A)| \cdot |v_q(B)|. \quad (13)$$

All the similarities of a particular pair of vectors $s_{p,q}$ are summed together to evaluate the similarity of the vector sets in general:

$$s(A, B) = \sum_p \sum_q s_{p,q} \quad (14)$$

The higher is the value of a particular $s_{p,q}$, the higher is their influence on the overall similarity value $s(A, B)$. The vector magnitude is proportional to the significance of the locally extracted feature that is represented by this vector. For this reason, magnitudes (13) are included in the similarity evaluation (10). Our experiments showed that, as a result of changes in the lightning conditions in the image acquisition stage, the same line-like objects appear differently on the filtered images: the positions of local maximums across the object vary. For this reason, we split the distance between vectors into parallel (to $v_p(A)$) and perpendicular parts, and evaluate them separately: the first mentioned is less critical for $s_{p,q}$ than the second one ($\sigma_{\parallel} > \sigma_{\perp}$). Both parts of the distance are found as the projections of the actual distance between $v_p(A)$ and $v_q(B)$ onto the vector $v_p(A)$. The similarity value $s(A, B)$ is influenced by the image contrast and the neighbourhood effect of many local maximums representing one and the same line-like object jointly compared with each other. In the evaluation stage it is normalized as described in [20].

$$S(A, B) = \frac{s(A, B)}{\sqrt{s(A, A) \cdot s(B, B)}} \quad (15)$$

The similarity index value $S(A, B)$ lies in the interval of $[0; 1]$ and is used for evaluation of similarity of two images. The similarity index doesn't have commutative properties, $S(A, B) \neq S(B, A)$.

However it is not safe to use raw biometric data. If the data is intercepted then the person's biometric information is stolen and cannot be changed. Therefore, in the next section, we look at the biometric data encryption method BioHash, and introduce enhancements of the encryption algorithm.

IV. THE BIOHASH ALGORITHM

Many methods that integrate biometrics with cryptography have been proposed. As an example, the fuzzy vault approach that binds a key to the biometric template [22] and a fuzzy extractors, that can be used to extract keys from a biometric template [23].

In this paper, we chose to encrypt the biometric data with a one-way hash function called biohash [24], [25]. It is a type of cancelable biometrics that has been introduced in [26], and it consists of an intentional, repeatable distortion of the original biometric template, satisfying the following properties:

- Renewability: it should be possible to revoke a compromised template and issue a new one based on the same biometric data (revocability);
- Security: the original biometric data cannot be recovered;
- Performance: the recognition performance of the protected system, in terms of Equal Error Rate (EER), should not degrade significantly with respect to a non-protected system.

Biohashing is a method that combines orthonormal random vectors with user specific biometric data. A token is a sequence of numbers or symbols that are used as a seed to generate random vectors. Each user can have its own private token or every person can use the public token.

The sequence of steps needed for biohash is the following:

- 1) acquisition of the biometric data and extraction of the most significant features to create a data vector;
- 2) usage of the token as a seed for the random number generator. Generating the random vectors and applying Gram-Schmidt vector orthonormalization process;
- 3) calculation of the dot product from each orthonormal vector and the biometric feature vector;
- 4) thresholding of the result by some value T .

After these steps, the result is a series of bits, or biocode that can be used in further authentication.

The enrolment and recognition stages are similar. Both follow the previously described sequence of steps, but in the enrolment stage, multiple biometric samples could be used to extract more precise features. The similarity H_b of two biocode samples is calculated by using Hamming distance D_h of those samples and the length of the feature vector l (16).

$$H_b = \frac{l - D_h}{l} \quad (16)$$

A. Basic Biohash Algorithm

The biohashing algorithm performs computations on the biometric data vector and creates a biocode. As previously explained, biometric data vector is created using the features extracted from the palmprint or palm vein images.

The extracted features are the R most significant vectors from the processed image. Each vector is described by 4 components: vector's starting position coordinates (u, v) and coordinates (du, dv) that represent vector's orientation and magnitude.

The data vector is constructed by putting the components of each vector in order of succession. The sequence starts with the coordinates of the first vector u_1, v_1, du_1, dv_1 , then the second vector and so on. The feature vector consists of $4R$ components (Fig. 4). Therefore, $4R$ orthonormal vectors will be created, and the resulting biocode will contain $4R$ bits.



Fig. 4. Biometric data vector

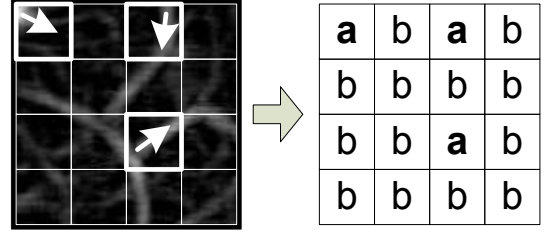


Fig. 5. Extraction of three most intensive vectors

B. Improved Biohash Algorithm

We have taken into account several observations and propose to use additional user specific information to improve the performance of the biohash algorithm. The improvements are achieved by using the statistical properties of the biometric data and the resulting biocodes.

In the captured palm images, some features (veins or ridges) are larger than others. As observed, environmental lighting and temperature will not affect the extraction of the main (larger) features. Therefore, one of the improvements of the algorithm presented in this paper is the use of additional information about the most intensive vectors that is added to the data vector. As previously described, most intensive vectors are found as follows: vector of maximal magnitude is found and the region with the size of o is excluded around it. The size of o defines how large is the region that represents the biometric feature. A limit on how many most intensive vectors need to be found can be set. Each cell of the matrix with the size of $k \times k$ describes the presence of at least one of the most intensive vectors w in the certain region of the image; value a means that the region contains the vectors, value b means that no vectors were found in that region. This process is shown in Fig. 5, where $k = 4$ and $w = 3$. Information about the most intensive vector positions is added to the biometric data vector. If the following method is used to describe the most intensive vectors in the sample, then, even if the order of vectors changes, the added information will be the same.

By analyzing the biocodes obtained, it was observed that they are not identical for the same person. This can be explained by the variable nature of biometric data. But some bits between one person's biocodes vary more than the others (Fig. 6). This bit variation further will be called bit stability. Bits that vary less are more stable ("4" in Fig. 6) than bits that vary more ("2" in Fig. 6).

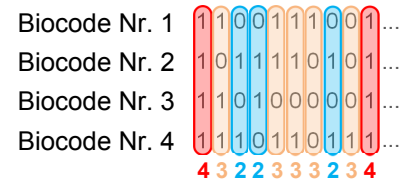


Fig. 6. Calculating the largest amount of equally valued bits

The stable bits of all biocodes are user specific. It is possible

to improve the method of biocode comparison by using information about the stable bits. For a more precise biocode comparison the errors of the more stable bits should be more emphasized than the errors of the unstable bits. Therefore, bits are arranged in descending order by bit stability. Training samples are used to obtain this information. Depending on the amount of samples used to sort the bits, more precise statistical information about the user can be obtained. After multiple biocode samples are collected, the bit stability is calculated (Fig. 6). First, the bits are sorted by the largest amount of equally valued bits between biocodes for a specific bit (Fig. 7 transform (a)). After that, the bits in each of the groups are sorted again, but this time by their total distance of equally valued bits from the threshold value T (Fig. 7 transform (b)). If the distance from the threshold T is larger, the bit is more stable than one which is closer to the threshold level.

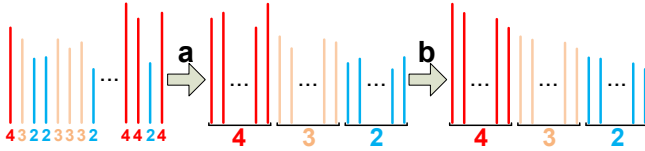


Fig. 7. Sorting the bits by their stability

After all these operations, a sequence of bit indexes b_i is obtained that allows comparison between two biocodes using specific weight coefficients that represent information about bit stability. i is the index of the bit in the original biocode (Fig. 7 before transform (a)), b_i is the position of the bit i in the sorted array (Fig. 7 after transform (b)), where the bits are sorted by their stability. This information does not reveal what values the specific bits should have, and it does not reveal the specific amount of bit stability. It only reveals which bits are more stable relatively to each other.

If Hamming distance is used to compare two biocodes, each difference between bits makes an equal contribution to the overall error. With the proposed comparison method the stable bits have a greater influence on the overall error if they differ. To calculate the error err_i for each bit an exponential function (17) is used. The value p describes how many of the bits are used in the comparison.

$$err_i = e^{-\frac{b_i^2}{p}} \quad (17)$$

V. EXPERIMENTAL RESULTS

To evaluate the previously described methods, a database of palmprint and palm vein images was used. This database consists of 250 images of each image type, containing 5 images of 50 different persons - a total of 500 images. First, image processing techniques, like CMF and gradient filtering are applied to each of the database image to acquire the most significant vector set. Next, the database images are mutually compared without using biocode. The results for non-hashed comparison can be seen in Table I.

TABLE I
RAW BIOMETRIC DATA COMPARISON RESULTS

	Veins	Palmprints	Fused data
EER	0.32	2.79	0.1

A crucial part of the bihashing algorithm is the tokens. As previously mentioned, a token is used as a seed for the random number generator to generate random vectors. Each set of generated random vectors is unique and with specific properties. Therefore, tests with different tokens must be employed. We performed 100 different tests, where each test employs its own, randomly generated token, and all persons in the database have the same token. As a result, from each test, the EER is calculated. The results including the mean value and the standard deviation of all EER can be seen in Table II.

TABLE II
BIOHASH ALGORITHM TEST RESULTS (EER)

	Veins	Palmprints	Fused data
Mean	14,043	12,073	6,190
StDev	1,152	1,102	0,803

In the advanced algorithm, the information about the most intensive vectors is added. Empirically it was determined that the best results are achieved if the information of 32 or half of the available most intensive vectors is added to the feature vector (Fig. 8). In our case, the following values of coefficients were used: $a = 1, b = -0.5, k = 8, w = 32$.

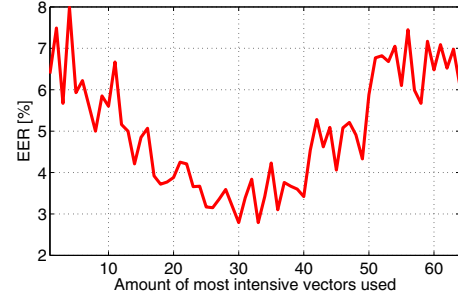


Fig. 8. Relationship between EER and the number of most intensive vectors described in the additional data vector

The advanced algorithm uses a training set; therefore, all the possible combinations of training sets have to be checked. Each person has 5 data samples, 2 of those samples are used for the training process, but the other 3 samples are used for the testing process. That gives a total of 10 different training sets. The calculated biocodes from the training samples are used to find the most stable bits. This stability information is used by the other 3 samples. For each training set 100 tests with different tokens were performed, where each person has the same token. The results can be seen in the Table III, where the training set number denotes a different training set. Results show that the improved bihashing approach achieves a lower EER than comparing the results of the basic biocode algorithm and EER is close to raw data comparison.

TABLE III
IMPROVED BIOHASH ALGORITHM TEST RESULTS USING DIFFERENT
TRAINING SETS (EER)

Training set Nr.	Veins		Palmprints		Fused data	
	Mean	StDev	Mean	StDev	Mean	StDev
1.	1,076	0,312	1,069	0,397	0,001	0,003
2.	1,122	0,289	1,369	0,439	0,003	0,006
3.	1,073	0,304	0,471	0,231	0,000	0,000
4.	1,440	0,609	0,712	0,382	0,000	0,000
5.	3,523	0,657	1,625	0,364	0,106	0,088
6.	4,051	0,568	1,326	0,023	0,035	0,029
7.	3,712	0,775	1,302	0,236	0,078	0,052
8.	3,876	0,843	5,195	0,658	0,217	0,118
9.	3,999	0,650	4,924	0,543	0,207	0,090
10.	4,721	0,834	5,475	0,915	0,297	0,174

VI. CONCLUSIONS

In this paper, we have discussed the four most important aspects of creating a biometric system: how to acquire biometric data, how to extract the valuable information from images, how to protect the biometric information and how to compare it. However, the main contribution is the biohashing algorithm enhancements. This research shows that by using the biohash algorithm, the statistical information obtained about the biocodes can improve the biocode comparison algorithm and obtain a lower EER. The proposed algorithm satisfies the requirements of the cancelable biometrics without sacrificing the recognition accuracy, compared to the raw biometric data comparison. The proposed biohashing approach could be useful for securing any type of biometric data.

Experimental results showed that results vary if different training samples are used. This can be explained by different lighting conditions that were employed to simulate real life conditions. For training, there should be more than two images to obtain more precise statistical information. This can improve the recognition results. It is once again shown that by using multimodality in biometric systems a lower EER can be achieved compared to using only one parameter. Our motivation is to build an easy to use, safe and reliable biometric system for everyday usage, and we believe that this article is a step towards achieving our goals.

REFERENCES

- [1] A. Kumar and D. Zhang, "Personal recognition using hand shape and texture," in *IEEE Transactions on Image Processing*, Vol. 15, Issue:8, IEEE, 2006, pp. 2454–2461.
- [2] M. G. K. Ong, T. Connie, A. T. B. Jin, and D. N. C. Ling, "A single-sensor hand geometry and palmprint verification system," in *WBMA '03: Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*. New York, NY, USA: ACM, 2003, pp. 100–106.
- [3] N. Nageshkumar, P. Mahesh, and M. Shanmukha Swamy, "An efficient secure multimodal biometric fusion using palmprint and face image," *International Journal of Computer Science Issues, IJCSI*, vol. 2, pp.49–53, 2009.
- [4] W. Yang, X. Yu, and Q. Liao, "Personal authentication using finger vein pattern and finger-dorsa texture fusion," in *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*. New York, NY, USA: ACM, 2009, pp. 905–908.
- [5] R. K. Rowe, U. Uludag, M. Demirkus, S. Parthasaradhi, and A. K. Jain, "A multispectral whole-hand biometric authentication system," in *ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences*. New York, NY, USA: ACM, 2009, pp. 1207–1211.
- [6] J.-G. Wang, W.-Y. Yau, A. Suwandy, and E. Sung, "Fusion of palmprint and palm vein images for person recognition based on "laplacianpalm" feature," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, june 2007, pp. 1–8.
- [7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Prentice Hall, 2007.
- [8] D. Zhang and W. Shu, "Two novel characteristics in palmprint verification: Datum point invariance and line feature matching," *Pattern Recognition*, vol. 32, pp. 691–702, 1999.
- [9] N. Duta, A. Jain, and K. Mardia, "Matching of palmprint," *Pattern Recognition Letters*, vol. 23, pp. 477–485, 2002.
- [10] J. Doi and M. Yamanaka, "Personal authentication using feature points on finger and palmar creases," in *Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. 32nd*, oct. 2003, pp. 282–287.
- [11] W. Li, D. Zhang, and Z. Xu, "Palmprint identification by fourier transform," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, pp. 417–432, 2002.
- [12] C.-C. Han, H.-L. Cheng, C.-L. Lin, and K.-C. Fan, "Personal authentication using palm-print features," *Pattern Recognition*, vol. 36, no. 2, pp. 371–381, 2003.
- [13] J. You, W.-K. Kong, D. Zhang, and K. H. Cheung, "On hierarchical palmprint coding with multiple features for personal identification in large databases," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 2, pp. 234–243, feb. 2004.
- [14] L. Zhang and D. Zhang, "Characterization of palmprints by wavelet signatures via directional context modeling," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 3, pp. 1335–1347, june 2004.
- [15] L. Zhang, Z. Guo, Z. Wang, and D. Zhang, "Palmprint verification using complex wavelet transform," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 2, october 2007, pp. 417–420.
- [16] W. K. Kong, D. Zhang, and W. Li, "Palmprint feature extraction using 2-d gabor filters," *Pattern Recognition*, vol. 36, no. 10, pp. 2339–2347, 2003.
- [17] A. Kong and D. Zhang, "Feature-level fusion for effective palmprint authentication," in *Biometric Authentication*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 3072, pp. 1–43.
- [18] Z. Sun, T. Tan, Y. Wang, and S. Li, "Ordinal palmprint representation for personal identification [representation read representation]," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, june 2005, pp. 279–284.
- [19] M. Greitans, M. Pudzs, and R. Fuksis, "Object analysis in images using complex 2d matched filters," in *EUROCON 2009: Proceedings of IEEE Region 8 conference*. IEEE, 2009, pp. 1392–1397.
- [20] M. Greitans, M. Pudzs, and R. Fuksis, "Palm vein biometrics based on infrared imaging and complex matched filtering," in *MM&Sec '10: Proceedings of the 12th ACM workshop on Multimedia and security*. New York, NY, USA: ACM, 2010, pp. 101–106.
- [21] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum, "Detection of blood vessels in retinal images using two-dimensional matched filters," in *IEEE transactions on medical imaging*, Vol.8, Issue:3. IEEE, 1989, pp. 263–269.
- [22] K. Nandakumar, A. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 4, pp. 744–757, dec. 2007.
- [23] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, pp. 97–139, March 2008.
- [24] R. Belguechi, C. Rosenberger, and S. A. Aoudia, "Biohashing for securing minutiae template," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ser. ICPR '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 1168–1171.
- [25] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, pp. 2245–2255, 2004.
- [26] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Syst. J.*, vol. 40, pp. 614–634, March 2001.

Plaukstu detektēšana un biometrisku parametru lokalizēšana

Problēma

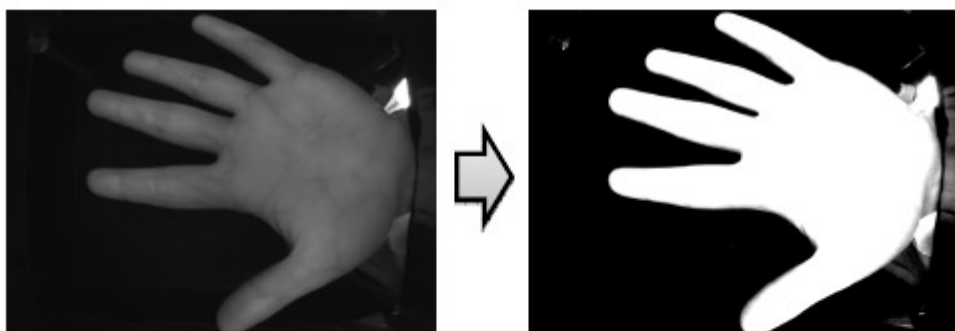
- Publiski pieejama plaukstu multispektrālu attēlu datubāze
- Lai apstrādātu šos attēlus pirmkārt ir nepieciešams automātiski izdalīt nepieciešamos reģionus (ROI)
- Plaukstu ģeometrisku parametru ROI, asinsvadu un rievojuma struktūras ROI.
- Pirmie soļi šīs problēmas risināšanai...

Attēls



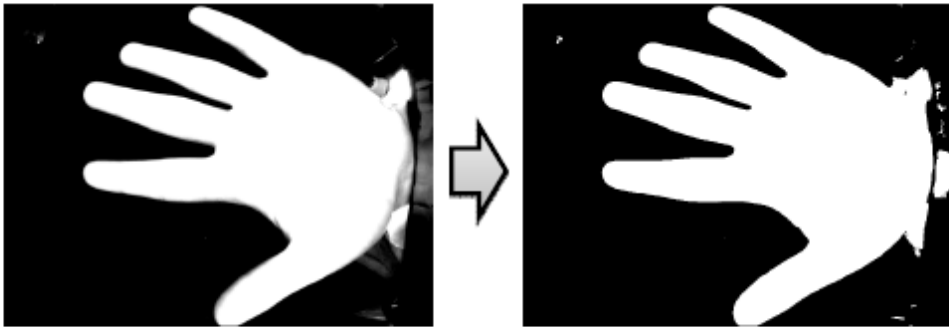
Algoritms

1.) Tiek veikta kontrasta transformācija



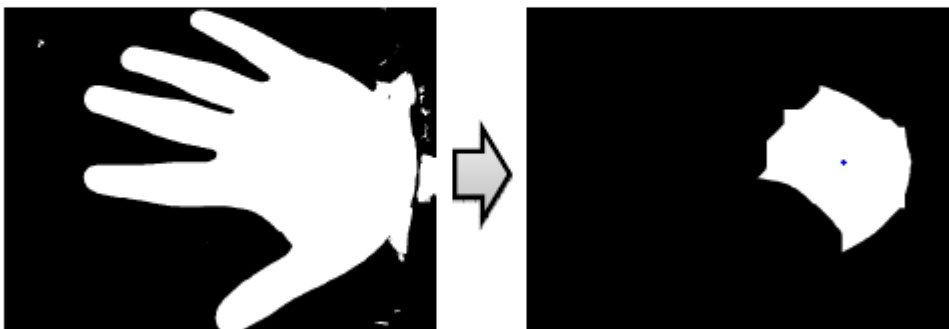
Algoritms

2.) Sliekšņošana



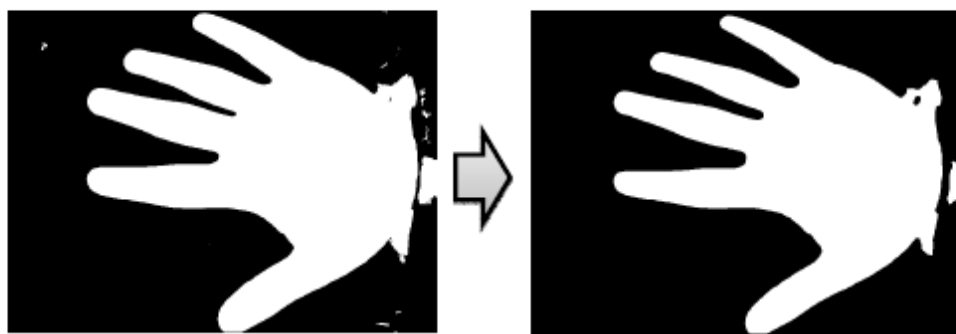
Algoritms

3a) Centroidas atrašana (novāc detaļas, kas šaurākas par 50 pix.) aprēķina centroidu.



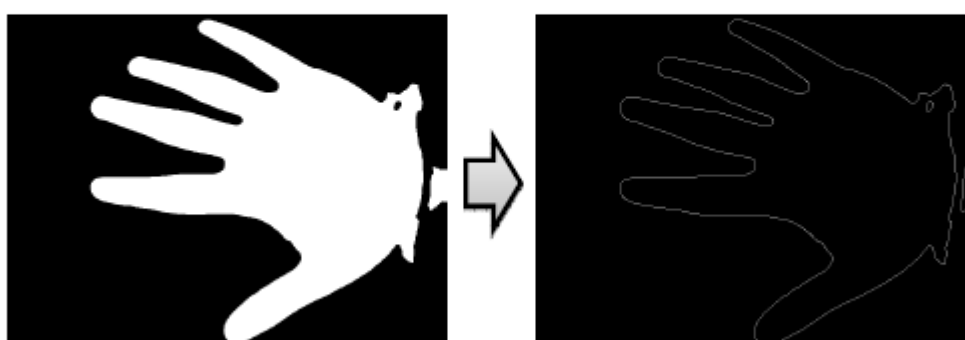
Algoritms

3b) Sliekšnotā attēla notīrīšana ar Erosion morfoloģisko operāciju



Algoritms

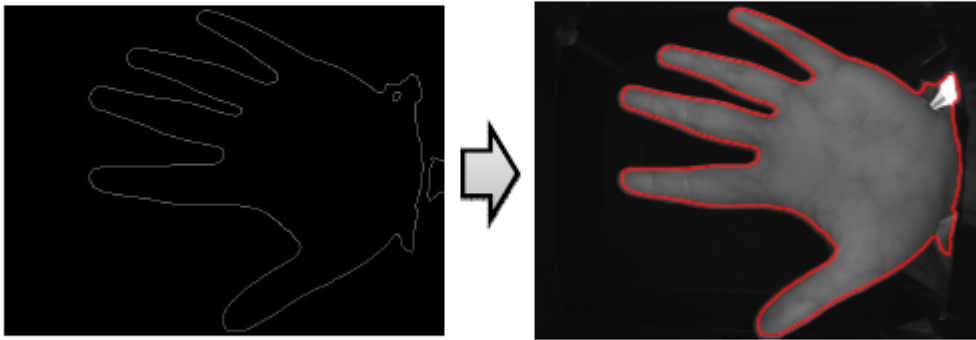
4.) Edge detection ar Sobel operatoru



Algoritms

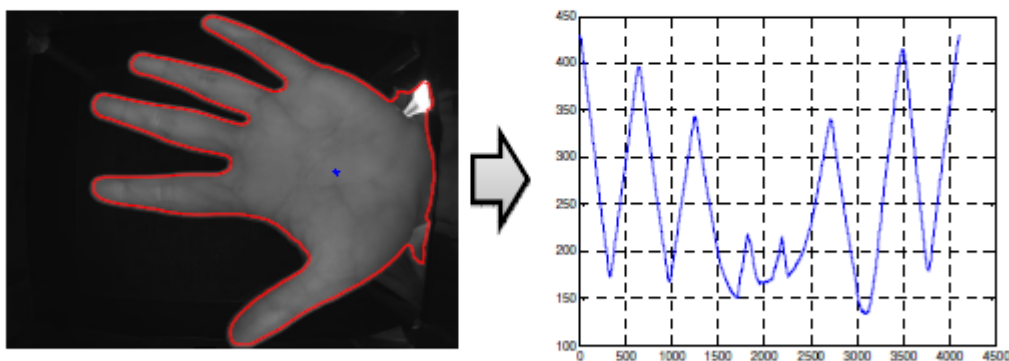
5.) Robežu meklēšana (boundary detection)

Lielākās robežas atstāšana



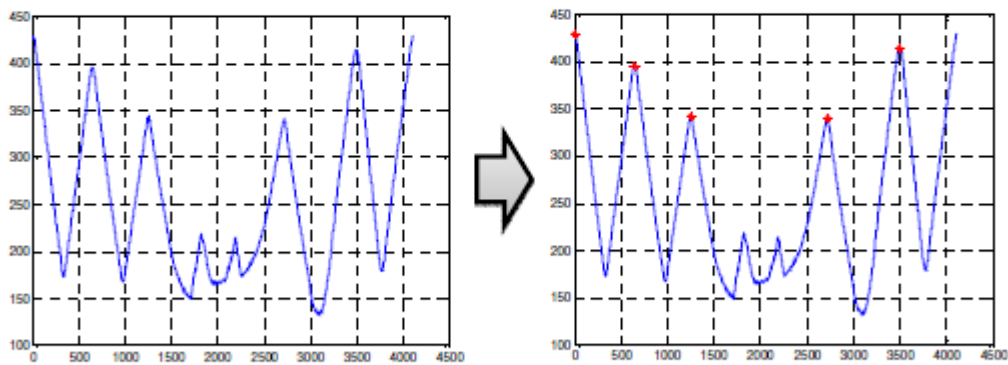
Algoritms

6.) Distances aprēķināšana no katra robežas punkta līdz centroidas punktam



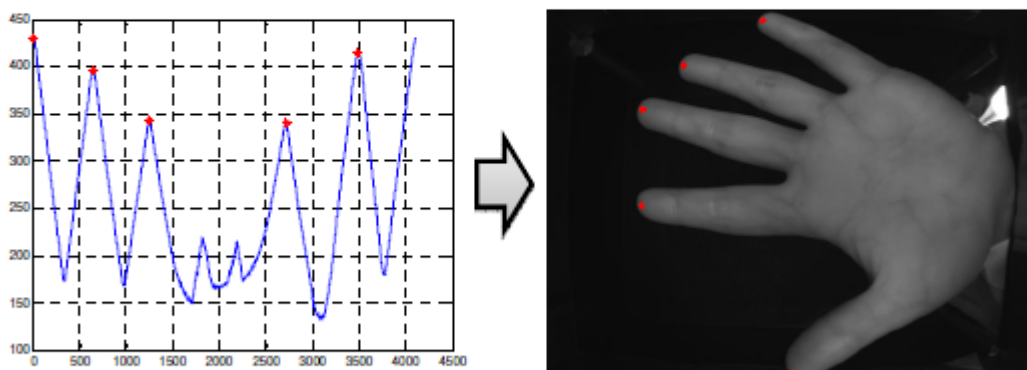
Algoritms

7.) Maksimumu detektēšana distances grafikā



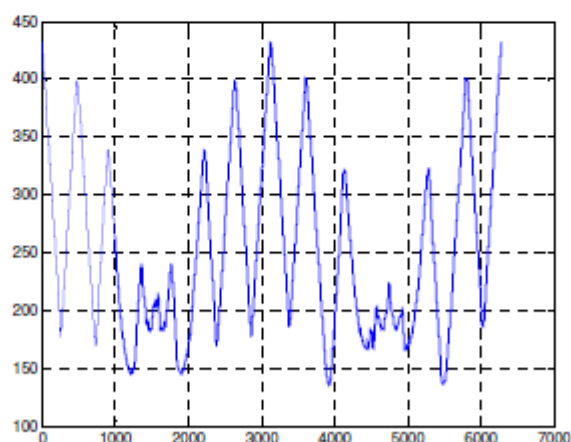
Algoritms

8.) Atrasto punktu attēlošana sākotnējā attēlā:



Problēmas

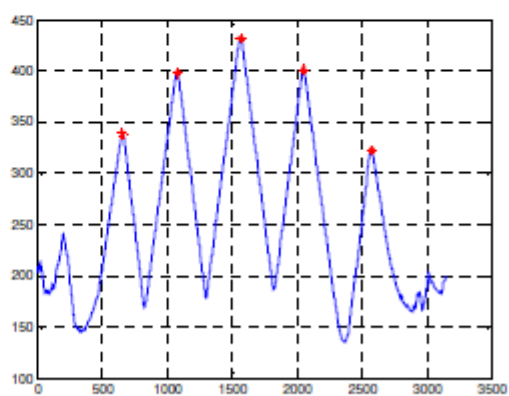
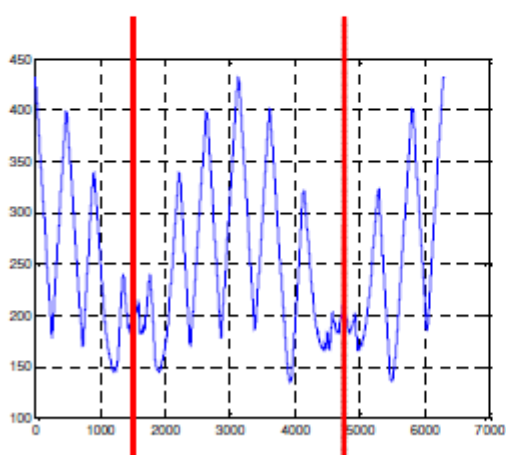
- Kontūru detektēšanas algoritms nestrādā precīzi, ja pēc edge detection kontūrs ir pārtrauts.



Problēmas risinājums

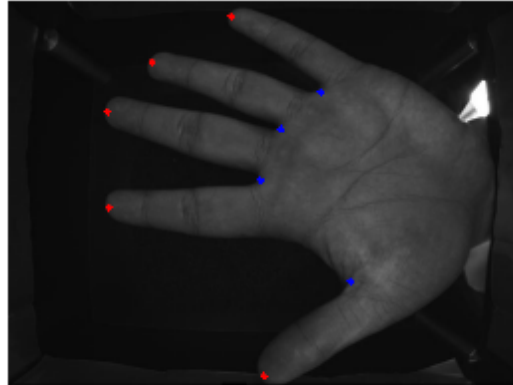
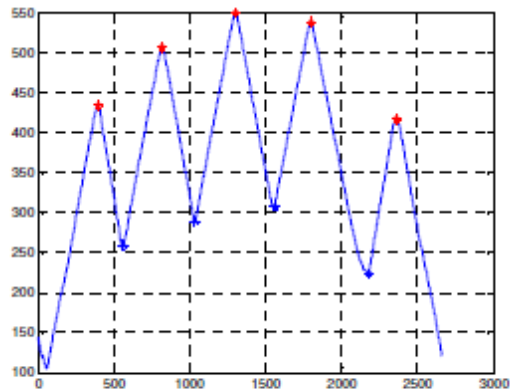
Tiek aprēķināti attālumi starp atrastajiem maksimumiem un atrasti divi lielākie, kas norāda uz to kur ir plaukstas kontūra sākums un beigas.

Tiek izgriezts kontūrs un atrasti 5 maksimumi, kas reprezentē pirkstu galapunktus.



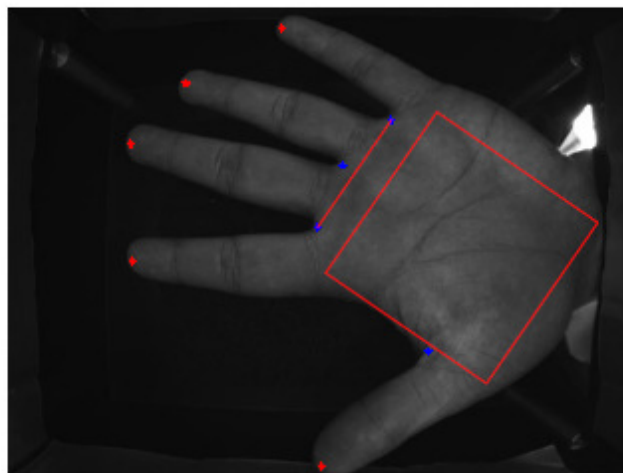
Algoritma turpinājums

9.) Pirkstu starpu detektēšana:



Algoritma turpinājums

10.) Reģiona (ROI) ievilkšana izmantojot atrasto punktu informāciju



Rezultāts

- Programma atgriež atrasto reģionu, kā arī leņķi, kādā šis reģions ir atrasts, lai pēc apstrādes ar CMF, vektorus varētu rotēt atpakaļ

