

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Využití významných bodů obličeje pro
detekování úhlu pohledu**
Gaze Detection using Facial Landmarks

Zadání diplomové práce

Student: **Bc. Michal Jamroz**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Využití významných bodů obličeje pro detekování úhlu pohledu**
Gaze Detection using Facial Landmarks

Jazyk vypracování: čeština

Zásady pro vypracování:

Detekce význačných obličejových bodů (tzv. landmarků) je v posledních letech stále více využívanou technikou pro detailnější analýzu lidské tváře. Za pomoci zájmových bodů můžeme rozpoznávat například výraz ve tváři, míru natočení obličeje nebo pozici očí. Tyto body mohou také napomáhat při určování směru pohledu. Cílem této práce je vytvoření aplikace, která bude využívat obličejových landmarků k vyhodnocování směru pohledu.

1. Seznamte se se základními pojmy v oblasti detekce důležitých obličejových bodů v obrazech.
2. Seznamte se s technikami detekování zornic a duhovek v obrazech.
3. Seznamte se s knihovnou OpenCV.
4. S pomocí knihovny OpenCV implementujte program, který bude vyhodnocovat úhel pohledu (využít můžete volně dostupné algoritmy pro detekování zájmových bodů obličeje).
5. Experimentálně ověřte funkčnost, přesnost a rychlost řešení.
6. Svě závěry řádně zdokumentujte v textu práce.

Seznam doporučené odborné literatury:

- [1] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. vol. 1, pp. I-511-I-518 vol.1 (2001)
- [2] Ren, S., Cao, X., Wei, Y., Sun, J.: Face alignment at 3000 FPS via regressing local binary features. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1685–1692 (June 2014)
- [3] Wísniowska, J., Rezaei, M., Klette, R.: Robust eye gaze estimation, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8671, pp. 636-644 (2014)

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Radovan Fusek**

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2016

A handwritten signature in cursive script, appearing to read 'Jan', is written over a horizontal dotted line.

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, ať radou nebo jen slovem podpory. Jmenovitě potom svému vedoucímu práce panu Ing. Radovanu Fuskovi za objektivní připomínky, předané zkušenosti a v neposlední řadě za trpělivost a motivaci v těžkých chvílích. Panu Ing. Michaelu Holušovi za poskytnutí testovacích dat a souhlasu s jejich zveřejněním. Dále bych chtěl poděkovat mým přátelům (především Bc. Radku Simkaničovi DiS. a Bc. Adamu Václavíkovi) a rodičům. V neposlední řadě bych rád poděkoval své sestře Ivetě Jamrozové a své snoubence Bc. Haně Vyvialové DiS.

Abstrakt

Tématem této diplomové práce je detekce úhlu pohledu za využití významných bodů obličeje. Vytvořený program využívá již hotové implementace detektoru významných bodů, pomocí kterého jsou nalezeny oční oblasti. Tyto oblasti jsou využity k detekci obvodu a středu duhovky pomocí Houghovy transformace. Střed duhovky je dále poměřován s oční oblastí a na základě jeho polohy je vyhodnocen úhel pohledu pozorované osoby. Z důvodu odstranění mírné nestability detekce úhlu pohledu, způsobené detektorem významných bodů, obsahuje detektor paměť předchozích detekcí, které aproximují detekci aktuálního snímku.

Teoretická část je zaměřena na uplatnění a popis konkrétních řešení detektoru významných bodů obličeje a detektoru duhovek a zornic. Praktická část popisuje samotný algoritmus detekce úhlu pohledu a testy, které na něm byly provedeny.

Klíčová slova: úhel pohledu, landmarky, významné body obličeje, detekce, OpenCV, obličej, oko, pohled

Abstract

The subject of this master thesis was a gaze detection using facial landmarks. The created program uses already finished implementation of facial landmarks detector. This detector is used to find eye regions. Using these regions the circuit and the center of iris is detected by Hough transform. Based on the position of iris center, inside the eye region, the person's eye gaze is calculated. A buffer which stores a specific number of detections is used. These past detections help to approximate the current image detection.

The theoretical part is focused on motivation and description of a specific solutions of facial landmarks detector and iris and pupil detector. The practical part describes the created eye gaze detection algorithm itself and the performed tests.

Key Words: angle of view, landmarks, facial landmarks, detection, OpenCV, face, eye, gaze

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
1 Úvod	12
2 Detekce významných bodů obličeje	14
2.1 Regrese lokálních binárních příznaků	16
2.2 Kombinační hledání a regrese tvaru	17
2.2.1 Detekce lokálních příznaků	18
2.2.2 Hledání shody množiny příznaků se statistickým modelem tvaru	18
2.2.3 Kombinační hledání vhodných příznaků	18
2.3 Použití Bayesovy věty	18
3 Detekce zornic a duhovek	20
3.1 Rozpoznávání vzorů	22
3.1.1 Haarovy příznaky	22
3.1.2 Detekce zornice s pomocí Haarových příznaků	26
3.2 Detekce založená na tvaru	26
3.2.1 Detekce částečně zakryté duhovky	27
3.2.2 Detekce duhovky metodou Houghových kružnic	30
4 Použité nástroje	35
5 Vlastní detektor úhlu pohledu	36
5.1 Vstupní data	37
5.2 Detekce významných bodů obličeje	37
5.3 Detekce oční oblasti	38
5.4 Ekvalizace histogramu	39
5.5 Morfologická operace eroze	39
5.6 Vyhlazení Gaussovým filtrem	40
5.7 Binární prahování	40
5.8 Detekce hran	41
5.9 Detekce kružnic	41
5.10 Určení úhlu pohledu	42
6 Testování vytvořeného detektoru	44
6.1 Přesnost detektoru	44
6.2 Vylepšení přesnosti detektoru	48

6.3	Rychlost detektoru	49
6.4	Problémy detektoru	51
6.4.1	Rozlišení vstupního obrazu	51
6.4.2	Detekce významných bodů obličeje	52
7	Závěr	54
	Literatura	56
	Přílohy	58
A	Příloha na CD	59
A.1	Diplomová práce	59
A.2	Implementace detektoru úhlu pohledu	59
A.3	Testovací videozáznam	59
A.4	Ukázka výstupu	59
B	Návod k použití detektoru úhlu pohledu	60
B.1	Knihovna OpenCV	60
B.2	CMake	60
B.3	Konfigurace CMakeLists.txt	60
B.4	Kompilace programu	60
B.5	Spuštění programu	60

Seznam použitých zkratk a symbolů

FPS	–	Frames Per Second - Snímky za vteřinu
RANSAC	–	Random sample consensus - Shoda náhodných vzorků
PCA	–	Principal Component Analysis - Analýza hlavních komponent
SIFT	–	Scale-invariant feature transform
AdaBoost	–	Adaptive Boosting
LoG	–	Laplacian of Gaussian - Laplaceův operátor s Gaussovým filtrem

Seznam obrázků

1	Ukázka detekce významných obličejových bodů.	14
2	Morfování tváří.	15
3	Lokální binární příznaky.	16
4	Nejlepší rozsahy lokálních regionů v krocích 1, 3 a 5.	17
5	Příklad detekce s neúplným souborem lokálních příznaků.	17
6	Výstup detektoru s použitím Bayesovy věty.	19
7	Porovnání dvou heatmap.	21
8	Haarovy příznaky pro detekci oka.	22
9	Příklady Haarových příznaků.	23
10	Příklad jednoduchého výpočtu integrálního obrazu.	23
11	Příklad složitějšího výpočtu integrálního obrazu.	24
12	Příklad zapojení klasifikátorů do kaskády.	25
13	Nalezení polohy zornice pomocí Haarových příznaků.	26
14	Detekce a vyznačení duhovky.	27
15	Postup detekce částečně zakryté duhovky.	30
16	Hranová detekce duhovky.	32
17	Parametrický prostor pro Houghovu kružnicovou transformaci.	33
18	Houghova kružnicová transformace.	34
19	Diagram detekce duhovky.	36
20	Příklad detekce významných bodů obličeje.	37
21	Příklad detekce oční oblasti.	38
22	Rozšíření oblasti oka. (a) původní oblast oka, (b) rozšířená oblast oka.	38
23	Ukázka ekvalizace histogramu oblasti oka.	39
24	Ukázka po aplikaci morfologické operace eroze na oblast oka.	40
25	Ukázka oblasti oka po vyhlazení Gaussovým filtrem.	40
26	Ukázka oblasti oka po provedení prahování.	41
27	Ukázka detekovaných hran v oblasti oka.	41
28	Ukázka detekce kružnic v oblasti oka. Detekovaná kružnice - červená, střed kružnice - žlutý bod.	42
29	Ukázka určení polohy středu duhovky v oční oblasti.	42
30	Ukázka detekce úhlu pohledu.	43
31	Úspěšnost algoritmu na úhlu pohledu $x = -10^\circ$, $y = -10^\circ$	44
32	Úspěšnost algoritmu na úhlu pohledu $x = -20^\circ$, $y = -10^\circ$	45
33	Úspěšnost algoritmu na úhlu pohledu $x = -30^\circ$, $y = -10^\circ$	46
34	Úspěšnost algoritmu na úhlu pohledu $x = -30^\circ$, $y = 0^\circ$	46
35	Chybovost detekce úhlu pohledu.	47

36	Úspěšnost algoritmu na úhlu pohledu $x = -30^\circ$, $y = -10^\circ$. S použitou pamětí 7 snímků.	48
37	Rychlost detekce úhlu pohledu s výchozím nastavením detektoru tváří.	50
38	Rychlost detekce úhlu pohledu s konfigurovaným nastavením detektoru tváří. . .	51
39	Ukázka detekce oční oblasti při nízkém rozlišení.	52
40	Detekovaná oční oblast při nízkém rozlišení.	52
41	Detekované významné body obličeje při nízkém rozlišení.	53

1 Úvod

Zrak je jedním ze smyslů, které lidem umožňují vzájemnou interakci. Informací, které je možné z pouhého pohledu získat existuje mnoho, jednou z hlavních je směr pohledu člověka. Se znalostí úhlu pohledu lze umožnit ovládání nejrůznějších zařízení bez nutnosti fyzického kontaktu s nimi. Tento druh ovládání může například usnadnit lidem s omezenou pohyblivostí těla provádět obvyklé každodenní činnosti.

Detekce úhlu pohledu vychází z vědního oboru počítačového vidění. Tento obor se obecně zaměřuje na zpracování a analýzu obrazu. Největší rozmach a postup vývoje v této oblasti je zaznamenáván právě v období posledních zhruba dvaceti let. To je způsobeno především větší dostupností dostatečně výkonného hardwaru.

Úloh, které do oblasti počítačového vidění spadají, existuje celá řada. Algoritmy bývají často naprogramovány tak, aby zvládaly jednu konkrétní úlohu, pro kterou jsou určeny. Tento typ nástrojů bývá používán v prostředích, ve kterých je předem definováno, jaké situace mohou nastat. Příkladem může být kontrola vad výrobku na sériové lince továrny. Druhým typem algoritmu je algoritmus učící. Ten potřebuje před nasazením do samotného provozu podstoupit krok trénování. Pro tento účel jsou používány trénovací sady. Ty se skládají z množin pozitivních a negativních případových snímků, ze kterých se algoritmus učí rozeznávat požadovanou událost nebo přítomnost určitého objektu v obraze.

Samotná detekce úhlu pohledu se skládá z několika témat zpracování obrazu. Zajímá-li nás úhel, kterým se pozorovaná osoba dívá, potřebujeme určit nějakou obrazovou stopu, podle které je možné tuto vlastnost odhalit. Jednoznačným ukazatelem jsou v tomto případě oči pozorované osoby. Další snažení tedy směřuje k nalezení očí v obraze. Samostatná detekce očí by ovšem byla poměrně náročnou úlohou, je tedy žádoucí provést jisté předzpracování obrazu. Napříč různými přístupy k detekci úhlu pohledu je jako první aplikována detekce tváří. Tímto krokem je určen relevantní prostor, ve kterém se oči nacházejí. Dalším krokem je samotné nalezení očních oblastí. Metod, které mohou být využity k detekci očí, je více. Patří mezi ně například detekce významných bodů obličeje. Tyto body pomáhají definovat části obličeje, včetně právě očí. Známe-li polohu oka, můžeme využít obrazových informací z těchto míst pro určení úhlu pohledu.

Cílem této práce je seznámení se s různými metodami detekce významných bodů obličeje a detekce duhovek a zornic v obraze. Na základě tohoto zkoumání je dále práce zaměřena na společné využití těchto dvou metod. V praktické části byl s ohledem na toto zadání vytvořen program, který se zabývá určováním úhlu pohledu pomocí detekce duhovek. Pro nalezení očních oblastí je využito významných bodů tváře. Jak je uvedeno v zadání, samotný vývoj detektoru významných bodů tváře není předmětem této práce. Z tohoto důvodu je využito jeho již hotového řešení a vývoj je zaměřen přímo na detektor duhovek a určování úhlu pohledu.

Teoretická část práce je rozdělena do dvou hlavních kapitol, které se postupně zaměřují na vybrané detektory významných bodů tváře a na detektory duhovek a zornic. V obou těchto

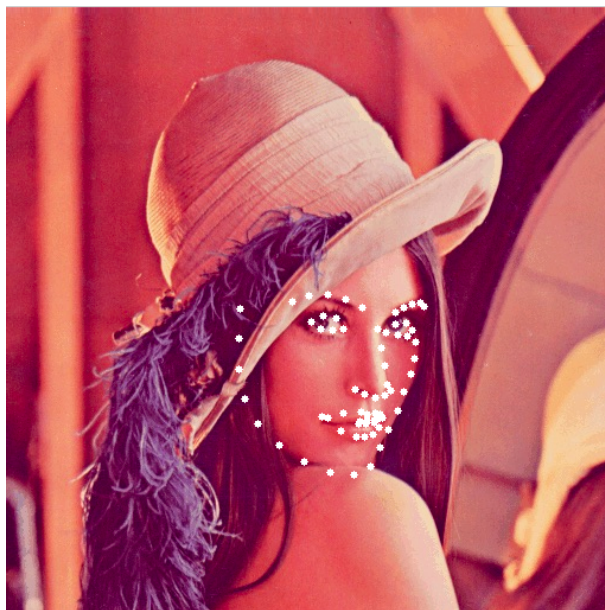
kapitolách je kromě samotných metod popsána také motivace k jejich využití.

Praktická část se zabývá popisem vytvořeného detektoru úhlu pohledu. Postupně jsou vysvětleny a popsány jednotlivé kroky algoritmu, včetně ukázek konkrétních příkladů. Vytvořený algoritmus je dále podroben testování, které se dělí na dvě části hodnotící přesnost a rychlost detektoru. V samotném závěru testovací části jsou uvedeny problémy detektoru, které byly během testů a samotného vývoje zjištěny.

2 Detekce významných bodů obličeje

V oboru počítačového vidění je hledání významných obličejových bodů (tzv. landmarků) jedním z aktuálních témat posledních let. Chceme-li se ale podívat na tento problém podrobněji, je vhodné si připomenout, co tomuto tématu předchází. Samotná detekce konkrétních landmarků nutně nemůže být prováděna bez určitého předzpracování obrazu, zvláště potom předpokládá-li se od algoritmu rychlost a přesnost. Před samotnou detekcí je tedy na vstupní obraz nejprve aplikována detekce tváří, jejímž výstupem je obdélníkový výřez obrazu obsahující nalezenou tvář. Nejznámějším řešením detekce tváří je bezpochyby přístup představený dvojicí Paul Viola a Michael Jones v roce 2001 v článku *Rapid Object Detection using a Boosted Cascade of Simple Features* [1]. Ten je s určitými rozšířeními stále využíván a je součástí knihovny OpenCV, zaměřené především na počítačové vidění a zpracování obrazu.

S vymezenou oblastí tváře je dalším zřejmým krokem snaha o nalezení příznaků tváře. Jako příznaky můžeme označit části obličeje (resp. části obrazu skládajících se ze skupiny pixelů), které jsou v různých tvářích do jisté míry jedinečné. Jde tedy o části obrazu, které nám umožňují rozlišit mezi dvěma tvářemi. Příklady příznaků nacházejících se v obličeji jsou například oblasti očí, nosu, úst, obočí, případně některé další potenciálně jedinečné oblasti (důlky ve tváři, vrásky, tvar brady, tvar celého obličeje, atd.). Zde se již dostáváme k detekci samotných landmarků, kterými lze oblasti těchto obličejových regionů popisovat.



Obrázek 1: Ukázka detekce významných obličejových bodů.

Uplatnění landmarků existuje celá řada. Hlavním tématem je již zmíněné využití při rozlišování tváří, kde umožňují vytvoření středního tvaru tváře, který pomáhá vylepšit jejich rozpoznávání. Princip středního tvaru tváře je založen na statistické veličině střední hodnoty, kterou můžeme jinými slovy označit také jako populační průměr. V případě tváře se podobně jako

u střední hodnoty jedná o jakýsi průměr všech tváří v dané populační množině. Na obrázku 1 je příklad tváře s nalezenými landmarky. V tomto příkladě je k popisu tvaru tváře použito 68 bodů. Při výpočtu střední hodnoty několika tváří se počítá průměr jednotlivých korespondujících bodů napříč množinou vstupních tváří.

Poměrně rozsáhlé využití, má také odhad orientace hlavy. Pohyby hlavy jsou součástí mezilidské komunikace a určitým způsobem doplňují mluvené slovo gesty. Pokývnutím hlavy signalizujeme svůj souhlas, zavrtěním do stran nesouhlas, atd. Tato gesta by mohla v budoucnu stejně tak doplnit i interakci s počítačem, kde se již nyní hojně rozvíjí hlasové ovládání. V případě detekce úhlu pohledu se orientace hlavy může využít jako podpůrná metoda, která pomáhá při zvýšení přesnosti detekce. Ne vždy se totiž tvář musí nacházet v poloze orientované přímo na kameru. Samozřejmě ale záleží na konkrétním použití detektoru. Odhad orientace hlavy je možné provádět i pomocí nižšího počtu určitých landmarků. K určení orientace hlavy jsou důležité informace o poloze uší, brady, nosu a případně i očí. Se znalostí polohy těchto obličejových příznaků jsme schopni učinit odhad orientace hlavy. Ať už pomocí poměrování polohy příznaků s odpovídajícími příznaky v referenční tváři, která je přímo orientovaná do kamery, nebo i srovnávání polohy příznaků v rámci stejného obličeje.

Další využití můžeme najít v oblasti digitálních úprav. Známe-li umístění dané části obličeje v obraze, můžeme na ní provádět různé vizuální modifikace. Je tak například možné vytvoření automatických funkcí pro upravení tváře zachycené na digitálním snímku. Příkladem takovéto automatizované funkce může být zvýraznění kontury rtů, případně upravení make-upu osoby na snímku. V praxi detektor nalezne landmarky popisující oči nebo ústa, určí konturu této oblasti a následně na ní provede požadovanou změnu.

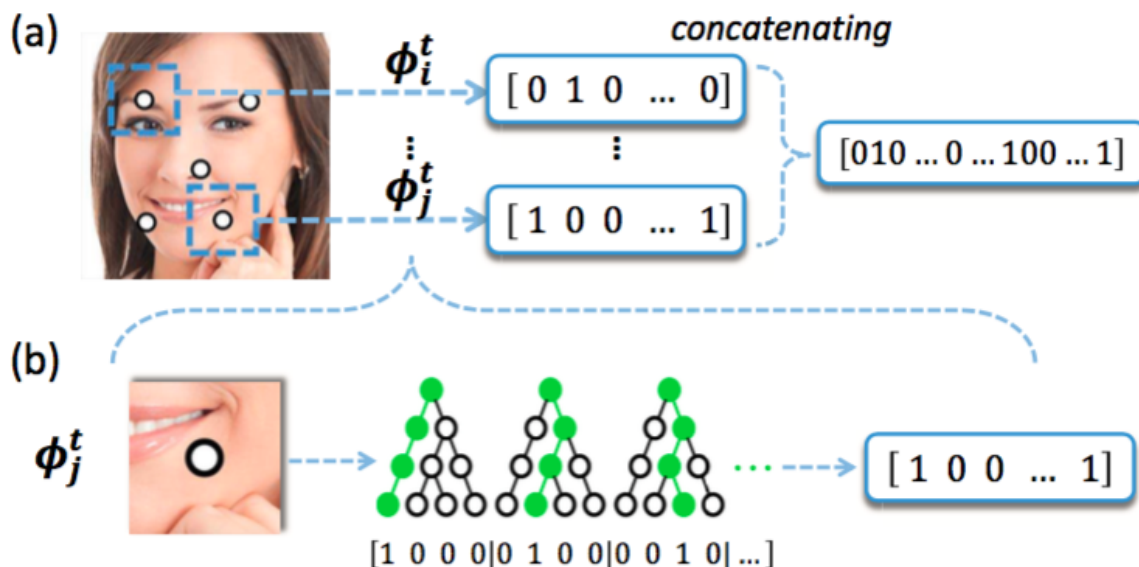


Obrázek 2: Morfování tváří.

Landmarky mohou sloužit při tzv. morfování tváří (obrázek 2), kde jsme schopni vytvořit animovaný přechod z jedné tváře do druhé. Tohoto efektu bývá již desítky let využíváno v odvětví filmového průmyslu se sci-fi tematikou. Kromě těchto úprav jsme také poměrně jednoduše schopni překrýt celou tvář jinou.

2.1 Regrese lokálních binárních příznaků

Metoda nalezení landmarků prostřednictvím regrese lokálních binárních příznaků byla představena v článku *Face Alignment at 3000 FPS via Regressing Local Binary Features* [2]. Autoři uvádějí, že metoda detekce využívá dvou nových přístupů. První novinkou, vyplývající již ze samotného názvu článku, je využití množiny tzv. *lokálních binárních příznaků*. Druhou je potom *princip lokality*, který popisuje vzájemně nezávislý způsob jejich učení pro jednotlivé landmarky. Binární příznaky vyprodukované tímto způsobem jsou využity k učení lineární regrese.

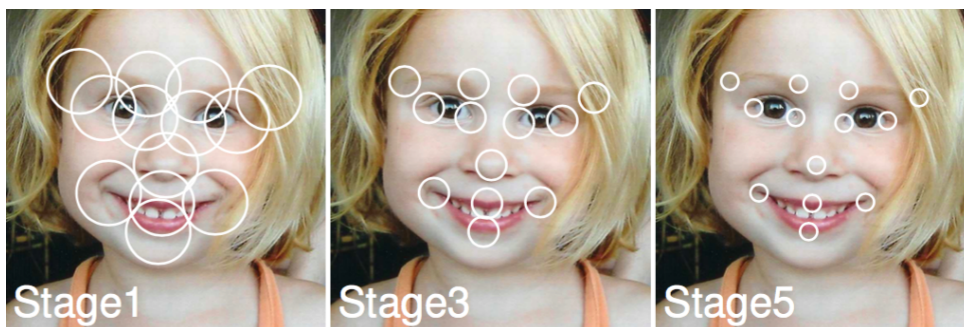


Obrázek 3: Lokální binární příznaky.

Proces extrakce příznaků je ilustrován na obrázku 3. Funkcí ϕ_i^t jsou jednotlivé oblasti převedeny do formy binárních příznaků. Ty jsou následně zřetězeny a vytváří vysokodimenzionální binární příznaky pro množinu landmarků celého obličeje. Příznaky jsou reprezentovány formou náhodného lesa. Každý získaný binární příznak potom určuje, jestli vstupní obrázek obsahuje nějaký lokální vzor nebo ne.

Princip lokality, uvedený jako jedna ze dvou novinek této metody, využívá kaskádového trénování. V každém kroku je odhadován nejlepší poloměr regionu pro šetření daného landmarku. Na obrázku 4 můžeme pozorovat postupně se snižující poloměr v jednotlivých krocích. Důvodem jeho snižování je odpovídající snížení variace tvaru obličeje v postupných kaskádových krocích algoritmu.

Způsob získání příznaků popsany v tomto článku i jejich následná regrese jsou, co se výpočetní rychlosti týče, poměrně jednoduchou operací. Autoři uvádějí, že celý proces nalezení 68 významných obličejových bodů dosahuje na stolním počítači s jednojádrovým procesorem rychlosti přes 3000 FPS.

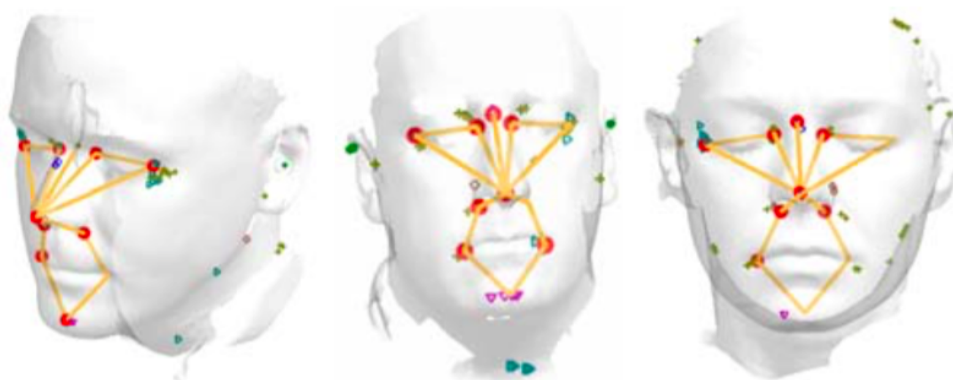


Obrázek 4: Nejlepší rozsahy lokálních regionů v krocích 1, 3 a 5.

2.2 Kombinační hledání a regrese tvaru

Detekce obličejových landmarků popsaná v článku *3D Facial Landmark Localization using Combinatorial Search and Shape Regression* [3] je založena na kombinačním hledání s použitím algoritmu shody náhodných vzorků (RANSAC - *Random sample consensus*).

Nejprve je pro každý landmark vybraným detektorem příznaků získána množina 3D kandidátů. Tyto body jsou použity jako vstup algoritmu, který provede kombinační hledání omezené deformovatelným modelem tvaru. Důvodem tohoto omezení je fakt, že ani jeden z kandidátů na určitý landmark nemusí být dostatečně přesný. V takovém případě je potom detekována jen část landmarků a zbylé, ke kterým nebyl nalezen kandidát s odpovídající přesností, jsou následně podle určitých pravidel odvozeny. Tento způsob podmíněné selekce landmarků z množiny kandidátů vede k maximalizaci přesnosti výsledného deformovatelného modelu tvaru. Protože je model schopen pracovat i s neúplnou množinou landmarků, máme možnost omezit počet kandidátů. S tímto omezením dochází rovněž k velmi výraznému snížení počtu možných kombinací, které musejí být algoritmem otestovány.



Obrázek 5: Příklad výstupu detekce s neúplným souborem lokálních příznaků. Výsledný tvar je vyznačen čarami, úspěšně detekované kandidátní body červenými značkami, ostatní barevné značky odpovídají dalším kandidátním bodům.

2.2.1 Detekce lokálních příznaků

Pro nalezení lokálních příznaků je použit systém vyhodnocení skóre daného kandidáta. Vysoké skóre by v ideálním případě mělo určit kandidáta blízkého cílovému bodu. Tento přístup však poměrně často naráží na problém tzv. *false positive* (tzn. chybné detekce), kdy kandidát sice má vysoké skóre, přesto je však jeho vzdálenost od cílového bodu vysoká. Pro odstínění tohoto trendu je žádoucí udržet velikost množiny kandidátů na takové úrovni, aby pokud možno alespoň jeden z nich byl dostatečně blízký cílovému bodu.

2.2.2 Hledání shody množiny příznaků se statistickým modelem tvaru

Detekované lokální příznaky jsou zřetězeny do jediného vysokodimenzionálního vektoru. Dále je na celou trénovací množinu aplikována analýza hlavních komponent (PCA - *Principal Component Analysis*). Výstup analýzy je poté používán v testovací fázi, kde se s jeho pomocí určuje, jestli testovaná množina odpovídá danému modelu PCA.

V případě, že vstupem je neúplná množina bodů, jsou vzaty pouze dostupné body a zbývající jsou odvozovány statistickým modelem. Cílem je odvodit chybějící body tak, aby se pravděpodobnost výsledného tvaru co nejvíce blížila modelu PCA a již známé body zůstaly pokud možno beze změn.

2.2.3 Kombinační hledání vhodných příznaků

Postup odhadování chybějících příznaků je založený na RANSAC. Algoritmus začíná s množinami kandidátů na jednotlivé významné obličejové body. Následně postupně vybírá jednotlivé kandidáty na odpovídající body a vyhodnocuje všechny jejich možné kombinace s ostatními. Hlavním parametrem indikujícím vhodnosti kandidáta je jeho vzdálenost od aktuálního odhadu daného landmarku.

2.3 Použití Bayesovy věty

Přístup využívající pro detekci landmarků Bayesovu větu byl představen v článku *Localizing Parts of Faces Using a Consensus of Exemplars* [4]. Podobně jako výše představené metody i tato využívá pro hledání obličejových bodů lokálních informací.

Zajímavým aspektem tohoto přístupu je ovšem fakt, že se trénovací množina skládá z obličejových vzorků s již anotovanými významnými body. Lokální příznaky získané tímto způsobem zajišťují neparametrické informace pro globální model.



Obrázek 6: Výstup detektoru s použitím Bayesovy věty.

Lokální detektor využívá posuvného okénka s velikostí úměrnou integrální optické hustotě a získává SIFT příznaky. Po získání lokálních příznaků jsou globálním detektorem modelovány možné konfigurace množiny landmarků. Podmíněná pravděpodobnost polohy landmarků a vektoru jejich lokálních příznaků je maximalizována. Co se přesnosti týče autoři této metody uvádějí, že ve srovnání s manuální detekcí je jejich přístup efektivnější ve většině z 29 hledaných významných bodů obličeje.

3 Detekce zornic a duhovek

Nalezení polohy zornice oka je klíčem k určení směru pohledu sledované osoby. Podobně jako detekce landmarků je rovněž detekce zornic a duhovek v posledních letech podrobena intenzivnímu zkoumání. Obecně existují dva základní přístupy k řešení této detekce. První je založen na poloze a naklonění hlavy, jak již bylo zmíněno v kapitole 2. Druhým způsobem je detekce úhlu pohledu právě pomocí určení polohy samotné oční zornice nebo duhovky. Nejvýhodnější je ovšem kombinace obou přístupů.

Můžeme nalézt mnoho různých využití informace o směru pohledu pozorované osoby. Všechny mají společný základ, a tím je určení místa, které daná osoba pozoruje. Se znalostí této informace je možné podniknout celou řadu akcí, od statistických průzkumů sledování reklamy až k přímému bezkontaktnímu ovládní nejrozličnějších elektronických spotřebičů.

V oblasti webdesignu nebo reklamy je žádoucí upoutat a udržet pozornost pozorovatele, resp. uživatele. V některých případech je dokonce možné vést pozorovatelův pohled směrem k nějakému konkrétnímu místu. K tomu slouží různá grafická „vodítka“, která pohled přirozeně směřují (např. výrazný nadpis, struktura textu nebo i fotografie tváří atraktivních osob). Za účelem zjištění nejsledovanějších míst se provádí průzkumy, které vyhodnocují buď přímo samotný pohled uživatele nebo pohyb ukazatele myši na obrazovce. Druhá z uvedených možností se ukázala být výrazně odpovídající skutečnému pohledu pozorovatele a není-li naším požadavkem striktní přesnost, jde rovněž o metodu výrazně snazší. Výstupem takovýchto měření bývají nejčastěji *heatmapy*, které znázorňují míru pozornosti drženou na jednotlivých částech obrazu. Na obrázku 7 si můžeme prohlédnout dva příklady takové heatmapy. Jak je již na první pohled patrné, levý obrázek využívá očí dívky k vedení pohledu pozorovatele směrem k prezentovanému produktu a také k jeho značce. Naproti tomu na pravém obrázku nám heatmapa ukazuje, že se uživatel na produkt vůbec nezaměřil a reklama je tedy v tomto případě neúčinná [5].

Dalším využitím detekce zornic je například kontrola pohledu řidiče automobilu pro zajištění jeho plné pozornosti. Nejen v prostředí auta se také nabízí možnost ovládní nejrozličnějších elektronických zařízení bez nutnosti fyzického kontaktu s nimi. S takovým způsobem interakce bychom se mohli v budoucnu setkat také v domácnosti při ovládní televize a ostatních spotřebičů.

Velmi zajímavou možností je využití sledování pohledu oka při chirurgických zákrocích. Při operacích, kde je přirozeně vyžadována vysoká přesnost, může příslušný detektor pohledu určit, jakou chirurgickou úlohu chce operatér provést. Tím se nabízí možnost bezdotykové manipulace s některými nástroji. Bylo také zjištěno, že z dat získaných sledováním úhlu pohledu chirurga lze získat informace vypovídající o jeho zručnosti. Při zpracování těchto informací chirurgickými nástroji můžeme znatelně vylepšit přesnost provedení dané operace [6].

Další perspektivní oblastí pro metody detekce úhlu pohledu jsou technologie usnadňující život hendikepovaným lidem. Přestože jsou tyto technologie stále ve fázi výzkumu, dosahují už nyní slibných výsledků. Ovládní pomocí zraku může poměrně efektivně pomoci lidem s určitou



Obrázek 7: Porovnání dvou heatmap s použitím vizuálního vodítka (levý obrázek) a bez něj (pravý obrázek) [29].

paralýzou těla zůstat v kontaktu s okolním světem. Tedy i přes neschopnost pohybu končetinami je takto hendikepovaný člověk schopen pracovat s počítačem pomocí tzv. „oční myši“. Tato alternativa ke klasické počítačové myši funguje shodně se svou předlohou, tedy pohyb kurzoru po obrazovce je umožněn samotným pohybem očí, vybrání požadované funkce mrknutím. Kromě klasického stolního počítače by samozřejmě tento způsob ovládání mohl být použit rovněž u mobilních zařízení, tabletů a další elektroniky [7, 8].

Určování úhlu pohledu může být mimo jiné nápomocné i při ovládání kolečkového křesla. Lidé s částečnou nebo úplnou paralýzou těla by mohli mít už brzy možnost, bez pomoci druhých osob, samostatně manipulovat s kolečkovým křeslem prostřednictvím „očních gest“. I v tomto případě se daná technologie nachází ve fázi vývoje. Princip ovládání je v zásadě jednoduchý, konkrétní gesta by mohla být například tato: pohyb očí doleva a doprava určí směr jízdy a např. pohyb jazyka směr potvrdí, mrknutím očí potom uvede vozík do chodu, resp. jej zastaví [9, 10].

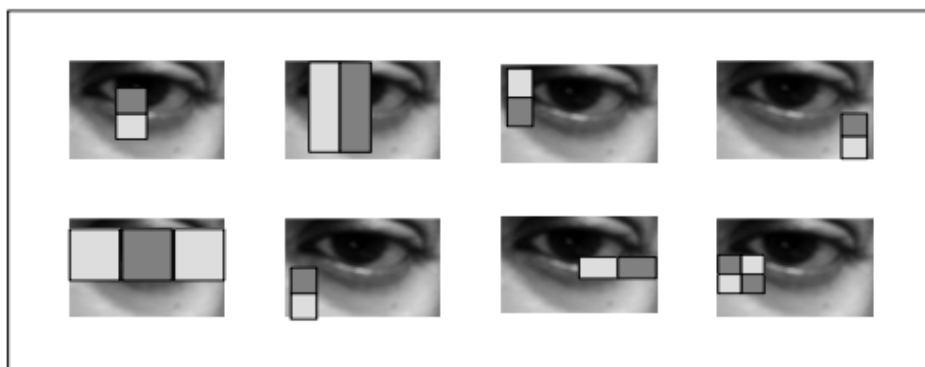
Následující podkapitoly budou zaměřeny na různé typy detektorů zornic a duhovek. Jednotlivé přístupy se od sebe, co do zpracování oblasti oka, navzájem kategoricky liší. Tento fakt však není ničím překvapivým vzhledem k tomu, že stále nemáme k dispozici dostatečně robustní a efektivní algoritmus, který by k této detekci sloužil. Z tohoto důvodu se výzkumná činnost zaměřuje na různé způsoby, kterými by potenciálně bylo možné určit přesnou polohu zornice.

Krokem předcházejícím detekci duhovky nebo zornice je hledání oka. V kapitole 2 jsme si ukázali poměrně efektivní metodu pro nalezení oblasti oka pomocí landmarků. Přestože se vývoj hledání landmarků již dostal do fáze, kdy máme k dispozici v podstatě *real-time* detektor, většina algoritmů pro detekci zornic a duhovek tohoto předzpracování obličeje nepoužívá. To je samozřejmě dáno tím, že oba tyto problémy počítačového vidění jsou poměrně aktuální a jejich společná aplikace je tedy otázkou těchto a následujících let. Dalším častým nástrojem pro detekci

očních oblastí jsou *Haarovy příznaky*. Ty budou podrobněji popsány v kapitole 3.1.1.

3.1 Rozpoznávání vzorů

Metoda rozpoznávání vzorů se hojně uplatňuje na různé problémy počítačového vidění. Detektor pracuje s obrazy ve stupních šedi. Pro klasifikaci se používají *Haarovy příznaky* s algoritmem *AdaBoost*, který kombinuje několik jednoduchých klasifikátorů do tzv. kaskády, což umožňuje efektivně vyřadit nezájímavé oblasti obrazu a soustředit se více na oblasti obsahující hledané objekty. Tyto detektory zpravidla využívají speciální reprezentace obrazu tzv. *integrální obrazy*, které umožňují velmi rychlý výpočet příznaků [13, 25].

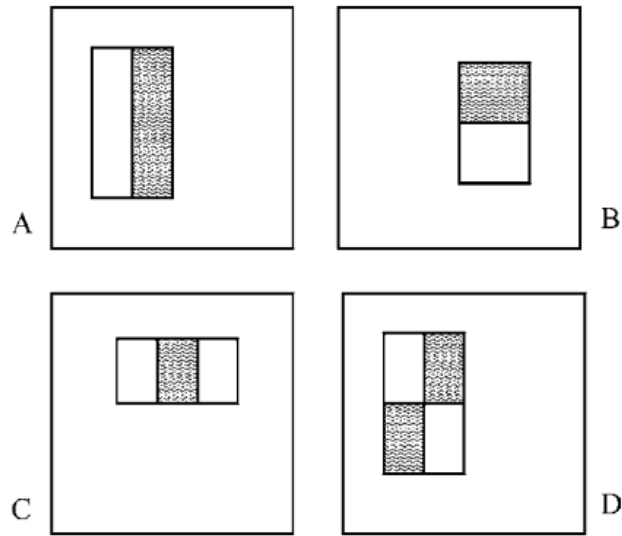


Obrázek 8: Haarovy příznaky pro detekci oka [16].

3.1.1 Haarovy příznaky

V úvodu trénovací procedury jsou vygenerovány Haarovy příznaky. Ty jsou obvykle obdélníkového tvaru a skládají se z několika dalších obdélníků. V závislosti na tom, jakou obrazovou informaci máme v úmyslu detekovat, můžeme využít různých typů těchto příznaků, např. dvoj-obdélníkové, troj-obdélníkové nebo čtyř-obdélníkové (obrázek 9). Výsledná odezva Haarova příznaku je potom v případě dvoj-obdélníkového příznaku vypočtena, jako rozdíl součtů pixelů v oblastech jednotlivých obdélníků daného příznaku. V případě troj-obdélníkového Haarova příznaku je vypočítán součet pixelů oblastí dvou vnějších obdélníků a následně je odečten od součtu pixelů oblasti vnitřního obdélníku. A nakonec čtyř-obdélníkový Haarův příznak je vypočítán, jako rozdíl součtů pixelů oblastí ležících v diagonále. Příklad Haarových příznaků pro detekci oka je ilustrován na obrázku 13.

Příznak je postupně hledán v celém obraze. Začíná se s nejmenšími rozměry detekčního okna 24×24 pixelů. Okno se postupně pixel po pixelu posouvá v horizontálním nebo ve vertikálním směru. Poté, co je již celý obraz prošetřen, je detekční okno zvětšeno a celá činnost se znovu opakuje. Extrakce příznaků je ukončena v momentě, kdy detekční okno již dosáhlo velikosti odpovídající obrazu a bylo zvětšeno nad rámec jeho velikosti. V případě rozlišení nejmen-

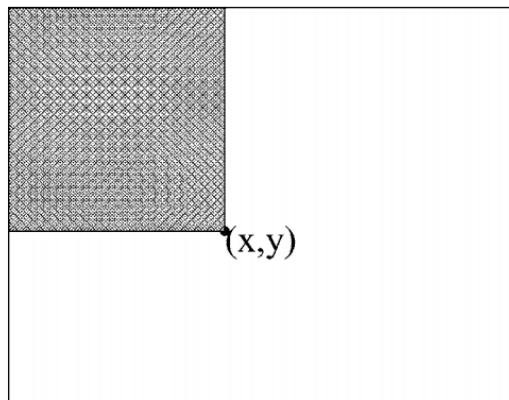


Obrázek 9: Příklady Haarových obdélníkových příznaků [13].

šího detekčního okna 24×24 dosahuje množina různých obdélníkových příznaků velikosti okolo 160 000 prvků [13, 14].

3.1.1.1 Integrální obraz

Při výpočtech odezvy Haarových příznaků je potřeba velmi často počítat součet hodnot pixelů obsažených v daných obdélnících. Z důvodu urychlení výpočtu je proto provedeno předzpracování obrazu v podobě převedení na integrální obraz. Výpočet hodnoty součtu pixelů v obdélníku je díky tomu možný až v konstantním čase. Hodnota na souřadnicích (x, y) je potom vypočtena, jako součet všech pixelů ve směru nahoru a vlevo od bodu (x, y) (obrázek 10).



Obrázek 10: Příklad jednoduchého výpočtu integrálního obrazu [13].

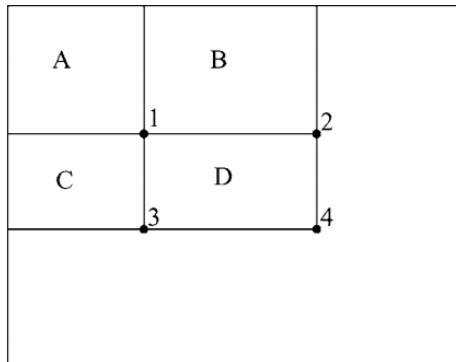
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (1)$$

Rovnice (1) popisuje výpočet integrálního obrazu $ii(x, y)$ z původního obrazu $i(x, y)$. Pro tento výpočet musí být splněny podmínky určené rovnicemi (2) a (3), kde $s(x, y)$ je kumulovaný součet jednotlivých pixelů v řádku, $s(x, -1) = 0$, a $ii(-1, y) = 0$ může být integrální obraz vypočten pouze jedním průchodem původního obrazu.

$$s(x, y) = s(x, y - 1) + i(x, y), \quad (2)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y), \quad (3)$$

Máme-li původní obraz převeden do integrálního, snadným výpočtem můžeme získat hodnotu obdélníku pro Haarův příznak. Vycházíme-li z příkladu na obrázku 11, je naším cílem získat hodnotu obdélníku D . Hodnota v bodě 1 je určena součtem hodnot pixelů v oblasti obdélníku A . Hodnota v bodě 2 je určena součtem hodnot pixelů v oblasti obdélníků A a B . Hodnota v bodě 3 je podobně jako předchozí určena jako součet hodnot pixelů v oblasti obdélníků A a C . A konečně hodnota v bodě 4 je určena součtem hodnot pixelů v oblastech všech obdélníků A, B, C a D . Výsledná hledaná hodnota obdélníku D je nyní vypočtena sčítáním a odečítáním získaných hodnot v jednotlivých bodech, tedy $4 - 3 - 2 + 1$ [13, 14].



Obrázek 11: Příklad výpočtu hodnoty součtu pixelů obdélníku v integrálním obraze [13].

3.1.1.2 Lineární klasifikátor

Klasifikátor má za úkol rozdělení příznaků do tříd. V případě hledání vzorů rozlišuje klasifikátor mezi dvěma třídami, pozitivní a negativní. Jde tedy v podstatě o určení, zdali daný příznak splňuje nějakou podmínku nebo ne. Slabý klasifikátor pracuje právě tak, že porovnává hodnotu příznaku s určitou prahovou hodnotou a podle toho přiřazuje příznaku odpovídající klasifikační

třidu. Skládá se z příznaku f , prahové hodnoty Θ a polariry p , která určuje směr nerovnítky. Rovnice (4) ukazuje rozhodovací mechanismus takového klasifikátoru.

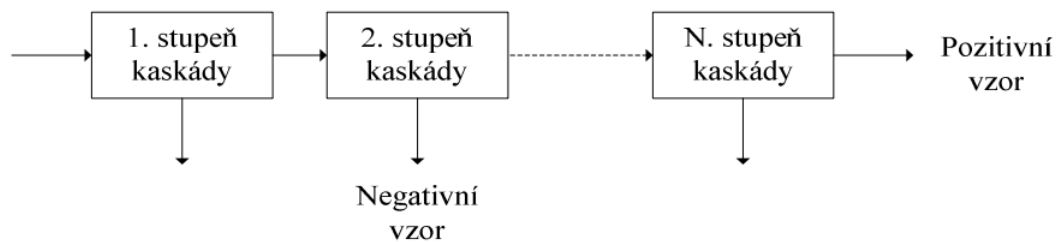
$$h(x, f, p, \Theta) = \begin{cases} 1 & pf(x) < p\Theta \\ 0 & jinak, \end{cases} \quad (4)$$

Proměnná x reprezentuje 24×24 pixelů vstupního obrazu. Samotná klasifikace probíhá tak, že se nejprve vypočte hodnota odezvy Haarova příznaku a podle polariry se určí, jestli se pozitivní třída klasifikátoru nachází pod ($p = -1$) nebo nad ($p = 1$) prahovou hodnotou. Nakonec zbývá určení příslušnosti k odpovídající klasifikační třídě. Hodnota klasifikátoru je porovnána s hodnotou prahu a výsledkem je informace o tom, jestli patří vstupní obraz x do pozitivní klasifikační třídy $h(x, f, p, \Theta) = 1$, nebo negativní klasifikační třídy $h(x, f, p, \Theta) = 0$ [13, 14].

3.1.1.3 Kaskáda klasifikátorů

Kvůli vysokému počtu detekčních oken používaných k prohledávání obrazu se detekce stává výpočetně velmi náročnou. Je tedy žádoucí snížit dobu detekce jednotlivých detekčních oken. K tomu slouží kaskádové zapojení klasifikátorů. Důležitým aspektem při detekci je fakt, že většina detekčních oken je negativní, což znamená, že neobsahuje hledaný objekt. Kaskáda klasifikátorů zapojuje jednotlivé slabé klasifikátory za sebe do kaskády. Jednotlivé klasifikátory potom postupně detekují pozitivní obrazy a spolu s nimi i malé množství chybně detekovaných pozitivních. Počet chybných pozitivních detekcí je v dalších krocích kaskády postupně zmenšován. Výsledný silný klasifikátor je vážená kombinace T slabých klasifikátorů s prahovou hodnotou P , která určuje, zda je aktuální detekční okno pozitivním nebo negativním obrazem. Silný klasifikátor může být interpretován rovnicí (5), kde h_t označuje slabý klasifikátor a α_t jeho váhu. Příklad zapojení klasifikátorů do kaskády je ilustrován na obrázku 12 [13, 14].

$$H(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \leq P \\ 0 & jinak, \end{cases} \quad (5)$$



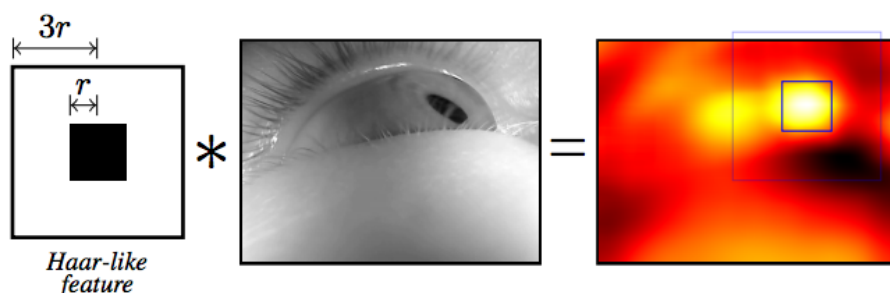
Obrázek 12: Příklad zapojení klasifikátorů do kaskády [14].

3.1.1.4 AdaBoost

Tento algoritmus se používá pro trénování kaskádových klasifikátorů. Každý stupeň kaskády je reprezentován silným klasifikátorem. AdaBoost (Adaptive Boosting) vybírá slabé klasifikátory $h(x)$, přičemž klade důraz na to, aby jejich kombinací vznikl silný nelineární klasifikátor $H(x)$ pro daný stupeň kaskády. Následný trénink používá na vstupu pozitivní a negativní snímky, kde pozitivní jsou snímky obsahující daný objekt a negativní snímky pozadí [13, 14].

3.1.2 Detekce zornice s pomocí Haarových příznaků

Haarovy příznaky (kapitola 3.1.1) mohou v případě detekce zornic poměrně rychle detekovat její přibližnou polohu. Myšlenka této detekce se zakládá na skutečnosti, že oblast zornice, případně celá oblast zornice a duhovky, může být popsána, jako tmavý objekt uprostřed světlejšího pozadí (očního bělma). Z tohoto pohledu na oblast zornice vyplývá také volba typu Haarova příznaku, kde je střední černý obdélník obklopen bílým. Na obrázku 13 můžeme vidět ilustraci tohoto Haarova příznaku a také výsledek konvoluce s obrazem oka. Jako místo polohy zornice je detekována oblast s nejsilnější odezvou Haarových příznaků mezi všemi pixely a poloměry Haarových příznaků [15, 16].



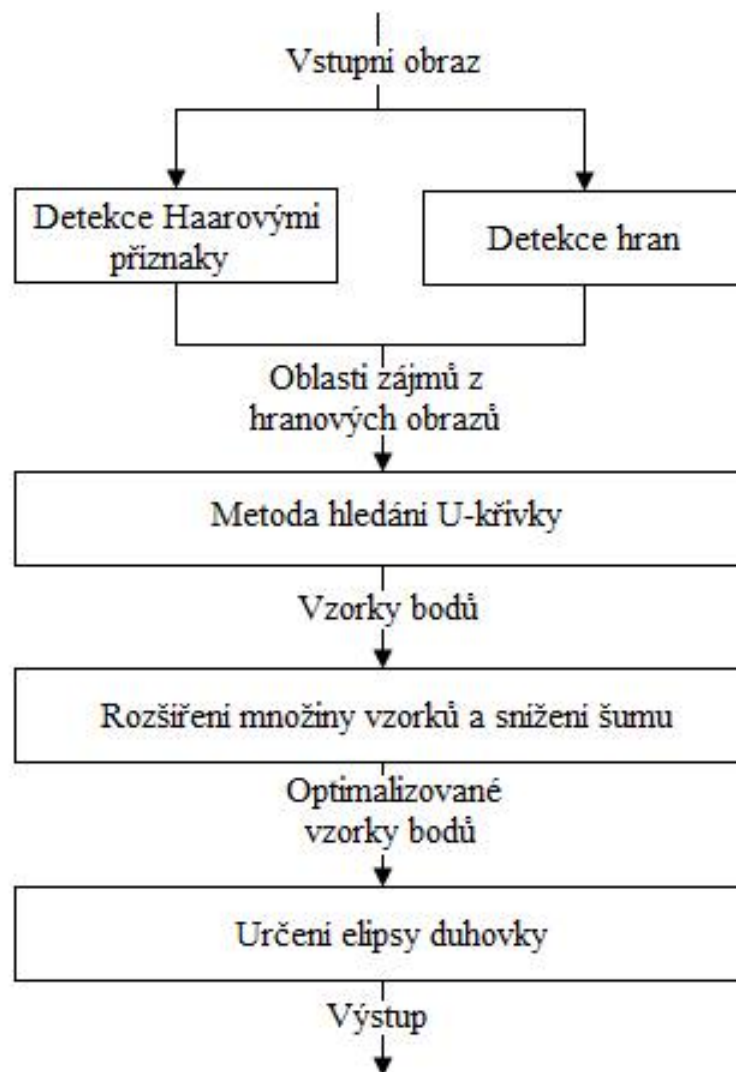
Obrázek 13: Nalezení polohy zornice pomocí Haarových příznaků [15].

3.2 Detekce založená na tvaru

Přístup k detekci zornic, který se zdá být v současnosti nejefektivnější, je založen na rozpoznávání jejich tvaru. Oproti metodám hledání vzorů jde v tomto případě o detekci založenou na jednoduchém principu hledání kruhového tvaru zornice, resp. duhovky. Napříč detektory zaměřenými na hledání duhovek nebo zornic pomocí tvaru se využívá metody Houghových kružnic. Ta se zaměřuje na hledání objektů v obraze, které se blíží kruhovému tvaru. Jak je z principu tvarové detekce zřejmé, tyto metody nevyžadují pro svou funkci žádná trénovací data. Tato vlastnost s sebou nese podstatné ušetření času, který by jinak musel být stráven shromažďováním potřebných trénovacích vzorků a samotným tréninkem [11, 12].

3.2.1 Detekce částečně zakryté duhovky

Tato kapitola je zaměřena na specifickou metodu detekce duhovky oka, konkrétně se zabývá přístupem publikovaným v článku *Iris Detection and Marking Based on Haar Features* [25]. Metoda detekuje oblast očí pomocí Haarových příznaků (kapitola 3.1.1) a souběžně s tímto krokem je prováděna i hranová detekce. Další zpracování již probíhá pouze v rámci detekovaných očních regionů, kde je na množinu nalezených hran aplikována metoda tzv. *U-křivky*, která vychází z toho, že část duhovky je překryta horním očním víčkem. Na obraz je aplikována redukce šumu a rozšíření množiny vzorků *U-křivky*. V závěru se algoritmus pokouší najít elipsu odpovídající *U-křivce*, která je považována za výstup a za odpovídající obvod duhovky. Celý algoritmus je graficky znázorněn na obrázku 14.



Obrázek 14: Detekce a vyznačení duhovky [25].

3.2.1.1 Cannyho detektor hran

Obraz je nejprve převeden do odstínů šedi, k tomu slouží rovnice (6) pro výpočet relativního jasu. Proměnné R , G a B reprezentují jednotlivé barevné složky daného pixelu, tedy červenou, zelenou a modrou a Y označuje výslednou hodnotu jasu [25].

$$Y = 0.212671 \times R + 0.715160 \times G + 0.072169 \times B \quad (6)$$

Dalším krokem je aplikování Gaussova filtru. Jinými slovy tedy provedeme konvoluci obrazu s Gaussovským jádrem 3×3 . Použití Gaussova filtru způsobí, že se obraz lehce rozostří, díky tomu dojde k odstranění případného šumu jednotlivých pixelů. Vzorec této konvoluce odpovídá rovnici (7), kde $D(u, v)$ je vzdálenost od počátku Fourierovy transformace a parametr σ označuje šířku Gaussovy křivky [25].

$$H(u, v) = e^{-D^2(u,v)/(2\sigma^2)} \quad (7)$$

Následujícím krokem je samotná hranová detekce. Použitím Sobelova operátoru jsou z obrazu extrahovány velikosti gradientů hran a jejich směry. Sobelův operátor je k tomuto použití vhodný především z důvodu nízké citlivosti na šum.

Po získání gradientu a jeho směru se Cannyho detektor zaměřuje na ztenčení hran. Mezi gradienty tedy hledá pouze lokální maxima, ostatní gradienty, které nejsou lokálními maximy, zahazuje. Zůstávají tedy pouze tenké linky, které jsou dále zpracovány jako kandidáti na hrany.

Posledním krokem hranového detektoru je prahování s hysterezí. V této fázi Cannyho detektor vybírá pouze „důležité“ hrany a zbavuje se těch nejmenších. Jsou zvoleny dvě prahové hodnoty $T1$ pro nižší práh a $T2$ pro práh vyšší. Dále algoritmus porovnává velikosti jednotlivých gradientů hran s prahy. Je-li gradient vyšší než práh $T2$, je označen jako hranový. Pokud je gradient nižší než $T1$, nejedná se o hranový pixel. V případě, že se hodnota gradientu pixelu nachází mezi oběma prahy $T1$ a $T2$, bude pixel označen jako hranový pouze v případě, že některý z jeho sousedních pixelů leží nad hranicí prahu $T2$. Doporučený poměr prahů $T2$ a $T1$ je mezi 2 : 1 a 3 : 1 [18, 24, 25].

3.2.1.2 Metoda U-křivky

Tato metoda vychází z faktu, že duhovka bývá mnohdy částečně zakryta horním očním víčkem. Okraj viditelné části duhovky potom připomíná tvar písmene „U“. Metoda U-křivky pracuje s oblastí oka detekovanou pomocí Haarových příznaků a s hranami v této oblasti nalezenými Cannyho detektorem. Cílem metody je nalezení nejdelší hrany připomínající tvar písmene „U“ [25].

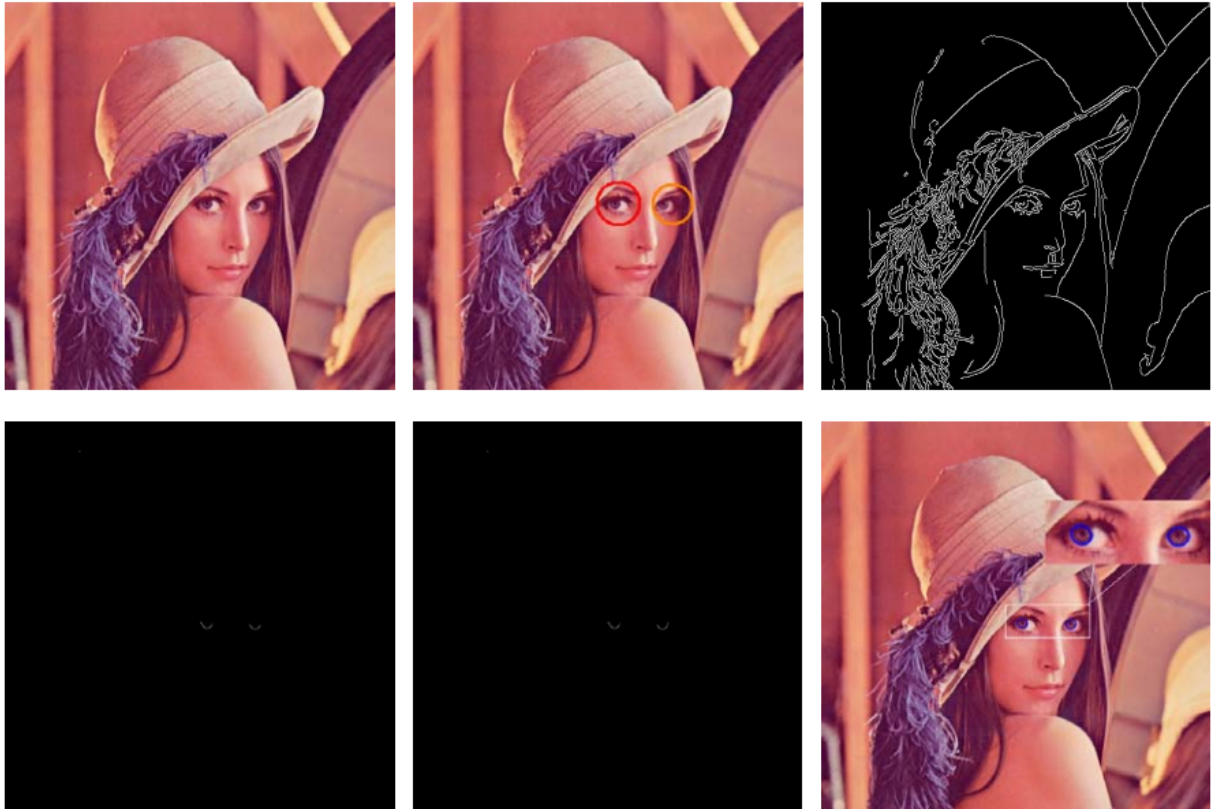
Algoritmus se skládá z pěti kroků, které na sebe navazují. Postupně je prováděn se všemi obrazovými body oka ve směru od levého horního rohu k pravému spodnímu:

1. První krok zkontroluje, jestli je daný obrazový bod hranou a jestli zatím nebyl řešen. Označí jej za vyřešený a přidá jej do nové hranové linie s délkou rovnou 1. Pokud pixel nespĺnil některou z podmínek, pokračuje dalším krokem.

2. V druhém kroku probíhá hledání pokračování hrany ve směru vpravo a dolů. Podmínky jednotlivých podbodů jsou postupně kontrolovány, pokud není jedna splněna, kontroluje se další, atd.
 - (a) Zkontroluje, jestli pravý sousední pixel je hranový a zatím nebyl řešen, pokud je to tak, je tento pixel označen jako vyřešený a je přidán k aktuální hranové linii, jejíž délka je zvětšena o 1. Pokračuje se znovu krokem č. 2.
 - (b) Zkontroluje, jestli spodní sousední pixel je hranový a zatím nebyl řešen, pokud je to tak, je tento pixel označen jako vyřešený a je přidán k aktuální hranové linii, jejíž délka je zvětšena o 1. Pokračuje se znovu krokem č. 2.
 - (c) Zkontroluje, jestli pravý spodní sousední pixel je hranový a zatím nebyl řešen, pokud je to tak, je tento pixel označen jako vyřešený a je přidán k aktuální hranové linii, jejíž délka je zvětšena o 1. Pokračuje se znovu krokem č. 2.
 - (d) Pokud nebyl proveden ani jeden z podbodů, pokračuje se krokem č. 3.
3. Ve třetím kroku probíhá hledání pokračování hrany ve směru vpravo a nahoru. Podmínky jednotlivých podbodů jsou postupně kontrolovány, pokud není jedna splněna, kontroluje se další, atd.
 - (a) Zkontroluje, jestli pravý sousední pixel je hranový a zatím nebyl řešen, pokud je to tak, je tento pixel označen jako vyřešený a je přidán k aktuální hranové linii, jejíž délka je zvětšena o 1. Pokračuje se znovu krokem č. 3.
 - (b) Zkontroluje, jestli horní sousední pixel je hranový a zatím nebyl řešen, pokud je to tak, je tento pixel označen jako vyřešený a je přidán k aktuální hranové linii, jejíž délka je zvětšena o 1. Pokračuje se znovu krokem č. 3.
 - (c) Zkontroluje, jestli pravý horní sousední pixel je hranový a zatím nebyl řešen, pokud je to tak, je tento pixel označen jako vyřešený a je přidán k aktuální hranové linii, jejíž délka je zvětšena o 1. Pokračuje se znovu krokem č. 3.
 - (d) Pokud nebyl proveden ani jeden z podbodů, pokračuje se krokem č. 4.
4. Délka hranové linie je porovnána s prahovou hodnotou, pokud je vyšší, je přidána na seznam kandidátů, pokračuje se krokem č. 5.
5. Na závěr je vybrána nejdelší hranová linie, která je považována za okraj duhovky.

3.2.1.3 Optimalizace vzorků

Množina pixelů tvořících hranovou křivku je dále podrobena optimalizačním úpravám. Nejprve je rozšířen počet vzorků křivky. Dále následuje snaha o snížení šumu, který se projevuje především u začátku a konce hranové křivky. Podle geometrických pravidel elipsy se všechny její body nacházejí na stejné straně tangenty kteréhokoli jejího bodu. Algoritmus se tedy postará o odstranění malého množství bodů, které tuto podmínku nesplňují [25].



Obrázek 15: Postup detekce částečně zakryté duhovky [25].

3.2.1.4 Určení elipsy duhovky

Na závěr je k optimalizované hranové křivce nalezena elipsa kopírující její tvar. K nalezení odpovídající elipsy se používá metoda nejmenších čtverců. Na obrázku 15 jsou ilustrovány jednotlivé kroky detektoru. V horní části se vlevo nachází původní obrázek, uprostřed obrázek s detekovanými očními oblastmi (detekce Haarovými příznaky kapitola 3.1.1 nebo pomocí landmarků, kapitola 2), vpravo detekované hrany (Cannyho detektor hran, kapitola 3.2.1.1). Ve spodní části obrázku je potom zleva snímek s *U-křivkami* (kapitola 3.2.1.2), uprostřed po optimalizaci vzorků (kapitola 3.2.1.3) a vpravo samotný výsledek detekce [25].

3.2.2 Detekce duhovky metodou Houghových kružnic

Dalším příkladem detektoru duhovky založeném na tvaru je metoda využívající detekci Houghových kružnic. Příkladem může být přístup publikovaný v článku *Eye state analysis using iris detection based on Circular Hough Transform* [26]. Autoři této práce uvádějí, že výsledky byly testovány na videozáznamech zachycených prostřednictvím levné webové kamery, což vypovídá o poměrně dobré použitelnosti této metody i se slabší kvalitou záznamu (jiné metody často využívají různých přísvitů a kamer s vysokým záznamovým rozlišením). Před samotnou detekcí

duhovek algoritmus nejprve detekuje obličej sledované osoby ve snímku a jeho oční oblasti. Dále pokračuje detekcí Houghových kružnic [26].

3.2.2.1 Detekce hran

Budeme-li se na hranu dívat jako na místo s výraznou změnou jasové funkce, bude po vypočtení první derivace v tomto bodě vysoká hodnota jasové funkce. Maximální hodnota potom bude ve směru kolmo k hraně. Pro zjednodušení a urychlení výpočtu hran se používá aproximace derivace konvolucí s určitým jádrem. Existuje celá řada operátorů, z nichž každý využívá různého konvolučního jádra. Mezi nejznámější a nejčastěji používané operátory patří:






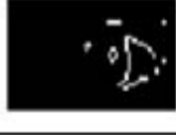






- Sobelův operátor $\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$,
- Prewittův operátor $\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$,
- Robertsův operátor $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$,
- Kirschův operátor $\begin{pmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{pmatrix}$,
- Laplaceův operátor $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$,
- Laplaceův operátor s Gaussovým filtrem (LoG - Laplacian of Gaussian)

$$\begin{pmatrix} 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -24 & -40 & -24 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \end{pmatrix}$$

Jako nejlepší algoritmus pro detekci hran bývá představován Cannyho hranový detektor, který kromě aproximace první derivace obrazu Gaussovým filtrem navíc využívá prahování s hysterezí (kapitola 3.2.1.1) [26, 27].

Autoři tohoto přístupu však uvádějí, že hrany detekované těmito klasickými filtry nebyly dostatečně uspokojivé pro účely detekce kruhového tvaru duhovky. Z tohoto důvodu představují vlastní specializovaný detektor hran duhovky. Ten převede obraz do odstínů šedé a vyřadí z detekce pixely, které mají intenzitu jasu vyšší než odhadnutý práh th_{edges} . Práh th_{edges} je zvolen tak, aby vyřadil z detekce pixely, které nesouvisí se samotnou duhovkou.

Po selekci pixelů prahem th_{edges} je pro každý jednotlivý pixel x z této množiny vybráno n levých sousedních pixelů a n pravých sousedních pixelů. Dále je vypočítán rozdíl daného pixelu x s jeho levými sousedy a analogicky rovněž rozdíl x s pravými sousedy. Tyto rozdíly jsou porovnávány se dvěma dalšími prahovými hodnotami th_{high} a th_{low} . Je-li vypočítaný rozdíl vyšší než th_{high} nebo nižší než th_{low} , jedná se o hranový pixel duhovky a je označen 1. Prahová hodnota th_{high} by měla být vhodně zvolena, aby byla schopna rozlišit rozdíly jasu mezi duhovkou a bělmem. Podobně je potřeba vybrat i vhodnou prahovou hodnotu th_{low} , která bude respektovat podobnost pixelů duhovky.

	Closed eye	Open eye
Original image		
Prewitt		
Sobel		
LoG		
Canny		
Our edge detector		

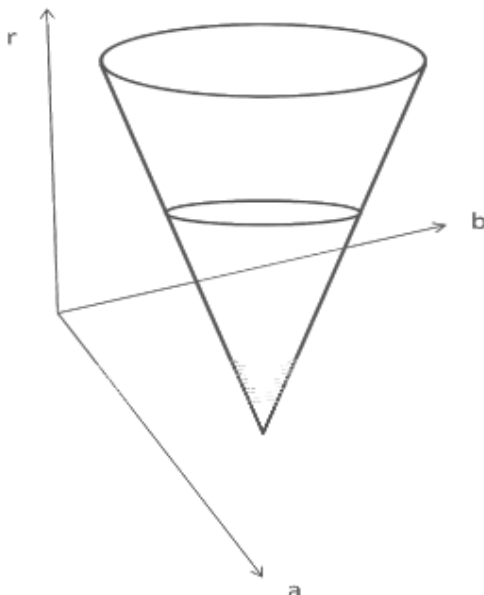
Obrázek 16: Hranová detekce duhovky. Srovnání metody představené v kapitole 3.2.2.1 a klasických metod [26].

3.2.2.2 Houghova kružnicová transformace

Jde o zvláštní formu Houghovy transformace zaměřující se na hledání obrazových objektů, jejichž tvar se blíží tvaru kružnice. Houghova transformace může být popsána, jako transformace bodu z kartézského prostoru do parametrického prostoru definovaného v závislosti na daném objektu zájmu. V případě kružnicové Houghovy transformace se při transformaci používá rovnice kružnice (rovnice (8)), kde r je poloměr kružnice, (a, b) je střed kružnice a (x, y) je bod na kružnici. Podle této rovnice by bod (x, y) po provedení Houghovy transformace měl ve výsledném parametrickém prostoru tvar (a, b, r) .

$$r^2 = (x - a)^2 + (y - b)^2 \quad (8)$$

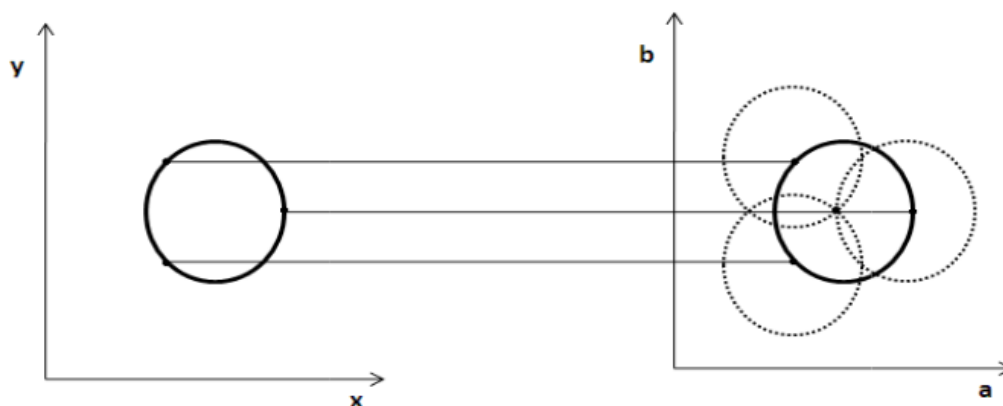
Algoritmus provádějící Houghovu kružnicovou transformaci na vstupním obraze by potom v prvním kroku detekoval všechny hrany, které se v obraze nacházejí (např. Cannyho hranovým detektorem, kapitola 3.2.1.1).



Obrázek 17: Parametrický prostor pro Houghovu kružnicovou transformaci [30].

V každém detekovaném hranovém bodě vyneseme kružnici se středem v tomto bodě a požadovaným poloměrem (poloměr nemusí být fixní, ale může být zadán intervalem, v tom případě se ve všech hranových bodech vykreslí kružnice pro každý z těchto poloměrů). Tyto kružnice jsou vykresleny v parametrickém prostoru (a, b, r) . Pro zjednodušení výkladu se dále počítá s fixním poloměrem kružnic.

V dalším kroku budeme inkrementovat hodnoty akumulární matice, která má stejné rozměry jako parametrický prostor. Pro všechny body, ležící na obvodu jednotlivých kružnic vykreslených v předchozím kroku, budou inkrementovány odpovídající body v akumulární matici. Stručně



Obrázek 18: Houghova kružnicová transformace [30].

vzato projdeme všechny hranové pixely, vytvoříme k nim kružnice požadovaných rozměrů a podle těchto kružnic inkrementujeme hodnoty odpovídajících bodů v akumulační matici.

Po projití všech hranových bodů a všech požadovaných poloměrů obsahuje naplněná akumulační matice informace o objektech ve vstupním obraze, které se svým tvarem blíží tvaru kružnice. Nejvyšší čísla v této matici odpovídají nalezeným kružnicím v obraze [26, 30].

3.2.2.3 Určení polohy duhovky

Algoritmus určuje obvod duhovky pomocí Houghovy kružnicové transformace (kapitola 3.2.2.2) a nalezených hranových pixelů detekovaných pomocí speciálního algoritmu (kapitola 3.2.2.1). V každé iteraci algoritmu jsou náhodně vybrány tři hranové pixely, které splňují následující dvě podmínky:

- vybrané pixely nejsou kolineární (neleží na stejné přímce),
- vzdálenost mezi každou dvojicí těchto tří pixelů je vyšší než fixní práh t_{dist_px} .

Pokud jsou tyto podmínky splněny, je vypočten poloměr a souřadnice středu kružnice definované těmito vybranými třemi body. Pokud jsou všechny možné parametry této kružnice v daném prahovém rozmezí, jsou přiřazeny do akumulační matice. V dalším kroku je vypočtena vzdálenost středu kružnice a všech hranových pixelů. Tato vzdálenost je porovnávána s další prahovou hodnotou t_{dist_cir} , pokud je menší než práh, je počítadlo pixelů kružnice zvýšeno o 1. Pokud toto počítadlo překročí hodnotu prahu $th_{counter}$, vytvořená kružnice může představovat duhovku. Zbývající hranové pixely jsou využity při následující iteraci k hledání dalších kružnic.

Algoritmus končí, když je množina hranových pixelů nižší než určité množství nebo bylo dosaženo stanoveného počtu iterací. Za výslednou kružnici a tedy obvod duhovky je z množiny detekovaných kružnic vybrána ta s největším poloměrem.

4 Použité nástroje

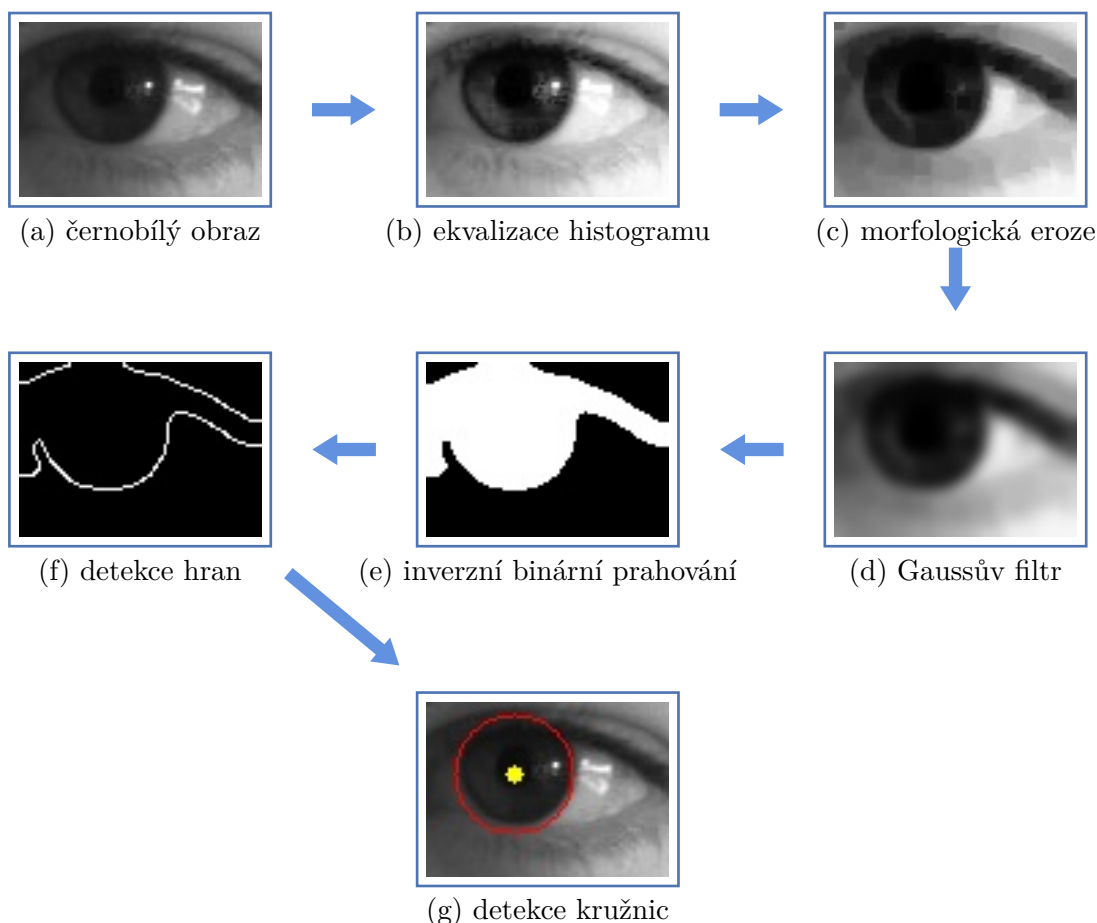
Veškerá implementace praktické části této práce byla provedena v programovacím jazyce C++. Tento jazyk byl zvolen především z toho důvodu, že umožňuje použití největší dostupné otevřené knihovny OpenCV pro zpracování obrazu, počítačové vidění nebo strojové učení. Tato knihovna se skládá z širokého množství metod, nejčastěji používaných v těchto oblastech. Počet optimalizovaných algoritmů této knihovny převyšuje 2500. Byla použita verze knihovny OpenCV 2.4.11 [17].

Pro vývoj byl použit notebook MacBook Air (začátek roku 2014) s procesorem CPU 1.4 GHz Intel[®] Core[™]i5, operační pamětí DDR3 RAM 8GB 1600 MHz a operačním systémem OS X El Captain verze 10.11.4.

5 Vlastní detektor úhlu pohledu

Předchozí kapitoly této práce byly postupně zaměřeny na detekci významných obličejových bodů, tzv. landmarků (kapitola 2) a na detekci zornic a duhovek (kapitola 3). Obě tyto části nejprve popisovaly motivaci k daným úlohám a jejich možné využití. Po tomto úvodu, který měl za úkol seznámit s užitečností daných metod, byly dále popsány některé známé přístupy detekcí jak landmarků, tak duhovek a zornic.

Toto úvodní zkoumání bylo zároveň průvodním krokem k vytvoření vlastního detektoru úhlu pohledu. Ten by se měl primárně zaměřovat na detekci duhovky (resp. zornice) a pomocí znalosti její polohy v oblasti oka určit úhel pohledu pozorované osoby. Jako metoda předzpracování detektoru duhovky je určena metoda detekce landmarků obličeje, která pomáhá definovat oční oblasti, na kterých následně detektor duhovky pracuje. Jak vyplývá ze samotného zadání této diplomové práce, je pro detekci landmarků možno využít volně dostupných algoritmů. Pro tento účel byl po úvodním zkoumání zvolen detektor popsáný v kapitole 2.1. Důvodem této volby je vysoká rychlost detektoru. Autoři pro zpracování jednoho snímku při konfiguraci s důrazem na rychlost uvádějí až 3000 FPS, při nastavení vyšší přesnosti 300 FPS.



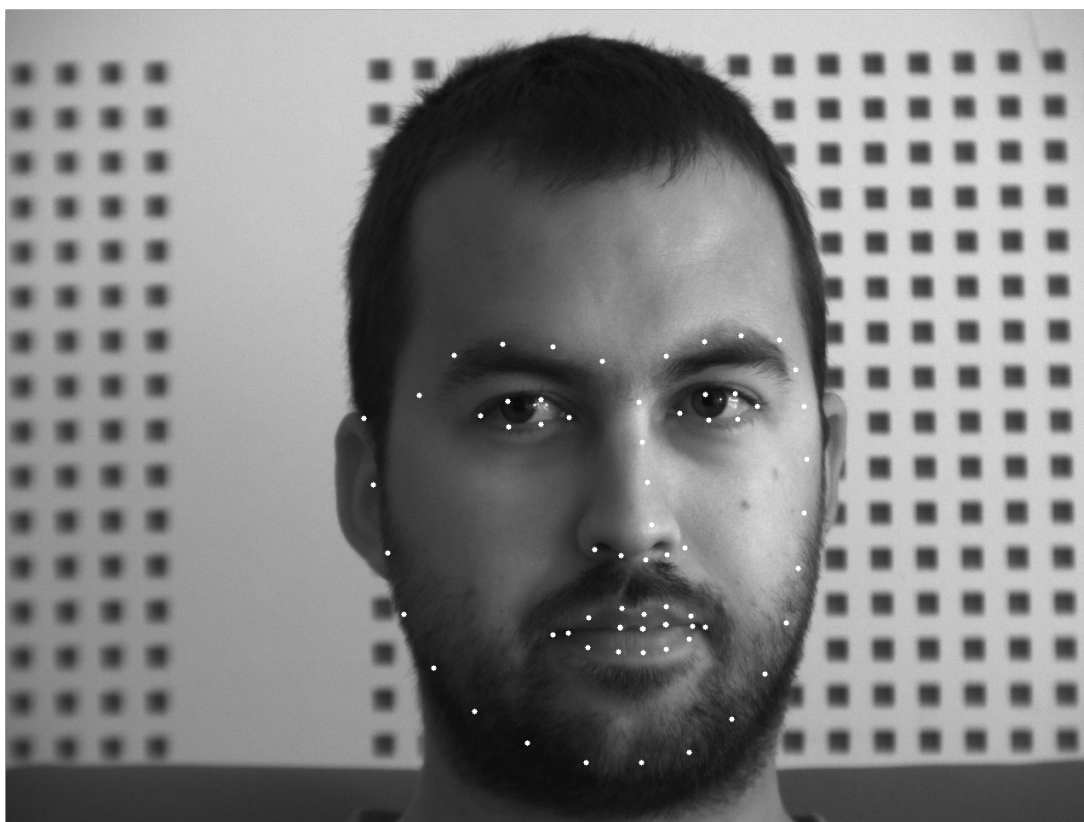
Obrázek 19: Diagram detekce duhovky.

5.1 Vstupní data

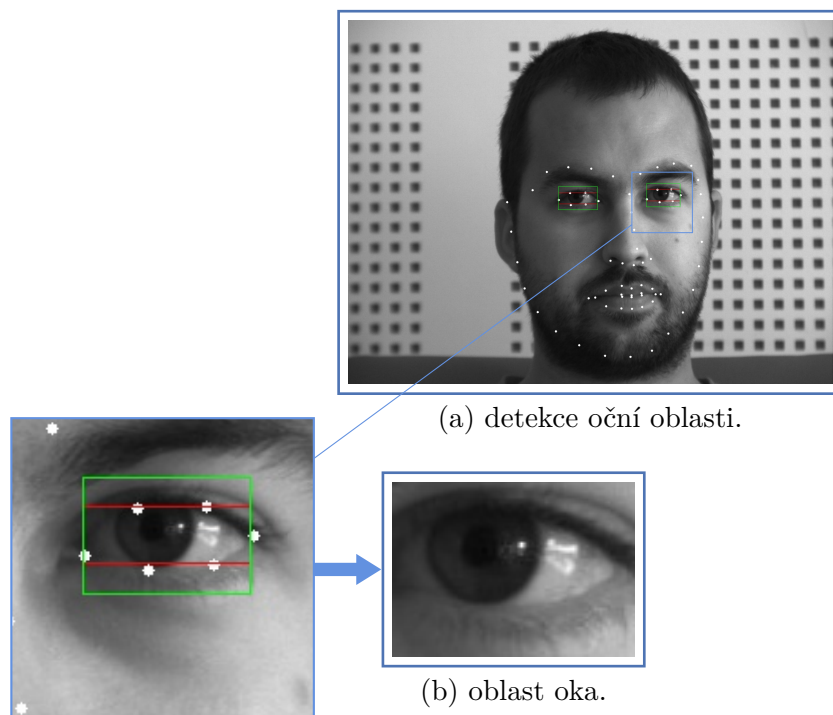
Datovým zdrojem vytvořené metody může být jak jednosnímkový obrázek, tak kontinuální obrazový záznam v podobě souboru videa nebo přenosu z kamery.

5.2 Detekce významných bodů obličeje

Jak již bylo zmíněno dříve, algoritmus detekce úhlu pohledu vytvořený v rámci této diplomové práce využívá k nalezení očních oblastí metodu detekce landmarků. Samotné vytvoření a implementace detektoru landmarků není cílem této práce, důraz je kladen především na její použití v kombinaci s detektorem duhovek a zornic. Z tohoto důvodu byla použita hotová implementace metody popsané v článku *Face Alignment at 3000 FPS via Regressing Local Binary Features* [2], která již byla podrobněji popsána v kapitole 2.1. Implementace tohoto detektoru landmarků byla získána z veřejně dostupného zdroje webové verzovací služby GitHub [28]. Toto softwarové dílo je svým autorem Lequan Yu licencováno svobodnou licencí MIT, která umožňuje jeho použití v proprietárním softwaru. Na obrázku 20 je příklad vzorového obrázku s detekovanými landmarky.



Obrázek 20: Příklad detekce významných bodů obličeje - vyznačeny bílými body.

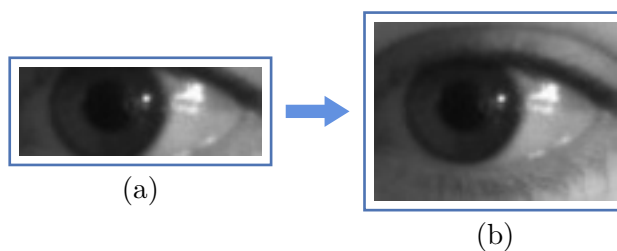


Obrázek 21: Příklad detekce oční oblasti. (a) významné body obličeje - bílé body, očních oblastí - červený obdélník, rozšířená oční oblast - zelený obdélník. (b) výsledná oční oblast.

5.3 Detekce oční oblasti

Dalším krokem algoritmu je získání očních regionů. Tento krok umožňuje detekci duhovky v relevantních oblastech oka, což představuje významné snížení detekční oblasti a tedy urychlení samotné detekce.

Daný detektor landmarků zaznamenává celkem 6 očních bodů. Pro získání výřezu oblasti oka je použit levý a pravý landmark, průměr dvou horních landmarků a průměr dvou spodních landmarků. Získané čtyři body určují oblast jednoho oka (vyznačena červeným obdélníkem na obrázku 21).



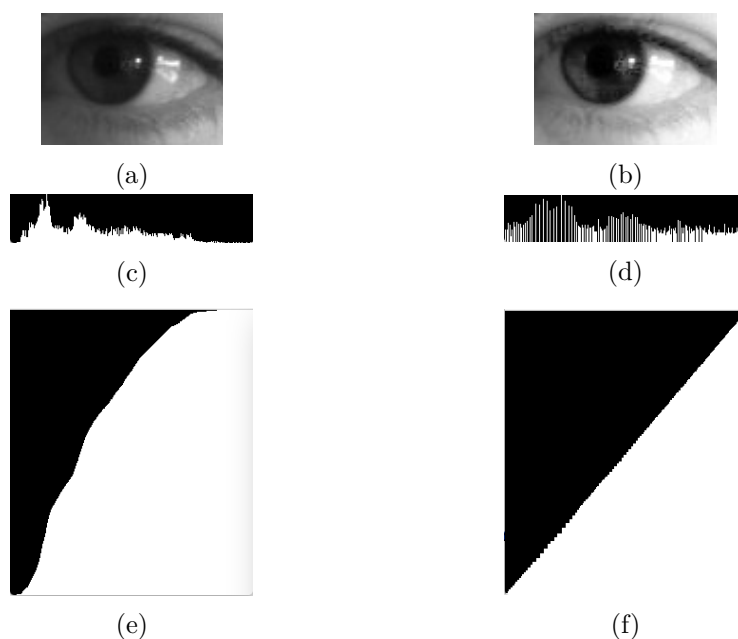
Obrázek 22: Rozšíření oblasti oka. (a) původní oblast oka, (b) rozšířená oblast oka.

Následný vývoj však ukázal, že detektor landmarků ne vždy přesně vyhodnotí oční oblast, tím často docházelo ke ztrátě informací potřebných k detekci duhovek. Z tohoto důvodu byla

oblast výřezu oka rozšířena na dvojnásobek své výšky (obrázek 22). Tato rozšířená oblast je vyznačena na obrázku 21 pomocí zeleného obdélníku.

5.4 Ekvalizace histogramu

V tomto kroku algoritmus již zpracovává samotnou oblast oka. Dříve než je přistoupeno k ekvalizaci, je obraz převeden do stupňů šedi. Dále tedy pracujeme s černobílým obrázkem a histogramem si můžeme vyjádřit četnost zastoupení jednotlivých intenzit jasu v obraze. Ekvalizací histogramu dosáhneme roztažení rozsahu těchto jasových intenzit. Jinými slovy dojde ke zvýšení kontrastu obrazu. Příklad této procedury včetně samotných histogramů a kumulativních histogramů je ilustrován na obrázku 23 [19].



Obrázek 23: Ukázka ekvalizace histogramu oblasti oka. (a) oblast oka před ekvalizací, (b) oblast po ekvalizaci, (c) histogram před ekvalizací, (d) histogram po ekvalizaci, (e) kumulativní histogram před ekvalizací, (f) kumulativní histogram po ekvalizaci.

5.5 Morfologická operace eroze

Dalším krokem algoritmu je aplikování morfologické operace eroze. Ta má v tomto případě za úkol odstranění nežádoucích odlesků světla od rohovky, které nepříznivě ovlivňují detekci duhovky. Eroze způsobí zvýraznění tmavých částí duhovky a zornice, její mírné zvětšení, ale především překrytí vnitřních odlesků. Jako jádro pro tuto morfologickou operaci byla použita matice jedniček 3×3 , která se během testů algoritmu ukázala jako nejefektivnější řešení. Ukázka aplikace eroze je na obrázku 24. Je možné použít i jádro větších rozměrů (např. 5×5), v tom případě ale může dojít k významějšímu ztmavení očního bělma v oblasti blízko duhovce. Při

následném binárním prahování (kapitola 5.7) potom může docházet až ke spojení oblasti duhovky s tmavou oblastí okraje očního víčka. Tento jev lze však částečně odstranit snížením prahové hodnoty při binárním prahování [20].



Obrázek 24: Ukázka po aplikaci morfologické operace eroze na oblast oka.

5.6 Vyhlazení Gaussovým filtrem

Po erozi obrazu následuje vyhlazení obrazu, které pomůže odstranit případný nežádoucí šum. K tomu je použit Gaussův filtr s jádrem o velikosti 9×9 a standardní odchylkou $\sigma = 2$, platnou pro obě osy matice jádra. Z obrázku 25, který ukazuje stav po aplikaci Gaussova filtru, je patrné snížení lokálních extrémů vysokého jasu části bělma a naopak mírné zvýšení v nejtmaší oblasti duhovky [21].



Obrázek 25: Ukázka oblasti oka po vyhlazení Gaussovým filtrem.

5.7 Binární prahování

Dále algoritmus provádí segmentaci obrazu v závislosti na hodnotě jasu. Tento krok vychází z toho, že oblast duhovky nebo případně zornice je dostatečně tmavší než zbytek oční oblasti. Je tedy aplikováno inverzní binární prahování, které rozdělí obraz na hodnoty pod a nad prahem. Hodnoty s vyšším jaselem, než je práh, jsou označeny 0, hodnoty s nižším jaselem 1. Je důležité stanovit takový práh, který bude správně vyhodnocovat duhovku jako tmavší a bělmo, jako světlejší. V závislosti na světelných podmínkách musí tedy být zvolena vyhovující prahová hodnota. Výchozí práh byl po testování algoritmu stanoven na hodnotu 60 (hodnota jasu pixelu je definována na rozsahu 0 – 255, kde 0 je černá barva a 255 bílá). Příklad prahování je ukázán na obrázku 26 [22].



Obrázek 26: Ukázka oblasti oka po provedení prahování.

5.8 Detekce hran

V dalším kroku následuje procedura hledání hran, která zpracovává binární obraz, jež je výstupem předchozího prahování. Pro tento účel je použit Cannyho detektor, který již byl popsán v teoretické části v kapitole 3.2.1.1. Výsledkem jsou nalezené hranové linie oční oblasti. Ukázka výstupu této detekce je na obrázku 27.



Obrázek 27: Ukázka detekovaných hran v oblasti oka.

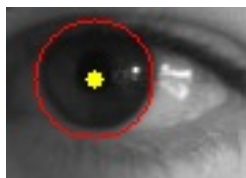
5.9 Detekce kružnic

Posledním krokem při detekci duhovky je hledání kružnic. Jako svůj vstup si bere obraz oční oblasti s hranovou reprezentací, který byl výstupem detektoru hran (kapitola 5.8). Jejím výstupem by v ideálním případě měl být zjištěný obvod duhovky, se středem v oblasti zornice. Pro nalezení této obvodové kružnice byla použita Houghova transformace specializovaná na kružnice. Tato metoda byla již rovněž popsána v teoretické části v kapitole 3.2.2.2.

Hodnoty výchozích parametrů byly stanoveny jako relativní. Důvodem je zaručení toho, aby se zajistilo stejné nastavení metody při různých rozměrech obrazu oční oblasti. V porovnání s parametry prahování a eroze se v tomto případě jedná o parametry, které mají závislost na velikosti obrazu oční oblasti. Ve výchozí konfiguraci musí být vzdálenost mezi detekovanými kružnicemi vyšší než $1/4$ výšky oční oblasti. Práh akumulární matice při určování středů kružnic byl stanoven na hodnotu 10 a poloměr detekovaných kružnic se musí nacházet v rozmezí $1/4$ až $2/5$ výšky oční oblasti.

Optimální nastavení těchto parametrů by mělo v ideálním případě zajistit detekci jediné kružnice, která bude správně detekována jako obvod duhovky. Přesto se může stát, že detektor naleznе více kružnic a je potřeba nějakým mechanismem určit, která z nich představuje požadovaný okraj duhovky. Pro tento účel je v metodě použito následující kontroly, která by měla tyto chybné detekce vyřadit. Pro každou z kružnic detekovanou Houghovou transformací

je z obrazu získaného v kroku prahování (kapitola 5.7) vzata čtvercová oblast s hranou velikosti $1/10$ šířky oční oblasti a středem odpovídajícím středu dané kružnice. Jedná se o binární obraz, kde oblast duhovky je reprezentována bílou barvou a pozadí černou, pokud tedy kružnice odpovídá duhovce, musí její střed obsahovat bílou barvu. Poměrné zastoupení bílých pixelů v tomto čtverci je poté porovnáno s konstantou 0.9, je-li vyšší, je čtverec považován za bílý a kružnice je schválena jako obvod duhovky. V opačném případě je kružnice zamítnuta. Ukázka výstupu této metody je na obrázku 28.



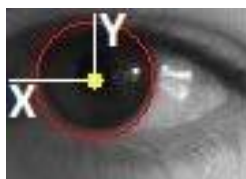
Obrázek 28: Ukázka detekce kružnic v oblasti oka. Detekovaná kružnice - červená, střed kružnice - žlutý bod.

5.10 Určení úhlu pohledu

Předchozí kroky sloužily k detekci obvodu duhovky a jejího středu. Pomocí těchto informací algoritmus vyhodnotí úhel, kterým se daná pozorovaná osoba dívá. Pro tento účel je využito polohy středu duhovky vzhledem k oční oblasti. Daný střed je porovnán s okraji oční oblasti a na základě toho je určeno, jakým směrem se osoba dívá. Úhel pohledu je vypočítáván pomocí rovnice (9) pro x -ovou osu a pomocí rovnice (10) pro y -ovou osu, kde $width$ je šířka oční oblasti a $height$ je výška oční oblasti. Hodnoty X a Y jsou potom ilustrovány na obrázku 29.

$$angle_X = \left(\frac{X}{width} - 0.5 \right) \times 150 \quad (9)$$

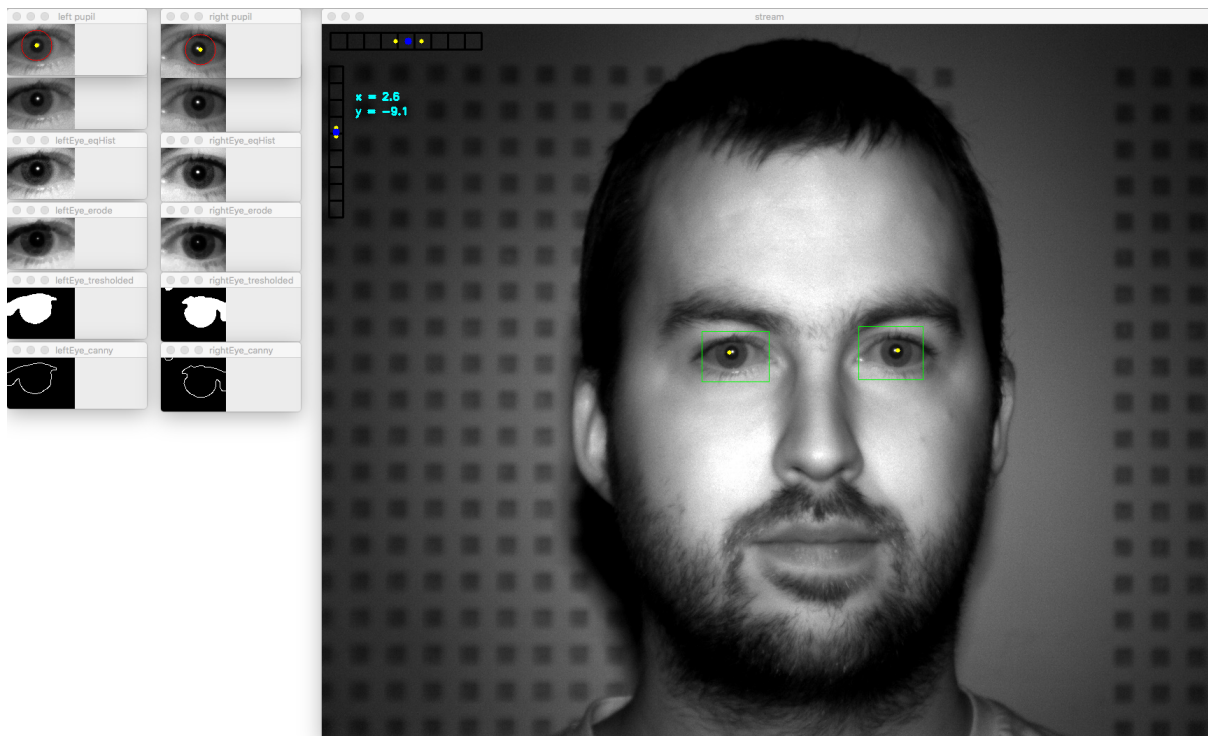
$$angle_Y = \left(\frac{Y}{height} - 0.5 \right) \times 150 \quad (10)$$



Obrázek 29: Ukázka určení polohy středu duhovky v oční oblasti.

Výsledná detekce úhlu pohledu se nachází na obrázku 30. V levé části obrázku jsou vykresleny jednotlivé kroky detekce pro oba oční regiony. V části pravé se potom nachází původní vstupní snímek s vyznačenými očními oblastmi a detekovanými středy duhovek. V levé horní části tohoto snímku jsou potom vykresleny dvě stupnice, které indikují směr pohledu v horizon-

tální a vertikální ose. Tato indikace je na každé ze stupnic prezentována dvěma žlutými body, z nichž každý odpovídá úhlu pohledu detekovanému v jednom očním regionu. Modrý bod potom představuje průměrnou hodnotu těchto dvou detekcí a zároveň výsledný úhel pohledu. Světle modrou barvou jsou v tomto snímku vypsány výsledné úhly.



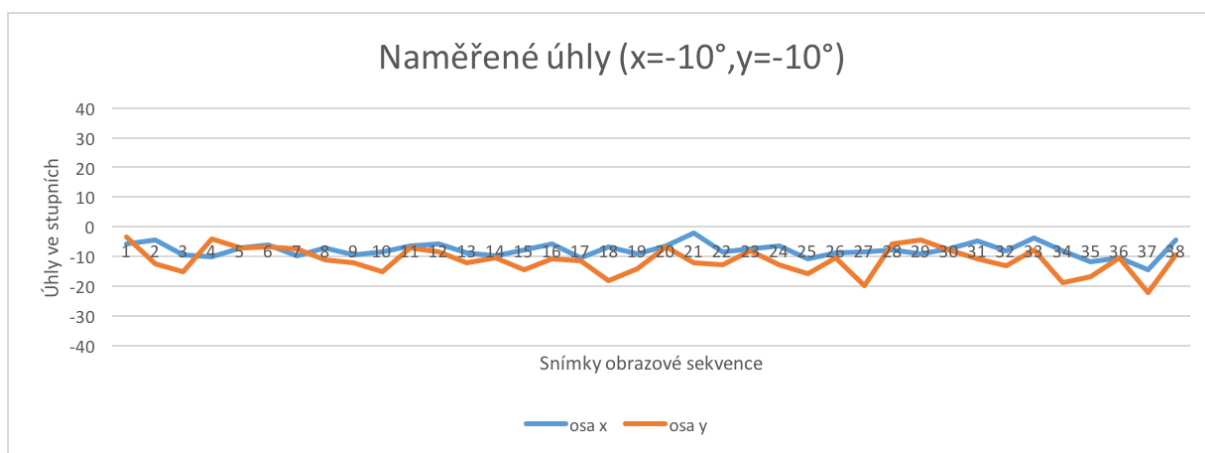
Obrázek 30: Ukázka detekce úhlu pohledu. V levé části jednotlivé kroky detekce, v pravé obraz se stupnicemi vyznačující směr pohledu a detekovanými úhly v obou osách - světle modře, vyznačenými očními oblastmi - zelené obdélníky a středy duhovek - žluté body.

6 Testování vytvořeného detektoru

Tato kapitola je zaměřena na experimentální testování vytvořeného algoritmu pro detekci úhlu pohledu. Na testovacích datech bude vyhodnocena úspěšnost detekce úhlu pohledu. Dále zde bude popsána zjištěná rychlost jednotlivých součástí algoritmu.

6.1 Přesnost detektoru

Přesnost detekce byla testována na videozáznamu pořízeném v laboratoři digitálního zpracování obrazu VŠB-TU Ostrava katedry Informatiky. Záznam byl pořízen v rozlišení 1280×1024 s použitím infračervené přísvitové lampy. K tomuto záznamu byl rovněž vytvořen soubor s pravdivými (tzv. *ground truth*) hodnotami úhlů pohledu pro jednotlivé snímky videosekvence. Porovnáním dat naměřených vytvořeným algoritmem detekce úhlu pohledu s těmito hodnotami lze určit, s jakou přesností detekce algoritmus pracuje.

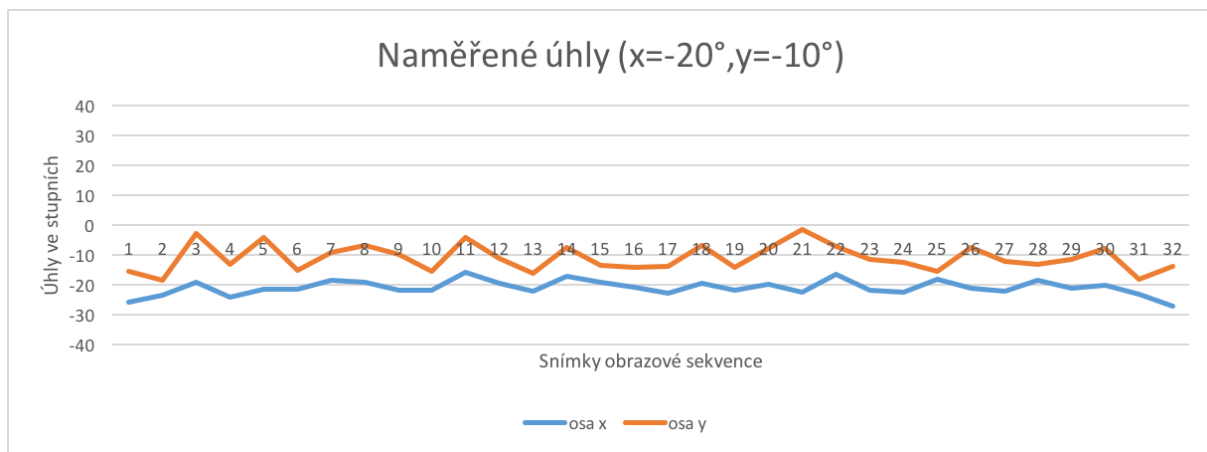


Obrázek 31: Úspěšnost algoritmu na úhlu pohledu $x = -10^\circ$, $y = -10^\circ$.

Graf na obrázku 31 popisuje přesnost detekce pro úhel pohledu $x = -10^\circ$, $y = -10^\circ$. Vyhodnocení detekce:

- průměrná hodnota úhlu v x -ové ose: -7.80° ,
- průměrná hodnota úhlu v y -ové ose: -11.17° ,
- směrodatná odchylka v x -ové ose: 2.36° ,
- směrodatná odchylka v y -ové ose: 4.40° .

Detekce pro úhel pohledu $x = -10^\circ$, $y = -10^\circ$ vykazuje v x -ové ose průměrnou chybu 2.20° se směrodatnou odchylkou 2.36° . Z toho vyplývá, že měření v horizontální rovině bylo v tomto případě poměrně stabilní a do jisté míry i přesné. V y -ové ose dosahuje průměrná chyba hodnoty



Obrázek 32: Úspěšnost algoritmu na úhlu pohledu $x = -20^\circ$, $y = -10^\circ$.

1.17° , což je ve srovnání s x -ovou osou téměř poloviční hodnota. Směrodatná odchylka 4.40° je naproti tomu ovšem známkou nižší stability detekce ve vertikální rovině.

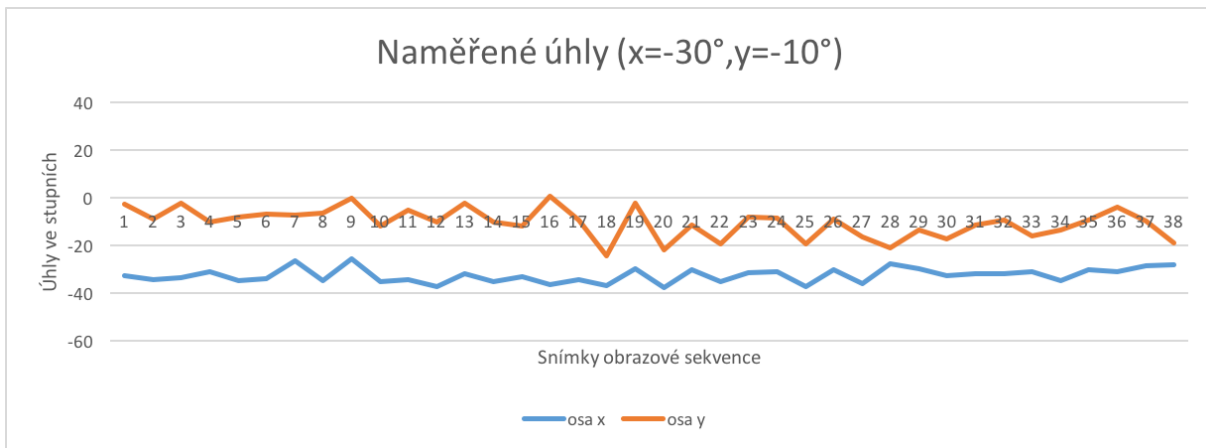
Další graf na obrázku 32 popisuje přesnost detekce pro úhel pohledu $x = -20^\circ$, $y = -10^\circ$. Vyhodnocení detekce:

- průměrná hodnota úhlu v x -ové ose: -21.03° ,
- průměrná hodnota úhlu v y -ové ose: -11.09° ,
- směrodatná odchylka v x -ové ose: 2.45° ,
- směrodatná odchylka v y -ové ose: 4.39° .

Výsledky detekce úhlu pohledu $x = -20^\circ$, $y = -10^\circ$ byly ve srovnání s předchozím úhlem $x = -10^\circ$, $y = -10^\circ$ velmi podobné. Lepší hodnoty ovšem nyní dosáhla průměrná chyba v x -ové ose, kde s hodnotou 1.03° vykazovala poměrně dobrou přesnost detekce. Směrodatná odchylka 2.45° opět ukazuje na poměrně stabilní detekci. Co se y -ové osy týče je průměrná chyba 1.09° velmi dobrým výsledkem, ale se směrodatnou odchylkou 4.39° se jedná opět o velmi nestabilní detekci.

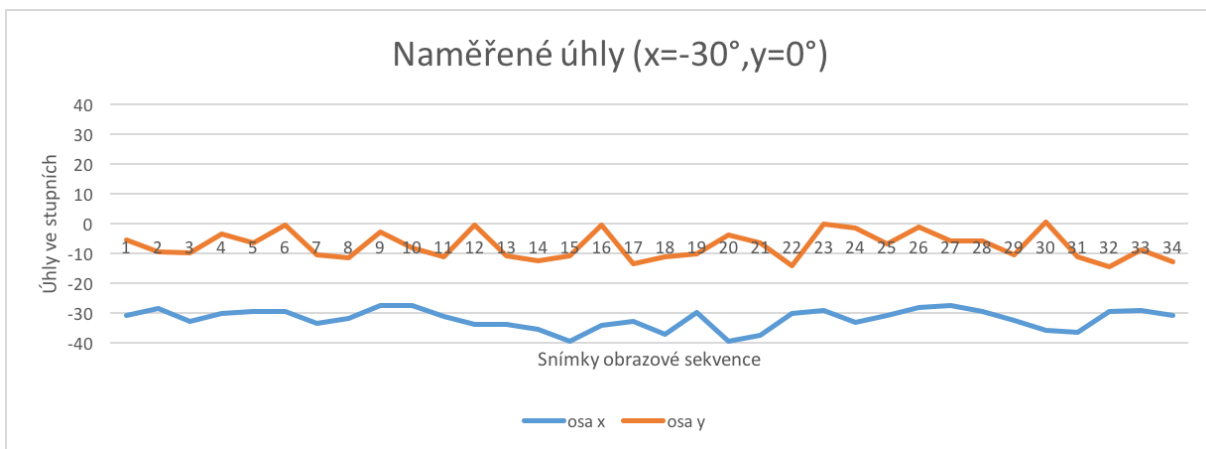
Následný graf na obrázku 33 popisuje přesnost detekce pro úhel pohledu $x = -30^\circ$, $y = -10^\circ$. Vyhodnocení detekce:

- průměrná hodnota úhlu v x -ové ose: -32.62° ,
- průměrná hodnota úhlu v y -ové ose: -10.51° ,
- směrodatná odchylka v x -ové ose: 3.11° ,
- směrodatná odchylka v y -ové ose: 6.15° .



Obrázek 33: Úspěšnost algoritmu na úhlu pohledu $x = -30^\circ$, $y = -10^\circ$.

Detekce úhlu pohledu $x = -30^\circ$, $y = -10^\circ$ byla naměřena s průměrnou chybou 2.62° na x -ové ose, tato chyba se již ukazuje jako vyšší ve srovnání s předchozími detekcemi. Podobnou tendenci zaznamenává také směrodatná odchylka, která ukazuje nižší stabilitu detekce. V y -ové ose se opakuje trend nízké průměrné chyby, která je jen 0.51° . Poměrně vysoké hodnoty však v tomto případě dosahuje směrodatná odchylka této vertikální osy, která s hodnotou 6.15° znamená doposud nejméně stabilní detekci.



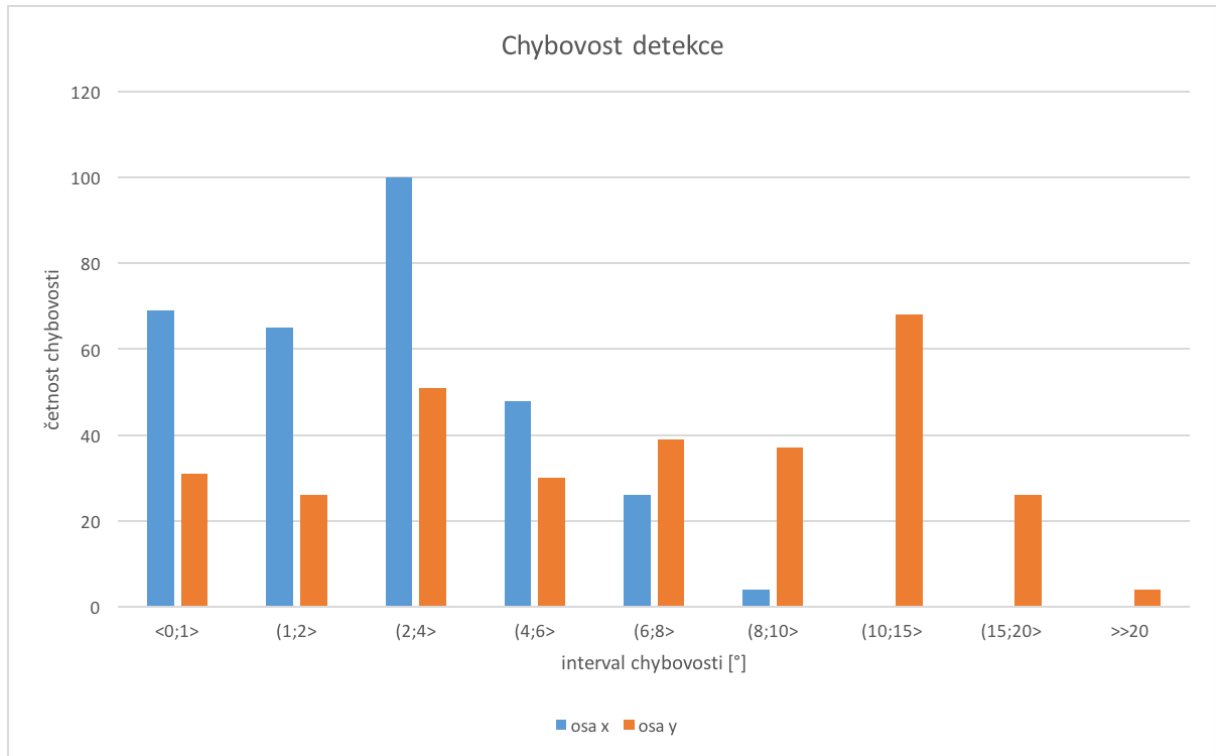
Obrázek 34: Úspěšnost algoritmu na úhlu pohledu $x = -30^\circ$, $y = 0^\circ$.

Posledním z grafů týkajících se přesnosti detekce úhlu pohledu je graf na obrázku 34, popisuje přesnost detekce pro úhel $x = -30^\circ$, $y = 0^\circ$. Vyhodnocení detekce:

- průměrná hodnota úhlu v x -ové ose: -32.09° ,
- průměrná hodnota úhlu v y -ové ose: -7.41° ,
- směrodatná odchylka v x -ové ose: 3.36° ,

- směrodatná odchylka v y -ové ose: 4.54° .

Při detekci v úhlu pohledu $x = -30^\circ$, $y = 0^\circ$ byla na x -ové ose zjištěna průměrná chyba 2.09° , která podobně jako u předchozího měření vypovídá o mírně vyšší průměrné chybě přesnosti. Směrodatná odchylka 3.36° ukazuje na sníženou stabilitu detekce. V y -ové ose je hodnota průměrné chyby dosahující 7.41° doposud nejvyšší průměrnou chybou vertikální osy. Směrodatná odchylka s hodnotou 4.54° zachovává trend nízké stability detekce v y -ové ose.



Obrázek 35: Chybovost detekce úhlu pohledu. Rozdělení do intervalů podle míry chybovosti jednotlivých snímků v $[\circ]$.

Z výsledků provedených detekcí je možné vyvodit, že detekce v x -ové ose dosahuje poměrně dobré průměrné přesnosti s chybovostí do 3° . Hodnota směrodatné odchylky v rovině stejné osy se přitom u úhlu -10° i -20° ground truth hodnoty v drží pod 2.5° , což vypovídá o poměrně dobré stabilitě, u ground truth hodnoty -30° už se však směrodatná odchylka blíží hranici 3.4° .

Výsledky přesnosti průměrné detekce na y -ové ose vycházely ve většině případů lépe, než na x -ové. Výjimkou je ovšem detekce v ground truth hodnotě úhlu 0° , kde detekce vykázala průměrnou chybu měření 7.41° . I přes relativně ucházející průměrnou přesnost detekce se však hodnota směrodatné odchylky úhlu y -ové osy napříč měřeními držela nad hranicí 4° . Tento jev ukázal na slabou stabilitu detekce v rovině této osy.

Na obrázku 35 se nachází graf četnosti chybovosti detekce. Jednotlivé sloupce představují četnost snímků, jejichž úhel pohledu byl detekován s chybou v odpovídajícím intervalu. Rozdělení je zvláště pro jednotlivé hodnoty osy x a osy y . Graf potvrzuje zhoršenou chybovost detekce

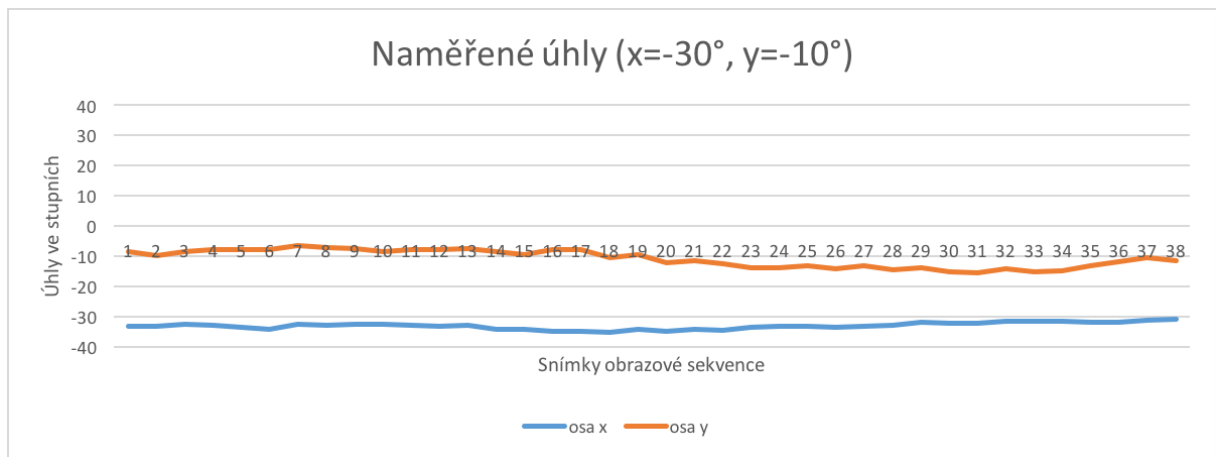
v rovině y -ové osy, kde největší četnost zaznamenala chybovost na hladině $10^\circ - 15^\circ$. Naproti tomu x -ová osa zaznamenává většinu snímků na hladině $0^\circ - 8^\circ$ a podstatná část se nachází dokonce pod hranicí chybovosti 6° .

6.2 Vylepšení přesnosti detektoru

Provedené testy přesnosti detektoru obecně ukázaly zhoršenou stabilitu detekce. Především v y -ové ose dosahovala směrodatná odchylka hodnot za hranicí 4° . Tato podkapitola tedy bude soustředěna na stabilizaci detekce. Jako prostředek k prevenci tohoto jevu bylo zvoleno použití paměti, která by uchovávala historii určitého počtu detekovaných snímků a na jejich základě vyvažovala detekci aktuálního snímku. Toto řešení má samozřejmě význam pouze v případě videozáznamu nebo přenosu z kamery.

Možnou nevýhodou tohoto řešení je pomalejší přesná detekce duhovky v případě prudkého pohybu očí. Detekce aktuálního snímku je totiž ovlivňována také detekcí několika předchozích poloh duhovky. Mírné zpoždění však při volbě správné velikosti této paměti nemusí představovat problém, protože standardní videozáznamy mají většinou více než 20 FPS.

Následující test byl proveden s použitím paměti snímků nastavené na velikost 7. Pro ověření stabilizace detekce byla vybrána stejná sekvence snímků, jako v předchozích testech pro úhel pohledu $x = -30^\circ$, $y = -10^\circ$ (graf na obrázku 33). V předchozím testu bylo během tohoto měření dosaženo směrodatné odchylky 3.11° v x -ové ose a v y -ové dokonce 6.15° . Výsledek nového měření s použitím paměti 7 snímků je prezentován na obrázku 36.



Obrázek 36: Úspěšnost algoritmu na úhlu pohledu $x = -30^\circ$, $y = -10^\circ$. S použitou pamětí 7 snímků.

Při tomto novém měření přesnosti detekce úhlu pohledu pro úhel $x = -30^\circ$, $y = -10^\circ$ byly naměřeny následující hodnoty:

- průměrná hodnota úhlu v x -ové ose: -33.03° ,
- průměrná hodnota úhlu v y -ové ose: -10.77° ,

- směrodatná odchylka v *x-ové* ose: 1.12° ,
- směrodatná odchylka v *y-ové* ose: 2.84° .

Výsledky ukazují, že v porovnání se stejným testem (graf na obrázku 33) došlo pouze k mírné změně průměrné chyby, která v případě obou os představuje rozdíl do 0.4° . Přesnost detekce tedy zůstala téměř nezměněna. Co se stability detekce týče, na *x-ové* ose došlo ke snížení směrodatné odchylky z 3.11° na 1.12° . Snížení dosáhla také směrodatná odchylka v *y-ové* ose, která se snížila z 6.15° na 2.84° . Stabilnější detekce je patrná už pouhým pohledem na graf na obrázku 36.

Z tohoto měření lze vyvodit závěr, že použitím paměti snímků lze efektivně snížit směrodatnou odchylku detekce úhlu pohledu, a tím pádem stabilizovat detekci.

6.3 Rychlost detektoru

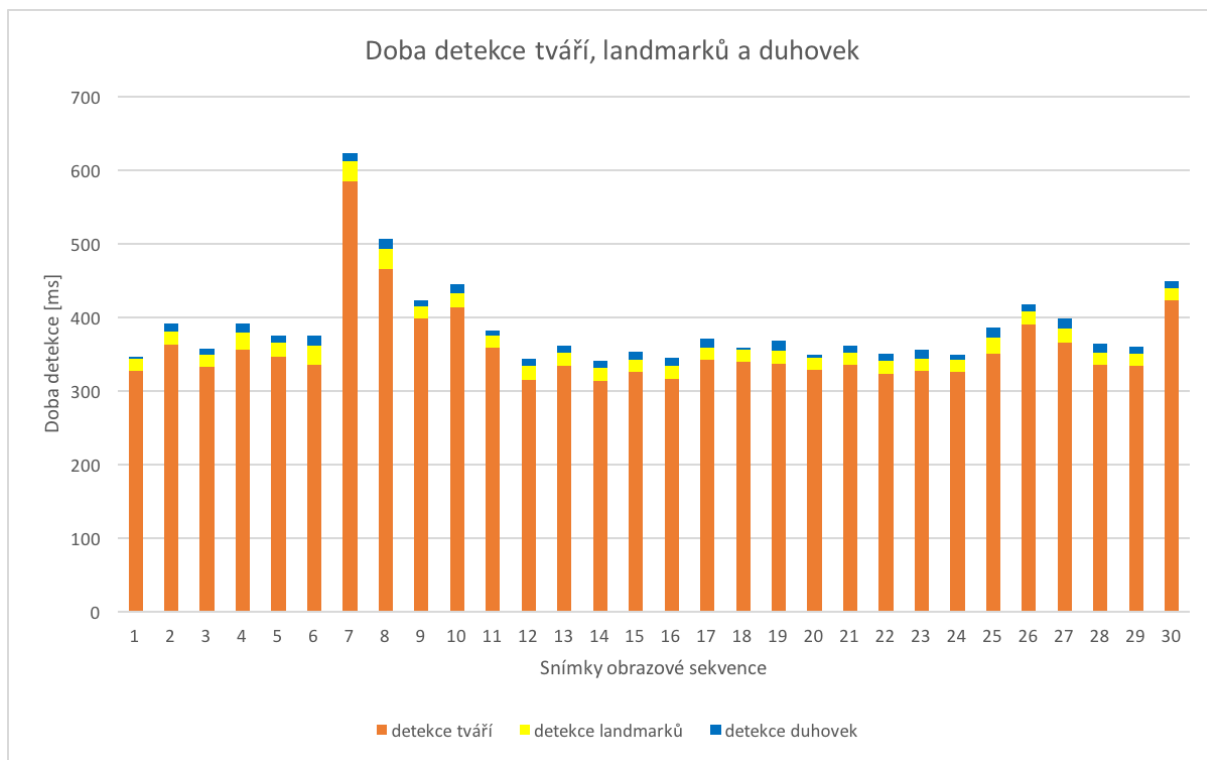
Při úvodním zkoumání a testování použité implementace algoritmu detekce landmarků [28] byla zjištěna poměrně velká časová náročnost. Tento problém byl soustředěn v oblasti detekce samotného obličeje, tedy ještě před hledáním landmarků. Pro detekci obličeje je zde použit kaskádový klasifikátor *CascadeClassifier::detectMultiScale* [23] knihovny OpenCV pro detekci tváří. Tento klasifikátor se svou výchozí konfigurací zapříčinil, že majoritní časový interval celkové detekce byl věnován právě hledání tváří. Přehled rychlostí detekce třiceti testovacích snímků je ukázán na obrázku 37, kde jednotlivé sloupce odpovídají rychlosti celkové detekce úhlu pohledu na jednom snímku. Každý ze sloupců je rozdělen do tří částí - detekce tváří, detekce landmarků a detekce duhovek.

Výchozí parametry detektoru tváří, které byly použity při měření doby detekce ilustrované na obrázku 37, byly nastaveny následovně:

- faktor zvětšení detekčního okna: 1.1,
- minimální počet sousedních kandidátů tváře: 2,
- minimální velikost tváře: 30×30 pixelů.

První parametr udává, že při detekci tváře bude detekční okno po prohledání celého obrazu zvětšeno před další iterací $1.1\times$. Další říká, že aby daná detekovaná tvář byla uznána, musí existovat nejméně dva další sousední kandidáti na tuto tvář. Poslední parametr udává nejmenší možnou velikost tváře. Vyhodnocení detekce s těmito parametry:

- průměrná rychlost detektoru tváří: 357.95 ms,
- průměrná rychlost detektoru landmarků: 18.59 ms,
- průměrná rychlost detektoru duhovek: 9.94 ms,
- celková průměrná rychlost detektoru úhlu pohledu: 386.48 ms.



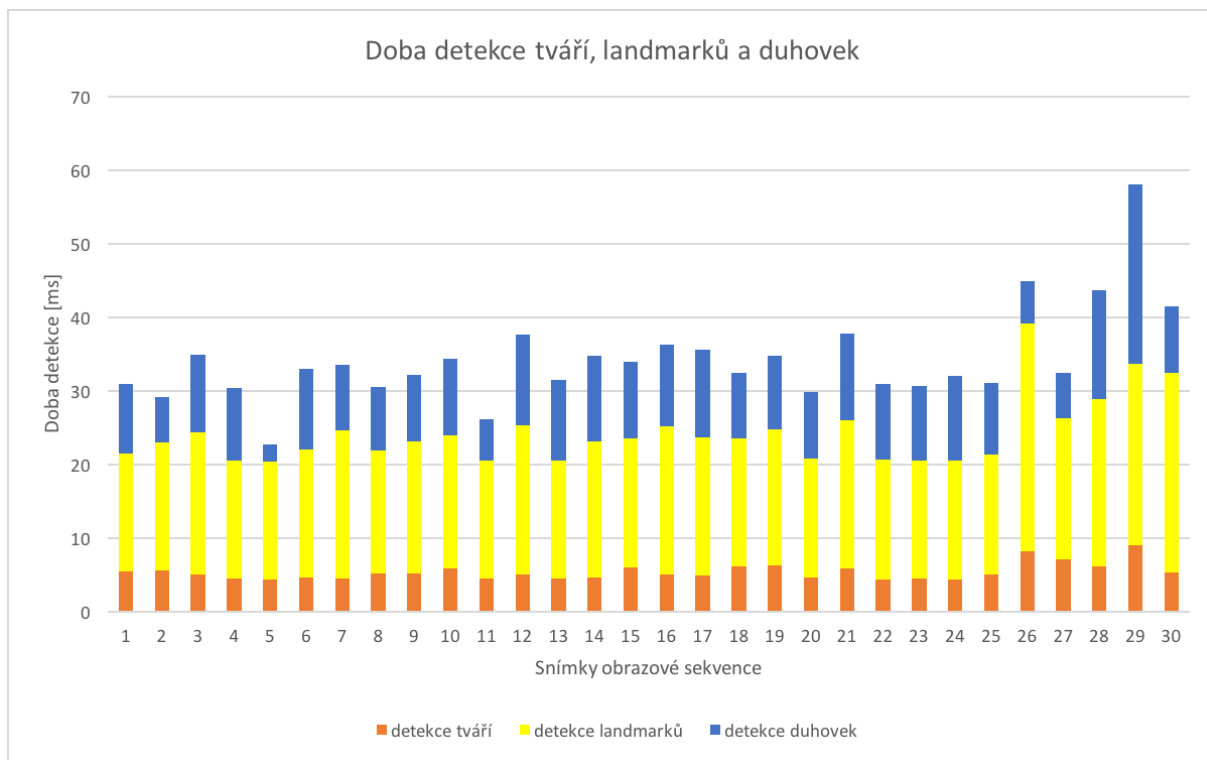
Obrázek 37: Rychlost detekce úhlu pohledu s výchozím nastavením detektoru tváří.

Dosažené rychlosti detekce (obrázek 37) v tomto případě nejsou příliš uspokojivé. Celková průměrná detekce jednoho snímku za 386.48 ms odpovídající přibližně 2.5 FPS neumožňuje dostatečně rychlé zpracování a detekci úhlu pohledu. Zřejmý podíl na tom má právě detekce tváří, která s průměrnou rychlostí 357.95 ms vyžaduje ke svému běhu více než desetinásobek času detekce landmarků a duhovek celkem. Z tohoto důvodu byly parametry detektoru tváří konfigurovány za účelem snížení detekčního času. Nově nastavené parametry detektoru tváří:

- faktor zvětšení detekčního okna: 1.2,
- minimální počet sousedních kandidátů tváře: 2,
- minimální velikost tváře: 300×300 pixelů.

Tato konfigurace zvětšila měřítko, kterým se zvyšuje velikost detekčního okna mezi jednotlivými iteracemi hledání tváří v obraze, z hodnoty 1.1 na 1.2. Minimální počet sousedních kandidátů zůstal zachován na hodnotě 2. Parametr, který doznal nejvýraznější změny, je minimální velikost tváře, která byla zvýšena z rozměrů 30×30 desetinásobně až na 300×300 pixelů. Na obrázku 38 je ukázán přehled doby detekce na stejných třiceti snímcích.

Zatímco rychlosti detekce landmarků a duhovek zůstaly na stejné úrovni, detektor tváří znamenal výrazné snížení doby detekce. Vyhodnocení rychlosti detektoru tváří s konfigurovanými parametry:



Obrázek 38: Rychlost detekce úhlu pohledu s konfigurovaným nastavením detektoru tváří.

- průměrná rychlost detektoru tváří: 5.36 ms,
- průměrná rychlost detektoru landmarků: 18.81 ms,
- průměrná rychlost detektoru duhovek: 10.07 ms,
- celková průměrná rychlost detektoru úhlu pohledu: 34.24 ms.

Nově naměřená rychlost detektoru tváří dosahuje ve srovnání s předchozí konfigurací přibližně 70× rychlejšího výsledku. Rychlosti detektorů landmarků a duhovek se podle očekávání významně nezměnily. Celková průměrná rychlost 34.24 ms odpovídající přibližně 29 FPS je v porovnání s původními 2.5 FPS výrazné zlepšení.

6.4 Problémy detektoru

V předchozích kapitolách byla postupně testována přesnost algoritmu (kapitola 6.1) a jeho rychlost (kapitola 6.3). Tato kapitola bude zaměřena na některé odhalené slabé stránky a nedostatky algoritmu a jejich možná řešení.

6.4.1 Rozlišení vstupního obrazu

Je velmi důležité, aby vstupní obrazový zdroj měl dostatečně velké rozlišení. Během testování byl používán obraz v rozlišení 1280×1080 a výsledná detekce v tomto případě dosahovala dobrých

výsledků. Problém nastává, je-li rozlišení příliš nízké. V tom případě totiž získaná oční oblast obsahuje jen málo obrazových bodů a detekce kružnic je v této oblasti velmi ztížená. Změnou parametrů metody sice lze detekci umožnit, to však s sebou nese její vyšší chybovost.



Obrázek 39: Ukázka detekce oční oblasti při nízkém rozlišení.

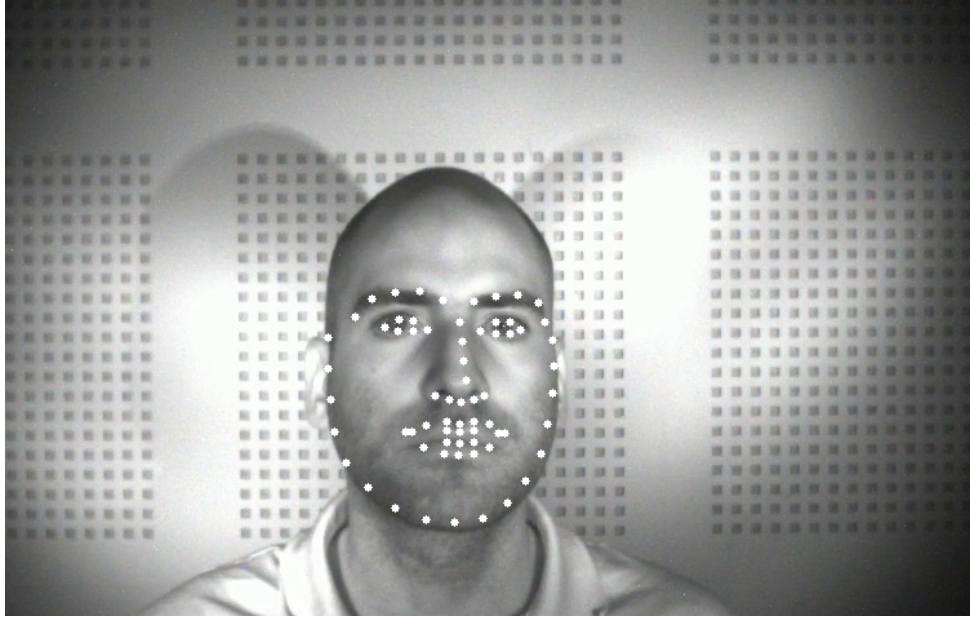
Pro příklad, oblast oka, která byla zpracovávána během testování, dosahovala rozlišení přibližně okolo 70×100 pixelů. V případě rozlišení vstupního obrazu 752×480 (obrázek 39) však oční oblast (obrázek 40) dosahuje pouze rozměrů 17×33 . Kromě nižšího rozlišení má negativní vliv na tyto rozměry také vzdálenější poloha hlavy od kamery.



Obrázek 40: Detekovaná oční oblast při nízkém rozlišení.

6.4.2 Detekce významných bodů obličeje

Použití landmarků pro nalezení oční oblasti má také jisté nevýhody. První z nich se vztahuje k předchozí kapitole 6.4.1, která se zabývala problémem nízkého rozlišení obrazu. Detektor landmarků v případě nižší kvality obrazu nepracuje příliš spolehlivě a má problémy s nalezením přesné polohy významných obličejových bodů. Na ukázce na obrázku 41 je patrné, že oční landmarky byly nalezeny s určitou chybou. Výsledný oční region se nachází na obrázku 40. Je na něm možné vidět, že oblast levého očního koutku zde zcela chybí.



Obrázek 41: Detekované významné body obličeje při nízkém rozlišení.

Dalším nedostatkem detektoru landmarků, při využití pro detekci úhlu pohledu, je stabilita jeho detekce. Tento problém se projevuje především při detekci úhlu pohledu v sekvenčním obrazovém záznamu. Landmarky jsou vypočítávány pro každý jednotlivý snímek zvlášť, což vede k jejich mírně rozdílným polohám napříč videosekvencí. Důsledkem je, že oční oblast má v jednotlivých snímcích mírně odlišnou polohu a rozměry. Vzhledem k tomu, že při určování úhlu pohledu je poloha středu duhovky poměřována s okraji oční oblasti, dochází ke zhoršené stabilitě detekce. Tento jev byl pozorován při testování přesnosti algoritmu detekce úhlu pohledu v kapitole 6.1. Jako poměrně úspěšné řešení, kompenzující tuto nestabilitu, se ukázalo použití paměti snímků prezentované v kapitole 6.2.

7 Závěr

Cílem této práce bylo popsat a seznámit se s problematikou detekce významných bodů obličeje a detekce duhovek a zornic. Na základě provedeného zkoumání měl být vytvořen program, který by umožnil detekci úhlu pohledu osoby v obraze. Jako způsob předzpracování byla v zadání specifikována metoda detekce významných bodů obličeje.

Pro implementaci algoritmu jsem si vybral programovací jazyk C++. Důvodem této volby byla široká podpora, kterou má tento jazyk v oblasti zpracování obrazu. Při vývoji jsem využil knihovnu OpenCV, která obsahuje velké množství metod z oblasti zpracování obrazu a je implementována právě v jazyce C++.

Protože zadání této práce přímo specifikuje, že může být využita hotová implementace detektoru významných bodů obličeje, vybral jsem pro tento případ metodu popsanou v kapitole 2.1. Autoři této metody uvádějí, že při důrazu na přesnost dosahuje rychlost detekce přes 300 FPS a při nastavení s prioritou rychlosti dokonce 3000 FPS. Implementaci této metody jsem stáhl z veřejného zdroje webového verzovacího systému GitHub. Její svobodná licence MIT přímo umožňuje její použití v proprietárním softwaru.

Detektor významných bodů obličeje jsem v programu použil pro nalezení očních oblastí tváře. V rámci této oblasti jsem dále prováděl detekci duhovky a následné stanovení úhlu pohledu pozorované osoby. Vytvořenou metodu jsem podrobil testům přesnosti, pro které byly použity videozáznamy s popsanými pravdivými úhly pohledu. Výsledek těchto testů ukázal jistou nestabilitu detekce, která se projevovala vyšší hodnotou směrodatné odchylky detekovaných úhlů pohledu. S tímto jevem jsem počítal již při vývoji algoritmu, protože bylo patrné, že detekce významných bodů tváře rovněž vykazuje mírnou nestabilitu. Ta byla způsobena tím, že významné body jsou v jednotlivých snímcích videosekvence detekovány pouze s přibližně přesnou polohou, která se tedy napříč videosekvencí mírně liší. Z tohoto důvodu jsem do programu implementoval paměť, která uchovává specifikovaný počet již nalezených středů duhovek a využívá jejich polohy k vyvážení stability detekce. Výsledky přesnosti detekce s použitím této paměti ukázaly, že se stabilitu podařilo vylepšit.

Další testování, kterému jsem metodu podrobil, se týkalo její rychlosti. Změřil jsem tedy dobu detekce jednotlivých částí metody - detekce obličeje (je součástí detekce významných bodů obličeje), detekce významných bodů a detekce duhovek. Z testů vyšlo, že majoritní časový úsek strávil detektor detekcí tváře - přes 350 ms oproti necelým 19 ms detekce významných bodů a zhruba 10 ms detekce duhovek. Z tohoto důvodu jsem přistoupil ke konfiguraci parametrů metody detekce tváře, což umožnilo snížení doby jejího běhu na pouhých 5.36 ms. Celková doba zpracování úhlu pohledu na jednom snímku potom vyšla na 34.24 ms, což odpovídá přibližně 29 FPS. S touto rychlostí je možno detekovat úhel pohledu real-time.

Budoucí vývoj detektoru by se mohl zabývat detekcí natočení hlavy. S touto vlastností by bylo možno detekovat přesnější úhel pohledu v případech, kdy hlava není orientována přímo do kamery. Pro řešení tohoto rozšíření by mohla být využita poloha významných bodů obličeje.

Druhým možným vylepšením by mohlo být nahrazení současného hranového detektoru. Algoritmus nyní využívá Cannyho detektor hran, který detekuje všechny hrany nacházející se v daném obraze. Zajímavou možností by mohlo být vytvoření vlastního hranového detektoru, který by se specializoval přímo na detekci duhovek. Inspirací může být například metoda tzv. *U-křivky* popsaná v kapitole 3.2.1.2. Příklad srovnání detekce hran Cannyho detektorem a detektorem *U-křivky* se nachází na obrázku 16.

Literatura

- [1] PAUL VIOLA, MICHAEL JONES *Rapid Object Detection using a Boosted Cascade of Simple Features*. Publikováno na *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. CVPR 2001
- [2] SHAOQING REN, XUDONG CAO, YICHEN WEI, JIAN SUN *Face Alignment at 3000 FPS via Regressing Local Binary Features*. Publikováno na *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE 2014
- [3] FEDERICO M. SUKNO, JOHN L. WADDINGTON, PAUL F. WHELAN *3D Facial Landmark Localization using Combinatorial Search and Shape Regression*. Publikováno na *IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume:35 , Issue: 12)*. IEEE 2013
- [4] PETER N. BELHUMEUR, DAVID W. JACOBS, DAVID J. KRIEGMAN, NEERAJ KUMAR *Localizing Parts of Faces Using a Consensus of Exemplars*. Publikováno na *ECCV 2012*.
- [5] G. GALFANO, M. DALMASO, D. MARZOLI, G. PAVAN, C. CORICELLI, L. CASTELLI *Eye gaze cannot be ignored (but neither can arrows)*. Publikováno v *The Quarterly Journal of Experimental Psychology, Apr. 2012*, strany 1895-1910.
- [6] N. AHMIDI, GD. HAGER, L. ISHII, G. FICHTINGER, GL. GALLIA, M. ISHII *Surgical task and skill classification from eye tracking and tool motion in minimally invasive surgery*. Publikováno v *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, strany 295 - 302, ISBN 978-3-642-15710-3.
- [7] P. TANGADE, S. MUSALE, G. PASALKAR *A Review Paper on Mouse Pointer Movement Using Eye Tracking System and Voice Recognition*. Publikováno v *International Journal of Emerging Engineering Research and Technology, Volume 2, Issue 8, November 2014.*, strany 135 - 138, ISSN 2349-4395.
- [8] A. CHO, J. CHUNG, L. YU, J. LEE, S. LEIGH *The eyeCan project*. Dostupné z: <http://www.eyecanproject.org/p/english.html>.
- [9] POONAM S. GAJWANI, SHARDA A. CHHABRIA *Eye Motion Tracking For Wheelchair Controll*. Publikováno v *International Journal of Information Technology and Knowledge Management July-December 2010, Volume 2, No. 2*, strany 185 - 187.
- [10] K. ARAI, R. MARDIYANTO *Eyes Based Eletric Wheel Chair Control System*. Publikováno v *International Journal of Advanced Computer Science and Applications, Vol. 2, No. 12, 2011*.

- [11] W. M. K WAN MOHD KHAİROSFAİZAL AND A. J. NOR AİN *Eyes detection in facial images using Circular Hough Transform*. Publikováno v *Signal Processing & Its Applications, 2009. CSPA 2009. 5th International Colloquium on*, strany 238 - 242.
- [12] N. ALİOUA, A. AMİNE, M. RZİZA, D. ABOUTAJDİNE *Eye state analysis using iris detection based on Circular Hough Transform*. Publikováno na *Multimedia Computing and Systems (ICMCS), 2011 International Conference on*, strany 1 - 5.
- [13] P. VIOLA, M. JONES *Robust Real-Time Face Detection*. Publikováno na *International Journal of Computer Vision 57(2), 2004*, strany 137 - 154.
- [14] J. PŘİNOSİL, M. KROLİKOWSKI *Využití detektoru Viola-Jones pro lokalizaci obličejů a očí v barevných obrazech*. Publikováno na *Elektrorevue, 2008*.
- [15] L. ŚWIRSKI, A. BULLİNG, N. DODGSON *Robust real-time pupil tracking in highly off-axis images*. Publikováno v *Proceedings of ETRA, 2012*.
- [16] S. V. SHEELA, P. ABHİNAND *Iris detection for gaze tracking using video frames*. Publikováno v *Advance Computing Conference (IACC), 2015 IEEE International, 2015*, strany 629 - 633.
- [17] OPENCV - OPEN SOURCE COMPUTER VISION. [online] [cit. 26.4.2016] Dostupné z: <http://opencv.org/about.html>.
- [18] OPENCV 2.4.12.0 DOCUMENTATION *Canny Edge Detector*. [online] [cit. 26.4.2016] Dostupné z: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html.
- [19] OPENCV 2.4.12.0 DOCUMENTATION *Histogram Equalization*. [online] [cit. 26.4.2016] Dostupné z: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html.
- [20] OPENCV 2.4.12.0 DOCUMENTATION *Eroding and Dilating*. [online] [cit. 26.4.2016] Dostupné z: http://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html.
- [21] OPENCV 2.4.12.0 DOCUMENTATION *Smoothing Images*. [online] [cit. 26.4.2016] Dostupné z: http://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html.
- [22] OPENCV 2.4.12.0 DOCUMENTATION *Basic Thresholding Operations*. [online] [cit. 26.4.2016] Dostupné z: <http://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html>.

- [23] OPENCV API REFERENCE *Cascade Classification*. [online] [cit. 26.4.2016] Dostupné z: http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html.
- [24] P. KALRA *Canny Edge Detection, 2009*. [online] [cit. 26.4.2016] Dostupné z: <http://www.cse.iitd.ernet.in/~pkalra/csl1783/canny.pdf>.
- [25] P. LU, H. XIANG, Y. DAI, W. ZHANG *Iris Detection and Marking Based on Haar Features*. Publikováno na *Audio Language and Image Processing (ICALIP), 2010 International Conference on, 2010*, strany 518 - 522.
- [26] N. ALIOUA, A. AMINE, M. RZIZA, D. ABOUTAJDINE *Eye state analysis using iris detection based on Circular Hough Transform*. Publikováno na *Multimedia Computing and Systems (ICMCS), 2011 International Conference on*, strany 1 - 5.
- [27] R. MAINI, H. AGGARWAL *Study and Comparison of Various Image Edge Detection Techniques*. [online] [cit. 26.4.2016] Dostupné z: <http://www.math.tau.ac.il/~turkel/notes/Maini.pdf>.
- [28] LEQUAN YU *A C++ implementation of face alignment in 3000fps*. [online] [cit. 26.4.2016] Dostupné z: <https://github.com/yulequan/face-alignment-in-3000fps>
- [29] M. LEEFLANG *Human gaze & eyetracking*. [online] [cit. 26.4.2016] Dostupné z: <http://www.frankwatching.com/archive/2013/05/01/human-gaze-en-eyetracking-zorgt-juiste-kijkrichting-voor-hogere-conversie/>.
- [30] LAKSHMI SWARUPA *Circular Hough Transform*. [online] [cit. 26.4.2016] Dostupné z <http://imageprocessingtechnique.blogspot.cz/2014/06/circular-hough-transform.html>.

A Příloha na CD

A.1 Diplomová práce

Diplomová práce ve formátu PDF se nachází v kořenovém adresáři CD s názvem `JAM0010.pdf`.

A.2 Implementace detektoru úhlu pohledu

Implementace vytvořeného detektoru úhlu pohledu se nachází v adresáři `GazeDetection`.

A.3 Testovací videozáznam

Testovací videozáznam, na kterém byly prováděny testy v kapitole 6, se nachází v adresáři `GazeDetection/test_video.mp4`.

A.4 Ukázka výstupu

Videozáznam prezentující ukázkou výstupu detektoru úhlu pohledu se nachází v adresáři `example/output.mp4`.

B Návod k použití detektoru úhlu pohledu

B.1 Knihovna OpenCV

Knihovna OpenCV verze 2.4.x je nutnou prerekvizitou pro spuštění vytvořeného softwaru. Její stažení je možné na adrese <http://opencv.org/downloads.html>.

B.2 CMake

Instalace programu CMake, který je možné stáhnout na adrese <https://cmake.org/download/>. Minimální požadovaná verze je 2.8.11.

B.3 Konfigurace CMakeLists.txt

Soubor `CMakeLists.txt` se nachází v adresáři `GazeDetection/CMakeLists.txt`. V tomto souboru v sekci `CONFIGURATION` je nutné nastavit správné cesty ke knihovně OpenCV.

B.4 Kompilace programu

Pro kompilaci programu se v konzoli přepnout do adresáře `GazeDetection` a provést následující dva příkazy:

```
$ cmake .  
$ make
```

Po doběhnutí kompilace by se měl v aktuálním adresáři `GazeDetection` objevit spustitelný soubor `gaze`.

B.5 Spuštění programu

Pro spuštění programu se zavolá spustitelný soubor a jeho parametry. První parametr je povinný a určuje adresu vstupního obrázku nebo videozáznamu. Pro nastavení vstupu z kamery je místo adresy k souboru zadán řetězec `camera`. Druhý parametr je nepovinný a udává velikost paměti předchozích detekcí (kapitola 6.2). Příklad spuštění detektoru se vstupním souborem `test_video.mp4` a velikostí paměti 10:

```
$ gaze test_video.mp4 10
```
