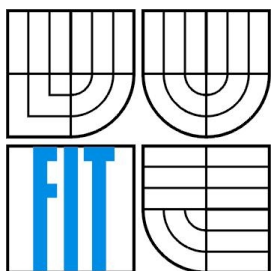


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IDENTIFIKACE OBLIČEJE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR ČÁSTEK

BRNO 2008

Abstrakt

Tento dokument se snaží čtenáře uvést do problematiky rozpoznávání obličeje spojeného s různorodostí snímací techniky a prostředí. Jsou zde rozebrány různé možnosti vytvoření unikátního otisku obličeje, aby byly co nejvíce potlačeny nežádoucí účinky prostředí a byla možná identifikace osob. Jdou zde rozebrány různé přístupy k problematice a použití různých modelovacích technik tváře.

Klíčová slova

identifikace obličeje, verifikace obličeje, popis rysů obličeje, strukturální metody, holistický přístup, normalizace, počítačové vidění, indexování, hashování, Analýza hlavních komponent (PCA), Diskriminační analýza (LDA), elastické shlukovací grafy, jet, strom, vzdálenost

Abstract

This document is trying to introduce the reader with issues of identifying the face connected with miscellaneous scanning technologies and environments. Inside this document there are mentioned some possibilities of creation unique print of a face so that there would be denied unwanted effects of environment and the identification of persons would be possible.

Keywords

identification of a face, verification of a face, definition of characteristics of a face, structural methods, holistic stance, normalization, computer sight, indexing, hashing, the analysis of main components, the discriminational analysis, the elastical clumping graphs, distance

Citace

Petr Částek: Identifikace Obličeje, diplomová práce, Brno, FIT VUT v Brně, 2008

Identifikace obličeje

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Petra Chmelaře.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
Datum

© Petr Částek, 2008.

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

Obsah.....	1
1 Úvod.....	5
2 Druhy rozpoznávání.....	8
2.1 Identifikace obličeje.....	8
2.2 Verifikace obličeje.....	9
2.3 Seznam sledovaných (watch list).....	11
3 Vzhled předloh.....	12
3.1 Normalizace.....	13
4 Reprezentační vektory.....	15
5 Lokalizace obličeje.....	16
5.1 Statistické (strukturální) metody.....	16
5.2 Strojové učení.....	16
5.2.1 Haarovy filtry.....	17
5.2.2 Strojové učení Haarových filtrů.....	17
5.2.3 Support Vector Machines (SVM).....	18
5.3 Počítačové vidění.....	19
5.3.1 Detekce Kůže.....	19
5.3.2 Gaborova vlnková transformace.....	21
5.3.3 Morfologické operace.....	24
5.3.4 Detekce úst.....	26
5.3.5 Detekce očí.....	29
5.3.6 Verifikace obličejové oblasti.....	30
6 Popis (rysů) obličeje.....	31
6.1 Statistický (strukturální) přístup.....	31
6.2 Lineární analýza.....	31
6.2.1 Analýza hlavních komponent (PCA).....	32
6.2.2 Srovnání Haarových filtrů a PCA.....	35

6.2.3	Analýza nezávislých komponent (ICA).....	35
6.2.4	Lineární diskriminační analýza (LDA).....	37
6.2.5	Použití.....	39
6.3	Nelineární analýza	39
6.3.1	Jádrová analýza hlavních komponent (KPCA).....	39
6.4	Modelový přístup.....	40
6.5	Metody založené na Elastických shlukových grafech.....	41
6.5.1	Jety.....	41
6.5.2	Porovnávání Jetů.....	43
6.5.3	Individuální obličej	43
6.5.4	Shlukový graf obličeje	44
6.5.5	Reprezentace tváře	44
6.5.6	Podobnostní funkce.....	44
6.6	2D modifikační model (AAM).....	45
6.7	3D modifikační model.....	45
7	Identifikace (vyhledávání) subjektů.....	46
7.1	Měření vzdálenosti vektorů.....	46
7.1.1	Manhattan vzdálenost.....	46
7.1.2	Euklidova vzdálenost.....	47
7.1.3	Minkowského vzdálenost.....	48
7.1.4	Mahalanobisova vzdálenost.....	48
7.1.5	Chebyshevova vzdálenost.....	49
7.2	Indexování.....	49
7.2.1	Bodové přístupové metody (PAM).....	50
7.2.2	Prostorové přístupové metody (SAM).....	50
7.2.3	B-strom.....	52
7.2.4	Mřížkový (grid) soubor.....	52
7.2.5	KD-strom.....	53
7.2.6	Quad-strom.....	53

7.2.7 R-strom.....	54
7.2.8 SS-strom.....	55
7.2.9 SR-strom.....	55
7.2.10 VP-strom.....	55
7.2.11 M-strom.....	55
7.3 Hashování.....	56
7.3.1 Metoda dělení.....	57
7.3.2 Metoda středních čtverců.....	57
7.3.3 Metoda násobení.....	58
7.3.4 Metoda rozkladu.....	58
7.3.5 Fibonacciho hashování.....	58
7.3.6 Cormackovo hashování.....	59
8 MPEG-7	61
8.1 Extrakce rysů a popis obsahu.....	63
8.1.1 Extrakce rysů.....	63
8.1.2 Extrakce tvaru.....	64
8.1.3 Extrakce pohybu a umístění.....	64
8.2 Komponentový LDA deskriptor tváře.....	65
8.2.1 Popis tváře.....	65
8.2.2 Podobnostní vyhledávání.....	67
8.2.3 Rekurzivní podobnostní vyhledávání.....	67
8.3 Použité Rozpoznání obličeje.....	69
9 Výsledky.....	70
9.1 Vstupní data.....	70
9.2 Implementace.....	72
9.2.1 Načtení snímků.....	73
9.2.2 Vektorová prezentace objektů (VOP).....	74
9.2.3 Lokalizace obličeje.....	74
9.2.4 Segmentace barev a extrakce rysů.....	74

9.2.5 Normalizace.....	75
9.2.6 Deskriptor	75
9.2.7 Vyhledávání	75
9.3 Výstupní data.....	77
9.3.1 Přímé výstupy.....	77
9.3.2 Nepřímé výstupy.....	78
9.4 Diskuse.....	78
9.4.1 Rozpoznání obličeje.....	78
9.4.2 Normalizované obličeje.....	81
9.4.3 Identifikace obličeje.....	82
10 Závěr.....	83
Literatura.....	84
Odkazy.....	90
Seznam příloh.....	91

1 Úvod

V posledních letech došlo k výraznému rozvoji technik rozpoznávání obličeje, což je způsobeno především masivním rozšířením a zlevněním výpočetní a snímací techniky pro širokou veřejnost. Díky tomu se stále více výzkumníků z oblastí biometrie a počítačového vidění obecně, zabývá problematikou identifikace obsahu získaného pomocí snímací techniky (typicky ve formě obrázků nebo videosekvencí). Strojové učení a počítačová grafika se stále více setkávají za jediným cílem, kterým je rozpoznávání obličeje.

Tento obecný zájem mezi výzkumníky, kteří pracují v různorodých oblastech informačních technologií, byl především motivován pozoruhodnou lidskou schopností rozpoznávat lidi. V neposlední řadě ovšem i skutečnost, že lidská aktivita je primárním zájmem v každodenním životě a ve světě počítačů. Již v minulosti jsme se setkali ve velké míře s požadavkem rozpoznávacích technologií například v kriminalistických aplikacích, bezpečnostních, ale i v komerčních.

Tyto aplikace představují automatizovaný přístup k identifikaci daného uživatele (pachatele) a v oblasti policejní identifikace zde máme ještě další rozšiřující požadavky, jako je rekonstrukce obličeje. V multimediálních komunikacích to může být vytvoření syntetických tváří z reálných a u multimediálních databází máme snahu mít možnost vyhledávat nejen podle popisu obrazových dat, ale i podle jejich fyzického obsahu.

Různé biometrické indikátory jsou vhodné pro různé druhy identifikačních metod, záleží pouze na konkrétních požadavcích na přesnost, cenu a v neposlední řadě i na uživatelské jednoduchosti snímání.

Dnes je známá celá řada lineárních a nelineárních strukturálních přístupů k lidské tváři. Později se oči odborníků zaměřili k 2D a 3D modelovacím technikám, které nastaly na základě možnosti modelovat lidskou tvář.

Chceme-li však rozpoznávat obličeje z obecně různých dat, nemůžeme pak předpokládat pouze vhodné a dobré normalizované snímky, ale obecně obrázky s obličejem v komplexním pozadí, na které budeme muset aplikovat sekvence různých filtrů pro detekci hran, barev a přechodů, po jejichž složení a dalších čištěních od chybných nebo nepoužitelných detekcí můžeme získat hranice

obličejové oblasti. Jelikož mnohdy nepůjde o čisté čelní pohledy obličejů, protože někteří lidé mohou chtít záměrně oklamat identifikaci, nebo ať půjde například o automatizované vyhledávání hledaných osob pomocí kamery na veřejném frekventovaném místě, musíme počítat s tím, že obličej bude nutné vhodně pootočit a transformovat tak do vhodné polohy.

Mnoho metod využívá možnosti převodu obrázku do vektorové podoby a jeho následné zobrazení do prostoru, kde jsou jednotlivé obličejové reprezentovány jako body, a my řešíme podobnost vůči referenční množině bodů.

Zde se pak dostáváme k problému vhodné indexace dat, kvůli potřebě rychlého přístupu k informacím a v neposlední řadě k postupu a principu měření vzdáleností (různorodosti) mezi obličejí. Již na první pohled je jasné, že tato úloha nebude triviální a rozhodnout, který rys nebo vlastnost ukládat či ne, porovnávat či z aktuálního porovnávání vyřadit, a mnoho dalších jsou problémy, u kterých není jednoduché rozhodnout, která metoda bude obecně lepší.

Samotné rozpoznávací metody jsou tedy založeny na extrakci podle barvy kůže, rtů i vlasů, což může mít různé výsledky u afroamerických jedinců v porovnání se severskými státy tudíž už jen nalezení odpovídajícího spektra barvy kůže je problematické.

Vzmemme-li v úvahu potřebu identifikovat všechny jedince na naší planetě, dojdeme k závěru že žádný rys nemůže být v podstatě sám o sobě identický (jednoduchou úvahou je snaha o vytvoření 6 miliard rtů s odlišnými vlastnostmi a schopnost je posléze identifikovat). Tedy můžeme uvažovat, že rozpoznání obličejí není závislé tedy na přesnosti identifikace daného rysu, ale spíše na vhodném určení antropometrických veličin, tedy vzdáleností těchto rysů.

Pokud se budeme dále zabývat touto problematikou, dojdeme k závěru, že existuje celá řada vhodných metod a postupů, ale záleží zřejmě na způsobu použití, kvalitě vzorků dat, formou reprezentace v databázi, vyhledávání podobnosti mezi obličejí a výsledné použitelnosti, na což se budeme snažit najít odpověď v nadcházejících kapitolách.

Na tuto problematiku bylo myšleno i při vytvoření MPEG-7, ve kterém jsou prostředky pro realizaci řešení porovnání obrázků s množinou normalizovaných obrázků. V této práci se pokusíme nalézt problémy a hledaná řešení jednotlivých problémových znaků.

V druhé kapitole se zamyslíme nad základními typy rozpoznávacích úloh a rozdělíme tak problematiku na identifikační a verifikační model. Třetí kapitola se zabývá vzhledem vstupních dat

vhodných pro identifikaci a jejich normalizace, tedy homogenní ukládání obličejů. Ve čtvrté kapitole představím způsob reprezentace obličejů ve vysokodimenzionálních prostorech, což je teoretický základ pro většinu reprezentačních metod dalších kapitol.

Další, tedy pátá kapitola se zabývá přípravou vstupních dat pro identifikační proces. Prochází problematikou lokalizace obličeje v obraze s komplexním pozadím a zamýšlí se nad rozhodovacími procesy, které vedou k nalezení obličeje s vysokou pravděpodobností úspěchu.

Jakým způsobem popsat rysy tváře je předmětem šesté kapitoly a metody na základě lineární a nelineární analýzy obličejů.

V sedmé kapitole se dostávám k hlavnímu tématu této práce, což je identifikace obličeje, kde nastíníme problematiku a možná řešení měření vzdáleností dvou vektorů v prostoru a indexovací techniky pro přístup k datovým strukturám.

Programovým jádrem mé aplikace je standard MPEG-7 rozebraný v osmé kapitole, kde čtenář nalezne podrobný popis tohoto standardu se zaměřením na použité extrakční metody, popisovací struktury tváře a podobnostní vyhledávání.

V deváté kapitole se pak zabývám konkrétními výsledky a testováním na různorodých datech, kterými jsou volně dostupné databáze obličejů s celkovým počtem 4200 snímků.

2 Druhy rozpoznávání

Scénáře rozpoznávání obličeje mohou být klasifikovány do dvou základních typů, verifikace obličeje (autentizace) a identifikace obličeje (rozpoznání) [12]. Ve Face Recognition Vendor Test 2002 [1], provedený Mezinárodním Institutem Standardů a Technologií (NIST), přidal třetí scénář nazvaný seznam sledovaných (watch list).

2.1 Identifikace obličeje

Jedná se o porovnávací proces 1:N, kde srovnáváme hledaný obraz s tváří, oproti všem referenčním obličejům uloženým nejčastěji v databázi a snažíme se určit podobnost mezi těmito obrazy. Tento proces bývá nejčastěji realizován tím způsobem, že se hledaný obraz uloží do databáze a má tedy nejvyšší podobnost s hledaným obličejem. Identifikační proces je „uzavřený“ proces, který porovnává různé vlastnosti obrazu a je individuální pro většinu databází. Testované rysy obličeje musejí být nejprve normalizované, tedy pootočený a zarovnaný podle os tak, aby bylo možné je porovnat s dalšími obličejí, a poté mohou být postupně srovnávány s dalšími rysy obličejů uloženými v databázi.

Výsledkem je tedy míra podobnosti pro každé srovnání, které jsou později srovnány sestupně tak, aby bylo možné jednoduše projít podobné obličeje od nejvíce odpovídajících, až po ty nejméně podobné. To procento obličejů, které bylo ohodnoceno nejvyšší mírou podobnosti je označeno jako seznam nejlepších srovnání. Pokud některé z nejlepších srovnání koresponduje s testovaným objektem, je považováno v rámci srovnávání za úspěšné. Dalším aspektem srovnávání je počet správně nalezených objektů, který bývá označován jako Souhrnný poměr porovnání, což je křivka hodnot podobnosti nejlepších nálezů oproti správným identifikacím [12].



Obrázek 2: Modelový příklad identifikační úlohy.

2.2 Verifikace obličeje

Jedná se o vyhledávání 1:1, tedy snaha o nalezení referenčního obličeje, jehož identita je již potvrzena, k našemu hledanému obličeji. Jelikož se nám nabízí hned celá řada různých verifikačních algoritmů založených na rozdílných metodách, máme tedy potřebu zhodnotit tyto metody podle nějakého verifikačního poměru.

Hodnocení probíhá tedy podle klasifikace výsledků do 4 základních tříd [57]:

- *Správná přijetí* (TP) - Pokud je testovaný subjekt vyhodnocen a identifikován správně.
- *Správná odmítnutí* (TN) – Tato třída je negací třídy TP, tedy jde o správně vyhodnocené odmítnutí testovaného subjektu (Obrázek 1a).
- *Chybná přijetí* (FP) – Zde se řadí případy, ve kterých byl nesprávně identifikován subjekt a identifikace tedy selhala. Zde se mohou objevit úspěšné podvrhy ale i nesprávně identifikované případy. Každý bezpečnostní systém se v zásadě snaží tuto třídu pokud možno minimalizovat, protože se jedná v podstatě o přístup do bezpečné zóny pro subjekt, který na to neměl získat oprávnění.
- *Chybná odmítnutí* (FN) – Do této třídy padnou takové případy, u kterých by měl být subjekt správně identifikován, což se ale nestalo a přístup byl zamítnut.

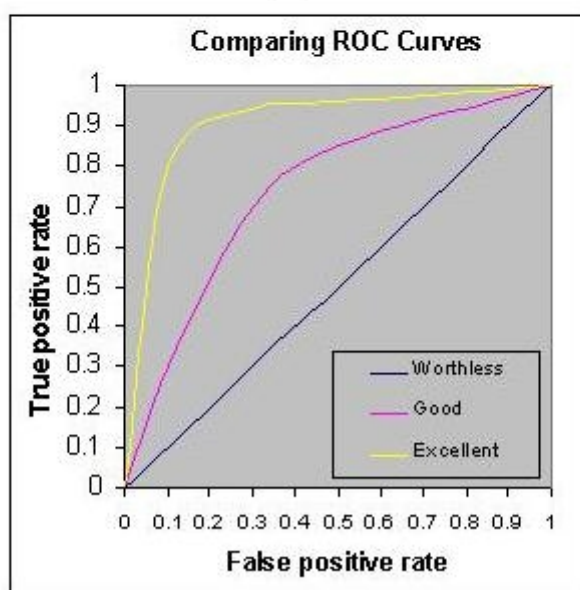
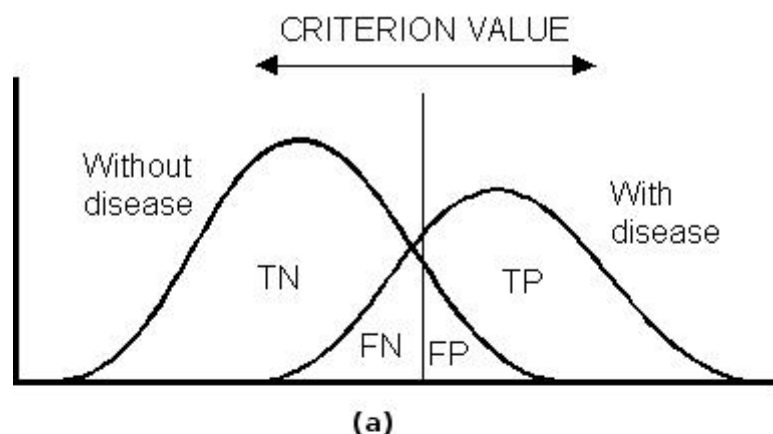
Dále mluvíme o poměru správných přijetí (TPR), což můžeme vypočítat jako poměr správných přijetí a celkového počtu přijetí, což je součet správných přijetí a chybných odmítnutí (Obrázek 1b).

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \quad (2.1)$$

Analogicky k poměru správných přijetí je definován poměr chybných přijetí (FPR), což je poměr mezi chybnými přijetími a počtem odmítnutí, což je součet chybných přijetí a správných odmítnutí (Obrázek 1b) [57].

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \quad (2.2)$$

Verifikační poměr tedy bude poměr mezi poměrem správně povolených přístupů (TPR) oproti poměru chybně povolených přístupů (FPR). Verifikační poměr pro konkrétní úlohu bývá zakreslen pomocí grafu, který se nazývá ROC křivka (Obrázek 1b). Zde mají špatně hodnocené systémy průběh grafu ve tvaru přímky, a čím vyšší je míra kvality testovaného systému, přímka se stále více limitně blíží maximální hodnotě. Dobré verifikační systémy balancují mezi těmito dvěma hodnotami, kde jejich konkrétní hodnoty závisí na konkrétním použití a stupni bezpečnosti.



Obrázek 1: a) Diagnostika pro vytvoření ROC. b) Příklad ROC křivky [13].

2.3 Seznam sledovaných (watch list)

Jedná se o otevřenou testovací metodu, kde testovaný jednotlivec může být (tedy nemusí) uložen v databázi. Hledaný obličej je postupně porovnáván s jinými obličejí uloženými v systémové databázi a míra podobnost je vypsána pro všechna porovnání, která jsou poté srovnána sestupně podle míry podobnosti, a tedy obličej nalezený v databázi, který má nejvyšší míru podobnosti je vypsán jako první. Pokud je míra podobnosti vyšší než přednastavený práh, je automaticky označen, který je indikátorem přítomnosti obličej v databázi.

Zde nás tedy budou zajímat dva aspekty. Prvním z nich bude počet úspěšných nálezů v databázi obličejů nazývaný „Detekční a Identifikační poměr“. Druhým aspektem bude počet systémem chybně identifikovaných testovaných obličejů, jako přítomný v databázi i když tomu tak nebylo, nazývaný „poměr chybného označení“.

3 Vzhled předloh

Velkou mírou se na úspěšnosti porovnávacího algoritmu podílí v neposlední řadě i vzhled obrazu obličeje. Ať už se jedná o obrázek, který chceme porovnávat s databází, nebo zdali se jedná o referenční obrázky umístěné v databázi. Tedy kvalita porovnání může být ovlivněna různými faktory, kterými jsou 3D fotografie obličeje, možnost různých módních doplňků, z nichž významné jsou například sluneční brýle, díky kterým je prakticky znemožněna (nebo alespoň částečně ztížena) možnost lokalizace očí [3].

Mezi další faktory můžeme počítat i růst vousů, které určitě změní uvažovaný tvar a zbarvení obličeje a v neposlední řadě stárnutí, které probíhá konstantě i u všech uložených referenčních obrazů v databázi vůči realitě. Tento fakt musíme u reálně použitých systémů také zohlednit, což můžeme provést vždy aktualizací zmíněné předlohy nebo akceptací určité tolerance v určitých částech obličeje. Dalšími problémy může být kvalita snímku, který může být přesvícený nebo naopak příliš tmavý. Takové předlohy by kvalitní systém měl zřejmě odmítnout, protože identifikační předlohy uložené v databázi by měly mít obecně vyšší kvalitu než obrazy snímané pro konkrétní identifikaci, abychom se pokud možno co nejvíce vyhnuli možnosti chybovosti z důvodů nekvalitních dat. Dalším rušivým elementem bude v neposlední řadě také emoční nálada snímaného subjektu, kterou nelze odstranit pouhými aktualizacemi, protože uvažujeme-li například systém pro identifikaci vstupu oprávněných osob do objektu firmy, může mít subjekt při každém vstupu jinou náladu [16].

Dále musíme počítat s tím, že i jediný člověk může mít v databázi referenční fotku s jinými doplňky než na testované fotografii. Ať už se jedná o brýle, kdy každý člověk, který nosí brýle může mít více rozdílných modelů, nebo ať uvažujeme obměnu módních doplňků, kdy kupříkladu u ženské části populace bývá v celku běžné, že v krátkých intervalech obměňují různé doplňky obličeje. Jedná

se například o náušnice, které nebudou zřejmě v našem případě hlavním problémem, nebo jedná-li se o různé oční stíny nebo rtěnku, které už by mohli zásadně posunout barevné spektrum k jiným než očekávaným (přirozeným) barvám těchto identifikovaných partií. Obzvlášť u mladší generace jsou nyní v oblibě extravagantní barvy (stříbrné, černé, zelené), což zřejmě nemůžeme jakkoliv automaticky rozpoznat a ani tuto chybu předzpracovat, protože barevné spektrum je teoreticky velice široké. Možná esteticky zajímavou, ale z hlediska naší problematiky ne příliš šťastnou moderní změnou je úprava obočí, které si dnes mnohé ženy nechají částečně nebo úplně vytrhat, což může vést k problémům s jedinečností tohoto rysu, protože tímto může být tento rys ovlivněn nanesením a (ne)přesností každé ženy při opakované úpravě. Pokud zajdeme do extrémů, může si takto subjekt změnit parametry tohoto rysu dle potřeb oklamání identifikačního systému.

Jelikož budeme využívat především digitální snímací techniky, musíme uvažovat i šum, který vzniká ovlivněním generovaného elektrického signálu (úměrný fotonu, dle vlnové délky světla) jiným elektromagnetickým zářením (rádiové vlny, mikrovlny) nebo teplotou polovodičových součástek a přidružených integrovaných obvodů. Vzniká tak šum, který se projevuje jako různobarevné zrnění zvláště patrný při špatných světelných podmínkách, což je způsobeno potřebou vyšší citlivosti pro dostatečnou expozici [3].

Pro potřeby rozpoznávání obličeje nejsou tedy potřeba vysoká rozlišení snímačů, ale spíše jejich kvalita, umístění kamery a adekvátního osvětlení scény. Pokud zajistíme vhodné osvětlení, vyhneme se tak vysoké míře šumu a bude tak možné lépe spočítat antropometrické veličiny subjektu .

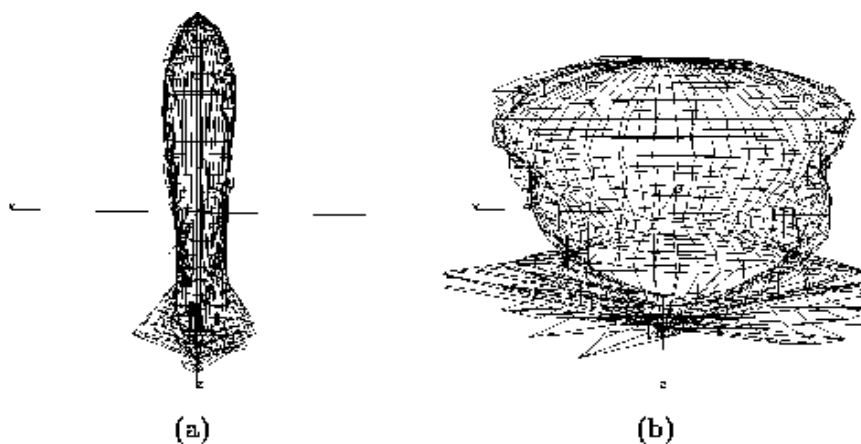
3.1 Normalizace

Pro normalizaci tváře byla navržena hned celá řada způsobů, které pro transformaci využívají zejména nalezení několika kotevních bodů. Základní metody používá zejména homogenní deformaci, která mapuje trojúhelník tvořený třemi vrcholy, které odpovídají očím a ústům. Tato normalizační metoda zachází s obrazem obličeje a jeho okolím jako s listem papíru, který můžeme libovolně podle aktuální potřeby deformovat například změnou měřítka, rotací nebo oříznutím. Tato technika může zaručit jen 2D úpravy, které nejsou závislé na 3D složce objektu.

Objekt v obraze je možné dále transformovat pomocí nehomogenních transformací, které však nemohou zaručit přesné zprohýbání obrazu a proto musíme nalézt důmyslnější přístupy, které očekávají v obraze objekt obecně shodných tvarů [13].

Jako alternativní model tváře se často využívá transformace na elipsoid nebo jednodušší geometrickou strukturu, kterou může být například válec. Na rozdíl od základní Akamatsovi metody má elipsoid schopnost napodobit nerovinné rotace tváře, což je způsobeno díky zakřivení elipsoidu, který realizuje nehomogenní zprohýbání obrazu. Bohužel elipsoid nemůže zahrnout všechna natočení tváře a plně tak normalizovat 2D obraz, což je způsobeno tím, že lidská hlava není docela elipsoidní a je tedy obtížné se co nejvíce přiblížit 3D modelu. Samozřejmě nejpřesnějšími daty by byl 3D obraz získaný pomocí 3D laserového skenování obličeje, což by bylo vhodné zejména u vzorových obličejů uložených v databázi při potřebě co nejpřesnějšího rozpoznání obličeje. Tento proces je však velice nákladný, časově náročný a nelze jej zcela automatizovat. Navíc je zde jasná potřeba sofistikovaného hardwaru, což jej dělá prakticky nedostupným a nelze tedy reálně mít pro každého fotografovaného jedince v databázi odpovídající 3D model.

Při použití jiné individuální 3D modelovací techniky pro normalizaci fotografie, předpokládáme určitý stupeň shody všech lidských tváří, kde deformujeme model podle jeho os. Jedná se především o natažení nebo zmáčknutí modelu lidské tváře a snažíme se tedy o modifikaci modelu libovolně s různými malými lokálními deformacemi (roztahování, prohýbání) tak, aby jej šlo přizpůsobit každému novému jedinci. Tato metoda nám dává docela dobré výsledky a je zdaleka přesnější než rovinné nebo elipsoidní modely. Minimální a maximální deformaci poskytující tento přístup je zobrazen níže na obrázku 2 [13].



Obrázek 2: Deformace 3D modelu podle vertikální osy (a) zmáčknutí (b) roztahení [13].

4 Reprezenční vektory

Obrazová data mohou být reprezentována pomocí vektorů, resp. pomocí bodu ve vysokorozměrném prostoru. Pokud tedy máme 2D modelový obrázek o rozměrech X a Y , můžeme jej mapovat jako vektor $v \in \mathcal{R}^{(XY)}$. Tento prostor můžeme definovat jako zřetězení každého řádku nebo sloupce našeho modelového obrazu.

Je ovšem jasné, že v reálném světě máme řadu omezení, která vyplývají z běžně dostupných technických prostředků. Data tedy budou ve skutečnosti ležet v méně rozměrných podprostorech, do kterých budou systematicky rozdělena. Hlavním cílem podprostorové analýzy tedy bude identifikovat konkrétní data, reprezentovat je a pokusit se parametrizovat rozdíly mezi obrázky nějakým optimálním řešením.

Vektor $X_i = (x_1, x_2, \dots, x_{(N-1)}, x_N)$, který reprezentuje matici $I \times N$, kde x_i je vektor dimenze n . Nyní se zaměříme na příklad z úvodu této kapitoly, kde jsme měli obrázek o rozměrech $n = x \times y$. Číslo n reprezentuje počet všech pixelů v obraze a je tedy i dimenzí vektoru v , dále číslo N je počet všech obrázků v databázi. Nyní můžeme s obrazovými daty pracovat jako s vektorem, což nám rozšiřuje spektrum použití a modelování různých systémů [52].

5 Lokalizace obličeje

Existuje mnoho přístupů určených k rozpoznávání objektů v obrazech, většina z nich závisí na převodu a reprezentaci obrazu do vektorového prostoru. Obraz je v této metodě považován za vysokorozměrný vektor, resp. bod ve vysokorozměrném prostoru.

Je jasné, že tento převod a reprezentace není triviální úlohou, existuje tedy hned celá řada metod a způsobů, jak jej reprezentovat (Obr. 3). Mnoho z těchto metod využívá pouze statické techniky analýzy a modeluje tak vektor v daném vektorovém prostoru, tudíž lze tedy vytvořit efektivně relativně přesné zobrazení ve vektorovém prostoru [3].

Vektorová reprezentace obrazu nám dále umožňuje užití různých učebních technik pro analýzu a syntézu z obrazů. Pokud se na rozpoznávání obličeje tedy podíváme v širším záběru a znalostí těchto vektorových metod reprezentace obrazu, můžeme jej pak pokládat za problém nalezení vhodného podprostoru řešení dané úlohy v kombinaci se strojovým učením.

5.1 Statistické (strukturální) metody

Metody tohoto přístupu se snaží o rozpoznávání jednotlivých dominantních částí obličeje (oči, ústa, nos...) předkládaného vzoru, mezi kterými jsou následně určeny neměnné antropometrické veličiny. Po nalezení těchto rysových vlastností tváře jej je možno normalizovat, což je snaha o oproštění od předpokládaných rušivých vlivů (šum, rušení, poloha ve scéně, velikost...). Zmíněné metody se prolínají do celé oblasti zájmu a tvoří kvalitní základ pro řešení problematiky. Zde narážím hlavně na principy detekce kůže, očí a úst, na základě vlnkových transformací obrazů.

5.2 Strojové učení

Bez strojového učení si asi nelze představit moderní identifikační systém, protože všechny systémy si ukládají obličeje do databází a ta tvoří trénovací základ pro úspěšnou identifikaci osoby.

Příznačné pro tento přístup jsou kombinace metody backpropagation (metoda zpětného učení neuronové sítě), základní analýzy komponent (principal component analysis - PCA) a dekompozice

jedinečných hodnot (singular value decomposition - SVD). Představa redukcionismu je obecná praxe v rozvoji inteligentních systémů - návrh řešení komplexních problémů prostřednictvím postupné dekompozice úkolu do následných modulů [2].

5.2.1 Haarovy filtry

Haarovy filtry jsou forma $(1, -1)^T$, kterou můžeme považovat za dvourozměrný vektor, kde T je transpozice. Pokud tento atomický předpoklad zobecníme, mají Haarovy filtry tvar M -rozměrného vektoru $(g_1, g_2, \dots, g_M)^T$, kde $g_i = 1$ nebo -1 . Tento zobecněný vektor nazýváme **Haarův dlouhý filtr**. Originální Haarův filtr je horní propust, protože $1 + (-1) = 0$. Haarův dlouhý filtr splňuje podmínku

$$\sum_{i=1}^M g_i = 0, \quad (5.1)$$

pak se stává horní propustí. Tato podmínka je však pro naše účely příliš přísná, později se budeme zabývat obecnější formou [13]

$$\sum_{i=1}^M g_i \approx 0. \quad (5.2)$$

5.2.2 Strojové učení Haarových filtrů

Nejprve zkonstruujeme N obdélníkové trénovací obrazy obličejů pro učení Haarových filtrů. Nyní tyto obdélníky rozdělíme do $M \times M$ čtvercových bloků z nichž každý má velikost $m \times m$. Nyní vypočítáme počet všech elementů obsažených v v -tém bloku obrazu obličeje a konstruujeme málo výskytový obraz $C_{(i,j)}^v, i, j=1, \dots, M$. Tento obraz může být považován za množinu M řádků vektorů. Reprezentujeme i -tý řádkový vektor jako sloupcový vektor

$$C_i^v = (C_{(i,1)}^v, C_{(i,2)}^v, \dots, C_{(i,M)}^v)^T, \quad (5.3)$$

Nyní můžeme vyjít z definice Haarova dlouhého filtru a můžeme definovat

$$g_i^T C_i^v = \sum_{j=1}^M g_{(i,j)} C_{(i,j)}^v \quad (5.4)$$

Pak tedy chceme definovat odchylku $g_i^T C_i^v, v=1, \dots, N$, tedy

$$o_i^2 = \frac{1}{N} \sum_{v=1}^N (g_i^T (C_i^v - C_i))^2, \quad (5.5)$$

kde

$$\bar{C}_i = \frac{1}{N} \sum_{v=1}^N C_i^v. \quad (5.6)$$

Odchylka o_i^2 může být přepsána jako

$$o_i^2 = g_i^T V_i g_i \quad (5.7)$$

Zde V_i označuje matici doplňků odchylky, a je tvořena

$$V_i = \frac{1}{N} \sum_{v=1}^N (C_i^v - \bar{C}_i)(C_i^v - \bar{C}_i)^T. \quad (5.8)$$

Naše metoda pro strojové učení dlouhého Haarova filtru maximalizuje odchylku v M -rozměrném binárním vektorovém prostoru $S = (g_1, g_2, \dots, g_M), g_i = \{1, -1\}$. Nyní se dostáváme ke kombinatorickému optimalizačnímu problému 2^M kombinací z binárních vektorů. Tato metoda je v podstatě stejná jako PCA přístup, kterou probereme níže. Tento problém se řadí do množiny NP problémů a pokoušíme se tedy získat reálné vektorové řešení, které se bude blížit binárnímu vektoru v obecnější formě. Nyní jej můžeme převést na následný minimalizační problém:

$$J = \min_M \left(-g_i^T V_i g_i + \frac{K_1}{2} \left(\sum_{j=1}^M g_{i,j} \right)^2 + \frac{K_2}{4} \sum_{j=1}^M (g_{i,j}^2 - 1)^2 \right) \quad (5.9)$$

Zde jsou K_1 a K_2 pokusné konstanty, kde K_1 vychází z obecné podmínky Haarova dlouhého filtru a druhá K_2 vychází z aproximace binárních vektorových řešení. Můžeme se pokusit nalézt lokální minima, kde používáme Newtonovu metodu, protože nalézt globální minima je obtížné [56].

5.2.3 Support Vector Machines (SVM)

Tvoří sadu nástrojů pro metody strojového učení užívaného pro klasifikaci a regresi. Zvláštní vlastností SVM strojů je souběžná minimalizace empirické klasifikační chyby a maximalizace geometrických okrajových oblastí. Podíváme-li se na vstupní data jako na dvě soustavy vektorů v n -rozměrném prostoru, SVM konstruuji oddělovací rovinu maximalizující okraj mezi dvěma

skupinami dat. Vytvoříme tedy takovou oddělovací rovinu, která má co největší vzdálenost k sousedním bodům obou tříd.

5.3 Počítačové vidění

Lidské obličejové mají zřetelnou strukturu, která se liší od ostatních objektů v obraze. Obvykle se oddělují do sebe kůže, vlasy a ostatní objekty v obraze. Využívá se barevné informace obrazu pro hledání obličejové textury. Přestože je tato metoda poměrně rychlá, bývá využívána spíše při složitějších realizacích [3].

5.3.1 Detekce Kůže

Jedná se v podstatě o základní a velice účinnou metodu využívanou v mnoha aplikacích. Výhodou této metody je především rychlost, malá citlivost na změnu světelných podmínek na scéně, necitlivost na natočení tváře vůči kameře. Tato metoda se uplatňuje i při detekci obličejové v obraz s velmi komplexním pozadím. Její nevýhoda je však celkem zřejmá a to chybná detekce v případě přítomnosti shodné barvy na pozadí v obraze. Proto se tato metoda kombinuje s dalšími, aby se tato vlastnost potlačila [3].

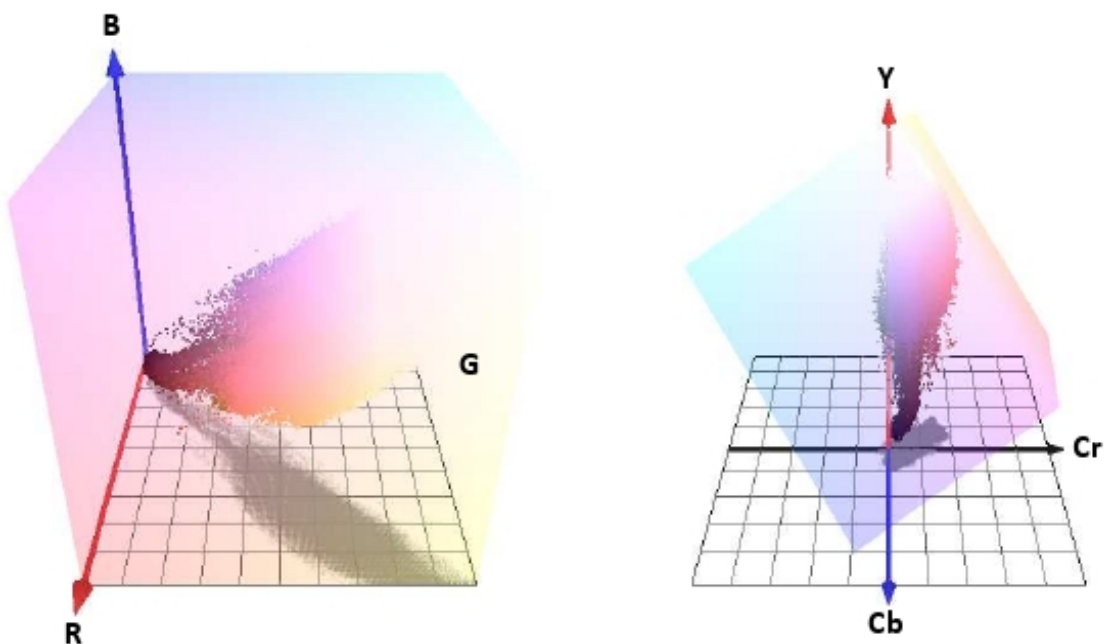
Princip této metody tedy v důsledku vychází z předpokladu, že barva lidské kůže se nachází v určitém ohraničeném podprostoru barevného prostoru. Pokud tedy můžeme vymezit barevnou paletu lidské kůže, můžeme ji tak modelovat a pak klasifikovat pixely obrazu.

Postupujeme tedy tak, že nejprve vybereme vhodný barevný model a následně zvolíme model barvy kůže a jeho přesnost a nakonec stanovíme kritéria pro klasifikaci.

5.3.1.1 Výběr vhodného barevného modelu a jeho přesnost

Snažíme se najít takový barevný model, který odpovídá barvě kůže a můžeme tedy rozlišit barvu kůže v obraze. Tato oblast zájmu se nazývá kolorimetrie. Základním barevným modelem používaným při zpracování obrazu je RGB, kde je barva definovaná jako složka tří barev (červená, modrá, zelená) o různé intenzitě. Problémem tohoto modelu je že explicitně neodděluje jasovou a barevnou složku. Pro práci s obrazem je lepší model YCbCr, kde Y reprezentuje jasovou složku, Cb a Cr barevné chrominační složky [3].

Nyní tedy potřebujeme definovat podprostor barvy kůže v barevném modelu. Můžeme ji vyjádřit explicitně na základě rozložení barev kůže v barevném modelu, nebo experimentálně srovnáváním různorodých testovaných vzorů. Výhodou je bezesporu jednoduchá implementace a rychlost klasifikace. Můžeme i výrazně specifikovat šířku pásma detekované oblasti a pokud bychom dále rozdělili tyto vzory například podle světlosti kůže do skupin.



Obr. 4: Podprostor barvy kůže v modelu RGB. **Obr. 5:** Podprostor barvy kůže v modelu YCbCr [3].

Mezi těmito dvěma barevnými modely existuje převodní vztah který má tvar

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad [3]. \quad (5.10)$$

Nyní chceme vyjádřit daný podprostor a zjistit tedy rozložení barvy kůže z trénovací množiny bez explicitního vyjádření pomocí pravidel. Dále následuje konstrukce pravděpodobnostní mapy, což je přidělení pravděpodobnosti jednotlivým bodům prostoru, s jakou daný bod spadá do barvy kůže. Tento přístup nazýváme neparametrický, oproti němu parametrický přístup se snaží popsat rozložení barvy kůže v daném barevném modelu za pomoci matematického modelu, která získáváme z trénovacích dat. Podprostor barvy kůže má obvykle eliptický tvar s normálním rozložením bodů.

5.3.2 Gaborova vlnková transformace

Princip metody spočívá ve vhodném výběru vrcholových bodů s vysokou energií jako hlavních bodů pro aproximaci. V porovnání s předdefinovanými uzly grafu modifikačních modelů má Gaborova transformace lepší reprezentaci kvalifikace. Tyto charakteristické body jsou automaticky extrahovány použitím lokálních charakteristik jednotlivých tváří (obrázek 12a) [15].

Při detekci rysů obličeje vycházíme z předpokladů, že v obličeji je ve vertikálním směru řada ostrých přechodů, kdežto v horizontálním směru jsou tyto změny v podstatě zanedbatelné. Použijeme tedy detailní koeficienty vlnkové transformace ve vertikálním směru prvního stupně rozkladu za použití daubeschiesové vlny 2. typu, kde vysoké hodnoty koeficientů určují přítomnost ostrého přechodu. Pro účely pokrytí větší části obličeje těmito vysokými hodnotami použijeme dále filtraci dolní propustí za předpokladu že důležité koeficienty roztáhneme spíše vertikálně, což vychází z antropologie obličeje. Proto použijeme průměrovací funkci sinusového tvaru. Nyní jsme získali pole koeficientů, které musíme upravit na velikost binární mapy [3].

Gaborova funkce byla nejprve navržena Denisem Gaborem jako nástroj pro rozpoznání signálu v šumu. Gabor ukázal existenci kvantového základu pro informaci. Gaborovy filtry byly v podstatě znovuobjeveny a zobecněny Daugmanem pro 2D a nyní se používají v různorodých oblastech počítačového vidění [15].

$$\Psi_i(\vec{x}) = \frac{\|\vec{k}_i\|^2}{\sigma^2} \cdot e^{-\frac{\|\vec{k}_i\|^2 \|\vec{x}\|^2}{2\sigma^2}} \cdot [e^{j\vec{k}_i \cdot \vec{x}} - e^{-\frac{\sigma^2}{2}}] \quad (5.11)$$

Každé Ψ je rovinná vlna charakterizované vektorem k reprezentovaný gaussovou funkcí, kde σ je její směrodatná odchylka. Střední výskyt i -tého filtru získáme z charakteristiky:

$$\vec{k}_i = \begin{pmatrix} k_{ix} \\ k_{iy} \end{pmatrix} = \begin{pmatrix} k_v \cos \Theta_\mu \\ k_v \sin \Theta_\mu \end{pmatrix}, \quad (5.12)$$

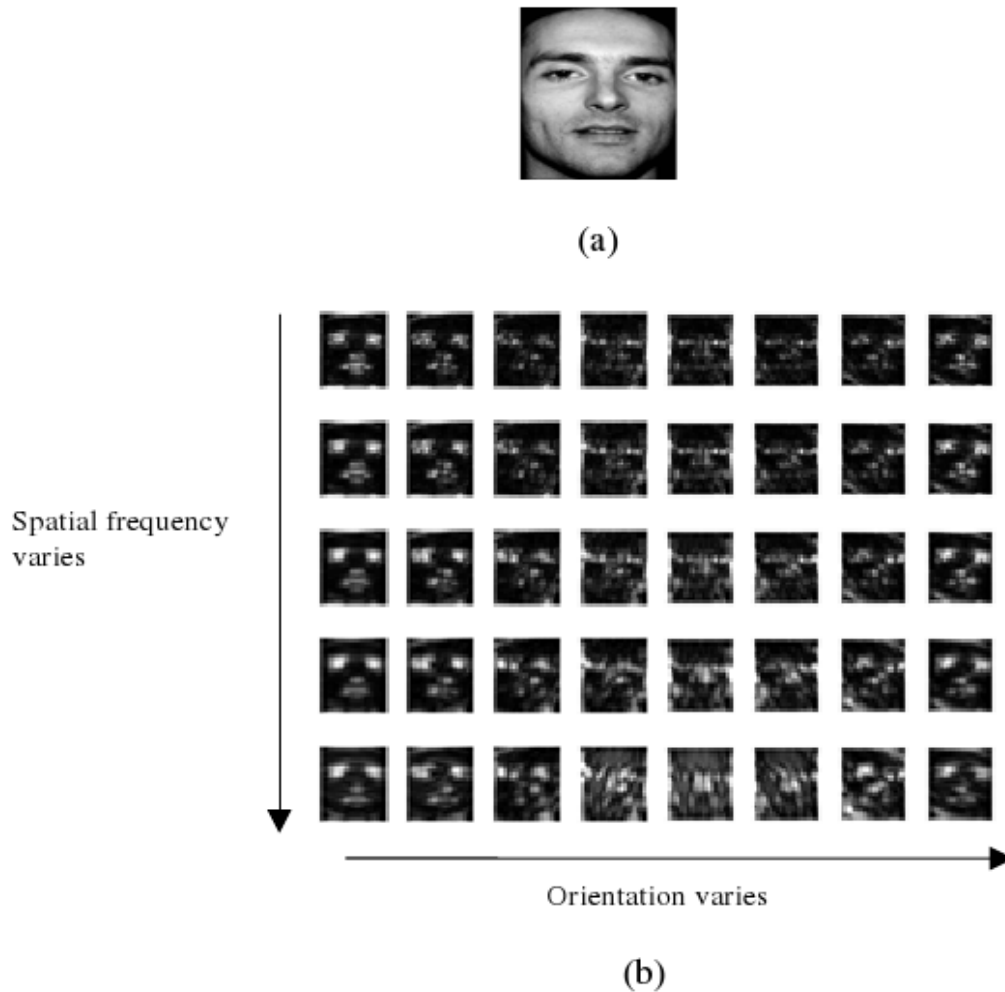
kde máme stupeň a orientaci danou (k_v, Θ_μ) a dále první proměnná v závorkách určuje oscilační část jádra a druhá proměnná kompenzuje ovládací hodnotu jádra. Při odečítání se stává Gaborův filtr necitlivý na celkovou úroveň osvětlení [15].

Daughman a jiní navrhovali, že soubor jednoduchých buněk je nejlepší variantou Gaborova vlnkového vzorkování kmitočtové domény chování. Tato třída je ekvivalentní k souboru souvislých stavů vygenerovaných za pomoci rotací a rozšíření (dilatací), kterou nazýváme Waveletova transformace obrazu, definovaná jako:

$$R_i(\vec{x}) = \int I(\vec{X}') \Psi_i(\vec{x} - \vec{x}') d\vec{x}', \quad (5.13)$$

kde $I(x)$ je intenzita obrazu vektoru x .

Úkoly rozpoznávání obličejů pro lidské bytosti patří mezi základní vlastnosti a schopnosti všech lidských osob, což můžeme chápat jako biologicky motivovaný Gaborův filtr, který je velice dobře adaptovaný na tuto problematiku. Tento proces probíhá ve zrakové kůře mozku, kde jsou jednoduché vlny nahrazeny Gaussovou obálkovou funkcí. Obraz může být tedy reprezentovaný pomocí Gaborovou vlnkovou transformací, která dovoluje popis prostorové struktury a vztahů v ní existujících. Na obrázku máme vstupní tvář a amplitudu Gaborovy filtrace [15].



Obrázek 6: Příklad obrazu obličeje a výsledků Gaborovy filtrace. a) originální tvář z databáze, b) výsledky Gaborova filtru [15].

Extrakční algoritmus této metody má dva hlavní kroky. V prvním kroku lokalizujeme hlavní body a v druhém kroku spočítáme hlavní vektory. Hlavní vektory jsou extrahovány tedy z bodů v obraze, které jsou nějakým způsobem významné. Jedná se například o body koutků očí a úst nebo spodní hrana nosu. Počet hlavních vektorů a jejich umístění v obraze se může měnit za účelem zlepšení různorodé reprezentace charakteristik obličeje různých tváří, jako jsou například důlky, což jsou rysy které také charakterizují konkrétní lidské tváře. Z výsledků Gaborova filtru jsou vrcholy nalezené hledáním pozic v okně W_0 o velikosti $W \times W$ pomocí následující procedury, kde hlavní body jsou (x_0, y_0) [15]:

$$R_j(x_0, y_0) = \max(R_j(x, y))$$

$$R_j(x_0, y_0) > \frac{1}{N_1 N_2} \sum_{x=1}^{N_1} \sum_{y=1}^{N_2} R_j(x, y) \quad , \quad (5.14)$$

kde R_j je výslednou Gaborovou transformací obrázku obličeje, N_1 a N_2 určují velikost obrazu tváře a střed okna W_0 je v (x_0, y_0) . Velikost okna W je jedním z důležitých parametrů. Obraz nesmí být příliš malý, aby byly zřetelné důležité rysy, a zároveň ne příliš velký, abychom se vyhnuly nadbytečným datům. Při konstrukci hlavní mapy tváře využijete všech 40 Gaborových filtrů.

Hlavní vektory jsou vygenerovány z hlavních bodů jako složení Gaborova k -tého transformačního koeficientu i -té tváře je vymezena jako :

$$v_{i,k} = \{x_k, y_k, R_{i,j}(x_k, y_k) | j=1, \dots, 40\} \quad [15]. \quad (5.15)$$

Máme tedy celkem 42 složek, protože musíme započítat i výchozí body. Tyto body jsou velice důležité zejména v době, když ještě nemáme žádné další informace o dalších vektorech, a pro porovnávací proces. Zbývajících 40 složek jsou vzorky Gaborových filtrací. Ačkoliv máme hned celou řadu metod využívající různé okrajové informace, výběr hlavních bodů je vždy realizován pomocí Gaborových vlnkových transformací. Poskytují reprezentující prostorové frekvenční struktury a prostorové vztahy lokálních oblastí kolem daných hlavních bodů.

5.3.3 Morfologické operace

Po redukci oblasti bodů pomocí Gaborovy vlnkové transformace mohou v mapě zůstat body, které ale nepatří do žádné obličeje oblasti a i přesto splňovaly námi stanovené podmínky. Může se jednat a menší oblasti bodů nebo celé plochy v případě tvarových či barevných shod s charakteristickými tvary obličeje. Musíme se tedy pokusit tyto chyby nalézt, zobecnit a odfiltrovat (Obrázek 7b).

Morfologie je obsáhlá sada operací pro zpracování obrazu založené na charakteristických tvarech v obraze. Jedná se v podstatě o transformaci vstupního obrazu na výstupní obraz o stejné velikosti. Při aplikaci těchto operací hodnota každého výstupního bodu závisí na porovnávání odpovídajícího pixelu s jeho sousedy ve vstupním obraze [17].

Provedeme tedy morfologické operace uzavření a otevření, které odstraní samostatné body a seskupí zbylé body do větších celků [3]:

$$\begin{aligned} X \cdot B &= (X \odot B) \ominus B \\ X \circ B &= (X \oplus B) \odot B \end{aligned} \quad [3], \quad (5.16)$$

kde \odot je operace dilatace a \ominus je operace eroze, což jsou základní morfologické operace. Operace dilatace přidává pixely hranicím objektů v obraze, zatímco eroze naopak odstraňuje pixely na hranicích objektů. Aplikujeme-li tedy morfologickou operaci dilatace na vstupní obraz, hledáme tedy maximální hodnotu všech pixelů sousedících s odpovídajícím pixelem [17, 18]. Potřebujeme-li operaci aplikovat na binární obraz, tak pokud je hodnota některého z pixelů 1, je i výstupní pixel roven 1. Při erozi naopak hledáme minimální hodnotu, takže analogicky je-li hodnota některého i sousedícího pixelu rovna 0, je i výstupní pixel roven 0 [17] (Obrázek 7c). Operace jsou definovány jako [3]:

$$\begin{aligned} X \odot B &= \{p \in \mathcal{E}^2 : p = x + b, x \in X, b \in B\}, \\ X \ominus B &= \{p \in \mathcal{E}^2 : p + b \in X, \forall b \in B\}, \end{aligned} \quad (5.17)$$

Nyní máme objekty v obraze označeny shluky pixelů a můžeme se pokusit použít hranový filtr pro detekci okrajů a obálkovým algoritmem ohraničíme kandidátní místa. Je asi zbytečné používat složité obálkové algoritmy postačí základní metody s hranami rovnoběžnými s osami, takže výsledky jsou obdélníkové obrazy. Nyní můžeme již hrubě oddělit příliš malé a poměrově nevhodné oblasti. Ponecháme tedy kandidátní jenom takové, které mají poměry hran obálky odpovídající obličejí (Obrázek 7c).



Obrázek 7: a) binární maska, b) vynásobená waveletovými koeficienty, c) po provedení morfologických operací s výběrem kandidátských oblastí [3].

5.3.4 Detekce úst

Pro detekci úst využíváme Fisherovy lineární diskriminace FLD (5.18), která vychází z předpokladu, že barevné spektrum rtů je složeno z vysokých hodnot červené a velmi nízké modré barvy modelu RGB [3]. Je jasné, že se toto barevné složení může u různých lidí měnit, ať už jenom částečnou odchylkou nebo úplnou změnou spektra, v případě použití různých líčení (extravagantní rtěnky černých a stříbrných odstínů) a podobně. Tyto problémové případy nemůžeme izolovat, protože by teoreticky mohly tedy rty nabývat všech dostupných hodnot barevného spektra. Budeme tedy uvažovat pouze přirozené barvy rtů a provedeme barevnou transformaci FLD, která je definována [3]:

$$FLD = \begin{bmatrix} -0.289 & 0.379 & 0.038 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (5.18)$$

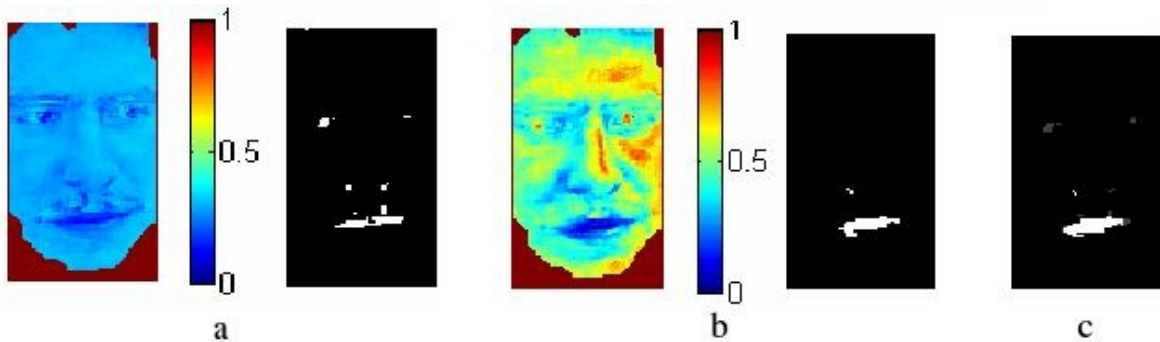
Touto transformací ale nejsme schopni postihnout všechny případy, proto ještě zavedeme pomocnou barevnou transformaci g :

$$g = \frac{G}{R+G+B}. \quad (5.19)$$

Výsledkem obou transformací jsou dvě podobné barevné mapy, ve kterých oblast úst dosahuje velmi nízkých hodnot. Zde opět narážíme na stejný problém jako u detekce kůže, kterou je přítomnost shodné barvy prohledávaného spektra v pozadí při rozlišení těchto rysů. Pokud budou tedy v komplexním pozadí přítomné shodné barvy s vysokou hodnotou červené a nízkou hodnotou modré, nerozpoznáme pouze rty v obličejí. Tento problém odstraníme opět kombinací více metod, tedy využijeme binární masky obličejí a zpracováváme pouze ty části obrazu, které obsahuje binární maska. Pokud se ale zamyslíme nad tímto řešením, dojdeme k názoru, že ovšem tento problém není zcela vyřešen. Pokud se subjekt na obrázku bude například červenat, jsme stále u stejného problému. Využijeme tedy opět kombinace s již zmiňovanou Gaborovu vlnkovou transformací ve vertikálním směru, aby se rty ve výsledném obraze jevíly jako ostrý přechod a Gaborovy koeficienty vynásobíme obě barevné mapy, které dále pomocí jednoduché metody prahování převedeme na binární obraz. Hraniční hodnoty prahových funkcí získáme na základě histogramů jednotlivých barevných map.

Nyní již můžeme binární obrazy sloučit a aplikovat algoritmus Mean-shift, který je předmětem další kapitoly, z něhož získáme souřadnice oblasti s vysokou pravděpodobností výskytu rtů [3].

$$BinObr. = \frac{1}{4} \cdot (BinObr.fld + BinObr.g) + \frac{3}{4} \cdot (BinObr.fld) \cdot (BinObr.g) \quad (5.20)$$



Obrázek 7: a) barevná mapa g a její binární obraz, b) barevná mapa FLD a její binární obraz, c) výsledný binární obraz dle rovnice [3].

5.3.4.1 Mean-shift segmentace a shlukování

Mějme obraz v odstínech šedi, kde každý pixelů $[x,y]$ má určitou intenzitu i . Obraz tedy můžeme reprezentovat jako $[x,y,i]$ v trojrozměrném prostoru, kde první dvě souřadnice jsou souřadnicemi pozice pixelu a třetí souřadnice je intenzita v konkrétním bodě. Pixely o podobné hodnotě intenzity obrazu zřejmě náležejí stejnému objektu v obraze a tedy budou v trojrozměrném prostoru vytvářet shluky. Naopak sousedící pixely, které mají rozdílnou hodnotu intenzity obrazu zřejmě náležejí různým objektům a tedy budou v trojrozměrném prostoru vzdálené [14].

Jedná se tedy o obecný algoritmus, který se snaží iterativně nalézt lokální maxima hustoty vzorků v trojrozměrném obraze. V každém kroku se posunujeme ve směru váženého průměru a hledáme v podstatě těžiště v oblasti kolem současné pozice. Tímto postupem získáme lokální maxima do nichž algoritmus dokonvergoval a dále shlukujeme vzorky, které dokonvergovaly s určitou tolerancí do stejného maxima.

Mean-shift tedy počítá v každém kroku gradient hustoty vzorku v aktuální pozici x ze vzorků v okolí, které je definované jádrem K – radiálně symetrickou funkcí. Často je výhodnější ve vzorcích použít profil kernelu K [14]:

$$K(x) = c_k k(\|x\|^2) \quad (5.21)$$

Nyní záleží na použití konkrétního jádra. Existuje jich hned celá řada [15]:

Uniformní

$$K(x) = \frac{1}{2} \mathbf{1}_{(|x| \leq 1)}$$

Trojúhelníkový

$$K(x) = (1 - |x|) \cdot \mathbf{1}_{(|x| \leq 1)}$$

Epanečnickův

$$K(x) = \frac{3}{4} (1 - x^2) \cdot \mathbf{1}_{(|x| \leq 1)} \quad (5.22)$$

Dále odvodíme profil kernelu k_E a jeho derivaci k'_E . Pro obecný kernel je odhad hustoty v bodě x

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n K \cdot \left\| \frac{x - x_i}{h} \right\| \quad (5.23)$$

kde h je šířka kernelu, n je počet bodů a $x_{1..N}$ jsou vzorky [14]. Nyní můžeme získat gradient dosazením základní rovnice profilu kernelu do předchozí rovnice, dostaneme tak

$$\nabla f(x) = \frac{2c_k}{nh^d} \sum_{i=1}^n (x - x_i) k' \left\| \frac{x - x_i}{h} \right\|^2 \quad (5.24)$$

Můžeme tedy spočítat gradient pro Epanečnickův kernel f_E , dosazením k'_E , c_k zvolíme a běh algoritmu Mean-shift můžeme popsat následujícím pseudokódem.

Pseudokód Mean-shift algoritmu:

```

i = 0;
repeat
{
 $\mathbf{y}_{i+1} = \mathbf{y}_i + \mathbf{f}(\mathbf{y}_i)$ ;
i++;
} until ( abs( $\mathbf{y}_{i+1} - \mathbf{y}_i$ ) < (h/1000))

```

Pokud chceme tuto metodu aplikovat v prostoru s různými veličinami na různých osách je vhodné použít kernel s různou šířkou pro každou veličinu. Pro segmentaci obrazu je vhodné mít

možnost měnit nezávisle šířku kernelu h_s ve směru pozice pixelu a šířku h_r ve směru intenzity, tedy

použijeme normalizovaný vektor $\left[\frac{x}{h_s}, \frac{y}{h_s}, \frac{i}{h_r}\right]$ [14].

Po spuštění Mean-shift získám pro každý pixel lokální maximum $z = [x, y, i]$ a budeme spojovat takové sousední pixely a, b , pro které platí

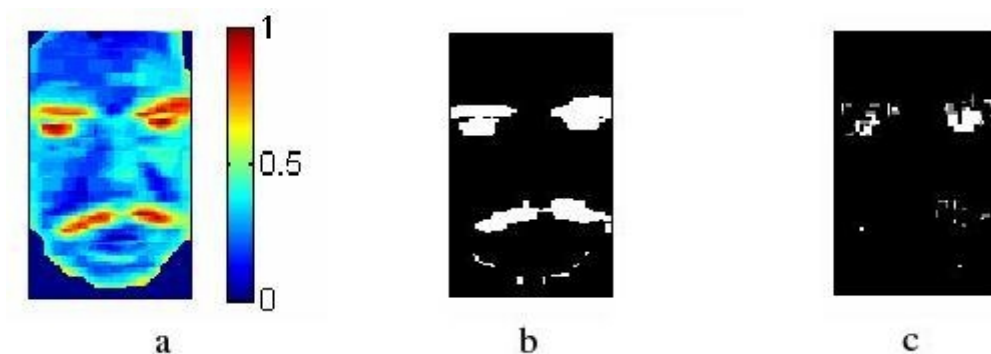
$$\left|z_a - z_b\right| \leq h = [h_s, h_s, h_r] \quad (5.26)$$

Vytvořím binární matici o dvojnásobné délce hran obrazu v němž budou reprezentovány pixely a hrany mezi nimi. V takovéto binární matici můžeme nalézt spojené komponenty [14].

5.3.5 Detekce očí

Chceme-li nalézt oči subjektů v obraze, můžeme postupovat obdobně jako při detekci rtů s tím, že postačí použít pouze jednu barevnou mapu, protože barva zornice a bělma je u většiny lidí velmi podobná [3]. Použijeme průměrnou binární masku obličeje a aplikujeme násobení Gaborovými vlnkovými koeficienty. Barevnou mapu opět převedeme na binární obraz pomocí prahování a algoritmem Mean-shift detekujeme pozici očí. Musíme však dbát i na pozici rtů, protože zuby mohou dosahovat shodných hodnot s barevnou mapou očí [3].

$$EyeMap = \frac{1}{3} C_b^2 + (256 - C_r)^2 + \frac{C_b}{C_r} \quad (5.27)$$



Obrázek 8: a) barevná oční mapa b) její binární obraz c) binární obraz vynásobený Gaborovými vlnkovými koeficienty [3].

5.3.6 Verifikace obličejové oblasti

Můžeme tedy pomocí různých skládání filtrů lokalizovat v obraze pozici rtů, očí a lokalizovat oblast tváře pomocí detekce barvy kůže. Nyní, pokud dáme všechny pozice dohromady uvažujeme předpokládané vertikální i horizontální rozměry obličej, můžeme s velkou určitostí stanovit, zda-li se jedná o obličej nebo ne. Lokalizované rysy musí logicky ležet v oblasti detekované jako kůže a antropometrické veličiny musí být v poměru. Navíc tímto rozdělíme nalezené obličej na ty, které můžeme dále použít (pokud nenalezneme například jedno oko, obličej vyřadíme) a ty které nesplňují základní požadavky [3].

6 Popis (rysů) obličeje

Metody obsažené v této kapitole jsou metody využívané obecně i v případě lokalizace obličejů v obraze. Principy některých přístupů jsou stejné. Zabývají se tedy pouze prohledáváním obrazu s komplexním pozadím a hledají libovolně umístěné markanty, nebo jak budeme nyní probírat, prohledáváme z menší obrázky za předpokladu normalizovaného obrazu a standardního rozmístění markantů v obraze (oči, ústa, nos).

Pro zjednodušení reprezentovaných dat se často využívá obalovacích metod, což je v podstatě snaha o nalezení minimální konvexní obálky jednoduchým objektem jako zobecnění složitějších. Pro příklad ji lze využít pro rychlou detekci kolizí objektů v prostoru, k tomuto účelu se využívají OBB (Oriented Bounding Box) algoritmy. Dalším použitím zejména ve světě 3D grafiky je nalezení množiny objektů pro zjištění viditelnosti daného objektu a filtrace všech objektů v obraze na ty, co se mají vykreslovat a ty, které nejsou buď vůbec v záběru, nebo jsou zakryté za jiné větší objekty.

6.1 Statistický (strukturální) přístup

Provádí se rozpoznávání jednotlivých dominantních částí obličeje (oči, ústa, nos...) předkládaného vzoru, změření antropometrických veličin, jejich normalizace vzhledem k předpokládaným rušivým vlivům (šum, rušení, poloha ve scéně, velikost...), dále porovnání s databází známých fotografií použitím klasifikačních algoritmů, statistické rozhodnutí o relativní podobnosti s takto vybranou množinou obrazů [2].

6.2 Lineární analýza

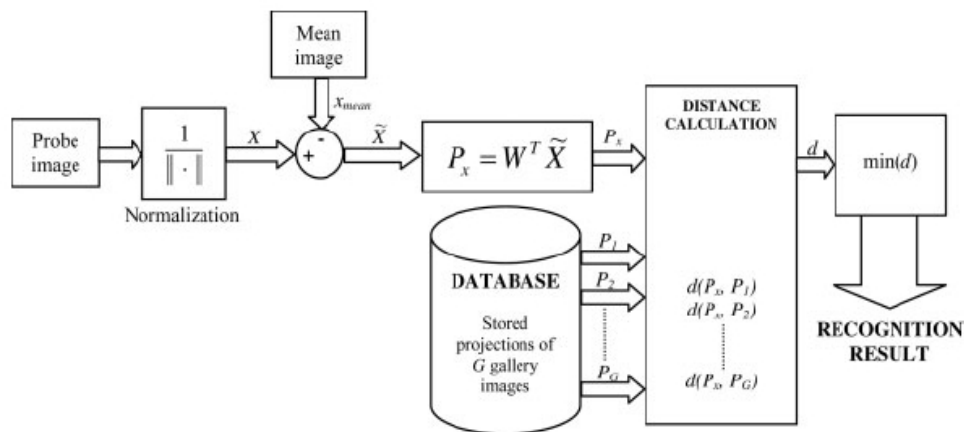
Mezi tři základní metody založené na lineárních vzhledových klasifikátorech, jsou analýza hlavních komponent (PCA), analýza nezávislých komponent (ICA) a diskriminační analýza LDA [4]. Každý ze základních klasifikátorů má svou vlastní reprezentaci (bázové vektory) vysoko-rozměrného vektorového prostoru, který je založený na různých statických metodách. Pomocí možnosti projekce

vektoru tváře na bázevé vektory získáme projekční koeficienty, které jsou základním typem reprezentace tváře v obrazu.

Díky těmto koeficientům můžeme porovnávat obličej a určit tak míru podobnosti mezi testovaným obličejem a porovnávaným obličejem. K tomuto účelu můžeme použít různé metody, mezi základní patří určení kosinu mezi jednotlivými vysokorozměrnými vektory testovaného a referenčního vektoru, který je uložen v databázi. Klasifikační metody tedy můžeme považovat za lineární transformaci z originálního obrazového vektoru do projekčního vektoru:

$$P_x = W^T \tilde{X} \quad (6.1)$$

kde P_x je $n \times N$ vektor matice, n je dimenze tohoto vektoru a W je transformační matice. Celý postup převodu a analýzy obličej je znázorněn na obrázku 5.



Obrázek 5: Hlavní etapy postupu rozpoznání obličeje. Testovaný obraz je nejprve normalizován a sečten s průměrným obrazem, který je převeden a vektorovou reprezentaci a dále porovnáván s ostatními vektory obrazů v databázi. Výsledkem tohoto porovnávání je posloupnost vzdáleností, ze kterých je vybrány vždy minimální hodnocení[4].

6.2.1 Analýza hlavních komponent (PCA)

Mezi hlavní myšlenky této základní lineární metody analýzy hlavních složek množiny bodů v prostoru je nalezení ortogonální báze, pomocí které bude možné co nejlépe vystihnout danou množinu vektorem ve vysokorozměrném prostoru [4].

Tato metoda se využívá především pro nalezení obálky obklopujícího všechny vnořené body, a tudíž ji lze využít pro jednoduchou a rychlou aproximaci složitějších objektů. V některých aplikacích se využívá i pro rychlou aproximaci hodnoty velikosti objektů [53].

Jako jednu ze základních způsobů pro vytvoření obálky kolem bodů ve vysokorozměrném prostoru je metoda AABB (Axis Aligned Bounding Box). Ta řeší obálku pomocí vektorů kolmých na osy daného souřadného systému a hledá tak nejmenší rozměry zvoleného objektu (nebo množiny bodů reprezentujícího objekt). Jeho využití je sporné, protože kvalitních výsledků dosahují pouze u objektů soustředěných kolem jednoho bodu.

Oproti němu metoda OBB řeší tento problém minimálností obálky pomocí os, které nemusejí být nutně kolmé na osy souřadného systému daného dimenzionálního prostoru. Lze tak tudíž vytvořit lepší aproximaci daného objektu v prostoru [4].

Mějme tedy množinu vstupních bodů x_1, \dots, x_n ve vysokorozměrném prostoru \mathbb{R}^d reprezentovaném jako:

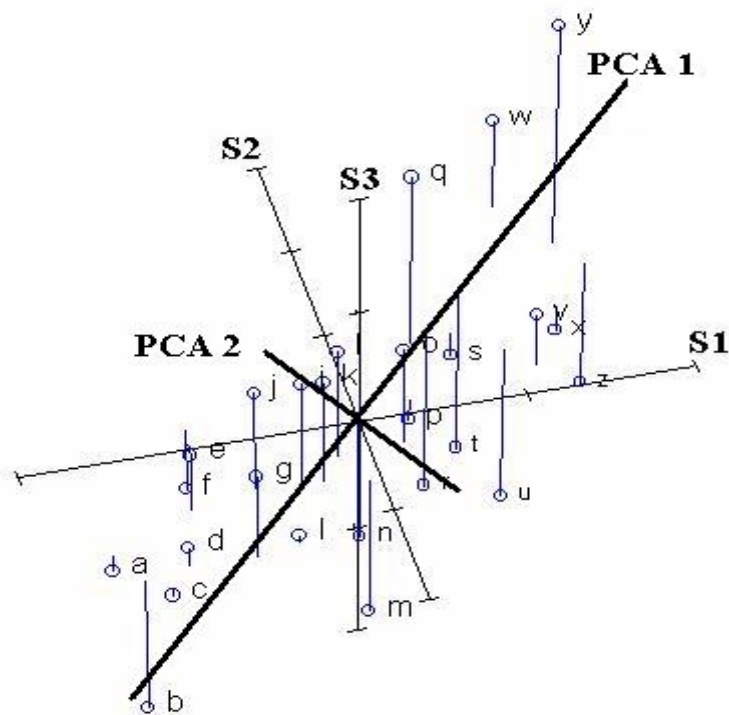
$$\begin{aligned} X_1 &= \langle x_1^1, x_1^2, \dots, x_1^{(d-1)}, x_1^d \rangle \\ X_2 &= \langle x_2^1, x_2^2, \dots, x_2^{(d-1)}, x_2^d \rangle \\ &\dots \\ X_n &= \langle x_n^1, x_n^2, \dots, x_n^{(d-1)}, x_n^d \rangle \end{aligned} \quad (6.2)$$

Pokud chceme nalézt bázi našich vektorů, hledáme vlastně těžiště těchto bodů a nalezneme jej

tak, že $t = \frac{1}{n} \sum_{i=1}^n X_i$ [4].

Máme tedy bod, který vyjadřuje daný podprostor bodů, který jsme si vybrali a pokud nyní vytvoříme průměrnou hodnotu mocnin rozdílu všech bodů tohoto prostoru a dostaneme hodnotu rozptylu všech bodů v prostoru. Můžeme jej vyjádřit jako

$$V(X) = \frac{1}{n} \sum_{j=1}^n \left(x_j - \frac{1}{n} \sum_{i=1}^n X_i \right)^2 . \quad (6.3)$$



Obrázek 9: Nalezení os pomocí metody PCA [52].

Z jiného hlediska lze pomocí této funkce měřit varianci dat v různých směrech a hledat symetrii množiny bodů. Pokud proložíme osu mezi body vysokorozměrného prostoru, můžeme na ni promítnout všechny uvažované body, které jsou nyní rozptýleny kolem ní. Vznikají tak nové body

X' , mezi kterými můžeme nyní zjistit vzdálenosti jednotlivých bodů. Jsou-li na přímce body shromážděny blízko u sebe, bude pak variance nízká, v opačném případě bude vysoká.

Uvažme tedy bod u a jednotkový vektor v takový, že $|v|=1$ a projekci libovolného bodu na

přímku může vyjádřit $\|X'_i - t\| = \frac{\vec{v}}{\|\vec{v}\|} \cdot (X_i - t) = \vec{v} \cdot \vec{y}_i = (\vec{v})^T \vec{y}_i$, jak je zřejmé z výše uvedeného na

obrázku 9. V podstatě můžeme každou symetrickou matici bodů v prostoru rozložit na dvě dílčí matice rotace a měřítka (scale), která má extrémy ve směru vlastních vektorů. Pomocí této vlastnosti můžeme zjistit směry maximální a minimální variance pro danou množinu objektů (bodů) v prostoru [4].

6.2.2 Srovnání Haarových filtrů a PCA

PCA přístup pro strojové učení klade důraz na splnění omezení $g_i^T g_i = 1$. Metoda PCA je založena na výpočtu hlavních komponent vektoru řešením problému. Získané komponenty jsou reálné hodnoty vektorů, z nichž každý zahrnuje základní prvky. V PCA metodách je hlavní báze vektor pro vstupní obrazy získáván expanzí obrazu s ohledem na hlavní komponenty vektorů. Naše idea tedy leží ve výpočtu filtru horní propusti, která je sekvencí 1 a -1. Hlavní vektor má tedy součásti, které vzniknou pomocí Haarových filtrů a řádkového vektoru obrazu. Zde není nutné počítat skalární součin obou vektorů, protože náš filtr sestává z 1 a -1 [56]. Více o PCA si můžete přečíst v další kapitole popisu rysů obličeje.

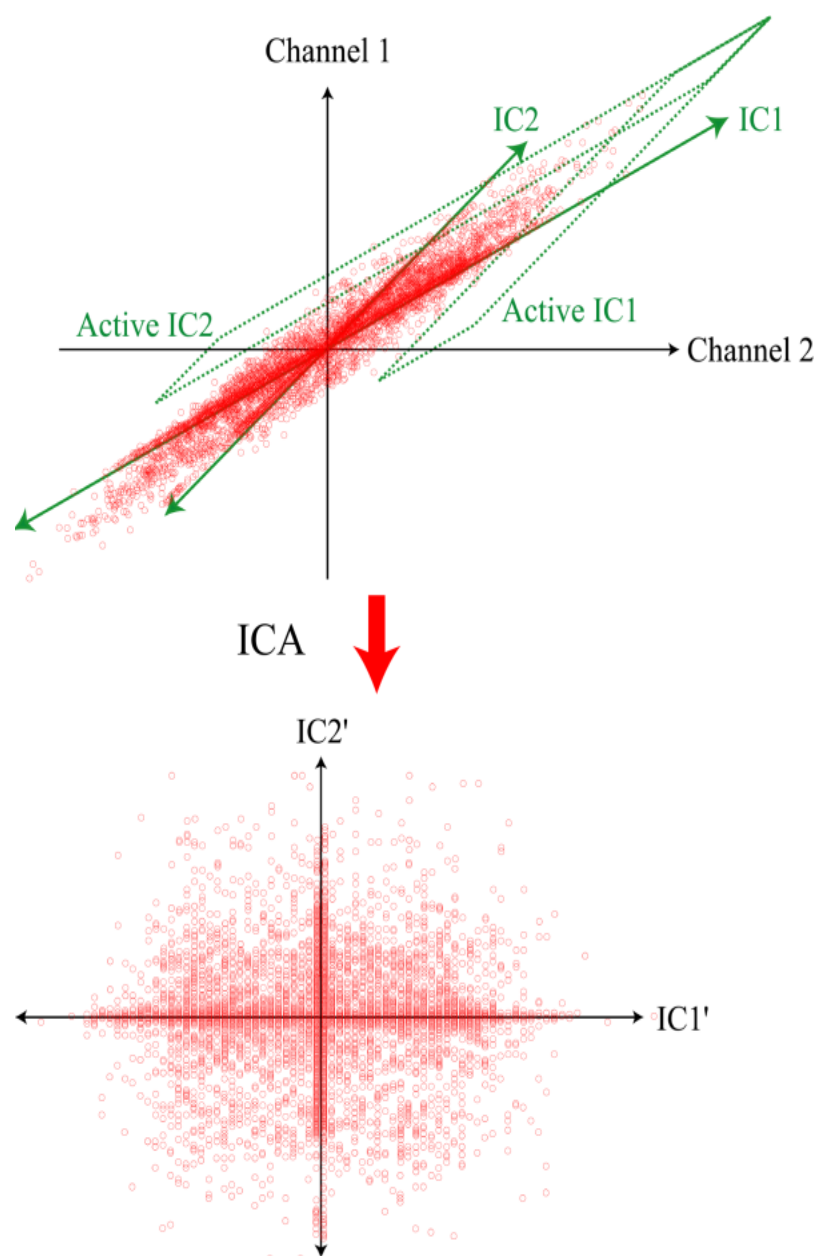
6.2.3 Analýza nezávislých komponent (ICA)

Jedná se o metodu, kde hlavní snahou je nalézt hlavní báze komponenty ze znalosti jejich kombinace. ICA definice předpokládá lineární kombinaci zdrojových vektorů (dat), které bývají velmi často označovány jako komponenty. Pokud budeme uvažovat n -rozměrný vektor vstupního obrázku obličeje a vhodnou míchací matici můžeme je aproximovat za n -rozměrný vektor zkoumaných vlastností. Tuto kombinaci tedy můžeme definovat jako $P_x = W^T \tilde{X}$ (6.1).

Nezávislá analýza komponent je v podstatě podobná jako PCA s tím rozdílem, že distribuce komponent jsou řešeny negaussovským způsobem. Je více architektur řešení, ale všechny jsou založeny na nezávislé analýze částí obličeje, čili staticky nezávislém základu obrazu obličeje a kódové reprezentace. Pokud bychom předpokládali, že směšovací matice W je ortogonální a X_i jsou gaussovské, pak i P_x jsou gaussovské, nekorelované a mají jednotkový rozptyl.

6.2.3.1 Odhad nezávislosti

Součet několika nezávislých náhodných proměnných jako jsou zde X_i , se blíží ke Gaussovskému rozdělení. Takže $x_i = a_1 s_1 + a_2 s_2$ je více gaussovské než rozdělení s_1 nebo s_2 . Z toho tedy plyne, že můžeme najít jakovou kombinaci měřených signálů z Y , které budou mít minimální gaussovské vlastnosti, pak tento signál bude jeden z nezávislých signálů. Aby mohl být nalezen tento signál, je potřeba měřit nenormality součinu [6].



Obrázek 10: Grafy analýzy nezávislých komponent [53].

6.2.3.2 Špičatost

V jistých situacích potřebujeme vědět jakým způsobem jsou data rozptýlena po ose. Je rozdíl mezi tím, jsou-li data rozptýlena rovnoměrně, což bude zřejmě případ ploššího rozptýlení bodů. V případech zjištění záporné špičatosti s jedná o subgaussovské, nebo-li platykurtické a ty, které mají kladnou špičatost jsou nazývány supergaussovskými či leptokurtickými.

Poznamenejme tedy ještě, že špičatost je velice citlivá na extrémny a chyby vznikající při měření dat, není tedy vhodná pro ICA metodu [53].

6.2.3.3 Negativní entropie

Druhou možností jak měřit nenormalitu je užitím negativní entropie. Entropie náhodné veličiny může být interpretována jako střední počet bitů, pomocí níž lze informaci zakódovat. Entropie H je pro diskretní signál x_i definována jako [6]:

$$H(x) = - \sum_{i=1}^n W(x_i) \log W(x_i) \quad . \quad (6.4)$$

Entropie dosahuje malých hodnot pro rozdělení, která jsou jasně koncentrována na jistých hodnotách. Tedy je proměnná jasně rozdělena a je velice ostrá. Pro měrnou nenormalitu, která je obvykle nulová pro gaussovskou proměnnou a vždy kladná, použijeme nepatrně modifikovanou definice rozdílové entropie zvané negativní entropie [6].

6.2.4 Lineární diskriminační analýza (LDA)

Tato metoda umožňuje rozlišení dvou nebo více skupin námi zkoumaných bodů ve vektorovém prostoru. Množina zvolených proměnných je určena svou odlišností v různých skupinách, tedy možnosti přiřadit objekty k jednotlivým typům. Snažíme se tedy vážit a lineárně kombinovat diskriminační proměnné s cílem rozčlenění prvků do skupin, které provádíme pomocí diskriminačních funkcí.

Asi mezi nejznámější diskriminační funkci patří nejspíše *Fisherova lineární diskriminační funkce (FLD)*, která využívá specifické informace třídy. Z různých definicí tříd pomocí různých statistik oddělíme obrazy do korespondujících tříd. Tato metoda je asi nejpřesnější ze všech uvedených lineárních metod.

Lineární diskriminační analýza nalézá ortogonální matici W_{LDA} , maximalizuje determinant rozptylové matice mezi třídami na determinant rozptylové matice třídy, kterou lze definovat takto:

$$W_{LDA} = \arg \max_w \frac{(W^T S_B W)}{(W^T S_W W)} = [w_1 w_2 \dots w_N] \quad , \quad (6.5)$$

kde S_B je definována jako mezi-třídní matice odpovídajícího rozptýlení a S_W je rozptylovou maticí konkrétní třídy:

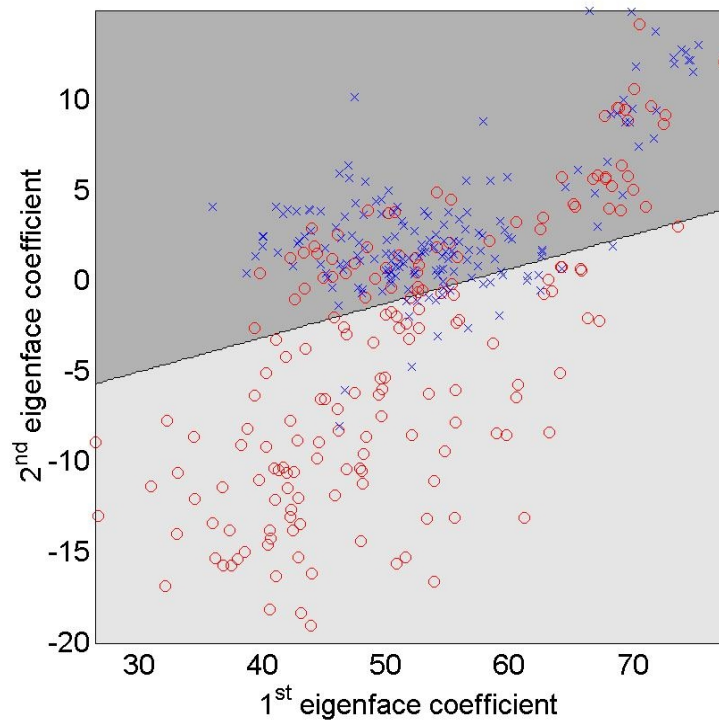
$$S_B = \sum_{i=1}^c N_i (u_i - \bar{x})(u_i - \bar{x})^T, \quad (6.6)$$

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} N_c (x_k - u_i)(x_k - u_i)^T, \quad (6.7)$$

kde N_i je počet trénovacích vzorků ve třídě i , dále c je počet odlišných tříd. Vektor u_i je vektor vzorků dané třídy i a X_i reprezentuje kolekci vzorků patřících do i [49].

Obecně můžeme říci, že PCA nejprve redukuje vektorovou dimenzi před uplatněním LDA. Každý hlavní vektor je reprezentován vektorovou projekcí

$$y_k = W_{LDA}^T x_k, k=1,2,..N \quad [49]. \quad (6.10)$$



Obrázek 11: Rozdělení diskriminační funkcí [54].

6.2.5 Použití

Problematika, kterou se zabývá lineární analýza je na omezené použití, a komplikovanost rozpoznávání obličejů nemůže postihnout v míře potřebné pro problematiku rozpoznávání tváří. Použití metod lineární analýzy je vhodná spíše na problematiku řešící aproximační funkce pro složitější modely a výhodnější tedy je použití nelineární analýzy, což je předmětem další kapitoly.

6.3 Nelineární analýza

Různorodost možností lidské tváře v kombinaci s možnými relevantními změnami, ať už krátkodobými nebo dlouhodobými, vyžaduje složitější modelovací techniky. Použití přímých modelovacích technik různorodosti bývá často využíváno k učení se pro tyto nelineární kopie.

6.3.1 Jádrová analýza hlavních komponent (KPCA)

Jádrové metody se stávají stále častěji populárnějšími a mají významné místo na poli mezi metodami strojového učení. Metody KPCA se používají hlavně pro nelineární mapování ze vstupního prostoru R^M do hlavního prostoru R^L , kde $L > M$. Toto mapování je základem všech funkcí [5]:

$$\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \langle \tilde{\lambda}(\mathbf{x}), \tilde{\lambda}(\mathbf{y}) \rangle_{\tilde{h}} \quad (6.11)$$

Hlavní myšlenkou tohoto teorému je, že v ucelené formě dovolí efektivně rozlišit různorodost nelineárních optimalizačních problémů, které jsme nuceni řešit při regresivním učení a dalších nepřímých problémech.

6.4 Modelový přístup

Rozpoznávací techniky tváře založené na holistickém přístupu směřují k výstavbě pokud možno co nejlepšího modelu odpovídajícího lidské tváři, která by měla zachycovat obměny různých variant obličeje. Všechny předchozí vlastnosti lidské tváře jsou využívány k návrhu modelu. Například metody založené na rysech lidského obličeje pracují tím způsobem, že se snaží nalézt poziční vzdálenost mezi rysy z jednotlivých markantů uvnitř lidského obličeje (oči, nos, rty).

První algoritmy, které se objevily, lokalizují polohu očí podle očních koutků, polohu nosu podle nosních dírek, které jsou výrazně tmavší než okolí, při čelním pohledu obličeje. Princip tohoto algoritmu je v postupném výpočtu všech parametrů pro každou tvář ve srovnání s parametry tváří již nalezenými a vloženými do databáze.

Většina následujících algoritmů pro vyhledávání rysů používaných dnes je založena na Elastických shlukových grafech.

Pokročilejší 3D modifikační modely jsou modelovací techniky pro vytvoření reálného modelu povrchu lidské tváře. Tyto metody modelování obvykle sestávají ze tří základních fází modelovací procedury. Nejprve je potřebné konstrukce odpovídajícího univerzálního modelu lidské tváře. Tento model se budeme následně snažit co nejlépe vymodelovat (přiblížit) reálnému modelu obličeje. Dále již stačí použít takto získané parametry vymodelované tváře jako hlavní vektor pro výpočet podobnosti mezi zkoumanou tvář a referenčními tvářemi z databáze.

6.5 Metody založené na Elastických shlukových grafech

Pokud se pokusíme zamyslet nad maximální různorodostí zkoumaného problému, dospějeme k názoru, že všechny modelované subjekty jsou shodné v topologii struktury obličejce. Wiskott prezentoval hlavní rozpoznávací metodu pro třídění členů známe třídy objektů. V těchto modelech je tvář reprezentována jako uzly, které jsou umístěny na bodech u kterých předpokládáme významný podíl pro nejlepší aproximaci obličejce. Vektory vzniklé propojením nejbližších vektorů na ploše obličejce jsou označovány jako distanční vektory.

6.5.1 Jety

Jet se snaží o vhodný popis malých kousků šedivých hodnot v obraze (obrázek 12b), kde se využívá principu Gaborovy vlnkové transformace, která je definována jako

$$\mathfrak{T}_j(\vec{X}) = \int \wp(\vec{x}) \psi_j(\vec{x} - \vec{X}) d^2 \vec{x} \quad (6.15)$$

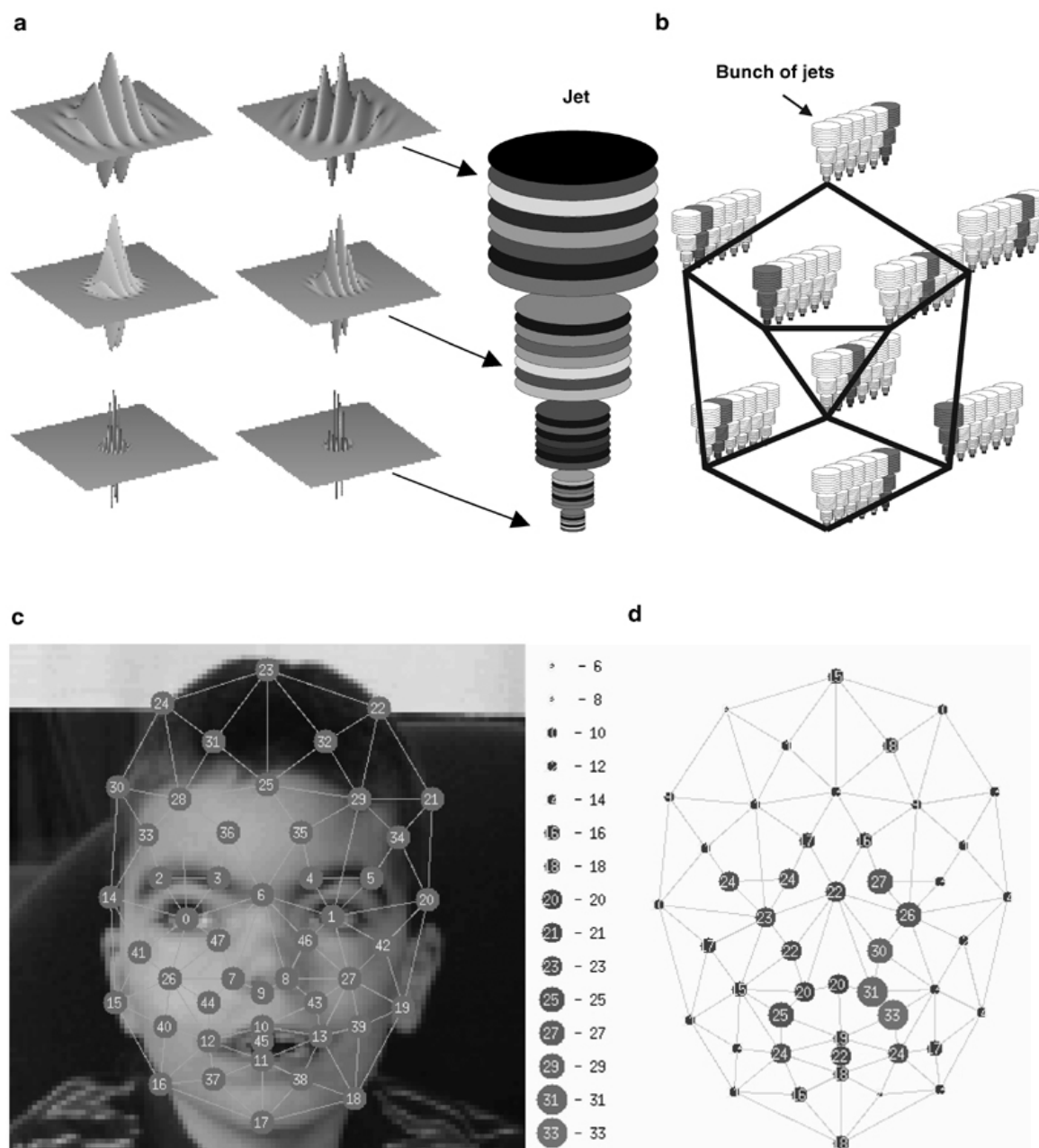
a Gaborovo jádro je určeno funkcí

$$\psi_j(x) = \frac{k_j^2}{\sigma^2} \exp(k_j^2 \frac{x^2}{2\sigma^2}) [\exp(i \vec{k}_j \vec{x}) - \exp(\frac{\sigma^2}{2})] \quad (6.16)$$

ve tvaru rovinných vln s vlnkovým vektorem k_j , pomocí Gaussovy obalovací funkce je

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos(\varphi_\mu) \\ k_v \sin(\varphi_\mu) \end{pmatrix}, k_v = 2^{\frac{-(v+2)}{2}} \pi, \varphi_\mu = \mu \frac{\pi}{8}. \quad (6.17)$$

Takto definované jety jsou základem pro vytvoření normalizované reprezentace obličejce použitím tzv. obličejového grafu, což není rovnoměrně rozložená mřížka (kdy může docházet k vysoké redundanci a nadbytečnosti v závislosti na počtu zvolených bodů), nýbrž v počtu 48 bodů definovaný shlukový graf (srovnávací shlukový graf). Tyto body jsou rozloženy v hranách a křivkách dominantních částí obličejce (obrázek 12c) [10].



Obrázek 12: (a) Gaborova vlnková transformace různých prostorových velikostí a orientací souměrnosti, která se používá k analýze digitální fotografie. Pro každý bod jsou výsledky velikosti a orientace uspořádány do charakteristického vektoru zvaného jet. (b) Schéma shlukového grafu. Tento graf zobrazuje 9 uzlů a 13 hran, každý uzel je označen jety, které jsou specifické pro 6 osob. Během porovnávání je v každém uzlu nezávisle vybrán jet s nejvyšší podobností, které identifikujeme v obraz obličeje transformovaného do odstínů šedi. (c) čelní pohled, tento graf vyznačuje 48 uzlů odkazujících se a pevné vyhodnocovací body. (d) každý uzel má specifikován počet příspěví k úspěšné identifikaci. Vnitřní obličejové uzly mají vyšší prioritu než vnější jsou stabilnější [10].

6.5.2 Porovnávání Jetů

Abychom mohli s jety efektivně pracovat, potřebujeme způsob, jakým vzájemně porovnávat parametry jetů mezi sebou. Díky fázovému posuvu jety získané z bodů umístěných jen pár pixelů od sebe mohou mít velmi různé koeficienty i když reprezentují téměř shodně lokální rysy, což může způsobit hned řadu problémů při jejich vzájemném porovnávání. Funkce podobnosti je definována, jako

$$S_a(J, \mathfrak{J}) = \sum_j a_j \frac{a_j}{\sqrt{\sum_j a_j^2 \sum_j a_j^2}} . \quad (6.18)$$

Je požadováno, aby konkrétní fázová informace rozlišovala mezi vyskytujícími se vzory s podobnými velikostmi. Mějme dva jety, které odkazují na stejnou lokalitu v objektu s malým relativním posuvem. Fázové posunutí obou by mohlo být v tomto případě přibližně vyvážené.

6.5.3 Individuální obličej

O různorodosti tváří je řečeno už mnoho, ať jsme mluvili o dlouhodobých a krátkodobých rozdílech, nebo možnostmi konkrétních snímacích senzorů a tím různě poškozené obrazy.

Definujeme tedy množinu kandidátních bodů, tedy ústní koutky, nosní dírky, uši apod. Pokusíme-li se sestavit graf reprezentující tvář sestávající z těchto výchozích bodů (Jetů), a spojíme mezi nimi. Počet těchto uzlů odpovídá množině uzlů a výchozí body odpovídají pozicím \mathbf{X}_n a hrany jsou označeny vzdálenostmi $\Delta \mathbf{X}_e = \mathbf{X}_n - \mathbf{X}_m$, kde hrana e spojuje uzel z prvního bodu n do bodu m .

Pokud odkazy na geometrickou strukturu grafu nebudou označeny jety, nazýváme jej mřížkou. Pokud se nyní zamyslíme nad podobou tváře, je to tedy množina bodů z tváře (výchozí body). Tyto body se liší v geometrii a umístění a ačkoliv výchozí body odkazují na odpovídající umístění v obraze, mohou být uzavřeny kvůli rotaci bodu. Aby tedy bylo efektivně možné srovnávat obrazce různé geometrie, musíme definovat ukazatele na odpovídající dvojici uzlů [55].

6.5.4 Shlukový graf obličej

Jelikož se na poli počítačů pohybujeme v časově omezených možnostech, je velice složitou úlohou pokrytí naprosto všech různorodostí variací lidské tváře. Takový systém by teoreticky musel mít databázi veškerých tváří celé planety a jelikož je tato množina pro reálné použití konečná, což je ve sporu s nově narozenými generacemi, které nejsou v databázi. Tento graf pokrývá potřeby širokého okruhu různých změn nálezu tváře, svou funkčností připomíná strukturu zásobníků jetů, což jsou jakési seřazené vhodné obrazce [55].

6.5.5 Reprezentace tváře

Nejpřesnější metodou je ruční vytvoření tváře, kterou jsme použili pro základní graf tváře. Nyní chceme vytvořit nový graf a v podstatě přidat nový obraz, což je možnost učít se nové markanty a nové možnosti. Z počátku, když máme k dispozici jen málo grafů k porovnávání a možnosti rozdílnosti jednotlivých zásobníků shluků, je nutné kontrolovat výsledky porovnání. Obsahuje-li již řádově desítky záznamů, již systém spolehlivě vytváří nové reprezentace tváří.

6.5.6 Podobnostní funkce

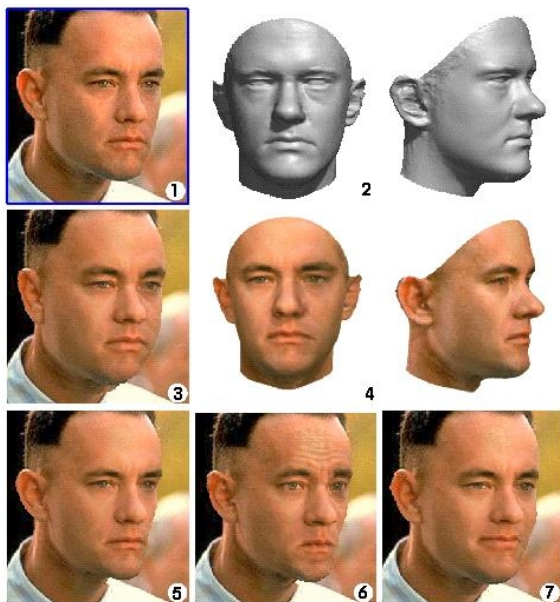
Mezi nutné funkce pro porovnávání obličejů je podobnostní funkce, která bude nějakým způsobem hodnotit podobnost mezi jednotlivými grafy. Rozdíly mezi jednotlivými jety souvisí s zakřivením mřížky grafu reprezentujícího tvář a pomocí rozdílů těchto grafů můžeme vyjádřit příbuznost mezi grafy. Pokud máme v databázi již uloženo potřebné množství obličejů, a jejich shlukových reprezentací, máme rozdíly vzdáleností mezi jednotlivými body struktury grafu uloženy a nejpoužívanější z nich se nazve porovnávacím. Tento nejlepší typ jetu souží jako expert daného shluku.

6.6 2D modifikační model (AAM)

Jedná se o statický model kombinace střídání odlišných tvarových vlastností s modelem změn tvaru obličeje. Statický model obsahuje model tvaru a definuje úroveň šedi pro vytvoření potenciálně nejzajímavějších dat pro rozpoznávací procedury. Odpovídající procedura se bude snažit nalézt minimální parametry modelu mezi rozdílem obrazu a umělým výchozím obrazem (Obrázek 12c, d) [55].

6.7 3D modifikační model

Nyní nás budou zajímat obrazy, které reprezentují lidský obličej v 3D prostoru a tedy v běžných podmínkách se kterými se setkáváme při pořizování fotografií uživatelů. V podstatě lze hlavní výhodu vystihnout v jednoznačně lepším postihnutí lidské tváře. Jako vždy budeme používat model s parametry specifikujícími konkrétní model a budeme mít vhodnou proceduru (funkci), pomocí níž budeme moci vyhledávat rozdíly mezi dvěma reprezentacemi 3D modelu tváře. K efektivnímu využití tohoto způsobu potřebujeme navíc parametry reprezentující tvar a vlastnosti textury, které jsou odděleny od obrazových dat.



Obrázek 13: Různé použití variačního modelu [55].

7 Identifikace (vyhledávání) subjektů

V této kapitole rozebereme automatizované hledání vazeb mezi testovaným subjektem a databází obličejů. Musíme tedy provést lokalizaci obličeje v obraze a následně nalézt specifické vlastnosti ve vyhledaných obličejích, ať už se jedná o základní antropometrické vlastnosti obličeje nebo vedlejší identifikační znaky. Promítneme tedy obličej do vysokorozměrného prostoru spolu s ostatními vektory obličejů z databáze a snažíme se najít co nejbližší a tedy nejpodobnější vektor. V ideálním případě nebo při použití identické fotky obličeje by měly testovaný a databázový vektor splývat, v opačném případě by měla být vzdálenost mezi koncovými body vektorů velice malá, a dále v ideálním případě by veškeré identifikační fotografie stejné osoby měly spadat do stále stejného podprostoru a tyto podprostory by měly být disjunktní. Potřebujeme tedy vhodně určit vzdálenost mezi koncovými body vektorů ve vysoko-rozměrném prostoru.

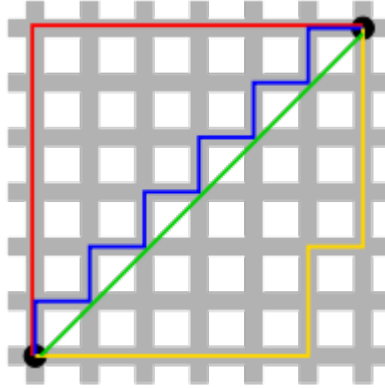
7.1 Měření vzdálenosti vektorů

Vzdálenost je numerický popis vzdálenosti dvou libovolných oddělených objektů. Ve fyzikálních oblastech nebo v každodenním hovoru se většinou odkazujeme na fyzickou vzdálenost dvou objektů, časových intervalů. V matematice chápeme vzdálenost podle více kritérií a chápeme mnohdy vzdálenost z bodu A do bodu B stejně jako vzdálenost z bodu B do bodu A. Analytická geometrie realizuje vzdálenost mezi body X a Y užitím vzdálenostního vzorce, tedy vzdálenost mezi (x_1, y_1) a (x_2, y_2) [23].

7.1.1 Manhattan vzdálenost

Tato vzdálenost bývá často označována jako taxikářská vzdálenost, protože je podobná algoritmu průjezdu kolem severoamerických městských bloků s kolmými ulicemi, od bodu A do bodu B. Jedná se tedy o vzdálenost mezi dvěma body v Euklidově prostoru s pevným kartézským souřadnicovým systémem, tedy součet délek projekce lineárních částí osy [26]. Jde tedy o jednoduchý výpočet ve tvaru

$$D_{Manhattan} = \sum_{i=1}^n |x_{iB} - x_{iA}| \quad [27]. \quad (7.9)$$



Obrázek 15: Červená, modrá a žlutá křivka mají stejnou délku v Euklidově i Manhattan geometrii, avšak v Euklidově můžeme nalézt ještě zelenou přímku, která je nejkratší cestou [26].

7.1.2 Euklidova vzdálenost

Z hlediska matematiky je Euklidova vzdálenost chápána jako ordinární vzdálenost mezi dvěma body, na kterou můžeme opakovaně aplikovat Pythagorův teorém. Pokud budeme tedy aplikovat tuto formuli vzdálenosti, Euklidův prostor se stává metrickým prostorem [22].

Euklidův prostor R_n je tedy n -rozměrný reálný vektorový prostor, který má definován skalární součin, stejně jako metrický prostor:

$$D_{euclid} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (7.2)$$

Euklidova vzdálenostní matice $\mathbb{R}^{N \times N}$ je obsáhlá tabulka vzdálenostních čtverců d_{ij} mezi body (po dvou) ze seznamu N bodů $\{x_l, l=1..N\} \in \mathbb{R}^n$. Čtvercová metrika míry vzdálenosti je definována jako [21]:

$$d_{ij} = \|x_i - x_j\|^2 = (x_i - x_j)^T (x_i - x_j) = \|x_i\|^2 + \|x_j\|^2 - 2x_i^T x_j = [x_i^T, x_j^T] \begin{bmatrix} I & -I \\ I & -I \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \vec{X}^T (\phi_{ij} \otimes I) \vec{X} = \langle \phi_{ij}, X^T, X \rangle \quad (7.3)$$

kde X je vektor a kde $\phi_{ij} \otimes \mathfrak{S}$ náležící do $I \in S^n$ v ii -tém a jj -tém bloku položek dokud $\mathfrak{S} \in S^n$ plní ij -tý a ji -tý blok

$$\phi_{ij} = \delta((\mathbf{e}_i \mathbf{e}_j^T + \mathbf{e}_j \mathbf{e}_i^T) \mathbf{1}) - (\mathbf{e}_i \mathbf{e}_j^T + \mathbf{e}_j \mathbf{e}_i^T) \in S^N = \mathbf{e}_i \mathbf{e}_i^T + \mathbf{e}_j \mathbf{e}_j^T - \mathbf{e}_i \mathbf{e}_j^T - \mathbf{e}_j \mathbf{e}_i^T = (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T, \quad (7.5)$$

kde $\{\mathbf{e}_i \in \mathbb{R}^N, i = 1..N\}$ je soubor základních vektorů a \otimes značí Kronekrův produkt, a každý vstup d_{ij} je konvexní čtvercová funkce vektoru X . Každý bod je označen řadovou číslovkou, což koresponduje s řádkovým nebo sloupcovým indexem EDM [21].

7.1.3 Minkowského vzdálenost

Výpočty Minkowského sumy nachází v oblasti informatiky své uplatnění například u plánování cesty robota, plnění kontejnerů a hledání offsetů [28].

$$D_{Minkowski} = \left(\sum_{i=1}^n (x_{iA} - x_{iB})^p \right)^{1/p} \quad [27] \quad (7.6)$$

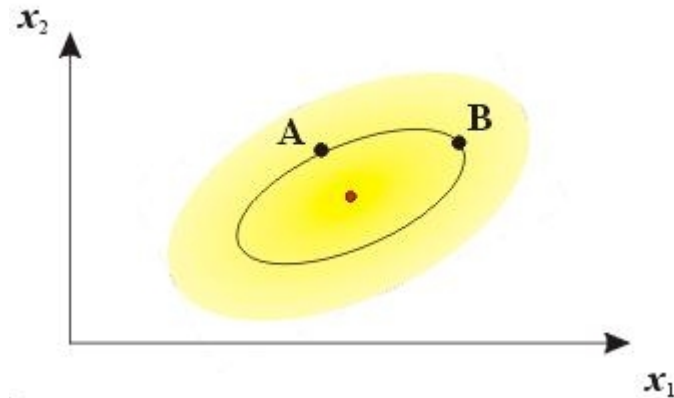
Jedná se tedy o zobecněný případ předchozích vzdáleností.

7.1.4 Mahalanobisova vzdálenost

Byla definována P.C. Mahalanobisem v roce 1936 a zaujímá významnou roli. Tato vzdálenost je měřena pozorováním okolí z výchozího bodu tímto způsobem [24]:

$$D_{Mahalanobis} = (\mathbf{x} - \mathbf{g})^T \Sigma^{-1} (\mathbf{x} - \mathbf{g}), \quad (7.7)$$

s kovariační maticí distribuce. Pozorovací regiony se stejnou hustotou jsou ve stejné Mahalanobisově vzdálenosti od výchozího bodu, ačkoliv Euklidovy vzdálenosti od výchozího bodu mohou být různé. Body v daných Mahalanobisových vzdálenostech leží na elipsoidu se středem ve výchozím bodě. Na obrázku můžeme vidět body A a B, které mají stejnou Mahalanobisovu vzdálenost od výchozího bodu [24].



Obrázek 14: Mahalanobisova vzdálenost dvou bodů [24].

7.1.5 Chebyshevova vzdálenost

Chebyshevova vzdálenost je metricky definovaná na vektorovém prostoru jako vzdálenost mezi dvěma vektory jako největší rozdíl podle určité dimenze. Je také známá jako šachovnicová vzdálenost, tedy minimální počet pohybů králem potřebný pro pohyb z nějakého čtverce šachovnice na nějaký další čtverec je úměrný vzdálenosti mezi těmito čtverci ve dvou dimenzích. Tedy Chebyshevova vzdálenost mezi vektory nebo body p a q se souřadnicemi p_i a q_i jako

$$D_{Chebyshev} = \max(|p_i - q_i|) = \lim_{k \rightarrow \infty} \left(\sum_{i=1}^n |p_i - q_i|^k \right)^{\frac{1}{k}} . \quad (7.8)$$

Ve dvourozměrném prostoru je definována jako $D_{Chess} = \max(|x_2 - x_1|, |y_2 - y_1|)$ [25].

7.2 Indexování

Pokud se podíváme na vývoj identifikačního software z hlediska nejprve objemu dat, budeme potřebovat udržovat a přistupovat teoreticky k více jak 6 miliardám obličejů v případě snahy o globální identifikační databázi. Vzniká nám tedy potřeba urychlit tento proces a indexovat obličej.

Indexování je snaha o přístup k datům přes index a navrácení relevantních dokumentů s určitými „falešnými poplachy“, kterých se ve druhé fázi verifikace snažíme zbavit. Data identifikujeme pomocí primárního klíče, což je jediný atribut bez možných duplicit, a sekundárními klíči, které se skládají z jednoho nebo více atributů s možnými duplicitami.

Pro dynamickou indexaci primárního klíče u relačních databází se využívá stromů a různých variant (B+, B*), kdežto pro indexaci sekundárních indexů používáme různá vektorová řešení organizace indexačního prostoru, abychom byli schopni odpovědět na různé variace dotazů.

Indexace dotazů je problematická díky své rozmanitosti. Rozlišujeme dotazy podle druhu výběrů na přesná srovnávání (například $\text{jmeno} = \text{“Novák“}$ a $\text{plat} = 30\,000$), nebo částečná (například $\text{jmeno} = \text{“Novák“}$ a $\text{plat} = *$). Dále můžeme rozlišovat rozsahové dotazy (například $\text{jmeno} = \text{“Novák“}$ a $20\,000 \leq \text{plat} \leq 30\,000$), nebo snaha nalezení nejbližších sousedů (například nalezení 10 nejvíce podobných obrazů). Jako indexovací strukturu využíváme invertovaných souborů, kde každý atribut ukazuje na seznam relevantních dokumentů. Rozdělují se podle způsobu přístupu na bodové (PAM), kde jsou data jsou body ve vysokodimenzionálním prostoru (Grid file, kd, kB, hB stromy) , a na prostorové metody (SAM), které data charakterizují linkami, a jinými geometrickými objekty (R-stromy a křivkami [31]).

7.2.1 Bodové přístupové metody (PAM)

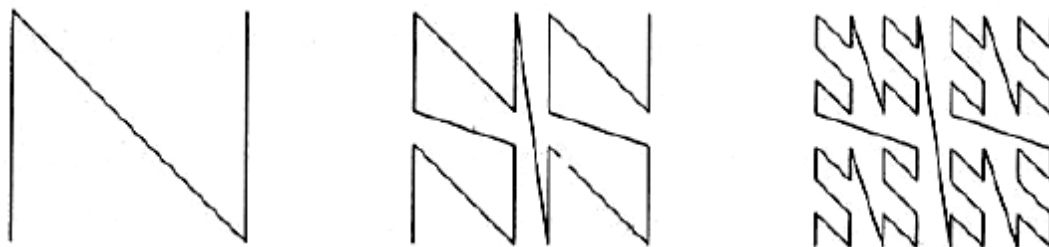
Jak jsme již zmínili, metody založené na chápání indexačního prostoru jako bodového systému, kde jsou data chápána jako N-dimenzionální prostor. Pro jejich uspořádání používáme Mřížkový přístup (grid file) a dále různé varianty B-stromu, který je v podstatě základní použitelnou stromovou strukturou. Níže rozebereme varianty typů kd-stromu, kB-stromu a hB-stromu.

7.2.2 Prostorové přístupové metody (SAM)

Struktura souboru jako body, linky a jiné geometrické objekty ve vysokodimenzionálním prostoru, ke kterému můžeme přistupovat jako ke křivkám vyplňující prostor, pro které se používají metody Z-postup, Gray nebo Hilbertovy křivky. Pro druhý přístup stromových struktur je využíván R-strom s jeho variantami. Dále je můžeme dělit podle typu dotazů na bodové dotazy, které se v podstatě snaží najít nejbližší oblasti obsahující vyhledávaný atribut, nebo metoda oken, která hledá křížící se geometrické objekty.

7.2.2.1 Z-postup

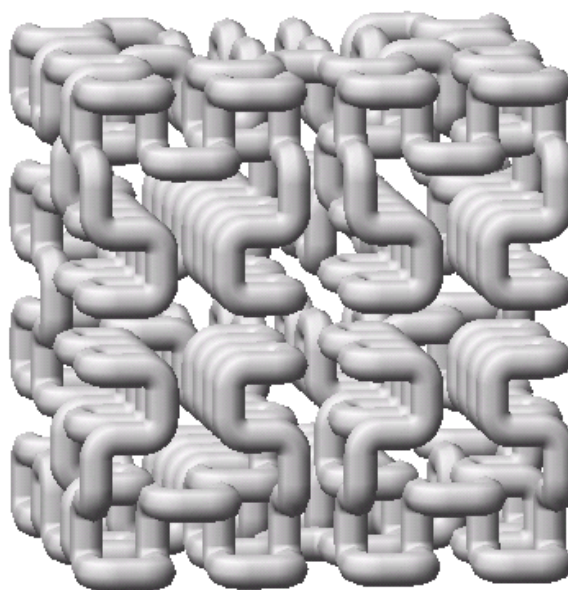
Název metody je odvozen od postupu po objektech podle tvaru písmene Z. Této metody se využívá například při rozlišování viditelných oken na obrazovce monitoru, kde jsou jednotlivá okna chápána jako plátna na různých souřadnicích osy Z (tedy v různé hloubce).



Obrázek 16: Ukázka Z-průchodu prostorem [31].

7.2.2.2 Hilbertův přístup

Hilbertova křivka je kontinuální fraktál vyplňující prostor, která byla poprvé popsána německým matematikem Davidem Hilbertem v roce 1891. Euklidova délka křivky je $2^n - \frac{1}{2^n}$, a narůstá exponenciálně [40]. Má lepší shlukovací schopnost než Z-průchod s efektivnějším trajektorií pohybu prostorem [31].



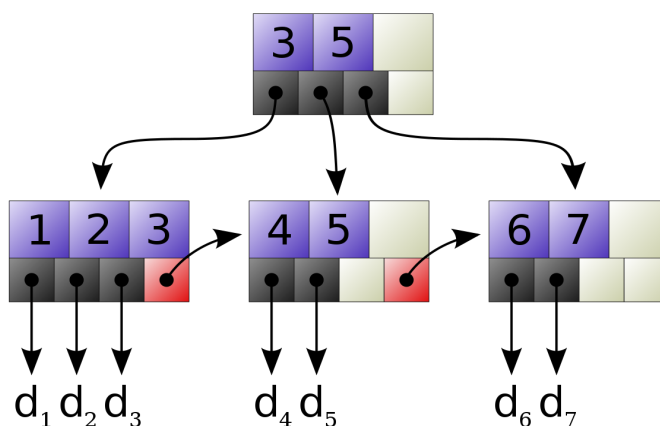
Obrázek 17: Hilbertova křivka ve třetí dimenzi [40].

7.2.3 B-strom

Strom je acyklický graf s jedním kořenem a vrcholy, které mají vždy jednu vstupní hranu grafu [39]. B-strom je druh stromu, který zavádí limity na maximální a minimální počet potomků vrcholu. Díky těmto základním omezením zůstává strom vyvážený a operace přidání, vyjmutí a vyhledávání tedy probíhají v logaritmickém čase. B-stromem stupně n rozumíme tedy stromovou strukturu, která splňuje [39]:

- Kořen má nejméně dva potomky, pokud není koncovým listem
- Uzly mají nejméně $\lceil n/2 \rceil - 1$ a nejvíce n potomků
- Všechny cesty od kořene k listům jsou stejně dlouhé
- Data v uzlu jsou logicky organizována do posloupností

Existuje celá řada variací tohoto stromu. Hojně používanou je B+strom, který má data uložena až v listových (koncových) uzlech grafu. U souborových systémů Reiser4, HFS(+), se využívá B*strom, který má specifikovanou minimální hranici počtu prvku v listu specifikovanou na $2/3n$.



Obrázek 18: Ukázka B+ stromu (variace na B-strom, v které musí být všechny hodnoty umístěny až v listech) [39].

7.2.4 Mřížkový (grid) soubor

Principem je položení sítě na adresový prostor, která se dynamicky přizpůsobuje lokální hustotě dat s vyšším dělením podprostoru. Dva nebo více prvků může sdílet jeden podprostor a řezy jsou povolené na předdeklarovaných místech kolmá vždy na jednu osu dimenze [31].

7.2.5 KD-strom

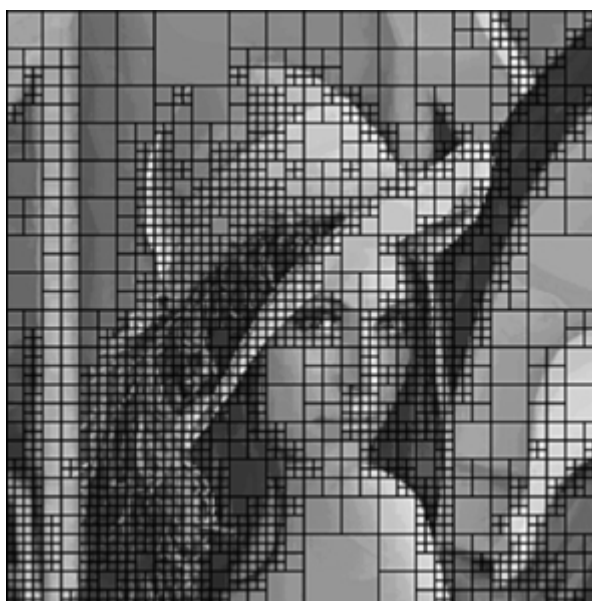
Dělení probíhá vždy podle roviny kolmé na jednu ze souřadnicových os, přičemž se vybírá vždy taková rovina, která dělí nejdelší hranu pomyslného kvádrů. Jde o binární soubor, kde vždy levá část podstromu obsahuje menší hodnoty konkrétního atributu, než pravá část část.

Jde o elegantní a intuitivní algoritmus s dobrým výkonem, vhodné pro dotazy extrakcí, rozsahové a nejbližších sousedů.

Existují různá rozšíření, jako KDB-strom, který dělí adresní prostor do intervalů pro každý uzel, nebo hB-strom, který dělí adresní prostor do regionů, které mají díry [31].

7.2.6 Quad-strom

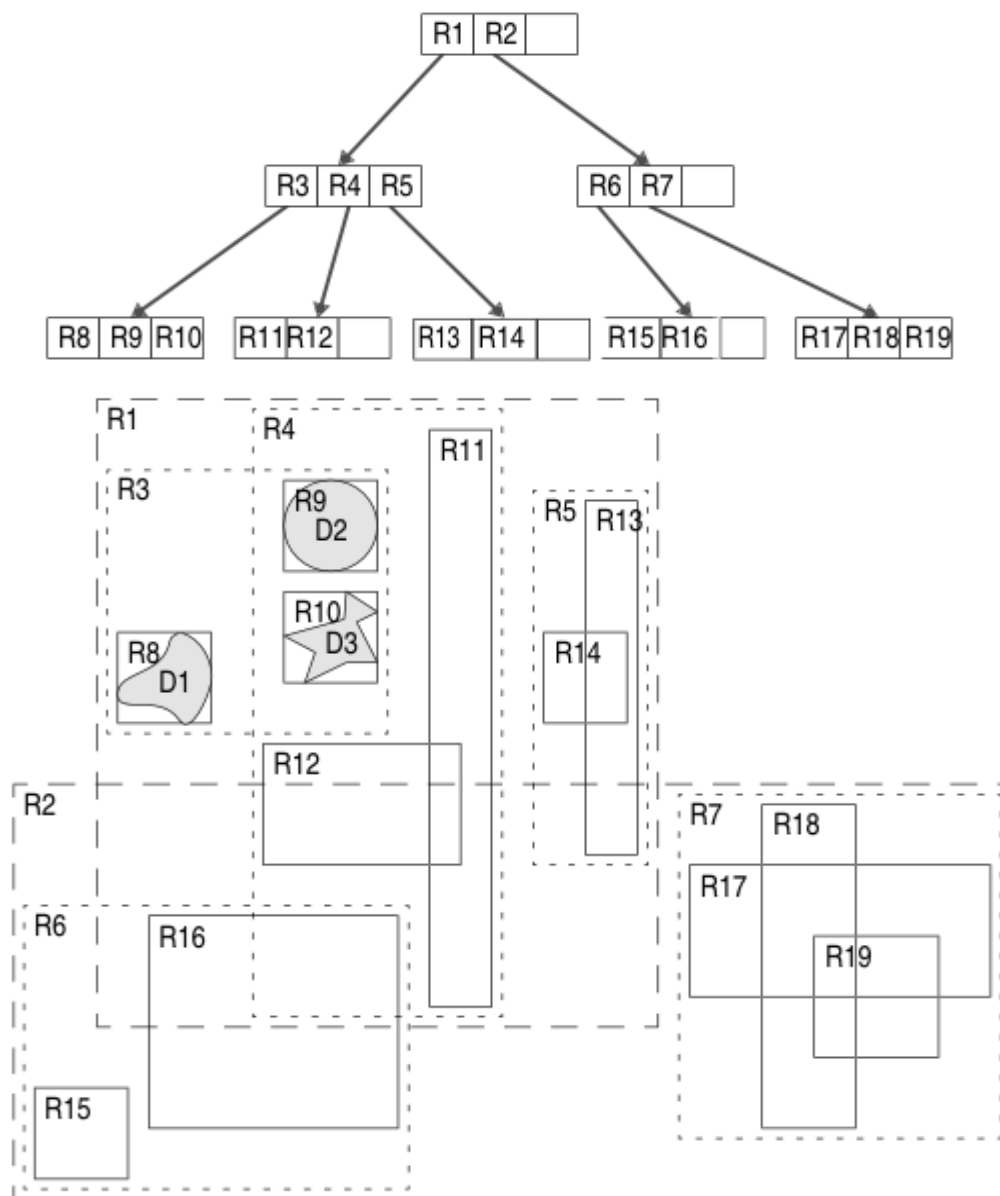
Quad stromy principiálně vycházejí z KD-stromů, ale vkládání body rozdělují prostor ve všech směrech. Quad stromy mají rychlejší vyhledávání, ale operace mazání jsou problémové, kvůli nalezení uzlu, který má být nahrazen. Další variací jsou MX-quad stromy, který navíc dohlíží, aby vzniklé části byly stejně velké a dělení probíhá rekurzivně. Je zde pevně určena granularita [31, 41].



Obrázek 19: Příklad použití metody Quad-stromu a vzorových datech [42].

7.2.7 R-strom

Stromy založeny na ukládání obdélníkových struktur, kde některé představují reálné objekty databáze, a jiné sdružují menší obdélníky do větších celků. Pokud v databázi existuje neobdélníkový objekt hledá se jeho minimální obdélník, nebo-li konvexní obálka obsahující všechny objekty. R-stromy jsou vhodným algoritmem pro velké objemy dat. Zásadním problémem metody je možnost překrývání obálek, tudíž při vyhledávání nelze zcela vyloučit některé větve. Tento problém se snaží řešit R*-stromy, které se snaží překrývání minimalizovat [31, 41].



Obrázek 20: Indexování pomocí metody R-stromu [43].

7.2.8 SS-strom

Další z variant R^* stromu, která namísto obdélníkových geometrických útvarů využívá kruhové, v případě třetího rozměru se pak jedná o kouli. Dosahuje kvalitních výsledků při hledání nejbližších sousedících bodů [31]. Tento přístup je výpočetně složitější než pravouhlé řešení obálek. Metoda je často využívána v kombinaci s dalšími stromy [46].

7.2.9 SR-strom

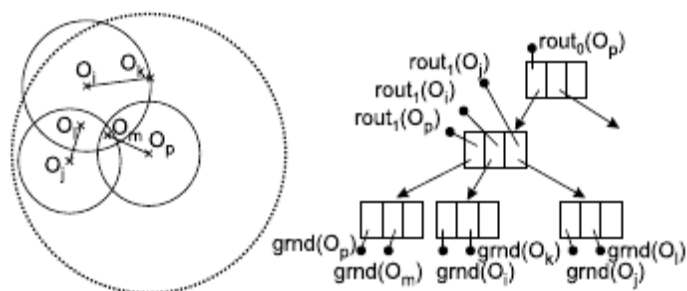
Jedná se o rozšíření R^* -stromu a SS-stromu. Příznačné je pro tuto funkci kombinované využití hranic koule a trojúhelníků, což zlepšuje výkon při dotazech na nejbližší sousedy. Tato metoda překonává obě rodičovská zmíněná řešení, obzvláště ve vysokodimenzionálních prostorech s nerovnoměrně rozloženými daty, což je u multimediálních dat velice pravděpodobné [44, 45].

7.2.10 VP-strom

Dělí prostor podle vzdáleností od výhodně postaveného výchozího bodu. Kořenem stromu je celý prostor, tedy veškeré databázové objekty. Levý podstrom je tvořen body s menší vzdáleností a pravý podstrom je tvořen body s větší vzdáleností. Jde tedy o statický binární strom s logaritmickým časem vyhledávání, dobrý pro využití pro operační paměti [31].

7.2.11 M-strom

Jedná se o metrickou přístupovou metodu v kombinaci s prostorovým přístupem k adresovému prostoru [31]. Vyvážený strom, který umožňuje indexovat pomocí objektů metrického prostoru. Podobně jako u mnoha jiných struktur je založena na B+stromu, kde ale konkrétní index představuje hierarchii metrických regionů (každý uzel je chápán jako jeden region). Listy stromu obsahují záznamy indexovaných objektů a vnitřní uzly obsahují tzv. směrovací záznamy, které popisují metrické regiony, vymezující v metrickém prostoru určitou oblast a popisujeme jej hyperkoulí [47].



Obrázek 21: Hierarchie regionů v metrickém prostoru a příslušný M-strom [47].

7.3 Hashování

Mluvíme-li o hašování, mámě na mysli hashovací funkce, což je reprodukovatelná metoda pro převod vstupních dat na relativně malé číslo. Tento vstupní vzorek dat můžeme charakterizovat a chápat jej jako otisk a označujeme jej jako extrakci či minimalizaci. Hashování je nedílnou součástí moderních kryptografických systémů které běžně využíváme pro digitální podpisy. Tuto funkci tedy můžeme použít například pro kontrolu integrity dat, rychlá porovnání dvou objemných informací, indexování a v neposlední řadě pro vyhledávání.

Důvodů proč se využívá hashování je hned několik. Pro jakkoliv dlouhý vstupní řetězec vždy poskytuje výstupní otisk konstantní délky, a navíc i při malé změně vstupního vzorku dat dojde k výrazné změně výstupního řetězce, tudíž je vysoká pravděpodobnost shody vstupních dat se stejným hashem. Z toho vyplývá řada výhod a možností pro srovnávání podle různých kritérií, ale i možnost kolize dvojic vstupních dat takových, že pro

$$x \neq y, hash(x) = hash(y) \quad , \quad (7.9)$$

tedy že pro různá vstupní data získáme stejný otisk. Tuto chybu nemůžeme z principu zcela odstranit, ale můžeme se snažit vytvořit takovou hashovací funkci, která nám s co nejvyšší pravděpodobností nedá stejné hashe. Pokud ovšem předem známe všechny platné vstupní vzorky, můžeme vytvořit dokonalou hashovací funkci, která bude tedy předem připravena na relevantní vstupní data a vyloučí se tak problematika případných kolizí [30].

Při indexování budeme často potřebovat strukturalizovaně ukládat hashovací klíče s odpovídajícími vstupními daty a dalšími vlastnostmi. Použijeme tedy hashovací tabulku pro asociaci

řetězce a vygenerovaného hashovacího klíče. Této možnosti se využívá především pro rychlé vyhledávání položky v homogenních datových typech [29].

7.3.1 Metoda dělení

Základní a nejjednodušší ze všech metod pro hašování celého čísla x , které dělíme celočíselně číslem M a použijeme zbytek po dělení jako výsledek hashovací funkce

$$H_{Division} = x \bmod M \quad [32]. \quad (7.10)$$

Obecně se jedná o docela dobrý přístup. Hodnotu M je potřeba explicitně určit. Většinou jako sudé číslo, což ale znamená, že $H(x)$ je sudé, když x je sudé, a $H(x)$ je liché, když x je liché. Pokud budou všechny pravděpodobné hodnoty, pak to není problém. V opačném případě nebudou klíče rovnoměrně rozprostřeny po celém rozsahu. Podobně tomu tedy bude, pokud necháme M rovno 2^k , pro nějaké číslo $k > 1$. Hashovací funkce extrahuje dolních k bitů z dvojkové reprezentace čísla x , což je výpočetně jednoduché, ale tento způsob není příliš vhodný, protože nezávisí na všech bitech dvojkové reprezentace x . Proto tedy budeme volit M jako prvočíslo, což zvýší pravděpodobnost rovnoměrného rozložení po celém rozsahu, protože dělení závisí na všech bitech čísla x , ne jen na postfixu jeho binární podoby pro libovolné malé k .

Výhodou metody dělení je absence nutnosti nastavení v době kompilace jako stálé konstanty, ale může být určeno až za běhu programu. Avšak nevýhodou plynoucí z konstrukce metody je vlastnost generování za sebou jdoucích mapování klíčů do po sobě jdoucích hodnot, což znamená, že po sobě jdoucí klíče nebudou kolidovat, ale po sobě jdoucí alokace polí budou obsazená. Je zřejmé, že to může vést k degradaci výkonu systému. Měly bychom tedy vědět řádově kolik klíčů budeme potřebovat použít [32].

7.3.2 Metoda středních čtverců

Tato metoda vychází myšlenkově z metody dělení a z faktu, že násobení je obecně rychlejší než dělení a tedy tím můžeme zkrátit dobu běhu hashovacího algoritmu. Mějme tedy M rovno 2^k , pro nějaké číslo $k > 1$. Pro hashovací funkci použijeme

$$H_{Square}(x) = \left[\frac{M}{W} (x \bmod M) \right] \quad [33], \quad (7.11)$$

kde M i W jsou mocniny dvou a koeficient $W/M = 2^{w-k}$ je také mocninou dvou. Tedy abychom zlepšily metodu dělení, uděláme ještě posun o $w-k$ bitů. Tato metoda má vlastnost dobré funkce, pokud celočíselné klíče jsou stejně pravděpodobné. Po sobě jdoucí klíče dobře rozptýlí. Tato metoda tedy bere v úvahu pouze určitou podmnožinu bitů uprostřed klíčů, což může vést ke konfliktu počátečních klíčů prefixem nul [33].

7.3.3 Metoda násobení

Velmi jednoduchá obměna metody středních čtverců, kterou zmírníme její chyby v extrémních případech. Násobíme klíč vybranou konstantou a pak vybereme střední bity z výsledku.

$$H_{\text{Multiplication}}(x) = \left[\frac{M}{W} ((a \cdot x) \bmod M) \right] \cdot \quad (7.12)$$

Jestliže zvolíme a , existuje pak i inverze a' , takže pokud chceme získat zpět originální hodnotu, můžeme jen vynásobit výsledek znovu touto inverzí. Existuje mnoho konstant, které přicházejí v úvahu, avšak jedna hodnota užívaná pro 32-bitová čísla je $a_{10} = 2\ 654\ 435\ 760$ s binární reprezentací $a_2 = 10011110001101110111100110111001$ [34].

7.3.4 Metoda rozkladu

Tato metoda dělí x do několika oddílů tak, že každá část má stejný počet číslic jako požadovaný hashovací klíč [35]. Matematická formulace metody je ve tvaru

$$H_{\text{Folding}}(x) = (a + b + c) \bmod M \quad [36], \quad (7.13)$$

kde číselné hodnoty a, b, c jsou části rozkladu vstupního klíče a M je velikost hashovacího prostoru. Výsledek po celočíselném dělení je hashovacím klíčem.

7.3.5 Fibonacciho hashování

Za finální variaci hashování považujeme Fibonacciho hashovací funkci, která se zakládá na metodě násobení se speciálním případem hodnoty pro a , které budeme vybírat okolo stejného čísla jako v předchozí kapitole. Mějme tedy pozitivně definovaná x a y , a poměr $\Phi = x/y$ a dále [35]

$$\begin{aligned} \frac{x}{y} = \frac{x+y}{x} &\Rightarrow 0 = x^2 - xy - y^2 \\ &\Rightarrow 0 = \Phi^2 - \Phi - 1 \quad , \\ &\Rightarrow \Phi = \frac{1 + \sqrt{5}}{2} \end{aligned} \quad (7.14)$$

kde Φ má úzkou vazbu mezi nejlepší konstantou a Fibonacciho čísly, která získáme ze vztahu

$$Fn = \frac{1}{\sqrt{5}}(\Phi^n - \hat{\Phi}^n) \quad . \quad (7.15)$$

Zde je $\Phi = \frac{(1 + \sqrt{5})}{2}$ a $\hat{\Phi} = \frac{(1 - \sqrt{5})}{2}$. V kontextu Fibonacciho hashování je pro nás

zajímavá Φ^{-1} , což můžeme vyjádřit následujícím způsobem:

$$\Phi^{-1} = \frac{2}{1 + \sqrt{5}} = \left(\frac{2}{1 + \sqrt{5}}\right) \left(\frac{\sqrt{5} - 1}{\sqrt{5} - 1}\right) = \frac{\sqrt{5} - 1}{2} \approx 0.618033887 \quad (7.16)$$

Fibonacciho hashování je v podstatě metoda násobení, ve kterém vyměníme konstantu a za vhodně zvolené celé číslo odvozené od W . Následující hodnoty jsou vhodné pro odpovídající W a blíží se k $\Phi^{-1}W$.

W	a
2^{16}	40503
2^{32}	2654435769
2^{64}	11400714819323198485

Tabulka 2: Příklady konstanty a pro různé W [35].

Pokud se tedy zamyslíme nad Fibonacciho hashováním, jde tedy opět o metodu násobení, která vhodně rozprostírá po sobě jdoucí klíče. Pokud tedy použijeme způsob výpočtu $a = \Phi^{-1}W$ pro hashování posloupnosti klíčů, padne hodnota hashe mezi dva již existující dříve vytvořené hashe [35].

7.3.6 Cormackovo hashování

Tato statická metoda je dnes hojně využívána v celé řadě aplikací. Je založena na existenci hlavní hashovací funkce a celé třídy vedlejších hashovacích funkcí. Metoda u každého hashovacího klíče ukládá odkaz do primárního souboru a kolikátá vedlejší hashovací funkce byla použita a dále počet položek, na které odkazujeme. Při vytváření dalších klíčů nejprve vypočteme hlavní funkci

hashovací klíč a pokud již existuje nějaká položka s tímto klíčem, snažíme se najít takovou vedlejší hashovací funkci, která nám vrátí odlišný hashovací klíč pro nově vkládanou položku a pro všechny ostatní položky v datovém bloku odkazů do primárního souboru.

Při vyhledávání začínáme analogicky vypočtením hodnoty hlavní hashovací funkce a sledujeme počet odpovídajících položek. Odpovídá-li více položek danému klíči spočteme vedlejší hashovací funkci daného bloku a takto najdeme odpovídající klíč primárního souboru. Často používanou sekundární hashovací funkcí je [37]

$$H_i(x, r) = (k \bmod (2i + 100r + 1)) \bmod r, \quad (7.17)$$

kde x je vstupní klíč, i je číslo použité hashovací funkce a r je počet již existujících položek stejného hashe.

8 MPEG-7

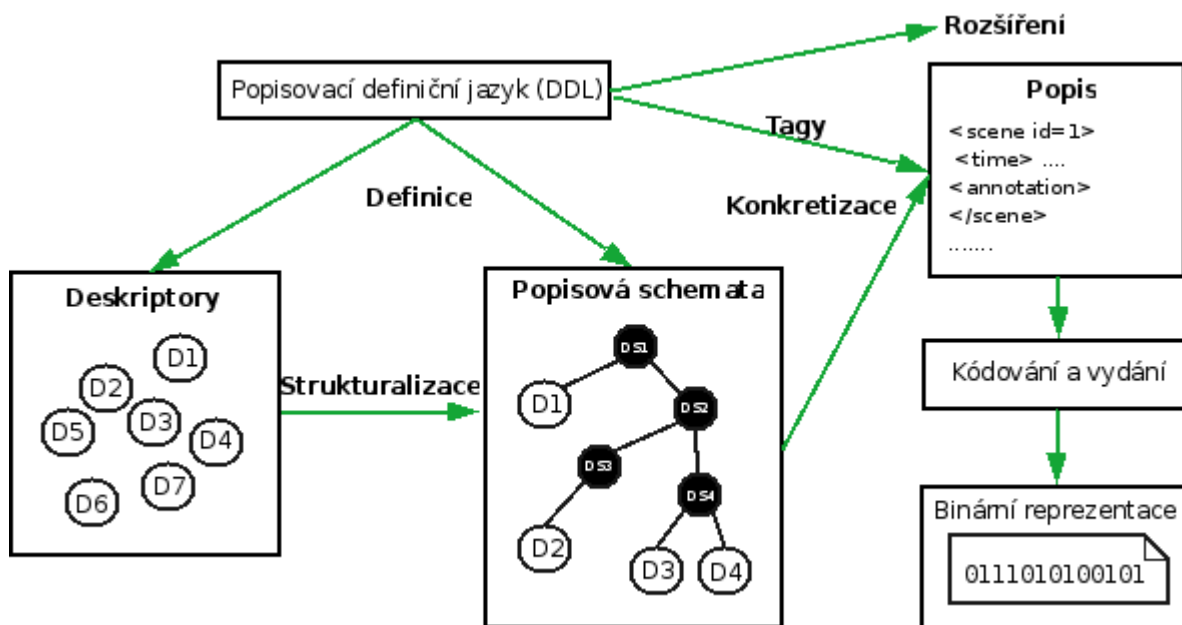
Tento formát byl přijat jako standard ISO/IEC, který vyvinulo Moving Picture Experts Group, která má na svědomí standardy jako MPEG-1 (1992) a MPEG-2 (1994), které jsou nám chronicky známé jako standardy používané pro VideoCD, MP3, DVD.

Jejich dalším počinem byl MPEG-4 (1998 a 1999), který je kombinací přirozeného a syntetického obrazu jako objekty. Je to standard funkcí umožňující integraci produkce a distribuci. Standard MPEG-7 se od svých předchůdců odlišuje především v jeho použití, které jsou určeny pro kódování multimédií. Standard MPEG-7 je souhrn standardizovaných prostředků pro popis obsahu multimediálních objektů, jako jsou statické obrazy v tištěné podobě nebo 3D grafika a její modely, zvuk řeč a video a pro náš účel zajímavá biometrie lidského obličeje [41].

Jde tedy o ucelený způsob reprezentace informací popisující název, autora nebo vztahy mezi objekty v obraze. Definuje tedy sadu informací ve struktuře s dalšími vztahy (deskriptory) mezi elementy, které ukládá ve formátu XML nebo do efektivnější binární podoby. Popis nesouvisí se způsobem uložení média, je tedy možné přiřadit deskriptor k obrazu, článku, filmu stejně jako streamu.

Deskriptory (Ds) reprezentují různé vlastnosti (rasy) a atributy multimediálního obsahu založené na katalozích (název, autor), sémantice (informace o objektech a událostech), syntaxi (barva obrazu nebo tón zvuku) a technologii (formát, velikost, vzorkovací frekvence). Jazykem pro definici deskriptorů definuje deskriptory, odvozené datové struktury (DSs) a typy. Definuje syntaxi a sémantiku s možností jejich změny a rozšíření založených na XML upraveném pomocí W3C pro MPEG-7. Pro podporu tvorby a přenosu popisů v binární podobě máme k dispozici systémové nástroje pro multiplexování s multimediálním obsahem, synchronizaci a formáty souborů [41].

Tento mohutný nástroj můžeme použít pro filtrování dat. Můžeme například z několika not poznat skladbu, z několika tahů štětcem poznat obraz. Hledáme podle podobností objektů a textur nebo v případě videa pohyb objektů v obraze specifickým způsobem [48].



Obrázek 21: Schéma prvků MPEG-7 [48].

Normu MPEG-7 můžeme rozdělit do tématických částí:

1. **Systém** – Specifikuje nástroje pro přípravu deskriptorů k přenosu, ukládání kompresi a pro synchronizaci s obsahem.
2. **DDL** – specifikace jazyka pro definici množiny deskriptivních a nových nástrojů, založených na XML.
3. **Vizuální** – obsahuje nástroje pro popis vizuální složky, specifikuje základní kategorie, jako jsou barva, textura, tvar, pozice a rozpoznávání obličejů.
4. **Zvuk** – definuje základní nástroje pro popis zvukových složek jako spektrální, časové, dynamické vlastnosti nebo rozpoznávání hlasu, barva nástroje a melodie.
5. **Multimediální** – Jde o schémata popisující multimediální obsah, tedy audio i video, použití obsahu, organizace, navigace a interaktivitu.
6. **Referenční software** – implementace standardu se stále vyvíjí, protože norma popisuje způsob uložení popisu, nikoliv způsob jeho získání.
7. **Testování ekvivalence** – poskytuje vodítka pro testování shody implementace deskriptorů.
8. **Extrakce a použití schémat** – pouze informativní charakter, není součástí normy, například implementace experimentálního modelu [41, 48].

8.1 Extrakce rysů a popis obsahu

Pokud se zamyslíme nad možnostmi realizace popisu media, můžeme jej provádět ručně, tedy vytvoření smysluplného objektivního názvu a popisu objektu, popřípadě i ruční přiřazení média do jisté kategorie, nebo automatizovaný popis vytvořením vektorů rysů (signatury a deskriptory).

Rysy jsou jisté charakteristické vlastnosti obsahu médií. Rysy u obličeje jsou například barva kůže, vlasů, očí, antropometrické vlastnosti, umístění nosu, tvar hlavy. Podobně realizujeme rysy i u ostatních obrázků. Rysy u videa mohou být charakteristické pohybem objektů. Rysy textu podle klíčových slov nebo zvukové rysy jsou frekvenční vlastnosti, tón, barva nástroje, hlas a rozpoznávání řeči [41].

8.1.1 Extrakce rysů

Extrakce nízkourovňových rysů obrazových dat jsou relativně jednoduché. Jde tedy například o dominantní barvy v obrázku nebo její histogram. Pro některé rysy nejsou klasické barevné modely příliš vhodné. Je tedy vhodnější použít model, který více odpovídá lidskému chápání barev, jakým je YCbCr. Texturové vlastnosti reálných povrchů, 2D vizuální vlastnosti, jako jsou kontrast, zaměřenost, pravidelnost, nebo hmatový charakter, jako hrubost nebo drsnost. V její počítačové reprezentaci je složená z opakujících se elementů, které se nazývají primitiva. Tato primitiva mohou být popsána jejich statickou, geometrickou nebo frekvenční charakteristikou.

Pro tvorbu deskriptorů můžeme také využívat vlnkové transformace nebo podobových rysů. Mají prostorový frekvenční charakter. Jejich výsledků je možné použít pro statické zpracování a při vhodné kombinaci tvarů, tedy rysů (Haarovy kaskády), a nalézt tak požadovaný objekt v obraze.

Těchto vlastností využijeme například při optické kontrole výrobní linky laků automobilů. Barva a textura jsou vhodné pro podobnostní vyhledávání objektů. Libovolný obrázek tedy představuje mozaiku barev, proto je velmi efektivní indexovat oblasti s danou texturou nebo barvou [41].

8.1.2 Extrakce tvaru

Hlavním problémem vizuálních informací je jejich přílišná rozmanitost a stejně tak tomu je u lidských tváří, které jsou obecně shodné (oči, ústa, nos), ale liší se vzájemnou polohou a tvarem. Vnímání objektů ve scéně je odlišné u člověka a u počítače, tedy jako oblasti s jistou barvou, texturou, hranami a dalšími vlastnostmi. Pokud nebudeme přemýšlet nad stupněm vhodné granularity přesnosti, je možné říci, že objekt se skládá z jednoho nebo více regionů v obraze. Oblasti je možné popsat jako plochu, nebo její hranici. V případě plochy bývá oblast reprezentována jako bitmapa, ve které hodnoty jasu „1“ značí příslušnost odpovídajících oblastí, zatímco „0“ směřuje k bílému pozadí. Tento způsob popisu oblastí je efektivní při vyhledávání v případě porušení hranic.

Reprezentace hranic je náročnější, ale lépe odpovídá lidskému vnímání a je více odolná vůči transformacím, nehledě na změnu tvaru objektu v čase [41].

8.1.3 Extrakce pohybu a umístění

Pohybem chápeme změnu umístění objektu v posloupnosti snímků, včetně dalších transformací objektu. Musíme počítat s přiblížením, zmenšením, otočením, perspektivní deformace, nebo i s pohybem kamery. Kamera má šest stupňů volnosti, může se pohybovat po svislé nebo vodorovné ose, přibližovat nebo vzdalovat se a v každé z těchto os může navíc ještě rotovat. Z těchto důvodů pohyb kamery není vždy jednoduché přesně rozpoznat, ale je to žádaná vlastnost. Pohyb kamery můžeme vyjádřit pomocí fundamentální matice, kterou odhadneme díky shodě několika náhodných bodů v pokračujících snímcích.

Když se nám podaří kompenzovat pohyb kamery odečtením vektoru pohybu a ohraničíme pohybující se oblasti, jsme schopni identifikovat pohyblivé objekty a určit jejich tvar. Takovýto objekt následně obalíme konvexní obálkou a zapíšeme souřadnice do databáze. Pokud navíc ještě doplníme časový údaj, dostaneme časoprostorový lokátor objektu.

Zobecněním chápání pohybu objektů na ploše vytvoříme perspektivní matici. Pokud budeme uvažovat nejjednodušší pohyb objektu v rovině, kdy připustíme pouze transformaci a rotaci objektu, těleso nemění příliš vzhled, tvar ani velikost. Takovéto objekty můžeme ve videu najít pomocí odhadu pohybu, které realizují zjištění změny odečtením vždy dvou po sobě jdoucích snímků. Z této

extrakce pak vytvoří vektor pohybu. Lepší výsledky dosahují prediktivní metody, například Kalmanův filtr [41].

8.2 Komponentový LDA deskriptor tváře

Pokusíme se využít lineárních vlastností a robustnosti pro obrazové variace spojením LDA s reprezentací založené na komponentovém přístupu. Pokusíme se aplikovat LDA na oddělené části obličeje a tím tedy zlepšit přesnost LDA metody. V této souvislosti rozlišujeme holistické LDA metody, které aplikujeme na celou tvář, a komponentové LDA metody, které jsou tedy aplikované na části (komponenty) obličeje. Oba tyto přístupy je možné spojit. V takovém případě mluvíme o kombinovaném LDA přístupu. Z pohledu matematiky pro holistický přístup je obraz tváře reprezentován pomocí LDA hlavním vektorem (6.10) s LDA transformační maticí W_{LDA} (6.5) [49].

Příklad deskriptoru tváře ve formátu XML:

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns:xsi = "http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:mpeg7 = "http://www.mpeg7.org/2001/MPEG-7\_Schema">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "FaceRecognitionType">
<Feature>14 21 9 11 8 24 9 18 7 24 16 9 22 18 17 16 13 10 12 7
20 20 8 22 27 3 11 10 19 14 12 20 15 3 17 16 17 10 15 11 20
12 26 7 18 6 23 1 </Feature>
</Descriptor>
</DescriptionUnit>
</Mpeg7>
```

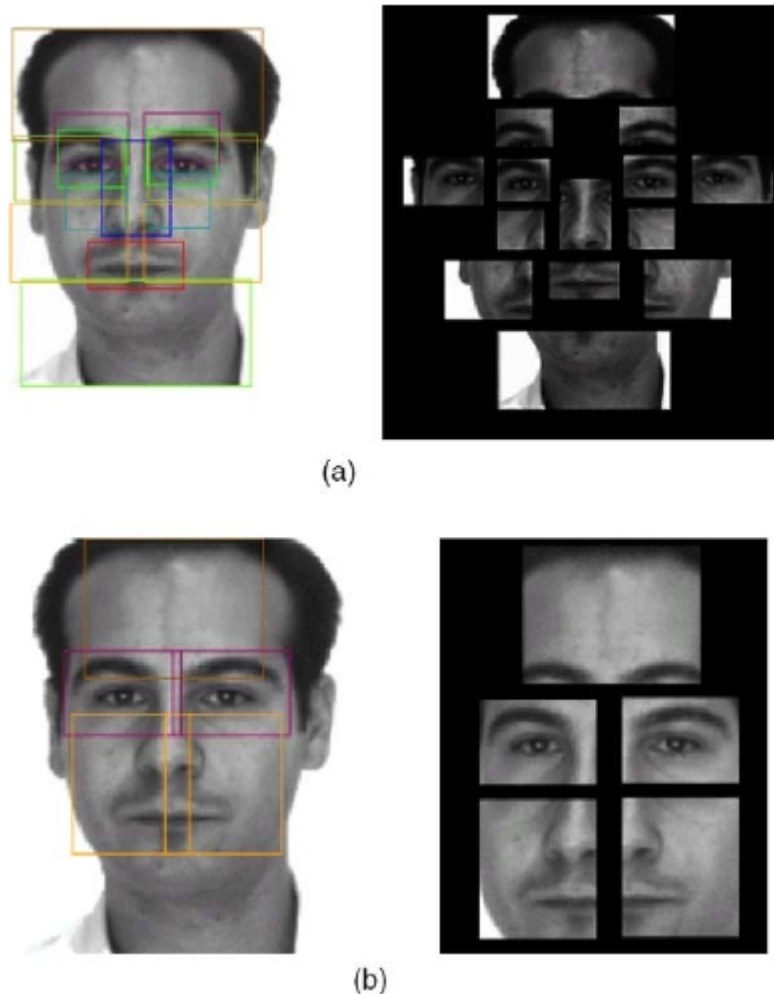
8.2.1 Popis tváře

Nejprve pro skupinu trénovacích dat provedeme extrakci LDA transformační matice. Tedy vezmeme soubor trénovacích obrazů $\{x_1, x_2, \dots, x_N\}$, které rozdělíme na L obličejových částí pomocí

rozdělovacího komponentního algoritmu. Všechny části jsou spojené dohromady a reprezentované ve vektorové formě $\{z^k_1, z^k_2, \dots, z^k_N\}$. Dále pro soubor každého komponentu vytvoříme LDA transformační matici, tudíž v podstatě dostáváme pro každý z rysů vlastní matici W^k . Nakonec vytvoříme vektor transformačních matic $\{W^1, \dots, W^L\}$. Na skupině dat je L vektorů $\{z^1, z^2, \dots, z^L\}$, které odpovídají konkrétním polím obličeje obrazu x . Množina hlavních transformačních vektorů $y = \{y^1, \dots, y^L\}$ je extrahováno pomocí transformace komponentových vektorů odpovídající LDA transformační matice. Hlavní vektory jsou pak vypočteny jako

$$y^k = (W^k)^T z^k, k = 1, \dots, L \quad (7.18)$$

Pro komponentovou metodu je obraz tváře kompaktní reprezentace množinou LDA hlavních vektorů $\{y^1, \dots, y^L\}$. Při využití kombinované metody je množina LDA hlavních vektorů v rozšířeném tvaru $\{y^0; y^1, \dots, y^L\}$, kde pro holistický přístup je tvář reprezentována vektorem y^0 [49].



Obrázek 22: Definice komponent na 14 separací (a) a 5 separací (b) [50].

8.2.2 Podobnostní vyhledávání

Mějme tedy dva obličejové obrazy x_1, x_2 reprezentované pomocí LDA množiny hlavních vektorů y_1 a y_2 , podobnost $d(y_1, y_2)$ pro komponentní LDA metodu je měřena váženou sumou vzájemných korelací mezi odpovídajícími složkami

$$d(y_1, y_2) = \frac{1}{L} \sum_{k=1}^L w_k \frac{y_1^k \cdot y_2^k}{\|y_1^k\| \|y_2^k\|}, \quad (7.19)$$

kde y_1, y_2 jsou hlavní LDA vektory k -té obličejové komponenty tváře a w_k jsou faktory z k -té obličejové komponenty. Pro kombinovanou LDA metodu k uvažuje od 0, pro holistickou metodu k je jen 0 [49].

8.2.3 Rekurzivní podobnostní vyhledávání

Rekurzivní vyhledávání je metodou s přijatelnými výsledky. Pokud obecně uvažujeme libovolný snímek tváře, jeho reprezentace obsahuje i proměnlivé změny prostředí stejně tak jako vlastnosti tváře. Při dotazu na databázi získáme tváře odrážející jistou chybu, která je tvořena změnami prostředí a snímací techniky. Pokud bychom získali shodně ovlivněný obraz jako je v databázi, bude vyhledání úspěšné, což je ale nepříjemné. Předpokládejme tedy, že většina z tváří zařazených na vrcholu podobnosti budou rozpoznány podle charakteristik tváře, kde je pozice obrazu obličeje ovlivněna mírně jinými vlivy prostředí než dotazovaný obraz [50].

Předpokládejme obrazový dotaz q (vektor popisu z), kterým chceme získat obrazy v seřazené posloupnosti z obrazové databáze $\{I_i \mid i = 1, \dots, N\}$. V prvním kroku získáme setříděnou množinu obrazů $\{q_i \mid i = 1, \dots, M\}$ z obrazové databáze, která koresponduje s polem ohodnocení $\{s(i) \mid i = 1, \dots, M\}$, kde $s(i) = D(q, q_i)$ vzdálenostní funkce $D(z_1, z_2)$ je Euklidova vzdálenost vektorů z_1 a z_2 .

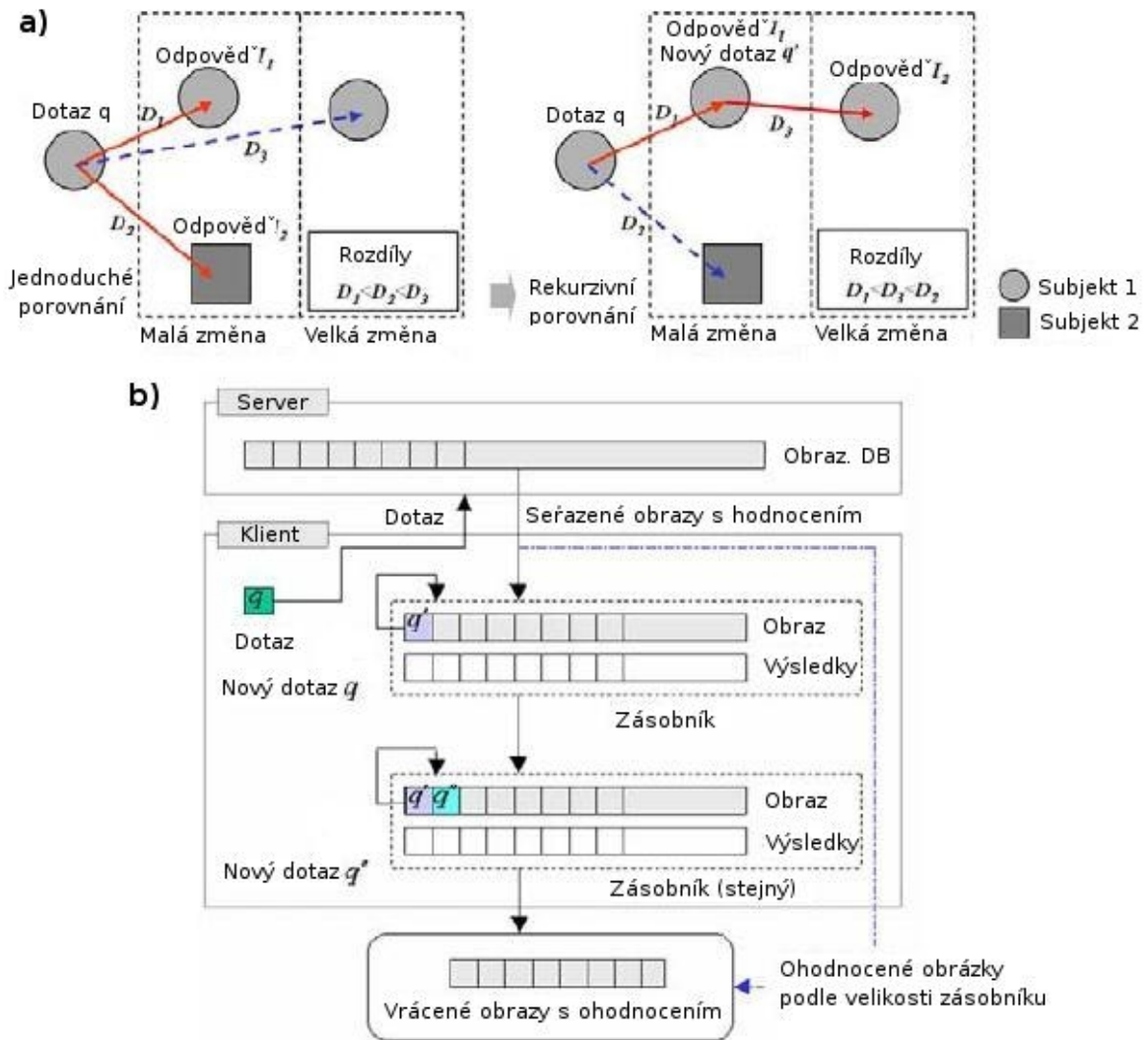
V dalším kroku je první získaný obraz q_1 zvolen za nový obraz q' a porovnávání a setřídění se opakuje jen uvnitř vyrovnávací paměti o velikosti M , ne s celou databází. Výsledkem je znovu setříděná množina obrazů $\{q'_i \mid i = 1, \dots, M\}$ s odpovídajícím polem ohodnocení $\{s'(i) \mid i = 1, \dots, M\}$. Jednotlivé rekurzivní kroky aktualizují ohodnocení obličejů a lze ji definovat jako

$$s'(i) = s(i) + w_1 \cdot D(q', q_i'), \quad (7.20)$$

kde w_i je váhová konstanta. Vytvoříme rekurzivní funkci, pomocí které po M krocích dostaneme seříděnou posloupnost, tedy

$$s^{(n-1)}(i) = s^{(n-2)}(i) + w_{(n-1)} \cdot D(q^{(n-1)}, q_i^{(n-1)}) \quad [50]. \quad (7.21)$$

Díky vlastnosti, že se neporovnává v rekurzivních krocích celá databáze, ale probíhá přepočít pouze ve vnitřní paměti, je výpočetní složitost rekurzivní metody zanedbatelná ve srovnání s jednoduchým srovnáváním.

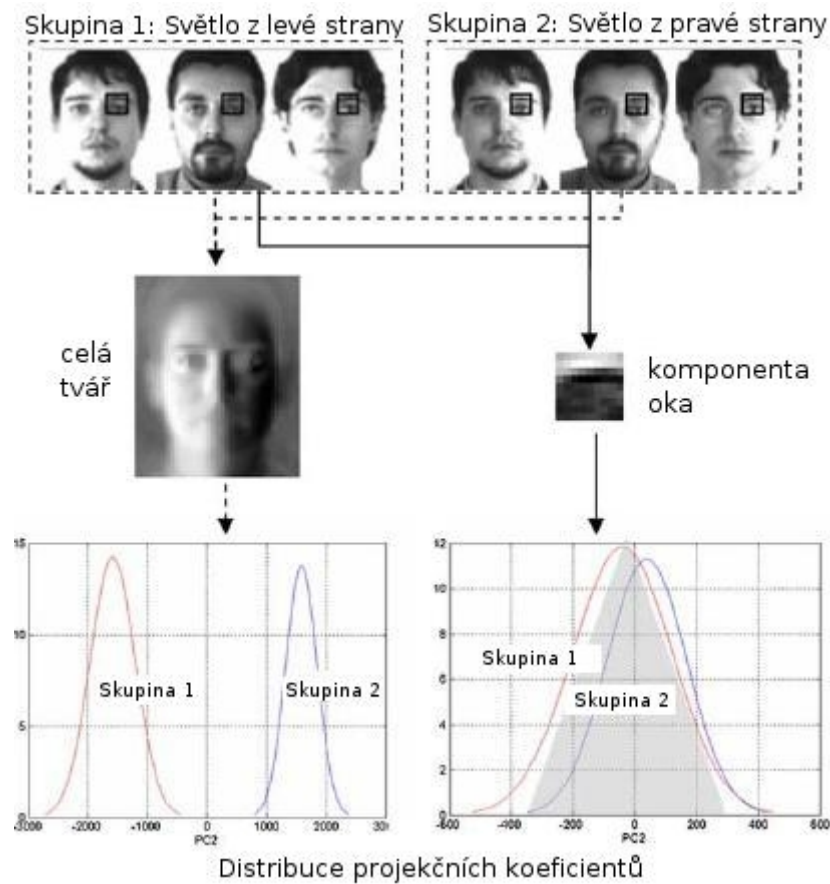


Obrázek 23: Rekurzivní porovnávání tváře. a) model koncepce b) blokový diagram [50].

8.3 Použité Rozpoznání obličeje

Deskriptory pro rozpoznávání obličeje jsou používány pro nalezení tváře odpovídající testovanému obličeji. Deskriptor v podstatě reprezentuje projekci vektoru tváře na soubor bazových vektorů, pomocí kterých jsou extrahovány z normalizovaných obličejů tváře.

Středy očí v každé tváři obrázků jsou umístěné na 24. řádku a 16. (pravé) a 31. (levé) sloupci. Normalizovaný obraz je pak používán pro zjištění odpovídajícího vektoru tváře. Výsledná množina je pak vypočtena jako projekce vektoru na prostor definovaný množinou bazových vektorů [12]. Rozpoznávání obličeje tedy transformujeme na vyhledání nejbližších sousedů daného bodu a měření vzdáleností mezi body, tedy obličejů.



Obrázek 23: Účinek osvětlení celého obrazu v porovnání s komponentou oka [50].

9 Výsledky

Cílem je vytvořit identifikační algoritmus s využitím datových struktur definic deskriptorů MPEG-7, jehož součástí je základní množina datových struktur pro zobrazování obrazů v vysokodimenzionálním prostoru využívající metody počítačového vidění. Využil jsem tedy těchto dostupných nástrojů vyvinutých předními odborníky a implementoval jsem ucelený nástroj pro identifikaci obličejů. Na množině testovacích obličejů chci zjistit kvalitu tohoto nástroje.

Nejprve se pokusím aplikovat detekci na jednotlivé galerie, kde je mým cílem výběr snímků, ve kterých je lokalizován obličej a bude použitelný pro účely identifikace. Na množině lokalizovaných obličejů budu následně zjišťovat, pro která natočení hlavy vůči obrazovce bude identifikace nejčtetnější, čímž zjistíme nejvhodnější natočení hlavy vůči kameře.

Dalším testem bude postupné porovnání všech obličejů s celou databází, tedy snaha o nalezení odpovídajících obličejů k předložené tváři. Tento test probíhá již na rozmanitějších vzorcích dat, proto předpokládám nízký výskyt odpovídajících shod, a měla by se také projevit špatná použitelnost nekvalitních vzorků dat.

Všechny níže zmíněné skripty jsou přiloženy na CD. Budete-li chtít provést všechny níže uvedené kroky, nemusíte jednotlivé zmiňované skripty spouštět ručně, ale výhodnější je spustit `sh build_all all`, který provede všechny kroky a vypíše tabulky s přehlednými statistikami. Tyto skripty jsou umístěny v rootu databáze, konkrétně `{projekt}/newsrc/testset/facedb/`.

9.1 Vstupní data

Jako množinu vstupních dat jsem vytvořil databázi o počtu 4200 snímků ve formátu PGM, které jsou přiloženy na CD. Tato databáze je tvořena z voně dostupných galerií jako jsou *AT&T*, *BioID*, *Yale* a *UseNIX* popsané níže.

Databáze obličejů *AT&T* {1} je tvořena galeriemi jednotlivců, kde v každé je série fotek s drobnou rotací hlavy (zhruba o 10°), ale vždy jsou viditelné důležité rysy obličeje. Jednotlivé galerie jsou označeny podle vzoru *P-rotace[-bryle]-s[1-40]*, kde *P* je označení pohlaví (*M*, *Z*) a na konci je

identifikace konkrétní galerie. Tato databáze je zřejmě pro účely testování identifikačního algoritmu nejvhodnější, protože směr pohledů má jen malý rozptyl, a v obrazech nejsou žádné rušivé elementy.

Galerie *bioID* {2} je galerií více osob, kde je od každé z nich vždy několik fotografií. Najdeme zde různé pohledy lidí do kamery, vždy s nějakou konkrétní chybou celé série. Některé mají obličej neúplný a excentricky umístěný, jiné jsou zaměřeny na emoční změny rozložení rysů obličeje a podobně.

Databáze *UseNIX* {4} je série po sobě jdoucích fotografií vždy s několika snímky jedné osoby se zaměřením na nějakou konkrétní chybu. Galerie jsou pojmenovány podle vzoru *P-rotate[bryle]-1[a-t]*. Snímky v galerii vždy tvoří posloupnost rotace hlavy vůči kameře, od polohy hlavy z profilu, tedy pohled subjektu svírá s kamerou úhel 90° , až do polohy, kdy se subjekt dívá přímo do kamery a pokračuje v pohybu dále až jeho pohled svírá s kamerou úhel 20° . Snímky jsou vždy pootočený o zhruba 5° .

Poslední dvě testovací galerie jsou z databáze *Yale* {3}, které jsou na první pohled s nevyhovující kvalitou, v nízkém rozlišení a některé snímky jsou příliš příliš tmavé. Zde předpokládám nízký poměr úspěšných rozpoznání obličeje v důsledku nepříznivých vlivů působících negativně na kvalitní identifikaci subjektu.

Na datovém základě těchto galerií vytvořím seznamy vstupních souborů pro jednotlivé galerie a pro celou databázi obrázků, se kterou budu dále porovnávat vyhledávané subjekty. K tomuto účelu můžete použít skript `create_all_lst_files`, který vytvoří v každé galerii soubor s názvem galerie a příponou *lst*, do kterého vloží úplné cesty k souborům. Dále vytvoří seznam *databaze.lst*, kde jsou cesty na všechny dostupné soubory celé vstupní databáze. Pro každý seznam je dále nutné vytvořit soubor s parametry pro spuštění. Pojmenování a umístění těchto souborů je analogické se seznamy souborů pouze s tím rozdílem, že mají vždy příponu *par*. Pro ukázkou je níže uveden soubor *databaze.par*, který je použit při aplikaci identifikace nad celou databází. V tomto souboru nejprve specifikujeme název aplikace pro jádro MPEG-7, následuje položka *ListFile*, ve které je cesta k souboru se seznamem snímků. *BitStream* soubor je databázovým souborem, do kterého jsou ukládány deskriptory obličejů v binární podobě nebo ve formátu XML, což lze nastavit posledním parametrem *CodingMode*. Zbývá už jen specifikovat soubor s histogramem barev *FaceHist* a počet srovnání *NoOfMatches*.

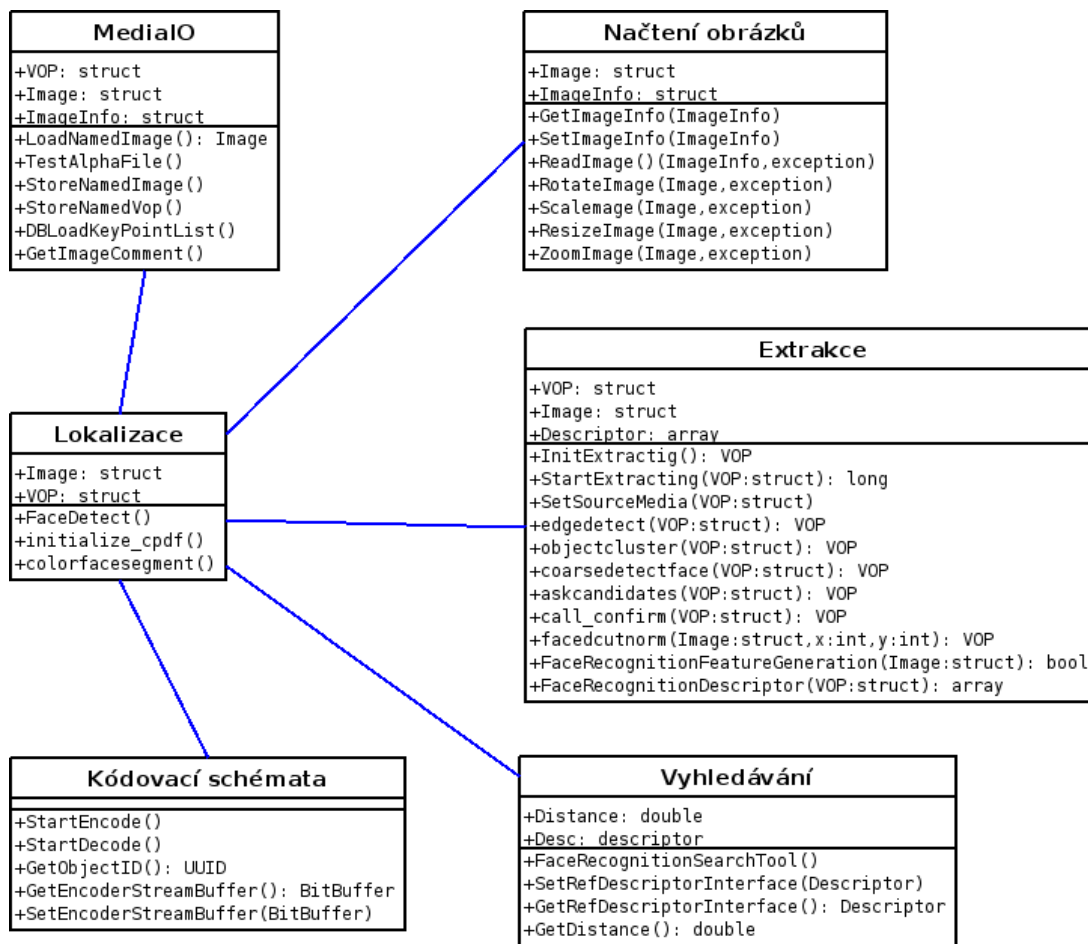
Konfigurační soubor `database.par`:

```
/* jméno spouštěné aplikace */
Application FaceRecognitionClient
/* soubor se seznamem obrázků v databázi*/
ListFile testset/facedb/database.lst
/* soubor s deskriptory přijatých obličejů z ListFile */
Bitstream Bitstreams/FaceRecognition.mp7
/* histogram barevné palety */
FaceHist Parfiles/FaceHist.raw
/* počet porovnání */
NoOfMatches 100
/* Mód ukládání dat DDL: 0 Binary: 2*/
CodingMode 2
```

Konfigurační soubory vytvořím současně se seznamy obrázků obdobným způsobem pomocí skriptu `create_par_file`, pro jednotlivé galerie, nebo `create_par_DB`, pro celou databázi. Budete-li chtít provést vytvoření všech konfiguračních a seznamových souborů, bude nejjednodušší použít hlavní skript s přepínačem `reload`, tedy konkrétně `sh build_all reload`.

9.2 Implementace

V této kapitole probírám konkrétní implementaci identifikačního procesu, který můžeme rozdělit na vytvoření databáze deskriptorů stranou serveru a porovnávání dvojic deskriptorů obličejů na straně klienta.



Obrázek 23: Schéma aplikace rozpoznávání obličeje.

9.2.1 Načtení snímků

Nejprve ze konfiguračního souboru zjistím seznam snímků a s využitím komponenty *ImageMagick* načítám snímky, což je zapouzdřeno ve funkci *LoadNamedImage()*. Tento proces probíhá tak, že je nejprve naplněna struktura *ImageInfo* s informacemi o obrázku funkcí *GetImageInfo()*. Doplní se informace o obraze, kterými jsou signatura, komprese, velikost, a fronta struktur *Image*. Ta je tvořena položkami s typem barevného prostoru, počtem řádků a sloupců obrazu, počtem barev, hloubkou a polem struktur s obrazovými daty *PixPacket*, kterou naplní funkce *ReadImage()*.

9.2.2 Vektorová prezentace objektů (VOP)

Následně převedu bitmapový obraz na vektorovou prezentaci objektu *VOP*, kde jsou položky důležité pro porovnávání, jako výška a šířka obličeje *width* a *height* a odpovídající horizontální a vertikální reference v obraze *hor_spat_ref* a *ver_spat_ref* s měřítkem *scaling*.

Hlavními daty jsou odpovídající kanály YUVA barevného modelu obrázku a dva seznamy *VOP* struktur. Prvním je seznam případů stejných obličejů a druhý seznam je případ podobných obličejů. Nejprve inicializuji nový *VOP*, do kterého následně konvertuji jednotlivé RGB kanály na kanály YUV a alpha složku A. Doplním zbylé výše zmíněné informace o rozměrových vlastnostech snímku a může začít lokalizace obličeje.

9.2.3 Lokalizace obličeje

Funkce *call_FaceDetect()*, nejprve zkontroluje existenci pomocné složky *obrazek.dir* se souborem *faces.txt*, který nabývá hodnot 0 nebo 1, podle úspěšnosti detekce obličeje, a popřípadě ještě souřadnice očí ve zdrojovém obraze. V takovém případě je zde navíc soubor *FaceXX.jpg* o velikosti 46 x 56 pixelů s normalizovaným obličejem subjektu. Při absenci těchto informací se volá samotná detekce rysů funkcí *FaceDetect()*. Zde se provede inicializace histogramu *initialize_cpdf()* a po úspěšném načtení souboru s histogramem se volá segmentace barev tváře, tedy funkce *colorfacesegment()*.

9.2.4 Segmentace barev a extrakce rysů

Nejprve proběhne detekce očí a využitím modré složky RGB modelu, převedeným na binární obraz vynásobená Gaborovými koeficienty (Obrázek 8). Převedu obraz na YCbCr (5.10) barevný model vhodný pro detekci barvy kůže a vyhledáme oblast obličeje (Obrázek 6). Poslední je lokalizace úst, na což je použita FLD transformace (5.18) a opět se vytvoří binární obraz (Obrázek 7).

Pokračuji extrakcí rysů *feature_extract()*, kde se provede především detekce očí v obraze *edgedetect()*, shlukování *objectcluster()* a pak hrubá detekce tváře *coarsedetectface()* v jednotlivých binárních mapách. Pokud zatím vše proběhlo úspěšně, přistoupím k vyhledání kandidátních oblastí

askcandidates(), ze kterých vyřadím nesmyslné oblasti a zavolám funkci *call_confirm()*, která verifikuje zbylé oblasti jako reálný obličej podle vzájemné vzdálenosti kandidátních oblastí v obraze.

9.2.5 Normalizace

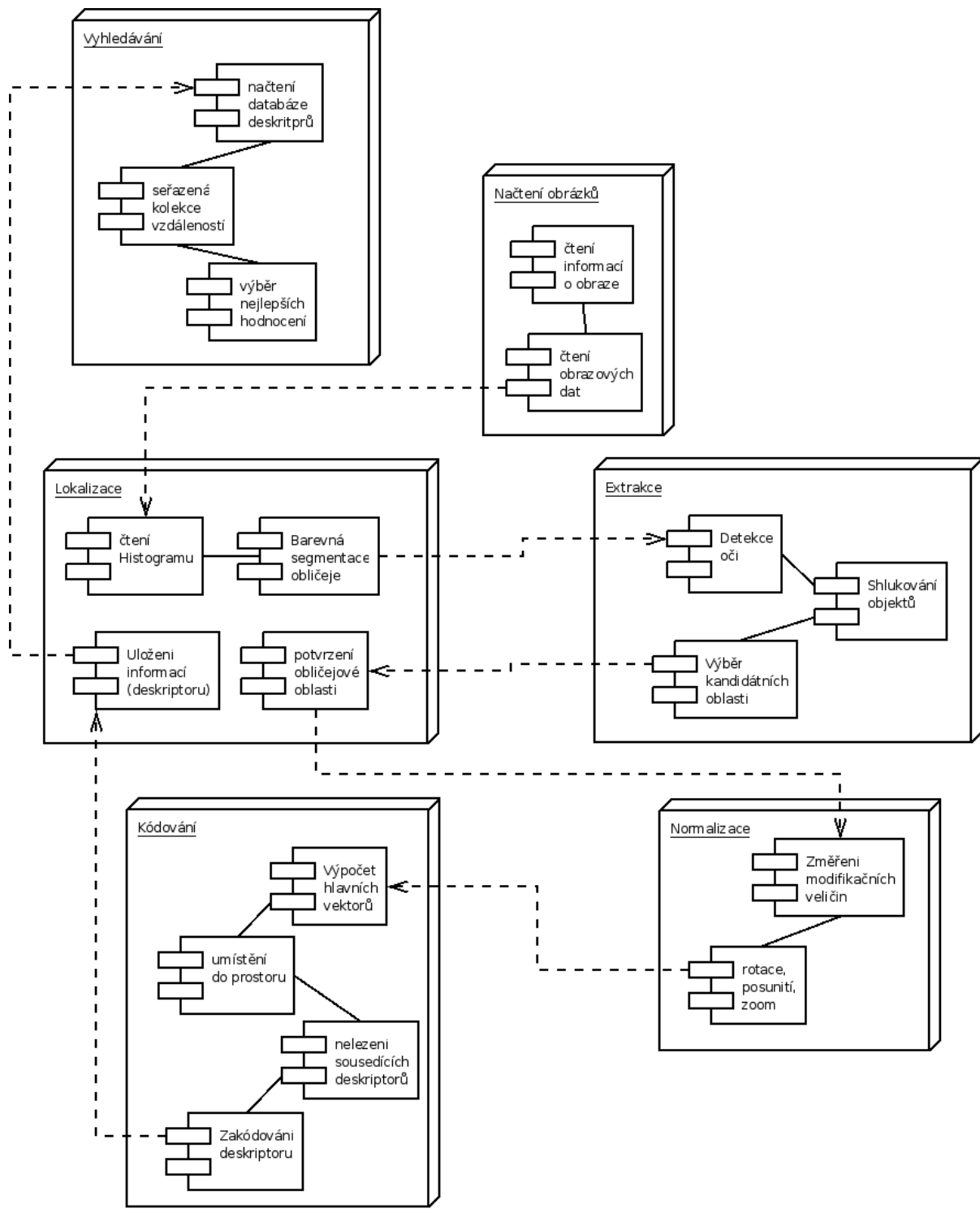
Provedu normalizaci *facedcutnorm()* obrazu podle úhlu, který svírá přímka protínající obě oči s horizontální osou a zmenšení podle poměru vzdálenosti očí ve vstupním obraze a v normalizovaném obraze. Tento obrázek následně uložím pod názvem *FaceXX.jpg* do odkládacího adresáře pro pozdější využití.

9.2.6 Deskriptor

Funkcí *InitExtracting()* vytvořím deskriptor odpovídající tváře, který nabývá 48 celých čísel a je reprezentací hlavních vektorů obličeje ve vysokodimenzionálním prostoru a vytvoří se voláním funkce *FaceRecognitionFeatureGeneration()*. Zde se po extrakci vektoru snažím najít nejvhodnější umístění vektoru ve vysokodimenzionálním prostoru a najít nejbližší sousední deskriptory, které mají vysokou podobnost. Funkcí *StartEncode()* jej převedu z číselného pole na řetězec oddělený dvěma mezerami. Nyní zbývá tento deskriptor uložit do databázového souboru *Bitstream*, v binárním formátu nebo ve formátu XML.

9.2.7 Vyhledávání

Vyhledávání probíhá na straně klientské části aplikace, která začíná obdobně jako server, tedy načtením obrázku do VOP, reprezentace a následné vytvoření odpovídajícího deskriptoru tváře. Deskriptory jsou po dvojicích porovnávány funkcí *FaceRecognitionSearchTool()*, která vrací Manhattan vzdálenost těchto vektorů jednotlivých složek ve vysokodimenzionálním prostoru.



Obrázek 24: Diagram spolupráce jednotlivých komponent.

9.3 Výstupní data

Po provedení identifikačního procesu získáme řadu přímých a nepřímých výstupních informací. Dále data zpracování *nazev.xxx.dir*, které jsou umístěny vždy ve složce se zdrojovými soubory, *nazev* je jméno souboru a *xxx* je přípona zdrojového obrázku. V těchto adresářích při úspěšném rozpoznávání jsou uloženy 2 soubory. Prvním souborem je obrázek typu JPEG s názvem *Face00.jpg*, který má rozlišení 46 x 56 pixelů a je v něm uložen normalizovaný obličej s umístěním očí na 24 řádku, 16 a 31 sloupci. V druhém textovém souboru s názvem *faces.txt* je na prvním řádku uložena informace, byl-li proces rozpoznávání úspěšný (0 nebo 1) a na třetím řádku jsou uloženy souřadnice očí ve zdrojovém obraze.

9.3.1 Přímé výstupy

Mezi hlavní přímá výstupní data se bezesporu řadí vytvořené deskriptory tváří. Jejich reprezentace může být binární, taková data pak nejsou přímo čitelná, nebo ve formátu XML, jak vidíme na příkladu níže.

Ze strany klientské části rozpoznávání získáme výstup v souboru *obrazek.cmp*, který je také uložen u zdrojového souboru. V něm najdeme na každém řádku nejprve cestu k porovnávaným obličejům v databázi a následně hodnotu shody (vzdálenosti) a status byl-li rozpoznán nebo ne. Tyto výstupy získáme jednoduše použitím příkazu `sh build_all compare`.

Příklad výstupu *10.pgm.cpm* porovnávání obrázku *galerie s4* (UseNIX):

```
testset/facedb/s4/3.pgm      dist=32248 OK
testset/facedb/s4/7.pgm      dist=33424 OK
testset/facedb/s4/8.pgm      dist=34808 OK
testset/facedb/s35/5.pgm     dist=74648
```

9.3.2 Nepřímé výstupy

Mezi nepřímé výstupy patří dílčí analýzy, jako je analýza lokalizace obličeje, kterou můžeme snadno vytvořit pomocí skriptu `eval_table`. Tento skript projde všechny výsledné *cmp* soubory a vytvoří statistiku počtu přijatých vzorků obličejů z jednotlivých galerií, která je vypsána na obrazovku a uložena v souboru *lokalizace.csv*. Výpis tohoto souboru je zapsán do tabulky 2, více o výsledcích budu diskutovat později.

Zajímavější dílčí analýzou je analýza počtu rozpoznaných obličejů pro jednotlivé obličeje ve srovnání s vlastní galerií nebo s celou databází. Pokud budeme porovnávat obličej s vlastní galerií, snažíme se zjistit pro který úhel natočení obličeje bude identifikováno nejvíce odpovídajících obličejů. Takováto analýza je důležitá zejména jako informaci pro uživatele systému, tedy v praxi by se zřejmě jednalo o nakreslení barevného bodu s nápisem „Dívejte se sem“, aby bylo dosaženo kvalitnějších vstupních dat díky intuitivnímu řešení. Na první pohled se jedná o drobnost, ale zamyslíme-li se nad dopadem tohoto aspektu, může to významně přispět ke kvalitě systému jako celku. Výpis analýzy některých galerií naleznete v tabulce 3.

Ve srovnání s celou databází už máme větší množství různorodých dat. Na těchto trénovacích datech již nebudeme dosahovat tak vysokého počtu nálezů k jednotlivým obrázkům. Tuto analýzu provedete jednoduše příkladem `sh build_all tests`.

9.4 Diskuse

V této kapitole se pokusím zamyslet nad poznatky a chybami vyplývajícími z výstupních dat. Mou snahou bude vyvodit závěry, které by mohly přispět ke zlepšení rozpoznávacího systému.

Budu se zde podrobně zabývat analýzou vhodných vstupních dat pro úspěšný identifikační proces a možnosti rozptylu čelního pohledu obličeje.

9.4.1 Rozpoznání obličeje

Jako přijatelné vstupní snímky mohou označit takové snímky, které mají úspěšně lokalizovaný obličej v obraze. Byly v nich tedy nalezeny informace potřebné pro správné určení antropometrických

veličin, a byla tedy lokalizována oblast hlavy, obě oči a ústa. V galeriích jednotlivců nejsou velké rozdíly úspěšnosti u mužů nebo žen.


Kód jedince	Počet snímků	Lokalizováno	Rozpoznáno	Vlastnosti
s1	10	100%	100%	muž s proměnlivým přiblížením
s2	10	40%	100%	muž s brýlemi
s3	10	100%	100%	muž, polovina snímku s přivřenými víčky
s4	10	90%	100%	muž s brýlemi i bez nich, 1x zavřené oči
s5	10	100%	100%	muž s úsměvem
s6	10	100%	100%	muž s brýlemi
s7	10	70%	100%	muž s vousy, některé s brýlemi, pohyb rtů
s8	10	100%	98%	žena s různým pohybem očí
s9	10	100%	100%	muž
s10	10	100%	100%	žena s pohledy jinam a úsměvem
s11	10	100%	100%	muž, 2 snímky se zavřenýma očima
s12	10	90%	100%	muž, různé emoce
s13	10	100%	100%	muž s brýlemi i bez nich, pleš
s14	10	60%	100%	muž s brýlemi a vousy
s15	10	100%	100%	muž, pohyb rtů
s16	10	100%	100%	muž s vousy
s17	10	100%	100%	muž s brýlemi i bez nich
s18	10	100%	90%	muž s úsměvem
s19	10	90%	100%	muž s brýlemi i bez nich, pohyb rtů
s20	10	90%	99%	muž s brýlemi i bez nich, grimasy
s21	10	100%	100%	muž s brýlemi
s22	10	100%	100%	muž , afroameričan
s23	10	100%	100%	muž
s24	10	100%	100%	muž, pohyb rtů
s25	10	100%	100%	muž s knírem
s26	10	100%	100%	muž s plnovousem
s27	10	100%	89%	muž s brýlemi, i se zavřenýma očima
s28	10	80%	99%	muž s brýlemi a plnovousem
s29	10	100%	98%	muž s různými pohledy
s30	10	100%	100%	muž, pohyb rtů
s31	10	40%	100%	muž s brýlemi
s32	10	100%	99%	žena, pohyb rtů
s33	10	100%	97%	muž, pohyb rtů
s34	10	60%	100%	muž s brýlemi
s35	10	100%	100%	žena, pohyb rtů
s36	10	90%	100%	muž, 1 snímek se zavřenýma očima
s37	10	50%	93%	muž s brýlemi a plnovousem
s38	10	100%	100%	muž, pohyb rtů
s39	10	100%	90%	muž, pohyb rtů
s40	9	100%	100%	muž, pohyb rtů

Tabulka 2 : Hodnocení lokalizace a rozpoznávání obličejů v jednotlivých galeriích.

Lokalizace obličeje dosahuje tedy nejlepších výsledků u čelních pohledů s malým rozptylem pohledů. Průměrná úspěšnost lokalizace dosahuje na testované databázi *AT&T* 91,25%, což je způsobeno zejména snímky s chybějícími rysy nebo natočením hlavy, a tyto snímky jsou tedy nepoužitelné pro identifikaci. Nejvhodnější data pro identifikaci poskytuje databáze *AT&T*, kde jsou čelní pohledy s malým rozptylem, a bylo rozpoznáno 98,8% z lokalizovaných obličejů. Tato databáze tvoří datový základ pro tabulku 2.

9.4.2 Normalizované obličeje

Vždy se musím rozhodnout, zda-li je snímek použitelný pro identifikaci nebo nikoliv. Rozpoznáváním v databázi *UseNIX* a *Yale* chci zjistit nejkvalitnější úhel pohledu obličeje, které mají ve vlastní galerii co nejvíce četností nálezu. Pokud neuvažuji jedinou fotku, ale například krátkou videosekvenci (lze chápat i jako sérii fotek), pak by takové analýzy mohlo být využito u automatizovaných systémů k výběru nejlepšího vzorku pro uložení do databáze, která by měla pouze tyto vzorky.

	Úhel natočení obličeje vůči kameře						
	-15°	-10°	-5°	0°	5°	10°	15°
Lokalizováno:	78,0%	82,0%	85,7%	89,0%	84,3%	83,3%	77,9%
Rozpoznáno:	72,0%	85,9%	91,1%	94,2%	92,0%	86,1%	76,5%
Ukázka:							

Tabulka 3 : Výpis úspěšných a neúspěšných rozpoznání obličejů v jednotlivých galeriích.

Na testovaných datech je úspěšně lokalizován obličej u 49,5% snímků. Výrazný rozdíl oproti poměru úspěšně lokalizovaných obličejů v *AT&T* je způsoben zejména rozmanitostí rotací hlavy a jen polovina snímků má viditelné všechny důležité rysy obličeje pro úspěšnou lokalizaci. Nejvíce četností správných identifikací se pohybuje opět okolo ideálního normalizovaného pohledu obličeje, kde úspěšná správná rozpoznání dosahují 92,3%.

9.4.3 Identifikace obličeje

Na vstupních datech jsem tedy provedl celou řadu testů, nejvýznamnějším z nich je porovnání snímků na množině různorodých vstupních dat. Ve velkém množství dat nenabývají nálezy vysokých počtů shod jako v rámci galerií. Za předpokladu výběru vhodných databází, jakou je databáze AT&T, je identifikace čelních a jim blízkých rotací pohledu s přesností skoro 99% a dosahuje výsledků, které je již možné využít v praxi jako jednu z bezpečnostních biometrických vlastností. Předpoklad, že přítomnost jedné tváře v databázi s různými úhly pohledu povede k vyšší pravděpodobnosti úspěšného rozpoznání se nepotvrdil, protože pouze zvyšuje rozptyl dat a nepřispívá pozitivně k úspěšné identifikaci jedince.

	TP	TN	FP	FN
AT&T	98,8%	99,6%	0,4%	1,2%
BioID	85,7%	91,0%	9,0%	14,3%
UseNIX	92,3%	94,7%	5,3%	7,7%
Yale	82,0%	79,8%	20,2%	18,0%

Tabulka 4 : Statistika výsledků identifikace rozdělená podle databází.

10 Závěr

Cílem je snaha o dokonalejší a neomylnější způsoby reprezentace a rozpoznání lidského obličeje v běžných situacích. Existuje řada metod používaných pro rozpoznávací úkoly, ale pouze některé jsou použitelné v reálných situacích, kde rozpoznávají na základě podnětů zkoumaných částí.

Mnoho z těchto algoritmů využívá metod vzhledu a základního schématu pro definici obličeje. Nevýhodou těchto metod je bezesporu velká chybovost při změně pohledu a změně světelných poměrů v obraze. Uživatel identifikačního systému tedy má jen malý prostor rozptylu pohledu tváře. Navrhované algoritmy musejí být tedy imunní proti vnímání poměrné světelné změny proti změnám rozpoznávanými mezi dvěma snímky.

Další otázkou je v podstatě určení postačující přesnosti samotného rozpoznávání, použití takové metody pro změření vzdáleností dvou bodů (obličejů) ve vysokodimenzionálním prostoru, který nám uchovává veškeré potřebné markanty obličejů.

Při realizaci identifikace obličejů malé lokální organizace, nebude podobnost obličejů mezi identifikovaným perzonálem až tak markantní, ale budeme-li přemýšlet v globálním světle, dostáváme v podstatě 6 miliard v zásadě podobných objektů s malými rozlišnostmi různě vzdálených bodů pro identifikaci, mezi kterými budou vznikat podobnosti podle stupně granularity ukládaných subjektů.

Při implementaci identifikačního algoritmu jsem využil jádra implementace standardu MPEG-7, která byla ve vývojové verzi uvolněna skupinou Moving Experts Picture Grup. Zde jsem se setkal s celou řadou chyb, z nichž většina se týkala chybné syntaxe spojené s kompatibilitou s aktuálními komponentami ImageMagick, Xerces a AFsp. Nevýhodou této implementace je potřeba starších verzí komponent, což přináší řadu problémů a tvoří tedy tento nástroj méně dostupným.

Ve skutečnosti je vzájemná vzdálenost dvou deskriptorů zjišťována pomocí Manhattan vzdálenosti. Využití jiných metod by mohlo vést ke zlepšení přesnosti algoritmu.

Na trhu máme hned celou řadu biometrických systémů pro identifikaci lidí, avšak většina z nich naráží na své limity. Zájem o lidskou tvář pramení zřejmě z její unikátnosti a ze snahy nalézt proces pro automatickou identifikaci, což patří bezesporu mezi základní lidské schopnosti.

Literatura

- [1] PHILIPS, Peter, MICHAELS, Richard, BLACKBUM, David, GROTHOR, Peter, TABAASI, Edward, BONE, John. *FRVT 2002: Overview and summary.*“, Březen 2003, Dostupný z WWW: <<http://www.frvt.org/FRVT2002/documents.htm>>.
- [2] HINNER, J. *Detekce a rozpoznávání obličejů osob a jejich identifikační význam, Kriminallistika* [cit.2006-09-10]. červen 2003, Dostupný z WWW: <http://www.mvcr.cz/casopisy/kriminallistika/2003/03_01/hinner.html>.
- [3] VLACH, Jan, PŘINOSIL, Jiří. *Lokalizace obličeje v obraze s komplexním pozadím. Elektrovue* [online]. 2007 [cit. 2008-05-01]. Dostupný z WWW: <<http://www.elektrovue.cz/cz/download/lokalizace-obliceje-v-obraze-s-komplexnim/>>. ISSN 1213-1539.
- [4] DELAC, Kim, GRGIC, Martin, GRGIC, Simon. *Independent comparative study of PCA, ICA and LDA on the FERET data set*, 2006, Dostupný z WWW: <http://www.vcl.fer.hr/papers_pdf/Independent%20Comparative%20Study%20of%20PCA,%20ICA,%20and%20LDA%20on%20the%20FERET%20Data%20Set.pdf>.
- [5] VIDAL, R. *HW 3: Learning Theory II*. [s.l.] : [s.n.], [2006?]. Dostupný z WWW: <<http://www.vision.jhu.edu/teaching/hw3-06.pdf>>. s. 3.
- [6] KONOPKA, Ondřej. *Analýza nezávislých komponent*. [s.l.], [2004?]. 36 s. Vedoucí oborové práce Ing. Jakub Šťastný. Dostupný z WWW: <<http://amber.feld.cvut.cz/bio/konopka/file/LBR-semesterka.pdf>>.
- [7] SCHLESINGER, Pavel. *Klasifikační postupy pro analýzu biologických dat*. [s.l.], [2004]. 72 s. Vedoucí diplomové práce Prof. RNDr. Jaromír Antoch, CSc. Dostupný z WWW: <http://ufal.mff.cuni.cz/~hladka/ML/FINAL_CUT.pdf>.
- [8] MINH, Ha Quang, NIYOGI, Partha, YAO, Yuan. *Mercer's Theorem, Feature Maps, and Smoothing*. [s.l.] : [s.n.], [2004?]. Dostupný z WWW: <<http://people.cs.uchicago.edu/~niyogi/papersps/MinNiyYao06.pdf>>. s. 15.
- [9] LAURENZ, Wiskott, FELLOUS, Jean-Marc, KRUGER, Norbert. *Face Recognition by Elastic*

Bunch Graph Matching : Intelligent Biometric Techniques in Fingerprint and Face

Recognition. [s.l.] : [s.n.], c1999. 23 s. Dostupný z WWW:

<<http://www.face-rec.org/algorithms/EBGM/WisFelKrue99-FaceRecognition-JainBook.pdf>>.

ISBN 0-8493-2055-0.

- [10] LOOS, Hartmut, WIECZOREK, Dagmar. *Computer-based recognition of dysmorphic faces* [online]. 2003 [cit. 2007-12-28]. Dostupný z WWW:
<<http://www.nature.com/ejhg/journal/v11/n8/full/5200997a.html>>. ISSN 1018-4813.
- [11] Martínez, J. *ISO/IEC JTC1/SC29/WG11. MPEG-7 Overview* [online]. Palma de Mallorca : 2004 [cit.2005-10-22].
Dostupný z WWW: <<http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>>.
- [12] SAVVIDES, Marios, KUMAR, Vijaya, KHOSLA, Pradeep. *Face Verification using Correlation Filters*. [s.l.] : [s.n.], [2002?]. 6 s.
Dostupný z WWW: <http://www.ece.cmu.edu/~kumar/Biometrics_AutoID.pdf>.
- [13] NICHOLLS, Anthony. *OpenEye ROC explanation* [online]. 1997-2008 , 2007 [cit. 2007-12-24].
Dostupný z WWW: <<http://www.eyesopen.com/about/events/presentations/acs01c/roc2.html>>.
- [14] DOUBEK, Petr. *Mean-Shift segmentace. CMP FEL ČVUT Praha* [online]. 2007 [cit. 2008-05-10]. Dostupný z WWW:
<cmp.felk.cvut.cz/cmp/courses/DZO/Cviceni/cv2/meanshift/meanshift.pdf>.
- [15] Wikipedia contributors. *Kernel (statistics)* [online]. Wikipedia, The Free Encyclopedia; 2008 March 18 [cit. 2008-05-10].
Dostupný z WWW: <[http://en.wikipedia.org/wiki/Kernel_\(statistics\)](http://en.wikipedia.org/wiki/Kernel_(statistics))>.
- [16] KEPENEKCI, Burcu. *Face recognition using gabor wavelet transform*. [s.l.], 2001. 118 s.
Vedoucí diplomové práce AydınAlatan A. Dostupný z WWW:
<www.eee.metu.edu.tr/~alatan/PAPER/MSburcu.pdf>.
- [17] *Morphology Fundamentals: Dilation and Erosion* [online]. The MathWorks, Inc., 2007 [cit. 2008-05-10]. Dostupný z WWW:
<<http://www.mathworks.com/access/helpdesk/help/toolbox/images/f18-12508.html>>.

- [18] FISHER, Richard, PERKINS, Samuel, WALKER, Andrew. *Erosion : Morphology* [online]. 2003 [cit. 2008-05-10]. Dostupný z WWW: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm>.
- [19] JANSSEN, Rik, VOSSEPOEL, Albert. *Adaptive Vectorization of Line Drawing Images*. Academic Press [online]. 1995 [cit. 2008-05-10]. Dostupný z WWW: www.ph.tn.tudelft.nl/People/albert/papers/IU97.pdf. ISSN 1077-3142.
- [20] LIPOCAK, Zdravko, LONCARIC, Sven. *A Scale-Space Approach to Face Recognition From Profiles*. University of Zagreb, Faculty of Electrical Engineering and Computing [online]. 1999 [cit. 2008-05-10]. Dostupný z WWW: <http://ipg.zesoi.fer.hr/papers/caip99.pdf>.
- [21] DATTORRO, Jon. *Convex Optimization & Euclidean Distance Geometry : Euclidean Distance Matrix*. [s.l.] : Meboo Publishing USA, 2005. Dostupný z WWW: www.stanford.edu/~dattorro/EDM.pdf. s. 96.
- [22] Wikipedia contributors. *Euclidean distance* [online]. Wikipedia, The Free Encyclopedia, 2004 , 27 April 2008 [cit. 2008-05-10]. Dostupný z WWW: http://en.wikipedia.org/wiki/Euclidean_distance.
- [23] Wikipedia contributors. *Distance* [online]. Wikipedia, The Free Encyclopedia, 2004 , 10 May 2008 [cit. 2008-05-10]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Distance>.
- [24] AI ACCESS. *Mahalanobis distance* [online]. 2004 [cit. 2008-05-10]. Dostupný z WWW: http://www.aiaccess.net/English/Glossaries/GlosMod/e_gm_mahalanobis.htm.
- [25] Wikipedia contributors. *Chebyshev distance* [online]. Wikipedia, The Free Encyclopedia, 2004 , 9 April 2008 [cit. 2008-05-10]. Dostupný z WWW: http://en.wikipedia.org/wiki/Chebyshev_distance.
- [26] Wikipedia contributors. *Taxicab geometry : Manhattan distance* [online]. Wikipedia, The Free Encyclopedia, 2004 , 6 April 2008 [cit. 2008-05-10]. Dostupný z WWW: http://en.wikipedia.org/wiki/Manhattan_distance.
- [27] *Fonetický přístup*. 2004. 28 s. Dostupný z WWW: www.ibot.sav.sk/karolx/Prednasky/ZaFeKla2.ppt.
- [28] VÝRUT, Radek. Minkowského suma. *Fakulta aplikovaných věd : Západočeská univerzita v Plzni* [online]. 2006 [cit. 2008-05-10].

- Dostupný z WWW: <home.zcu.cz/~rvyrut/WWW-KMA/publications/gcg2006.pdf>.
- [29] Wikipedia contributors. *Hashovací tabulka* [online]. Wikipedia, The Free Encyclopedia, 2004, 9. 4. 2008 [cit. 2008-05-01].
Dostupný z WWW: <http://cs.wikipedia.org/wiki/Hashovac%C3%AD_tabulka>.
- [30] Wikipedia contributors. *Hashovací funkce* [online]. Wikipedia, The Free Encyclopedia, 2004, [cit. 2008-05-10].
Dostupný z WWW: <http://cs.wikipedia.org/wiki/Hashovac%C3%AD_funkce>.
- [31] PETRAKIS, Euripides. *Database Indexing Methods*. [s.l.] : [s.n.], 2005. 41 s.
Dostupný z WWW: <www.intelligence.tuc.gr/~petrakis/courses/multimedia/indexing.pdf>.
- [32] BRUNO, Preiss. *Data Structures and Algorithms with Object-Oriented Design Patterns in C++ : Division Method* [online]. 1997 [cit. 2008-05-10].
Dostupný z WWW: <<http://www.brpreiss.com/books/opus4/html/page211.html>>.
- [33] BRUNO, Preiss. *Data Structures and Algorithms with Object-Oriented Design Patterns in C++ : Middle Square Method* [online]. 1997 [cit. 2008-05-10].
Dostupný z WWW: <<http://www.brpreiss.com/books/opus4/html/page212.html>>.
- [34] BRUNO, Preiss. *Data Structures and Algorithms with Object-Oriented Design Patterns in C++ : Multiplication Method* [online]. 1997 [cit. 2008-05-10].
Dostupný z WWW: <<http://www.brpreiss.com/books/opus4/html/page213.html>>.
- [35] ELLIS, Heidi, WEATHERBY, Gerard. *Hashing*. [online]. 2004 [cit. 2008-05-10].
Dostupný z WWW: <www.rh.edu/~gerardw/fa02d/hasheDictionaries.pdf>.
- [36] University of Saskatchewan. *Hashing : The Folding Method* [online]. 1998 [cit. 2008-05-10].
Dostupný z WWW:
<https://www.cs.usask.ca/resources/tutorials/csconcepts/1998_1/gen_hashing/2-3.html>.
- [37] Wikipedia contributors. *Cormackovo hashování* [online]. Wikipedia, The Free Encyclopedia, 2004, 28. 9. 2007 [cit. 2008-05-10]. Dostupný z WWW:
<http://cs.wikipedia.org/wiki/Cormackovo_hashov%C3%A1n%C3%AD>.
- [38] BRUNO, Preiss. *Data Structures and Algorithms with Object-Oriented Design Patterns in C++ : Fibonacci Hashing* [online]. 1997 [cit. 2008-05-10].
Dostupný z WWW: <<http://www.brpreiss.com/books/opus4/html/page214.html>>.

- [39] Wikipedia contributors. *Strom (datová struktura)* [online]. Wikipedia, The Free Encyclopedia, 2004 , 21. 2. 2008 [cit. 2008-05-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Strom_%28datov%C3%A1_struktura%29>.
- [40] Wikipedia contributors. *Hilbert curve* [online]. Wikipedia, The Free Encyclopedia, 2004 , 9 May 2008 [cit. 2008-05-10]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Hilbert_curve>.
- [41] CMELARĚ, Petr. *Multimediální databáze*. 2006. 57 s. Dostupný z WWW: <www.fit.vutbr.cz/study/courses/VPD/public/0506VPD-Chmelar.pdf>.
- [42] KATAMAYA, Norio. *Sintesis de estructuras fractales deterministas* [online]. 2004 [cit. 2008-05-01]. Dostupný z WWW: <<http://coco.ccu.uniovi.es/geofractal/capitulos/03/03-09.shtm>>.
- [43] IBM. *R-Tree Index Structure* [online]. IBM Informix User's Guide. 2004 [cit. 2008-05-10]. Dostupný z WWW: <<http://publib.boulder.ibm.com/infocenter/idshelp/v10/topic/com.ibm.rtree.doc/rtree29.htm>>.
- [44] KATAMAYA, Norio, SHINICHI, Saton. *The SR-Tree: An Idnex Structure for High-Dimensional Nearest Neighbor Queries*. *NACISIS* [online]. 2002 [cit. 2008-05-10]. Dostupný z WWW: <<http://research.nii.ac.jp/~katayama/homepage/papers/SIGMOD-97.ps.gz>>.
- [45] KATAMAYA, Norio. *The SR-Tree (Sphere/Rectangle-Tree)* [online]. 2003 , 30 Apr 2003 [cit. 2008-05-10]. Dostupný z WWW: <<http://research.nii.ac.jp/~katayama/homepage/research/srtree/English.html>>.
- [46] WHITE, David, JAIN, Ramesh. Similarity Indexing with the SS-tree. *IEEE* [online]. 1996 [cit. 2008-05-10]. Dostupný z WWW: <make.cs.nthu.edu.tw/alp/course/cs5730/teacher%20paper/SS-Tree.pdf>. ISSN 1063-6382.
- [47] SKOPAL, Tomáš, KOLOVRAT, Michal, SNÁŠEL, Václav. Využití LSI a M-stromu při indexování a vyhledávání obrázků. Katedra softwarového inženýrství, MFF UK Praha [online]. 2004 [cit. 2008-05-10]. Dostupný z WWW: <<http://siret.ms.mff.cuni.cz/skopal/pub/znal2005.pdf>>.
- [48] LOBAZ, Petr. *Multimediální a Hypermediální systémy*. [s.l.] : [s.n.], 2004. 23 s. Dostupný z WWW: <www.kiv.zcu.cz/~lobaz/mhs/prednasky2004/mhs_2004_02_neg2up.pdf>.

- [49] TAE-KYUN, Kim, HYUNWOO, Kim, WONJUN, Hwang. Component-based LDA Face Descriptor for Image Retrieval. *Human Computer Interaction Lab* [online]. 2002 [cit. 2008-05-10]. Dostupný z WWW: <www.bmva.ac.uk/bmvc/2002/papers/85/full_85.pdf>.
- [50] TAE-KYUN, Kim, HYUNWOO, Kim, WONJUN, Hwang. Component-based LDA face description for image retrieval and MPEG-7 standardisation. *Image and Vision Computing 23* [online]. 2005 [cit. 2008-05-10]. Dostupný z WWW: <svr-www.eng.cam.ac.uk/~tkk22/doc/ivc05_published.pdf>. ISSN 631-642.
- [51] JAE-HO, Lee. Automatic Video Management System Using Face Recognition and MPEG-7 Visual Descriptors. *ETRI Journal : Volume 27* [online]. 2005 [cit. 2008-05-10]. Dostupný z WWW: <<http://etrij.etri.re.kr/Cyber/servlet/GetFile?fileid=SPF-1134115008222>>.
- [52] PALMER, Michael. *Principal Components Analysis* [online]. 2007 [cit. 2007-12-25]. Dostupný z WWW: <<http://ordination.okstate.edu/PCA.htm>>.
- [53] PCA: Principal Component Analysis & ICA: Independent Component Analysis [online]. 1999-2005 [cit. 2007-12-22]. Dostupný z WWW: <<http://www.his.kanazawa-it.ac.jp/~idurumi/ica/pcaica.html>>.
- [54] DIACO, Anthony, DICARLO, Jeffrey, SANTOS, Juan. Gender Recognition : Linear discriminant analysis [online]. 2001 [cit. 2007-12-22]. Dostupný z WWW: <<http://scien.stanford.edu/class/ee368/projects2000/project15/algorithms.html>>.
- [55] BLANZ, Volker. A Morphable Model for the Synthesis of 3D Faces [online]. 1998 , [cit. 2007-12-20]. Dostupný z WWW: <<http://www.kyb.tuebingen.mpg.de/bu/people/volker/>>.
- [56] HIGASHIJIMA, Yamato , TAKANO, Saton, NIIJIMA, Kim. Face Recognition Using Long Haar-like Filters. *Department of Informatics : Kyushu University* [online]. 2005 [cit. 2008-05-10]. Dostupný z WWW: <pixel.otago.ac.nz/ipapers/09.pdf>.
- [57] MedCalc corp.. *ROC curve analysis : Introduction* [online]. 2002 , 22 April 2008 [cit. 2008-05-10]. Dostupný z WWW: <<http://www.medcalc.be/manual/roc.php>>.

Odkazy

- {1} <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- {2} <http://www.bioid.com/>
- {3} <http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html>
- {4} <ftp://ftp.uu.net/published/usenix/faces>

Seznam příloh

Příloha 1. DVD

- Upravené zdrojové kódy MPEG-7
- Kompatibilní verze ImageMagick, Xerces a AFsp
- Databáze obličejů BioID, Yale, UseNIX, AT&T
- Skripty pro analýzu programu
- Vygenerované soubory pro analýzu
- Soubory s výsledky analýzy