

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁNÍ LIDSKÉ TVÁŘE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL BURIÁN

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁNÍ LIDSKÉ TVÁŘE

HUMAN FACE RECOGNITION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL BURIÁN

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ADAM HEROUT, Ph.D.

BRNO 2010

Abstrakt

Tato bakalářská práce se zabývá problematikou rozpoznávání lidské tváře v obraze. Popisuje principy detekce tváře, metody pro rozpoznávání a existující datové sady tváří. Zaměřuje se na rozbor, srovnání a testování algoritmů eigenface, fisherface a KDDA (Kernel Direct Discriminant Analysis).

Abstract

The bachelor thesis deals with the problematic of the human face recognition in an image. It describes the face detection principles, methods used in recognition problematic and existing face data sets. Thesis is aimed on analysis, comparison and testing of the eigenface, fisherface and KDDA (Kernel Direct Discriminant Analysis) algorithms.

Klíčová slova

Rozpoznání lidské tváře, detekce tváře, eigenface, PCA, analýza hlavních komponent, fisherface, KDDA, databáze tváří.

Keywords

Human face recognition, face detection, eigenface, PCA, principal component analysis, fisherface, KDDA, face database.

Citace

Pavel Burián: Rozpoznání lidské tváře, bakalářská práce, Brno, FIT VUT v Brně, 2010

Rozpoznání lidské tváře

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana
Ing. A. Herouta, Ph.D.

.....

Pavel Burián
18. května 2010

© Pavel Burián, 2010.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Detekce lidské tváře	4
2.1 Statisticky orientované metody	4
2.1.1 Metoda podprostoru	4
2.1.2 Metoda neuronových sítí	4
2.2 Znalostní metody	5
2.2.1 Metody založené na rozpoznávání obličejových obrysů	5
2.2.2 Metody založené na informaci o barvách	6
2.2.3 Metody založené na informaci o pohybu na scéně	6
2.2.4 Metody založené na symetrii	6
2.3 Shrnutí	6
3 Metody pro rozpoznání lidské tváře	7
3.1 Metody založené na rozložení odstínů šedi v obraze	7
3.2 Metody založené na geometrických tvarech a identifikačních markantech	7
3.3 Metoda optických toků	8
3.4 Metoda deformačních modelů	8
3.5 Metody neuronových sítí	9
3.6 Shrnutí	9
4 Vybrané algoritmy	10
4.1 Použité nástroje	10
4.2 Eigenface	10
4.2.1 Implementace	12
4.2.2 Použití	13
4.3 Fisherface	13
4.3.1 Implementace	14
4.3.2 Použití	15
4.4 KDDA	17
4.4.1 Implementace	18
5 Datové sady	20
5.1 Feret	20
5.2 Umist	21
5.3 ORL	22

6	Výsledky	23
6.1	DataMaker	23
6.2	Hodnocení	24
7	Závěr	30
A	Obsah CD	33
B	Tabulky hodnot	34

Kapitola 1

Úvod

Rozpoznání lidské tváře patří do oblasti biometrie, která se zabývá identifikací osob podle fyziologických znaků, jež jsou pro každého člověka jedinečné.

V dnešní době se hojně pracuje na vývoji biometrických systémů. Z velké části se o to zapříčinily teroristické útoky. Tyto systémy jsou pak nasazovány, aby těmto situacím předcházely, či je eliminovaly. Do kategorie biometrie spadá např. porovnávání otisků prstů, oční duhovky, oční sítnice, geometrie ruky, krevního řečiště dlaně i hřbetu ruky, DNA, tvaru ucha, hlasu atd. Většina zmíněných potřebuje ke své činnosti speciální zařízení. Kdybychom však toto zanedbali, nelze opomenout skutečnost, že k těmto systémům je vyžadován aktivní osobní přístup každého jedince, kterému nemusí být proces identifikace příjemný.

Rozpoznání (identifikace) podle lidské tváře nabízí neinvazivní přístup. Snímání pak probíhá fotoaparátem či kamerou, která je nainstalovaná na monitorovaných místech, kterými mohou být letiště, zastávky hromadné dopravy, sportovní haly, stadiony apod. Mnohdy lidé vůbec ani netuší, že jsou zachyceni objektivy a tak nejsou vystaveni možné, s identifikací spojené, stresové situaci.

Biometrický systém založený na rozpoznání lidské tváře není sice dokonalý, ale pro svou jednoduchost stále více uplatňovaný. Příkladem mohou být notebooky s vestavěnou kamerou v displeji, kde lze pro autentizaci do systémů použít místo hesla tvář.

Obecně identifikace nebo ověření totožnosti osob pomocí biometrických systémů nabízí vyšší možnost zabezpečení, než např. čipové karty, klíče apod., které mohou být zneužity při odcizení. Odpadá také nutnost pamatovat si nebo mít bezpečně uložené různé autentizační řetězce.

Tato práce se zpočátku věnuje popisu metod na detekci tváře. V druhé kapitole jsou probrány základní přístupy k rozpoznání lidské tváře. Následující kapitola je věnována algoritmům eigenface, fisherface a KDDA (kernel direct discriminant analysis), které byly vybrány pro srovnání a testování. Práce také poskytuje informace o použitých datových sadách tváří, jež jsou obsahem kapitoly č. 5. Způsob testování a dosažené výsledky lze nalézt v kapitole č. 6.

Kapitola 2

Detekce lidské tváře

Detekce je základním předpokladem pro následné rozpoznání lidské tváře. Jde o proces, v němž se snažíme říci, zda-li se na předloženém obrazu (fotografii) nachází lidská tvář. Pokud ano, dokážeme ji lokalizovat a tuto oblast vyříznout (vyextrahovat) a postoupit dalšímu zpracování, kterým může být ono rozpoznání. Ne vždy totiž platí, že data pro automatické rozpoznání získáváme z portrétů lidí. Nejdříve získáme nějaká surová data pořízená fotoaparátem či kamerou, na nichž se nachází i prostředí, do jehož kontextu jsou tváře zasazeny.

Detekci tváře lze rozdělit do dvou hlavních kategorií, jimiž jsou statisticky orientované metody a znalostní metody, které vychází z [1]. Ukázkou detekce lze vidět na obrázku 2.1, který byl vytvořen programem *facetedetect*, coby hotovým příkladem knihovny OpenCV (více o knihovně v části 4.1).

2.1 Statisticky orientované metody

2.1.1 Metoda podprostoru

Metoda podprostoru je založena na hledání markantních charakteristik (tzv. markantů), které jsou typické pro lidskou tvář (např. oči, ústa, nos, uši).

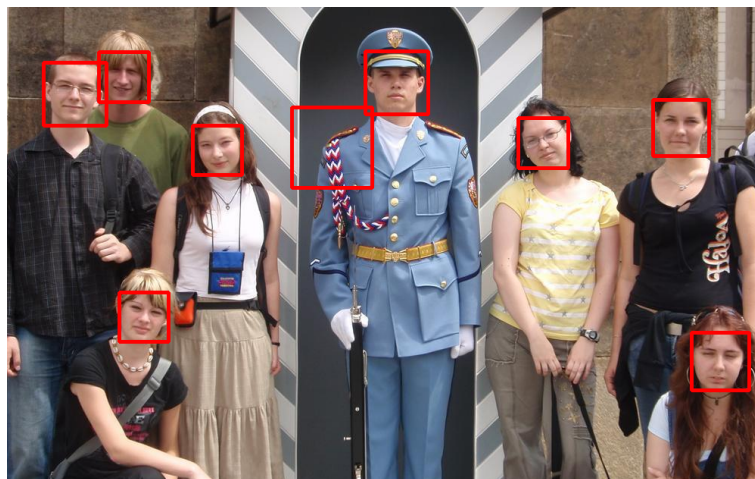
Každý 2D obraz lidské tváře si lze představit jako vícerozměrný vektor (např. obraz o velikosti 92×112 jako vektor $1 \times (92 * 112)$). Pokud se zmíněné charakteristiky nachází ve společném podprostoru daného vícerozměrného prostoru, lze detekci tváře považovat za detekci tohoto podprostoru.

Pro detekci se používá Karhunen-Loeve transformace, kde se setkáváme s normalizovaným obrazem tváře - eigenface.

Normalizace obrazu je proces, během kterého může být provedena změna měřítka obrazu, převedení barevného obrazu do odstínů šedi, redukce bitové hloubky, úprava jasu či nalezení význačných bodů (např. oči), od kterých se odvíjí orientace tváře v prostoru. Všechny takto normalizované obrazy pak mají stejné vlastnosti, a tudíž je lze efektivně porovnávat. Normalizované tváře mohou měnit svůj vzhled pomocí různých doplňků nebo měnit výraz obličeje. Více o eigenface lze nalézt v části 4.2 zabývající se rozpoznáváním lidské tváře.

2.1.2 Metoda neuronových sítí

Metoda neuronových sítí si bere za cíl naučit se rozpoznávat obrazy, na nichž se buď nalézají nebo nenalézají lidská tvář. Tato síť by měla s jistotou rozeznat obě kategorie a proto se



Obrázek 2.1: Detekce lidské tváře v obraze.

trénuje na specifické obrazy každé z nich. U dané metody však vyvstává problém rozpoznat obrazy, na kterých tvář není. Je to dáno tím, že takových je mnoho. Úspěšnost detekce tváře pak vychází ze schopnosti rozpoznávat tyto obrazy. Tudíž je důležité zvolit správný reprezentativní vzorek trénovacích dat.

2.2 Znalostní metody

Metody založené na rozložení odstínu šedi v obraze

Metoda vychází z obecných pravidel distribuce šedé barvy v obraze. I když dva jedinci mají rozdílný vzhled, tak lze vidět, že oblast očí je vždy tmavší než například čelo.

Jako nejznámější metodu můžeme uvést metodu mozaiky, která je založena na přirozeném způsobu rozpoznání tváře lidským mozkiem. Zpracováváný obraz rozdělí na bloky 4×4 . Pokud se v tomto obraze vyskytuje lidská tvář, pak se v definovaných blocích nacházejí identifikující markanty tváře (očí, ústa, nos). Ty lze určit z charakteristické distribuce šedé barvy. Hledáme tedy pouze bloky s markanty, které následně opět rozdělíme, ale ještě detailněji na větší počet bloků (8×8). Pro určení pozice markantů v blocích lze využít metod detekce hran.

Protože snímky lidských tváří nejsou vždy pořizovány za konstantního osvětlení, ovlivňují vrhané stíny samotnou distribuci odstínů šedé barvy. Bloky se pak zpracovávají pomocí adaptivního přizpůsobování.

Metodu mozaiky lze upravit, aby obraz dělila do bloků 3×3 , čímž se zvýší efektivita zpracování. I tak je ale pomalá a proto se kombinuje s dalšími metodami za účelem snížení doby zpracování.

2.2.1 Metody založené na rozpoznávání obličejových obrysů

Tento způsob detekce je založen na nalezení obrysů (kontur) tváře, jež jsou další důležitou charakteristikou. Nejčastěji se využívá v kombinaci s jinými metodami. Nejdříve nalezneme jednotlivé části obličeje (oči, nos, ústa) pomocí detekce kontur a tyto charakteristiky slouží jako vstup následující metodě.

Dalšími způsoby, jak nalézt obrysy či kontury, je využití prahování, detekce hran, segmentace narůstáním oblastí apod.

2.2.2 Metody založené na informaci o barvách

Takové metody si kladou za cíl nalézt obličej v obraze podle jeho barev. Sice všichni lidé nemají totožnou barvu kůže obličeje, ale platí určité zásady, díky kterým lze odlišit obraz bez a s tváří. Pro každou lidskou rasu je rozložení barev v obličejí velice podobné. Můžeme zde tedy najít typické oblasti s určitou barvou. Pokud se podíváme na oblast očí, jsou charakterizovány tmavší barvou (barvou stínů). Oblasti nosu mohou vévodit světlejší barvy a lze nalézt jeho ohraničení tvořené stínem.

Metody založené na informaci o barvách pracují efektivně za dobrých světelných podmínek. Barvy je těžké rozlišovat při velice jasném nebo naopak temném osvětlení a při měnícím se úhlu osvětlení.

2.2.3 Metody založené na informaci o pohybu na scéně

Již v 60. letech 20. století se taková metoda detekce objektů používala ve vojenství např. pro sledování přiletů a odletů letounů. Informaci o pohybu na scéně lze také využít pro detekci a lokalizaci tváře. K tomuto potřebujeme časovou sekvenci snímků pohybujících se osob vůči pozadí. Do této kategorie patří metoda optických toků (optic flow). Ta krom lokalizace tváře nabízí také její rozpoznání.

Detekce tváře založené na pohybu mají však svá úskalí a omezení. Problémem je, když se tváře překrývají nebo jsou zakryty jinými objekty. Dalším omezujícím faktorem je různá rychlost pohybu osob na scéně.

2.2.4 Metody založené na symetrii

Můžeme říct, že lidská tvář je do určité míry symetrická. Pak hledáme v obraze symetrické obrazce, které odpovídají charakteristikám lidské tváře.

Existuje více přístupů jak nalézt tyto symetrie. Jedním je rozdělení obrazu do kruhových oblastí, v nichž se zkoumá, zda-li jsou totožné se symetrickou tváří. Dalším způsobem je využití symetrické transformace, pomocí níž je možné popsat symetrické tvary.

2.3 Shrnutí

Výčet výše uvedených metod není úplný. Existuje mnoho variací pro jednotlivé přístupy a každý má své kladné, ale i záporné vlastnosti. Chybnost detekce byla také vidět na obr. 2.1. Sice byly veškeré tváře detekovány, ale použitá metoda také chybně označila jednu oblast, na níž se tvář nevyskytuje.

V praxi se pak tedy nejčastěji využívají kombinace více metod pro zlepšení výsledků detekce.

Problematika detekce lidské tváře nenachází uplatnění jen v komplexních zabezpečovacích či autentizačních systémech. Široká veřejnost má možnost se s touto technologií setkat např. v digitálních fotoaparátech pro automatické zaostřování na tvář.

Kapitola 3

Metody pro rozpoznání lidské tváře

Pokud jsme získaly obrazy lidských tváří pomocí jejich detekce na scéně nebo jiným způsobem, jakým může být cílené pořizování portrétů osob, můžeme přejít k jejich rozpoznávání. V následujícím textu jsou popsány jednotlivé metody, vycházející z [1].

3.1 Metody založené na rozložení odstínů šedi v obraze

Jako tomu bylo u detekce tváře v 2.2, i sem patří metoda mozaiky, která je však již použita pro rozpoznávání tváří. Opět rozdělí obraz do několika bloků. To provede nad rozpoznávanou (neznámou, testovací) tváří a stejným způsobem i nad známou (trénovací) tváří. Poté probíhá porovnávání vzájemně odpovídajících si bloků a zkoumá se jejich bezprostřední okolí. K tomuto se využívá n -rozměrných vektorů. Když se provede veškeré porovnání, lze říct, zda oba obrazy tváří patří stejné osobě.

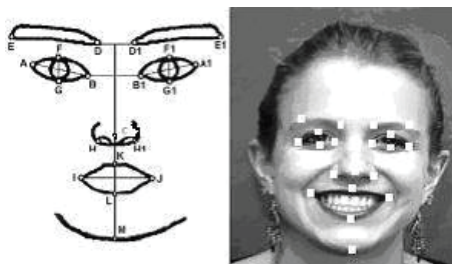
Metoda je výpočetně náročná a její rychlost a úspěšnost ovlivňuje počet známých tváří.

3.2 Metody založené na geometrických tvarech a identifikačních markantech

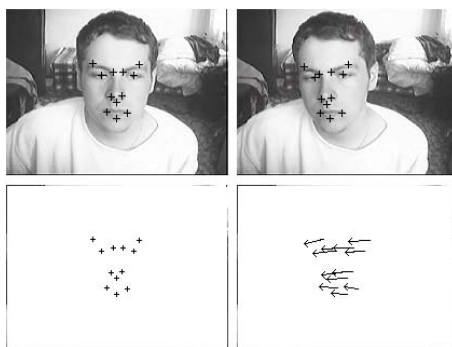
Tyto metody vycházejí z geometrických a vizuálních charakteristik, založených na antropometrických poznacích, a detekce geometrických rysů obličeje. Ty velkou vahou identifikují jednotlivé osoby. Mezi charakteristiky patří vzdálenosti a úhly mezi identifikačními body.

Základní antropometrické body jsou definovány vnitřními a vnějšími koutky očí, vnějšími horizontálními body rtů, bodem přechodu nosu v čelo a bodem spodní hrany nosu. Vyznačené antropometrické body lze vidět na obrázku 3.1.

Metody této skupiny dominují ve forenzní analýze a jsou stále využívány bezpečnostními složkami. Potýkají se však s problémem při rozpoznávání tváře z fotografie, pořízené za špatných světelných podmínek. Při různém úhlu pozorování tváře také dochází k zakrytí markantů, čímž se redukuje jejich počet. Některé charakteristiky se stávají nezajímavými, neboť je nelze zjistit přes konstantní zakrytí. Tomu odpovídá oblast uší.



Obrázek 3.1: Obličejové markanty. Obraz převzat z [2].



Obrázek 3.2: Optický tok markantních částí obličeje charakterizovaný vektorem. Obraz je převzat z [3].

3.3 Metoda optických toků

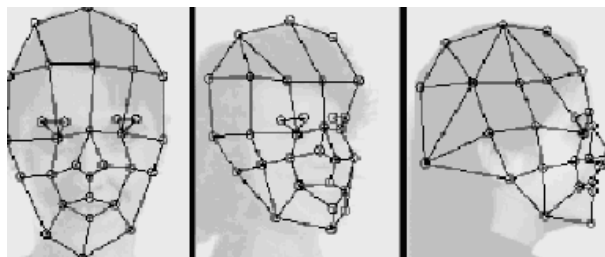
Opět jako u detekce je potřeba mít k dispozici sekvenci snímků hlavy v čase. Pokud srovnáme dva snímky jdoucí po sobě, nalezneme určité dynamické změny. Těmi jsou změna světelné intenzity vzájemně odpovídajících si bodů a jejich pohyb v prostoru. Z těchto změn lze určit směr pohybu. Můžeme tedy říci, že v obraze probíhají změny texturální (změny intenzity) a strukturální (změny prostorové).

Každá osoba má rozlišené umístění markantních bodů a jejich optické toky dané rozdílem dvou snímků určují jejich emocionální výraz tváře. Ty lze využít pro účely rozpoznávání. Jak takový optický tok vypadá můžeme vidět na obrázku 3.2.

3.4 Metoda deformačních modelů

Jedná se o metodu pracující v 3D prostoru, která využívá zakřivení invariantního (neměnného) objektu. Pokud povrch tváře pokryjeme sítí horizontálních a vertikálních čar, tak je tato síť při pohledu z větší vzdálenosti zakřivena. Protože povrch tváře netvoří rovinu, hustota čar přiložené sítě bude pro různé části obličeje odlišná. Podle měnícího se obrazu se také mění tyto křivky a vzdálenosti nebo úhly mezi nimi. Změnou může být proměnný výraz tváře.

Pokud uvedeného chceme využít pro rozpoznávání tváří, jedná se o elastické srovnávání dvou 3D zakřivených povrchů - algoritmus EBGM (Elastic Bunch Graph Matching). U každé osoby musí být známy určité změny povrchu její tváře, na základě kterých ji lze charakterizovat.



Obrázek 3.3: Elastická síť [4].

3.5 Metody neuronových sítí

Neuronové sítě (ANN, artificial neural network) nachází uplatnění, pokud nejsme schopni určit jakou metodou řešit určitý problém nebo pokud je vyžadována práce s porušenými či neúplnými daty (těch je většinou značné množství). Jedná se totiž o systém, který se dokáže sám učit.

Jejím vzorem je lidský mozek, neboť znalosti získává právě učením a tyto ukládá do spojení mezi jednotlivými neurony.

První neuronová síť se objevila již roku 1958, jejímž tvůrcem byl psycholog Frank Rosenblatt a dostala jméno Perceptron. Měla modelovat způsob činnosti lidského mozku při rozeznávání objektů.

Tuto síť tvoří, jak z názvu vyplývá, neurony, které plní funkci jako v lidském mozku. Obsahuje mnoho vstupních signálů a na základě vnitřního systému vytváří signál výstupní, zasílaný jiným neuronům.

Celek neuronů tvoří hierarchickou strukturu, v níž nelze předvídat tok dat.

Podle způsobu školení je možno neuronové sítě rozdělit na samoorganizující se a zpětně šířící. Vždy se nejdříve všechny neurony náhodně inicializují. Princip první z nich spočívá ve vystavování velkému množství dat za účelem nalezení souvislostí v datech. Zpětně šířící se neuronové sítě vyžadují spolupráci tzv. učitele, který vyhodnocuje její výstup. Podle toho, zda-li je správný či naopak chybný, se upravují neuronové váhy.

Neuronové sítě lze kombinovat s jinými metodami, které nejprve najdou charakteristiky obličeje a teprve poté je síť použita pro klasifikaci. Není to však základní předpoklad, neboť neuronové sítě mohou funkci těchto metod zastoupit.

3.6 Shrnutí

Uvedené metody nejsou jediné používané. Existují další způsoby jak se postavit k rozpoznávání tváří. Každý přístup přináší své výhody, ale i nedostatky vycházející z principu jeho činnosti a vyznačuje se svými charakteristickými vlastnostmi a rozdílným procentem úspěšnosti. V praxi se pak často využívají kombinace těchto metod, díky kterým je možné dosáhnout lepších výsledků.

Kapitola 4

Vybrané algoritmy

Tato kapitola je zaměřena na přiblížení algoritmů, které byly zvoleny pro účely srovnání a testování. Jako první byl neodmyslitelně vybrán *Eigenface* (PCA - Principal component analysis - analýza hlavních komponent), který lze považovat za základní algoritmus na poli rozpoznávání. Dále byl zvolen algoritmus *Fisherface*, jehož základem je LDA (Linear Discriminant Analysis - Lineární diskriminační analýza). Třetím a posledním je KDDA (Kernel Direct Discriminant Analysis), který byl zvolen pro implementaci.

Všechny uvedené lze rozdělit do dvou stěžejních částí, jimiž jsou učení a rozpoznávání. Toto hrubé rozdělení obecně platí pro všechny algoritmy. V obou fázích se pracuje s obrazy tváří. Množinu z nich, které zahrneme do fáze učení (trénování), nazvěme trénovací data a ty, jež zahrneme do rozpoznávání (testování) - testovací data.

4.1 Použité nástroje

Pro implementaci byla využita svobodná a otevřená multiplatformní knihovna pro manipulaci s obrazem - OpenCV (Open Source Computer Vision) verze 2.0, která byla původně vyvíjena firmou Intel. Obsahuje více jak 500 optimalizovaných algoritmů pro počítačové vidění a zpracování obrazu v reálném čase. Knihovnu lze využít zdarma jak pro nekomerční, tak ale i pro komerční účely [5].

Dalšími nástroji jsou QT Toolkit verze 4.5.2, v kterém je implementace realizována. Matlab, který byl využit pro algoritmus Fisherface, který je napsán v kódu pro tento nástroj a Microsoft Visual Studio 2008.

4.2 Eigenface

O tento algoritmus se zasadili pánové Turk a Pentland, kteří jej popsali již v roce 1991 [6]. Nyní tvoří základ i pro jiné metody či je jimi využíván.

Eigenface (též PCA) patří mezi základní a jednoduché algoritmy pro rozpoznání lidské tváře a jak již bylo zmíněno, i pro detekci.

Nejprve se podívejme na tento algoritmus obecněji. Eigenface snižuje rozměrnost obrazů. Rozměrnost je rovna počtu pixelů v daném obrazu. Snižování probíhá během fáze učení, jejíž výsledkem je reprezentace, která se nazývá podprostor. I během fáze rozpoznávání se redukuje rozměrnost obrazu (testovacích dat) jeho projekcí (promítnutím) do zjištěného podprostoru. Následně vypočítáme vzdálenost mezi tímto obrazem (testovacími daty) a

trénovacími daty. Vybereme nejmenší vzdálenost a řekneme, že daný obraz tváře patří osobě, která je v tomto podprostoru nejbližší.

Nyní se podíváme na bližší popis, který lze najít v [6].

Mějme obraz tváře o velikosti $v \times s$. Ten lze považovat za sloupcový vektor dimenze n o velikosti součinu jeho rozměrů v pixelech. Takové vektory pak představují body v daném vícerozměrném prostoru, jejichž distribuce není náhodná. PCA pak následně hledá vektory, které splňují podmínku nejlepší distribuce obrazů tváří v celém prostoru obrazů. Získané vektory, které jsou lineární kombinací původních obrazů tváří, tvoří podprostor, tzv. prostor obrazů tváří. Název Eigenface získaly tyto vektory na základě podobnosti obrazům tváří a také proto, že jsou vlastními vektory kovarianční matice odpovídající původním obrazům tváří.

Zvolme trénovací sadu obrazů tváří jako $\Gamma_1, \Gamma_2, \Gamma_3 \dots \Gamma_M$. Nejdříve vypočítáme průměrnou tvář sady, kterou získáme aritmetickým průměrem tváří přes jejich všechny obrazové body:

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (4.1)$$

Jak se každá tvář liší od průměrné, zjistíme jejich vzájemným rozdílem:

$$\Phi_i = \Gamma_i - \Psi \quad (4.2)$$

Získaná množina vektorů je vstupem pro analýzu hlavních komponent, která pro popis distribuce dat hledá množinu M ortonormálních vektorů u_n .

Vlastní vektory a hodnoty lze zjistit z kovarianční matice:

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = AA^T \quad (4.3)$$

kde matice $A = [\Phi_1 \Phi_2 \dots \Phi_M]$ má rozměr $n \times M$ a matice C je symetrická o velikosti $n \times n$, která určuje n vlastních vektorů a hodnot. Většinou se setkáváme s případem, kdy dimenze prostoru je mnohem větší než počet předložených obrazů ($M \ll n$) a je dokázáno, že matice C bude obsahovat jen $M - 1$ významných vlastních vektorů. Nemusíme počítat vektory z matice o velikosti $n \times n$, ale stačí je vyřešit z matice L o velikosti $M \times M$. $L = A^T A$, kde $L_{mn} = \Phi_m^T \Phi_n$ a nalezneme M vlastních vektorů v_l matice L . Odpovídající vlastní hodnoty nám umožní seřadit vlastní vektory podle charakteristické změny mezi obrazy. Nyní můžeme vypočítat vlastní tváře (eigenfaces) u_l :

$$u_l = \sum_{k=1}^M v_{lk} \Phi_k \quad (4.4)$$

kde $l = 1, \dots, M$. Pokud tedy chceme klasifikovat obraz tváře (Γ), tak jej projekcí transformujeme do podprostoru:

$$\omega_k = u_k^T (\Gamma - \Psi) \quad (4.5)$$

pro $k = 1, \dots, M'$, kde M' je počet největších vlastních čísel. $\Omega^T = [\omega_1, \omega_2 \dots \omega_{M'}]$ - vektor vah, který udává přínos každé eigenface reprezentující vstupní obraz tváře. Tento vektor pak může být použit pro nalezení třídy obličeje (osoby) k , která nejlépe popisuje tvář a má nejmenší *Euklidovskou vzdálenost*.

$$\epsilon_k^2 = \|\Omega - \Omega_k\|^2 \quad (4.6)$$

4.2.1 Implementace

Tento algoritmus, který je k nalezení na [7], je napsán v jazyce C s využitím knihovny OpenCV. Jedná se o konzolovou aplikaci. Celá implementace je obsažena v jediném modulu *eigenface.c*.

Program vyžaduje, aby všechny obrazy měly stejnou velikost a také přítomnost dvou souborů - *train.txt* a *test.txt*. Ty obsahují řádky, které mají následující formát:

```
<číslo osoby> <cesta k obrazu>
např.: 2 C:/data/beh1/2/test/s1/1.pgm
```

Jsou to tedy seznamy obrazů, které budou použity pro trénování (*train.txt*) a testování (*test.txt*).

Hlavní program *main()* pouze zpracovává vstupní parametry a podle jejich hodnoty volá funkce *learn()* - pro učení a *recognize()* - pro testování.

Ve fázi učení jsou nejdříve načteny obrazy jednotlivých tváří do globální proměnné *faceImgArr* (*loadFaceImgArray()*) a také čísla osob na základě obsahu *train.txt*. Poté je volána funkce *doPCA()*, která nalezne PCA podprostor. Ta využívá vestavěné funkce OpenCV *cvCalcEigenObjects()*, jež přijímá celkem 9 parametrů.

- *nObjects* - počet zdrojových objektů (obrazů)
- *input* - ukazatel na pole vstupních obrazů typu *IplImage*
- *output* - ukazatel na pole objektů (vlastních vektorů)
- *ioFlags* - vstupní/výstupní flag
- *ioBufSize* - vstupní/výstupní velikost bufferu v bytech
- *userData* - ukazatel na strukturu obsahující data po návratu z funkce
- *calcLimit* - kritérium zastavení výpočtu vlastních objektů
- *avg* - průměrný objekt
- *eigVals* - ukazatel na pole vlastních hodnot seřazených sestupně [8]

Po výpočtu podprostoru se do něj provádí projekce trénovacích dat. K tomuto účelu je opět využita funkce OpenCV - *cvEigenDecomposite()*. Před ukončením trénovací fáze je volána funkce *storeTrainingData()*, která do souboru *facedata.xml* uloží počet vlastních čísel, počet trénovacích dat (fotografií), matici obsahující čísla, která charakterizují adresáře trénovacích dat, vektor vlastních hodnot, matici projekcí trénovacích dat a průměrný obraz tváře.

Výše zmíněná funkce pro fázi testování - *recognize()*, provádí načtení testovací sady tváří. Pomocí funkce *loadTrainingData()* opět načte všechna uložená data z trénovací fáze a znovu využije *cvEigenDecomposite()* pro projekci testovacích dat do podprostoru.

Nalezení nejmenší vzdálenosti testovací tváře od trénovacích je realizována metodou nejbližšího souseda a nachází se ve funkci *findNearestNeighbor()*. Výstupem programu je výpis řádku pro každý testovací obraz, obsahující číslo (příp. název) adresáře nejbližší trénovací tváře a číslo adresáře, kam testovací tvář skutečně patří. V případě shody je tvář správně klasifikována. Pro rychlý přehled výsledku bylo do této části připojeno počítadlo.

4.2.2 Použití

Pro překlad zdrojového kódu byl použit nástroj Microsoft Visual Studio 2008, v kterém je potřeba nastavit pracovní cesty ke knihovnám a hlavičkovým souborům OpenCV. Hlavičkové soubory lze nastavit v:

Tools->Options->Projects and Solutions->VC++ Directories->Include files
např. C:\OpenCV2.0\include\opencv

Statické knihovny:

Tools->Options->Projects and Solutions->VC++ Directories->Library files
např. C:\OpenCV2.0\lib\Release

Pro samotný projekt je ještě potřeba doplnit *cv200.lib cxcvcore200.lib highgui200.lib* do:

Project->Properties->Linker->Input->Additional dependencies

Následné použití je jednoduché. Aplikace přijímá jeden parametr, kterým může být *train* nebo *test*. V konzoli lze tedy zadat:

- *eigenface train* - spustí fázi tréningu,
- *eigenface test* - spustí fázi testu.

Soubory *train.txt* a *test.txt* musí být přítomny před spuštěním. Pro jejich automatizovanou tvorbu byl vytvořen program *DataMaker*, který bude popsán částí 6.1.

4.3 Fisherface

Tato metoda, která je založena na lineární diskriminační analýze, vybírá matici ortonormálních sloupcových vektorů W tak, že poměr rozptylu mezi třídami a rozptylu uvnitř tříd je maximalizovaný. Matice rozptylu mezi třídami je definována jako

$$S_B = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (4.7)$$

a matice rozptylu uvnitř tříd je definována

$$S_W = \sum_{i=1}^C \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \quad (4.8)$$

kde μ_i je střední obraz třídy X_i , μ je střední obraz celé trénovací sady a N_i je počet vzorků ve třídě X_i . Je-li matice S_W regulární (čtvercová matice, jejíž determinant je různý od nuly a její sloupce či řádky jsou lineárně nezávislé [9]), je optimální projekce W_{opt} volena jako matice ortonormálních vektorů, která maximalizuje poměr determinantů matic S_B a S_W .

$$W_{opt} = \operatorname{argmax} \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1 w_2 \dots w_m] \quad (4.9)$$

kde $\{w_i | i = 1, 2, \dots, m\}$ je množina zobecněných vlastních vektorů S_B a S_W , které odpovídají m největším vlastním hodnotám $\{\lambda_i | i = 1, 2, \dots, m\}$

$$S_B w_i = \lambda_i S_W w_i \quad (4.10)$$

kde $i = 1, 2, \dots, m$. Pokud označíme počet tříd jako C , nabývá m hodnoty $C - 1$.

Metoda má ovšem problém. Matice S_W je vždy singulární (není regulární). To je způsobeno tím, že počet obrazů tváří z trénovací sady je menší než jejich dimenze. Může se tedy stát, že vybraná matice W bude obsahovat vzorky, které budou nulové v rámci rozptylu uvnitř tříd. Řešením je projekce obrazu do nižšího dimenzionálního prostoru, který získáme použitím PCA ke snížení rozměrnosti [10]. Matice W tedy bude obsahovat vypočítané vlastní vektory z trénovacích dat pomocí techniky PCA (Eigenface, viz. 4.2). Z matic $W^T S_B W$ a $W^T S_W W$ se pak vypočítají zobecněné vlastní vektory, které se seřadí sestupně podle hodnot zobecněných vlastních čísel.

Projekce dat (trénovacích i testovacích) se provádí do této množiny vektorů a pro klasifikaci lze využít metodu nejbližšího souseda, jako tomu bylo u Eigenface.

4.3.1 Implementace

Tento algoritmus pro rozpoznání lidské tváře je napsán v kódu pro Matlab. Implementace, kterou lze nalézt na [11], obsahuje dva soubory - *RunFaceRecog.v* a *subfigure.m*.

RunFaceRecog.v tvoří celé jádro funkčnosti, do kterého spadá načítání trénovacích a testovacích dat, učení algoritmu (trénování) na zvolených tvářích, následné testování a výpis výsledku. V jednotlivých fázích probíhá zobrazování grafických informací, o kterých si povíme v následujícím textu.

Při spuštění je nejdříve volána funkce *RunFaceRecog*. Obsahuje parametry běhu algoritmu, které je možné editovat a dosáhnout tak požadovaných vlastností. Proměnná *frAlgorithm* obsahuje textový řetězec a může nabývat dvou hodnot - *eigen* - pro rozpoznávání lidských tváří bude použit algoritmus eigenface, či *fisher* - bude použit fisherface. Pro naše účely je tedy nastavena druhá možnost. Pro manipulaci se zobrazováním grafických informací (graf, fotografie) je přítomna struktura *plots*. Obsahuje:

- *intermediateOn* - zobrazení trénovací a testovací sady, průměrné tváře, eigenfaces, fisherfaces, ...
- *finalOn* - zobrazení výsledků testování
- *savePlotsOn* - nastavení ukládání grafických informací

Aktivaci požadované funkce lze provést nastavením proměnné na hodnotu "1". Parametr *finalOn* doporučuji při velké testovací sadě tváří deaktivovat, neboť by se zobrazilo značné množství grafických výsledků.

Nastavení cesty k databázi tváří je provedeno v *basePath*. Pro načítání obrazů je nutné uvést jejich formát v proměnné *imType*. Tato hodnota není neměnná, neboť zvolené datové sady pro testování obsahují fotografie ve formátu PGM a JPG. V následující části probíhá načítání obrazů trénovací a testovací sady. Náhled na první sadu je vidět na obrázku 4.1, druhá není uvedena pro rozměrnost obrazu obsahující 320 fotografií.

Dále *RunFaceRecog* pokračuje voláním funkcí *computeEigenfaces*, *computeFisherfaces* a *classifyFaces* a zobrazí výsledek testování, jak je uvedeno na obrázku 4.2. O layout (rozvržení) fotografií na obrázku se stará modul *subfigure.m*. V levém sloupci jsou zobrazeny



Obrázek 4.1: Trénovací sada fotografií databáze ORL - 2 pro každou osobu.

tváře z testovací sady a ke každé z nich jsou ukázány 3 nejbližší z trénovací sady. Nápisy nad fotografiemi jsou jejich názvy a barva písma vyjadřuje, zda-li tyto nejbližší sousedé patří do stejné třídy jako testovací tvář. Pokud ano, jsou modré, v opačném případě červené. Jako stěžejní fotografie, podle které je určen výsledek, se bere v úvahu první z dané trojice. Z ostatních lze usuzovat jakousi míru jistoty, že algoritmus správně rozezná vstupní (testovanou) tvář.

Před ukončením programu vzniká v lokálním adresáři se zdrojovými kódy soubor *workspace_dump.mat*, kde jsou uloženy všechny proměnné, použité za běhu programu.

Funkce *computeEigenfaces* realizuje analýzu hlavních komponent (PCA). Jak bylo uvedeno výše, je využita pro snížení dimenze prostoru. Vypočítá matici vlastních tváří z trénovacích dat, která slouží jako vstupní parametr funkce *computeFisherfaces*. Dalším jejím vstupem je struktura obsahující matici trénovacích dat a počet tříd (osob) sady. Výsledkem je matice P vlastních vektorů - fisherfaces, jež je na obrázku 4.3 a matice projekcí trénovacích dat do P uložená v *train.wt*.

Funkce *classifyFaces* zajišťuje samotnou klasifikaci. Pro svou činnost vyžaduje strukturu obsahující matici testovacích tváří (*recog*), strukturu obsahující matici trénovacích tváří (*train*) a matici vypočítaných vlastních vektorů, do kterých provádí projekci testovacích dat.

Určení správnosti rozpoznání lidské tváře probíhá na základě srovnání textových řetězců názvů testovacího obrazu a obrazu k němu nejbližšímu (pomocí Euklidovské vzdálenosti). Program byl modifikován pro práci se specifickými názvy fotografií. Ty tvoří čtyřciferné číslo, které je jedinečné pro každou osobu, následované libovolným řetězcem pro odlišení tváří v rámci jedné osoby. Ten již na správnou funkčnost nemá vliv (např. 1001-1.pgm).

4.3.2 Použití

Pro testování algoritmu je potřeba připravit si data. Všechny fotografie musí mít stejnou obrazovou velikost a musí mít specifický formát názvu, který byl zmíněn v části 4.3.1, tj. název začínající čtyřciferným číslem, jedinečným pro každou osobu. Trénovací data jsou



Obrázek 4.2: Výsledek testování algoritmu fisherface.



Obrázek 4.3: Fisherfaces.

uložena v adresáři *training* a testovací v adresáři *test*. Ve zdrojovém kódu je nutné uvést cestu k datové sadě a formát jejich fotografií.

Po provedené přípravě je možné v prostředí Matlab spustit proces rozpoznávání příkazem *RunFaceRecog*. Výstup byl doplněn o počítadlo správně rozpoznaných tváří pro rychlý přehled úspěšnosti.

4.4 KDDA

Tento algoritmus je publikován v [12] a [13]. Autoři popisují, jak se algoritmy založené na LDA potýkají s problémem, pokud počet vzorků pro trénování je menší než rozměrnost vzorků, a jak může být těžké od sebe lineárně oddělit dvě a více tříd. Proto se zde využívá *kernel* funkce pro nelineární mapování vstupních vzorků do více-dimenzionálního prostoru, kde jsou lépe oddělitelné.

Nejprve označme počet trénovacích dat jako L a jejich množinu $\{z_i\}_{i=1}^L$. Ty jsou reprezentovány jako vektory. Dále počet tříd (osob) C - $\{Z_i\}_{i=1}^C$ a hledáme transformaci φ , která mapuje každý trénovací obraz do $y_i = \varphi(z_i)$, která umožní data lépe oddělit.

Příznakový prostor F je vypočítán pomocí kernel funkce $k(z_i, z_j)$, která nahrazuje skalární součin příznakových vektorů $\phi(z_i) \in F$ a $\phi(z_j) \in F$, které odpovídají vstupním obrazům z_i a z_j . Pro kernel lze použít např. polynomiální funkci či radiální bázovou funkci (RBF). V tomto případě byla použita druhá zmíněná:

$$k(z_i, z_j) = \exp\left(\frac{-\|z_i - z_j\|^2}{\sigma^2}\right) \quad (4.11)$$

Celý algoritmus můžeme rozdělit do tří částí, jimiž jsou vlastní analýza rozptylové matice mezi třídami S_b v příznakovém prostoru, vlastní analýza rozptylové matice S_w uvnitř tříd a redukce dimenze a extrakce příznaků.

V první části se nejdříve vypočítá kernel matice K o velikosti $L \times L$. Poté se snažíme nalézt vlastní čísla a jim odpovídající vektory matice $S_b = \Phi_b \Phi_b^T$, kde $\Phi_b = [\phi_1 \dots \phi_c]$ a $\phi_i = \sqrt{\frac{C_i}{L}}(\phi_i - \phi)$. $C - 1$ nejvýznamnějších vlastních vektorů lze také nepřímo odvodit z matice $\Phi_b^T \Phi_b$ velikosti $C \times C$.

$$\Phi_b^T \Phi_b = \frac{1}{L} B \cdot (A_{LC}^T \cdot K \cdot A_{LC} - \frac{1}{L} (A_{LC}^T \cdot K \cdot 1_{LC}) - \frac{1}{L} (1_{LC}^T \cdot K \cdot A_{LC}) + \frac{1}{L^2} (1_{LC}^T \cdot K \cdot 1_{LC})) \cdot B \quad (4.12)$$

Matice $B = \text{diag}[\sqrt{C_1} \dots \sqrt{C_c}]$ je diagonální a na hlavní diagonále obsahuje druhé odmocniny počtů tváří každé osoby. 1_{LC} je matice jedniček o velikosti $L \times C$, stejně jako $A_{LC} = \text{diag}[a_{c1} \dots a_{cc}]$, jež představuje blokově diagonální matici se sloupcovými bloky (vektory) velikosti $C_i \times 1$ nabývajících hodnot $\frac{1}{C_i}$. $C - 1$ vypočítaných vlastních vektorů matice $\Phi_b^T \Phi_b$, které jsou seřazeny sestupně podle velikosti odpovídajících vlastních hodnot označme jako E a diagonální matici vlastních hodnot jako Λ_b .

Vlastní analýza matice rozptylu uvnitř tříd S_w v příznakovém prostoru zahrnuje výpočet

$$U^T S_w U = (E \Lambda_b)^T (\Phi_b^T S_w \Phi_b) (E \Lambda_b) \quad (4.13)$$

$$\Phi_b^T S_w \Phi_b = \frac{1}{L^2} B \cdot (A_{LC}^T \cdot K2 \cdot A_{LC} - \frac{1}{L} (A_{LC}^T \cdot K2 \cdot 1_{LC}) - \frac{1}{L} (1_{LC}^T \cdot K2 \cdot A_{LC}) + \frac{1}{L^2} (1_{LC}^T \cdot K2 \cdot 1_{LC})) \cdot B \quad (4.14)$$

B , A_{LC} a 1_{LC} známe již z předchozího kroku. Matici K^2 získáme ze vztahu $K \cdot (I - W) \cdot K$, kde I je jednotková matice velikosti $(C - 1) \times (C - 1)$ a $W = \text{diag}[w_1 \dots w_c]$ je blokově diagonální matice ($L \times L$) s bloky w_i velikosti $C_i \times C_i$ obsahující hodnoty $\frac{1}{C_i}$. Nyní opět vypočítáme matici vlastních vektorů P a matici vlastních čísel Λ_w , ale tentokrát matice $U^T S_w U$. Tyto vektory setřídíme podle hodnot čísel vzestupně.

V poslední části získáváme projekci vstupního vzorku z do podprostoru příznakových vektorů jako $y = \Theta \cdot \gamma(\phi(z))$, kde

$$\Theta = \frac{1}{\sqrt{L}} (E \cdot \Lambda_b \cdot P \cdot \Lambda_w)^T (B \cdot (A_{LC}^T - \frac{1}{L} 1_{LC}^T)) \quad (4.15)$$

je matice o velikosti $(C - 1) \times L$ reprezentující podprostor a $\gamma(\phi(z)) = [\phi_{11}^T \phi(z), \phi_{12}^T \phi(z), \dots, \phi_{CC}^T \phi(z)]^T$ je kernel vektor $L \times 1$.

4.4.1 Implementace

Implementace vychází z principu uvedeného výše a lze ji také nalézt v matlabovkém kódu na [12], odkud byl čerpán způsob práce s kernel maticí a parametry *threshold* (práh) a *update* pro úpravu hodnot vlastních čísel. Celý program obsahuje čtyři moduly psané v C++ s využitím toolkitu QT a jedná se o grafickou aplikaci. První spouštěný je *main.cpp*, který vytváří instanci třídy *FaceRec* (z *FaceRec.cpp*) a zobrazí grafické rozhraní.

Tento objekt tvoří jádro algoritmu, stará se o nastavování parametrů (cesty k datovým sadám, viditelnosti tlačítek apod.) a zobrazuje výsledky v textové i grafické podobě. Stěžejními funkcemi jsou *training()*, která zastřešuje fázi tréningu a *test()* pro fázi testu. Na jejich začátku probíhá načítání dat díky *nactiTestovacíData()* či *nactiTrenovacíData()* přijímajících jako parametr cestu k adresáři. Fyzické načítání však provádí třída *nacitanidat*, kterou obě využívají s jinými parametry.

Tato třída obsahuje funkci *nactiData()*, která do proměnné typu *IplImage*** načítá jednotlivé obrazy pomocí vestavěné funkce OpenCV (*cvLoadImage*). Do *tvareNaOsobu*, což jest matice typu *CvMat* o jednom řádku a se sloupci o počtu osob, ukládá počet fotografií, kterým každá osoba disponuje. Dvou-průchodové procházení souborů na disku umožní nejprve zjistit počet obrazů, na které se naalokuje místo v paměti pro proměnnou kde budou ukládány, a teprve poté jsou načteny. Procházení adresářů se provádí se zanořením do hloubky 1 a hledají se soubory ve formátu PGM, PPM a JPG.

Po této akci funkce *training()* pokračuje výpočtem matice *kernel*, obsahující hodnoty radiální bázové funkce. Parametr této funkce (σ^2) nabývá hodnoty 1.3389e7, která je považována za optimální. Zjištěná matice je poté využita k analýze rozptylu mezi třídami. Matice *Sb* je výsledkem rovnice 4.12, z níž jsou pomocí *cvEigenVV()* počítány vlastní vektory čísla. Ty jsou uloženy v maticích *e_vectors* a *e_values*, seřazené sestupně podle velikosti vlastních hodnot již jako produkt této funkce. Jejich počet je stále C a tudíž je provedena redukce matice o vektor s nejmenším vlastním číslem včetně jeho samého. Protože jsou čísla realizována jako sloupcový vektor, je z něj vytvořena diagonální matice.

Trénink pokračuje analýzou rozptylu uvnitř tříd. Výsledkem je matice S_w získaná z rovnice 4.14. $U^T S_w U$ v implementaci zastupuje matice $U S_w U$ typu *CvMat* a je spočítána z S_w a vlastních čísel a vektorů předešlé analýzy. Tato matice slouží opět jako vstupní parametr *cvEigenVV()*.

Matice *KDDASubspace* představuje vypočítaný podprostor z trénovacích dat. Pro následné využití ve fázi testování je uložena do lokálního adresáře ve formátu XML (*KDDASubspace.xml*). Díky tomuto je možné spouštět opakované testování nad různými

daty (v rámci jedné datové sady, jinak efekt ztrácí význam) bez nutnosti trénování se stále dokola na stejné obrazy. Mimo tuto matici je také ukládán obsah proměnné *tvareNaOsobu* do *TrainVector.xml*. Do souboru *path.txt* je v textové podobě ukládána cesta k trénovacím datům. Toho lze využít při opakovaném spuštění aplikace. Ta totiž pro fázi testování vyžaduje znát tuto informaci.

Před ukončením je volána funkce *test()*, která provádí projekci dat do podprostoru. Ta je také volána samostatně na podmět uživatele, který určuje cestu k testovacím datům. V případě volání této funkce v rámci *trening()* však promítá do podprostoru trénovací data. Opět načte *KDDAsubspace* a vypočítá kernel matici (*kernel*) z testovacích a trénovacích dat. Projekce je pak dána součinem těchto matic, která se nachází v matici *vysledek*. Pokud byl test volán z funkce *trening()*, tak je uložena do souboru *TrainData.xml*. Nyní již tedy známe umístění trénovacích obrazů ve více-dimenzionálním prostoru.

Pokud je *test()* volán na žádost uživatele, tak již je využita jím zvolená cesta k testovacím datům a tyto se promítají do podprostoru. Také se do hry dostává modul *hodnoceni.cpp* obsahující třídu *Hodnoceni*. Její konstruktor přijímá tři parametry, jimiž jsou matice *vysledek*, počet tváří na osobu (pro testovací data) a celkový počet fotografií. Funkce *vyhodnotData()* vrací celočíselnou hodnotu charakterizující počet správně rozpoznávaných tváří. Načte matici projekcí trénovacích dat, v níž se hledá tvář, která má nejmenší vzdálenost vůči každé projekci testovací tváře. Tváře v maticích projekcí jsou indexovány a na základě znalosti počtu fotografií na osobu a celkového počtu fotografií lze stanovit správnost klasifikace. Všechny dobře, ale i chybně rozpoznané tváře jsou zvlášť ukládány do textových řetězců. Části těchto řetězců obsahují adresáře a názvy testovací tváře a k ní nejbližší trénovací tváře. Z objektu realizujícího vyhodnocování dat lze tyto řetězce získat funkcí *vratVysledek()*. Toho je využito pro zobrazování grafické informace výsledku rozpoznávání. O výpis seznamu jednotlivých klasifikací se v rámci třídy *FaceRec* stará funkce *zobrazTabulku()* a o zobrazení náhledu rozpoznávané a rozpoznané tváře funkce *zobrazFotky()*. Uživatel má možnost přepínat se mezi seznamem dobře a chybně rozpoznávaných.

Kapitola 5

Datové sady

Pro testování výše zmíněných algoritmů bylo potřeba opatřit si dostatečné množství fotografií tváří, neboli datové sady. K těmto účelům lze nalézt na internetových stránkách mnoho různých sad. Některé jsou zaměřeny na zachycení lidské tváře z různých úhlů pohledu fotografa nebo také na různé vzdálenosti osoby na fotografii. Jiné obsahují fotografie tváří z čelního pohledu za proměnného osvětlení. Existují i sady tváří lidí, které jsou určitým způsobem pozměněny. Za změnu můžeme považovat proměnlivý výraz obličeje (radost, zamračenost, unavenost apod.), ale také přidání slunečních či dioptrických brýlí, šátku či klobouku.

Různé sady nabízí různý počet osob, u kterých bylo focení provedeno, a různý počet jejich nasnímaných tváří.

Pro ukázkou můžeme uvést několik těchto sad:

- Feret
- Umist
- ORL
- Japanese Female Facial Expression (JAFFE)
- Purdue U
- Yale Face
- atd.

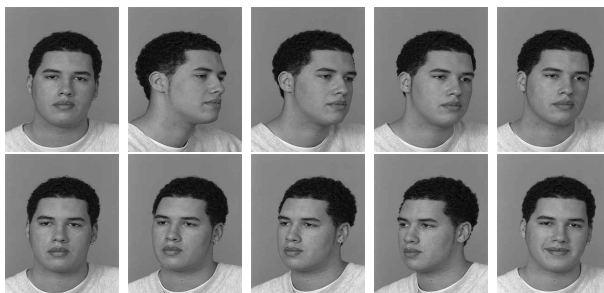
První tři uvedené byly využity pro testování vybraných algoritmů, proto si je přiblížíme.

5.1 Feret

Feret (Facial Recognition Technology) - program, jehož cílem je vyvinout nové techniky a algoritmy pro automatické rozpoznání lidské tváře. Fotografie byly pořízeny v letech 1993 - 1996 [14], jejichž počet je 14126, které byly získány od 1199 osob. Na obrázku 5.1 je ukázáno 10 fotografií jednoho člověka této databáze.

Obrazy jsou ve formátu JPG o velikosti 256 x 384 pixelů s barevnou hloubkou 8 bitů. Z celé sady byly vybrány pouze ty osoby, které splňovaly kritérium, kterým bylo minimální počet fotografií pro daného jedince. Minimum bylo zvoleno 8, neboť je potřeba právě 7

fotografií pro trénování algoritmů a alespoň jedna pro testování. Z této podmnožiny datové sady byla vybrána část, která obsahuje 124 osob s celkovým počtem fotografií 2044.



Obrázek 5.1: Ukázka datové sady Feret.

Tyto fotografie jsou bez normalizace. Na [15] lze však nalézt program na testování algoritmů pro rozpoznání lidské tváře - *CSU Face Identification Evaluation System* (CSU - Colorado State University), který obsahuje mimo jiné zdrojový kód *csuPreprocessNormalize.c*, jež lze použít pro normalizaci této datové sady. Normalizace zajišťuje konverzi obrazu do 256 stupňů šedi. Geometrickou normalizaci podle souřadnic očí a následné oříznutí podle eliptické masky. Obraz tak získává nový rozměr 130×150 a formát PGM. Výsledkem normalizace není stejně velká datová sada, ale redukovaná, neboť jsou vyřazeny všechny laterální (boční) pohledy tváří na fotografiích.

Pro testování algoritmů byly zvoleny i takto připravené obrazy. Ovšem i ony musely splňovat zmíněné kritérium, čímž se do užšího výběru dostalo pouze 283 fotografií od 30 osob. Výsledek normalizace je na obrázku 5.2.

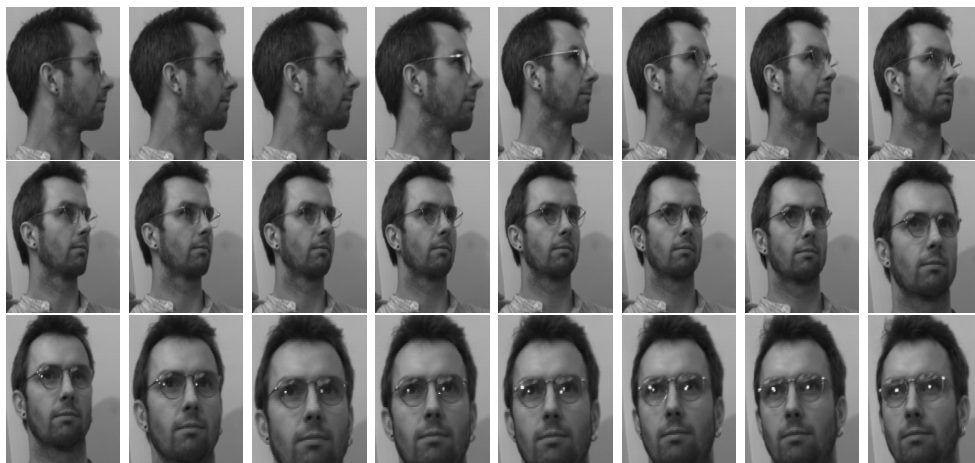
Celou datovou sadu je možné stáhnout po zaslání e-mailu podle instrukcí na [16].



Obrázek 5.2: Ukázka normalizované datové sady Feret.

5.2 Umist

Databáze Umist, která je k nalezení na [17], obsahuje 1012 fotografií od 20 lidí. Ty jsou ve formátu PGM a nemají stejnou obrazovou velikost. Šířka se pohybuje v rozmezí 161 – 252 pixelů a výška 185 – 270 pixelů s barevnou hloubkou 8 bitů. Tváře jsou nafoceny z různých úhlů. Náhled lze vidět na obrázku 5.3.



Obrázek 5.3: Ukázka datové sady Umist.

5.3 ORL

Tato datová sada se skládá ze 40 osob, kde každá obsahuje právě 10 fotografií. Celkem tedy ORL skýtá 400 obrázky. Ty jsou rovněž jako databáze Umist ve formátu PGM s pevnou velikostí 92×112 pixelů a barevnou hloubkou 8 bitů. Na obrázku 5.4 je opět vidět 10 fotografií jedné osoby této sady. Stažení je volně dostupné na [18].



Obrázek 5.4: Ukázka datové sady ORL.

Kapitola 6

Výsledky

V části 4 byly popsány algoritmy, které byly použity pro účely testování. Tato kapitola zahrnuje popis programu *DataMaker* pro přípravu datových sad, vysvětluje způsob testování a ukazuje dosažené výsledky.

6.1 DataMaker

Pro efektivní přípravu trénovacích a testovacích množin fotografií z původních datových sad byl napsán program *DataMaker*. Hlavním důvodem je skutečnost, že celkový počet tváří je značně vysoký a tak je nereálné připravovat tato data ručně. Krom tohoto faktoru hraje nezanedbatelnou roli také specifický formát uložených dat, který není pro uvedené algoritmy totožný a tak je třeba jednu fotografii upravit vícekrát.

Jedná se o GUI (Graphical User Interface) aplikaci s intuitivním ovládáním. Celé jádro implementace (*datamaker.cpp*) je napsáno v C++ v toolkitu QT s využitím knihovny OpenCV. Mimo základních funkcí pro ovládání aplikace se tu nachází *vytvorData()*, jež je volána po stisku tlačítka *Vytvoř*. Ta vyžaduje, aby zvolený zdrojový adresář obsahoval podadresáře fotografií lidských tváří. Pro každou osobu jeden. Poznámka: tohle hierarchické uspořádání je již při obstarání si konkrétní datové sady většinou dáno. Pokud je skutečnost jiná, je nutné před použitím tohoto programu onu sadu patřičně upravit.

V adresáři, uvedeném jako cílový, vytvoří adresáře s názvem 2, 3, až maximálně 7 podle nastavených parametrů *Od* a *Do*. Číslo značí počet tváří na osobu, které budou zahrnuty do trénovací množiny. V těchto adresářích poté vzniknou další - *fisher*, *test* a *training* a také dva textové soubory *test.txt* a *train.txt*. Program náhodně vybírá zvolený počet fotografií každé osoby ze zdrojového adresáře a kopíruje je do adresáře *training*. Cestu cíle také zapíše spolu s číslem osoby do souboru *train.txt*. Zkopírovanou fotografii díky funkcím OpenCV načte ve stupních šedi s osmibitovou hloubkou, změní velikost obrazu na 92×112 a uloží. Fotografie se také kopíruje do adresáře *fisher/training*, kde je podrobena stejným úpravám s výslednou velikostí 40×60 .

Zbylá množina tváří po výběru trénovacích dat je analogicky zkopírovaná do adresářů *test* a cesty k nim uloženy do souboru *test.txt*.

Pro algoritmus fisherface jsou u všech fotografií v adresáři *fisher* pozměněny jejich názvy, na základě kterých probíhá vyhodnocení správnosti klasifikace.

Výsledná adresářová struktura, která je znázorněna níže je využita následovně: Pro algoritmus fisherface jsou vstupem *../cislo/fisher/training* a *../cislo/fisher/test*, pro KDDA *../cislo/training* a *../cislo/test* a algoritmus eigenface vyžaduje pro svou činnost soubory

test.txt a *train.txt*.

Poznámka: Data vytvořená tímto programem mohou být libovolně kopírována do kteréhokoliv adresáře jen pro algoritmy KDDA a fisherface. Pro eigenface to však není možné, neboť zmíněné textové soubory obsahují absolutní cesty k obrazům.

cílový adresář

```
| - 2
|   | - fisher
|   |   | - test      (obsahuje např. soubory: 1002-1.pgm, 1002-10.pgm, ...)
|   |   | - trening
|   | - test
|   |   | - s1        (názvy adresářů původní datové sady,
|   |   | - s10       obsahují např. soubory: 1.pgm, 10.pgm, ...)
|   |   | - ...
|   | - trening
|   |   | - s1
|   |   | - s10
|   |   | - ...
|   | - test.txt
|   | - train.txt
| - 3
| - ...
```

6.2 Hodnocení

Algoritmy byly testovány na čtyřech datových sadách. Jak již bylo řečeno, každá byla rozdělena na trénovací a testovací množinu, kde ta první obsahovala pro každou osobu vždy stejný počet fotografií. Účelem bylo sledovat, jak se mění klasifikační schopnost algoritmů v závislosti na počtu předložených obrazů, na kterých měly možnost se učit. Tento počet, v tabulkách uveden jako K , nabývá hodnot 2 - 7, čili bylo provedeno 5 testů na každé datové sadě. Sloupec označený jako L charakterizuje počet testovacích dat a je roven $L = N - K * C$, pokud N je celkový počet dat a C počet osob (tříd) sady. V konferenčních člancích zabývajících se rozpoznáváním lidské tváře je vidět, že v testování figuruje parametr chybovost klasifikace (CER - classification error rate) a také míra správného rozpoznání (CRR - correct recognition rate). Tabulky hodnot, obsažené v této práci, zobrazují úspěšnost algoritmů uváděnou v procentech, která odpovídá právě správnému rozpoznání.

Data pro trénování a testování byla pro každou datovou sadu (s uvedeným rozložením počtu obrazů na osobu) náhodně generována celkem čtyřikrát. Naměřené hodnoty v následujících tabulkách jsou tedy aritmetickým průměrem úspěšností nad takto vytvořenými množinami. Všechny získané hodnoty lze najít v příloze této práce, kde 4 tabulky pro každou datovou sadu jsou označeny postupně jako Běh 1 - Běh 4.

Výsledky jsou také zobrazeny v podobě grafů, které byly vytvořeny pomocí nástroje Gnuplot. Jednak jsou vyneseny závislosti úspěšností algoritmů na počtu trénovacích dat na osobu pro každou datovou sadu, ale také porovnání úspěšností každého z nich skrz paletu všech datových sad.

Další pohled na posouzení algoritmů se ubírá směrem k časové náročnosti výpočtů. V tomto případě se již nejedná o aritmetický průměr, neboť měření bylo provedeno jen jednou. Pro ukázkou byly vybrány jen dvě datové sady. Tou první je ORL, která není tak

rozsáhlá a tou druhou je Feret. U KDDA a eigenface je měřena doba tréningu i testu zvlášť, u fisherface celková doba.

Testování proběhlo na architektuře AMD Sempron 3100+ (1,8 GHz) s operační pamětí 512 MB.

Když začneme touto časovou náročností (tabulky 6.2 a 6.2), tak algoritmus KDDA a eigenface tráví nad fází testování pro malou sadu téměř konstantní čas pro všechna K . KDDA však svou celkovou dobou výpočtu přesahuje výslednou dobu eigenface $2\times$ (pro $K = 2$) až $3\times$ (pro $K = 7$). Nejlepšího hodnocení dosáhl na této ORL databázi právě eigenface a nejhůře fisherface. Podíváme-li se ale na hodnocení poměrně velké sady, jakou je Feret, tak opět vedoucím algoritmem je eigenface. Kdyby počet trénovacích dat na osobu nekončil na hodnotě 7, lze z trendu nárůstu času pro fisherface odhadnout, že by se tento přesunul na přední místo v žebříčku hodnocení. KDDA však již vyžaduje mnohem delší čas pro celkový výpočet. Pokud jej opět porovnáme s eigenface, tak náročnost oproti ORL se se zvyšujícím se K klesá. KDDA vyžaduje $4,98\times$ více času pro $K = 2$ a $3,62\times$ pro $K = 7$. Výsledné grafy jsou na obrázku 6.1.

Nyní se tedy zaměříme na hlavní cíl testování algoritmů pro rozpoznání lidské tváře a to na úspěšnost klasifikace. Ze zjištěných hodnot je vidět, že má na ni parametr K významný vliv. Pokud bylo algoritmům předloženo více dat, na kterých se mohly natrénovat, jejich korektní klasifikace se zvýšila.

Tabulka 6.3 ukazuje úspěšnosti vybraných algoritmů na datové sadě ORL. S malým počtem trénovacích dat si nejlépe poradil KDDA (80%). Při $K = 7$ jsou již výsledky téměř vyrovnané. Z globálního pohledu (grafy na obrázku 6.3) je tato sada nejúspěšnější v porovnání s ostatními. Jak bylo vidět na obrázku 5.4, v kapitole datové sady, obsahuje tváře s relativně malou změnou úhlu natočení. Takovou vlastnost má i normalizovaná datová sada Feret, ale KDDA již nedosahuje úspěšnosti jako na ORL. Naproti tomu fisherface zde podal vůbec nejlepší výsledek ze všech algoritmů napříč všemi sadami pro $K = 6$. Průměrná správná klasifikace tu získala 98,06% (101 tváří ze 103). V příloze, kde jsou uvedeny všechny naměřené hodnoty, lze najít, že ve dvou testech ze čtyř (pro zvolené K) chybně klasifikoval pouze jedinou tvář.

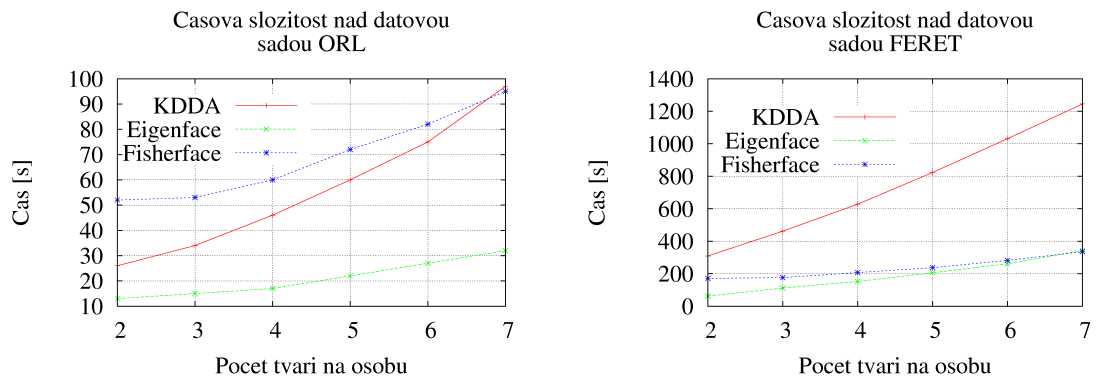
Nejhorší výsledky se dostavily pro datovou sadu Feret bez normalizace (tabulka 6.5), která skýtá 2044 fotografií patřících 124 osobám. Eigenface byl z testovaných algoritmů, v rámci této sady, nejúspěšnější se svými 55,76 procenty, kdy korektně rozeznal 655 tváří z 1176. KDDA 40,05% a fisherface pouhých 29,49%.

Datová sada ORL (40 osob, 400 fotografií)								
K	L	KDDA			Eigenface			Fisherface
		Trening	Test	Celkem	Trening	Test	Celkem	Celkem
2	320	8	18	26	5	8	13	52
3	280	15	19	34	7	8	15	53
4	240	26	20	46	9	8	17	60
5	200	40	20	60	14	8	22	72
6	160	56	19	75	18	9	27	82
7	120	79	18	97	23	9	32	95

Tabulka 6.1: Časy potřebné pro výpočet nad datovou sadou ORL.

Datová sada FERET (124 osob, 2044 fotografií)								
K	L	KDDA			Eigenface			Fisherface
		Trening	Test	Celkem	Trening	Test	Celkem	Celkem
2	1796	62	247	309	18	44	62	171
3	1672	139	323	462	38	74	112	176
4	1548	238	390	628	66	86	152	207
5	1424	375	448	823	104	102	206	237
6	1300	541	490	1031	156	105	261	282
7	1176	726	519	1245	218	125	343	335

Tabulka 6.2: Časy potřebné pro výpočet nad datovou sadou Feret.



Obrázek 6.1: Závislost času potřebného pro výpočet na počtu fotografií pro osobu nad datovými sadami. Vlevo ORL, vpravo Feret.

Datová sada ORL (40 osob, 400 fotografií)							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	320	257	80,39	245	76,48	236	73,83
3	280	247	88,04	244	87,14	244	86,96
4	240	214	89,17	219	91,15	220	91,56
5	200	186	93,00	188	94,00	190	95,00
6	160	153	95,31	153	95,63	155	96,88
7	120	114	94,79	116	96,46	115	95,83

Tabulka 6.3: Úspěšnost algoritmů na datové sadě ORL.

Datová sada UMIST (20 osob, 1012 fotografií)							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	972	712	73,28	707	72,76	655	67,36
3	952	733	77,02	778	81,72	781	82,04
4	932	796	85,43	823	88,25	843	90,42
5	912	799	87,64	830	91,04	850	93,23
6	892	825	92,46	852	95,49	854	95,71
7	872	811	93,03	825	94,58	832	95,44

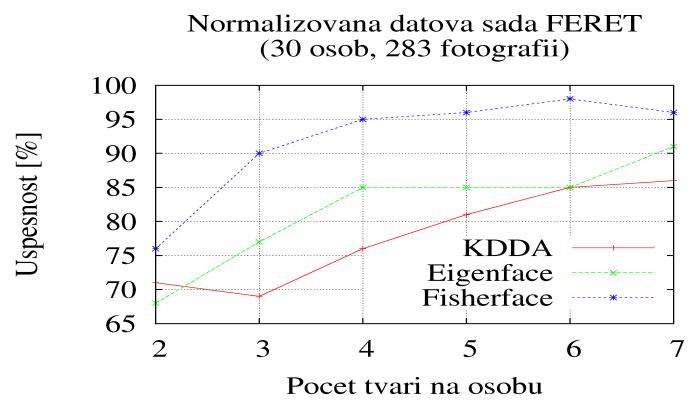
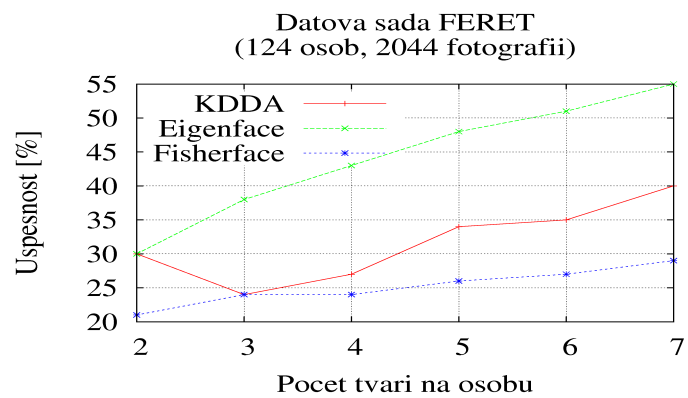
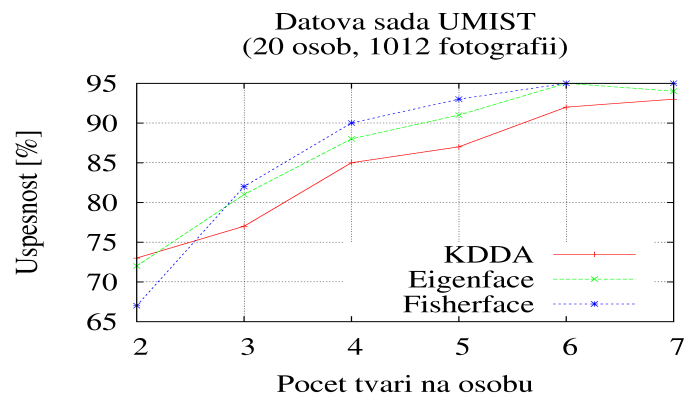
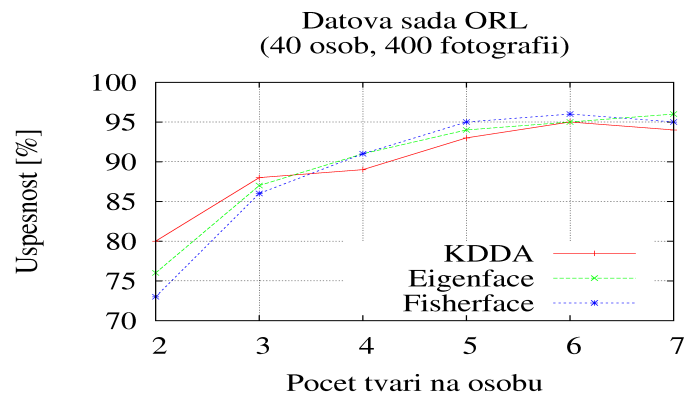
Tabulka 6.4: Úspěšnost algoritmů na datové sadě UMIST.

Datová sada Feret (124 osob, 2044 fotografií)							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	1796	547	30,44	556	30,93	380	21,13
3	1672	413	24,69	649	38,83	402	24,03
4	1548	420	27,12	666	43,02	375	24,22
5	1424	497	34,90	694	48,70	376	26,39
6	1300	465	35,77	669	51,42	356	27,37
7	1176	471	40,05	656	55,76	347	29,49

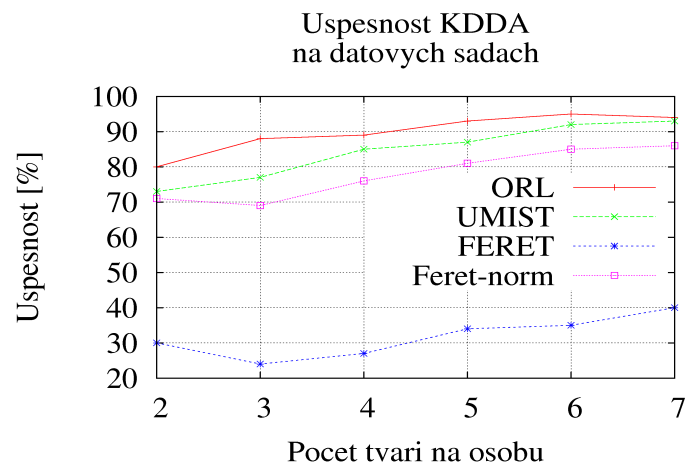
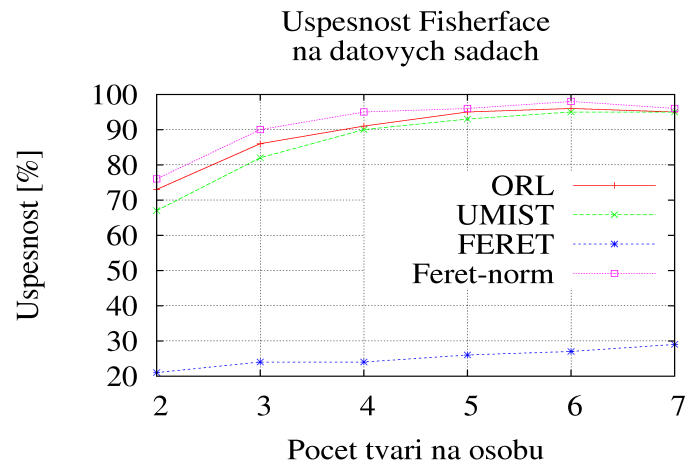
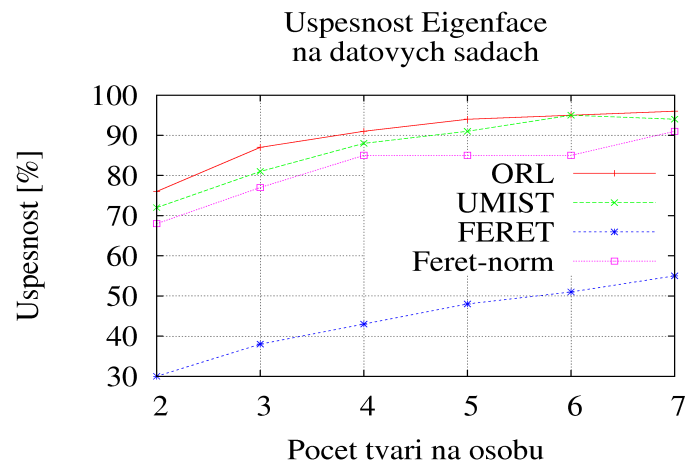
Tabulka 6.5: Úspěšnost algoritmů na datové sadě FERET.

Datová sada Feret - normalizovaná (30 osob, 283 fotografií)							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	223	159	71,19	152	68,05	171	76,57
3	193	134	69,30	149	77,33	175	90,54
4	163	124	76,23	139	85,12	156	95,55
5	133	108	81,20	114	85,53	128	96,05
6	103	88	85,19	89	85,92	101	98,06
7	73	63	86,64	67	91,10	71	96,58

Tabulka 6.6: Úspěšnost algoritmů na normalizované datové sadě FERET.



Obrázek 6.2: Závislosti úspěšností na počtu fotografií na osobu použitých pro trénink.



Obrázek 6.3: Porovnání úspěšností algoritmů na jednotlivých sadách.

Kapitola 7

Závěr

V rámci této práce jsem se seznámil s problematikou detekce lidské tváře v obraze a vyzkoušel jsem si ji pomocí programu *facetedetect* na množině fotografií, z nichž zde byla jedna uvedena.

Seznámil jsem se s metodami a konkrétními algoritmy pro rozpoznání lidské tváře. Opatřil jsem si již existující algoritmy, které jsem upravoval pro korektní práci nad připravenými daty a implementoval algoritmus KDDA, zasazený do grafického prostředí. Celkem 4 moduly i s hlavičkovými soubory obsahují 1300 řádků opět včetně prázdných a komentářů.

Obstaral jsem si datové sady tváří a provedl výběr vhodných osob z databáze Feret tak, aby byly splněny podmínky pro zvolené testování. Tuto databázi jsem také podrobil procesu normalizace programem z dostupného systému CSU Face Identification Evaluation System.

Vytvořil jsem program *DataMaker*, který automaticky připravuje datové sady pro trénování a testování algoritmů, jež jsou uvedeny v této práci. Modul s hlavičkovým souborem obsahuje přibližně 230 řádků včetně prázdných a komentářů.

Provedl jsem sadu testování, jak z hlediska úspěšnosti tak časové náročnosti.

Zřejmě nejtěžší částí bylo zorientovat se v matematických popisech jednotlivých algoritmů. O pochopení určitých zásad a principů se zasloužil předmět vyučovaný na zdejší škole - Klasifikace a rozpoznávání.

Co se týče dalšího vývoje této práce, bylo by možné program pro rozpoznávání obohatit o detekci tváře s jejím následným oříznutím a normalizací. Další možností je také zaměřit se na algoritmy pracující v 3D prostoru a porovnat je se zde uvedenými.

Počítačová oblast, zabývající se rozpoznáváním a biometrií obecně, se mi jeví jako velice zajímavá. Navíc v dnešní době nachází značné uplatnění.

Literatura

- [1] RAK Roman, MATYÁŠ Václav, ŘÍHA Zdeněk, and kolektiv. *Biometrie a identita člověka ve forenzních a komerčních aplikacích*. Grada Publishing a.s., 2008. ISBN 978-80-247-2365-5.
- [2] DELAG K., GRGIC M., and BARLETT M.S. *Recent advances in face recognition*. IN-TECH, December 2008. ISBN 978-953-7619-34-3.
- [3] ŠPANĚL Michal. *Rozpoznávání gest ve video sekvencích* [online], 2003 [cit. 2010-04-25]. 83 s. Diplomová práce na Fakultě informačních technologií, VUT Brno. Vedoucí diplomové práce Doc. Dr. Ing. Pavel Zemčík.
<http://www.fit.vutbr.cz/~spanel/dp/spanel_dp_2003.pdf>.
- [4] ŠČUREK Radomír. *Biometrické metody identifikace osob v bezpečnostní praxi* [online], 2008 [cit. 2010-05-12].
<http://www.fbi.vsb.cz/shared/uploadedfiles/fbi/biometricke_metody.pdf>.
- [5] PISAREVSKY Vadim. *OpenCV* [online], poslední revize 2010-04-06 [cit. 2010-04-25].
<<http://opencv.willowgarage.com/wiki/>>.
- [6] TURK Matthew and PENTLAND Alex. *Eigenfaces for Recognition* [online], 1991 [cit. 2010-04-25]. <<http://www.face-rec.org/algorithms/PCA/jcn.pdf>>.
- [7] HEWITT Robin. *Seeing With OpenCV - A Five-Part Series* [online], 2007 [cit. 2010-04-25].
<http://www.cognotics.com/opencv/servo_2007_series/part_5/part5-src.zip>.
- [8] *Experimental and Obsolete Functionality Reference* [online], poslední revize 2006-10-31 [cit. 2010-04-25].
<http://www.cognotics.com/opencv/docs/1.0/ref/opencvref_cvaux.htm>.
- [9] *Regulární matice* [online], [cit. 2010-04-25].
<<http://www.math.muni.cz/kadourek/regulmat.pdf>>.
- [10] BELHUMEUR P.N., HESPANHA J.P., and KRIEGMAN D.J. *Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection* [online], 1996 [cit. 2010-04-30].
<<http://www.face-rec.org/algorithms/LDA/belhumeur96eigenfaces.pdf>>.
- [11] BROOKS Alan and GAO Li. *RunFaceRecog_vV.m* [online], 2004-06-01 [cit. 2010-04-25].
<http://dailyburrito.com/projects/facerecog/code/RunFaceRecog_vV.m>.

- [12] LU Juwei, PLATANOTIS K.N., and VENETSANOPOULOS A.N. *Face Recognition Using Kernel Direct Discriminant Analysis Algorithms* [online], 2002-08-12 [cit. 2010-04-25]. <<http://www.face-rec.org/algorithms/Kernel/r-kdda.zip>>.
- [13] LU Juwei, PLATANOTIS K.N., and VENETSANOPOULOS A.N. *Kernel Discriminant Learning with Application to Face Recognition* [online], 2004-10-04 [cit. 2010-04-25]. <http://www.dsp.utoronto.ca/juwei/Publication/JLu_KP_ANV.pdf>.
- [14] *The Color FERET Database* [online], verze 2, poslední revize 2008-01-31 [cit. 2010-05-09]. <<http://face.nist.gov/colorferet/>>.
- [15] *Evaluation of Face Recognition Algorithms* [online], poslední revize 2007-05-01 [cit. 2010-04-25]. <<http://www.cs.colostate.edu/evalfacerec/algorithms5.html>>.
- [16] *Color FERET Download Requests* [online], verze 2, poslední revize 2007-09-18 [cit. 2010-04-25]. <<http://face.nist.gov/colorferet/request.html>>.
- [17] The University Of Sheffield. *Face Database* [databáze online], c2010 [cit. 2010-04-25]. <<http://www.shef.ac.uk/eee/research/vie/research/face.html>>.
- [18] AT&T Laboratories Cambridge. *The Database of Faces* [online], c2002 [cit. 2010-04-25]. <<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>>.

Příloha A

Obsah CD

- Písemná zpráva ve formátu PDF
- Zdrojový tvar písemné zprávy
- Algoritmy Eigenface, Fisherface a KDDA
- Program pro tvorbu datových sad - DataMaker
- Datová sada ORL
- Datová sada ORL, zpracovaná pomocí DataMaker
- Náhledové listy datových sad
- Script pro Gnuplot, vektorové grafy
- Plakát

Příloha B

Tabulky hodnot

- Datová sada ORL (40 osob, 400 fotografií)
- Datová sada UMIST (20 osob, 1012 fotografií)
- Datová sada FERET (124 osob, 2044 fotografií)
- Datová sada FERET - normalizovaná (30 osob, 283 fotografií)

Běh 1							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	320	248	77,50	245	76,56	229	71,56
3	280	245	87,50	243	86,79	247	88,21
4	240	222	92,50	223	92,92	224	93,33
5	200	186	93,00	187	93,50	190	95,00
6	160	154	96,25	154	96,25	155	96,88
7	120	115	95,83	116	96,67	113	94,17

Běh 2							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	320	263	82,19	253	79,06	239	74,69
3	280	244	87,14	244	87,14	242	86,43
4	240	209	87,08	212	88,33	214	89,17
5	200	191	95,50	192	96,00	191	95,50
6	160	152	95,00	150	93,75	154	96,25
7	120	110	91,67	115	95,83	117	97,50

Běh 3							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	320	246	76,88	232	72,50	238	74,38
3	280	244	87,14	244	87,14	246	87,86
4	240	208	86,67	216	90,00	218	90,83
5	200	189	94,50	193	96,50	193	96,50
6	160	153	95,63	155	96,88	155	96,88
7	120	117	97,50	117	97,50	116	96,67

Běh 4							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	320	272	85,00	249	77,81	239	74,69
3	280	253	90,36	245	87,50	239	85,36
4	240	217	90,42	224	93,33	223	92,92
5	200	178	89,00	180	90,00	186	93,00
6	160	151	94,38	153	95,63	156	97,50
7	120	113	94,17	115	95,83	114	95,00

Tabulka B.1: Datová sada ORL.

Běh 1							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	972	711	73,15	667	68,62	595	61,21
3	952	771	80,99	777	81,62	819	86,03
4	932	785	84,23	814	87,34	860	92,27
5	912	751	82,35	805	88,27	818	89,69
6	892	843	94,51	854	95,74	870	97,53
7	872	786	90,14	812	93,12	810	92,89

Běh 2							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	972	700	72,02	722	74,28	700	72,02
3	952	736	77,31	754	79,20	735	77,21
4	932	807	86,59	822	88,20	836	89,70
5	912	827	90,68	842	92,32	870	95,39
6	892	810	90,81	853	95,63	858	96,19
7	872	818	93,81	831	95,30	844	96,79

Běh 3							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	972	699	71,91	691	71,09	635	65,33
3	952	735	77,21	806	84,66	806	84,66
4	932	789	84,66	813	87,23	815	87,45
5	912	845	92,65	847	92,87	868	95,18
6	892	822	92,15	848	95,07	839	94,06
7	872	834	95,64	840	96,33	843	96,67

Běh 4							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	972	739	76,03	749	77,06	689	70,88
3	952	691	72,58	775	81,41	764	80,25
4	932	804	86,27	841	90,24	860	92,27
5	912	774	84,87	827	90,68	845	92,65
6	892	824	92,38	852	95,52	848	95,07
7	872	807	92,55	816	93,58	832	95,41

Tabulka B.2: Datová sada Umist.

Běh 1							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	1796	576	32,07	575	32,02	364	20,27
3	1672	412	24,64	654	39,11	372	22,25
4	1548	472	30,49	657	42,44	410	26,49
5	1424	430	30,20	674	47,33	386	27,11
6	1300	434	33,38	649	49,92	331	25,46
7	1176	490	41,67	662	56,29	324	27,55

Běh 2							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	1796	571	31,79	587	32,68	405	22,55
3	1672	425	25,42	647	38,70	401	23,98
4	1548	373	24,10	671	43,35	342	22,09
5	1424	571	40,10	702	49,30	380	26,69
6	1300	468	36,00	649	49,92	356	27,38
7	1176	407	34,61	641	54,51	358	30,44

Běh 3							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	1796	499	27,78	520	28,95	368	20,49
3	1672	442	26,44	659	39,41	423	25,30
4	1548	392	25,32	658	42,51	378	24,42
5	1424	498	34,97	702	49,30	406	28,51
6	1300	479	36,85	681	52,38	396	30,46
7	1176	500	42,52	675	57,40	363	30,87

Běh 4							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	1796	541	30,12	540	30,07	381	21,21
3	1672	372	22,25	637	38,10	411	24,58
4	1548	442	28,55	678	43,80	370	23,90
5	1424	489	34,34	696	48,88	331	23,24
6	1300	479	36,85	695	53,46	340	26,15
7	1176	487	41,41	645	54,85	342	29,08

Tabulka B.3: Datová sada Feret.

Běh 1							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	223	164	73,54	157	70,40	167	74,89
3	193	134	69,43	145	75,13	179	92,75
4	163	139	85,28	139	85,28	156	95,71
5	133	113	84,96	114	85,71	129	96,99
6	103	90	87,38	94	91,26	100	97,09
7	73	64	87,67	68	93,15	71	97,26

Běh 2							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	223	157	70,40	147	65,92	175	78,48
3	193	137	70,98	152	78,76	168	87,05
4	163	112	68,71	136	83,44	156	95,71
5	133	98	73,68	113	84,96	124	93,23
6	103	89	86,41	88	85,44	100	97,09
7	73	67	91,78	70	95,89	70	95,89

Běh 3							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	223	162	72,65	153	68,61	168	75,34
3	193	126	65,28	149	77,20	175	90,67
4	163	130	79,75	142	87,12	156	95,71
5	133	108	81,20	113	84,96	129	96,99
6	103	79	76,70	81	78,64	102	99,03
7	73	59	80,82	64	87,67	69	94,52

Běh 4							
K	L	KDDA		Eigenface		Fisherface	
		Rozpoznáno	%	Rozpoznáno	%	Rozpoznáno	%
2	223	152	68,16	150	67,26	173	77,58
3	193	138	71,50	151	78,24	177	91,71
4	163	116	71,17	138	84,66	155	95,09
5	133	113	84,96	115	86,47	129	96,99
6	103	93	90,29	91	88,35	102	99,03
7	73	63	86,30	64	87,67	72	98,63

Tabulka B.4: Datová sada Feret-normalizovaná.