

# SISTEM ZA VERIFIKACIJO OSEBE NA PODLAGI PRSTNEGA ODTISA

Uroš Klopčič, Peter Peer

Laboratorij za računalniški vid

Fakulteta za računalništvo in informatiko

E-pošta: uros.klopacic@gmail.com, peter.peer@fri.uni-lj.si

---

**POVZETEK:** V tem delu je predstavljen sistem za verifikacijo oseb na podlagi prstnega odtisa. V osnovi gre za prototipni sistem, katerega namen je postavitev temeljev in ogrodja, katero bi omogočalo nadaljnji razvoj in testiranje. Sistem sestoji iz korakov segmentacije, izboljšave kvalitete prstnega odtisa z Gaborjevimi filtri, tanjšanja, iskanje značilk, klasifikacije ter primerjanja. Algoritem smo testirali na štirih testnih množicah s tekmovanja Fingerprint Verification Competition 2002 (FVC2002). Algoritem daje zadovoljive rezultate, več težav se pojavi predvsem pri prstnih odtisih slabe kvalitete v koraku izboljšanja kvalitete prstnega odtisa. Čas procesiranja 3 sekund ustreza zahtevam enostavnega sistema za verifikacijo oseb na podlagi prstnega odtisa.

---

## 1. UVOD

Prstni odtisi že dolgo veljajo za eno od najbolj zanesljivih in natančnih biometričnih značilnosti, zato so se že zgodaj uveljavili pri procesu ugotavljanja identitete posameznika. S pojavom ter razvojem računalniške tehnologije, se je zamudna ročna identifikacija posameznika zamenjala s hitrimi in zmogljivi sistemi za identifikacijo oseb na podlagi prstnega odtisa (angl. *AFIS*). Danes so prstni odtisi že popolnoma uveljavljeni, uporabljajo se predvsem pri kontroli dostopa ter policijskem delu.

Za namene kontrole dostopa v laboratorij smo tudi sami želeli razviti sistem za verifikacijo oseb na podlagi prstnega odtisa. Pri tem si kot glavni cilj nismo zastavili 100% zanesljivost sistema, temveč dobro ogrodje ter modularno zgradbo, na kateri bi bilo uveljavljanje prihodnjih nadgradenj čim bolj enostavno. Šele skozi proces posodobitev pa bi sistem pridobil potrebno zanesljivost ter natančnost.

Sistem bi lahko v grobem ločili v tri korake: zajem prstnega odtisa, iskanje značilk ter odločanje. Za zajem prstnega odtisa smo uporabili dva čitalca prstnih odtisov, vendar smo zaradi lažjega dela pri samem razvoju sistema uporabili v naprej zbrano zbirko slik prstnih odtisov. Večji poudarek je bil predvsem na koraku iskanja značilk, od katerega je odvisen tudi končni rezultat v koraku odločanja.

## 2. OBDELAVA PRSTNEGA ODTISA

### 2.1 Segmentacija

Pri segmentaciji poskušamo čim bolj ločiti področje, na katerem se nahaja prstni odtis, od ozadja. Pri tem smo uporabili metodo, ki je modificirana verzija pristopa opisanega v [3]. Kot prvo se poišče vrednost, ki predstavlja ozadje. V ta namen izračunamo histogram negativna slike. Nato potujemo po histogramu od indeksa 0 do 255 in opazujemo vrednost histograma. Indeks, kjer je ta vrednost prvič večja od praga  $ThrF$  (1000 v našem primeru), predstavlja vrednost ozadja. Ta vrednost se odšteje od vsakega slikovnega elementa  $(i, j)$  na sliki. Nato z enačbo (1) povečamo dinamično razdaljo med grebeni in dolinami in še enkrat izračunamo negativno sliko.

$$neg\_stretched(i, j) = \frac{neg\_subtracted(i, j)}{\max_{i, j}(neg\_subtracted(i, j))} \cdot 255 \quad (1)$$

V drugem delu sliko razdelimo na bloke enake velikosti  $W \times W$  ( $W = 9$  v našem primeru). Za vsak blok s središčem v  $(i, j)$  se izračuna štiri parametre: srednja vrednost, varianca, gradient in  $ROI$ , prikazan v enačbi (2).

$$ROI(i, j) = \sqrt{\frac{1}{W^2} \cdot \frac{(V_x(i, j)^2 + V_y(i, j)^2)}{V_e(i, j)}}, \quad (2)$$

kjer velja

$$V_x(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2 \cdot \partial_x(u, v) \cdot \partial_y(u, v), \quad (3)$$

$$V_y(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (\partial_x^2(u, v) - \partial_y^2(u, v)), \quad (4)$$

$$V_e(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (\partial_x^2(u, v) + \partial_y^2(u, v)) \quad (5)$$

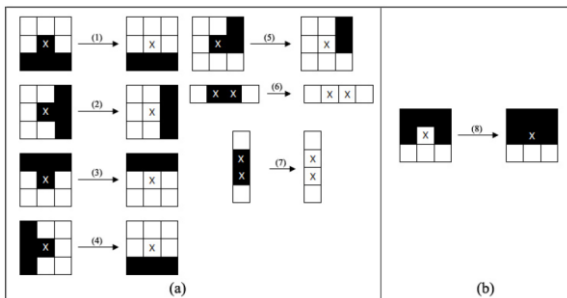
$ROI$  poda stopnjo gotovosti, ki nam glede na prag pove, ali slikovni element pripada ozadju ali ne.  $\partial_x$  in  $\partial_y$  sta jakosti gradienta v  $x$  in  $y$  smeri, katera pridobimo s Sobelovim operatorjem [8]. Če je srednja vrednost večja in ostali parametri manjši od nastavljenih pragov, potem je blok označen za ozadje.

S tem korak segmentacije še ni končan. Pogosto se namreč zgodi, da so nekateri bloki napačno označeni za ozadje oziroma ospredje. Zato je potrebna še naknadna obdelava, kjer se take bloke prepozna in popravi. V ta namen se uporabi množica hevrističnih pravil (slika 1).

### 2.2 Izboljšava kvalitete slike prstnega odtisa

Pri prepoznavanju prstnih odtisov je zelo pomembna predhodna obdelava zajete slike prstnega odtisa, saj slaba kvaliteta negativno vpliva na postopke iskanje značilnik. Pri

zajemu namreč pogosto pride do napak kot so šum in smeti, vzrok je lahko tudi prevelik pritisk na čitalec, pogoste pa so tudi mehanske poškodbe samega prsta (urez, praska). Zaradi tega smo v našem sistemu uporabili postopek izboljšanja kvalitete opisan v [4].



Slika 1: Hevristična pravila za odstranjevanje (a) blokov, označenih za ospredje, (b) blokov, označenih za ozadje

Nad prstnim odtisom se najprej izračuna orientacijsko polje s pomočjo metode gradientov. Orientacijsko polje nam za vsak del slike pove, v katero smer je prstni odtis usmerjen. Dobljeno orientacijsko polje pa ni povsem zanesljivo, saj vsebuje preveč šuma. Zato se izvede glajenje orientacijskega polja. Pri tem smo uporabili metodo [5], ki velikost okolice glajenja  $\Omega(s)$  pametno prilagaja glede na doslednost orientacije. Za izračun slednje se uporabi enačba (6).

$$Cons(s) = \frac{\sqrt{(\sum_{(i,j) \in \Omega(s)} \cos(2\theta(i,j)))^2}}{O(s)} \quad (6)$$

$O(s)$  predstavlja število orientacij  $\theta(i,j)$  v  $\Omega(s)$ . Kadar je doslednost majhna, se uporabi veliko okolico glajenja, s čimer omejimo vpliv šuma, obratno pa uporabimo majhno okolico, saj vemo, da je orientacija zanesljiva. Za vsak blok izračunamo  $Cons(s)$  in njegovo orientacijo na podlagi zunanjih  $\delta \cdot s$  blokov iz njegove  $(2 \cdot s + 1) \times (2 \cdot s + 1)$  velike okolice. Koraki glajenja so naslednji:

1. Izračunaj enotski vektor z dvakratno orientacijo  $[\cos(2\theta(i,j)), \sin(2\theta(i,j))]$ . Nastavi  $s = 1$  in uporabi vrednosti enotskega vektorja pri izračunu  $Cons(s)$ .
2. Povečaj  $s$  za 1. Izračunaj  $Cons(s)$  v  $\Omega(s)$ , kjer je  $\Omega(s)$   $\delta \cdot s$  zunanjih blokov.
3.  $Cons(s)$  doseže vrednosti na intervalu  $[0,1]$ . Če je  $Cons(s)$  manjši od praga (0,5 v našem primeru) ali pa manjši od  $Cons(s - 1)$ , potem pojdi na korak 2; na korak 2 se vračaš, dokler ne dosežeš maksimuma vrnitev (5 v našem primeru).
4. Če je  $s$  enak maksimumu, nastavi  $\Omega(s)$  na okolico velikosti  $3 \times 3$  blokov, sicer pojdi na korak 5.
5. Izračunaj zglajeno orientacijo po sledeči enačbi:

$$\bar{\theta}(i_c, j_c) = \frac{1}{2} \arctan\left(\frac{\sum_{(i,j) \in \Omega(s)} \sin(2\theta(i,j))}{\sum_{(i,j) \in \Omega(s)} \cos(2\theta(i,j))}\right) \quad (7)$$

kjer sta  $i_c$  in  $j_c$  koordinati točke, ki predstavlja središče trenutno obravnavanega bloka.

Izračunu orientacije sledi še izračun frekvence [4], nato pa se izvede dejansko filtriranje s pomočjo Gaborjevih filtrov.

### 2.3 Binarizacija in tanjšanje grebenov

Binarizacija se izvede s pomočjo prosto dostopnega orodja AForge .NET. Kot rezultat dobimo sliko, kjer so grebeni predstavljeni z bitom 1 in ozadje z bitom 0.

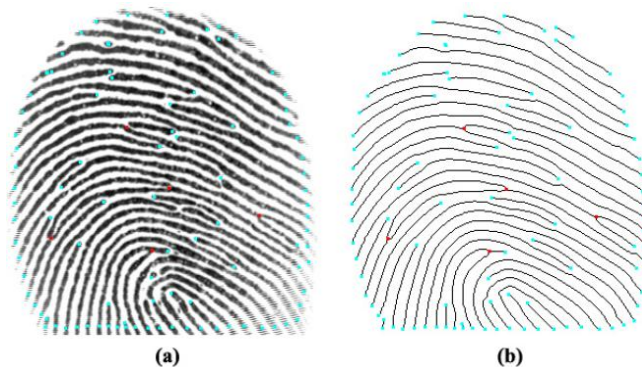
Namen tanjšanja grebenov je pridobiti skelet prstnega odtisa, kjer so grebeni široki natanko 1 slikovni element. V ta namen smo uporabili algoritem v [6], ki prične odstranjevati slikovne elemente na zunanjih robovih grebenov, dokler le-ti niso debeli zgolj 1 slikovni element. To dosežemo z uporabo 21 pravil, ki jih uporabimo v vsaki iteraciji nad vsakim slikovnim elementom. Algoritem uporablja tudi 4 posebna pravila za tanjšanje diagonalnih črt, poleg tega pa vsebuje tudi 12 pravil za predpripravo na tanjšanje.

### 2.4 Iskanje značilk

Naloga algoritma iskanja značilk je poiskati osnovne značilke: razcep (angl. *bifurcaton*) in zaključek (angl. *ridge ending*). Kadar imamo na voljo skelet prstnega odtisa, je naloga razmeroma enostavna. Pri iskanju zaključkov grebenov se opiramo na število križišč [10] v trenutni točki (angl. *Crossing Number*). Enačba (8) definira število križišč  $cn(\mathbf{p})$  točke  $\mathbf{p}$  v binarni sliki kot polovico vsote razlik med pari sosednjih točk v okolici točke  $\mathbf{p}$ .

$$cn(\mathbf{p}) = \frac{1}{2} \sum_{i=1}^8 |\text{val}(\mathbf{p}_{i \bmod 8}) - \text{val}(\mathbf{p}_{i-1})|, \quad (8)$$

kjer so  $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_7$  točke, ki pripadajo okolici točke  $\mathbf{p}$ ,  $\text{val}(\mathbf{p}) \in \{0, 1\}$  pa predstavlja vrednost točke. Za točko  $\mathbf{p}$  velja, da je zaključek grebena, če je  $cn(\mathbf{p}) = 1$ .



Slika 2: Najdene značilke, prikazane na (a) sivinski sliki in (b) sliki s skeletom prstnega odtisa

Pri iskanju razcepov si pomagamo z množico pravil. Za vsako točko se njena  $3 \times 3$  okolica primerja s 24 možnimi maskami za razcep na sliki 2. Če najdemo ujemanje, potem je ta kandidat dejansko razcep.

Kljub temu, da smo kvaliteto slike prstnega odtisa izboljšali in da smo iskali značilke le na področju, kjer se nahaja prstni odtis, algoritem še vedno najde večje število nepravilnih značilk, ki niso primerne za nadaljnjo obdelavo. Zato vse značilke preverimo še z algoritmom verifikacije značilk [1]. S tem odstranimo nepravilne strukture (špice, mostove, kratke greben, itd.) in posledično povečamo natančnost sistema.

## 2.5 Klasifikacija

V postopku klasifikacije poskušamo prstnemu odtisu določiti razred glede na obliko grebenov. Pri tem smo uporabili 5 najbolj uporabljenih razredov Galton-Henryevega sistema: lok, šotorast lok, leva in desna zanka ter spirala. Določitev razreda temelji na lokaciji in številu singularnih točk (jedra in delte). Te lahko določimo z uporabo polja ukrivljenosti (angl. *curvature map*) [7].

Za izračun polja ukrivljenosti sliko razdelimo na bloke velikosti  $W \times W$  ( $W = 7$  v našem primeru). Ukrivljenost bloka se izračuna glede na ukrivljenost 8 sosednjih blokov v njegovi  $3 \times 3$  okolici. Natančneje, ukrivljenost bloka je enaka polovici razlike ukrivljenosti blokov, na katere kaže smer obravnavanega bloka (enačba 9).

$$B_k = \frac{|O(B_1) - O(B_2)|}{2}, \quad (9)$$

kjer sta  $B_1$  in  $B_2$  bloka, na katera kaže smer središčnega bloka. Singularne točke iz polja ukrivljenosti dobimo na naslednji način:

1. Določimo prag ukrivljenosti  $T_c$  ter prag razdalje  $T_d$  (0,35 in 4 v našem primeru)
2. Označimo z  $S_c$  množico singularnih točk, katerih ukrivljenost je večja od  $T_c$ .
3. Izberemo katerakoli dva bloka  $B_i$  in  $B_j$  iz množice  $S_c$ . Če je razdalja med njima manjša od  $T_d$ , se blok z manjšo ukrivljenostjo odstrani iz množice  $S_c$ .
4. Pojdi na korak 3, dokler velikost množice  $S_c$  ni stalna.
5. Množica  $S_c$  vsebuje resnične singularne točke.

Za prstne odtise tipa lok je znano, da ne vsebujejo izrazitih singularnih točk. Ker pa naš postopek zahteva določitev referenčne točke prstnega odtisa, se za jedro vzame kar blok z najvišjo ukrivljenostjo.

Tip dobljenih singularnih točk (jedro, delta) določimo z metodo Poincare Index [9]. Ko poznamo število jeder  $N_c$  in delt  $N_d$ , lahko klasificiramo prstni odtis glede na naslednja pravila:

1. Če je  $N_c = 2$ , potem je prstni odtis tipa spirala.
2. Če je  $N_c = 0$  in  $N_d = 0$ , potem je prstni odtis tipa lok.
3. Če je  $N_c = 1$  in  $N_d = 1$ , potem je prstni odtis lahko tipa šotorast lok ali zanka. Potrebna je nadaljnja analiza. Med jedrom in delto potegnemo ravno črto. Pri šotorastem loku je črta vzporedna z lokalnimi orientacijami, med tem ko jih pri

zanki seka. Naj bo  $\lambda$  kot, ki ga tvori daljica med jedrom in delto in  $\eta_1, \eta_2, \dots, \eta_n$  koti lokalnih usmeritev vzdolž omenjene daljice. Če je povprečna vsota (enačba 10)

$$\sum_{i=1}^n \sin(\eta_i - \lambda) \quad (10)$$

manjša od pragu (0,3 v našem primeru), potem je prstni odtis tipa šotorast lok, sicer je tipa zanka. Leve zanke ločimo od desnih s pomočjo enačbe (11) - če je rezultat manjši od 0, potem je leva, sicer desna zanka.

$$(B_x - C_x)(D_y - C_y) - (B_y - C_y)(D_x - C_x) \quad (11)$$

kjer indeksa  $x$  in  $y$  označujeta  $x$  in  $y$  koordinati točk.

4. Sicer tipa prstnega odtisa ni mogoče določiti.

Referenčno točko tako predstavlja kar jedro. V primeru dveh jeder pa v ta namen izberemo kar središčno točko med obema jedroma.

## 2.6. Primerjanje

Primerjanje temelji na primerjavi lokacije, kota, usmeritve ter tipa značilk. V našem algoritmu se značilke najprej pretvorijo v polarni koordinatni sistem, kjer se za središče vzame referenčno točko.

Zadnji manjkajoči podatek je orientacija referenčne točke. Če obstaja dve jedri, je orientacija enaka orientaciji premice skozi ti dve točki. Sicer se okoli referenčne točke definira 16 smeri, ločenih s korakom  $\pi/8$ . Orientacija referenčne točke je smer, kateri se lokalne orientacije najbolj prilagajajo. Značilke nato predstavimo v polarnem koordinatnem sistemu s štirimi parametri ( $r, e, s, \Phi$ ): razdaljo od referenčne točke, polarnim kotom, tipom značilke ter kotom med  $r$  in smerjo značilke.

Primerjanje med vhodnim vektorjem značilk  $P_i = \{(r_{i1}, e_{i1}, s_{i1}, \Phi_{i1})^T, \dots, (r_{iM}, e_{iM}, s_{iM}, \Phi_{iM})^T\}$  in tistim iz podatkovne zbirke  $Q_j = \{(r_{j1}, e_{j1}, s_{j1}, \Phi_{j1})^T, \dots, (r_{jN}, e_{jN}, s_{jN}, \Phi_{jN})^T\}$  poteka na naslednji način:

1. Vsaka značilka iz  $P_i$  se primerja z vsako značilko iz  $Q_j$ .
2. Naj  $M_0$  in  $N_0$  označujeta število značilk v prvem in drugem prstnem odtisu,  $M_N$  pa število uspešnih ujemanj. Kadar je  $M_N / \sqrt{N_0 M_0}$  večji od praga  $\beta$ , potem smatramo obravnavana prstna odtisa za enaka.
3. Vzemi naslednji vektor  $Q_{j+1}$ .

Zaradi šuma in nelinearnih deformacij je nemogoče natančno določiti lokacijo značilk glede na referenčno točko. Zato mora primerjalni algoritem upoštevati določeno toleranco. V našem primeru to dosežemo z uporabo dinamične mejne škatle (angl. *Variable Bounding Box*). Naj kot  $\alpha_b$  in radij  $r_b$  označujeta velikost škatle. Kadar je radij značilke majhen, lahko majhna deformacija povzroči veliko spremembo v kotu radija značilke, medtem ko na radij nima vpliva. V tem primeru bi moral biti  $\alpha_b$  večji in  $r_b$  manjši. Kadar pa je radij značilke velik, lahko majhna sprememba v kotu radija značilke povzroči veliko spremembo v lokaciji značilke. Takrat  $\alpha_b$  pomanjšamo in  $r_b$  povečamo,

saj se vse deformacije med referenčno točko ter značilko seštevajo, kar se pozna na radiju.

### 3. REZULTATI

Sistem smo testirali na štirih množicah slik prstnih odtisov iz FVC2002. Vsaka množica je vsebovala 800 slik. Slike so bile pridobljene z optičnim (DB1, DB2) in kapacitivnim (DB3) čitalcem, zbirka DB4 pa je vsebovala sintetično generirane prstne odtise.

Vsi koraki z izjemo koraka izboljšanja kvalitete so dali zadovoljive rezultate. Pri izboljšanju kvalitete smo imeli probleme predvsem s prstnimi odtisi slabe kvalitete, saj algoritem ni bil zmožen ločevati med sosednjimi grebeni. V nekaterih primerih prstnih odtisov slabe kvalitete naš sistem ni bil sposoben podati natančne lokacije in števila singularnih točk, zaradi česar je bila klasifikacija prstnega odtisa nemogoča. Za primerjavo z ostalimi sistemi smo izmerili stopnji FNMR (angl. *False Non-Match Rate*) - in FMR (angl. *False Match Rate*). Prva označuje neujemanje dveh vzorcev istega osebk, druga pa ujemanje dveh vzorcev različnih osebkov. Rezultati v tabeli 1 nam povejo, da naš sistem ni popolnoma zanesljiv. Zaradi velikega vpliva kvalitete na rezultate smo testirali sistem tudi s 64 slikami boljše kakovosti. Slednji rezultati so podani v tabeli 2. Opazimo veliko izboljšanje FNMR pri DB1 ter DB2, malo manj pa pri ostalih množicah, saj so najboljše slike iz teh množic še zmeraj vsebovale preveč šuma. Na koncu smo s sliko 3 podali še skupne rezultate za vsako testno množico. Predstavljeni so bili s krivuljo ROC (angl. *Receiver Operating Characteristic*), ki prikazuje FNMR in FMR v odvisnosti od praga. Kjer se krivulji FNMR in FMR sekata, je t.i. točka EER (angl. *Equal Error Rate*). Na sliki sta s praznim kvadratom in polnim krogcem označeni oceni EER tudi za predzadnji in najboljši algoritem iz FVC2002 [2]. Povprečen čas procesiranja je znašal 3 sekunde.

Tabela 1: Stopnji FNMR in FMR pri pragu  $\beta = 0.3$ .

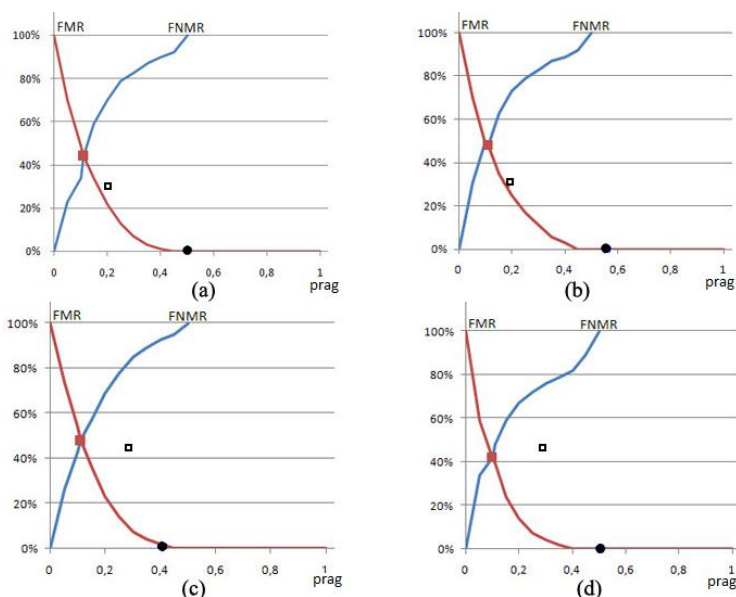
Stopnja	TESTNE MNOŽICE			
	DB1	DB2	DB3	DB4
FNMR (%)	89.3	88.6	91.2	81.3
FMR (%)	1.7	3.7	2.4	0.9

Tabela 2: Stopnji FNMR in FMR pri pragu  $\beta = 0.3$  za slike dobre kakovosti.

STOPNJA	TESTNE MNOŽICE			
	DB1	DB2	DB3	DB4
FNMR (%)	6.1	41.1	42.6	8.2
FMR (%)	0	0	0	0

### 4. ZAKLJUČEK

Sistem še ni dovolj zanesljiv in potrebuje nekaj popravkov. Največ napak se zgodi predvsem zaradi slik slabe kakovosti. Zato bo poudarek v nadaljnjem razvoju predvsem pri jasni določitvi frekvence grebenov, ki mora biti robustna in odporna na šum. Prav tako potrebuje nekaj izboljšav tudi postopek določitve lokacije singularnih točk, kjer trenutno obravnavamo nekaj različnih pristopov. Izdelan sistem predstavlja dobro osnovo in ponuja še veliko možnosti za nadgradnjo.



Slika 3: ROC krivulje za testne množice (a) DB1, (b) DB2, (c) DB3 in (d) DB4.

## LITERATURA

1. M.U. Akram, A. Tariq, S.A. Khan, S. Nasir, Fingerprint image: pre and post-processing, *International Journal of Biometrics* 2008, vol. 1(1), str. 63-80, 2008
2. <http://bias.csr.unibo.it/fvc2002/>  
Informacije o temovanju Fingerprint Verification Competition 2002
3. M.M. Hadhoud, W.S. El Kilani, M.I. Samaan, An adaptive algorithm for fingerprints image enhancement using gabor filters, *ICCES '07. International Conference on Computer Engineering & Systems*, str. 227-236, November 2007
4. L. Hong, Y. Wan, A. Jain, Fingerprint image enhancement: algorithm and performance evaluation, *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 20(8), str. 777-789, 1998
5. M. Liu, X. Jiang, A.C. Kot, Fingerprint Reference Point Detection, *EURASIP Journal on Applied Signal Processing*, vol. 2005, str. 498-509, Januar 2005
6. P.M. Patil, S.R. Suralkar, F.B. Sheikh, Rotation invariant thinning algorithm to detect ridge bifurcations for fingerprint identification, *17<sup>th</sup> IEEE International Conference on Tools with Artificial Intelligence*, str. 641-649, 2005
7. J. Qi, M. Xie, A robust algorithm for fingerprint singular point detection and image reference direction determination based on the analysis of curvature map, *IEEE Conference on Cybernetics and Intelligent Systems*, str. 1051-1054, 2008
8. J. Qi, M. Xie, Segmentation of fingerprint images using the gradient vector field, *IEEE Conference on Cybernetics and Intelligent Systems*, str. 543-545, September 2008
9. T. Tang, X. Wu, M. Xiang, An improved fingerprint singular point detection algorithm based on continuous orientation field, *International Symposium on Computer Science and Computational Technology ISCCST, 2<sup>nd</sup> Edition*, str. 454-457, 2008
10. F. Zhao, X. Tang, Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction, *Pattern Recognition*, vol. 40(1), str. 1270-1281, 2007