

ROZPOZNÁVÁNÍ OTISKŮ PRSTŮ

Petr Dvořáček <xdvora0n@stud.fit.vutbr.cz>

18. Prosinec 2013

Úvod

Tato dokumentace popisuje implementaci biometrického systému, jenž rozpoznává otisky prstů. Jako programovací jazyk jsem zvolil Python. Hlavně kvůli tomu, že mi BIO-Framework padal na řádku s komentářem TODO FIXME.

Implementace

Nejprve dojde k předzpracování dat. Vstupní obrázek se rozdělí na segmenty. V těchto blocích respektive segmentech se pak určí orientace, z níž se získá frekvence papilárních linií. Tyto informace jsou důležité pro aplikaci Gáborových filtrů, které zvýrazní papilární linie. Pak se tento obrázek skeletonizuje. Takovým způsobem, že se binarizuje otisk prstu a ztenčí se zvýrazněné linie. Následně probíhá hledání markantů.

Určení orientace segmentu

Orientaci papilárních linií můžeme zjistit pomocí takzvaných gradientů. Každý segment proženeme Sobelovými filtry, jenž vypadají následovně:

$$A = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Jak použít filtry, abychom dostali orientaci papilárních linií? Díky aplikaci filtru B získám gradient G_x v daném bodě x a y . Aplikací filtru A získám gradient G_y v daném bodě. Úhel θ papilárních linií v segmentu lze dostat pomocí těchto rovnic:

$$G_{xy} = \sum_{i=-1}^1 \sum_{j=-1}^1 G_x(x+i, y+j) G_y(x+i, y+j)$$

$$G_{xx} = \sum_{i=-1}^1 \sum_{j=-1}^1 G_x(x+i, y+j)^2$$

$$G_{yy} = \sum_{i=-1}^1 \sum_{j=-1}^1 G_y(x+i, y+j)^2$$

$$\theta = \frac{\pi}{2} + \frac{2 G_{xy}}{G_{xx} - G_{yy}}$$



1) Použití Sobelova filtru na otisk prstu

Určení frekvence papilárních linií v segmentu

Počet papilárních linií byl získán následně. Nejdříve rotnu obrázek tak, aby papilární linie segmentu byly svisle. Úhel rotace získám z odhadu orientace papilárních linií. Natočený obrázek se projde po řádcích. Na každém řádku určím počet přechodů z tmavé barvy na světlou pomocí prahování. Hodnota prahu se nachází mezi odstínem nejtmařejšího a nejsvětějšího bodu v daném segmentu. Matematicky řečeno:

$$Threshold = \frac{|the\ darkest\ point - the\ lightest\ point|}{2} + the\ darkest\ point$$

Je nutno podotknout, že při otočení se některé body v řádku nevyužívají. Tudíž je zapotřebí každý řádek váhovat a vypočítat váženou průměrnou hodnotu. Frekvence se pak vypočítá pomocí velmi známého vzorce $f = 1/T$.

Obrázek k této části bohužel nemám. Frekvenci segmentu jsem si tiskl na standardní výstup.

Aplikace Gáborových filtrů

Pro každý segment se vytvoří vlastní Gáborův filtr nebo se použije filtr z menší databáze nejčastěji používaných filtrů. Tento filtr má frekvenčně a směrově orientované vlastnosti. To způsobí, že vyjasní papilární linie. Díky tomu můžeme takový obrázek lehce binarizovat. Gáborův filtr můžeme sestavit pomocí vzorce:

$$G(x, y; \theta, f) = \exp\left\{-\frac{(x \cos \theta + y \sin \theta)^2}{\sigma_x^2} - \frac{(y \cos \theta - x \sin \theta)^2}{\sigma_y^2}\right\} \cos(2\pi f(x \cos \theta + y \sin \theta))$$

Kde x a y je velikost filtru, θ je úhel Gáborova filtru, f značí frekvenci kosinusové funkce.

Ovšem může nastat situace taková, že špatně určíme orientaci, či frekvenci v daném segmentu. Jak zachycuje obrázek 2.



2) Vlevo špatné použití Gáborova filtru na otisk prstu. Vpravo správné.

Ztenčení linií

$$C = \begin{bmatrix} 1 & 1 & 1 \\ x & 1 & x \\ 0 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} x & 0 & 0 \\ 1 & 1 & 0 \\ x & 1 & x \end{bmatrix}$$

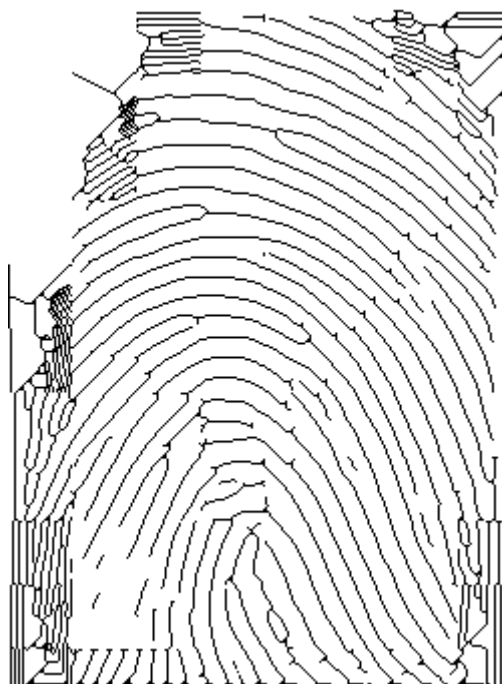
Ztenčení linií probíhá pomocí algoritmu hit-and-miss na binarizovaný obrázek. C a D jsou různě natočené okolí bodu. Může-li být aplikován tento filtr, pak se smaže daný bod. Znak x značí, že na daném bodu může být 1 nebo 0. Skeletonizace se pak provádí tak dlouho, dokud se má, co ztenčovat.

Nalezení markantů

Hledání markantů probíhá obdobně. Nachází-li se v okolí bodu jedna jednička, jedná se o markant ukončení. Nachází-li se v okolí bodu tři jedničky jedná se o markant vidličky. Nachází-li se v okolí bodu jiný počet jedniček, pak je to s největší pravděpodobností papilární linie.

Následující matice zachycují příklady markantů vidličky a konce.

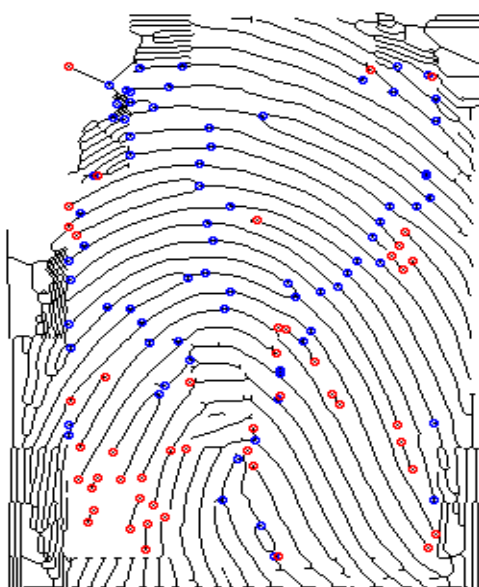
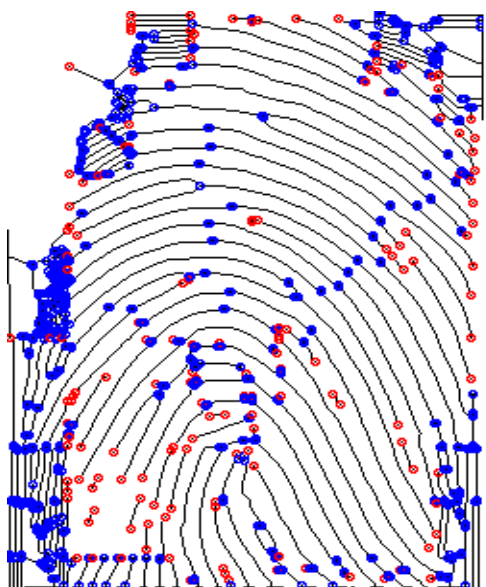
$$\text{Vidlička} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{Konec} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$



3) Ztenčení linií

Redukce markantů

V každém segmentu je vypočtena suma vidliček a konců. Je-li počet markantů 2% vůči ploše segmentu, pak jsou všechny markanty na daném segmentu zanedbány. Jedná se totiž o šum daný neúspěšnou aplikací Gáborova filtru na daný segment. Dále združují blízké vidličky k sobě. Díky těmto filtracím se původních 800 markantů redukovalo na asi 100 markantů, jak můžete vidět na obrázku 4. Ovšem takový počet dimenzí je podle mého názoru vysoký pro klasifikátory.



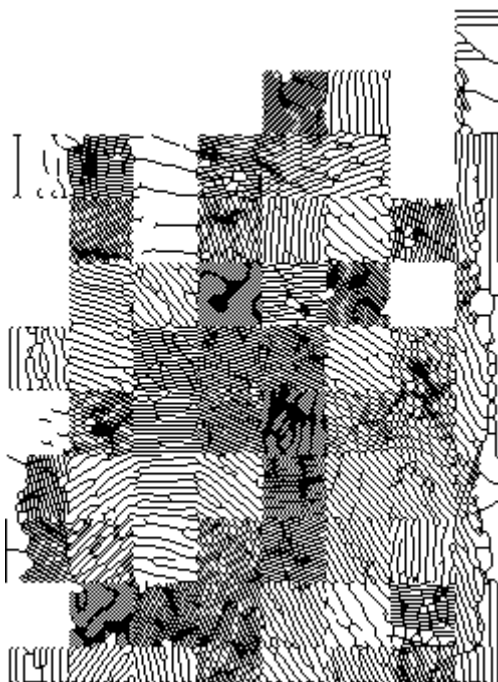
4) Na obrázku vlevo jsou zachyceny všechny markanty. Vpravo jsou tyto markanty redukovány.

Klasifikace

Jako klasifikátor jsem využil support vector machines (SVM). Jako vstupní vektor jsem použil sumu markantů na daných segmentech. V druhém biometrickém systému jsem použil orientaci papilárních linií.

Vyhodnocení biometrického systému

Trvalo to dlouho, avšak o rychlosti pojednává další podkapitola. Kolem 30 obrázků ze 112 bylo po skeletonizaci nevhodných pro další použití. To je nejspíše zapříčiněno tím, že byly špatně použity Gáborovy filtry. Osobně se domnívám, že chyba je v špatném výpočtu frekvence. Každopádně tyto „nevhodné“ obrázky se dále používaly v klasifikaci.



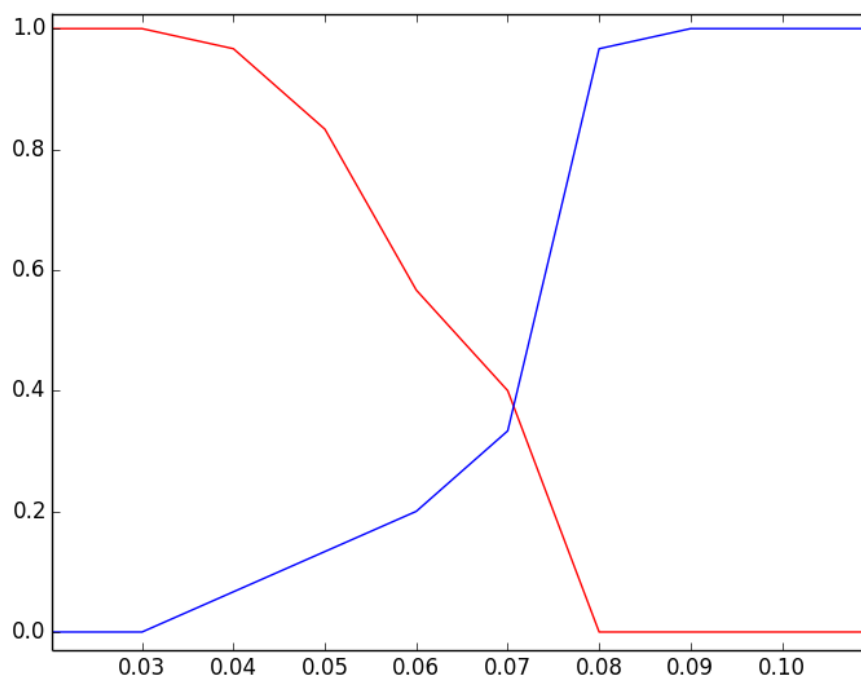
5) Příklad nevhodného skeletonizovaného obrázku.

Vyhodnocení biometrického systému založeného na sumě markantů

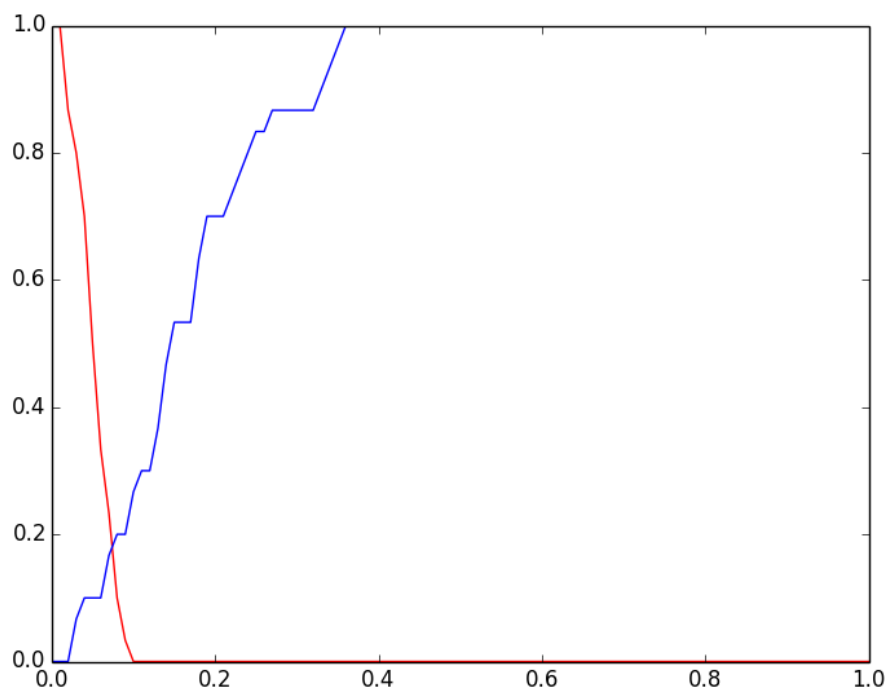
Pro každou třídu (celkem 15) bylo použito 5 trénovacích a 4 data testovací. Dvě data jež spadají do dané třídy a dvě data, která nespádají do dané třídy. Z grafu 6 můžeme zjistit, že EER je přibližně 39% pro práh na 7%. Úspěšnost při identifikaci pak byla 23%, což je lepší než pseudonáhodný generátor čísel.

Vyhodnocení biometrického systému založeného na orientaci papilárních linií

Pro každou třídu (celkem 15) bylo použito 6 trénovacích dat a 4 data testovací. Dvě data byly v dané třídě a dvě data nespádají do dané třídy. Z grafu 7 můžeme zjistit, že EER je přibližně 20% pro práh na 7,5%. Úspěšnost při identifikaci činila 53%.



6) Graf s mírou vyrovnaní chyb pro biometrický systém založený na sumě markantů. Červená křivka - FMR. Modrá křivka - FNMR. Osa x značí míru prahu. Osa y značí skóre.



7) Graf s mírou vyrovnaní chyb pro biometrický systém založený na orientaci papilárních. Červená křivka - FMR. Modrá křivka - FNMR. Osa x značí míru prahu. Osa y značí skóre.

Rychlost Pythonu a algoritmů

Díky pomalému předzpracování dat jsem se rozhodl měřit rychlost. Vytváření specifického Gáborova filtru má za následek to, že výsledné zpracování otisku prstu je pomalejší. Lepší by bylo vytvořit malou databázi několika Gáborových filtrů před spuštěním aplikace. Ztenčování linií ovšem nelze změnit, neboť se provádí iterativně. Viz Implementace - Ztenčení linií.

Průměrná doba segmentace, určení orientace a frekvence	1,21s
Průměrná doba vytváření a aplikace Gáborova filtru	75s
Průměrná doba ztenčování linií	11s
Celková průměrná doba předzpracování	87s

Z tabulky výše můžeme zjistit, že doba předzpracování je pro moderní biometrické systémy nepřijatelné, však pro studijní účely dostatečné. Podotýkám, že extrakce příznaků ze 112 obrázků trvala přibližně 2 hodiny. Proto nebylo možno využít ostatní databáze otisků prstů.

Závěr

Největším problémem při tvorbě projektu byla rychlost Pythonu. Doba strávená při extrakci markantů by se dala redukovat lepším použitím knihoven Scipy a Numpy, či interpretu PyPy. Co se týče implementace programu, tak nejvíce času zabralo vytváření a aplikace Gáborových filtrů. Tato doba by se dala zredukovat takovým způsobem, že by se vytvořila malá databáze nejčastěji používaných filtrů.

Jako další práci na tomto projektu bych viděl v lepším předzpracování markantů. Dále by bylo zajímavé porovnat různé klasifikátory a jejich úspěšnost při problematice rozpoznávání markantů. Rovněž by nebylo na škodu rozšířit databázi o více otisků prstů.

Zdroje

a) David Filla, Biometrie otisku prstu, Diplomová práce 2011

b) Thinning algorithm, <http://fourier.eng.hmc.edu/e161/lectures/morphology/node4.html>

c) Slidy k předmětu Biometrie.

Databáze otisků

- www.advancedsourcecode.com/fingerprintdatabase.asp
Mirror: <http://eva.fit.vutbr.cz/~xdvora0n/bio.zip>
Obsahuje 16 otisků. Jeden otisk je sejmutý osmkrát => Celkem 128 obrázků.

Instalace

Python 2.7.6

Knihovny

- `scipy 0.13.2` - scipy.org
natočení obrázku, při odhadu frekvence papilárních linií
- `numpy 1.8.0` - scipy.org
Prerekvizita knihovny `scipy`.
- `PIL 1.1.7` - effbot.org/zone/pil-index.htm
Manipulace s obrázky.
- `scikit 0.14.1` - scikit-learn.org
SVM
- `matplotlib 1.3.1` - matplotlib.org
Vykreslení grafu
- `python-dateutils` - <http://www.lfd.uci.edu/~gohlke/pythonlibs/>
prerekvizita modulu `matplotlib`
- `pyparsing`
prerekvizita modulu `matplotlib`
- `pytz`
prerekvizita modulu `matplotlib`
- `six`
prerekvizita modulu `matplotlib`

Použití programu

Před použitím vytvořit složku s názvem „db“, ve které je daná databáze.

<code>bio.py</code>	vyhodnotí biometrický systém založený na orientaci papilárních linií
<code>bio2.py</code>	profilování programu (chcete-li změnit obrázek... goto 508 řádek v tomto programu)
<code>bio3.py</code>	vyhodnotí biometrický systém založený na sumě markantů na daných segmentech.