

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SLEDOVÁNÍ OBLIČEJOVÝCH RYSŮ V REÁLNÉM ČASE

REAL-TIME FACIAL FEATURE TRACKING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

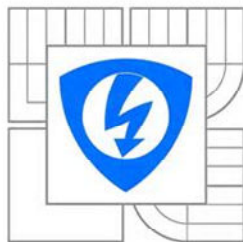
Bc. JAN PELOUŠEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ PŘINOSIL, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jan Peloušek
Ročník: 2

ID: 72753
Akademický rok: 2010/2011

NÁZEV TÉMATU:

Sledování obličejových rysů v reálném čase

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte moderní metody číslicového zpracování obrazových signálů orientované na detekci a sledování klíčových bodů v obrazové scéně. Na základě získaných teoretických znalostí navrhnete metodu pro sledování aktuální pozice a tvaru jednotlivých rysů lidského obličeje (oči, rty, apod.). Navrženou metodu následně implementujte ve vhodném programovacím jazyku s ohledem na zpracování videosignálu v reálném čase.

DOPORUČENÁ LITERATURA:

- [1] Přinosil, J.: Effective Facial Feature Keypoint Detection Using Active Shape Models Composition. In In Proceeding of the 33rd International Conference on Telecommunications and Signal Processing - TSP 2010, s. 200-203. ISBN: 978-963-88981-0- 4, 2010.
- [2] Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Academic Press, ISBN: 978-59749-272-0, 2009.
- [3] Parker, J.R.: Algorithms For Image Processing And Computer Vision, John Wiley& Sons, Inc., ISBN: 0-471-14056-2, 1997.

Termín zadání: 7.2.2011

Termín odevzdání: 26.5.2011

Vedoucí práce: Ing. Jiří Přinosil, Ph.D.

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Anotace:

Tato práce pojednává o problematice rozpoznávání objektů v obraze se zaměřením na rozpoznání lidské tváře a jejích součástí. Je zde popsány základní principy počítačového vidění, objektový detektor Viola-Jones, jeho programová realizace pomocí knihoven OpenCV. Dále je v této práci popsán systém přesné detekce obličejových částí pomocí algoritmu aktivních modelů tvaru (ASM) a s ním související mechanismy trénování klasifikátorů. Nakonec je popsána i praktická realizace včetně softwarové implementace.

Klíčová slova:

Rozpoznávání, objektový detektor, lidský obličej, detektor Viola-Jones, počítačové vidění, aktivní modely tvaru, ASM, Gaborovy filtry, klasifikátor, Principal Component Analysis, PCA, AdaBoost, OpenCV.

Abstract:

This thesis considers the problematics of the object recognition in a digital picture, particularly about the human face recognition and its components. There are described the basics of the computer vision, the object detector Viola-Jones, its computer realization with help of the OpenCV libraries and the test results. This thesis also describes the accurate system of the facial features detection per the algorithm of the Active Shape Models and also related mechanism of the classifier training, including the software implementation.

Keywords:

Recognition, object detector, human face, Viola-Jones detector, computer vision, Active Shape Models, ASM, Gabor Filters, classifier, Principal Component Analysis, PCA, AdaBoost, OpenCV.

Bibliografická citace této práce:

PELOUŠEK, J. Sledování obličejových rysů v reálném čase. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 44 s. Vedoucí diplomové práce Ing. Jiří Přinosil, Ph.D..

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma *Sledování obličejových rysů v reálném čase* jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu práce Ing. Jiřímu Přinosilovi, Ph.D., za velmi užitečnou metodickou pomoc a cenné rady při zpracování mé diplomové práce.

V Brně, dne

Podpis

OBSAH:

1 Úvod	- 1 -
2 Teorie rozpoznávání	- 2 -
2.1 Přístupy:.....	- 2 -
2.2 Strojové učení:.....	- 3 -
2.2.1 Druhy strojového učení:	- 3 -
2.2.2 Klasifikátor:.....	- 4 -
2.2.3 Extrakce příznaků	- 4 -
2.2.4 Předzpracování	- 5 -
2.2.5 Neuronové sítě pro klasifikaci:	- 5 -
2.2.6 Formální neuron:	- 5 -
2.3 Detektor Viola-Jones:	- 6 -
2.3.1 AdaBoost:	- 7 -
2.3.2 Haarovy vlnky:	- 9 -
2.3.3 Integrální obraz:	- 10 -
2.3.4 Detekce objektů v obraze:.....	- 11 -
3 Metody hledání obličejových rysů.....	- 13 -
3.1 Segmentace obrazu	- 13 -
3.1.1 Deformovatelné modely.....	- 14 -
3.2 Aktivní modely tvaru	- 14 -
3.3 Aktivní modely vzhledu	- 15 -
4 Přesné vyhledávání obličejových rysů pomocí ASM	- 16 -
4.1 Popis ASM.....	- 16 -
4.2 Důležité součásti ASM	- 17 -
4.2.1 Analýza hlavních komponent	- 17 -
4.2.2 Jasové profily	- 20 -
4.2.3 Gaborovy filtry.....	- 21 -
4.2.4 GentleBoost	- 22 -
5 Předzpracování obrazu.....	- 23 -
5.1 Barevné modely.....	- 23 -
5.1.1 RGB	- 23 -
5.1.2 Šedotónový obraz.....	- 24 -
5.1.3 YUV	- 24 -

5.2 Ekvalizace histogramu	- 25 -
6. Programová realizace objektového detektoru Viola-Jones.....	- 28 -
6.1 OpenCV:.....	- 28 -
6.2 Objektový detektor pomocí OpenCV	- 28 -
6.3 Popis aplikace FaceRec:.....	- 30 -
7 Praktická realizace ASM.....	- 33 -
7.1 Vytvoření trénovacích dat pro tvorbu modelů a klasifikátorů.	- 33 -
7.2 Výpočet modelů	- 34 -
7.3 Extrakce deskriptorů a příznaků významných bodů.....	- 34 -
7.4 Trénování klasifikátorů	- 35 -
7.5 Postup aplikace ASM	- 35 -
8 Programová realizace ASM.....	- 40 -
9 Závěr	- 42 -
Použitá literatura:.....	- 43 -
Seznam použitých zkratk	- 45 -

Obsah přílohy CD:

- a) Elektronická verze Diplomové práce ve formátu *.pdf*.
- b) Zdrojové kódy aplikace FaceRec
- c) Trénovací databáze 200 olabelovaných obrázků
- d) Zdrojové kódy k aplikaci ASM včetně programů *ViolaLabeler* a *ViolaExtractor*

1 Úvod

Dnešní život se odehrává ve světě, ve kterém jsou často informace cennější, nežli hmotné majetky. Ovšem zároveň je tento svět informacemi doslova zahlcen a není možné vnímat všechny naráz a je potřeba vyhledat a filtrovat pouze ty, které zrovna potřebujeme. Ve výpočetní technice existují poměrně pokročilé techniky pro vyhledávání v textu, kde pracujeme s jasnými požadavky v podobě konkrétních dotazů.

Vnímáme-li ovšem obraz, například fotografii, jako rastr bodů, je pro stroj jednoduše nemožné určit, co se zrovna v obraze nachází. Vektorový popis jednotlivých objektů v obraze by bylo nutné vytvářet ručně, což je při větším množství zpracovávaných obrazů velmi neefektivní. Z tohoto důvodu je nutná určitá dávka strojové automatizace pomocí mechanismů, řadících se do odvětví umělé inteligence - počítačového vidění. Tato práce je zaměřena zejména na popis lidské tváře a a detekci jejich charakteristických znaků (jako oči, nos, ústa a další).

První část práce je zaměřena na detekci obličeje a jeho částí pomocí objektového detektoru Viola-Jones, který bude popsán včetně jeho součástí. Dále bude stručně popsána programová realizace tohoto detektoru a výstupy z testování. Jako součást, nezbytná pro zvýšení robustnosti systému je popsána také metoda ekvalizace histogramu.

V této práci je dále popsáno přesné určení polohy jednotlivých obličejových rysů pomocí algoritmu aktivních modelů tvaru včetně jeho součástí, jako je metoda PCA, filtrování obrazu pomocí Gaborových filtrů, je také popsán navržený systém pro samotnou detekci včetně praktické programové realizace.

Rozpoznání výskytu, polohy a lidské tváře a jejích součástí může mít rozsáhlé využití v oblasti komunikace, zpracování obrazu, bezpečnosti, v biometrice, vytváření ovládacího rozhraní člověk-počítač a dalších.

2 Teorie rozpoznávání

Aby se mohly tyto metody rozšiřovat, musely být vyvinuty rozpoznávací metody ať už s vyšší, či nižší efektivitou a přesností.

Tyto metody se dají rozdělit do čtyř kategorií podle přístupu k získávání požadované informace.

2.1 Přístupy:

- Znalostní metody – využívají pravidla, určující poziční vztahy jednotlivých obličejových rysů.
- Invariantní příznakové metody – uvažují detekci pomocí obecně známých příznaků lidské tváře, které nepodléhají světelným podmínkám, ani pohybu tváře (barva lidské kůže, obličejová textura).
- Metody porovnávání šablon – hledají tváře porovnáváním vstupního obrazu s určitou předdefinovanou šablonou lidské tváře, nebo její částí.
- Metody využívající modely vzhledu – porovnávají části vstupního obrazu s určitým modelem obličeje, generovaným pomocí strojového učení z trénovací množiny vzorků lidských tváří.

Z důvodu vysoké variability vstupního obrazu, ať už se jedná o světelné podmínky, kvalitu obrazu, zašumění, rozdílnosti pozadí, pozice snímací kamery a nastavení objektivu, nebo o samotné rozdíly mezi lidskými tvářemi, není možné použít spolehlivě jedinou metodu. V praxi se používá spíše kombinace jednotlivých metod s ohledem na poměr přesnosti a rychlosti. Tyto algoritmy jsou však poměrně výpočetně náročné a nelze je provozovat plynule v reálném čase. Pro plynulost, jakou si žádá video se proto detekce provádí v každém n-tém snímku a poloha v ostatních snímcích se odhaduje na základě poslední známé pozice. To je dostačující například v případě aplikací pro automatické zaostřování kamery. Pro složitější aplikace, jako gesturální rozhraní, je však tento systém nedostačující a chybějící detekce se doplňují pomocí metod sledování pohybu lidského obličeje.

2.2 Strojové učení:

Je podoblastí umělé inteligence, zabývající se algoritmy a technikami, umožňujícími počítačovému systému „učit se“, neboli získávat nové poznatky a zkušenosti. Podle [8] se formálně strojové učení definuje následovně: Počítač se učí řešit úlohu třídy T se zkušeností E a úspěšností měřenou kritériem P , jestliže se jeho úspěšnost se vzrůstající zkušeností zvyšuje.

Nutností je určit druh znalostí, které se bude stroj učit a jejich reprezentaci včetně návrhu konkrétního algoritmu učení. Jedná se většinou o aproximaci určité cílové funkce, realizující úkol, který chceme počítač naučit řešit. V případě rozpoznávání objektů je nutné počítač naučit, jak je daný objekt reprezentován v obrazové formě.

2.2.1 Druhy strojového učení:

Bez učitele: V tomto případě učící algoritmus nemá k dispozici žádná trénovací data a nezná druh výstupních dat. Vstupy učícího algoritmu jsou shromažďovány a na základě, většinou statistických zákonitostí, vyhodnocovány.

S učitelem: Vstupem algoritmu je soubor trénovacích dat (např. obrazů, vektorů apod.) včetně požadovaného výstupu. Na základě těchto dat je poté vytvářen klasifikátor. Funkce klasifikátoru je mapování vstupních hodnot na výstupní (např.: vstupem je obraz a výstupem logická 0, nebo 1 podle shody). Kvůli omezení velikosti trénovací množiny není možné dosáhnout úplné přesnosti a vždy je nutné předpokládat určitou chybu. Proces učení je ovšem výpočetně náročný proces, proto je nutné použít takovou trénovací množinu, aby byla co nejmenší a zároveň dokázala pokrýt co nejvyšší množství možných případů. Před samotným trénováním dochází k extrakci samotných příznaků a jejich uložení (do vektoru). Trénování pak probíhá ve fázích klasifikace (aktuální verze klasifikátoru zpracovává vektor příznaků a je vyhodnocena chyba) a úprava klasifikátoru (na základě chyby procesu klasifikace se upravuje stávající klasifikátor ke zmenšení chyby).

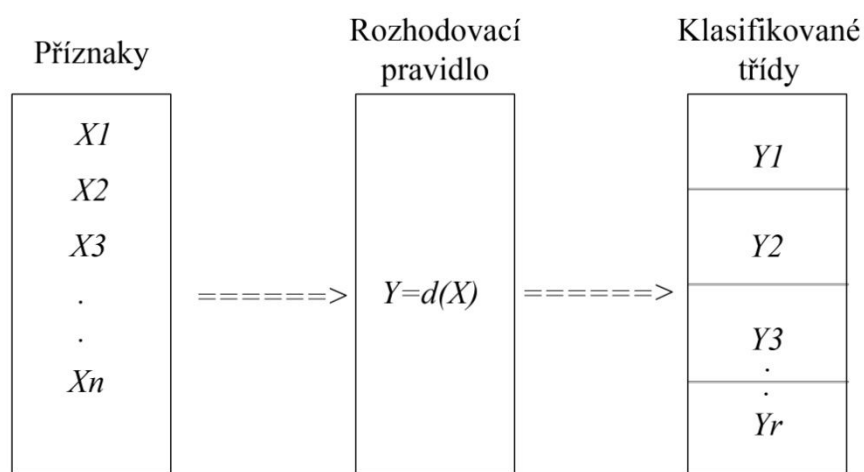
Proces trénování probíhá až do splnění kritéria zastavení. To může být naplněno buď faktem, že již neexistují další trénovací vzorky, nebo je dosažena požadovaná minimální chyba klasifikátoru a další trénování již není potřeba, dále je možné trénovat, dokud chyba mezi dvěma kroky má klesající tendenci s určitou směrnicí. Ve chvíli, kdy začne chybovost opět stoupat, jsme již dosáhli stavu, kdy je klasifikátor přetrénován a je nutné tento proces zastavit.

Od tohoto okamžiku totiž již ztrácí klasifikátor schopnost „generalizace“ a není schopný správně vyhodnocovat jiné vzorky, než právě ty trénovací.

2.2.2 Klasifikátor:

Klasifikátory se řadí mezi neuronové sítě, které rozpoznají vstupní vzor podle příznaků a zařadí ho do příslušné vybrané třídy.

Úlohou klasifikátorů je zařadit vzor na základě podobnosti. Typickým zástupcem tohoto druhu sítí je např. Hammingova síť.



Obr. 2.1: Vstupy a výstupy klasifikátoru

V případě, že se počet výstupních tříd rovná dvěma (výstup je logická 1, nebo 2 – náš případ), jedná se o tzv. úlohy *dichotomické* klasifikace.

2.2.3 Extrakce příznaků

Obrazová data obsahují pro daný účel vysoké množství redundantních informací a není možné je efektivně zpracovávat všechny s ohledem na výpočetní náročnost. Pomocí procesu extrakce příznaků zjednodušíme množinu zpracovávaných dat na minimum tím, že z obrazu vytáhneme pouze popisné veličiny, které potřebujeme (příznaky). Tímto procesem se vyhneme práci s objemnými daty a přitom zachovááme popis daných příznaků v dostatečné přesnosti.

2.2.4 Předzpracování

- Selekce příznaků: Vybereme pouze příznaky, které potřebujeme pro náš účel.
- Selekce vzorů: Je možné vyjmout ze vstupního obrazu některá data, která se vyskytují mimo určitý rozsah. Někdy je k modelu možné i vzory přidat.
- Transformace příznaků: Často je potřeba příznaková data před dalším zpracováním transformovat to jiné podoby pomocí například normalizace, nebo škálování.
- Diskretizace: Problémy s reálnými číselnými atributy rozhodovacích stromů jsou řešeny diskretizačními algoritmy. Výsledky jsou nominální hodnoty, představující příslušnost do určitého intervalu.

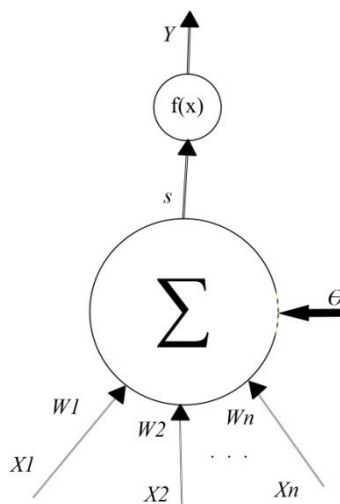
2.2.5 Neuronové sítě pro klasifikaci:

Klasifikátory, založené na neuronových sítích jsou inspirovány postupy v lidském mozku, kde probíhá klasifikace na základě znalostí a zkušeností. Biologická podstata klasifikace, resp. rozpoznávání spočívá v rozsáhlé síti paralelně propojených elementárně propojených výpočetních prvků – neuronů. Spojení je realizováno axony (dendrity), které tvoří vstupy a výstupy jednotlivých buněk.

Definice: Umělá neuronová síť je podle [5] paralelní distribuovaný systém výkonných prvků, modelujících biologické neurony účelně uspořádaný tak, aby byl tento systém schopen požadovaného zpracování informací.

2.2.6 Formální neuron:

Základem matematického modelu neuronové sítě je tzv. formální neuron. Formální neuron má n obecně reálných vstupů X_1, X_2, \dots, X_n , které modelují dendrity. Vstupy jsou ohodnoceny reálnými synaptickými vahami W_{1j}, \dots, W_{ij} , které určují jejich propustnost [12] a mohou mít i zápornou hodnotu. Jádro neuronu pak provádí vážený součet všech vstupních hodnot. Výstup je poté zpracováván určitou přenosovou funkcí, která může mít různé průběhy, např.: lineární, skokový, sigmoida a další.



Obr. 2.2: Model neuronu

První stupeň zpracování vstupních informací tedy realizuje vážený součet vstupů vztažený k prahové hodnotě Θ .

$$s = \sum_{i=1}^R X(i) \cdot W(i) - \Theta \quad (2.1)$$

Prah neuronu Θ určuje míru aktivace přenosové funkce a někdy se zavádí jako nultý prvek vstupního vektoru $X(0)$ s vahou $W(0)$. Aktivace výstupního neuronu je dána jeho přenosovou funkcí f :

$$Y = f(s) = f\left(\sum_{i=1}^R X(i) \cdot W(i) - \Theta\right) \quad (2.2)$$

2.3 Detektor Viola-Jones:

Jedná se o objektový detektor, prezentovaný v roce 2001 Paulem Violou a Michaellem Jonesem. Systém byl tehdy testován na počítači Intel Pentium III, 700MHz a na tomto počítači se detekce prováděla rychlostí 15 snímků za vteřinu při rozlišení vstupního obrazu 384 x 288 pixelů. Dřívější systémy byly schopny pracovat i s vyšší rychlostí, ale bylo zde použito dodatkových informací, jako změna polohy pixelů určité barvy a podobně. Detektor Viola-Jones pracuje v reálném čase s šedotónovými obrazy, a zpracovávání probíhá snímek po snímku. Výhodou tohoto detektoru je tedy rychlost, dostatečná spolehlivost a robustnost, neboť je algoritmus nezávislý na změně osvětlení scény a velikosti sledovaného objektu[10]. Z těchto důvodů je tento algoritmus často využíván právě pro detekci tváře a objevují se i různé modifikace tohoto algoritmu.

Tento detektor se skládá ze tří částí: Integrovaného obrazu, klasifikačního algoritmu Ada Boost, Haarovy vlnky[10].

2.3.1 AdaBoost:

AdaBoost, (co by zkratka z Adaptive Boosting) je klasifikační algoritmus, vycházející z metody strojového učení, zvané Boosting, jehož účelem je zlepšení klasifikační přesnosti algoritmů strojového učení a je určena právě pro úlohy dichotomické klasifikace. Tato metoda byla v roce 1999 publikována Yoavem Freundem a Robertem Schapirem a v roce 2003 oceněna tzv. Gödel praze.

Základem je vytvoření více klasifikátorů, vznikajících pomocí výběru ze základní trénovací množiny, označovaných jako *weak learners* (slabí žáci). První klasifikátor má přesnost o něco větší, než je přesnost odhadu (tj. minimálně 50% v případě dvoustavového klasifikátoru)[3]. Postupným přidáváním dalších klasifikátorů s podobnou mírou přesnosti se vytváří soubor klasifikátorů, označovaný jako *strong learner* (silný žák) s celkovou klasifikační přesností libovolně vysokou vzhledem ke vzorkům trénovací množiny. Jedná se tedy o zesílení klasifikace (boosting).

AdaBoost navíc narozdíl od základního boostingu trénovací množinu váhuje vahami D_t , ze začátku rovnoměrně nastavenými. V každé smyčce algoritmu pak provádí:

Vybírá slabý klasifikátor s nejnižší chybou klasifikace vzhledem k vahám D_t ,

Ověřuje, že chyba klasifikátoru není $>0,5$,

Počítá koeficient slabého klasifikátoru v lineární kombinaci $H(x)$,

Aktualizuje váhy D_t .

Aby bylo zaručeno, že učicí algoritmus bude konvergovat, musí být chyba učení $<0,5$. To se zajistí právě aktualizací vah, kdy se zvýší váha špatně klasifikovaných měření a naopak sníží váha dobře klasifikovaných. Dále hledáme klasifikátor, který bude klasifikovat doposud chybně provedená měření. *AdaBoost* pak podle [10] redukuje chybu trénování exponenciálně v závislosti na rostoucím počtu klasifikátorů. K přetrénování vlivem zvyšujícího se počtu klasifikátorů však dochází podle [10] jen velmi zřídka.

Proces učení klasifikačního algoritmu AdaBoost:

1. *Vstup:*

$$S = \{(x_1, y_1), \dots, (x_m, y_m)\}, \text{počet iterací } T \quad (2.3)$$

2. *Inicializace vah:*

$$D_1(i) = \frac{1}{m} \quad (2.4)$$

3. *Cyklus pro $t=1, \dots, T$:*

a. *Výběr klasifikátoru na základě vážené trénovací chyby*

$$\varepsilon_j = \sum_{i=1}^m D_t(i) \ell[y_i \neq h_j(x_i)] \quad (2.5)$$

$$h_t = \operatorname{argmin}_{h_j \in H} \varepsilon_j$$

b. *Pokud*

$$\varepsilon_t = 0 \text{ nebo } \varepsilon_t \geq \frac{1}{2}, \text{ pak konec cyklu}$$

c. *Nastavení*

$$\alpha_t = \frac{1}{2} \log \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \quad (2.6)$$

d. *Úprava vah*

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}, \quad (2.7)$$

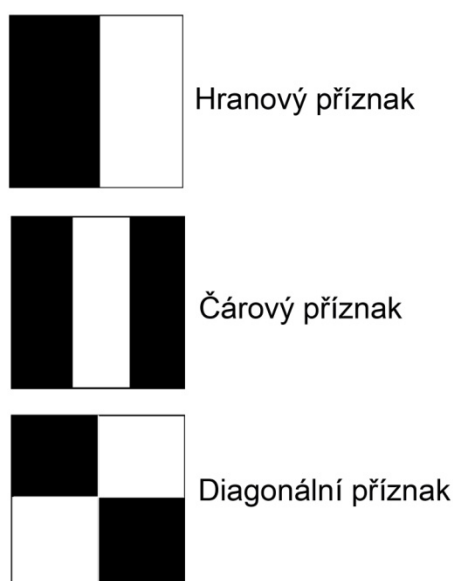
$$\text{Kde } Z_t = \sum_{i=1}^m D_t(i) e^{-\alpha_t y_i h_t(x_i)}$$

4. Výstup

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (2.8)$$

2.3.2 Haarovy vlnky:

Vstupem učicího procesu klasifikačního algoritmu je množina příznaků. S rostoucí velikostí této množiny roste i pravděpodobnost výběru slabého klasifikátoru s vyšší mírou přesnosti. Snahou detektoru Viola-Jones je získat velkého počtu příznaků za vynaložení nízkých výpočetních prostředků. Právě takovými příznaky mohou být právě příznaky založené na principu podobném definici Haarových vlnek, neboli *Haar-like features*. Podle [10] se hodnota takového příznaku vypočítá jako suma pixelů odpovídající světlé části, od které je odečtena suma pixelů tmavé části. Jsou definovány tři druhy příznaků podle počtu a rozvržení světlých a tmavých částí, a to hranový příznak(tvořený dvěma), čárový příznak(třemi) a diagonální(tvořený čtyřmi obdelníkovými oblastmi) , viz Obr. 2.3.



Obr. 2.3: Příznaky podobné Haarově vlnce

Tyto příznaky hledáme v celém vstupním obraze, přičemž se mění velikost daných příznaků a to od velikosti 1x1 pixel až po velikost celého vstupního obrazu, což znamená, že podle [10] pro vstupní obraz o velikosti 19x19 pixelů získáváme přibližně 64 tisíc příznaků, které nám tvoří vstup pro učicí klasifikační algoritmus *AdaBoost*. Ten z nich potom vybírá malé

množství příznaků, společně se stejným počtem natrénovaných slabých klasifikátorů (stejný příznak se ve výsledném silném klasifikátoru může vyskytovat i několikrát, ovšem pokaždé s jiným nastavením slabého klasifikátoru). Pouze toto malé množství příznaků je pak použito při samotné detekci.

2.3.3 Integrální obraz:

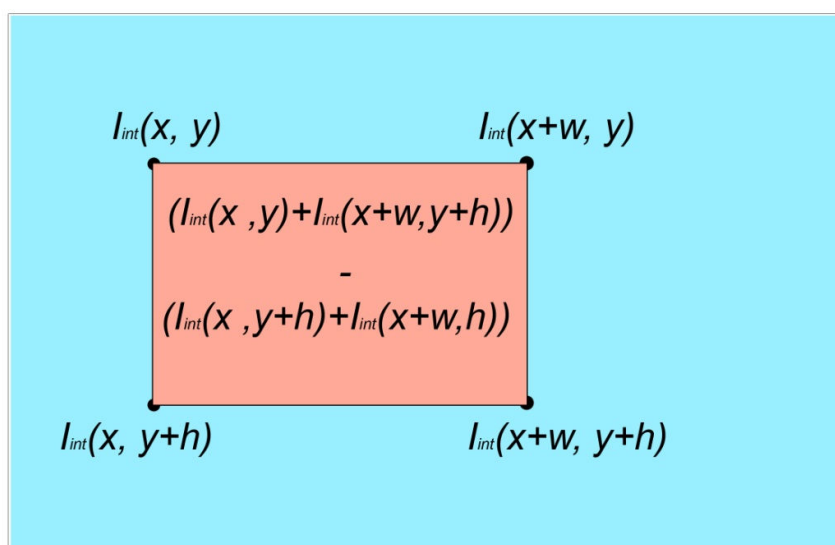
Slouží k efektivnímu a rychlému výpočtu hodnot jednotlivých příznaků v obraze. Kvůli zjednodušení je v reprezentaci integrálního obrazu každý bod obrazu určený součtem hodnot bodů všech bodů předcházejících podle rovnice [13]:

$$s(x, y) = s(x, y - 1) + I(x, y), \quad (2.9)$$

$$I_{\text{int}}(x, y) = I_{\text{int}}(x - 1, y) + s(x, y), \quad (2.10)$$

Kde $s(x)$ je kumulovaný součet hodnot v řádku obrazu, $I(x, y)$ představuje hodnoty intenzit jednotlivých bodů vstupního obrazu a $I_{\text{int}}(x, y)$ jsou jednotlivé hodnoty integrálního obrazu v daném bodě. Platí $s(x, 0) = 0$ a $I_{\text{int}}(0, y) = 0$.

Výpočet hodnot jednotlivých příznaků se pak výrazně zjednoduší, protože na výpočet sumy libovolného obdélníku v obraze postačí dvě operace sčítání a jedna operace odčítání viz obr. 2.4, kde x a y jsou počáteční souřadnice a w , h jsou šířka a výška požadovaného obdélníku.



Obr. 2.4: Příklad výpočty sumy obdelníku v integrálním obraze

2.3.4 Detekce objektů v obraze:

Pro detekci konkrétních objektů v obraze je nutné nejdříve vytvořit databázi pozitivních vzorů právě těchto objektů a negativních vzorů pozadí. Z těchto vzorů je poté pomocí klasifikačního algoritmu *AdaBoost* získána množina samotných optimálních příznaků společně s natrénovanou množinou slabých klasifikátorů a jejich vah. Všechny vzory by měli mít stejný vertikální i horizontální rozměr ($w \times h$, pro zjednodušení uvažujeme ještě $w=h$). Tento rozměr by měl být zároveň co nejmenší z důvodu výpočetní náročnosti trénování s ohledem na dostatečné optické rozlišení sledovaného objektu.

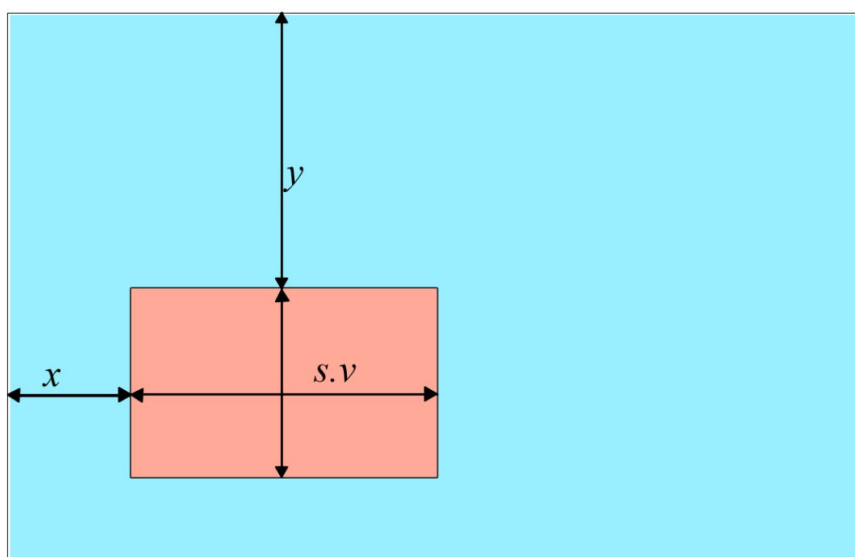
Pro realizaci detektoru tváří je nutné použít soubor trénovacích dat, v našem případě soubor vzorů různých lidských obličejů různé velikosti, barvy kůže (ikdyž detektor pracuje v šedotónovém barevném prostoru), natočení a při různých světelných podmínkách.



Obr. 2.5: Ukázka souboru fotografií obličejů, použitých pro trénování

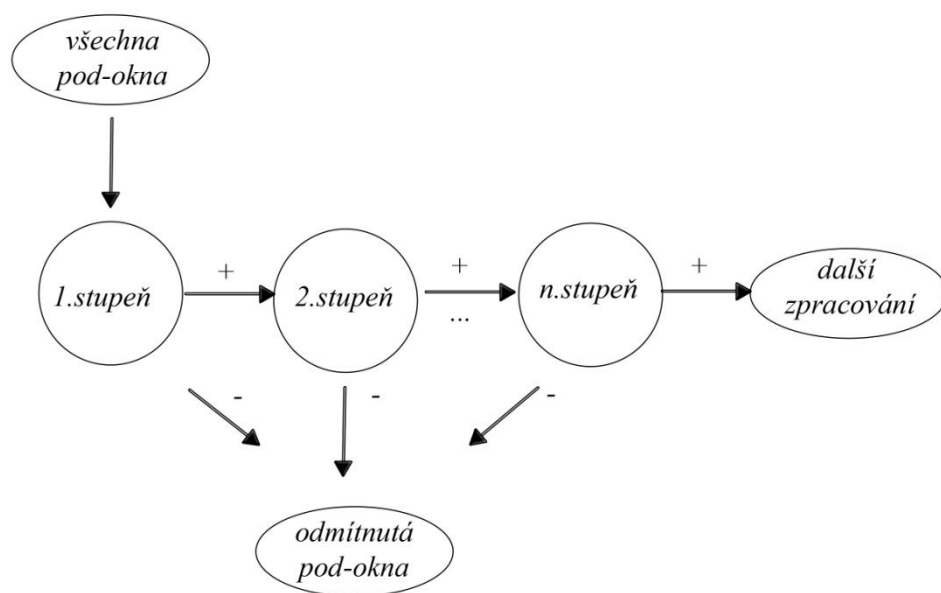
Stejně tak je pro trénování nutné použít soubor vzorů pozadí, které nebudou při detekci vyhodnocovány jako obličeje (negativní vzory).

Při detekci je pak obraz zpracováván po částech, vymezených posuvnými pod-okny. Ty mění v obraze svoji polohu (v osách x a y) i velikost ($s \times w$) v závislosti na předpokládané velikosti detekovaného objektu viz Obr. 2.6, kde w je základní velikost pod-okna, s je faktor zvětšení okna, x a y je poloha pod-okna. Vstupem tohoto klasifikátoru je vybraný příznak pod okna a klasifikátor rozhoduje, zda se příznaky tohoto podokna shodují s příznaky určitého pozitivního vzoru, tedy zda toto pod-okno daný objekt obsahuje, či nikoliv.



Obr. 2.6: Umístění a velikost vyhledávacího pod-okna

Díky zpracování obrazu pomocí velkého množství pod-oken je tento proces vysoce výpočetně náročný (např. podle [10] je při velikosti vstupního obrazu 320x240 pixelů nutné zpracovat asi 500 000 pod-oken s počáteční velikostí $w=19$). Náročnost se dá částečně eliminovat omezením doby, potřebné pro prohledávání jednotlivých pod-oken. To se dá realizovat například kaskádovým zapojením klasifikátorů. Vycházíme z předpokladu, že k vytvoření klasifikátoru, který dokáže vybrat všechny pozitivní a zároveň množství (20-50%) negativních případů, stačí jen několik příznaků. V každém stupni kaskády se také snažíme zamítnout co nejvíce negativních pod-oken, která daný vzor neobsahují a do dalších stupňů kaskády pak vstupují pouze pod-okna obsahující daný vzor. Neboť většina negativních pod-oken byla zamítnuta již na začátku, není nutné pro tato pod-okna extrahovat příznaky pro další stupně kaskády.



Obr. 2.7: Zapojení klasifikátorů do kaskády

2.3.5 Struktura kaskádového detektoru:

Původní výsledný detektor je podle [14] 32 stupňová kaskáda klasifikátorů. První stupeň kaskády byl navržen tak, aby pomocí dvou příznaků dokázal zamítnout zhruba 60% negativních případů pozadí, zatímco dokáže potvrdit téměř 100% případů výskytu obličeje. Druhý stupeň pak obsahuje pět příznaků, přičemž odmítá asi 80% negativních případů při zjištění téměř všech obličejů v obraze. Další tři stupně pracují s 20, následující dva pak s 50, pět se 100 zbývajících dvacet stupňů s 200-příznakovými klasifikátory. Počet příznaků v jednotlivých stupních kaskády byl určen experimentálně vzhledem k chybovosti. Počet stupňů kaskády pak byl zvyšován, dokud chybovost detekce nedosáhla takřka nulové hodnoty za zachování přijatelné rychlosti práce detektoru.

3 Metody hledání obličejových rysů

3.1 Segmentace obrazu

Základní metody hledání obličejových rysů v obraze spadají do kategorie segmentace obrazu. Jednoduchý přístup spočívá v překrytí vztažného obrazu přes obraz, ve kterém hledáme shody. Díky variabilitě obrazových textur a významných bodů je však tato metoda nevhodná. Pro obrazovou segmentaci existuje řada dalších metod, například detekce hran, přímek,

kružnic a další. Tyto techniky dávají základ profilovým modelům, které jsou součástí aktivních modelů tvaru (viz. dále).

3.1.1 Deformovatelné modely

Nejznámějším deformovatelným modelem je *had*, neboli *aktivní model obrysu*. Snahou tohoto modelu je minimalizace vnitřní a vnější energie. Vnější energie je minimalizována za předpokladu, že had zaujme tvar požadovaného tvaru v obraze. Vnitřní energie pomocí omezení zakřivení.

Problémem je, že jednotlivé body hada se mohou navázat na špatné obrazové orientační body. Obtížné je také lokalizovat a opsat ostře zakřivené tvary. Určitými modifikacemi je možné tyto problémy potlačit.

3.1.2 Distribuce bodů v modelu

Tvary jsou reprezentovány pomocí vektorů, můžeme na ně aplikovat standardní statistické modely. Tyto modely jsou naučeny na určitý tvar, získány z trénovací množiny dat. Pomocí využití hlavních komponent (PCA=Principal Component Analysis) jsou pak vytvářeny distribuční modely.

3.2 Aktivní modely tvaru

Ideální distribuční modely se deformují pouze způsobem, charakteristickým pro daný objekt. První parametrický statistický model tvaru pro obrazovou analýzu za použití vnitřních vzdáleností mezi orientačními body modelu prezentovali Cootes a Taylor, nazýváme ho aktivní model tvaru (ASM = Active Shape Models).

Popisuje polohu obličejových částí pomocí rozeznatelných orientačních bodů v oblasti obličeje. Tyto body hledáme pouze v obličejové oblasti, je tedy nutné předzpracování například pomocí obličejového detektoru Viola Jones. Pomocí trénování jsou vypočteny charakteristiky profilu a distribuce orientačních bodů modelu.

3.3 Aktivní modely vzhledu

Aktivní modely vzhledu (AAM = Active Appearance Models) jsou metodou lokace obličejových částí, vycházející z originální metody ASM, publikované Cootesem a Taylorem. Tato metoda spojuje model tvaru a textury to jednoho modelu vzhledu. Zpracovává se textura celého objektu (v našem případě obličeje). AAM potřebuje menší počet orientačních bodů, než ASM, neboť i samotný model textury zachovává velké množství informací.

4 Přesné vyhledávání obličejových rysů pomocí ASM

V předchozím textu byl popsán mechanismus vyhledávání lidské tváře. Dá se tedy předpokládat, že známe její polohu. Další text se zaměří vyhledání přesné polohy jednotlivých obličejových rysů pomocí metody aktivních modelů tvaru (ASM) a jejích hlavních součástí.

4.1 Popis ASM

Model tvaru můžeme uvažovat jako dvojrozměrný vektor (x_i, y_i) N orientačních bodů, které jsou k sobě invariantně vztaženy tak, že rotace ani změna měřítka nemá vliv na změnu modelu.

Mimo vyšší efektivity oproti ostatním lokalizačním algoritmům ASM nepotřebuje ručně definovaný model, přestože trénovací snímky musí být manuálně indexovány. ASM vytváří modely trénováním pomocí sady referenčních snímků, tedy mechanicky extrahují důležité charakteristiky trénovací sady. Ruční doladění je stále potřeba pro lepší výsledky vzhledem k účelu.

Uvažujeme-li pro reprezentaci tvaru jednorozměrný vektor $x' = (x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N)$, pro N bodů, definujících tvar, dostaneme pro M vytvořených modelů M jednorozměrných vektorů x_i' , pro $i=1 \dots M$, o délce $2xN$. Z těchto modelů poté vytvoříme normalizované modely tvaru $x_i'^{(n)}$ a průměrný model tvaru, vypočítaný podle euklidovské vzdálenosti:

$$D_E(\bar{x}', x_i'^{(n)}) = \sqrt{(\bar{x}' - x_i'^{(n)})^2} \quad (4.1)$$

Je pak možné vyjádřit normalizovaný model tvaru díky zarovnání všech modelů tvaru, získaných z trénovacích dat, pomocí funkce transformace podobnosti, řídící rotaci modelu tvaru u úhel θ , změnu měřítka s a translaci o x_p, y_p .

$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} s \cdot \cos \theta & s \cdot \sin \theta \\ -s \cdot \sin \theta & s \cdot \cos \theta \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \quad (4.2)$$

Proces normalizace modelů tvaru a vytvoření průměrného modelu je možné shrnout do postupu o 6 bodech [9]:

1. Translace souřadnic každého modelu tak, aby jejich těžiště $t = (t_x, t_y)$ leželo v počátku souřadnicového systému, tj.

$$t_x = \sum_i^{2 \cdot N} x_i = 0 \text{ a } t_y = \sum_i^{2 \cdot N} y_i = 0. \quad (4.3)$$

2. Náhodný výběr jednoho modelu tvaru jako počátečního průměrného modelu tvaru $\overline{x_0'} = x_i'$ a normalizace jeho velikosti tak, aby platilo

$$|\overline{x_0'}| = \sqrt{\sum_i^{2 \cdot N} \overline{x_{0,i}'} \cdot \overline{x_{0,i}'}} = 1. \quad (4.4)$$

3. Zarovnání všech modelů tvaru k průměrnému modelu tvaru $\overline{x_0'}$ pomocí transformace podobnosti. Zarovnáním vznikají nové normalizované modely tvaru $x_i'^{(n)}$, které nahrazují původní modely tvaru.
4. Výpočet průměrného modelu tvaru odpovídajícímu střední hodnotě všech normalizovaných modelů tvaru

$$\overline{x'} = \frac{1}{M} \cdot \sum_i^M x_i'^{(n)}. \quad (4.5)$$

5. Zarovnání průměrného modelu tvaru $\overline{x'}$ k počátečnímu průměrnému modelu tvaru $\overline{x_0'}$ pomocí transformace podobnosti a normalizace jeho velikosti tak, aby platilo

$$|\overline{x'}| = \sqrt{\sum_i^{2 \cdot N} \overline{x_i'} \cdot \overline{x_i'}} = 1. \quad (4.6)$$

6. Opakování bodů 4. a 5., dokud nedojde k požadované konvergenci (hodnoty průměrného modelu se již nadále příliš nemění).

4.2 Důležité součásti ASM

V následujícím textu budou popsány nejdůležitější součásti mechanismu aktivních modelů tvaru.

4.2.1 Analýza hlavních komponent

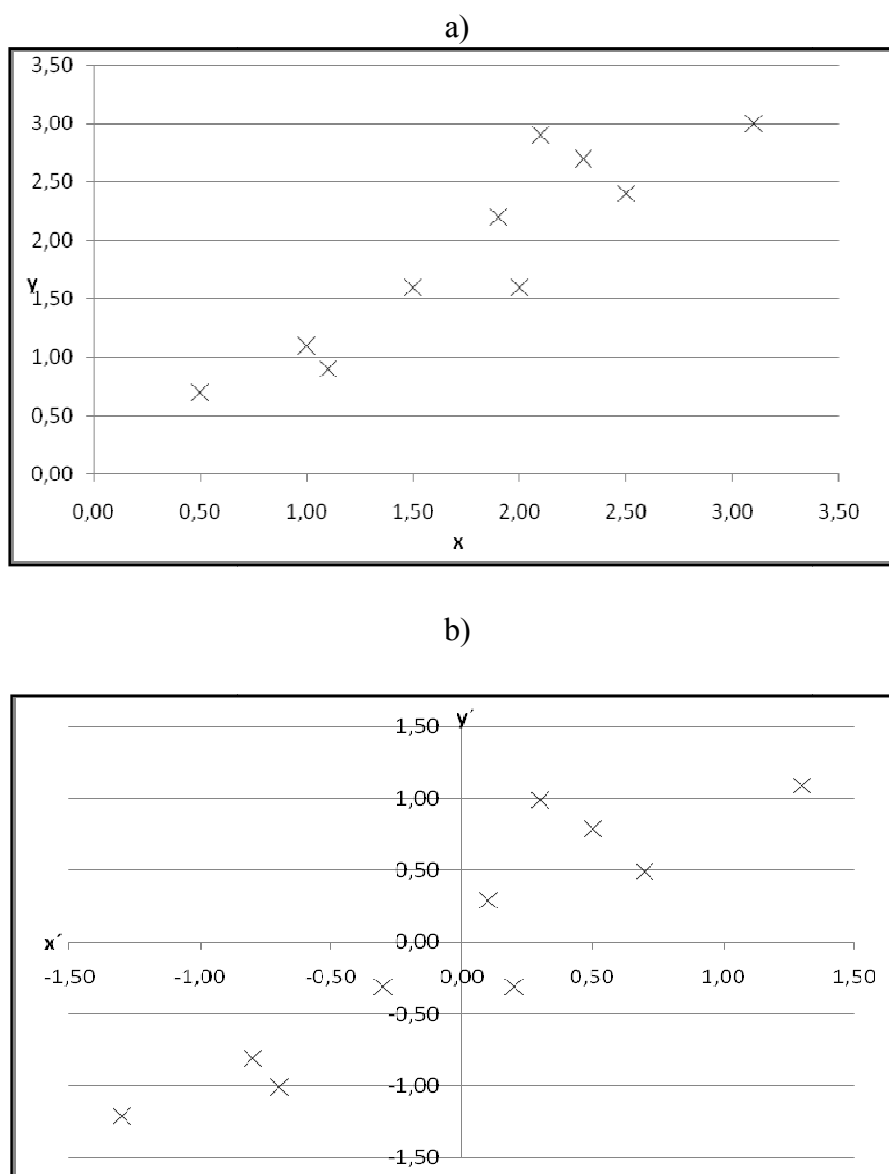
(PCA = Principal Component Analysis)

PCA je transformací, sloužící k de Korelaci dat. Snižuje dimenzionalitu dat za minimalizace ztráty informace. Jedná se o transformaci, při níž se soubor zdrojových dat, obsahující velké množství navzájem souvisejících proměnných přemění na sadu tzv. hlavních komponent takovým způsobem, že je zachováno co největší množství informace, sloužící k separaci jednotlivých kategorií dat [9]. Prvních několik komponent obsahuje nejvyšší množství informace pro účel separace. PCA je založena na rozptylu vstupní sady dat a je definována jako ortogonální lineární transformace.

Nejvyšší variance vstupních dat se nachází na první souřadnici nového systému (první hlavní komponenta), druhá nejvyšší na druhé a tak dále [6].

Postup vysvětlíme na příkladu z literatury [7]:

1. Mějme sadu dat, v dvojrozměrném systému souřadnic x, y o rozměrech $w \times h$ (horizontální x vertikální směr).



Obr.4.1: PCA a) Originální vstupní data, b) Centralizace kolem aritmetického průměru

2. Pro každou souřadnici provedeme výpočet aritmetického průměru a od tohoto průměru odečteme příslušné hodnoty jednotlivých souřadnic, čímž dojde k centralizaci.

Výpočet průměru:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ a } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i. \quad (4.7)$$

Nové souřadnice bychom pak spočítali jako: $x' = x - \bar{x}$ a $y' = y - \bar{y}$. Budeme však kvůli odstranění souvislostních informací uvažovat transformaci zpracovávaných M obrázků do M normalizovaných jednorozměrných vektorů L_1, L_2, \dots, L_M , délky $N = w \times h$. Díky normalizaci pak máme podle [9] zajištěnu invariabilitu transformace vůči světelným podmínkám.

Průměrný vektor pak vypočítáme jako

$$\Psi = \frac{1}{M} \sum_i^M L_i. \quad (4.8)$$

Odečtením tohoto vektoru od všech normalizovaných vektorů L_i získáme nový prostor, ve kterém jsou hodnoty jednotlivých vektorů nekorelované a variance dosahuje maximální hodnoty.

$$\Phi_i = L_i - \Psi, i = 1..M. \quad (4.9)$$

Následně je potřeba nalézt N ortogonálních vektorů u_n , kde $n = 1..N$ právě pomocí analýzy hlavních komponent. Vektory u_n jsou vlastními vektory kovarianční matice C a zároveň hlavními komponenty.

3. Výpočet kovarianční matice:

$$C = \frac{1}{M} \sum_i^m \Phi_i \cdot \Phi_i^T = \frac{1}{M} \cdot A \cdot A^T, \quad (4.10)$$

kde A je matice sloupcových vektorů Φ_i o rozměrech $N \times M$. Výpočet vlastních vektorů kovarianční matice C je definován vztahem:

$$U^{-1} \cdot C \cdot U = D, \quad (4.11)$$

U je N rozměrná diagonální matice vlastních vektorů kovarianční matice C , D je N -rozměrná diagonální matice vlastních čísel kovarianční matice C . Platí zároveň, že n -té vlastní číslo odpovídá n -tému vlastnímu vektoru.

Vektor příznaků ω_i je získán za použití k vlastních vektorů u_n odpovídajících k nejvyšším vlastním číslům.

$$\omega_{i,j} = u_i^T (L_j - \Psi), \quad i = 1..k, j = 1..M. \quad (4.12)$$

Jelikož je určení vlastních vektorů a vlastních čísel kovarianční matice C pro obrazové vzory s vysokým rozlišením výpočetně velmi náročný, lze podle [9] za předpokladu, že počet vzorů bude výrazně nižší, než jejich velikost, určit hodnoty vlastních vektorů i vlastních čísel pomocí lineární kombinace vektorů, získaných výpočtem vlastních čísel matice.

Nalezení vzorů pomocí PCA

Podle [7] mějme sadu obrazů O_1, O_2, \dots, O_n o rozměrech $N \times N$. Každý obraz je reprezentován svým vektorem. Všechny obrazy pak můžeme vložit do matice E pro účely analýzy PCA:

$$E = \begin{pmatrix} O_1 \\ O_2 \\ \vdots \\ O_n \end{pmatrix} \quad (4.13)$$

Po provedení PCA máme sadu vlastních vektorů, určených z kovarianční matice. Úkolem je určit, kterému vzoru náleží zkoumaný objekt (například obličej). Samotné určování pak probíhá na základě výpočtu rozdílnosti jednotlivých vzorů od zkoumaného objektu pomocí dat, získaných z PCA, díky němuž jsme získali originální data, ovšem ve formě vzájemné rozdílnosti a podobnosti mezi vzory.

Jestliže jsou všechny vektory o rozměrech $N \times N$, dostaneme N^2 vlastních vektorů. V praxi můžeme podle [7] vynechat některé méně důležité vlastní vektory za zachování vysoké přesnosti rozpoznávání. Tato metoda v případě zpracování obrazu pracuje pouze s hodnotami jasů v určitém bodě. To může být problém z hlediska nerovnoměrnosti nasvícení. Z tohoto důvodu je vhodná vstupní filtrace obrazu, která toto potlačí.

4.2.2 Jasové profily

Původní ASM, navržená Coatesem a Taylorem využívala pro extrakci příznaků profily jasových intenzit, měřených podél zvolených segmentů [1]. Těmito profily jsou myšleny vektory hodnot jasových intenzit pixelů v okolí daného bodu. Tento vektor je kolmý na

spojnici dvou sousedních bodů. Při vyhledávání pak bylo postupováno podél spojnice těchto bodů v obou směrech.

Tato metoda však byla příliš náchylná ke změnám světelných podmínek a byla později nahrazena metodou, využívající odezvy vstupních obrazů na filtraci Gaborovými filtry a redukcí vstupních dat pomocí PCA.

4.2.3 Gaborovy filtry

Gaboruv filtr je lineární filtr, jehož impulsní odezva je definována harmonickou funkcí, vynásobenou funkcí Gaussova okna. Těchto filtrů se často využívá například při detekci hran, případně významných bodů, neboť s jejich pomocí nepracujeme přímo s jasovou hodnotou, nýbrž její změnou vzhledem k okolním pixelům.

$$g(x, y, \lambda, \theta, \psi, \sigma) = e^{-\frac{x'^2 + y'^2}{2\sigma^2}} \cdot \cos\left(2\pi \frac{x'}{\lambda} + \psi\right), \quad (4.14)$$

kde λ udává vlnovou délku, resp. řád Gaborova filtru, θ udává úhel natočení, ψ fázový posun a σ šířku gaussova okna, na nějž se filtr aplikuje.

$$x' = x \cdot \cos(\theta) + y \cdot \sin(\theta), \quad (4.15)$$

$$y' = -x \cdot \sin(\theta) + y \cdot \cos(\theta), \quad (4.16)$$

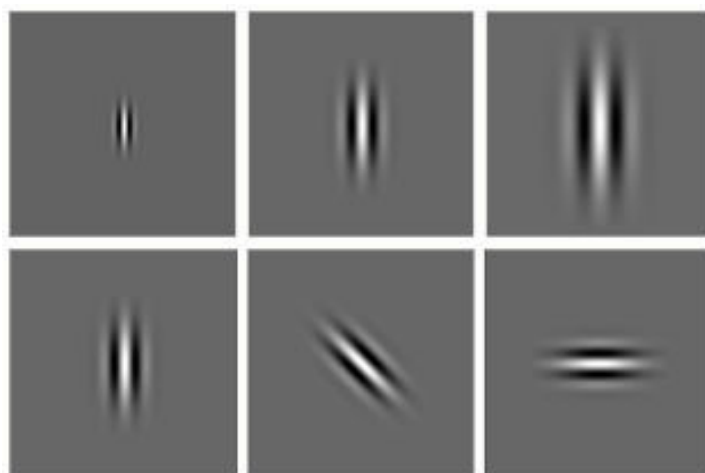
Podle [9] jsme změnou délky Gaborova filtru a jeho prostorové orientace schopni zachovat určitá prostorově-frekvenční pásma zatímco ostatní části jsou potlačeny. Změnou délky filtru λ dochází ke změně velikosti propustného pásma a změnou prostorové orientace θ dochází k frekvenčně-prostorovému posunu propustného pásma.

V dalším textu budeme uvažovat sadu 48 Gaborových filtrů vytvořenou kombinací filtrů s následujícími parametry:

$$\theta = \left\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\right\},$$

$$\lambda = \{2\sqrt{2}, 4, 4\sqrt{2}, 8, 8\sqrt{2}, 16\},$$

$$\sigma = \lambda \text{ a } \psi = 0.$$



Obr. 4.2: Různé vlnové délky a různé úhly natočení Gaborových vlnek



Obr. 4.3: Filtrace Gaborovými filtry s různými vlnovými délkami (vlevo originál)

4.2.4 GentleBoost

Pro účel trénování klasifikátorů byl oproti případu detektoru Viola-Jones použit trénovací algoritmus Gentle Boost, který je modifikací zde již popsaneho algoritmu AdaBoost.

Hlavní rozdíl podle [4] je ve způsobu, jakým využívá vážený odhad pravděpodobnosti k aktualizaci vah a klade nižší důraz na hodnoty, ležící mimo předpokládaný rozsah. Podle [11] tento algoritmus používá jako výstup reálné hodnoty, kde vyšší hodnota odpovídá vyšší pravděpodobnosti správné pozice, dosahuje rychlejší konvergence a je snadněji implementovatelná.

5 Předzpracování obrazu

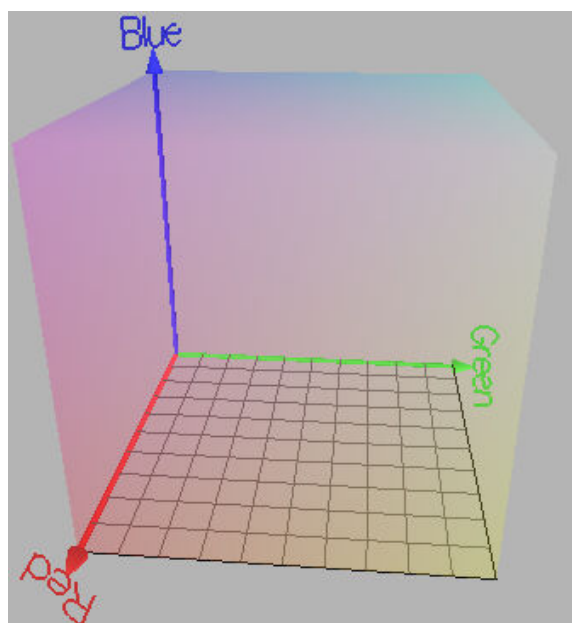
V této práci jsou využívány algoritmy, pro jejichž správnou funkci je potřeba vstupní obraz určitým způsobem pozměnit, neboť vstupní formát algoritmu nemusí být shodný se vstupem, který máme k dispozici. Jedná se o konverze mezi jednotlivými barevnými prostory, v našem případě je velice důležitá také ekvalizace histogramu.

5.1 Barevné modely

Pro různé účely byly postupem času a s vývojem analogových a digitálních technologií stanovené různé barevné modely, neboli způsob, jakým je reprezentována informace o optických vlastnostech předmětů. Výsledná subjektivní reprezentace jednotlivých barevných modelů však závisí na použitém zobrazovacím zařízení, které je použito (tiskárna, monitor, televize, apod.).

5.1.1 RGB

Nejznámějším a v počítačovém světě asi nejpoužívanějším barevným modelem je model RGB a je definován třemi barevnými složkami R (Red = červená), G (Green = zelená) a B (Blue = modrá). Jednotlivé barvy v daném modelu je možné reprezentovat krychlí v trojrozměrném souřadnicovém systému, na níž každá barevná složka tvoří jednu osu.



Obr.5.2: Barevný prostor modelu RGB (vygenerováno programem ColorSpace)

5.1.2 Šedotónový obraz

Pomocí barevného modelu RGB je možné definovat i barevný prostor, který je v rámci této práce nejvyužívanější pro použité algoritmy. Tento model (BW = Black and White) je definován pouze jasovou složkou obrazu, která má prakticky rozsah 0-255 v celých číslech.



Obr.5.3: Šedotónová stupnice

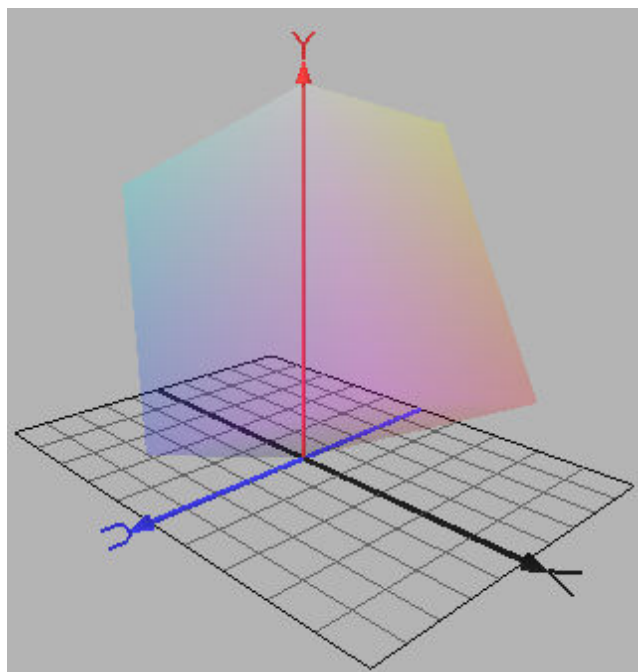
Pomocí barevných složek RGB lze jasovou intenzitu I odvodit, jako:

$$I = 0,299 * R + 0,587 * G + 0,114 * B \quad (5.1)$$

Jak je vidět z výše uvedeného vzorce, tak nejvíce jasové informace přenáší zelená barva.

5.1.3 YUV

Tento barevný model uvádím zejména kvůli častému použití ve webkamerách a pro další zpracování obrazu bývá nutná konverze do RGB. V případě digitálního signálu bývá též nazýván, jako YCbCr a v případě analogového YPbPr. Jednotlivé barvy je taktéž možné vyjádřit pomocí krychle.



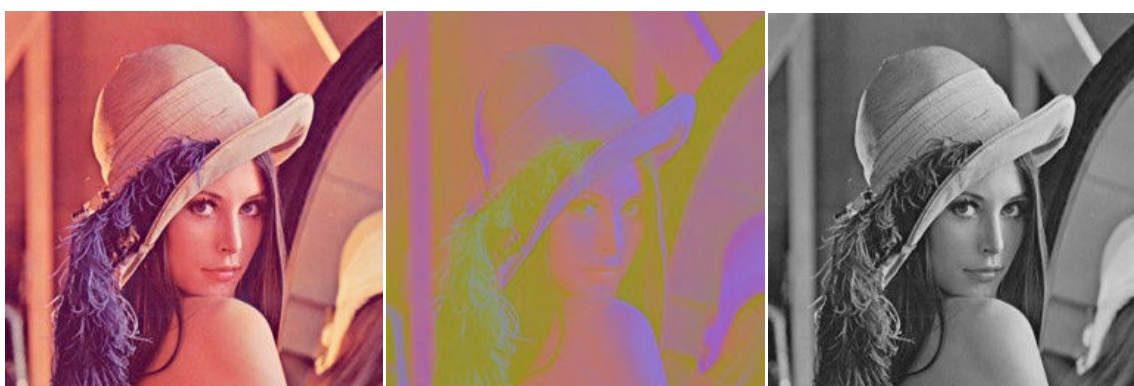
Obr.5.4: Barevný prostor YUV (vygenerováno programem ColorSpace)

Je také možné definovat vazbu na model RGB, která se využívá při vzájemných převodech.

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,100 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (5.2)$$

A naopak

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1,137 \\ 1 & -0,397 & -0,580 \\ 1 & 2,034 & 0 \end{pmatrix} * \begin{pmatrix} Y \\ U \\ V \end{pmatrix} \quad (5.3)$$



Obr.5.5: Porovnání obrazu v interpretaci různých modelů (po řádcích): RGB, YCbCr, BW

5.2 Ekvalizace histogramu

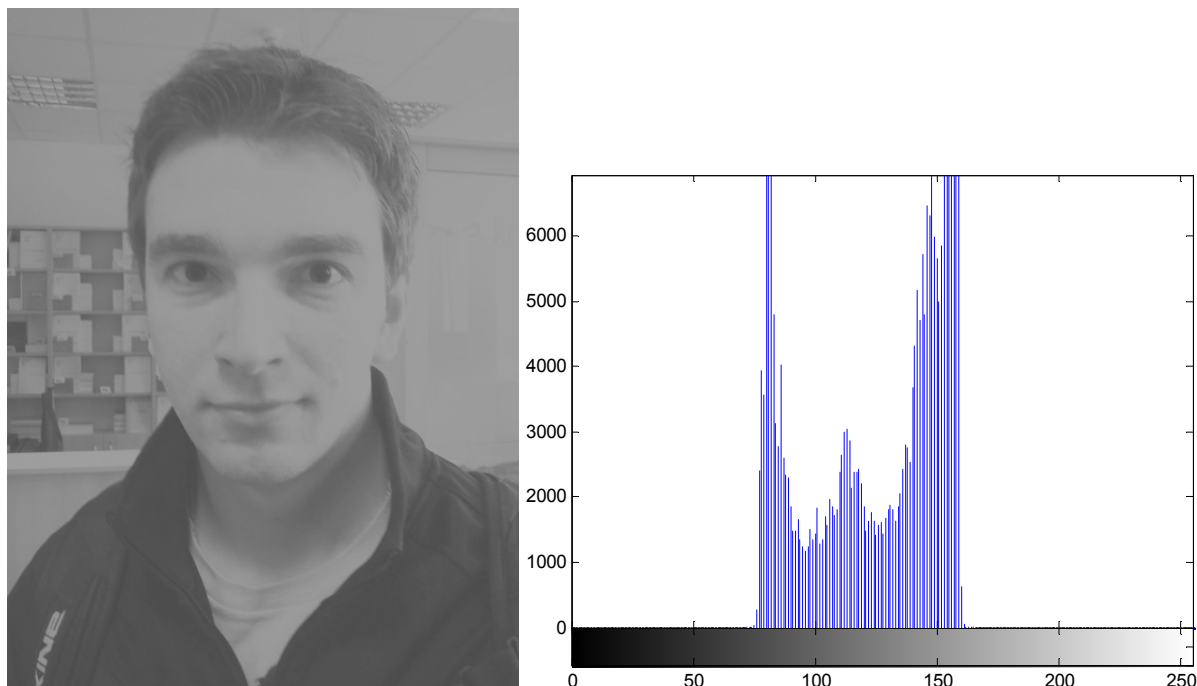
Jedná se o mechanismus, který je v rámci této práce použit, jako prostředek, přispívající ke zvýšení odolnosti celého řetězce zpracování obrazu vůči světelným podmínkám.

Pomocí rovnoměrnějšího využití celé jasové stupnice dojde k pozorovatelnému optimálnímu zvýšení kontrastu šedotónového obrazu.

Máme-li šedotónový rastrový obraz, kde n_i je počet výskytů úrovně šedé i , pak pravděpodobnost výskytu pixelu s i -tou úrovní šedé je $p(x_i) = \frac{n_i}{n}$, $i = 0, \dots, L-1$, kde n je celkový počet bodů v obraze a L je počet různých úrovní šedé (v našem případě 256). Součet všech hodnot pravděpodobnosti v takto definovaném histogramu je tak $\sum_{i=0}^N p(x_i) = 1$.

Histogram je tedy odhadem hustoty pravděpodobnosti, že pixel bude mít určitou jasovou hodnotu.

Následující obrázek ukazuje nekонтastní, přexponovaný obrázek.



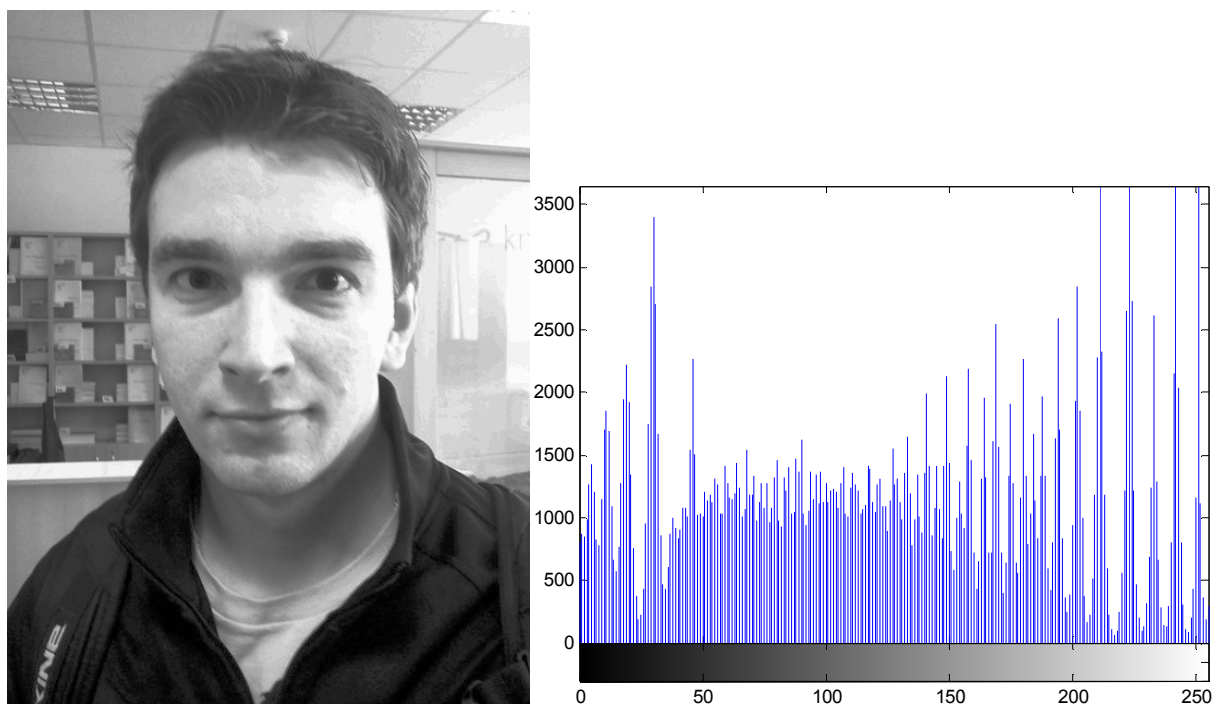
Obr.5.6: Histogram šedotónového obrazu

Z obrázku je patrné, že jsou využity jasové hodnoty jen ve velmi malém rozsahu uprostřed stupnice šedých tónů.

Úkolem tohoto algoritmu je dosáhnout toho, aby se jednotlivé odstíny šedi (obecně jakékoliv barvy) vyskytovaly v obraze s přibližně stejnou četností.

Cílem ekvalizace je však dosáhnout toho, aby byla využita celá jasová stupnice a pravděpodobnosti výskytu jednotlivých hodnot byly přibližně stejné, tedy aby histogram byl co nejvíce rovný. Vstupní hodnoty pixelů jsou mapovány na výstupní pomocí transformační

$$\text{funkce } s_k = \sum_{i=0}^N \frac{n_i}{n}, \quad i = 0, 1, 2, \dots, L-1.$$



Obr.5.7: Obrázek po ekvalizaci a jeho histogram

Je vidět, že po ekvalizaci se stal obrázek podstatně kontrastnějším. Sice na přesvícené ploše došlo k nepatrné ztrátě informace, nicméně jelikož detekce významných bodů probíhá zejména v oblastech, kde se jas mění s vysokou strmostí, tak tato skutečnost není kritická.

Ekvalizaci je možné obecně provést i pro vícekanálový obrázek. V případě modelu RGB se provádí pro každý kanál zvlášť. Tato metoda je poměrně jednoduchá i na hradwarovou implementaci a bývá využívána i v různých zařízeních, pracujících v reálném čase. Například některé fotoaparáty, nebo videokamery.

6. Programová realizace objektového detektoru Viola-Jones

6.1 OpenCV:

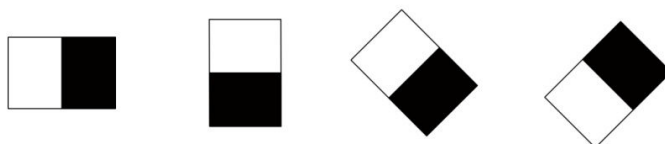
OpenCV je zkratka z anglického Open Source Computer Vision. Jedná se o rozsáhlý soubor funkcí a knihoven pro počítačové vidění, původně vyvíjený firmou Intel a v současnosti poskytuje podporu laboratoř Willow Garage, zabývající se robotikou a tvorbou aplikací. Knihovny OpenCV jsou nezávislé na platformě a dají se využít pro jazyky C++, Java, Python a další. V této práci se budeme nadále zabývat pouze spoluprací s jazykem C++. OpenCV obsahuje mimo jiné také funkce pro podporu objektového detektoru.

6.2 Objektový detektor pomocí OpenCV

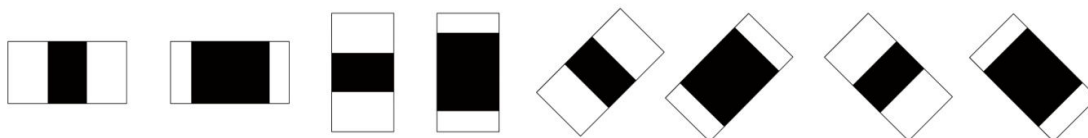
Tento detektor byl podle prvotně navrhnut Paulem Violou a rozšířen Rainerem Lienhartem o další typy diagonálních příznaků včetně ekvalizace histogramu jasové složky podle[2] pro dosažení vyšší robustnosti vůči změnám světelných podmínek.

OpenCV pracuje s rozšířenou sadou Haarových příznaků:

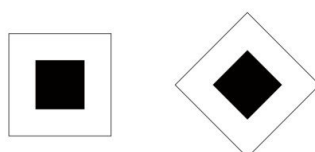
- a) Hranové příznaky:



- b) Čárové příznaky:



- c) Středové příznaky:



Obr.6.1 a,b,c: Rozšířená sada Haarových příznaků

Hodnota příznaku se vypočítá tak, že v místě přiložení příznaku na obraz se vypočítá suma všech pixelů pod bílou částí, suma pixelů pod černou částí a tyto sumy se od sebe odečtou. Tento proces se neprovádí na celý obraz zároveň, ale v rámci posuvných podoken. Pro zjednodušení a zefektivnění výpočtu je uvažována integrální reprezentace obrazu.

Nejdříve je klasifikátor (resp. kaskáda klasifikátorů, pracujících s Haarovými příznaky) trénován několika stovkami pozitivních vzorů určitého objektu, který chceme sledovat (např.: obličej, oko, auto, číslo). Všechny mají stejnou velikost 20x20 pixelů. Poté trénování probíhá i s negativními vzory, libovolnými obrázky stejné velikosti.

Po natrénování může být klasifikátor aplikován na oblast našeho zájmu (stejně velikosti jako při trénování) ve vstupním obraze. Výstup klasifikátoru je logická 1 v případě podobnosti oblasti s hledaným objektem (obličej, atd.), nebo 0 při neshodě.

Pro vyhledávání objektu v celém obraze můžeme přesouvat vyhledávací okno v obraze a kontrolovat každou oblast pomocí klasifikátoru. Hodnotící mechanismus byl navrhnut tak, aby mohlo být vyhledávací okno snadno zvětšeno, či zmenšeno, což je podstatně efektivnější než změna velikosti vstupního obrazu. Pro detekci objektu neznámé velikosti je tedy nutné proceduru provést vícekrát v různých měřících. Pro potlačení mnohonásobných pozitivních detekcí je podle [11] vhodné dále začlenit za samotný proces detekce shlukovou analýzu pracující s Eukleidovskou vzdáleností pozic a velikostí jednotlivých oblastí.

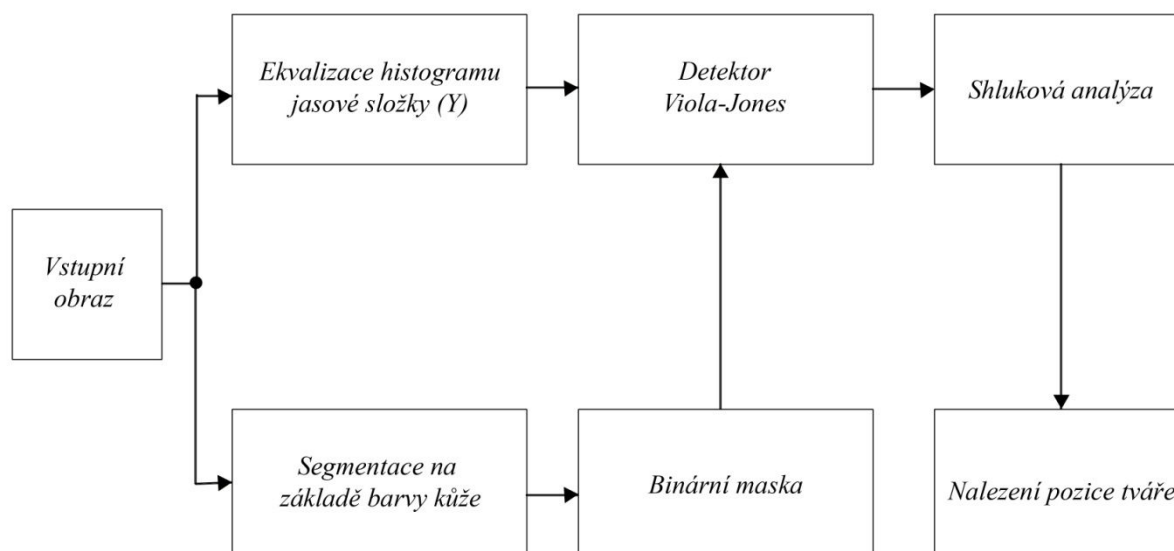
Pro snížení velikosti vstupního datového toku a zvýšení rychlosti detekce, je možné vstupní obraz podrobit předzpracování pomocí detektoru barvy lidské kůže, kde barva lidské kůže je explicitně vyjádřena rovnicí:

$$b_{skin}(x, y) = (77 < Cb(x, y) < 127) \wedge (133 < Cr(x, y) < 173), \quad (6.1)$$

kde Cb a Cr jsou chrominační složky obrazu v barevném modelu YcbCr a b_{skin} je binární maska barvy lidské kůže.

Po této proceduře pak již není prohledáván celý vstupní obraz, ale pouze oblasti, vyhodnocené podle barvy, jako lidská kůže. Jelikož detektor Viola-Jones pracuje se šedotónovými obrazy, omezí se tím také množství chybných detekcí na pozadí (majících odlišnou barvu).

Výsledný blokový diagram detekce poté bude podle [11] vypadat následovně:



Obr. 6.2: Blokové schéma obličejového detektoru

Pro detekci obličejových částí, jako oči, nos a ústa lze použít tento detektor velice obdobně, pouze je použit jiný soubor trénovacích dat, obsahujících pouze příslušnou část obličeje.

Při detekci očí je možné omezit obdobným způsobem vyhledávací barevný prostor vytvořením *barevné oční mapy* podle [10] pomocí chrominačních složek Cb a Cr :

$$EM(x, y) = \sqrt{\frac{1}{3} \cdot Cb^2(x, y) + (255 - Cr(x, y))^2} + \frac{Cb(x, y)}{Cr(x, y)} \quad (6.2)$$

Mapa je posléze ještě prahována na základě hodnoty jejího histogramu, kde distribuční funkce tohoto histogramu je rovna 90%. Tím se sníží výpočetní nároky, neboť je již pro polohu očí prohledávána již velmi malá část obličeje.

Tímto způsobem je možné vyhledávat takřka jakýkoliv objekt v obraze za použití příslušným způsobem natrénované klasifikátorové kaskády.

6.3 Popis aplikace Facerec:

K tématu byla vytvořena aplikace, založená na knihovnách OpenCV a demonstrující funkce objektového detektoru. Tato aplikace je schopna detekce jak ze statického obrazu, tak i z pohyblivého obrazu, jehož vstupem je webkamera. Implementováno bylo rozpoznávání

obličeje, zvláště pravého a levého oka, páru očí úst a nosu. Pro každý z těchto detektorů byla použita speciální klasifikátorová kaskáda, která je již součástí OpenCV. Pro přehlednost je možné jednotlivé detektory libovolně vypínat a zapínat pomocí klávesnice, aktivace a deaktivace probíhá klávesami f – pro obličejový detektor, r – pro detektor pravého oka, l – detektor levého oka, p – pár očí, m – ústa a n – nos. Ve stavovém okně se taktéž zobrazuje doba, potřebná pro detekci jednotlivých entit.

Výsledky detekce pomocí objektového detektoru pomocí OpenCV:

Pro přesnost uvádím, že testování bylo prováděno na Notebooku Fujitsu Siemens AMILO Pa1510 s Windows XP + SP3, procesor AMD Mobile Sempron 3200+, taktovací frekvence 1600MHz, 2GB RAM. Byly použity knihovny OpenCV verze 2.1, jako vývojové prostředí a kompilátor byl použit program Microsoft Visual Studio 2008. Jako zdroj obrazu byla použita webkamera Genius Slim 2020 AF.

Při testování jednotlivých detektorů bylo zjištěno, že u obličejového detektoru dochází jen k velmi malému množství chybných detekcí. Nejvyšší chybovost byla zaznamenána při použití detektoru úst. Zde docházelo k chybným detekcím zejména v oblasti očí a krku, při použití očního detektoru obzvláště při nepříznivých světelných podmínkách, nebylo možné správně určit, zda se jedná o levé, či pravé oko. Stejně tak docházelo k chybným detekcím očí na pozadí. Oproti tomu při vyhledávání očního páru k chybám téměř nedocházelo, pouze tento detektor není tak robustní vůči změně velikosti a při vzdálení již přestával pracovat, stejně jako při větším úhlu natočení hlavy. Mnohonásobné, či chybné detekce bylo možné částečně eliminovat pomocí nastavení shlukové analýzy.

Orientační doby detekce obličejových entit:

Byly testovány jednotlivé detektory zvláště na uvedené konfiguraci počítače při velikosti vstupního obrazu 320 x 240 pixelů. U všech detektorů bylo použito shodné nastavení shlukové analýzy, kde pro pozitivní vyhodnocení detekce bylo nutné nalézt alespoň dva kandidáty. Při použití více detektorů naráz se detekční doby sčítaly a vykreslování výsledného obrazu se tím zpomalovalo:

Typ detektoru	Přibližná doba detekce [ms]
Detektor obličeje:	500
Detektor pravého, nebo levého oka:	200
Detektor očního páru:	40
Detektor úst:	400
Detektor nosu:	960

Tab. 6.1: Porovnání výpočetní doby jednotlivých objektových detektorů

Při detekci více objektů stejného druhu(například více obličejů v obraze) se detekční doba nezvyšovala, neboť prohledávána jsou vždy všechna podokna.

Z těchto údajů vyplývá, že detektor Viola-Jones jako takový není bez úprav schopný práce v reálném čase(uvažujeme-li snímkovou rychlost 25snímků/sekundu, neboli délka trvání jednoho snímku je 40ms). Při doplnění o omezení vyhledávacího prostoru pomocí detektoru barvy kůže a barevné mapy by se tato doba pravděpodobně rapidně snížila

7 Praktická realizace ASM

Při zpracování této práce bylo vytvořeno několik aplikací a provedeny jisté postupy pro praktickou realizaci a znázornění funkce aktivních modelů tvarů.

7.1 Vytvoření trénovacích dat pro tvorbu modelů a klasifikátorů.

Pro samotnou aplikaci ASM byly využity již existující implementace, vytvořené Ing. Jiřím Přinosilem, PhD. V prostředí MATLAB. Tato implementace pracuje s obrazovými daty o rozměrech 120 x 120 pixelů. Z tohoto důvodu musela být vytvořena i trénovací množina o těchto rozměrech.

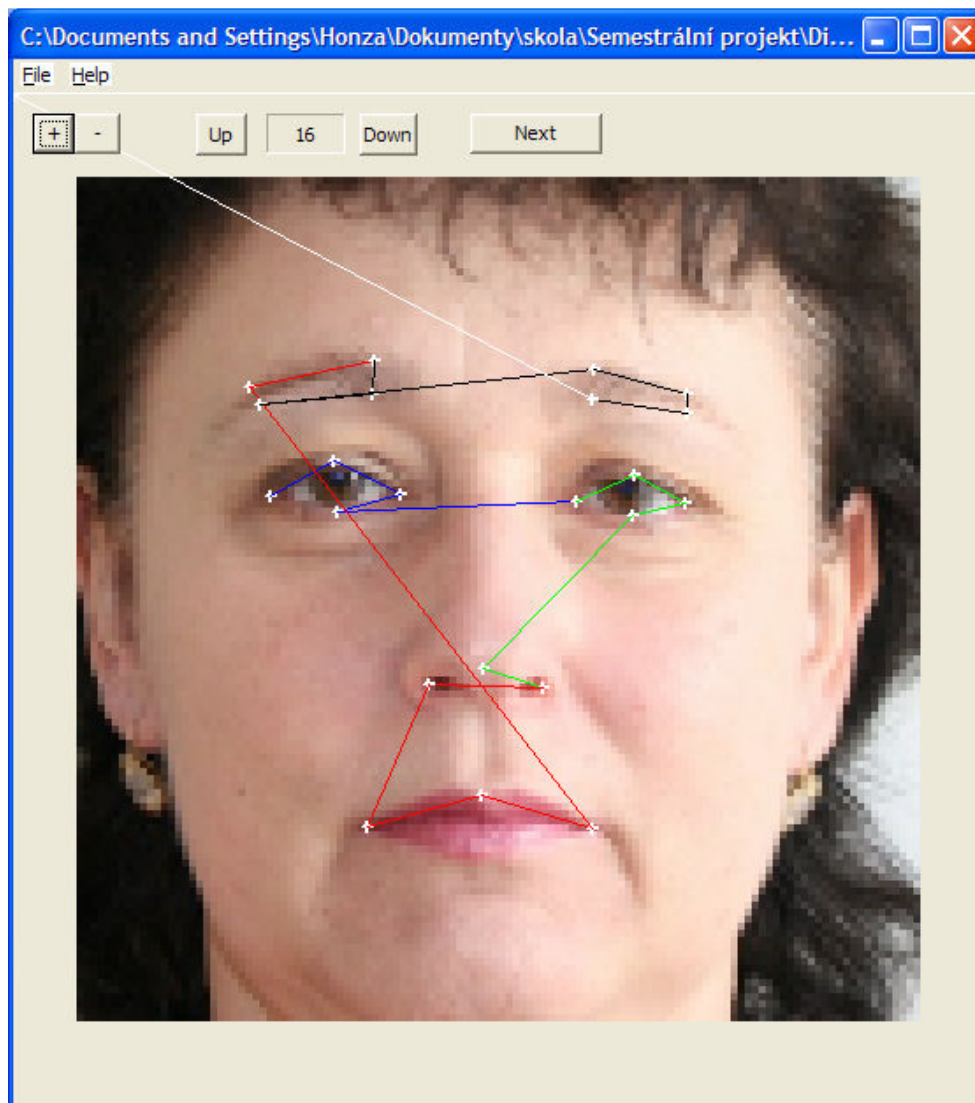
Základ trénovací množiny, čítající celkem 200 obrazových vzorů byl z velké části vytvořen z již existující databáze „*The IMM Face Database*“ (vytvořena v roce 2001 Michael M. Nordströmem, Mads Larsenem, Janusz Sierakowskim, Mikkel B. Stegmannem na z Dánské technické univerzity) a částečně z nově pořízených snímků. Tyto obrázky obsahují lidské obličeje různých lidí, při různých emociálních stavech a různých úhlech natočení hlavy ve všech osách.

Pro účel tvorby trénovací množiny jsem vytvořil jednoduchou aplikaci *ViolaExtractor*. Vstupem této aplikace může být obrázek ve formátu „*jpg*“, nebo „*bmp*“, prakticky jakékoliv velikosti (ovšem s rostoucí velikostí obrázků roste i výpočetní náročnost, viz. dále). Pomocí tohoto programu je v těchto obrázcích, obsahujících lidský obličej a za použití objektového detektoru Viola Jones detekován lidský obličej. V místě detekce je obraz oříznut na čtverec a následně zmenšen (případně zvětšen) na velikost 120 x 120 pixelů. Uložen je pak ve formátu „*bmp*“.

Ve výsledných obrázcích je poté potřeba ručně vyznačit významné body, které budou v budoucnu tvořit právě nově vytvořený aktivní modely.

Pro tuto práci byl zvolen zjednodušený model lidské tváře, skládající se celkem ze 22 bodů, a to 4 body pro každé oko (koutky očí a středy očních víček), 3 body pro nos (špička nosu a vnější okraje nosních dírek), 3 body pro ústa (ústní koutky a prostředek „propadu“ horního rtu a 4 body pro každé obočí (nejlépe vystihující tvar obočí, neboť jako takový je tvar obočí ne úplně jednoznačný a není přesně ohraničen). Tyto body byly pomocí programu Landmarks vyznačeny ve všech ze zmíněných 200 obrázcích a poloha těchto bodů byla uložena ve

formátu „.txt“ do souboru pro účely extrakce příznaků a následného samotného trénování klasifikátorů.



Obr. 7.1: Vytváření trénovací databáze

7.2 Výpočet modelů

Normalizované průměrné modely jsou vypočítány pomocí dat, získaných z trénovací množiny v předchozím bodě. Výpočet je popsán v kapitole *Aktivní modely tvaru*.

7.3 Extrakce deskriptorů a příznaků významných bodů

Jako vstup pro trénovací algoritmus je potřeba vytvořit soubor trénovacích dat. Každý vzor je transformován do šedotónového barevného prostoru a je na něj postupně aplikována již výše zmíněná sada 48 Gaborových filtrů o různých délkách a orientacích.

Pro každý z těchto 22 bodů (uvažujeme x_i, y_i), pro všechny filtrované obrazy a pro všechny vzory bylo následně vytvořeno okno v okolí bodu o souřadnicích rohů $A[x_i-6, y_i-6]$, $B[x_i+6, y_i+6]$, $C[x_i-6, y_i+6]$ a $D[x_i+6, y_i+6]$, tedy obraz o velikosti 13x13 pixelů. Jasové informace jednotlivých pixelů těchto oken byly následně uloženy do příslušných textových souborů pro každý bod zjednodušeného modelu. Tato informace bude následně využita pro trénování klasifikátorů pomocí algoritmu GentleBoost.

7.4 Trénování klasifikátorů

Pro každý z uvažovaných 22 bodů dílčího modelu byl pomocí algoritmu GentleBoost natrénován vlastní klasifikátor, obsahující v našem případě 8 až 48 příznaků pro každý bod. Z hlediska počtu klasifikátorů, je nejsložitější je popis obočí, kde pro 4 body určující obočí nám vyšlo celkem 123 příznaků pro levé obočí a dokonce 178 pro 4 body určující pravé obočí. Pro popis 4 bodů, určujících tvar levého oka je to 65, pro pravé oko 62, pro 3 body určující tvar nosu 50 a pro 3 body tvaru úst celkem 42 příznaků.

Stojí za zmínku, že trénování klasifikátorů je výpočetně velice náročný proces. Klasifikátory byly trénovány na požadovanou přesnost 93%. Tato hodnota byla zvolena experimentálně vzhledem k prakticky dosažitelné přesnosti a potřebnému výpočetnímu času. Pro výpočet byly použity 3 počítače o celkem 7 procesorových jádrech (výpočet každého klasifikátoru byl přiřazen konkrétnímu jádru) a výpočet všech 22 klasifikátorů trval zhruba 14 hodin, což vychází přibližně na 4 dny trénování při použití jednoho běžného procesorového jádra.

7.5 Postup aplikace ASM

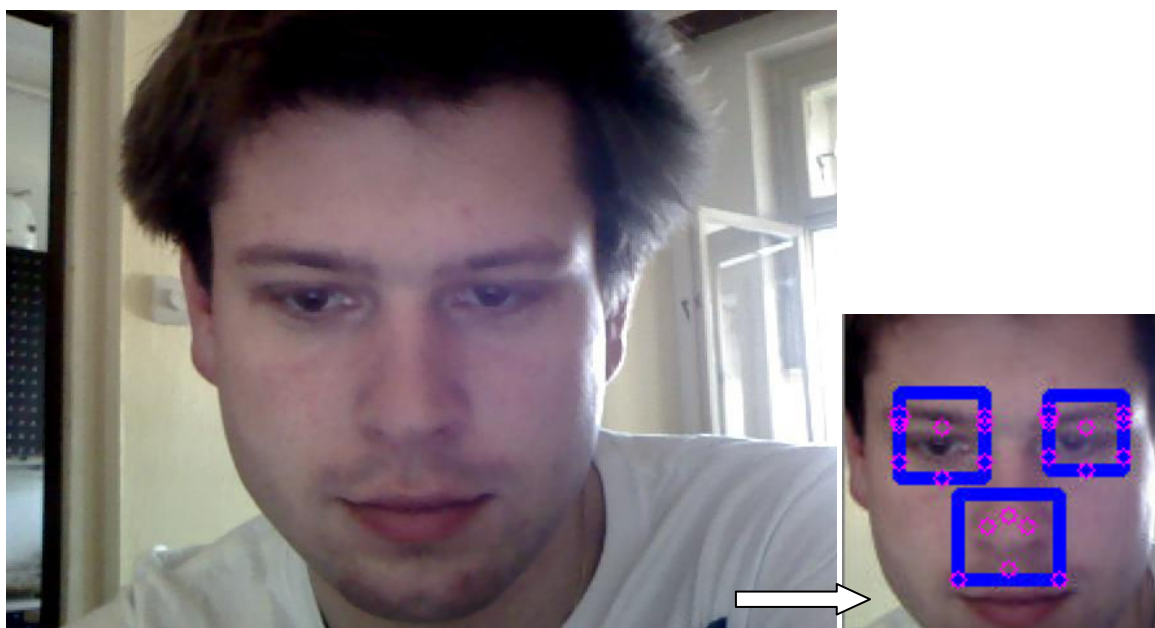
Aby bylo možné přistoupit k samotné aplikaci aktivních modelů tvaru, je potřeba nejdříve určit přibližnou polohu oblasti, kam bude model umístěn.

Pro názornost byl zvolen následující postup:

1. Ze souboru, případně po inicializaci webkamery je do paměti uložen snímek pro další zpracování.
2. Tento snímek pro předzpracování vstupuje do programu *ViolaLabeler*, který byl pro tento účel vytvořen. V tomto programu pomocí objektového detektoru Viola-Jones dojde k nalezení tváře, k následnému oříznutí obrazu podle nalezené obličejové oblasti a změně velikosti na

rozměr 120 x 120 pixelů kvůli dalšímu zpracování. Pro vyšší robustnost vůči světelným podmínkám zde byla použita technika ekvalizace histogramu. Jelikož detektor Viola-Jones pracuje se šedotónovými obrazy, je tato operace poměrně efektivní a žádoucí.

3. V oříznutém snímku je pomocí dalších dílčích detektorů nalezena poloha obou očí a nosu. Podle souřadnic nalezených oblastí je pak vypočítána přibližná poloha všech 22 bodů. Poloha obočí je vždy vypočítána podle polohy očí a poloha úst podle polohy nosu. Tato kombinace byla zvolena jako nejvhodnější z hlediska chybovosti.



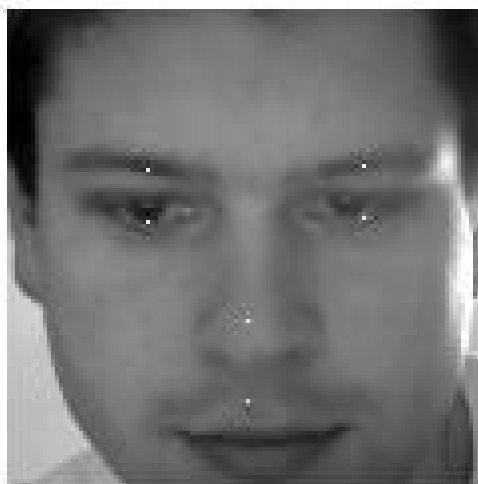
Obr. 7.2: Předzpracování vstupního obrázku programem *ViolaLabeler*

V obrázku je čtvercem označená oblast, která byla objektovým detektorem Viola-Jones vyhodnocena, jako oči a nos. Kružnice pak znázorňují přibližně vypočítanou polohu 22 bodů, tvořících obličejový model. Tato poloha není přesná a byla určena experimentálně vzhledem k výstupu detektoru a variabilitě vzhledu lidské tváře, nicméně se ukázala jako dostatečně přesná pro následné umístění aktivních modelů tvaru. Za zmínku stojí jistě i fakt, že samotná tvář byla přesně nalezena na různorodém pozadí a za špatného osvětlení. Bohužel se ukázalo, že při špatných světelných podmínkách (například jedna část obličeje je výrazně přesvícená oproti druhé části a je tedy málo kontrastní a bez kresby) není spolehlivost detektorů uspokojivá. Přesněji je například nalezena poloha tváře, ale není například možné nalézt jedno, či obě oči, popřípadě nos. V takovémto případě již není možné přistoupit k bodu aplikace ASM, neboť chybí přibližné vstupní určení polohy jednotlivých částí. Jestliže volíme

jako vstupní signál obraz z webkamery, je po nevykonané detekci uložen další snímek a vstupuje zpět do bodu 2.

Jestliže detekce obou očí i nosu (tedy i dopočet všech 22 bodů) proběhla bezchybně, je možné přistoupit již k samotnému umístění modelu.

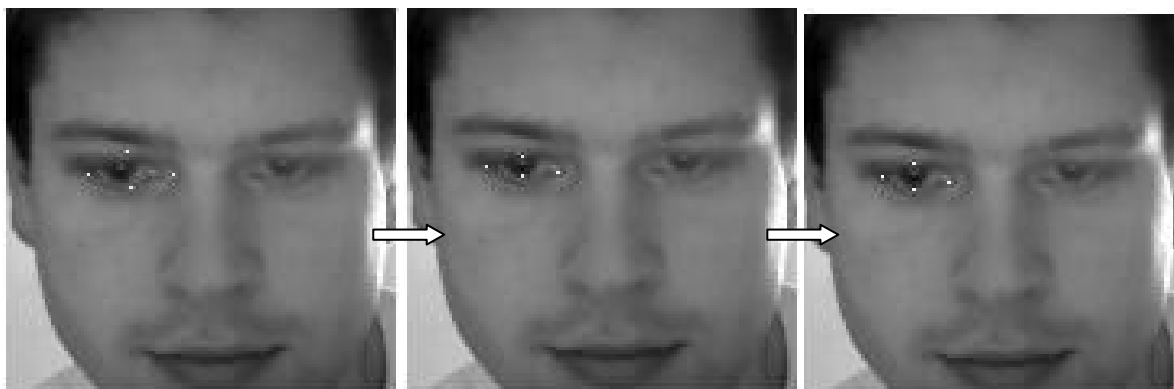
Nejdříve je pro každý z šesti dílčích modelů vypočítáno předpokládané těžiště modelů podle předpokládané polohy jednotlivých bodů, získaných v předchozím bodě a do paměti jsou načteny též průměrné modely, vypočítané z dat trénovací množiny podle výše uvedeného postupu. Je také vypočten obecný, neboli centrální model, definující umístění dílčích modelů podle jejich těžišť. Tento centrální model je definován vždy tolika body, kolik je dílčích modelů.



Obr. 7.3: Umístění centrálního modelu do oříznutého obrazu

Pomocí příslušných deskriptorů jsou pak pro každý dílčí model detekovány pozice nejlépe vyhovujících jasových vzorů. Samotná detekce probíhá pouze v nejbližším okolí daných bodů, neboť se podle [11] předpokládá, že v okolí správné pozice tohoto bodu se bude vykytovat více hodnot s vyšší pravděpodobností. Je tak zvolena pozice s nejvyšší akumulovanou hodnotou hodnot sousedních za využití plovoucího Gaussova okna.

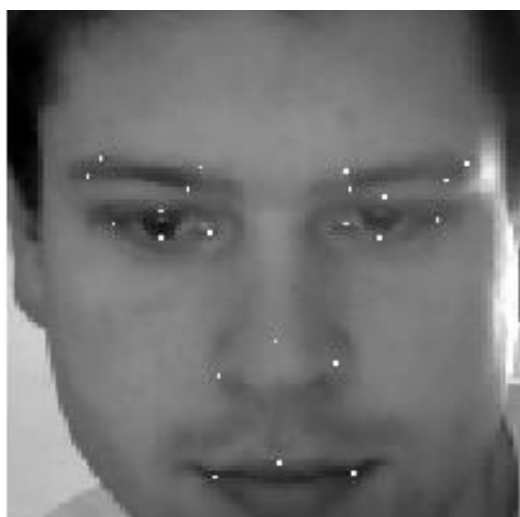
Následně jsou v obraze opět upraveny pozice nových bodů podle vzájemných pozičních vztahu aktivních modelů tvaru a opakuje se vyhledávání příslušných jasových vzorů. Tento postup se opakuje v několika iteracích, dokud nedojde k požadovanému stupni konvergence, což znamená, že se nově určované pozice bodů modelu nadále nebudou již příliš měnit. Na obr. je znázorněn zkrácený postup několika iterací uchycení modelu oka.



Obr. 7.4: Počáteční umístění očního modelu, druhá iterace, konečné umístění

Tímto způsobem je potřeba nalézt přesnou pozici bodů popisujících tvar pro všechny dílčí modely.

Na dalším obrázku je již vidět rozmístění všech 22 bodů modelu na oříznutém obrázku.



Obr. 7.5: Konečné umístění všech 22 bodů oříznutého obrázku

Pro zjednodušení výpočtů a snížení výpočetních nároků byl používán obrázek, který byl s pomocí objektového detektoru Viola-Jones oříznut pomocí aplikace *ViolaLabeler*. Informace o poloze nalezených bodů však prakticky pro další využití postrádá hlubší smysl bez kontextu polohy těchto bodů v obraze originálním. Z tohoto důvodu aplikace *ViolaLabeler* poskytuje nejen informaci o přibližných polohách hledaných bodů, ale i o poloze a velikosti výřezu v originálním obraze. Za použití těchto údajů je potom již poměrně snadno určit souřadnice nalezených bodů.

Nejdříve je nutné určit míru změny velikosti čtvercového výřezu obličejové oblasti S .

$$S = \frac{w}{120}, \quad (7.1)$$

Kde w je délka strany čtvercového výřezu obličejové oblasti a 120 je délka strany oříznutého obrázku.

Pak je již na základě informací o poloze výřezu možné vypočítat polohu bodů v původním obraze.

$$x_o = x_v \cdot S + T_x \quad (7.2)$$

$$y_o = y_v \cdot S + T_y \quad (7.3)$$

Kde x_o a y_o jsou hledané souřadnice bodů v originálním obraze, x_v a y_v jsou pozice bodů ve výřezu, S je míra změny velikosti a T_x a T_y značí, o kolik pixelů bylo výřezové okno posunuto oproti počátku souřadnicového systému.

Na následujícím obrázku je již patrný výsledek přepočtu a poloha hledaných bodů je již zaznačena v souřadnicovém systému originálního obrazu.



Obr. 7.6: Nalezené body modelu v originálním obraze

8 Programová realizace ASM

Součástí přílohy je programová realizace postupu aplikace aktivních modelů tvaru, navrhnutý v rámci této práce. Jelikož se vycházelo již z dřívějšího návrhu realizace, vytvořené Ing. Jiřím Přinosilem, PhD. v prostředí MATLAB, pro pokračování v rámci této práce bylo zvoleno stejné prostředí a v následujícím textu bude stručně popsán princip, jakým tento program pracuje. Jako vstup pro aplikaci byl zvolen výstup z webkamery.

1. Po spuštění skriptu *Run.m* jsou nastaveny potřebné údaje pro vstup webkamery, jako vstupní formát a pracovní barevný prostor. Jestliže webkamera pracuje v jiném režimu, nežli RGB (často modifikace YUV formátu, např. YUY2), je obraz do RGB konvertován. Dále je zobrazeno náhledové okno. Program počká 6 sekund na inicializaci, zaostření a další automatické nastavení kamery
2. Následně je uložen aktuální snímek s názvem *I.jpg*.
3. Tento snímek následně vstupuje do programu *ViolaLabeler*, v němž je pomocí objektového detektoru Viola-Jones nalezena lidská tvář. Takto nalezená oblast je následně oříznuta a obraz je zvětšen, či zmenšen na rozměr 120 x 120 pixelů a uložen jako *R.jpg*. Do pomocného souboru *facepos.txt* je následně uložena poloha a šířka čtvercového výřezu detekované oblasti. V tomto výřezu je poté detekována poloha obou očí a nosu a je vzhledem k výstupu z dílčích detektorů vypočítána přibližná poloha všech 22 bodů zjednodušeného modelu tváře. Jejich souřadnice jsou uloženy do pomocného souboru *R.txt*. Jestliže nějaká část není nalezena, je přesto na místo jejich souřadnic zapsána 0.
4. Následně se testuje pomocí průchodu souborem *R.txt*, zda byly nalezeny obě oči i nos, tedy jestli se v souboru nachází číslo 0. Jestliže ano, pak je uložen nový aktuální snímek a opakuje se bod 3.
5. Jestliže byly všechny tři obličejové entity nalezeny, pak jsou tyto souřadnice použity jako počáteční polohy pro algoritmus ASM. Volitelně je možné ještě použít techniku ekvalizace histogramu oříznutého obrazu.
6. Po nalezení přesné polohy všech 22 bodů jsou podle výše popsaného postupu souřadnice těchto bodů transformovány do souřadného systému originálního obrazu a je zobrazeno okno s výsledkem.

7. V programu je vytvořen cyklus, což v tomto případě znamená, že dokud není zavřeno náhledové okno, dochází k opakování bodů 2. - 6.

Zjištěné nevýhody navrženého systému:

-Navzdory použitým mechanismům není systém příliš robustní vůči velkým rozdílům osvětlení mezi jednotlivými částmi obličeje. Zpravidla tak nedojde vůbec k detekci buď samotného obličeje, nebo některé z jeho částí.

-Systém je stavěný na rozpoznání pouze jednoho obličeje v obraze. Při výskytu více obličejů je vždy vybrán jen jeden z nich.

-Občas dochází k chybnému určení polohy koutků úst, zejména při výskytu tzv. strniště na obličeji. Dochází k uchycení těchto bodů na předělu vousů a kůže. U oholených tváří se tento problém nevyskytoval.

-Systém byl na testovací konfiguraci pomalý. Proces detekce obličeje a následné finální umístění všech 22 bodů trvá několik vteřin. Tuto dobu by bylo možné překlenout například pomocí pyramidální implementace sledovacího algoritmu Lucas-Kanade, jehož vstupem by byly souřadnice 22 bodů obličejové masky, jejichž poloha by se pomocí metody optického toku sledovala. Tato metoda byla pouze otestována v implementaci knihoven OpenCV a jeví se z hlediska rychlosti a přesnosti, jako velmi efektivní.

-Občas dochází k paměťovým chybám a systém se zastavuje. Příčina však nebyla zjištěna.

-Systém by bylo pravděpodobně vhodné také v budoucnu implementovat celý v jazyce C++, zejména z důvodu výpočetní efektivity oproti prostředí MATLAB.

9 Závěr

V této práci byly popsány základní principy počítačového vidění pro rozpoznávání objektů, jakožto i základy neuronových sítí. Byl taktéž popsán objektový detektor Viola-Jones a jeho součásti – klasifikační algoritmus AdaBoost, druh reprezentace integrálního obrazu a způsob extrakce obrazových příznaků pomocí Haarových vlnek. Byla taktéž vytvořena aplikace *Facerec*, která demonstruje principy objektového detektoru Viola-Jones.

V rámci této práce byl také navrhnut a realizován postup použití aktivních modelů tvaru. Byly taktéž popsány způsoby extrakce deskriptorů a příznaků, výpočtu klasifikátorů a modelů. Rovněž byla vytvořena trénovací databáze obličejů, čítající 200 obrazů a v každém z nich bylo ručně označeno 22 bodů zjednodušeného modelu lidské tváře. Na základě takto vytvořené databáze bylo vypočteno 6 dílčích a 1 centrální model, určující jejich vzájemnou polohu a na základě vyextrahovaných příznaků vypočteny klasifikátory pro každý bod. Následně byla ověřena správnost postupu při umístění aktivních modelů tvaru do vstupního obrazu pomocí aplikace, která byla pro tuto práci realizována v prostředí MATLAB s využitím funkce objektového detektoru Viola-Jones, implementovaného za pomoci knihoven OpenCV v jazyce C++. Největší nevýhodou navrženého systému je pravděpodobně spolehlivost, zejména při nepříznivých světelných podmínkách a výpočetní náročnost, tedy i nedostatečná rychlost. Vyšší výpočetní efektivita by pravděpodobně do budoucna mohlo být dosaženo pomocí doplnění o sledovací algoritmus Lucas-Kanade.

Použitá literatura:

- [1] BARTLETT, M. S. *Face Image Analysis by Unsupervised Learning and Redundancy Reduction*. PhD thesis, University of California, san Diego. 1998.
- [2] Dokumentace projektu OpenCV, dostupná na: <http://opencv.willowgarage.com/wiki/FaceDetection>
- [3] FREUND, Yoav; SCHAPIRE, Robert E. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*. 1999.
- [4] FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. *Additive logistic regression: a statistical view of boosting*. Technical Report, 1998.
- [5] HORÁK, K.; KALOVÁ, I.; PETYOVSKÝ, P.; RICHTER, M. *Počítačové vidění*. Brno: 2007. s. 1-132.
- [6] JOLLIFFE I.T. *Principal Component Analysis*, New York, Springer Verlag, 1986.
- [7] LINDSAY, I.; SMITH, A. *Tutorial on Principal Components Analysis*, 2002.
- [8] MITCHELL Tom M. *Machine learning*. McGraw-Hill, Maidenhead, U.K., International Student Edition, 1997. ISBN: 0-07-115467-1, s. 1-414.
- [9] PŘINOSIL, J. *Analýza emocionálních stavů na základě obrazových předloh*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008.
- [10] PŘINOSIL, J.; KROLIKOWSKI, M. Využití detektoru Viola-Jones pro lokalizaci obličeje a očí v barevných obrazech. *Elektrorevue - Internetový časopis* (<http://www.elektrorevue.cz>), 2008, č. 31, s. 1-16. ISSN: 1213-1539.
- [11] PŘINOSIL, J.; MÍČA, I. Efektivní detekce významných bodů částí obličeje. *Elektrorevue - Internetový časopis* (<http://www.elektrorevue.cz>), 2010, roč. 2010, č. 57, s. 1-5. ISSN: 1213- 1539.
- [12] ŠÍMA J.; NERUDA R. *Teoretické otázky neuronových sítí*. Matfyzpress, Praha 1997.

[13] VIOLA P.; JONES M.. Rapid object detection using a boosted cascade of simple features. *In Proc. CVPR*, 2001, s. 511–518.

[14] VIOLA P.; JONES M. Robust real-time object detection. *Intl. J. Computer Vision*, 2004, s. 137–154.

Seznam použitých zkratk

AAM	Aktivní model vzhledu (Active Appearance Model)
ASM	Aktivní model tvaru (Active Shape Model)
BMP	Rastrový, nekomprimovaný grafický formát (BitMaP)
BW	Zde označení pro šedotónový barevný prostor (Black and White)
JPG, JPEG	Barevný komprimovaný grafický formát (Joint Photographic Experts Group)
OpenCV	Soubor knihoven pro zpracování obrazu (Open Source Computer Vision)
PCA	Analýza hlavních komponent (Principal Component Analysis)
RGB	Barevný model reprezentující aditivní míchání červené (Red) , zelené (Green) a modré (Blue)
YCbCr	Barevný model skládající se z jasové složky Y, modré a červené chrominační složky Cb a Cr
YUV	viz. YCbCr