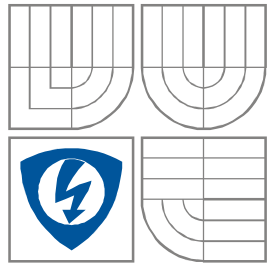


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ROZPOZNÁVÁNÍ VÝRAZU TVÁŘE U NEZNÁMÝCH OSOB

FACIAL FEATURES RECOGNITION OF UNKNOWN PERSONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAL BARTONČÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. KAREL HORÁK, Ph.D

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Michal Bartončík
Ročník: 2

ID: 73094
Akademický rok: 2010/2011

NÁZEV TÉMATU:

Rozpoznávání výrazu tváře u neznámých osob

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a implementovat algoritmy pro rozpoznávání výrazu tváře jako je např. smích v živém videu. Posloupnost rozpoznání výrazu tváře je dána čtyřmi základními bloky, jež je třeba komplexně vyřešit. Jde o bloky detekce barvy kůže, identifikace obličejové části, detekce obličejových rysů a samotné rozpoznávání výrazu tváře. Navržené metody musí být implementovány pro použití v aplikacích reálného času.

DOPORUČENÁ LITERATURA:

- [1] VERNON, David. Machine Vision : Automated Visual Inspection and Robot Vision. Hemel Hempstead : Prentice Hall International (UK) Ltd., 1991. 260 p. ISBN 0-13-543398-3.
- [2] SONKA, Milan, HLAVAC, Vaclav, BOYLE, Roger. Image Processing, Analysis and Machine Vision. 3rd edition. Toronto : Thomson, 2008. 829 p. ISBN 978-0-495-08252-1.
- [3] RUSS, J.C. The Image Processing Handbook. Boca Raton : CRC Press, 1995. 674 p. ISBN 0-8493-2516-1.

Termín zadání: 7.2.2011

Termín odevzdání: 23.5.2011

Vedoucí práce: Ing. Karel Horák, Ph.D.

prof. Ing. Pavel Jura, 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.

Abstrakt

Tato práce popisuje jednotlivé části a fáze pro vyhledávání a rozpoznání výrazu tváře u neznámých osob. Jsou zde uvedeny i možná řešení a metody řešící jednotlivé fáze projektu. Moje diplomová práce je zaměřena na rozpoznávání výrazu tváře u neznámých osob.

Pro tuto práci mi byla zapůjčena průmyslová videokamera, počítač a místo v laboratoři. Dále se seznámíme s barevnými prostory a jejich využitím. Z hlavních zastupitelů vybere ten nejvhodnější za pomoci využití programu MatLab a navrženého algoritmu.

Po nalezení vhodného barevného prostoru vysegmentujeme barvu kůže v obraze. Kůže obepíná ovšem celé tělo a tak je zapotřebí nalézt, na oddělených částech obrazu zastupující barvu kůže, hledaný obličej.

Po nalezení obličeje je zapotřebí nalézt významné body pro identifikaci deformace obličeje a následného výrazu tváře. Definujeme zde i samotné svalové posuny při různých výrazech.

Klíčová slova

Výraz obličeje, Významný bod, Segmentace, Kůže, Obličej, Detektor

Abstract

This paper describes the various components and phases of the search and recognition of facial expressions of unknown persons. They are presented here as well as possible solutions and methods of addressing each phase of the project. My master's thesis is designed to recognize facial expressions of unknown persons.

For this thesis, I was lent industrial video camera, computer, and place in a laboratory. Furthermore, we introduce the color spaces and their use. From the lead representatives selects the most appropriate assistance for the use of Matlab and the proposed algorithm.

After finding a suitable color space segments skin color in the image. The skin, however, surrounds the entire body and so need to be found, the separated parts of the image representing the color of skin, a face.

Once you find a face is needed to find relevant points for the identification subsequent deformation to definition of facial expressions. We define here the actual muscle movements in different expressions.

Keywords

Facial expression, Keypoints, Segmentation, Skin, Face, Detector

Bibliografická citace:

BARTONČÍK, M. *Rozpoznávání výrazu tváře u neznámých osob*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 68s. Vedoucí diplomové práce byl Ing. Karel Horák, Ph.D..

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Rozpoznávání výrazu tváře u neznámých osob 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 této diplomové práce 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **23. května 2011**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Karlu Horákovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **23. května 2011**

.....
podpis autora

Obsah

1	Úvod.....	10
2	Snímací aparatura.....	12
2.1	Počítač.....	12
2.2	Kamera.....	12
2.3	Objektiv.....	13
3	Barevné prostory pro detekci kůže.....	14
3.1	Model RGB.....	14
3.2	Model HSV.....	15
3.3	Model HLS.....	15
3.4	Model YUV.....	16
3.5	Model YIQ.....	17
3.6	Model YCbCr.....	17
3.7	Model Lab.....	18
3.8	Model Luv.....	18
4	Segmentace modelu kůže.....	19
4.1	Výpočet reprezentantů barvy kůže.....	20
4.2	Hustota reprezentantů modelu kůže.....	21
4.3	Geometrické těleso reprezentující lidskou kůži.....	22
4.4	Klasifikace kůže.....	23
5	Nalezení obličejové části.....	26
5.1	Teoretické rozdělení.....	26
5.2	Implementace vyhledání obličeje.....	27
5.2.1	Boosting.....	28
5.2.2	LogitBoost.....	28
5.2.3	Adaboost.....	29
5.3	Metoda Haarových příznaků.....	31
6	Významné body na obličeji.....	34
6.1	Detektor MORAVEC.....	34
6.2	Detektor HARRIS.....	36
6.3	Transformace SIFT.....	40
6.3.1	Vytvoření měřítkového prostoru a vyhledání lokálních extrémů.....	41
6.3.2	Přesnější lokalizace klíčových bodů.....	42
6.3.3	Přiřazování orientací.....	42

6.3.4	Tvorba deskriptorů	43
6.3.5	Zhodnocení.....	43
6.4	Metoda SURF	44
6.4.1	Měřítkový prostor.....	44
6.4.2	Fast Hessian.....	44
6.4.3	Přiřazování orientací.....	45
6.4.4	Tvorba deskriptorů	46
6.4.5	Zhodnocení.....	46
7	Rozpoznání výrazu tváře	47
7.1	Způsoby detekce obličejových rysů.....	47
7.1.1	AAM metoda.....	47
7.1.2	Model SVM.....	49
7.2	Metody rozpoznávání výrazu tváře	50
7.2.1	Nejjednodušší metody rozpoznávání výrazu	50
7.2.2	Rozpoznávání za pomoci masek	51
7.2.3	Rozpoznávání za pomoci neuronových sítí.....	52
8	Experiment	52
8.1	Postup	52
8.1.1	Omezení významných bodů pro definici výrazu tváře.....	53
8.1.2	Symetrie	54
8.1.3	Definice výrazů	57
8.2	Program	58
8.3	Výsledky.....	61
9	Závěr.....	63

1 ÚVOD

Automatická detekce obličejů z obrázku nebo z videosekvence spojená také s rozpoznáváním je dnes již velmi rozšířená v mnoha aplikacích. Detekce a rozpoznávání obličejů má svůj význam všude tam, kde je třeba rozpoznávat identitu osob, např. v kriminalistice, kde hraje důležitou roli v identifikaci pohřešovaných osob, případně k usvědčení osob z trestného činu. Dokonce i někteří výrobci běžných počítačů umožňují přihlašování do operačního systému na základě rozpoznávání obličejů. Další z možných oblastí využití jsou inteligentní roboti, bezpečnostní kamery apod. Musíme si ale uvědomit, že obličej může být mírně deformovaný díky aktuálnímu stavu uživatele. Samotné rozpoznávání výrazu tváře se může použít právě k určení deformace, pro internetové sociální sítě pro aktualizaci stavů nebo pro statistické či informační účely.

Tato práce popisuje jednotlivé části a fáze pro vyhledávání a rozpoznání výrazu tváře u neznámých osob. Jsou zde uvedeny i možná řešení a metody řešící jednotlivé fáze. Moje práce je zaměřena na rozpoznávání výrazu tváře u neznámých osob. Pro pochopení následujícího dokumentu si stručně popíšeme jednotlivé kapitoly. Pro můj projekt mi byla zapůjčena průmyslová videokamera, počítač a laboratoř. U výrazu tváře je zapotřebí rozeznávat jednotlivé deformace na tváři a proto mi byla doporučena právě průmyslová videokamera. Celá aparatura je popsána v kapitole dva.

Ve třetí kapitole se seznámíme s jednotlivými barevnými prostory a jejich využitím. V našem algoritmu jsme chtěli vyhledávat napřed barvu kůže pro rychlejší se zpracování obrazu. V prvním kroku tedy vysegmentujeme barvu kůže a s ostatními daty obrazu už nepotřebujeme pracovat. Za tímto účelem je ale zapotřebí vybrat vhodný barevný prostor. Je vyvinuto totiž mnoho různých barevných prostorů pro různé účely a s různými výhodami a zpracování. Cílem barevných modelů je co nejlépe modelovat skutečnou barvu pomocí určitých základních barev a jejich míchání.

V další kapitole rozebíráme už samu segmentaci barvy kůže. Kůže je sama o sobě dosti nejednoznačná. Člověk může být bledý, opálený nebo něco mezi a stále se bude jednat o toho samého člověka. Rozdíly jsou i v národech a jejich pigmentu. Afričané a jejich potomci mají tmavší pleť a lidé pocházející z Asie naopak světlejší. Už sama o sobě není kůže homogenní a obsahuje několik různých barev, které se v lidském oku skombinují a vytvoří v mozku odezvu odpovídající barvy. Obdobný princip funguje u televizoru, kde máme jednotlivé barevné body, které nám podle jasu v mozku vytvoří požadovanou barvu. V případě kůže jsou ale tyto body mikroskopické. Změny barvy kůže mohou nastat i při samotném snímání díky rozdílným osvětlením a nastavení snímací aparatury. Proto jsme si vytvořili galerii, ze které jsem si oddělil vzorky kůže a za jejich pomocí extrahoval samotný model kůže. Tento model kůže představuje geometrické matematicky popsatelné těleso reprezentující co možná všechny barvy kůže.

Když máme nalezenou a vysegmentovanou barvu kůže, nalezneme kde v obraze se vyskytuje hledaný obličej. Řešení tohoto problému jsme popsali v kapitole pět. Využili jsme zde velmi účinný a používaný způsob vyhledávání obličeje za pomoci Boosting algoritmů a Haarových příznaků.

V šestá kapitola je zaměřena na vyhledávání významných bodů. Významné body se využívají ve velkém měřítku od sledování pohybu nebo 3D modelování až k tvorbě panoramatických fotografií nebo rozpoznávání objektů. Seznámíme se zde se základními detektory významných bodů jako je Moravcův nebo Harrisův detektor, ale také novější detektory. Jako nejlepší metoda s ohledem na přesnost a korespondenci bodů je transformace SIFT, která má velmi dobré výsledky, ale je dosti náročná na čas zpracování. Tento nedostatek se snaží opravit metoda SURF.

Po nalezení významných bodů je zapotřebí body omezit pouze na ty důležité pro naši identifikaci výrazu tváře. Způsoby tohoto omezení jsou popsány v následující kapitole i s definicemi a možnými metodami rozpoznávání výrazu tváře.

V předposlední kapitole popisujeme už náš zaměřený na rozpoznávání výrazu tváře pro aplikace pracující v reálném čase. Je tady zapotřebí brát ohled na rychlost zpracování a na dostatečné kvalitní výsledky hodnocení. Výrazy ovšem nejsou pevně dané u všech osob na planetě a tak je zapotřebí nalézt určitý algoritmus nebo rozpoznávací charakteristiky pro každý požadovaný výraz obličeje.

2 SNÍMACÍ APARATURA

Pro sestavení své aparatury jsem dostal místo v laboratoři E617. Pro jednodušší nastavení by bylo vhodnější vybrat místo které by nebylo proti oknu ale tuhle menší nevýhodu můžeme upravit nastavením vyvážení bíle u kamery.

2.1 Počítač

Aby byl program kompaktní využil jsem stolní počítač zapůjčený od vedoucího jehož konfigurace je:

- Windows XP Profesional SP2
- Celeron CPU 2,80 GHz
- Operační paměť 480 MB RAM
- Grafická karta Intel 82865G - 64MB
- zapojení USB

2.2 Kamera

Dále jsem musel vybrat kameru. Rozpoznávání výrazu tváře závisí na rozlišení kamery. Jelikož chceme nejen nalézt obličej a také detekovat v obličejí významné body pro identifikaci výrazu tváře. Empiricky bylo zjištěno že pro rozeznání obličej na fotografii nebo na videu je potřeba minimálně 400 pixelů (20x20) [23]. Tento případ se ovšem zabýval pouze rozeznání obličej a ne identifikaci nebo ještě následnou detekci významných bodů. S rozlišením také přichází otázka vzdálenosti, pokud budeme chtít rozeznávat výraz u vzdálenějších osob od kamery, tím budeme potřebovat vyšší rozlišení.

Použil jsem tedy kameru od ImagingSource DFK 41BU02.



Obrázek 2.1: Zapůjčená průmyslová kamera

Charakteristika kamery DFK 41BU02:

Čip: CCD

Typ: Barevná

Kabel: USB

Video formát / frame rate:

1280x960 UYVY @ 7.5, 3.75 fps

1280x960 BY8 @ 15, 7.5, 3.75 fps

Citlivost: 0,15 lx (lux)

Dynamický rozsah: 8 bitu

Velikost pixelu: H: 4.65 μm , V: 4.65 μm

Napájecí napětí: 4.5 až 5.5 VDC

Hmotnost: 265 g

Další vlastnosti a použití si můžete přečíst na stránkách dodavatele [1].

2.3 Objektiv

Předpokládám použití u stolního PC u kterého člověk sedí a komunikuje s nějakou svojí skupinou po internetu. Podle tohoto předpokladu jsem vybral i objektiv. Pro mou aplikaci mi vyhovoval objektiv s následujícími parametry:

- clona 1:1,4
- ohnisková vzdálenost 6-15 mm.

U svého projektu počítám, že kamera bude umístěna přímo na obrazovce nebo vedle ní. Pro zjednodušení uvažujeme že uživatel bude mít v obrazu jak hlavu tak i ramena pro případný menší pohyb.



Obrázek 2.2: Stativ

3 BAREVNÉ PROSTORY PRO DETEKCI KŮŽE

Barevný prostor popisuje základní barvy a model mísení těchto základních barev do viditelné (výsledné) barvy. Barva je v přírodě dána směsicí světél různých vlnových délek (od 380nm do 780 nm což odpovídá viditelnému spektru) a různé barevné modely se snaží napodobit barvu co nejdříveji. V praxi se používají modely, u kterých je zvolen vhodný kompromis mezi přesností podání barevného dojmu a složitostí konkrétního modelu. [31]

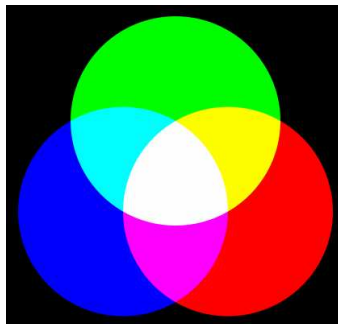
Modely mísení barev se rozdělují do pár hlavních kategorií.

Aditivní modely – barevný model pracující se světelnými zdroji barev (např. monitor nebo projektor)

Subtraktivní modely – barevný model pracující s odrazem světla – bílého (např. různé druhy tiskových technik, viz tiskárna)

Intuitivní modely – barevný prostor který je popsán ne matematicky, ale podle lidské intuice (př.: HSV, HLS)

a další



Obrázek 3.1 :Aditivního mísení barev

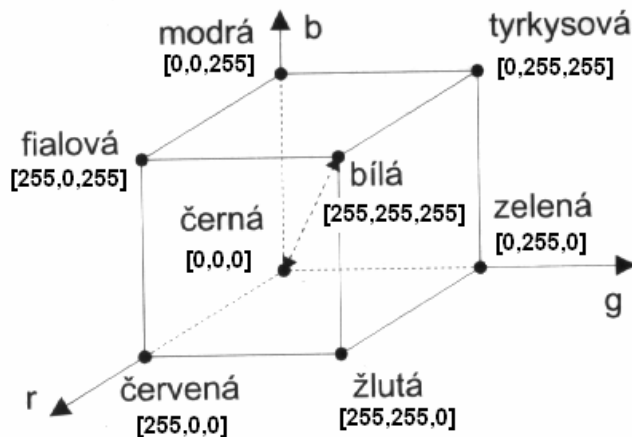
Pro vyhledání nejlepšího barevného prostoru pro segmentaci kůže jsem vycházel z vybraných zastupitelů barevných prostorů [2] [3] [4] [5].

3.1 Model RGB

Název modelu pochází z počátečních písmen tří základních barev R (red) –červená, G (green) – zelená, B (blue) - modrá. Někdy je používán model RGBA, kde čtvrtá složka znamená Alpha. Nejedná se o rozdílný model ale reprezentativní. Složka Alpha je použita pro množství průhlednosti.

Každá barva je udávána množstvím jednotlivých základních barev o vlnových délkách 780, 546,1 a 435,8 nm (tyto hodnoty jsou v mnoha publikacích popsány různě). Množství (nebo mohutnost) se udává různě, buď v procentech (dekadický způsob) nebo

podle použité hloubky jako určitý počet bitů vyhrazených pro barevnou komponentu (pro 8 bitů na komponentu je rozsah hodnot 0-255, pro 16 bitů 0-65535). Čím větší je mohutnost, tím s vyšší intenzitou se barva zobrazuje.



Obrázek 3.2: Model RGB zobrazený do krychle

3.2 Model HSV

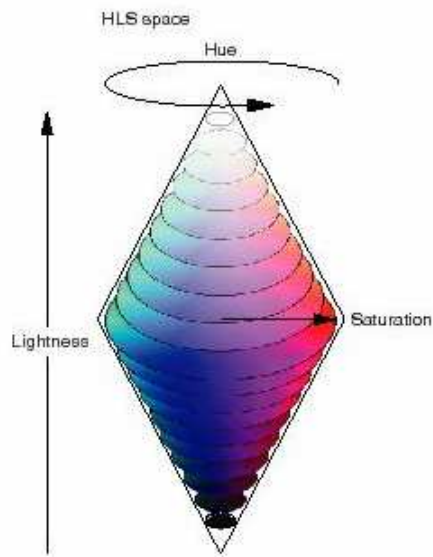
Hue – Barevného tónu, Saturation – Sytosti a Value – Jasové hodnoty. Tento model je známý taky jako HSB, kde místo Value se značí Brightness. Vznikl roku 1978 a nejvíce odpovídá lidskému vnímání barev. Hodnoty jednotlivých parametrů se udávají následujícím způsobem: Hue - (odstín) 0° - 360° nebo se označuje názvem barvy, Saturation – (chroma) 0 % - 100 %, Value – množství bílého světla.

Někdy je tento model nevhodný pro svůj jehlanovitý tvar, který způsobuje, že ve vodorovném řezu se musí bod o konstantní hodnotě S pohybovat při změně H po dráze ve tvaru šestiúhelníku a nikoliv po kružnici, jak by bylo přirozené. Dalším záporným jevem je nesymetrie modelu z hlediska jasu.

3.3 Model HLS

Hue, Saturation a Lightness a je také nazývaný HSL. Velmi používané označení je také HSI, kde třetí hodnota značí Intenzitu. Název není standardizovaný a tak může dojít k zaměnění některých zkratk.

Stejně jako u HSV je určitá barva definována úhlem (Hue), vzdáleností od osy (Saturation, Chroma) a vzdáleností podél osy (Lightness, Intensity, Brightness). Zatím co odstín je u obou modelu HSV a HSL definován stejně, v definici saturace je dramatický rozdíl.

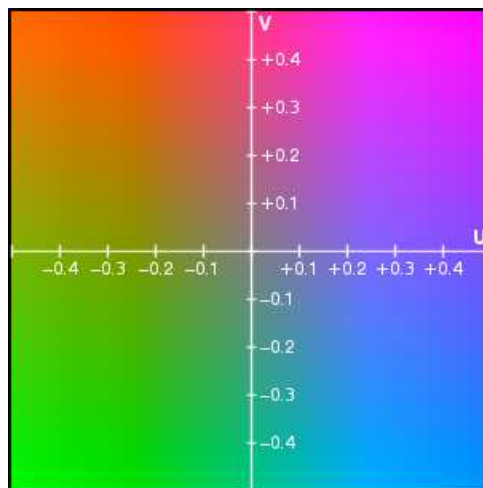


Obrázek 3.3: Reprezentace HLS modelu

3.4 Model YUV

Model YUV je někdy také označován jako Y, B-Y, R-Y. Kóduje podobným způsobem jako vnímá lidské oko. Složka Y má informaci pouze jasovou (černobílou) jako tyčinky v lidském oku, a zbylé dvě barevnou informaci, kterou detekují čípky. Barevný model YUV je používán v televizních vysílání v normě PAL i HDTV. Z tohoto modelu je odvozeno mnoho dalších jako například Y'UV, YCbCr, YPbPr atd.

Složky U a V nabývají hodnot od -0,5 do +0,5 a jasová složka Y nabývá hodnot od 0 do 1.



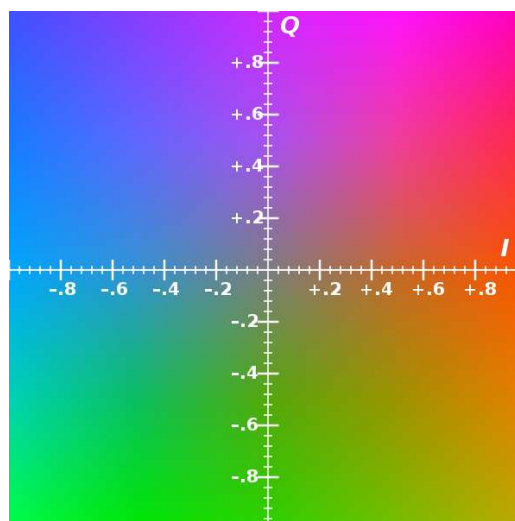
Obrázek 3.4: Ukázka kodování složek U a V

3.5 Model YIQ

Y složka opět představuje jasovou složku a složky I,Q barevnou informaci. I a Q mohou být interpretovány podobným způsobem, umístěním do stejného prostoru jen otočeny o 33° . Tedy IQ a UV představují odlišný souřadnicový systém ve stejném rovině.

Model YIQ je navržen tak, aby vylepšil barevnou citlivost lidského oka. Lidské oko mnohem citlivěji vnímá změny v rozsahu oranžová-modrá (I složka), než změny v rozsahu fialová-zelená (Q). A z toho důvodu je u vysílačů požadována menší šířka pásma pro Q složku.

V modelu YUV šly UV složky ukázat jako X a Y souřadnice v rámci barevného prostoru. I a Q mohou být interpretovány podobným způsobem, umístěním do stejného grafu jen otočeny o 33° . Tedy IQ a UV představují odlišný souřadnicový systém ve stejném rovině.

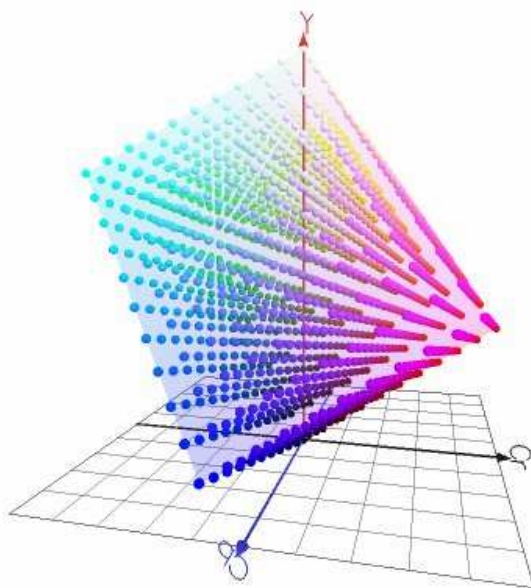


Obrázek 3.5: Ukázka složek I a Q v barevné rovině

3.6 Model YCbCr

Jeho složky jsou Y – jasová, Cb – rozdílový modrý obraz (modrý chromatický) a Cr – rozdílový červený obraz (červený chromatický). Hodnoty složek Y jsou od 0 do 1, Cb s Cr mají rozsah hodnot od -0,5 do 0,5.

YCbCr je někdy zkracován na YCC, ale zde by mohlo docházet k nejasnostem. Vzhledem k tomu, že rovnice, které definují YCbCr jsou tvořeny tak, že natáčejí celý prostor RGB a vzniká tak nový prostor, jsou pak určité barvy v RGB které nemohou být zobrazeny YCbCr jelikož jsou mimo jeho gamut. Toto způsobuje určité potíže při určování toho, jak správně interpretovat a zobrazit některé signály YCbCr.



Obrázek 3.6: Vyobrazení RGB krychle do prostoru YCbCr

3.7 Model Lab

Prostor Lab nebo též CIE 1976 (L^*, a^*, b^* nebo zvaný CIELAB). Lab je absolutní model vyjadřující barvy nezávisle na zařízení. Používá tyto složky: L – lightness, světelnost (0...100), a – osa zeleno-červená, b – osa modro-žlutá.

CIELAB je nejkompexnější barevný prostor určený Mezinárodní komisí pro osvětlení. Na rozdíl od RGB a jiných modelů, Lab je navržen tak, aby se přiblížil blíže k lidskému vidění. Prostor Lab je mnohem větší než gamut jiných modelů nebo dokonce lidského vidění. Mnoho barev prostoru Lab spadá mimo gamut lidského vidění a tak jsou čistě fiktivní, nemohou být reprodukovány ve fyzické podobě. Dnes se tyto fiktivní barvy nahradí tou nejbližší ve viditelném spektru.

3.8 Model Luv

Luv, CIE 1976 (L^*, u^*, v^*), CIELUV je model velmi používaný v počítačové grafice, zabývající se barevným osvětlením, pro svou jednoduchost. LUV je aktualizací barevného prostoru z roku 1964 CIEUUVW, rozdíly jsou menší změny v měřítku L a barevnost (osy U a V) jsou 1,5krát větší.

4 SEGMENTACE MODELU KŮŽE

Za účelem oddělení bodů kůže v barevném prostoru od okolí, jsem si vytvořil galerii fotek pro různá osvětlení a nastavení. Fotil jsem digitálním fotoaparátem CANON PowerShot A75 v laboratoři E617 na Kolejní. Pro větší kompatibilitu jsem při možných osvětleních, denní světlo, halogeny, zářivky a žárovky, fotil s každým možným nastavením fotoaparátu. Jiná osvětlení jsem vypustil s ohledem a použitím programu a možnosti fotoaparátu.

Scénu jsem si vždy připravoval tak, abych co nejvíce zamezil ostatním parazitním osvětlení. Při rozmístování světel jsem se snažil osvětlit celý obličej rovnoměrně, světla jsem rozmístil v rozmezí $\pm 45^\circ$ od osy pohledu.

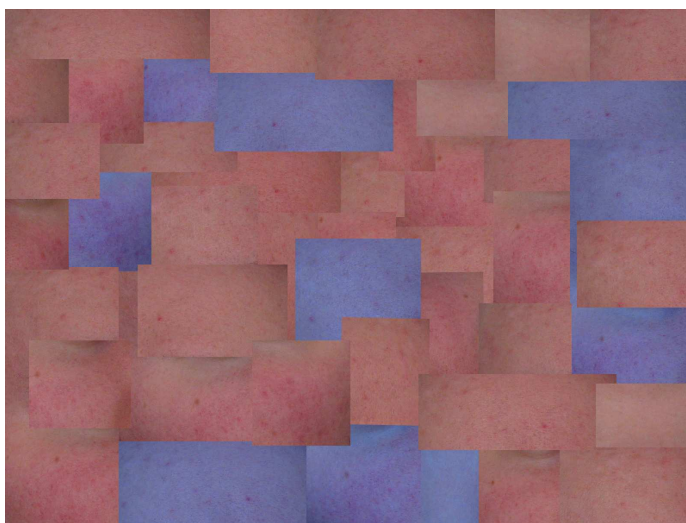
Pro takto osvětlené scény jsem tedy udělal vždy pět fotek pro každé nastavení fotoaparátu (CD/Galerie/Nafocené vzorky).

Fotoaparát umožňoval následující vyvážení bílé:

- Vnitřní osvětlení
- Denní osvětlení
- Zářivky
- Halogeny
- Blesk.

Fotky s bleskem byli ovšem nepoužitelné díky přesvětlení fotky a nemožnosti oddělení vzorků kůže.

Pro každé osvětlení jsem si vytvořil koláž vzorků kůže. Z každé fotky jsem si vyřízl kousek kůže a tyto kousky hodil do jednoho obrázku tzv. koláže (CD/Galerie/kolaze/).

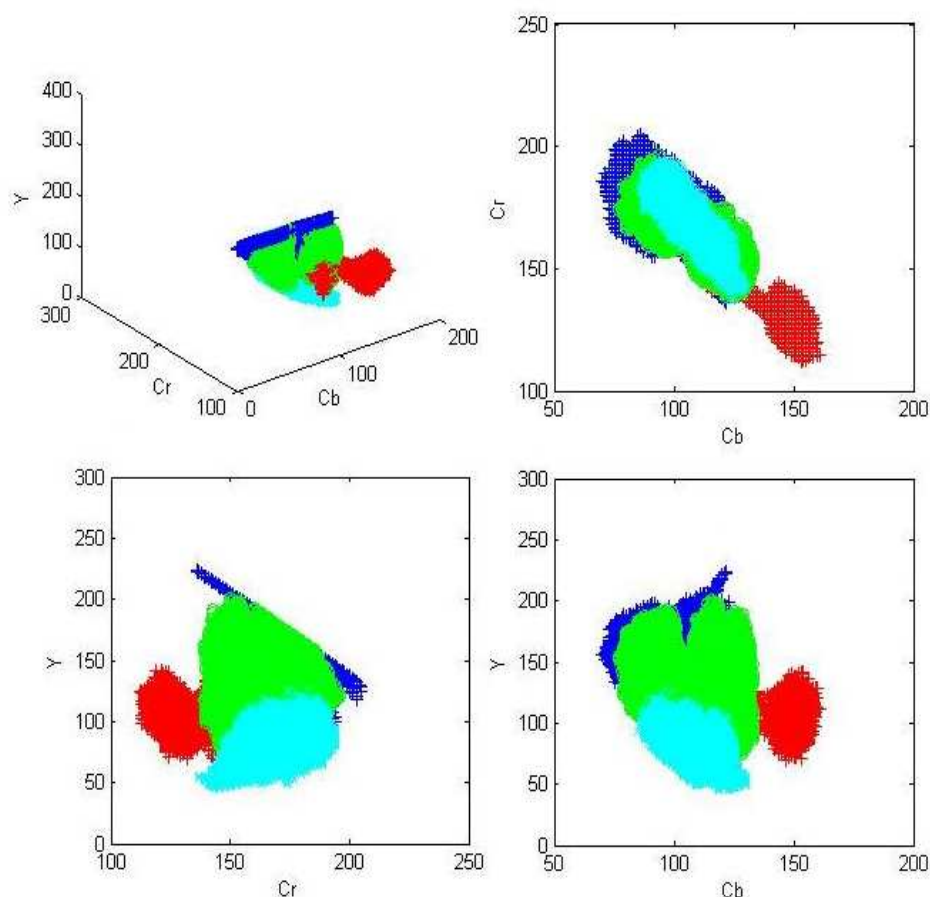


Obrázek 4.1: Koláž kůže

4.1 Výpočet reprezentantů barvy kůže

Když jsem si udělal koláž pro každé osvětlení začal jsem tyto vzorky kůže zobrazovat ve vybraných barevných modelech. V prvním vykreslení jsem nechal pouze vykreslit jednotlivé body do 3D grafu. Toto řešení bylo ovšem velmi časově náročné díky velkému množství bodů. Například u obrázku 1200x1600 je 1 920 000 bodů a každý má tři souřadnice (barevné složky). Samozřejmě mnoho bodů (barev) se opakuje tak jsem vytvořil M-File “MnStBodu.m” za účelem pouze nalézt jednotlivé barvy, později jsem přidal spočítání jejich množství (umístění CD/MatLab/...).

Použijeme-li tento M-File například na koláž pro denní světlo o velikosti 1200x1600 v RGB tak z 1 920 000 bodů dostáváme 48 513 barev, což je velmi dobrá redukce (pozn. Nejčastější barva má četnost 1711). Po této redukci a ujištění, že výsledný tvar je totožný s pouhým vynesem bodů, jsem si převedl všechny koláže do vybraných barevných prostorů. Poté aplikoval M-File a vynesl do grafů (redukované hodnoty lze nalézt: CD/MatLab/MnStBodu hodnoty/).



Obrázek 4.2: Ukázka grafu pro osvětlení žárovkou - prostor YCbCr (červená – denní osvětlení, modrá – halogeny, zelená - zářivky, tyrkysová - žárovky)

Tyto grafy nám ukazují celkové rozmístění bodů barvy kůže v daných barevných prostorech. Navzájem se přepisují v případech kdy při jednom osvětlení obsahuje kůže stejnou barvu jako při jiném, proto můžete v grafech vidět převahovat barvu poslední vykreslované koláže (ostatní grafy můžete vidět: CD/obrázky reprezentantů kůže v barevných prostorech/). Z těchto 3D rekonstrukcí vysegmentované kůže jsem vybral ty, které šlo nejlépe popsat matematickým tělesem tak, aby obsahovali co nejvíce z vykreslených bodů. Vybral jsem tedy prostory YCbCr, Lab a YIQ.

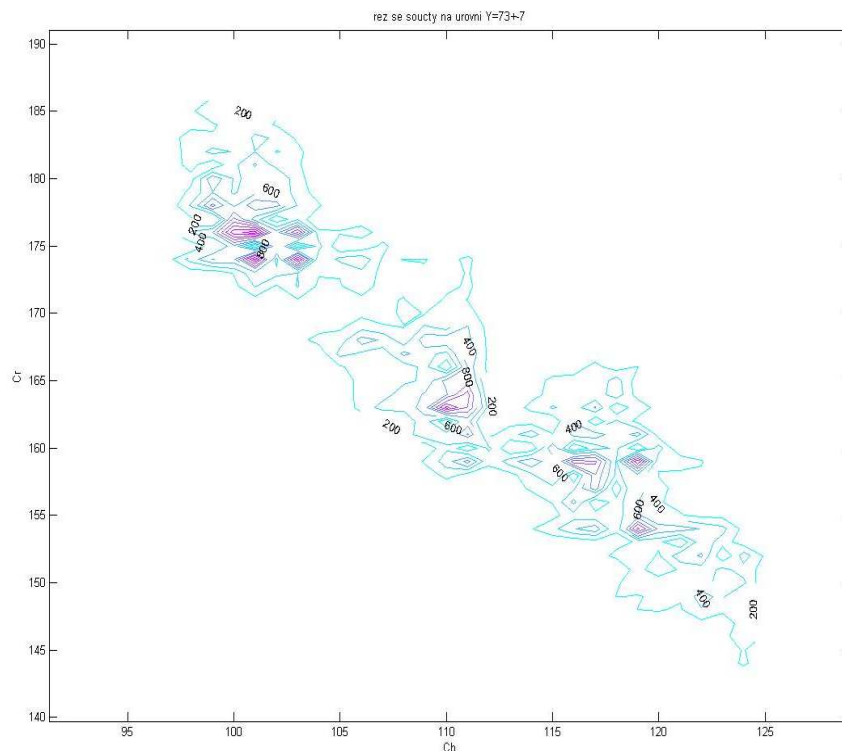
Nesmíme ale pozapomenout na barvu kůže jako takovou. Barva kůže je nehomogenní (není na celé ploše stejná) a vyskytuje se v ní mnoho různých barev z pohledu barevných modelů. Barvu kůže tvoří šum několika různých a však podobných barev. Musíme ale brát také ohled na to, že každý člověk má jiný poměr četností těchto barev ale v základu se vyskytují opakovaně stejné barvy. Myslím tím například člověka, který je hubenější tak mu jdou vidět žíly které zbarvují části kůže trochu do modra.

Z tohoto důvodu bylo zapotřebí vynést četnosti reprezentantů barvy kůže (viz kapitola 4.2).

4.2 Hustota reprezentantů modelu kůže

Pro lepší zobrazení jsem vykreslil u vybraných prostorů i jednotlivé četnosti barev. Je to nutné s ohledem na charakter barvy kůže. Dal by se popsat jako šum několika různých barev. Tyto barvy jsou si navzájem podobné, ale můžou se tam vysvitnout i barvy dost nepravděpodobné nebo až náhodné (s ohledem na osvětlení, pohyb, vzorkování kamery atd.).

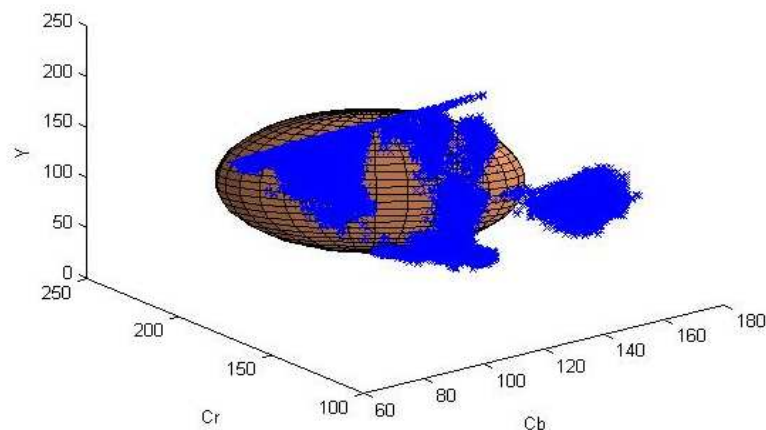
Vykreslením četností zjistíme, kde jsou největší shluky nejvíce se opakujících barev, které je potřeba obklopit daným virtuálním tělesem. Jelikož se často v prostorech vyskytovali barvy s minimální četností, tak jsem pro lepší zviditelnění vždy sečetl několik úrovní v určitém rozmezí (např.: `ContourYCbCr.m - CD/MatLab/...`). Z průřezů je i patrné že v případě YCbCr červené body vystupující z celkové shluku jsou oproti četnostem ve shluku zanedbatelné.



Obrázek 4.3: Ukázka četností v prostoru YCbCr na úrovních 66-80

4.3 Geometrické těleso reprezentující lidskou kůži

Pro nalezené virtuální tělesa jsem vzal v úvahu obrázky vynesných bodů, ve kterých jsem sestrojil elipsoidy tak aby obsahoval většinu vykreslených bodů. Menší úpravy jsem prováděl pro pokrytí i nejčetnějších míst např: viz. Obrázek 4.3.



Obrázek 4.4: Elipsoid pro YCbCr

YCbCr						
Střed			délka os			Natočení
Y	Cb	Cr	Y	Cb	Cr	
125	105	170	70	45	13	$\pi / 4$
Lab						
střed			délka os			Natočení
L	a	b	L	a	b	
125	157	150	73	34	15	$5\pi / 8$
YIQ						
střed			délka os			Natočení
Y	I	Q	Y	I	Q	
0,5	0,23	0,0475	0,3	0,17	0,0725	0

Tabulka 4.1: Hodnoty elipsoidů pro vybrané modely

Zde se samozřejmě jedná o teoretické středy těles a jejich os. Jejich přesnost lze vylepšit postupnými testy a úpravami vlastností těles.

Možná by bylo vhodnější zkusit obklopit všechny body pomoci matematicky popsatelného ale nepravidelného tělesa, nebo popřípadě obklopit pouze jednotlivé velké shluky jednoduchými geometrickými tělesy a podle nich vyhledávat barvu kůže. Tyto způsoby by ovšem byli dosti náročné. V případě nepravidelného tělesa tu figuruje problém matematického popisu samotného tělesa, aby obklopovalo všechny body. Pokud se podíváme na jednotlivé modely (CD/ obrázky reprezentantů kůže v barevných prostorech/.) můžeme vidět jejich silnou nepravidelnost a těžkou popsatelnost. Ve druhém případě bychom měli několik například elipsoidů. Každý elipsoid by byl pospán třemi osami a třemi natočeními podle daných os. Což by podstatně zvýšilo problém vyhledání výskytu bodu v daných tělesech.

4.4 Klasifikace kůže

Když jsem si určil tyto tělesa mohl jsem začít vyhledávat kůži v obraze. Napřed jsem použil nalezený tvar jako pevný práh a podle něho prahoval. Tyto výsledky ovšem neměli moc dobré výsledky (viz Obrázek 4.5 - vlevo). Přidal jsem proto lineární funkci určující vzdálenost od středu tělesa. Využil jsem pro to samotnou funkci elipsoidu, kde pokud se bod pohybuje od středu k povrchu výsledek rovnice narůstá od 0 do 1.

$$\left(\frac{x}{xc}\right)^2 + \left(\frac{y}{yc}\right)^2 + \left(\frac{z}{zc}\right)^2 \leq 1 \quad (4.1)$$

Na následujícím obrázku můžete vidět výsledky (Obrázek 4.5 - vpravo). Je zde viditelné že model YIQ není příliš vhodný.



Obrázek 4.5: Pevné prahování podle tělesa (nalevo) s pravděpodobností (napravo) – prostor YIQ

Zde můžete vidět porovnání dvou zbývajících modelů YCbCr a Lab.



Obrázek 4.6: Porovnání detekce kůže pomocí prostorů Lab s prahem 0,5 a s YCbCr s prahem 0,6

Na obrázku 4.6 můžete vidět, že prostor Lab nebere více osvětlenou kůži a při zvýšení jasové složky nebo zmenšení prahu se tento model už chytá světlejších oblastí jako šatů nebo betonu. Jako nejlepší se ukázal model YCbCr.

Na obrázkách můžeme vidět že se nám celkově podařilo vysegmentovat hledanou kůži ale ne zcela dokonale. Má to podstatu v už zmiňovaném charakteru kůže. Kůže obsahuje různé barvy, které jsou rozmístěné náhodně (pro představu obdobně jako šum). Tyto barvy se mi podařilo vysegmentovat, ale jednotlivé prvky barvy kůže se mohou vyskytovat i jinde než na lidské kůži. Například velmi světlá dřeva jsou tvořena podstatnou, velmi četnou barvou vyskytující se v kůži. Také opečené či jinak připravené maso na konzumaci obsahuje velmi podobné barvy (viz obrázek 4.7). Některá dřeva díky své barevné členitosti (tmavější místa letokruhů) se mi podařilo díky geometrickému tělesu vymezující barvu kůže odstranit ze segmentace.



Obrázek 4.7: Ukázka segmentování barvy kůže a zachycení i připraveného masa

Poslední nedostatek se týká samotného osvětlení. Pokud totiž osvětlíme obličej příliš silným světlem nebo dotyčný či dotyčná má mastnou pleť, světlo se odráží ve své podstatě v bílé barvě. Existují lidé s opravdu světlou pletí mající tzv. Albinismus. Jejich pokožka je velmi světlá a citlivá na sluneční světlo. Ale přesto jejich pokožka dosahuje světle růžové barvy, bílé barvy dosahují pouze jejich vlasy atd. Ukázku špatného nasvětlení a následní špatné segmentace můžeme vidět na následujícím obrázku 4.8.



Obrázek 4.8: Ukázka silně osvětleného obličeje

Tento nedostatek by šel opravit pomocí zvýšení prahu ale na úkor segmentování dalších oblastí už nezastupující pokožku ale jiná tělesa v obraze. Například na Obrázku 4.5 si můžeme všimnout zachytávání se o okraj červeného sudu vyskytující se v levé části fotografie.

5 NALEZENÍ OBLIČEJOVÉ ČÁSTI

5.1 Teoretické rozdělení

Pro detekci obličeje ve snímané scéně se využívá algoritmů, které lze rozdělit do určitých oblastí podle využívané vlastností obrazu. Mezi nejpoužívanější metody detekce obličeje patří [24]:

a) Detekce obličeje ve snímcích se známým pozadím

V tomto případě se jedná o velmi jednoduchý přístup, protože se snímají objekty s předem známou barvou pozadí. Lze tedy snadno oddělit pozadí od objektu, který vždy odpovídá obličeji.

b) Detekce obličeje pomocí barvy

Metoda spočívající v detekci lidské kůže. Problém této metody je především v malé robustnosti navrženého algoritmu. Hlavní problém představuje fakt, že lidé nemají jednotnou barvu kůže. Dalším problémem je pak náchylnost na různé osvětlení snímané scény.

c) Detekce obličeje pomocí pohybu

Pokud máme k dispozici video, můžeme využít faktu, že obličej se ve snímané scéně neustále pohybuje. Problémy vznikají, je-li ve scéně více pohybujících se objektů, pak je potřeba všechny „pohybové“ oblasti správně klasifikovat podle specifických kritérií.

d) Kombinace předchozích metod

Spojením několika vhodných přístupů lze dosáhnout lepších výsledků.

e) Detekce obličeje v komplexním pozadí

Nejobtížnější případ detekce obličeje nastane tehdy, neznáme-li předem scénu, ve které jej máme detekovat, a tudíž nemáme žádné informace o barvě kůže, pozadí nebo pohybu v této scéně. V takovém případě se využívají např. neuronové sítě, které se daným podmínkám přizpůsobí.

Na rozdělení se můžeme dívat i z pohledu přístupu a zpracování dat z obrazu [30]

Nízko úroňňová analýza – Tato skupina přístupů využívá nízko úroňňové příznaky jako například barvu pokožky, hrany v obraze, symetrie oblastí nebo pohyb z videosekvence. Tyto příznaky jsou poté jednoduše zpracovány v závislosti na tvaru a velikosti.

Příznaková analýza – Některé příznaky získané pomocí předchozí skupiny metod nemusí jednoznačně souviset s lidskou tváří. Tato neurčitost se dá odstranit analýzou pozice jednotlivých příznaků v závislosti od geometrie lidské tváře. Tato skupina tedy obsahuje metody pro dohledávání příznaků, které se na základě příznaků (pár očí, kontura tváře atd.)

Tvarové modely – V této skupině jsou metody, které se používají hlavně na detekci přesné polohy jednotlivých částí takže i celého obličeje. Nutnou podmínkou použití metody je informace o přibližné pozici tváře. Na tuto pozici je potom v rámci procesu vyhledávání umístění tvarových modelů.

Vzhledové modely – Jde o skupinu s podobnou podstatou s Tvarovými modely . Na rozdíl od nich však nepracuje jen na základě příznaků v okolí vybraných tvarových částí, ale na základech příznaků z celé oblasti. Tato skupina také potřebuje informaci o přibližném umístění tváře. V průběhu iteračního procesu se mění parametry vzhledového modelu.

Obrazově orientované přístupy - se snaží o získání statistické charakteristiky tváří v obraze jako celku. Obrazy tváří mají při tom přesně definované atributy – velikost, natočení, osvětlení atd. Na základech této charakteristiky pracuje klasifikátor, který dokáže prohlásit že v obraze je lidský obličej.

5.2 Implementace vyhledání obličeje

Mnou vybraný způsob by se zařadil mezi Obrazově orientované přístupy využívající barvu kůže. Obličej vyhledává za pomoci existence orientovaných umístění kontrastních regionů v obraze. Soubor těchto vlastností může být už samostatně použit pro vyhledávání obličejů nebo tvarů jim podobných.

Klasifikátor byl trénován s několika stovkami vzorů, které byli zmenšeny na minimální velikost pro rozpoznání obličeje. Pro vyhledávání ve větších obrazech je možné masku vzoru postupně posouvat a prohledat tak celý obraz. V případě že nalezneme v oblasti obličej výstup klasifikátoru se nastaví na “1“ pokud ne “0“. Klasifikátor je upraven tak, aby bylo možné měnit velikost hledaného obličeje, tento způsob je mnohem praktičtější než měnit velikost obrazu v němž hledáme objekt.

Využití slovo kaskáda v kódu (řádek 222) znamená že klasifikátor se skládá z několika jednodušších fází, které jsou postupně použity pro oblasti zájmu až v určité fázi kdy je daná oblast zamítnuta. Třídění v každé fázi kaskády je složité a samy o sobě jsou postaveny mimo klasifikátor pomocí jedné ze čtyř posilujících technik.

Použitá verze obsahuje tyto techniky:

Discrete Adaboost

Real Adaboost

Gentle Adaboost

Logitboost

5.2.1 Boosting

Boosting představuje strojové učení pomocí tzv. meta-algoritmu (metody jak nejlépe učit klasifikátory). Boosting je založen na otázce může několik weak learners (špatných žáků) vytvořit jednoho strong learner (výborného žáka)?

Weak learner můžeme chápat jako klasifikátor, který jen trochu koreluje se skutečnou klasifikací (jeho výsledky jsou lepší než náhodné hádání). Naproti tomu, Strong learner svévolně dobře koreluje s opravdovou klasifikací. [6]

Zatím co boosting není algoritmicky omezen, většina boosting algoritmů tvoří iterační učení slabých klasifikátorů, pokud jde o distribuci a jejich vlivu do finálního silného klasifikátoru. Po jejich přidání jsou k nim určitým způsobem přiřazeny váhy, které odráží přesnost jednotlivých slabých žáků. Tyto váhy jsou postupně pozměňovány podle učení. [7]

Existuje mnoho boosting algoritmů, ten originální byl navržen Robertem Schapire (Rob Schapire, Strength of Weak Learnability. Machine Learning Vol. 5, 1990) a podpořil ho Yoav Freund (Yoav Freund. Boosting a weak learning algorithm by majority. Proceedings of the Third Annual Workshop on Computational Learning Theory, 1990). Tento první algoritmus nebyl adaptivní a nemohl plně využít slabých studentů.

5.2.2 LogitBoost

Je boostovací algoritmus formulován Jeromem Friedman, Trevorem Hastie, a Robertem Tibshirani. Původní práce přiřazuje AdaBoost algoritmus do statistického rámce. Konkrétně, pokud vezmeme v úvahu AdaBoost, že je generalizovaný aditivní model tak pak platí funkce logistické regrese, pak tedy můžeme odvodit algoritmus LogitBoost.

5.2.3 Adaboost

Je zkratka pro Adaptive boosting a je adaptivní v tom smyslu, že je postaven na zlepšování klasifikátorů v případech kdy předchozí klasifikátory chybně rozpoznají objekt. AdaBoost je citlivý na šum a příliš vzdálená (odlehlá) data.

Výběr základních klasifikátorů v rámci AdaBoost algoritmu probíhá následovně

- 1) Přidělení vah každému vzorku v rámci množiny M
- 2) Trénování klasifikátorů p_i , $i = 1, \dots, K$ s použitím vzorků z množiny M a k nim příslušných vah.
- 3) Výpočet hodnot e_i představující chybovost jednotlivých klasifikátorů p_i . Jednotlivé hodnoty e jsou vypočítané na základě vzorkovací množiny M a k nim příslušných vah.
- 4) Výběr klasifikátoru p_i , který dosahuje minimální chybovosti, do množiny základních klasifikátorů v AdaBoost algoritmu.
- 5) Změna vah jednotlivých vzorků v množině M na základě předpisu určeného AdaBoost algoritmem.
- 6) Pokud není proces AdaBoost algoritmu zastavený tak se vrátí na krok 2.

Výsledkem AdaBoost algoritmu je silný klasifikátor [25], [26]:

$$P(Z) = \text{sign} \left(\sum_{t=1}^{BC_{num}} \alpha_t h_t(Z) \right) \quad (5.1)$$

Kde BC_{num} představuje výsledný počet základních klasifikátorů použitých v silném klasifikátoru a α_t zase koeficient důležitosti pro jednotlivé základní klasifikátory.

Zde můžeme rozhodovat pomocí hranice q , která je v případě funkce nastavená na 0, měnit a tím pádem řídit úspěšnost silného klasifikátoru na testovacích množinách ve kterých je poměr mezi „tvarovými“ a „netvarovými“ vzorky jiný jako v trénovací množině. Tím nám vzniká změna ve formulaci silného klasifikátoru.

$$P(Z) = \text{sign} \left(\left(\sum_{t=1}^{BC_{num}} \alpha_t h_t(Z) \right) + q \right) \quad (5.2)$$

Vraťme se k trénovací množině M , která je z normalizovaných tvarových a netvarových vzorků. Pokud mají tyto vzorky velikost $W \times W$ obrazových bodů, výběr vhodné velikosti W má vliv na výslednou dobu trénování AdaBoost algoritmu. Také velikost W má vliv na možnou velikost detekovaného obličeje (detektor nenajde obličej jiné velikosti než na jakou velikost byl natrénován).

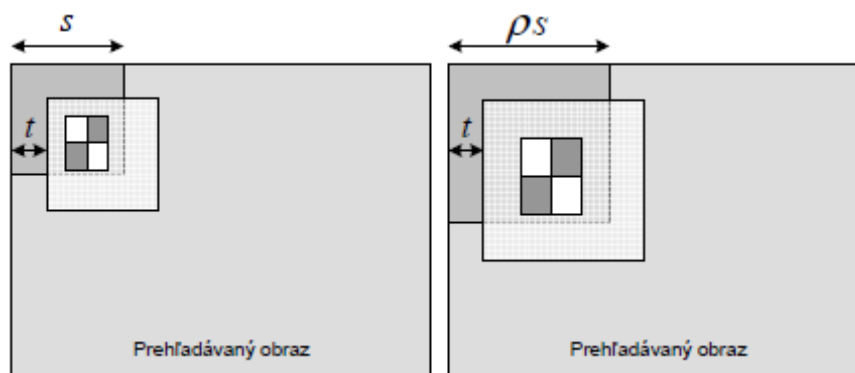
Situace by šla řešit pyramidovým zmenšením vstupního obrazu a aplikací detektoru konstantní velikosti. To by však vyžadovalo další zpracování, související s zmenšováním obrazu, přitom by integrální obraz musel být vytvořen na každé úrovni pyramidu.

Jiné řešení je postavené na zajímavé vlastnosti Haarových příznaků. Ty totiž mají přibližně stejné hodnoty pro ten samý obraz v různé velikosti. Tím pádem nemusíme zmenšovat nebo zvětšovat prohledávaný obraz, ale naopak měnit velikost vybrané oblasti a s ní souvisejících příznaků.

S prohledáváním vstupního obrazu tedy souvisí následující parametry.

- s – počáteční velikost hledaných tváří v obraze (velikost prověřované oblasti)
- t – posun při prohledávání vstupního obrazu
- ρ – koeficient zvětšení velikostí hledaných tváří v obraze

Význam parametrů je zobrazen na následujícím obrázku 5.1. Zvětšování oblasti klasifikátoru AdaBoost algoritmu končí, když je vybraná oblast větší jak prohledávaný obraz [27].



Obrázek 5.1: Ukázka významů parametrů s , t , ρ

Tváře v trénovací množině nebo v samotných prověřovaných obrazech mohou být osvětleny různým způsobem. Bylo by tedy vhodné normalizovat vliv různorodého osvětlení na jednotlivé trénovací vzorky respektive prověřované oblasti. Jedním způsobem je normalizace rozptylu hodnot intenzit v obraze. Necht' obraz Z je tvořen z $W \times W$ bodů, který mají hodnotu intenzit $I(x,y)$. Potom střední hodnota těchto hodnot μ a rozptyl hodnot σ^2 jsou

$$\mu = \bar{I} = \frac{1}{nm} \sum_{i=0}^m \sum_{j=0}^n I(i, j) \quad (5.3)$$

$$\sigma^2 = \frac{1}{nm} \sum_{i=0}^m \sum_{j=0}^n (I(i, j) - \mu)^2 \quad (5.4)$$

S využitím informací o rozptylu a střední hodnotě můžeme hodnoty intenzit v bodě (x,y) normalizovat následovně.

$$\frac{I(x, y) - \mu}{\sigma} \quad (5.5)$$

Takto navržený postup by se dal úspěšně použít při normalizaci trénovacích dat, kde nezáleží na rychlosti normalizace procesu. Při samotné detekci by však šlo o výrazné zpomalení, když by hodnoty intenzit vstupního obrazu museli být normalizovány pro každý příznak zvlášť.

Vytvoření kaskádového zapojení

Rychlost klasifikátoru trénovacího pomocí AdaBoost algoritmu je lineárně závislá od počtu použitých základních klasifikátorů. Úspěšnost detekce je také závislá od počtu základních klasifikátorů. Snaha je použít co nejvíc základních klasifikátorů a při tom zachovat dostatečnou rychlost zpracování. Tento problém řeší kaskádové zapojení, jehož podstata spočívá v tom, že důkladně prověřuje jen ty oblasti, které vykazují tvarové příznaky a naopak oblasti, které je nevykazují prověří velmi rychle. Například homogenní plochy na vstupním obraze určitě neobsahují lidské tváře a proto je na jejich zamítnutí potřeba jen několik vyhodnocených příznaků (základní klasifikátor).

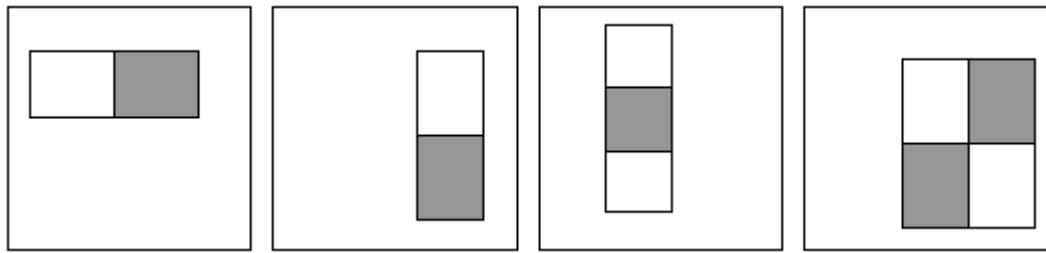
5.3 Metoda Haarových příznaků

Haarovy příznaky

Při klasifikaci obrazu můžeme použít více přístupů. Jedním z nich je klasifikace přímo na základě hodnot intenzit klasifikovaného obrazu. Jiný přístup je klasifikace za pomoci příznaků.

Její výhodou spočívá v tom, že příznaky v sobě obsahují informaci o charakteru vybrané oblasti obrazu. Pokud se dá tato informace vyjádřit jednoduchým způsobem, dostáváme velmi rychlou metodu.

Při použití Boosting kaskádové detekce se využívají příznaky extrahované pomocí Haarových bázeových funkcí. Příznak v tomto případě znamená rozdíl sum intenzit dvou nebo více sousedních oblastí (suma intenzit světlejší oblasti je vždy odčítána od sumy intenzit tmavší oblasti viz obrázek 5.2).



Obrázek 5.2: Příklady typů Haarových příznaků (zleva: leva hranice, dolní hranice, horizontální přímka, přímka s uheml 45°)

Uvažme že chceme vypočítat hodnoty všech příznaků (pro všechny velikosti a pozice), který mají charakter totožný s příznaky znázorněným na obrázku 5.2. Pro čtvercovou oblast velikosti 24x24 bodů dostáváme celkově víc jako 45000 různých příznaků (mění se typ, velikost a pozice). Pokud to zobecníme na oblast velikosti MxN obrazových bodů, tak dostáváme množinu příznaků $P = p_i$, $i = 1, 2, \dots, K$ kde p_i představuje i-ty typ příznaku a K zase celkový počet příznaků pro oblast. Proto je nutné zvolit rozsah přeměřované oblasti, ale i parametry příznaků (minimální a maximální velikost, krok posunu ve vybrané rámci oblasti) tak, aby výsledný počet prověřovaných příznaků nebyl příliš velký.

Každý vytvořený příznak pro stanovenou oblast nese informaci o tom, zda vybraná oblast obsahuje nebo neobsahuje tvář. To vytváří otázku, které příznaky nesou tuto informaci a jak tuto informaci vyhodnotit

Integrální obraz

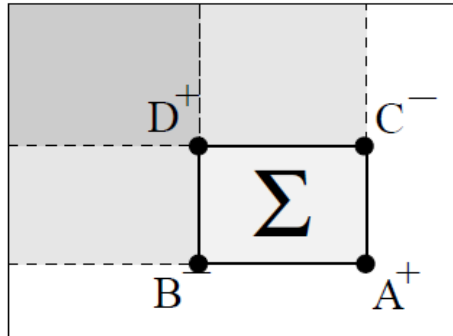
Integrální obraz je struktura vhodná pro velice rychlé zjištění součtu hodnot všech bodů uvnitř libovolné obdélíkové oblasti zdrojového obrazu. Integrální obraz představuje transformaci vstupního obrazu, kde se hodnota intenzity v původním obrazu $I(x,y)$ nahradí sumou intenzit $I_{int}(x,y)$. Tato transformace se počítá pro každý bod. Obraz intenzit má tedy stejné rozměry jako původní obrázek jen jeho hodnoty jsou určeny součtem všech hodnot bodů uvnitř obdélíku ohraničeného levým horním rohem obrázku a bodem, který je počítán. Počítat pro každý bod vždy součet celé této oblasti by bylo výpočetně náročné, naštěstí lze dynamicky používat již vypočtené hodnoty a složitost tvorby integrálního obrázku je tak lineární.

Výhodou integrálního obrazu je, že z něj můžeme jednoduše vypočítat sumu intenzit dané oblasti v obraze. Proto se často využívá pro Haurovy příznaky.

$$S(x, y) = \sum_{i=1}^x I(i, y) = S(x-1, y) + I(x, y) \quad (5.6)$$

$$I_{int}(x, y) = \sum_{i=1}^x \sum_{j=1}^y I(i, j) = I_{int}(x, y-1) + S(x, y) \quad (5.7)$$

Po skončení generování je možné zjistit součet libovolného obdélníkového regionu v původním obrázku jen za pomoci sečtení čtyř čísel, jak je naznačeno na následujícím obrázku. Součet bodů v regionu vyznačeném Σ je dán jako $A - B - C + D$, kde A,B,C,D představuje hodnoty bodů v integrálním obrázku.



Obrázek 5.3: Vysvětlující obrázek k ke integrálnímu obrazu

Klasifikace na základě Haarových příznaků

Nechť $M_{\text{face}} = \{X_1, \dots, X_N\}$ je množina obrazů X_i , které mají stejné rozměry a obsahují obrazy normalizovaných lidských tváří (velikost, natočení v rovině, pozice). Množina $M_{\text{noface}} = \{Y_1, \dots, Y_L\}$ je množina obrazů Y_i , které mají stejnou velikost jako X_i , ale na rozdíl od nich, obrazy tváří neobsahují. Definujme si množinu M jako průnik množin a a $f_i(Z_j)$ je hodnota i -tého příznaku na j -ty obraz, kde $i = 1, \dots, K$ a $j = 1, \dots, N + L$.

Pro každý typ příznaku f_i , $i = 1, \dots, K$, dokážeme potom na základě množiny M vytvořit klasifikátor $h_i(Z)$, kde Z je obraz o stejných rozměrech jako X_i , které dokáže vyhodnotit jestli Z obsahuje, a nebo neobsahuje lidskou tvář. Tento klasifikátor definujeme jako:

$$h_i(Z) = +1 \text{ (oblast } Z \text{ obsahuje tvář) pokud } p_i f_i(Z) < p_i \theta_i$$

$$h_i(Z) = -1 \text{ (oblast } Z \text{ neobsahuje tvář) jinak}$$

přičemž θ_i je rozhodovací práh a p_j paritní funkce nabývající hodnot $\{-1, +1\}$. Optimální hodnoty parametrů klasifikátoru $h_i(Z)$, θ_i a p_i , můžeme pro data z trénování množiny M určit například pomocí „Information Gain“ algoritmu nebo algoritmu dělení oblastí.

Kvalitu získaných klasifikátorů h_i můžeme ověřit na množině M a tím pádem pro každý klasifikátor získat jeho chybovost, e_i , $i = 1, \dots, K$. Ten klasifikátor, který dosáhne nejmenší chybovosti představuje základní klasifikátor pro AdaBoost algoritmus.

6 VÝZNAMNÉ BODY NA OBLIČEJI

Vyhledávání významných bodů se velmi často využívá v počítačovém vidění. Za jejich pomoci se získávají různé rysy a následně odvozuje obsah obrazu. Využívají se například při sledování pohybu, natočení či pohybu ve fotografii, tvorby panoramatických fotografií, 3D modelování, rozpoznávání objektů a další. Významný bod obecně označuje místo, které má:

- jasnou a matematicky dobře podloženou definici
- jasnou definovanou pozici v obrazovém prostoru
- lokální struktura v obraze kolem významného bodu je bohatá na informace vhodné pro pozdější zpracování vizuálním systémem
- je stabilní z hlediska působení lokálních a globálních deformací v obrazové doméně, tak aby byl bod opět nalezen s vysokým stupněm opakovatelnosti

Významný bod může být definován na předělu dvou hran. Významný bod v obraze je bod, který má přesně definovanou pozici a může být opakovaně detekován. Z toho vyplývá že hledaným bodem může být roh nebo také izolované místo maximální nebo minimální intenzity, konec úsečky atd.

V praxi většina tak zvaných metod pro detekci významných bodů řeší problematiku spíše obecně než hledání specifických rohů. Jako důsledek je nutné provést lokální analýzy pro vybraní a nalezení požadovaných bodů. V mém případě jako koutky očí, úst atd.

Metody vyhledávání významných bodů nejsou často dostatečně robustní a často potřebují odborný dozor nebo velkou redukci nalezených bodů. Jeden z bodů kvality takových detektorů je schopnost odhalit stejné body ve více podobných snímcích a to za podmínek různých osvětlení, rotace a dalších translací. Velmi jednoduchý přístup k vyhledávání bodů je za pomoci korelace, ale tahle metoda je velmi výpočetně náročná.

6.1 Detektor MORAVEC

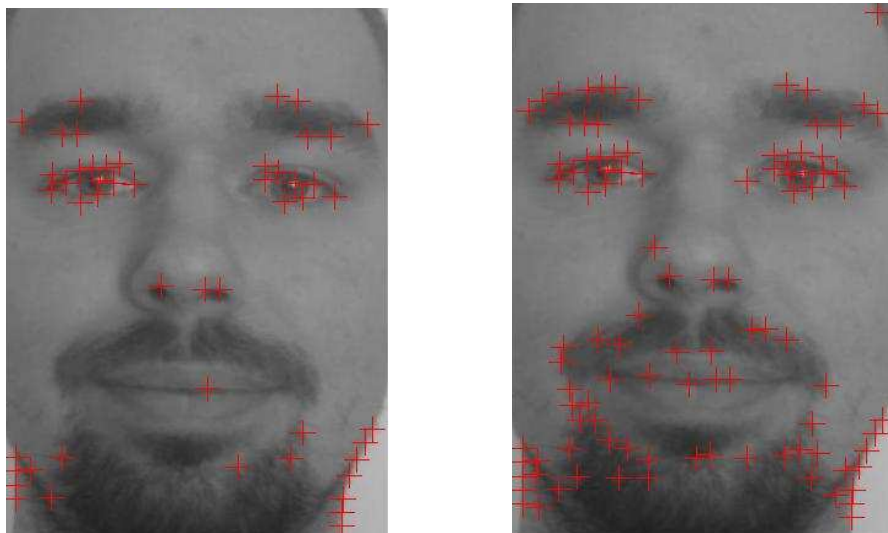
Moravcův algoritmus je jeden z nejstarších algoritmů pro detekci bodů (Hans Moravec - 1977). Algoritmus prochází každý pixel v obraze a testuje, jestli v daném pixelu není roh (za pomoci diference jasové úrovně) z oblasti kolem. Podobnost se měří pomocí součtu čtverců rozdílu mezi dvěma oblastmi. Čím nižší číslo tím je podobnost vyšší. [8]

$$f(i, j) = \frac{1}{8} \sum_{k=i-1}^{k=i+1} \sum_{e=j-1}^{e=j+1} |g(k, e) - g(i, j)| \quad (6.1)$$

Pokud je pixel v oblasti jednotné intenzity okolí vypadá velmi podobně. Pokud je pixel na hraně pak okolí v jednom směru vypadá zcela jinak, ale ve směru shodném se směrem hrany bude jen minimální změna. Pokud je pixel testován funkcí s variací ve všech směrech nebude zde už taková velká změna.



Obrázek 6.1: Ukázka výstupu Moravcova operátoru ve kterém se vyhledávají maxima (pro lepší zviditejnění jsou barvy invertovány, Černá = Max, Bílá = Min)



Obrázek 6.2: Detekce Moravcova operátoru (vlevo 50bodů, vpravo 100 bodů)

Tento detektor je ovšem velmi pomalý a tak není vhodný pro real-.time aplikace.

6.2 Detektor HARRIS

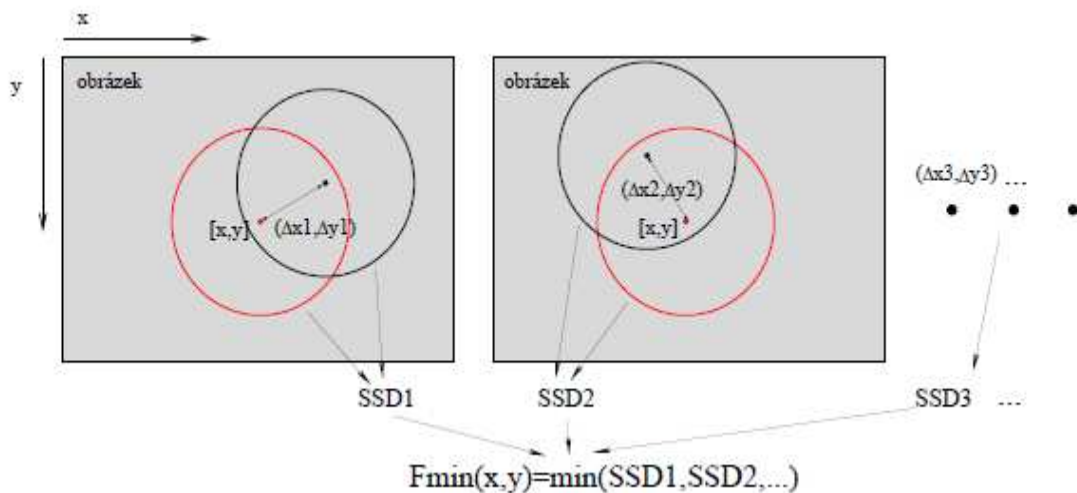
Na tomto detektoru pracovali Chris Harris a Mike Stephens – 1988. Tito dva muži zlepšili Moravcův operátor tím, že přidali zvažování diferenciálu rohu s ohledem přímo na směr, namísto použití masek. Tento způsob je často označován jako autokorelační. Přesně matematicky jde o sumu čtverců odchylek tedy podobnost jasové funkce při posunu okénka.[9]

Takovou jasnou a matematicky dobře podloženou definicí významného bodu je definice pomocí maximalizace minima autokorelační sumy. Souřadnice v obrazové oblasti označím jako x, y a okolí bodu o těchto souřadnicích označím O . Významné body budou ležet v lokálních maximech funkce F_{\min} , která má tvar:

$$F_{\min}(x, y) = \min_{(\Delta x, \Delta y) \in D} \sum_{(x, y) \in O} (I(x, y) - I(x + \Delta x, y + \Delta y))^2 \quad (6.2)$$

kde Δx a Δy představují elementární posuvy ve směru x a y . Jsou vybírány z množiny všech elementárních posuvů o velikosti, definované jako:

$$D = \{(\delta x, \delta y) \in R^2 : \|(\Delta x, \Delta y)\| = e, e > 0\} \quad (6.3)$$



Obrázek 6.3: Princip autokorelační funkce. Součet čtverců rozdílů (SSD) se počítá ze dvou oblastí vyznačených kroužky, které jsou posunuté o $(\Delta x, \Delta y)$. Pro bod $[x, y]$ se spočte se všemi možnými posuny a funkce $F_{\min}(x, y)$ má hodnotu nejmenších z nich.

Nechť $I(x, y)$ jsou hodnoty pixelů obrázku a x, y elementární posuny z množiny D . Pomocí Taylorova rozvoje aproximujeme $I(x + \Delta x, y + \Delta y)$ jako:

$$I(x + \Delta x, y + \Delta y) \approx I(x, y) + \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^T \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (6.4)$$

Použitím autokorelační funkce dostáváme

$$F_{\min}(x, y) \approx \min_{(\Delta x, \Delta y) \in D} \sum_{(x, y) \in O} (\Delta x, \Delta y) \cdot \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right) \cdot \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^T =$$

$$\min_{(\Delta x, \Delta y) \in D} (\Delta x, \Delta y) \cdot \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (6.5)$$

Kde matice M označuje autokorelační matici.

$$M = \sum_{(x, y) \in O} \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right) \cdot \begin{pmatrix} \frac{\partial I(x, y)}{\partial x} \\ \frac{\partial I(x, y)}{\partial y} \end{pmatrix}^T = \quad (6.6)$$

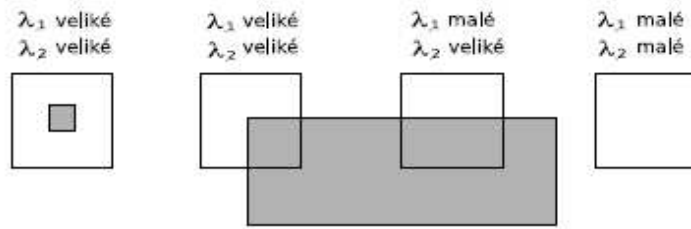
Matice M obsahuje všechny diferenční operátory a její vlastní čísla určují hlavní křivost plochy v okolí bodu (x, y) . Toho se dá využít při rozhodování, zda daný bod je významný a jestli se jedná o hranu či roh.

Vlastní čísla matice M označíme jako α_1 a α_2 a mohou nabývat těchto hodnot:

- Obě jsou relativně malá, potom je okolí bodu relativně ploché a nejedná se o významný bod.
- Jedno vlastní číslo je malé a druhé veliké, potom se jedná o hranu
- Obě vlastní čísla jsou veliká, potom je na okolí bodu významná křivost ve více směrech a jedná se tedy o roh nebo izolovaný bod.

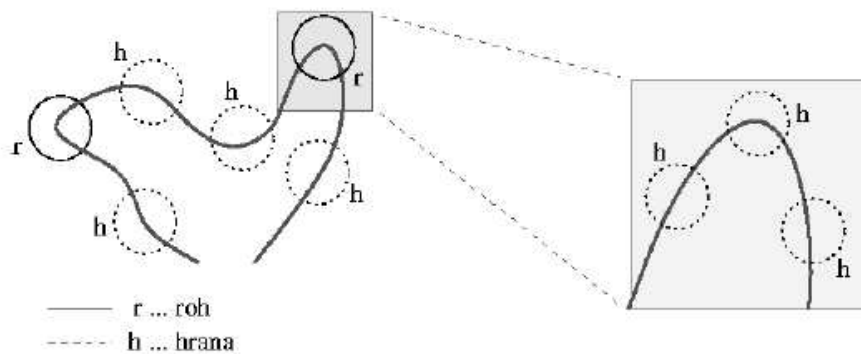
Pro získání významných bodů lze opět použít vyhledání maxim v mapě "rohovitosti" R známé z Moravcova detektoru. Pro její generování se použije determinant a stopa matice M podle vztahu

$$R(x, y) = \det(M) - k \cdot \text{trace}(M)^2 \quad (6.7)$$

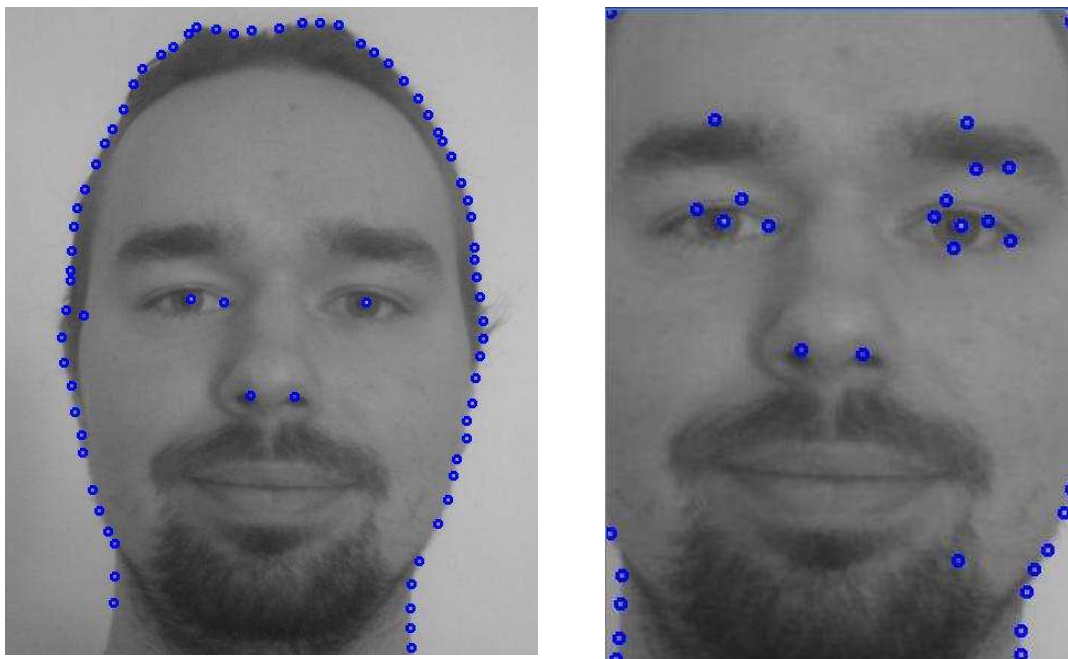


Obrázek 6.4: Vlastní čísla autokorelační matice M podle typu bodu

Použitím autokorelace dosahuje Harrisův detektor výborných výsledků. Je nezávislý na rotaci, posunu a odolný vůči šumu. Se zvětšujícím se nároky na zpracování snímků reálných scén ovšem začalo být důležité vyhledávat významné body i v závislosti na měřítku. Je potřeba aby byly nalezeny stejné obrazové struktury, i když se liší v měřítku a v závislosti na jejich měřítku je poté porovnávat. S tímto problémem si ovšem Harrisův detektor ve své základní podobě nedokáže poradit a musely se tady nalézt jiné metody.



Obrázek 6.5: Znárodnění problému vyhledávání bodů pomocí měřítkové závislého Harrisova detektoru



Obrázek 6.6: Ukázka detekce významných bodů pomocí Harrisova detektoru, vidíme že musíme brát v ohledu měřítko jelikož v obličejí potřebujeme více bodů a na okraji hlavy méně.

Multi-rozsah pro Harrisův detektor [10]

Výpočet druhého momentu matice (někdy označován jako strukturální tenzor) v Harrisově operátoru, vyžaduje derivací obrazu I_x a I_y , stejně jako součet nelineárních kombinací těchto derivací přes lokální okolí. Pro tento účel může posloužit Integrální obraz popsany v kapitole 5.3.

Jádro Gaussiánu, kde t je měřítko

$$g(x, y, t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/2t} \quad (6.8)$$

Vytvoříme měřítkový prostor L za pomoci konvoluce s jádrem Gaussiánu

$$L(x, y, t) = g(x, y, t) * I(x, y) \quad (6.9)$$

Kde L značí měřítkový prostor, g jádro Gaussiánu a I původní obrázek.

Zavedeme že $L_x = \partial_x L$ a $L_y = \partial_y L$ označuje parciální derivace L . Kromě toho zavedeme Gaussovou funkci $g(x, y, s)$ s parametrem s jako integrací stupnice. Poté můžeme vyjádřit druhý moment multi-rozsahové matice jako ...

$$\mu = (x, y; t, s) = \int_{\xi=-\infty}^{\infty} \int_{\eta=-\infty}^{\infty} [X] g(\xi, \eta; s) d\xi d\eta \quad (6.10)$$

Kde

$$[X] = \begin{bmatrix} L_X^2(x - \xi, y - \eta; t) & L_X^2(x - \xi, y - \eta; t) L_Y^2(x - \xi, y - \eta; t) \\ L_X^2(x - \xi, y - \eta; t) L_Y^2(x - \xi, y - \eta; t) & L_Y^2(x - \xi, y - \eta; t) \end{bmatrix} \quad (6.11)$$

Pak můžeme definovat multi-měřítkového Harrise

$$M_C(x, y; t, s) = \det(\mu(x, y; t, s)) - \kappa \cdot \text{trace}^2(\mu(x, y; t, s)) \quad (6.12)$$

Pokud jde o výběr místního měřítka t a parametru integrace s , tyto měřítkové parametry jsou často spojeny s relativním měřítkem integrace γ tak, že $s = \gamma^2 \cdot t$ kde obvykle γ volíme v rozsahu [1,2]. Můžeme tedy detekovat rohy pomocí Harrise a za přispění $M_C(x, y; t, s)$ v libovolném měřítku t .

V praxi je často toto řešení doplněno o výběr kroku měřítka, kde je měřítkový prostor normalizován pomocí Laplaciánova operátoru. Ten je počítán na všech úrovních měřítkového prostoru.

Tím tu vzniká Hartus-Laplaciánův operátor který je počítán přes všechny body, které současně splňují následující dva aspekty:

- maximum v měřítkovém prostoru opatřeném $M_C(x, y; t, s)$
- je v lokálním maximu nebo minimu ve všech měřítkách normalizovaných pomocí Laplaciánova operátoru.

6.3 Transformace SIFT

SIFT je jedna z novějších metod vyvinutá roku 2004. Název je odvozen z použitého principu, transformuje obrazová data do měřítkově nezávislých souřadnic. Byla navržena pro extrakci významných bodů z obrázků a následně spolehlivé vyhledávání společných rysů mez nimi. Svou výkonností už se blíží k možnosti využití v real-time aplikacích pro zpracování obrazu.[15]

Proces získávání významných bodů z analyzovaného obrázku lze rozdělit do 4 fází.

Vytvoření měřítkového prostoru a vyhledání lokálních extrémů:

Vypočítá se měřítkově nezávislá reprezentace obrázku. V něm se dále vyhledávají přes všechna měřítka lokální extrémů. Pro zlepšení výkonnosti je pro generování tohoto prostoru použita aproximace pomocí rozdílu Gaussových funkcí (DoG) na místo Laplacian Gaussovy funkce (LoG).

Přesnější lokalizace klíčových bodů:

Zde se podrobněji zkoumají body nalezené v první fázi. Ty s nedostatečnou stabilitou se vyloučí a u zbylých se pomocí interpolace upřesní jejich poloha v prostoru.

Přiřazování orientací:

Pro zajištění nezávislosti na rotaci, se ke každému bodu přiřadí dominantní orientace na základě orientací gradientů v okolí bodů.

Tvorba deskriptorů:

Na základě orientací gradientů v okolí bodu je pro každý významný bod vypočten deskriptor. Pomocí něj jsou potom porovnávány významné body mezi analyzovanými snímky.

6.3.1 Vytvoření měřítkového prostoru a vyhledání lokálních extrémů

Jako první se data obrazu nejprve převedou do reprezentace nezávislé na měřítku. Tímto převodem se zabývalo mnoho osob (např. Lindeberg, 1993; Lowe, 1999) a všechny výsledky potvrzují, že nejlepší vlastností se dosáhlo za pomoci generování využívající LoG. Jeho použití je ale výpočetně náročné, proto se hledají jednodušší filtry. Nejčastěji je použita aproximace pomocí DoG. Rozdíl Gaussových funkcí je získán odečtením dvou na sobě nezávislých Gaussových funkcí o různém měřítku. Generace se provádí za pomoci konvoluce s obyčejným Gaussem a měřítková forma se získá později rozdílem po sobě jdoucích vrstev.

Vystává ovšem otázka, kolik vrstev se má vytvářet. Tato volba bude mít rozhodující vliv na počet nalezených struktur. Struktury mohou být v měřítku blízko za sebou a při zvolení příliš velkého kroku se některé z nich nenaleznou. Ovšem s množstvím vrstev roste výpočetní náročnost.

Po vygenerování měřítkového prostoru je už možné začít s vyhledáváním významných bodů. Body které by mohli být označeny jako významné se budou vyskytovat na vrstvách jako lokální extrémů. Hledání se provádí ve všech vrstvách kromě krajních (které neobsahují rozdíl dvou vrstev). Každý bod je porovnán se svými

osmi sousedy ve stejné vrstvě a s devíti sousedy v obou sousedních vrstvách. Pokud je vyšší nebo menší než všichni jeho sousedi, je označen jako významný.[13]

6.3.2 Přesnější lokalizace klíčových bodů

Pro zajištění vysoké stability nalezených bodů použitých pro hledání korespondencí mezi obrázky se ukázalo jako výhodné používat přímo celočíselné souřadnice bodů v obrazu a měřítko vrstvy, ve které byl bod nalezen. Brown vyvinul metodu pro upřesnění polohy lokálního extrému pomocí aproximace funkce popisující okolí bodu měřítkovém prostoru a to jak v souřadnicích obrazu, tak také v měřítku.

Pro aproximaci využívá Taylorův rozvoj. Sub-pixelová a sub-měřítková pozice je určena jako extrém rozvoje. Pro vyjádření pozice extrému je funkce derivována a určena offsetem vůči zkoumanému bodu. Pokud je offset větší než 0,5 tak to značí extrém ležící blíže k sousednímu bodu. Klíčový bod je tedy vyměněn k souřadnicím zkoumaného bodu. [11]

Výhody určení polohy na těchto úrovních se projevuje ve vyšších vrstvách, kde už je vzorkovací krok vůči základnímu obrazu relativně veliký.

6.3.3 Přiřazování orientací

Tato část slouží pro dosažení nezávislosti transformace na rotaci obrázku. Získaným bodům z předchozích dvou částí se přiřadí jedna nebo více orientací vypočítaných z okolí daného bodu. Podle měřítka se vybere Gaussem filtrovaná plocha, která je svým měřítkem zkoumanému bodu nejbližší. Tím se zajistí měřítková nezávislost. Pro každý obrázek bodu $Im(x,y)$ se poté pomocí rozdílu pixelů přepočítá velikost gradientu $g(x,y)$ a orientaci $\theta(x,y)$.

$$g(x, y) = \sqrt{(\text{Im}(x+1, y) - \text{Im}(x-1, y))^2 + (\text{Im}(x, y+1) - \text{Im}(x, y-1))^2} \quad (6.13)$$

$$\theta(x, y) = \tan^{-1}((\text{Im}(x, y+1) - \text{Im}(x, y-1)) / (\text{Im}(x+1, y) - \text{Im}(x-1, y))) \quad (6.14)$$

Pro zkoumaný významný bod se sestrojí kruhový histogram orientací, z gradientů vypočtených z jeho okolí. K histogramu s odpovídající orientací se přičte gradient vážený pomocí kruhového okénka s Gaussovským rozložením vah.

Dominantní orientace je určena z vrcholů histogramu. Orientace nejvyššího bodu histogramu představuje i dominantní orientaci bodu. Pokud je detekováno více vrcholů (nad 80%) je vytvořen nový bod se stejným umístěním ale s jinou orientací. Tato úprava

zajišťuje větší stabilitu při hledání korespondencí. Tyto multi-orientační body se vyskytují poměrně málo (15%).

6.3.4 Tvorba deskriptorů

V předchozích fázích jsme získali množinu dobře lokalizovatelných bodů nesoucích důležitou informaci pro pozdější zpracování obrazu. Nyní ke každému bodu z této množiny se přiřadí deskriptor. Deskriptor v sobě nese informace o vzhledu obrazového okolí kolem významného bodu a umožňuje porovnávání s jinými vyhledávanými body a určování korespondencí. Metoda SIFT přišla s novým systémem deskriptoru vycházejícím z modelu biologického vidění. Metoda je založena na popisu okolí bodu pomocí orientací gradientů a umožňuje drobné posuvy v jejich umístění. Není tedy závislá na jejich přesné lokalizaci. To má za následek odolnost proti drobným geometrickým změnám a deformování v obrazu.

Pro zachování nezávislosti na rotaci se využije přiřazených orientací bodů v předchozí fázi. Podle těchto orientací se natáčí orientace gradientů, aby v případě odlišné orientace korespondujícího významného bodu v jiném obrazu byly výsledná orientace gradientů stejná. Pro účely výpočtu se okolí bodu rozdělí na několik čtvercových oblastí a pro každou se sestrojí histogram orientací. Veškeré informace obsažené v těchto histogramech tvoří data deskriptoru. Tyto deskriptory jsou tedy poměrně obsáhlé.

Další nevýhoda transformace SIFT spočívá v tom, že pokud se bod posune tak, že se gradient dostane do jiného okolí a nebo se změní jeho orientace, by mohl být přidán do jiného histogramu (jednoho z histogramů po rozdělení oblasti okolí). [12]

6.3.5 Zhodnocení

Tato metoda zaručuje nezávislost na:

- posunu
- rotaci
- měřítku
- změnách jasu a kontrastu
- drobných deformací

Podle experimentu u [14, str.20] vyhledávání bodů na fotografiích o rozlišeních 640x480 trvalo na počítači s procesorem 2GHz zhruba 5 sekund. Po použití DoG aproximace Laplaciánu se metoda sice urychlila, ale stále není vhodná pro použití v real-time aplikacích.

6.4 Metoda SURF

Speeded Up Robust Features je nejnovější metoda určená pro vyhledávání korespondencí mezi dvěma obrazy. Byla publikována 2006 autory H. Bay.

Název vypovídá že se tvůrci zaměřili na rychlost zpracování, vzali tedy známé způsoby a snažili se zkrátit jejich výpočetní náročnost. V mnoha ohledech se nechali inspirovat transformací SIFT (Kapitola 6.3). Tato transformace ukázala velmi dobré výsledky za použití aproximace Laplaciánu pro generování měřítkového prostoru a představilo (dnes už velmi používané) nové deskriptory. Tyto deskriptory jsou poměrně datově velké a tak je jejich zpracování ve větším množství náročnější. Na tuto část se zaměřili tvůrci metody SURF. [16] [17]

6.4.1 Měřítkový prostor

Vývoj metody SURF probíhal spíše experimentální cestou se snahou o co největší zjednodušení. SURF kombinuje dvě metody. SIFT, který má ale vysoké odezvy i kolem hran a proto se tyto body odliší za pomoci determinantu Hessianu a odfiltrování. Druhou metodu používali Mikolajczyk a Schmid [18], kteří použili Hessian pro lokalizaci bodů v obrázku a pomocí Laplacianu se mu pokusili přiřadit měřítko. Měřítkový prostor je generován za pomoci Hessianu podle vztahu.

$$L(x, y, \sigma) = \sigma^4 \det(H) = \sigma^4 \det \begin{pmatrix} I_{xx}(x, y, \sigma) & I_{xy}(x, y, \sigma) \\ I_{xy}(x, y, \sigma) & I_{yy}(x, y, \sigma) \end{pmatrix} \quad (6.15)$$

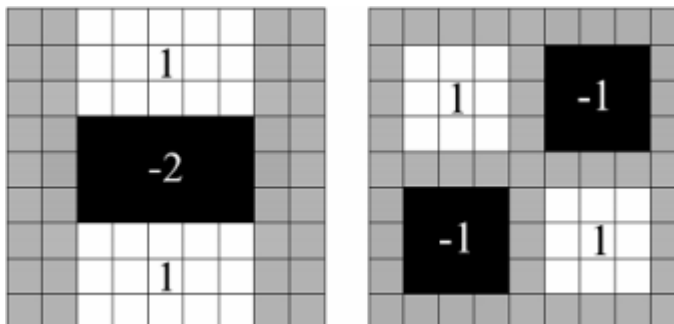
Kde $L(x,y,\sigma)$ představuje bod v měřítkovém prostoru o souřadnicích x,y a měřítku σ . I_{xx} je konvoluce druhé derivace Gaussovy funkce s obrázkem v bodě (x,y) a podobně pro I_{yy} a I_{xy} .

Metoda SURF využívá také Integrálních obrazů popsanych v kapitole 5.3. Pro jejich velmi rychlý výpočet jasu v obdélníku.

6.4.2 Fast Hessian

Podstatnou část výpočtu při hledání významných bodů tvoří konvoluce obrazu s jádry derivovanými od Gaussovy funkce. Snahy o zrychlení se zaměřují hlavně na tuto část úlohy. Rozdíl Gaussových funkcí se snažil o věrohodné napodobení Laplaciánu a přesnost aproximace bylo možné řídit pomocí konstantního faktoru k určujícího měřítkovou vzdálenost generovaných ploch. Pro maximální zkrácení časové náročnosti využili autoři právě integrální obraz, díky němuž se výpočet konvoluce pro libovolně velkou plochu provádí konstantní rychlostí. Metody je založena na výpočtu

determinantu Hessianovy matice a pro její konstrukci jsou potřeba konvoluční jádra pro směr x , y a xy . Tyto jádra jsou podobný Haarovým příznakům. Jejich diskrétní podoby mají tvar. [15]



Obrázek 6.7: Ukázka konvolučních jader pro velikost 9x9

Odezvy od takto konstruovaných jader je nutné ještě měřítkově normalizovat. Pro normalizaci se využívá Froebeniova norma, definovaná nad maticí $m \times n$ jako odmocnina ze součtu absolutních čtverců všech prvků matice. Na obrázku XX tvoří prvky matice jednotlivé hodnoty konvolučních jader. Pro původní jádra generovaná ze spojitých průběhů měřítkově normalizovaných derivací Gaussovy funkce platí, že kolikrát se zvětší měřítkový parametr, tolikrát se zmenší Froebiova norma.

Po vygenerování měřítkového prostoru se významné body najdou jako lokální maxima, stejně jako tomu bylo u metody SIFT. Při konstrukci je možné použít pouze jádra o dané velikosti, což má za následek velký měřítkový krok mezi jednotlivými vrstvami měřítkového prostoru.

6.4.3 Přiřazování orientací

Postup je opět podobný jako u metody SIFT, ale ze získaných orientací kolem významného bodu se uvažuje pouze nejdominantnější směr a ostatní se zanedbávají. Jako struktura pro záznam orientací jsou použita posuvná okénka, orientací pokrývající dohromady 360 stupňů a o velikosti 60 stupňů. Do okének se vkládají hodnoty gradientů získané v kruhovém okolí bodu a vážené pomocí hodnoty Gaussovy funkce centované v souřadnicích významného bodu. Pro zrychlení výpočtu gradientu jsou opět použity obdélníkové aproximace a s výhodou lze tedy použít již vygenerovaný integrální obrázek. [18]

Po vyšetření všech zkoumaných bodů v okolí významného bodu se vytvoří jednotlivé součty dx a dy . Tyto vektory se skombinují s orientací a vybere se z nich ten

největší vektor, ostatní se dále už neuvažují. Orientace tohoto největšího vektoru určuje orientaci významného bodu.

6.4.4 Tvorba deskriptorů

Jedna ze základních problémů SIFT je velikost deskriptorů a tak metoda SURF přichází s novým typem deskriptorů a také s novým způsobem korespondence. Pro popis okolí jsou využity gradienty spočtené na čtvercovém okolí. Tento čtvercový region je poměrně veliký s délkou strany $20s$, s značí měřítko, a je natočený podle orientace zpracovávaného významného bodu. Toto natočení je důležité pro zachování nezávislosti na rotaci obrázku.

Stejně jak v metodě SIFT i zde je použito dělení na sub-oblasti, ale informace v nich jsou jinak zpracovávány. Díky tomuto jsou odolné vůči drobným geometrickým deformacím v rámci sub-oblasti

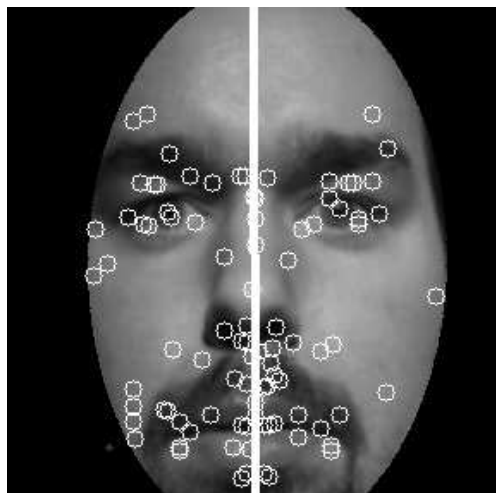
Postup konstrukce deskriptoru:

- sestrojení čtvercové oblasti s natočením podle orientace významného bodu
- celá oblast se rozdělí na 4×4 sub-oblasti
- v každé sub-oblasti vybereme 5 pravidelně rozmístěných bodů
- na vytvořené body se aplikují filtry pro výpočet gradientu
- pro každý bod jsou vypočteny odezvy dx , dy a přičteny k sub-deskriptoru pro danou oblast ... vznikne tak z 16 sub-oblastí 64 sub-deskriptorů

Velikost deskriptoru ovlivňuje rychlost jejich porovnávání při hledání korespondencí a má tudíž vliv na použitelnost metody v real-time aplikacích.

6.4.5 Zhodnocení

Výsledky jsou porovnatelné z metodou SIFT. Rozdíl je v deskriptorech které jsou popsány v kapitole 6.4.4. Deskriptory jsou méně objemné, což slouží pro rychlejší zpracování ale na úkor přesnosti a kvality deskriptorů. Korespondence je zde složitější a méně přesná.



Obrázek 6.8: Ukázka nalezených významných bodů pomocí SURF metody

7 ROZPOZNÁNÍ VÝRAZU TVÁŘE

Abychom mohli vymezit, které body potřebujeme a které ne, musíme si určit stanovený cíl. Chceme abychom mohli vypočítat výraz tváře neznámých osob. Musíme tedy brát v úvahu to že chceme, aby naše aplikace probíhala v real-time čase a fungovala i u neznámých osob. Každý člověk je rozdílný a své emoce projevuje jinak. Pokusíme se tedy sjednotit a nelézt řešení pro identifikaci různých emocí. Mnoho takových aplikací využívá různé metody a principy, které si zde uvedeme. Ale všechny využívají stejné významné body, jen je jinak vyhledávají, definují a následně zpracovávají.

7.1 Způsoby detekce obličejových rysů

Jedna z nejdůležitějších částí je definování, které body reprezentují obličej. Jedná se tu i o kompresi dat (vybrání jen významných a znovu lokalizovatelných bodů)

7.1.1 AAM metoda

AAM vyhledávání se skládá ze dvou modelů. Modelu tvaru a vzhledu. [19][20]

Model Tvaru

Tvar určitého objektu můžeme definovat jako geometrickou informaci, která zůstává po normalizaci pozice, rotace a velikosti daného objektu. Pokud předpokládáme tvar v rovině, tak změny jeho pozice, natočení v rovině a velikosti můžeme nazvat transformací podobnosti.

Tvar tváře můžeme pospat pomocí množiny tzv. Tvarových bodů $p_i = (x_i, y_i)$ kde $i = 1..n$ určující pozici důležitých rysů v obraze tváře. Ze souřadnic x_i a y_i bodů p_i můžeme zformovat nenormalizovaný tvarový vektor $x = (x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n)$. Metoda obsahuje několik takových vektorů pro trénování.

Při vyhledávání je zapotřebí provádět transformaci podobnosti (zarovnání dvou tvarů x_A a x_B) která by minimalizovala $E = |T(x_A) - x_B|^2$. Pro 2D obraz a jeden bod (x, y) tvaru x_A můžeme definovat transformaci podobnosti následovně:

$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} z \cos \theta & z \sin \theta \\ -z \sin \theta & z \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_p \\ y_p \end{pmatrix} \quad (7.1)$$

Kde parametr θ a z určují natočení a velikost zmenšení/zvětšení. Parametry x_p a y_p určují posunutí jednotlivých bodů tvaru. Pokud zavedeme substituci $a = z \cos \theta$ a $b = z \sin \theta$, tak dostaneme následující minimalizační problém:

$$E(a, b) = |T(x_A) - x_B|^2 = \sum_{i=1}^n (ax_{A,i} - by_{A,i} - x_{B,i})^2 + (bx_{A,i} - ay_{A,i} - y_{B,i})^2 \quad (7.2)$$

Řešením, které minimalizuje $E(a, b)$ je následující rovnice:

$$\begin{aligned} a &= \left(\sum_{i=1}^n x_{A,i} x_{B,i} + y_{A,i} y_{B,i} \right) / \left(\sum_{i=1}^n x_{A,i}^2 + x_{B,i}^2 \right) = \frac{x_A \cdot x_B}{|X_A|} \\ b &= \left(\sum_{i=1}^n x_{A,i} y_{B,i} - y_{A,i} x_{B,i} \right) / \left(\sum_{i=1}^n x_{A,i}^2 + x_{B,i}^2 \right) = \frac{1}{|X_A|} \left(\sum_{i=1}^n x_{A,i} y_{B,i} - y_{A,i} x_{B,i} \right) \end{aligned} \quad (7.3)$$

Pomocí metody PCA a množiny normalizovaných tvarů můžeme vytvořit statistický model tvaru lidské tváře pomocí které dokážeme tvar tváře reprezentovat podstatně menším počtem hodnot.

Model Vzhledu

AAM model vzhledu (Active Appearance Model) je vytvořený na základě kombinace modelů tvaru a modelu textury

Model textury je vytvořený z normalizovaných textur příslušejícím jednotlivým tvarům použitým v procesu trénování. Normalizace textur má dvě části. První je normalizace rozměru a vzhledu textury. Ve druhé části je upravovaný globální vliv osvětlení.

Plocha obrazu, které zabírá objekt tváře se dá popsat prostřednictvím vektorů tvaru X . Prostřednictvím těchto vektorů, procesu triangulace plochy tváře a metod deformace obrazu dokážeme do jisté míry normalizovat vzhled textury a přizpůsobit ji tak, aby

zodpovědně našla střed (přesné umístění). Hodnoty intenzit tvořících texturu vytváří vektory, pomocí kterých můžeme následující transformací upravit globální vliv osvětlení.

Oba modely, tvarový i texturový, můžeme sloučit použitím další PCA. Aplikováním tak odstraníme korelaci mezi vektory parametrů tvaru a textury.

Zhodnocení

Vyhledávání pomocí této metody je iterativní proces, který má určitý počet prověřovaných „masek“. Díky tomu je tato metody dosti pomalá a pro zpracování v reálném čase dosti nepraktická.

7.1.2 Model SVM

[21][22]Problém klasifikace můžeme nadefinovat jako hledání pravidla, pomocí kterého dokážeme zařadit objekt do jedné z daných výstupních tříd. Přičemž hledání probíhá na základě příkladů, daných v trénování množině. V nejjednodušším případě uvažujeme jen o zařazování do dvou výstupních tříd. Tato úloha se dá formalizovat jako snaha o nalezení rozhodovací funkce, která patří do třídy funkci F . Hledáme ji na základně vstupně výstupních trénovacích párů nazývaných trénování vzorky, tak aby funkce dokázala korektně klasifikovat i testovací vzorky. Trénování i testovací vzorky jsou přímo generovány z toho stejného neznámého rozdělení pravděpodobnosti a distribuční funkcí $P(x,y)$. Po nalezení klasifikační funkce jsou vzorky přiřazené k třídám. Hledání optimální rozhodovací funkce je přitom úspěšné pokud minimalizujeme tzv. předpokládanou chybu.

Bohužel hodnotu předpokládané chyby nemůžeme minimalizovat přímo, když nepoznáme distribuční funkci $P(x,y)$. Proto se při hledání funkce, která se bude aspoň přibližovat k optimální funkci snažíme využít informace, které máme k dispozici a to sice charakter trénovacích dat a vlastností třídy funkcí F .

Za poměrně jednoduché řešení problému můžeme považovat aproximaci předpokládané chyby prostřednictvím tzv. empirické chyby. Pokud se počet trénovacích vzorků n blíží k nekonečnu, tak se hodnota empirické chyby přibližuje k hodnotě předpokládané chyby. V tomto případě ovšem hrozí, že pokud bude třída funkcí F příliš komplexní, tak v konečném důsledku dojde minimalizací empirické chyby k přeučení a klasifikátor bude mít zlé generalizační vlastnosti.

7.2 Metody rozpoznávání výrazu tváře

7.2.1 Nejjednodušší metody rozpoznávání výrazu

Počítá s jednoduchými rozdíly v čase. Tato metoda ovšem vyžaduje abychom přistoupili ke kameře s „neutrálním výrazem“. Toto kritérium ale nemůžeme nikdy určitě splnit. Pro lepší vyobrazení a pochopení se podívejme na následující fotky.



Obrázek 7.1: Ukázka různých deformací obličeje

Uvažujme jakoby o počátku souřadnic významných bodů neutrální obličeje. Při pohybu koutků úst směrem k okrajům obrazu a mírně nahoru může být definován jako úsměv ale mnoho lidí se usmívá různými způsoby.



Obrázek 7.2: Druhy úsměvu

Jediná vlastnost která zůstává pro všechny totožná je prodloužení oblasti úst.

Například překvapení se u některých osob vyjadřuje pomocí otevření úst, zvednutí obočí tím i zvětšení očí. Osoba k tomu sama může uskočit nebo se celkově velmi rychle pohnout. V tomto případě by opět mohli použít vzdálenost jednotlivých víček, detekce otevření úst nebo vzdálenost obočí od očí. Skombinováním těchto jevů by mohli vytvořit jednotnou závislost, která by při splnění zvyšovala pravděpodobnost jednotlivých výrazů.



Obrázek 7.3: Překvapení

7.2.2 Rozpoznávání za pomoci masek

Další možnost jak rozeznávat výrazy jen za pomoci předem definovaných šablon. Tato metoda obsahuje několik šablon.

Tyto šablony se poté přiloží k obrázkům s detekovanými body nebo detekcí hranic za použití určitých filtrů jako Canny a další. Tato šablona se po obrazu posouvá a své správně umístění hledá pomocí nejmenší odchylky vypočítané od nalezených bodů nebo detekované hranice filtrem. Tímto způsobem ovšem může vzniknout chyba pro případnou pozdější rozpoznání výrazu. Výraz se totiž poté vyhodnocuje pomocí šablony, která měla nejmenší odchylku od obrázku.

Z popisu také vyplývá že se musí ještě pro případy různého přiblížení měnit měřítko šablony a to ještě za předpokladu že se uživatel dívá do objektivu přímo. Tato metoda je časově dosti náročná a nepřesná. Dá se využít například u pasových fotek nebo fotek kde je známá vzdálenost člověka od objektivu.

Velmi podobný způsob využívá AAM model pro definování pozice obličeje a významných bodů.

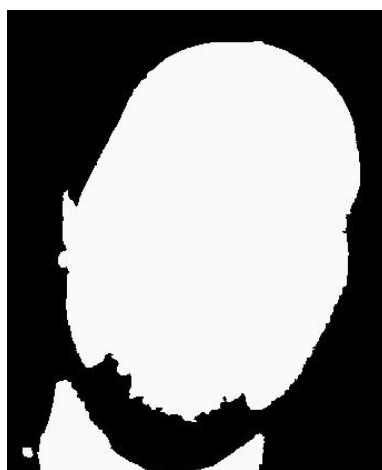
7.2.3 Rozpoznávání za pomoci neuronových sítí

Toto řešení čerpá z předešlých kapitol ale otázku samotného rozpoznávání výrazu už nechává na neuronové síti. Uživatel si sedne před kameru. Program nalezne významné body v obraze a pomocí nich se učí a určuje samotný výraz uživatele. Některé metody se učí samy za pomoci definování změn popsanych v předešlé kapitole nebo si uživatel sám (po prvním spuštění pro naučení neuronové sítě) odsouhlasí nebo nastaví kdy se usmívá a jiné emoce. Neuronová síť se to pak jen naučí a následně předpovídá. Je zde ale předpokládáno že počítač nebo program používá omezený počet lidí. Tato metoda není určena pro rozpoznávání výrazu tváře u neznámých osob ale u stále se opakujících.

8 EXPERIMENT

8.1 Postup

Moje diplomová práce začala vypůjčením potřebným zařízením a sestavení aparatury podle kapitoly 2. Pomocí sestavené aparatury jsem si vyfotografoval několik fotek pro různá nastavení fotoaparátu a osvětlení. Po otestování několika vybraných modelů (viz. Kapitola 3.) jsem nakonec vybral barevný prostor YCbCr. Jako dostačující se ukázali také modely YIQ a Lab. Poté jsem provedl segmentaci kůže (viz kapitola 4.). Pro vymezení části obličeje jsem zvolil metodu popsanou v kapitole 5, za pomoci Adaboostingu a Haarových detektorů jsem dostával velmi přijatelné výsledky i s dostatečnou rychlostí.



Obrázek 8.1: Ukázka správné segmentace kůže

Pro následné zpracování jsem nepotřeboval znát umístění obličeje vůči celkovému rozložení scény a tak pro lepší klasifikaci a rychlejší zpracování jsem si oblast obličeje

vyřízl a pracoval už jenom s ním. Tímto řešením můžeme porovnávat velmi jednoduše pozici významných bodů jelikož nemusíme přepočítávat pozici vůči celkovému zornému poli kamery. Změny ve výřezu nastávají pouze s přiblížením nebo oddálením uživatele od kamery. Tento efekt ovšem nenastává příliš často tak ho můžeme teoreticky zanedbat, ale pro lepší výsledky jsem omezení vztáhnul na velikost nalezeného obličeje (viz kapitola 8.1.1).

Dále bylo zapotřebí nalézt správnou metodu pro detekci významných bodů. Vyzkoušené metody jsou popsány v 6. kapitole. Moravec se ukázal jako dobrý detektor ale velmi pomalý, takže jeho využití real-time aplikacích můžeme zavrhnout. Hartus byl v detekování lepší, ale jeho rychlost nebyla o moc lepší než Moravec. Jeho využití tedy nebylo vhodné. Poté jsem vyzkoušel transformaci SIFT. Tato metoda dávala velmi dobré výsledky, ale díky velké velikosti deskriptorů byla jeho výpočetní náročnost velmi vysoká. Poslední metoda SURF se ukázala jako dostatečně rychlá, ale na úkor efektivnosti deskriptorů. Tato metoda definovala opravdu hodně významných bodů, které musejí být určitým způsobem omezeny. (kapitola 8.1.1 a kapitola 8.1.2)

Po omezeních významných bodů už zbývalo pouze nalézt nějakou charakteristiku jednotlivých výrazů, definovat je a zavést do algoritmu.

8.1.1 Omezení významných bodů pro definici výrazu tváře

Metoda SURF nám našla mnoho významných bodů, které je nutno omezit. K zredukování počtu nalezených významných bodů jsem přistoupil ze dvou pohledů. První pohled bude popsán této kapitolou a druhý v následující kapitole.

Z postupu můžeme vidět, že nyní pracujeme pouze s výřezem obličeje. Po mnoha pokusech s tímto algoritmem jsem se ubezpečil, že obličej bude stále uprostřed vyříznuté oblasti. Díky tomuto kroku můžeme body omezit vůči okrajům výřezu.

Napřed si popíšeme základní znalosti o obličejí. Celý obličej se skládá z čela, očí, nosu, úst a brady. Tyto části mají pevné umístění. Není člověk který má například ústa nad nosem neb jiné rozložení obličeje. Můžou se vyskytnout výjimky s ohledem na velikost, tvar nebo trošku jiné rozložení. Někteří lidé s nádory na obličejí atd. Tyto nemoci jsou ovšem dosti náhodné a nepravděpodobné, tak je můžeme zanedbat.

Rozdělil jsme si tedy výřez na oblasti ve kterých se budou hledat body zastupující určité části obličeje. Obličej jsme si rozdělili na levé oko, pravé oko a ústa. Pomocí jejich deformací se můžou určit následné výrazy tváře. Jelikož tu musíme brát v úvahu velikost nalezeného výřezu, který se mění s přiblížením obličeje ke kameře, jsou omezení ustálena vůči samotné velikosti detekovaného obličeje.

V mé práci jsem si meze nastavil:

- Vertikální okraje: od 1/5 do 4/5 šířky obrazu výřezu (v rovině X)
- Definování oblasti očí:
 - levé – od 1/2 do 3/4 v rovině X a více než 1/2 v rovině Y
 - pravé – od 1/4 do 1/2 v rovině X a více než 1/2 v rovině Y
- Část úst: od 1/2 do 7/8 osy Y

Tyto rozmezí jsou znázorněna na následujícím obrázku 8.2.



Obrázek 8.2: Ukázka vymezení oblastí pro definici bodů zastupující oči (obrázek vlevo) a zastupující ústa (obrázek vpravo)

8.1.2 Symetrie

Další vlastností obličeje je jeho symetričnost. Žádný obličej není zcela symetrický díky přírodním zákonům a životnímu stylu (například jizvy). Ale s ohledem na významné body na obličeji jsou tyto body symetrické. Zkusili jsme tedy přistoupit k tomuto způsobu řešení čistě geometricky (matematicky). Musíme počítat, že body mohou být různě od sebe vzdáleny pod různým úhlem.

Napřed jsem zkusil jednodušší geometrický způsob detekování symetrie:

Určení středu celého obličeje a souřadnice dvou porovnávaných bodů

Střed můžeme zjistit velmi jednoduše za pomoci vysegmentované části a následného výpočtu pomocí momentů nebo si určit pevný bod na výřezu obličeje.

$$x_c = \frac{M_{10}}{M_{00}} \quad y_c = \frac{M_{01}}{M_{00}} \quad (8.1)$$

Kde M_{xy} jsou momenty vypočítané z segmentované oblasti

Vypočítáme si vzdálenost obou bodů od středu

$$vzd = \sqrt{(x_c - x_b)^2 + (y_c - y_b)^2} \quad (8.2)$$

Porovnáme vzdálenost

Pokud je vzdálenost od středu stejná víme že se body nacházejí na kružnici o nalezeném poloměru.

Zjistíme osu symetrie

Zjistíme si úhel mezi body a středem. Osa by měla vést středem tohoto úhlu. Druhým způsobem určení je vypočítat střed úsečky definované dvěma prozkoumávanými body ležícími na kružnici. Nalezenou osu protáhneme nalezeným bodem a středem kružnice.

Otestujeme nalezenou osu s osou vysegmentované kůže

Porovnáme nalezenou osu s osou vypočtenou pomocí momentů.

$$\alpha = \arctan\left(\frac{2 * \mu_{11}}{(\mu_{20} - \mu_{02})}\right) \quad (8.3)$$

Kde μ_{xy} jsou centrální momenty

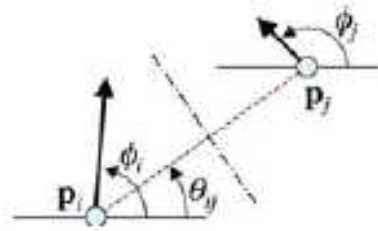
Konečná podmínka

Pokud je i tato podmínka splněna našli jsme Symetrické body pokud ne přejdi na další body.

Zhodnocení

Tato metoda je matematicky jednoduchá, ale nastává zde problém s použitím momentů. Momenty vysegmentované části neměli dostatečnou přesnost a často se pohybovali jen díky menším odchylkám v segmentaci způsobené pohybem stínů a jiných věcí, které zapříčinili rozdíly v jednotlivých snímacím při segmentování. Tento způsob se tedy ukázal jako nevyhovující.

Další způsob byl za využitím deskriptorů metody SURF popsané v kapitole 6.4. Každý body je v SURF definován jako $p_i(x,y,\Phi,s)$ kde x a y popisují souřadnice daného bodu, Φ orientaci okolí bodu a s jeho měřítko. Pomocí těchto vlastností můžeme velmi dobře vypočítat body které jsou symetrické.



Obrázek 8.3: Ukázka dvou bodů popsaných deskriptory

[28] [29] Pokud je obličej symetrický znamená to, že jedna jeho polovina je zrcadlovým obrazem druhé poloviny. Z toho vyplývá, že u deskriptorů metody SURF bude orientace dvou zkoumaných bodů taky zrcadlově otočená, ale měřítko bude stejné. Z těchto důvodů můžeme odvodit následující.

Váhová úhlová symetrie Φ_{ij} dosahuje hodnot $[-1, 1]$ vypočteme jako

$$\Phi_{ij} = \cos(\varphi_i + \varphi_j - 2\theta_{ij}) \quad (8.4)$$

Kde φ znázorňuje orientaci jednotlivých úhlů a θ úhel svírající spojnice mezi body s osou x (zobrazeno na obrázku 8.3). Váhové měřítko dosahuje hodnot S_{ij} dosahuje hodnot $[0, 1]$ kvalifikuje relativní podobnost dvou vektorů v měřítku.

$$S_{ij} = \exp\left(\frac{-|s_i - s_j|}{\sigma_s(s_i + s_j)}\right)^2 \quad (8.5)$$

Kde σ_s ovládá vektor měřítkových možností, které budou přijaty. V mém u jsem použil nastavení $\sigma_s = 1$.

V některých publikacích používají ještě jednu váhu a to pomocí vzdálenosti. Vzdálenější body od osy jsou méně důležité než ty bližší. V našem případě by se jednalo například o to, že by nosní dírky měli mít větší váhu než okraje očí. Tyto všechny tři váhy nakonec navzájem vynásobí a porovnávají jednotně. Z mého experimentu se ukázalo, že lepších výsledků se dosahuje pokud jsou jednotlivé váhy porovnávány odděleně. Díky tomu vysoká hodnota jedné neovlivní druhou. Je zapotřebí ovšem určité množství metodiky a trpělivosti. Hodnoty v programu jsem nastavoval empiricky pomocí vyzkoušených hodnot.

Takto jsme získali několik možných os mezi všemi možnými dvojicemi. Kdybychom znali úhel obličej (například pomocí momentů) porovnali bychom nalezené úhly a jejich umístění s úhlem obličej. Vypočítaný úhel podle momentů ovšem často skákal a byl nepoužitelný pro aplikaci. V některých publikacích vyhledávali hlavní osu pomocí Hooghovy lineární transformace. Toto hledání ovšem trvalo poměrně dlouhou dobu (jeden snímek 1 vteřinu). Pro urychlení a zjednodušení jsem omezil rotaci (naklonění) obličej pouze o malý úhel. Nepočítám zde tedy s možností, že člověk například před kamerou leží a opírá si hlavu nebo ji má nakloněnou pod vyšším úhlem než je 10° .



Obrázek 8.4: Ukázka os jednotlivých dvojic(vpravo), osy omezené úhlem tváře vypočítaným pomocí momentů (vlevo)

8.1.3 Definice výrazů

Usměv

Pokud se podíváme na obrázky 7.2 můžeme vidět že charakteristickým znakem úsměvu je rozšíření úst. Výjimky se vyskytují naprosto všude, ale převládá sociální předpoklad že tento argument je pro úsměv typický. Proto jsem jako úsměv definoval určité rozšíření úst. Bohužel každý člověk má jinak široká ústa a tak tento algoritmus není přímo ideální. Za předpokladu že by uživatel přišel ke kameře s neutrálním výrazem šlo by uložit počáteční hodnoty a podle nich zkorigovat jednotlivé výrazy, ale s tímto předpokladem nemůžeme počítat. Proto mi přijde jako nejlepší řešení přirovnávat to k poměru šířce hlavy.

Smutek

Smutek nejlépe člověk identifikuje z očí. Ale ne za pomoci čistě vzhledových vjemů. Člověk to vycítí. Oči mohou být rozšířeny, ale v mnoha případech zase nemusí. V některých náznacích se může přibližovat k překvapení. Člověk může i plakat a lomení úst by mohlo mít opačný znak půlměsíce než je u úsměvu, ale to se vyskytuje už poměrně dosti vzácně.

Překvapení

Z obrázku 7.3 můžeme vidět typické ukázky výrazu překvapení. Typickými znaky jsou rozšíření očních víček, zvednutí obočí a otevření úst. Tato definice je ovšem trochu zkruslená trochu melancholičtí nebo klidní lidé nevykazují třeba žádný z těchto aspektu a tváří se stále klidně. Proto je velmi obtížné definovat výraz překvapení.

Zlost

Typickým znakem zlosti nebo zuřivosti je zúžení očních víček, zvýraznění obočí (přiblížení obočí k očím) a v některých případech i ukázaní zubů. Tento fakt je v lidském charakteru zakódován už od počátku lidské rasy. Tyto rysy jsou ale dnes stále víc potlačovány a vznikají už různé znázornění vzteku.

Malátnost

Malátnost by se dala předpovídat ze všech výrazu nejhůře. Člověk je pokleslý unavený a pomalý. Proto by tento výraz nejvíce odpovídal neutrálnímu výrazu.

8.2 Program

Program jsem psal jednoduše krok po kroku pro dobrou přehlednost, jednoduchost a rychlé doladování.

Jednotlivé části programu.:

- a) Samotná deklarace, načtení knihoven a vytvořená třída pro práci se snímky
- b) Inicializace proměnných, odebrání snímku a jejich vlastností (výška, šířka)
- c) Vyhledávání barvy kůže (Kapitola 4.)
- d) Vyhledání krajů nalezené kůže a vyplnění vnitřní oblasti
- e) Detekce obličejů (Kapitola 5.)
- f) Detekce významných bodů (Kapitola 6.)
- g) Omezení bodů
 - nalezení dvojic a jejich symetrie (Kapitola 8.1.2)
 - korespondence
- h) Rozpoznání výrazu tváře (Kapitola 7.)

- a) Deklarace, knihovny, třída

Pro funkčnost programu a funkcí jsem načel řadu knihoven a definoval konstantu π . Hned poté se nastavuje proměnná „KAM“, pokud je nastavena na 0 načítá se videosekvence, pokud 1 program automaticky vyhledá kameru a spustí program. Jelikož algoritmus využívám kaskádovou detekci s AdaBoosting algoritmem potřeboval jsem alokovat paměť pro výpočty „storage“, definoval jsem si Haar klasifikátor využívající Haarových příznaků.

Pak mám vytvořenou třídu pro operace s obrázky a jejich indexaci pomocí názvů složek barev. Struktury jak pro RGB tak YCbCr kvůli konvertování a černobílý obrázek pro segmentaci. Na konec deklarace matematických funkcí cosinus a sinus za použitím knihovny „math.h“.

b) Inicializace proměnných

Samotný program začíná deklarací potřebných proměnných

- arg1, arg2, arg3 -
- vzdálenost – určuje vzdálenost od středu elipsoidu a je to výsledek rovnice elipsoidu
- i,j,n – jsou pouze pomocné proměnné pro procházení „for“ cyklu
- stred[3] – jedná se o souřadnice středu elipsoidu
- osy[3] – udávají délku os elipsoidu
- prah – je hodnota prahu pro prahování

I když ještě nejsme v cyklu pro odebírání jednotlivých snímků, jeden odeberu pro načtení hlavních vlastností. Využil jsem k tomu funkci „cvCaptureFromCAM“ Výsledek jsem uložil do „capture“, které jsem hned poté otestoval pro jistotu, že se obrázek skutečně načel z kamery. Poté jsem ho přiřadil do mé třídy pro jeho zpracování a vyčtení vlastností. Vlastností jako výška („height“), šířka („width“), hloubka („depth“) a počet barev („nchannels“).

Vytvořil jsem si místo také pro kopii originálního obrázku, černobílý obrázek pro segmentaci. Definoval barvu pro vyplňování oblastí kůže u segmentace a alokoval místo pro uložení nalezených okrajů kůže („storage“).

Zde začíná opakující se cyklus pro jednotlivé snímky

c) Vyhledávání barvy kůže

Teorie k této části je popsána v kapitole 3. Znovu odebírám jednotlivé obrázky z videokamery a pomocí funkcí řečených v předchozí části b). Kopíruji originální obrázek pro zachování celkového obrazu. Poté převádím kopii (pracovní obrázek) do barevného prostoru YCbCr za pomocí funkcí OpenCV „cvCvtColor“.

Vytvořil jsem si dva „for“ cykly pro procházení obrázku a vypočítal funkci elipsoidu. Pokud byl výsledek větší než 1, pak bod leží mimo daný elipsoid, takže na danou pozici do černobílého obrázku uložíme 0. Pokud byl výsledek menší nebo roven 1, už se bod vyskytoval v elipsoidu. Pro zjednodušení jsem zde obrátil značení vzdálenosti (původní: 0...střed, 1...povrch elipsoidu; změna - prohození). Pak jsem to už jen prahoval mnou definovaným prahem v inicializační části.

d) Vyhledávání okrajů kůže

Zde jsem využil opět nabídnutou funkci OpenCV „cvFindContours“, která nalezne jak vnitřní tak i vnější okraje (okraje a i vnitřní díry). Jelikož nepotřebuji znát umístění nosních dírek, úst, očí atd. potřebuji pouze obličej našel jsem funkci pro vyplnění vnější oblasti „cvDrawContours“ mnou definovanou barvou. Pak už jen zbývá segmentovanou barvu kůže zkombinovat s kopií originálního obrázku.

e) Detekce obličeje

Teorie je popsána v kapitole 5. Vytvářím si nový obrázek pro hledání obličeje, dva body reprezentující umístění „pt1“, „pt2“. Načítám Haar kaskádu, tu poté testuji jestli byla skutečně načtena. Alokuji místo pro výpočty a pak ji pro jistotu promazávám.

Zde začíná „for“ cyklus pro hledání obličeje, využiji k tomu funkci „cvHaarDetectObjects“ jejíž výsledky nalezení obličeje mi poslouží pro vyřiznutí obličeje a tedy omezení dat (výřez jsem pojmenoval „cropGRAY“ a uložil v šedotónových barvách pro následné zpracování).

f) Detekce významných bodů

Pro vyhledávání významných bodů jsem využil metodu SURF (Kapitola 6.4). Tato metoda je v prostředí OpenCV zakomponována pomocí funkce „cvExtractSURF“. Pro zviditelnění jsem si při testování vykresloval všechny nalezené body.

g) Omezení bodů

- nalezení dvojic a jejich symetrie

Napřed jsem si definoval velikost výřezu „VH“ pro výšku a „VW“ pro šířku. Dále jsem pomocí „for“ cyklů projížděl významné body a zároveň s tím je omezoval v okrajích. Ujistil se že se nebude testovat bod sám se sebou a poté vypočítal souřadnice středu jejich os. Vypočítal pomocí deskriptorů jejich váhy (viz kapitola 8.1.2). Odpovídající váhy jsem testoval a nejlépe dopadly váhy mající větší hodnotu než 0,5. Jelikož jsem nestihl dořešit problém větších náklonů hlavy, zvolil jsem dál jen určité rozmezí středů od X-ové souřadnice středu obličeje a maximální vzdálenost bodů od sebe na 2/5 šířky obrazu.

Vymezené body ukládám do pole „Actual“.

Pokud se jedná o první snímek, pole „Actual“ se uloží do „Past“ pro další zpracování korespondence.

- korespondence

Jako první jsem zkusil vypočítat korespondenci s ohledem na postup popsany na přednáškách, ale ten se ukázal dosti výpočetně náročný, tak jsem přešel na korespondenci za pomoci deskriptorů. Nejedná se tu o korespondenci dvou statických obrazů, takže hodnoty nebudou na tolik stejné a je zapotřebí zde hodit určitá omezení rozmezí, kdy se jedná o tentýž bod. Pokud bod splnil korespondenční podmínky byl uložen do pole „Nalezene“ a z těchto bodů už určujeme samotné výrazy.

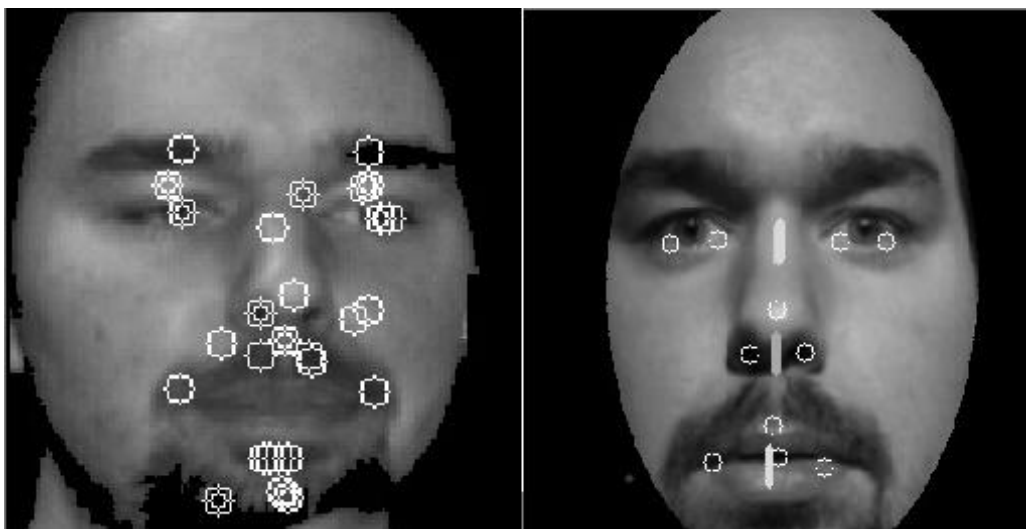
h) Rozpoznání výrazu tváře

Zde jsem vyhledal požadované okrajové body. U oblastí očí maximální a minimální hodnotu souřadnice Y. V oblasti úst souřadnice X. Napřed jsem pomocí pevně stanovených prahů definoval, kdy se uživatel usmívá a kdy se tváří neutrálně. Po testování jsem usoudil, že zavedením pravděpodobnosti bychom mohli vymežit případy, kdy jednotlivé části vykazují náznaky různých výrazů. Pomocí velikosti deformace se nastaví pravděpodobnosti výrazů. Pokud tedy úsměv bude výraznější algoritmus mu dá větší váhu a definuje jej jako hledaný výraz.

Na konci programu už jen vykresluji body po korespondenci, osu a do konzole zapisuji pravděpodobnosti jednotlivých výrazů. Program se ukončí po stisku klávesnice ESC.

8.3 Výsledky

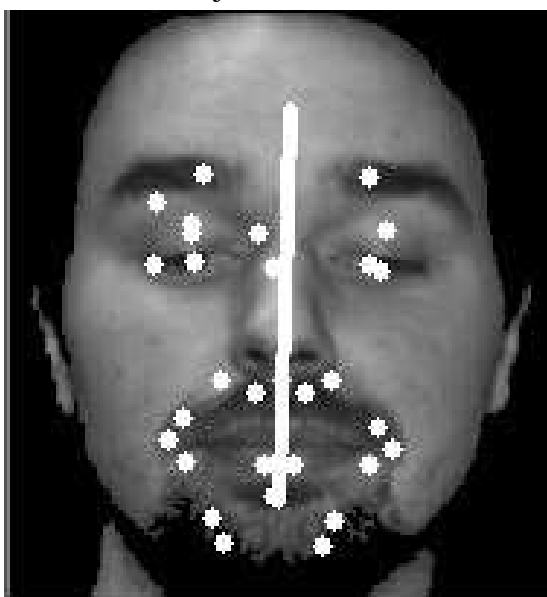
Metoda SURF se ukázala dostatečně rychlá pro aplikace v reálném čase a po omezení jsme dosáhli velmi dobrých výsledků s ohledem na definici významných bodů.



Obrázek 8.5: (Vlevo) ukázka vybraných významných bodů za pomoci pozičního a symetrického omezení (Vpravo) Začátky testování na videu se stacionárním obrazem.

Nejllepších výsledků jsme dosáhli při stacionárních snímcích (Zobrazeno na obrázku 8.4. vpravo). Nemůžeme ovšem počítat s tím, že člověk bude naprosto nehybný a tak musíme řešit korespondenci definovaných bodů které jsou zobrazeny na obrázku 8.4 vlevo.

Metoda SURF ovšem generuje velký počet významných bodů a i po omezeních vznikají jisté odchylky. Při pohybu jsou generovány nové body a některé se zase ztratí. Pro jejich znovu lokalizaci jsem vždy porovnával dva snímky jdoucích po sobě. Při definování jsem se snažil, aby opravdu významné body pro identifikaci výrazu tváře se vyskytovali na každém snímku. Tohoto úkolu jsem dosáhl s přesností 90%-95%, proto se může na jednom z 10 snímků vynechat významný bod, ale po následujících dvou snímcích se opět najde a řeší se jeho korespondence. Ukázku korespondujících bodů a vykreslení osy můžete vidět na následujícím obrázku 8.6.



Obrázek 8.6: Ukázka vykreslení sledovaných bodů a osy obličeje vypočtené pomocí nalezených bodů

Můžeme si zde všimnout i menších chyb, bodů které nemají svůj vlastní bod s ohledem na osu symetrie. Domnívám se, že je to způsobené nalezené dvojice, kde mají jeden bod společný.

Tyto chyby jsou ovšem zanedbatelné a pomocí těchto bodů a omezeného vyhledávání popsaného v kapitole 8.1. můžeme už definovat výrazy tváře. Za pomoci pevně stanovených hranic jsme dosahovali dobrých výsledků, ale problém byl ve velmi jemném přeskakování kolem hranice. Se zavedením pravděpodobnosti s ohledem na velikosti se výsledky trochu zlepšili, ale nastal tu problém s určením, při jaké pravděpodobnosti se člověk skutečně usmívá a kdy ne. Tato pravděpodobnost zde byla taky zavedena pro odlišení více výrazu od sebe. Z hlediska náročnosti a složitosti, nebylo možné nadefinovat rozeznávání dalších výrazů.

9 ZÁVĚR

Cílem této práce bylo navržení metody pro identifikaci výrazu tváře z videosekvence u neznámých osob. V kapitolách 3 a 4 je popsáno získání subprostoru barev odpovídající modelu lidské kůže. Seznámil jsem se zde s barevnými prostory, jejich vlastnostmi a také jejich využitím v praxi. Kůže je ovšem velmi nejednotvárná s hledem na její barvu. S využitím matematicky popsatelného geometrického tělesa jsem chtěl dosáhnout odstranění problému při vyhledávání pokožky za pomoci barvy

Problém byl v označení ne jen barvy lidské kůže, ale i barev podobných, jako jsou některé dřeviny nebo barvy blízké v daném barevném prostoru. K tomuto nedostatku se také vztahuje nutnost znalosti daného osvětlení. Obeznámil jsem se tedy s vlivy týkajícími se problému s osvětlení scény. Po použití vyhledaného geometrického tělesa došlo k omezení těchto problémů, ale ne zcela k jejich odstranění. Z důvodů obsazení mnoha barev na lidské pleti se podařilo odstranit všechny chybné segmentace. S ohledem na osvětlení se ukázali být výsledky přijatelné, až na případ snímání barvy lidské kůže při denním osvětlení a nastavení snímací aparatury na osvětlení pomocí žárovky. Pokud se podíváme na obrázek (4.2) můžeme vidět, že tyto nastavení způsobí velký rozdíl v barvě kůže. Při samotném pohledu člověka už ani on sám nerozpozná jedná-li se je o lidskou kůži či ne. K problému některých barev se ukázal model být kvalitní s výjimkou světlých dřevin. Tento problém ovšem nemůže být odstraněn pomocí detekce lidské kůže využívající barvy, jelikož lidská kůže obsahuje prvky barvy světlých dřevin, kde se tento prvek barvy vyskytuje převážně jako homogenní většinová barva.

Kvalitní segmentace kůže je základem pro následující detekci obličeje, pro kterou je využita metoda Adaboosting algoritmu s pomocí Haarových příznaků. Tato metoda se ukázala poměrně přesná a účinná ovšem za omezení popsaných v kapitole 5. Jedno z nich je například aby dotyčná osoba se vyskytovala naproti kameře a upínala pohled směrem ke kameře (pro snímání celého obličeje a ne jen určité části).

Po splnění vyhledávání obličeje, bylo možné přejít na vyhledávání významných bodů v obličeji. Zde bylo napřed potřebné si uvědomit, které významné body budeme pro identifikaci výrazu tváře potřebovat. Seznámil jsem se zde se základními detektory jako jsou Moravcův a Harrisův detektor významných bodů, jejich principu, vlastnostmi a nedostatky. Výsledky jednotlivých vyhledávání významných bodů můžete vidět na obrázcích (6.4) a (6.5). Jejich výsledky dostačující s ohledem na pozici nalezených bodů, ale jejich časová náročnost neumožňuje jejich použití v aplikacích reálného času. Jejich principy, nedostatky a důvody časové náročnosti jsou popsány v kapitole 6.

Dále jsem se obeznámil s novějšími metodami jako jsou transformace SIFT a metoda SURF (Speeded Up Robust Features). Obě metody a jejich rozdíly jsem popsal v kapitole 6. SIFT i SURF označovali podobné významné body, ale jejich hlavní rozdíl je v popisu deskriptorů. SIFT je přesnější díky skoro datově dvojnásobným

deskriptorům než obsahuje metoda SURF. Díky tomuto aspektu je SURF méně přesný k deskriptorům, ale dostatečně rychlý pro jeho přední místo pro použití v real-time aplikacích.

Problém rozeznání výrazu obličeje je stále ve vývoji. Zatím se podařilo rozeznávat výraz u stálých uživatelů dané aplikace, s použitím trénování a následné predikce za pomoci stromových struktur nebo neuronových sítí. V některých případech se zde využívají také deformovatelné šablony. Šablony se nejčastěji využívají při rozeznávání deformace obličeje na fotkách, kde mají dostatečný čas pro zpracování.

Tato práce byla zadána jako rozeznání výrazu tváře v aplikacích reálného času, což nám dávalo požadavek na rychlost zpracování, a výrazu u neznámých osob což nám zamezilo využití trénovacích algoritmů a přidalo otázku definice výrazu tváře. Tuto poslední otázku jsem popsal v kapitole (8.1.3).

Moje výsledky jsem popsal v kapitole 8.3. Pro jednoduchý a klidný postoj jsou výsledky přijatelné. Počítám zde pouze s jedním uživatelem používajícím aplikaci. Při rychlých pohybech hlavy se ukázal problém vzorkování pomocí zapůjčené kamery. Rychlost zpracování je závislá na rozlišení kamery, ale i při rozlišení 1280x960 je rychlost dostačující. Díky komplexnosti problematiky a menším výskytem problémů se nepodařilo zcela zdokonalit korespondenci bodů a definici výrazů tváře.

Proto bych se pro další vývoj zaměřil na zlepšení korespondence nalezených bodů, jejich znovu lokalizaci při rotaci obličejové části. Přidal bych do algoritmu rozeznávání více výrazů než jen úsměv a překvapení. Dále bych přidal možnost rozeznání více obličejů paralelně, pokud se v obrazu vyskytuje víc jak jeden obličej. Poté by se dal už algoritmus použít v různých specifických projektech. Dal by se použít v mnoha aplikacích sociálních sítí jako automatický zadavatel statusu/stavu uživatele, pro komunikaci po internetu s nevidomými nebo pro průzkumné, statistické či jen informační účely.

Literatura

- [1] Stránky výrobce kamery
Dostupné na URL: http://www.theimagingsource.com/en_US/products/cameras/usb-ccd-color/dfk41bu02/
- [2] Kuehni R. G.: *Color Spaces*. 8. 2003
Dostupné na URL: <http://www4.ncsu.edu/~rgkuehni/PDFs/CloSp.pdf>
- [3] Tkalčič M., Tasič J.F.: *Colour spaces – perceptual, historical and applicational background*. University of Ljubljana: 2003 Dostupné na URL: http://ldos.fe.uni-lj.si/docs/documents/20030929092037_markot.pdf
- [4] Barevné modely
Dostupné na URL: <http://gis.zcu.cz/studium/pok/Materialy/Book/ar03s01.html>
- [5] Charles Poynton, *Frequently Asked Questions about Color* 1997-03-02
Dostupné na URL: <http://www.poynton.com/PDFs/ColorFAQ.pdf>
- [6] Schapire R.E., *The Boosting Approach to Machine Learning An Overview*, 12. 2001
Dostupné na URL: <http://www.cs.princeton.edu/courses/archive/spr08/cos424/readings/Schapire2003.pdf>
- [7] Freund Y., Schapire R.E., *A Short Introduction to Boosting*
Dostupné na URL: <http://www.site.uottawa.ca/~stan/csi5387/boost-tut-ppr.pdf>
- [8] Moravec H.P., *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*, Carnegie-Mellon University, 9. 1980 Dostupné na URL: http://www.ri.cmu.edu/publication_view.html?pub_id=22
- [9] Harris C., Stephens M., *A Combined and edge detektor*, United Kingdom, 1988
Dostupné na URL: <http://www.bmva.org/bmvc/1988/avc-88-023.pdf>
- [10] Lindeberg T., *Scale-space*, Stockholm, Sweden, 9. 2008
Dostupné na URL: <ftp://ftp.nada.kth.se/CVAP/reports/Lin08-EncCompSci.pdf>
- [11] Brown M., Lowe D., *Invariant features from interest point groups*. In *Proceedings of British Machine Vision Conference*, pages 5–6, 2002. Dostupné na URL: http://read.pudn.com/downloads80/doc/309729/SIFT04_AffineMatch_bmvc2002.pdf
- [12] Edelman S., Intrator N., Poggio T., *Complex cells and object recognition*, 1997.
Dostupné na URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.412&rep=rep1&type=pdf>
- [13] D. G. Lowe. *Object recognition from local scale-invariant features*. 1150–1157, 1999.
Dostupné na URL: <http://www.cs.ubc.ca/~lowe/papers/iccv99.pdf>
- [14] Bílek P., *Významné body v obraze: detekce, korespondence a lokalizace ve 3D*, ČVUT, 8.2007,
Dostupné z URL: http://dip.felk.cvut.cz/browse/pdfcache/bilekp3_2007bach.pdf
- [15] D. G. Lowe. *Distinctive image features from scale-invariant keypoints* 91–110, 2004.

- [16] H. Bay, T. Tuytelaars, and L. van Gool. SURF: Speeded up robust features, demonstration. Dostupné na URL: <http://eccv2006.tugraz.at/files/demos/Bay.pdf>
- [17] H. Bay, T. Tuytelaars, and L. van Gool. SURF: Speeded up robust features. 404–417, 2006.
- [18] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. 63–86, 2004. Dostupné na URL: http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/mikolajczyk_ijcv2004.pdf
- [19] Stegmann M.B., Gomez D.D., A Brief Introduction to Statistical Shape Analysis Dostupné na URL: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=403
- [20] Cootes T.F., Taylor C.J., Statistical Models of Appearance for Computer Vision, 3. 2004 Dostupné na URL: http://www.face-rec.org/algorithms/AAM/app_models.pdf
- [21] Smola A.J., Schölkopf B. , A tutorial on support vector regression, 2004, 199-222.
- [22] Müller K. R., Mika S., Rätsch G., K. Tsuda, and B. Schölkopf, An introduction to kernel-based learning algorithms, 2001, 181-201.
- [23] Lao S., Kotáru T., Okamoto T., Yamashita T., Tabata N., Kasace M, A Fast 360-Degree Rotation Invariant Face Detection System, Dostupné na URL: <http://lear.inrialpes.fr/people/triggs/events/iccv03/cdrom/demos/lao.pdf>
- [24] Frischnolz R., Face Detection [online]. c1999, poslední revize 16.9.2008. Dostupné na URL: <http://www.facedetection.com/>
- [25] Matas J., Šochman N J., AdaBoost, České Vysoké Učení Technické, Praha, Dostupné na URL: http://www.robots.ox.ac.uk/~az/lectures/cv/adaboost_matas.pdf
- [26] Hongbo D.: A Brief Introduction to Adaboost , 2007 Dostupné na URL: <http://dtpapers.googlecode.com/files/Hongbo.ppt>
- [27] Viola P. , Jones M. , 'Fast and robust classification using asymmetric adaboost and a detector cascade', Canada., 2001 , Dostupné na URL: http://research.microsoft.com/en-us/people/viola/Pubs/Detect/violaJones_NIPS2002.pdf
- [28] Loy G. , Eklundh J., 'Detecting Symmetry and Symetric Constellations of Features, 2006
- [29] Wai H. L., ZHANG A.M., Kleeman Lindsay , Bilateral Symmetry Detection for Real Time Robotics Applications
- [30] Vernon D., Machine vision: Automated visual inspection and robot vision, 1991
- [31] SKALA V. Světlo, barvy a barevné systémy v počítačové grafice; Academia 1993

Seznam příloh

Příloha 1. CD/DVD

Příloha 2. Obsah přiloženého disku

Příloha 2.

CD-ROM /

LITERATURA

TESTOVACI VIDEO

GALERIE

VYTVOŘENÉ VZORKY

- vyfocené fotky pro vytvoření vzorků kůže

RŮZNÉ

- obrázky detekce významných bodů jejich omezení
atd.

MATLAB

3D MODELY - funkce pro vykreslení 3D modelu

MNSTBODU HODNOTY - obsahuje už vybrané hodnoty a jejich četnost pro pozdější zpracování

MODELY - M-Files pro vykreslení vybraných vzorků kůže ve 3D i ve formě nárysu půdorysu a bokorysu

POUZE 3D - soubory pro vykreslení 3D modelů

TĚLESA - soubory pro vykreslení těles se zastupujícími body

OBRÁZKY REPREZENTANTŮ LIDSKÉ KŮŽE V BAREVNÝCH PROSTORECH

– obrázky navzorkovaných barev vzorků kůže ve vybraných barevných prostorech

PROGRAMY - Obsahuje programy zhotovený v prostředí OpenCV:

MORAVEC_HARRIS ... ukázka detekování významných bodů

ZkusebníVerze2,3 ... postupný progres a ukázky vyzkoušených kódů

FINAL_Aplikace ... konečný zhotovený kód