

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



Projektová praxe (IPP)
Technická zpráva k projektu Morphing otisků prstů

Květen 2020

Daniel Kolínek (xkolin05)

1 Abstrakt

Cílem práce bylo implementovat aplikaci, která umožňuje generování otisků ze dvou zadaných. Takový otisk poté může obsahovat dostatečné informace z obou otisků prstů, aby seděl na oba zároveň.

2 Úvod

Na morphingu obrazu se pracovalo již v 80. letech minulého století, kdy v roce 1989 filmový průmysl byla použita technika mesh warping technique ve filmu Indiana Jones and the Last Crusade. A proto není divu, že v posledních letech, kdy se používají pro zabezpečení biometrické údaje jako je otisk prstu nebo obličej, dochází k morfologickým útokům. Nebezpečná je tato technika z toho důvodu, že dvě osoby po morphingu svých biometrických údajů mohou při správném morfování dosáhnout toho, že oba mohou použít např. jeden doklad totožnosti, či přístupovou kartu.

3 Předzpracování otisků

Otisk prstu je stopa, kterou zanechávají papilární linie. Na běžném snímku otisku je papilární linie zaznamenána pomocí černé barvy (stupně šedi) a údolí mezi nimi pomocí bílé barvy, s čímž počítá i má implementace.

Pro co nejreálnější výsledky morphingu je zapotřebí, před jeho provedením, zajistit, aby si otisky byly co nejvíce podobné. Prvně se tedy obrázky otisků normalizují pomocí funkce OpenCV `cv2.normalize`. Po normalizaci je provedeno oříznutí pozadí.

3.1 Zarovnání otisků

Pro najetí nejlepší pozice zarovnání dvou otisků prstů byla použita technika pracující na úrovni orientovaných polí. Hlavně z toho důvodu, že dva odlišné otisky mohou mít úplně jiné rysy, jako například jeden může obsahovat core point, druhý ne. O tom píší v kapitole 3.3. Nejprve tedy něco k orientovanému poli.

3.1.1 Orientované pole

Nechť $[x, y]$ je bod (pixel) v obrazu otisku. Pak lokální orientace linie v bodě $[x, y]$ je úhel $\theta_{x,y}$, který svírají linie, procházející skrz libovolně velké okolí se středem v bodě $[x, y]$, s vodorovnou osou. Protože hřebeny otisku jsou nesměrové, $\theta_{x,y}$ je neorientovaný směr v rozmezí $[0..\pi]$. Orientované pole je poté matice D , v jejíž prvcích je uložena lokální orientace linií v daném okolí. Každý prvek $\theta_{x,y}$ matice D udává průměrnou orientaci linií v okolí bodu $[x, y]$, kde bod $[x, y]$ reprezentuje čtvercovou mřížku umístěnou nad tímto bodem viz. obrázek 1. S každým prvkem $\theta_{x,y}$ je často asociována hodnota $r_{x,y}$, která označuje míru spolehlivosti orientace.



Obrázek 1: Vlevo původní obraz otisku, vpravo orientované pole

Popis algoritmu:

1. Rozdělíme normalizovaný obraz \mathcal{G} na bloky o velikosti $w \times w$. V aktuální implementaci pokud se na celém bloku nevyskytuje žádná linie, poté je ihned označena hodnota bloku za 0 a r nastaveno na 0, v opačném případě se r nastaví na 1 a pokračuje se ve výpočtu.
2. Vypočítáme gradienty $\partial_x(i, j)$ a $\partial_y(i, j)$ pro každý pixel (i, j) pomocí Sobelova operátoru.
3. Stanovíme lokální orientaci pro každý blok se středem v pixelu (i, j) , pomocí následujících rovnic:

$$\mathcal{V}_x(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u, v)\partial_y(u, v) \quad (1)$$

$$\mathcal{V}_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial_x^2(u, v) - \partial_y^2(u, v)) \quad (2)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{\mathcal{V}_y(i, j)}{\mathcal{V}_x(i, j)} \right) \quad (3)$$

Kde $\theta(i, j)$ je nejmenší čtvercový odhad lokální orientace linií v bloku se středem v (i, j) .

4. Z důvodu výskytu šumu, nepravých linií, atp. nemusí vždy být lokální orientace $\theta(i, j)$ stanovena správně. Díky tomu, že se lokální orientace ve svém okolí mění pouze mírně, kde se neobjevují žádné singulární body, je pro opravu (vyhlazení) použit filtr s dolní propustí. Pro provedení operace filtrem je zapotřebí převést θ do vektorového pole, které je definováno následovně:

$$\Phi_x(i, j) = \cos(2\theta(i, j)) \quad (4)$$

$$\Phi_y(i, j) = \sin(2\theta(i, j)) \quad (5)$$

Následné filtrování filtrem s dolní propustí je provedeno:

$$\Phi'_x(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} \quad (6)$$

$$W(u, v) \Phi_x(i - uw, j - vw)$$

$$\Phi'_y(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} \quad (7)$$

$$W(u, v) \Phi_y(i - uw, j - vw)$$

Kde W je 2D filtr pro dolní propust. Defaultní velikost filtru je 5x5

5. Výpočet vyhlazené lokální orientace se následně provádí pomocí:

$$\mathcal{O}(i, j) = \frac{1}{2} \tan \left(\frac{\Phi'_y(i, j)}{\Phi'_x(i, j)} \right) \quad (8)$$

[6]

Po stanovení orientovaného pole obou otisků prstů je konečně možné se pustit do jejich zarovnání. Metoda se snaží maximalizovat podobnost orientovaných polí v jejich průsečících.

Orientovaná pole \mathcal{O}_1 a \mathcal{O}_2 jsou vypočítána po blocích $w \times w$ a na každé pozici (i, j) obsahuje hodnotu velikosti úhlu θ a $r = [0, 1]$. Poté je podobnost mezi orientovanými poly vypočítána:

$$S(\mathcal{O}^1, \mathcal{O}^2) = \frac{\sum_{(i,j) \in (V_{\mathcal{O}^1} \cap V_{\mathcal{O}^2})} (r_{i,j}^1 + r_{i,j}^2) \cdot \psi(\theta_{i,j}^1, \theta_{i,j}^2)}{\sum_{(i,j) \in (V_{\mathcal{O}^1} \cap V_{\mathcal{O}^2})} (r_{i,j}^1 + r_{i,j}^2)} \quad (9)$$

, kde $\psi(\theta_1, \theta_2)$ je podobnost dvou úhlů z orientovaných polí:

$$\psi(\theta_1, \theta_2) = 1 - \frac{2 \cdot |\theta_1 - \theta_2|}{\pi} \quad (10)$$

a $V_{\mathcal{O}}$ obsahuje pouze souřadnice z popředí:

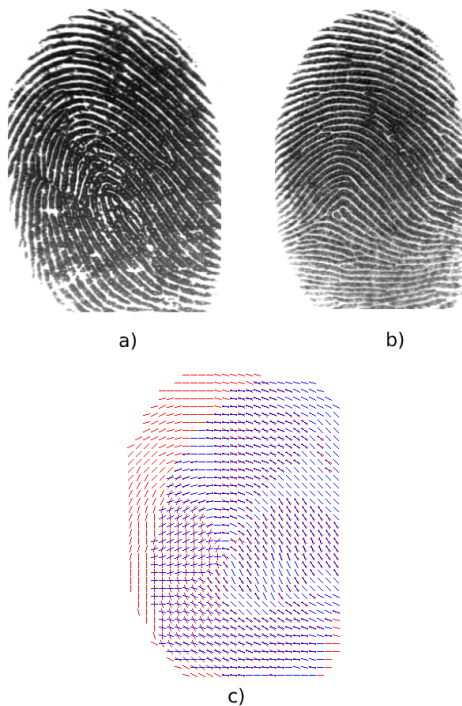
$$V_{\mathcal{O}} = \{(i, j) | o_{i,j} \in \mathcal{O} \wedge r_{i,j} > 0\} \quad (11)$$

Následně pro optimalizaci se přeskakují pozice, dvou orientovaných polí, na kterých se dostatečně nepřekrývají.

$$\frac{|V_{\mathcal{O}^1} \cap V_{\mathcal{O}}|}{\max(|V_{\mathcal{O}^1}|, |V_{\mathcal{O}^2}|)} \geq \min_{VR} \quad (12)$$

V mé implementaci uvažuji pouze natočení v rozsahu $< -45^\circ; 45^\circ >$ a posun o maximálně 50% šířky při posunu vpravo nebo vlevo a 50% výšky při posunu nahoru, nebo dolů. Větší změny vedly buď k nepřesnostem, nebo neměly dopad na výsledek. Dále se snažím pokaždé nasadit menší otisk prstu na větší. Po zarovnání otisků se provede oříznutí částí, které nejsou jejich průnikem.

Příklad výsledku je vidět na obrázku 2.



Obrázek 2: a) otisk 1 b) otisk 2 c) výsledek průniku orientovaných polí

3.2 Stanovení frekvenční charakteristiky otisku

Úrovně šedé v okolí linií mohou být modelovány jako sinusovka, pro případy kde se nevyskytuje v lokálním okolí minutiae nebo singulární bod. Pro tyto případy je lokální frekvence další důležitou složkou obrazu otisku prstu. Mějme \mathcal{G} značící normalizovaný obraz otisku a \mathcal{O} orientované pole vypočítané jako je popsáno v minulé sekci. Poté:

1. Rozdělíme si \mathcal{G} na $w \times w$ bloky, kde $w = w$ z prvního kroku orientovaného pole.
2. Pro každý blok s prostředkem v (i,j) , spočítáme pole $l \times w$, otočené o úhel $\mathcal{O}(i,j)$:

$$X[k] = \frac{1}{w} \sum_{d=0}^{w-1} \mathcal{G}(u, v), \quad k = 0, 1, \dots, l-1 \quad (13)$$

$$u = i + \left(d - \frac{w}{2}\right) \cos \mathcal{O}(i, j) + \left(k - \frac{l}{2}\right) \sin \mathcal{O}(i, j) \quad (14)$$

$$v = j + \left(d - \frac{w}{2}\right) \sin \mathcal{O}(i, j) + \left(k - \frac{l}{2}\right) \cos \mathcal{O}(i, j) \quad (15)$$

3. Pokud se žádné singulární ani minutiae body neobjeví v orientovaném poli, x-signature formuje sinusový tvar vlny. Nechť $\mathcal{T}(i, j)$ je průměrný počet pixelů mezi dvěma vrcholy. Poté je frekvence stanovena jako:

$$\Omega(i, j) = 1/\mathcal{T}(i, j) \quad (16)$$

[6]

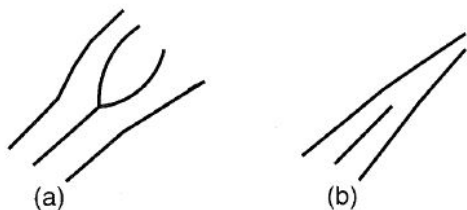
Příklad výsledku je vidět na obrázku 3.



Obrázek 3: Frekvenční charakteristika otisku z obrázku 1

3.3 Minutiae

Linie a údolí se na otisku prstu střídají a proudí jednotným směrem. Federal Bureau of Investigation roku 1984 našel 18 vlastností charakterizujících otisk prstu. Aktuálně známe přes 150 různých lokálních vlastností linií. Tyto lokální vlastnosti nejsou rovnoměrně rozloženy. Detekce většiny z nich převážně záleží na kvalitě snímku otisku a jsou málokdy viděny. Dvě vlastnosti s největším výskytem jsou zakončení linie a rozdvojení linie, viz. obrázek 4 [2]. Hromadně se tyto vlastnosti nazývají minutiae. Většina aplikací pro extrakci a porovnání minutiae používají právě dvě uvedené vlastnosti. Na otisku prstu zaznamenaném v dobré kvalitě může být nalezeno okolo 40 až 100 minutiae.



Obrázek 4: a) zakončení linie, b) rozdvojení linie [2]

Další typy minutiae jsou kombinací dvou uvedených. Každé zakončení linie a rozdvojení linie se skládá ze tří vlastností a to (x,y) souřadnice, kde se nachází s lokální orientace θ . Tato reprezentace otisků prstů je používána z důvodu zmenšení dat v databázi, ale také bezpečnosti, kdy sestavit otisk prstu ze získaných minutiae je prakticky nemožné.

V projektu je použit algoritmus pracující na extrakci minutiae pomocí thinningu. Pro extrakci minutiae je zapotřebí prvně spočítat orientované pole popsané výše. Následně rozlišit pozadí a otisk. Důležitým krokem je extrakce linií, neboli vytvořit černobílý obrázek, kde linie jsou tvořeny černou barvou a údolí s pozadím bílou. Posledním krokem extrakce linií je thinning, kde je extrakce převedena na binární matici, kde linie jsou tvořeny čarami o šířce jeden pixel s hodnotou 1 a pozadí má hodnotu 0. Dosažení thinningu je pomocí použití morfologických operací skeletonizace a vzdálenostní transformace na inverzní výsledek extrakce linií. Následně je výsledek thinningu očištěn od osamocených pixelů. Výsledek thinningu může být vidět v obrázku 5.



Obrázek 5: Výsledek operace thinningu otisku z obrázku 1

Následná extrakce minutiae je provedena pomocí použití metody "crossing number" na okolí bodu P ve vzdálenosti 1. Neboli na jeho 8 okolních pixelů:

$$CN = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}|, \quad P_9 = P_1 \quad (17)$$

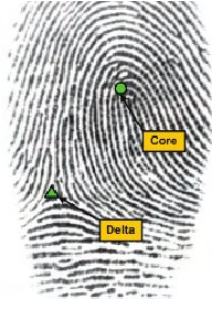
Následně dojde v postprocessingu k odstranění falešných minutiae. První částí je ridge counting. Pomocí ridge counting se určí, kolik linií prochází nebo se dotýká přímky, která je spojnicí mezi jádrem a deltou. Pro tuto operaci je zapotřebí co nejpresněji určit deltu a jádro otisku. Následně:

- Pokud přímka prochází místem rozdvojení linie, poté jsou dvě linie započítány.
- Pokud se přímka dotýká ostrovu (místo mezi dvěma rozdvojeními), nebo tečky, poté je započítána jedna linie
- Údolí musí být přerušeno linií mezi deltou a jádrem, jinak je jedna linie odečtena. (Vztahuje se pouze na linie ve tvaru smyčky.

Následně je vypočítán průměr počtu pixelů mezi dvěma přilehlými liniemi. Neboli je podělen výsledek ridge counting vzdáleností mezi jádrem otisku a deltou. Vzdálenost mezi dvěma minutiae musí být poté větší než tato průměrná hodnota. Algoritmus použitý v projektu pro extrakci minutiae je implementován dle paperu [5].

3.3.1 Jádro a Delta

V použitém algoritmu je ještě zapotřebí popsat jádro otisku a deltu. Jedná se o skupinu high-level vlastností, tzv. singulární body. Otisk prstu může mít žádné nebo více jader. Jádro se vyskytuje u středu otisku prstu a je tvořeno ze "závitu" - whorl, kdy jako jádro je detekován střed závitů, nebo ze "smyčky" - loop, kdy jako core je detekován nejvyšší bod nejnižší smyčky. Delta je poté tvořena liniemi ve směru trojúhelníku, které z ní vyznařují. Delta a jádro jsou vidět na obrázku 6.



Obrázek 6: Ukázka detekované delty a jádra (core) [7]

Pro detekci jádra v otisku byl použit algoritmus založený na metodě Poincaré Index [3], který na základě okolí ve vzdálenosti 1 (8 okolních pixelů) zjistí, zda se jedná o delta / jádro / nic.

4 Cutline

Jedná se o přímku okolo které se vyskytuje nejvíce minutiae a při generování nového otisku zajišťuje, že v otisku bude dostatečné množství minutiae z obou otisků. Přímka se počítá po provedení všech dříve popsaných operací tedy na zarovnaných a oříznutých otiscích od částí neprůniku i s jejich orientovanými poli. Minutiae extrahované z těchto otisků označíme jako T^1 pro otisk 1. a T^2 pro otisk 2.

Následně označíme $\rho = (\rho_x, \rho_y)$ jako barycentrum. Ve článku není uvedeno, o co se přesně jedná, tak jsem použil jádro / deltu otisku, přičemž pokud vypadne při průniku je za barycentrum označen střed otisku.

Poté má přímka l (cutline) parametry:

$$\begin{aligned} a_l \cdot x + b_l \cdot y + c_l &= 0 \\ a_l &= \sin(\beta), \quad b_l = \cos(\beta) \\ c_l &= -\rho_x \cdot \sin(\beta) - \rho_y \cdot \cos(\beta) \end{aligned} \quad (18)$$

Nyní je přímka točena okolo barycentra předurčeným krokem v rozsahu $(0^\circ, 180^\circ)$. Pro každý úhel β je určeno skóre přímky:

$$S_c = \omega_o \cdot S_o + \omega_v \cdot S_v + \omega_m \cdot S_m \quad (19)$$

Kde S_o je podobnost orientovaných polí, S_v je podobnost frekvenčních charakteristik, S_m je skóre, které slouží k ohodnocení vzhledem k generování otisku podobnému oběma na základě nalezených minutiae otisku 1: T_1 , a otisku 2: T_2 . $\omega_o, \omega_v, \omega_m$

jsou poté váhy jednotlivých složek v rozsahu $< 0; 1 >$, $\omega_o + \omega_v + \omega_m = 1$.

$$S_o = \frac{\sum_{(i,j) \in C} (r_{i,j}^1 + r_{i,j}^2) \cdot \psi(\theta_{i,j}^1, \theta_{i,j}^2)}{\sum_{(i,j) \in C} (r_{i,j}^1 + r_{i,j}^2)} \quad (20)$$

$$S_v = \frac{\sum_{(i,j) \in C} \left(1 - \frac{|v_{i,j}^1 - v_{i,j}^2|}{(\max_F - \min_F)} \right)}{|C|} \quad (21)$$

C obsahuje pouze prvky, které jsou od cutline l vzdáleny maximálně d_{max} a jsou průnikem orientovaných polí \hat{O}^1, \hat{O}^2 tedy :

$$C = \{(i, j) | (i, j) \in (V_{\hat{O}^1} \cap V_{\hat{O}^2}) \wedge \text{dist}_l(i, j) \leq d_{max}\} \quad (22)$$

$$\text{dist}_l(x, y) = \frac{|a_l \cdot x + b_l \cdot y + c_l|}{\sqrt{a_l^2 + b_l^2}} \quad (23)$$

$$S_m = \max(\zeta_m(T^1, T^2), \zeta_m(T^2, T^1)) \quad (24)$$

$$\begin{aligned} \zeta_m(A, B) &= \\ \frac{Z(|A|_l^P, \mu_m, \tau_m) + Z(|B|_l^N, \mu_m, \tau_m)}{2} \end{aligned} \quad (25)$$

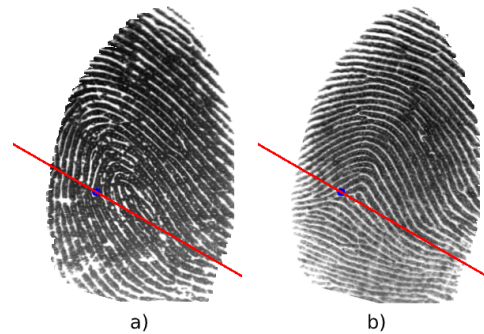
kde $|T|_l^P$ a $|T|_l^N$ označují minutiae, které spadají nad přímku a pod l . Neboli:

$$\begin{aligned} |T|_l^P &= |\{m \in T | \phi_l(m_x, m_y) \geq 0\}| \\ |T|_l^N &= |\{m \in T | \phi_l(m_x, m_y) < 0\}| \end{aligned} \quad (26)$$

Z je sigmoidní funkce kontrolována parametry μ a τ , které zajistí, že výsledek bude vždy mezi 0 a 1:

$$Z(v, \mu, \tau) = \frac{1}{1 + e^{-\tau(v-\mu)}} \quad (27)$$

Jako cutline je označena cutline s nejvyšší hodnotou skóre S_c . [4] Příklad cutline je vidět na obrázku 7.



Obrázek 7: a) cutline na otisku 1 b) cutline na otisku 2

5 Generování otisku dvojí identity 7 Dataset

Zvolil jsem metodu generování otisku na základě obrazu. Otisk je generován pomocí sloučení průniků otisků. Nechť \hat{F}^p je část otisků nad cutline a \hat{F}^n je část otisků pod cutline, které určíme následovně:

$$(p, n) = \begin{cases} (1, 2) & \text{if } \zeta_m(T^1, T^2) \geq \zeta_m(T^2, T^1) \\ (2, 1) & \text{otherwise} \end{cases} \quad (28)$$

Poté je otisk generován následovně:

$$D^I(x, y) = w_{x,y}^{l_{\max}} \cdot \hat{F}^p(x, y) + (1 - w_{x,y}^{l_{\max}}) \cdot \hat{F}^n(x, y) \quad (29)$$

kde $w_{x,y}^{l_{\max}}$ je váha pro vyvážení smíchání otisků v blízkosti cutline.

$$w_{x,y}^{l_{\max}} = \begin{cases} 1 - \max\left(0, \frac{d_{\max} - \text{dist}_{\max}(x, y)}{2 \cdot d_{\max}}\right) & \text{pokud } a_{\ln ax} \cdot x + b_{\ln ax} \cdot y + c_{\ln ax} \geq 0 \\ \max\left(0, \frac{d_{\max} - \text{dist}_{\ln ax}(x, y)}{2 \cdot d_{\max}}\right) & \text{jinak} \end{cases} \quad (30)$$

Výsledek takovéto operace je vidět na obrázku 8.



Obrázek 8: Výsledek morpingu otisků z obrázku 2

6 Implementace

Aplikace je převážně založena na článku [4] a je implementována v jazyce Python verze 3.7.5. Dále byly použity knihovny Numpy pro práci s maticemi, SciPy a OpenCV pro práci s obrazem. Pro thinning a minutiae byla použita knihovna z git repozitáře [8].

Dataset byl použit pro zkušební generování otisků prstů zde popsanou metodou. Bylo využito malé databáze s obsahem 80 otisků prstů Set "B" of Database 1 z [1].

Použitá literatura

- [1] Fvc2004 - third international fingerprint verification competition. <http://bias.csr.unibo.it/fvc2004/download.asp>. (Přistoupeno 05/24/2020).
- [2] Minutiae-based fingerprint extraction and recognition — intechopen. <https://www.intechopen.com/books/biometrics/minutiae-based-fingerprint-extraction-and-recognition>. (Přistoupeno 05/16/2020).
- [3] Ali Ismail Awad and Kensuke Baba. Singular point detection for efficient fingerprint classification. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 1:1–7, 01 2012.
- [4] M. Ferrara, R. Cappelli, and D. Maltoni. On the feasibility of creating double-identity fingerprints. *IEEE Transactions on Information Forensics and Security*, 12(4):892–900, 2017.
- [5] Shohreh Kasaei, Mohamed Deriche, and Boualem Boashash. Fingerprint feature extraction using block-direction on reconstructed images. pages 303 – 306 vol.1, 01 1998.
- [6] Lin Hong, Yifei Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.
- [7] Ishmael Msiza, Jaisheel Mistry, Brain Leke-Betechuoh, Fulufhelo Nelwamondo, and Tshilidzi Marwala. *On the Introduction of Secondary Fingerprint Classification*. 07 2011.
- [8] Dragos Tudor. Fingerprint matcher, June 2019.