

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Automatická detekce pupily v obraze

Bc. Jonáš Erlebach
Kybernetika a robotika

Květen 2016
Vedoucí práce: Ing. Jaromír Doležal, Ph.D.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Jonáš E r l e b a c h
Studijní program: Kybernetika a robotika
Obor: Robotika
Název tématu: Automatická detekce pupily v obraze

Pokyny pro vypracování:

1. Seznamte se s metodami používanými pro automatickou detekci pupily v obraze.
2. Demonstrujte princip metod pro detekci pupily a srovnajte jejich výhody a nevýhody.
3. Implementujte vybranou nejspolehlivější metodu v prostředí Matlab. Při implementaci dbejte na snadnou převoditelnost metody do jazyka C.
4. Srovnajte úspěšnost metody na databázi záznamů oka oproti referenční metodě EyeGaze SDK.

Seznam odborné literatury:

- [1] Hlaváč V., Sedláček M.: Zpracování signálů a obrazů, skriptum, Vydavatelství ČVUT, 2009
- [2] Fisher, R. B., et al.: Dictionary of Computer Vision and Image Processing, New York: Wiley, 2005
- [3] Šonka M., Hlaváč V., Boyle R.: Image Processing, Analysis and Machine Vision, Thomson, 3rd edition, 2007
- [4] Duschoswky T.: Eye tracking Methodology theory and practice, Springer, 2007

Vedoucí diplomové práce: Ing. Jaromír Doležal, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017

L.S.

prof. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 22. 12. 2015

Poděkování / Prohlášení

Chtěl bych poděkovat svému vedoucímu diplomové práce panu Ing. Jaromíru Doležalovi, Ph.D. za odborné vedení a cenné připomínky při vypracování této práce. Děkuji též společnosti Medicton Group, s.r.o. za poskytnutí dat ke zpracování. Nemohu zapomenout poděkovat rodinným příslušníkům za podporu při studiu i psaní této práce.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 27. 5. 2016

.....

Abstrakt / Abstract

Diplomová práce popisuje detekci zornice v obraze pro aplikaci při detekci očních pohybů. Jedná se o nalezení zornice v obraze z kamery, která sleduje oko. Práce se zabývá metodami zpracování obrazu užívanými k detekci příznaků v dané problematice. Dále porovnává tři zvolené metody detekce. První metodou je detekce na základě hledání hran v obraze. Druhou metodou je využití kaskádního detektoru (Viola-Jones) k nalezení zornice. Posledním přístupem je potom využití segmentace (Mean shift). Práce se též zabývá finální detekcí elipsy, v oblasti s předpokládanou polohou zornice. Závěr práce obsahuje porovnání úspěšnosti těchto metod, ve srovnání s daty získanými pomocí EyeGaze SDK.

Klíčová slova: detekce zornice; hrnová detekce, kaskádní detektor, segmentace; nalezení kružnice a elipsy

Diploma thesis describes detection of the pupil position in eye images, used in eye tracking. The goal of the thesis is to find pupil in camera recording of the eye. Thesis analyzes some image processing methods used for feature detection in described issue and compares three chosen detection methods. The first method is based on edge detection. The second method uses cascade object detector (Viola-Jones) for pupil detection. And the last method is based on Mean shift segmentation. The thesis is moreover dealing with final ellipse detection in region of eye pupil expected location. Success rates of all method are compared to reference EyeGaze SDK method.

Keywords: pupil detection; edge detection, cascade detector, segmentation; circle and ellipse fit

Title translation: Automatic pupil detection in image

Obsah /

1 Úvod	1
2 Možná řešení problematiky	2
2.1 Princip sledování očních pohybů	2
2.2 Přístupy detekce zornice	3
3 Teorie použitých metod	6
3.1 Detekce hran	6
3.2 Houghova transformace	6
3.3 Segmentace	7
3.3.1 K-means	7
3.3.2 Mean-shift	7
3.4 Učící algoritmy	8
3.4.1 Kaskádní detektor	8
3.4.2 Neuronová síť	11
3.5 Fitování kružnice, elipsy	11
3.5.1 Kritérium podobnosti kružnici, podobnost	11
3.5.2 Kritérium podobnosti kružnici, poměr plochy a obvodu	12
3.5.3 Fitování elipsy	12
4 Praktická část	13
4.1 Použitá data	13
4.2 Analýza dat	14
4.3 Metodika hodnocení úspěšnosti	17
4.4 Určení věrohodnosti řešení	18
5 Metoda na principu hranové detekce	22
6 Metoda učení na základě vzoru v obraze	26
7 Metoda založená na segmentaci a následné klasifikaci	32
7.1 Mean-shift segmentace	32
7.2 Další zpracování	32
7.2.1 Míra kruhovitosti	33
7.2.2 Volba segmentu příslušejícího zornici	34
8 Závěr	39
Literatura	42
A Softwarová příloha	45
B Porovnání parametrů všech metod	46
B.1 Středy	46
B.2 Doba zpracování	47
B.3 Kvalita detekce plochy	48
B.4 Tabulka zkoumaných parametrů	48

Tabulky / Obrázky

5.1. Výsledky hranové detekce.....	25
6.1. Porovnání metod kaskádního detektoru	31
7.1. Porovnání metod segmentace..	38
8.1. Porovnání metod detekce	40
B.1. Porovnání všech metod detekce	48
2.1. Měřicí brýle	2
2.2. Distanční tracker	2
2.3. Odrazivost zornice	3
3.1. Haar příznaky	9
3.2. Integrální snímek.....	9
3.3. Schéma detektoru	10
3.4. HOG příznaky.....	11
4.1. Brýle pro měření	13
4.2. Ukázka záznamu	14
4.3. Histogram vodorovné pozice...	15
4.4. Histogram svislé pozice	15
4.5. Histogram výšky zornice.....	16
4.6. Histogram šířky zornice.....	16
4.7. Histogram excentricity zornice	17
4.8. Histogramy vodorovné pozice segmentů	19
4.9. Histogramy svislé pozice segmentů	19
4.10. Histogramy poloměru segmentů	19
4.11. Histogramy kritéria kruhovitosti	20
4.12. Histogramy poměru jasu.....	20
5.1. Vstupní obrázek hranové detekce	23
5.2. Snímek nalezených hran	23
5.3. Vstupní obrázek hranové detekce	23
5.4. Snímek nalezených hran	23
5.5. Nalezené hrany pozitivní.....	23
5.6. Nalezené hrany negativní	23
5.7. Zvýraznění hran	24
5.8. Detekce Houghovou transformací, pozitivní příklad.	25
5.9. Detekce Houghovou transformací, negativní příklad.....	25
6.1. Detekce zornice	26
6.2. Detekce zornice.....	26
6.3. Porovnání kaskádního detektoru s referencí, pozitivní	27
6.4. Porovnání kaskádního detektoru s referencí, negativní.....	27
6.5. Porovnání metod	28
6.6. Histogram vzdáleností středů od reference LBP.....	29

6.7.	Histogram doby zpracování LBP	29
6.8.	Histogram vzdáleností středů od reference LBP + neuronová síť	30
6.9.	Úspěšnost v závislost na ploše, kaskádní detektor.....	30
6.10.	Špatně vepsaná elipsa	31
7.1.	Vstup segmentace	32
7.2.	Výsledek segmentace	32
7.3.	Segmentace s minimální velikostí ploch	33
7.4.	Segmentace s hodnotami kritéria kruhovitosti.....	33
7.5.	Segmentace s hodnotami kritéria kruhovitosti.....	34
7.6.	Správná detekce	35
7.7.	Negativní detekce	35
7.8.	Detekovaná zornice metodou segmentace	35
7.9.	Detekovaná duhovka metodou segmentace.....	36
7.10.	Histogram přesnosti středů segmentace	36
7.11.	Histogram doby zpracování segmentace	37
7.12.	Úspěšnost segmentačních metod v závislost na ploše	37
8.1.	Úspěšnost v závislost na ploše .	40
B.1.	Rozdíl středů Haar	46
B.2.	Rozdíl středů HOG	46
B.3.	Rozdíl středů LBP	46
B.4.	Rozdíl středů LBP + neuronová síť	46
B.5.	Rozdíl středů segmentace s detekcí elipsy	47
B.6.	Rozdíl středů segmentace s detekcí kružnice	47
B.7.	Doba zpracování Haar	47
B.8.	Doba zpracování HOG	47
B.9.	Doba zpracování LBP.....	47
B.10.	Doba zpracování LBP + neuronová síť	47
B.11.	Doba zpracování segmentace s detekcí elipsy	48

B.12.	Doba zpracování segmentace s detekcí kružnice	48
B.13.	Úspěšnost všech metod v zá- vislost na ploše	48

Kapitola 1

Úvod

Cílem práce je vytvořit automatickou detekci pupily v obraze. Tato detekce má sloužit k využití v brýlích, které se vyvíjejí pro sledování očních pohybů. Úkolem je tedy ve snímku oka, pořízeném pomocí kamery sledující oko, nalézt zornici. Další využití je plánováno v jiných verzích trackeru (označení zařízení pro záznam v souvislosti se sledováním očních pohybů), k detekci pupily ve výřezu snímku s celým obličejem.

Metoda sledování očních pohybů je dnes často řešenou problematikou v širokém spektru aplikací. Mezi obory využití patří například zdravotnictví (odhalení duševních či vývojových poruch), marketing (sledování účinnosti reklamy), bezpečnostní aplikace (k detekci duhovky jako identifikace, nejprve se detekuje zornice, kolem které se hledá duhovka) a v neposlední řadě též herní průmysl (další možnost ovládání počítačů nebo mobilních zařízení).

Pro účely vývoje a ověření funkčnosti využijeme výstupů systému I4Tracking® od firmy Medicton Group, který využívá k detekci zornice softwarové řešení od Firmy Eyedea Recognition, EyeGaze SDK. Při testování se využívá stejných vstupních obrázků, získaných z kamery umístěné na brýlích sledované osoby, jako využívá dosavadní řešení. Výstupy našeho algoritmu následně porovnáme s parametry detekované zornice získané pomocí tohoto komerčního produktu.

Úkolem je porovnat několik přístupů k detekci zornice a snaha nalézt nejlepší a nejoptimálnější řešení vzhledem k době zpracování a přesnosti detekce zornice. Tyto parametry jsou velmi důležité vzhledem k vysoké vzorkovací frekvenci kamery a přesnosti ovlivňující výsledek detekovaného pohledu.

Práce je rozčleněna do následujících kapitol. **Možná řešení problematiky** 2, která shrnuje metody využívané k detekci zornice. Další je **Teorie použitých metod** 3, kde jsou rozebrány jednotlivé metody zpracování obrazu, které jsou dále využity při realizaci vlastního zpracování dat. Pokračujeme konkrétním popisem dostupných dat v kapitole **Praktická část** 4. Následují tři kapitoly popisující testované postupy řešení problematiky (**Metoda na principu hranové detekce** 5, **Metoda učení na základě vzoru v obraze** 6, **Metoda založená na segmentaci a následné klasifikaci** 7). Práce je zakončena porovnáním testovaných metod, závěrečným výběrem nejlepší implementované metody a zhodnocením v kapitole **Závěr** 8.

Kapitola 2

Možná řešení problematiky

Tato kapitola popisuje možnosti náhledu na řešení problému. Zabývá se detaily pořízení dat ke zpracování. Dále se zaměřuje na některé přístupy detekce pupily popsané v dostupných zdrojích.

2.1 Princip sledování očních pohybů

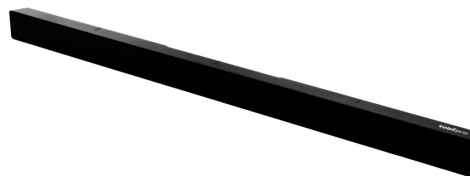
Pro sledování očních pohybů je možné využít několika metod [7]. Jednou z metod je elektro-okulografie, kdy se měří činnost očního svalstva. Další možností je využití speciálně upravené kontaktní čočky s využitím jejího pohybu v elektromagnetickém poli. Nejrozšířenější je však využití foto- nebo video-okulografie kdy se snímá oko pomocí kamery.

Při sledování očních pohybů chceme obvykle získat polohu bodu na který se člověk dívá. K tomuto účelu je třeba zjistit střed zornice a transformaci mezi body zájmu a polohou pupily. Další problematika, kterou se v této oblasti můžeme zabývat, je takzvaná pupilometrie [20]. Ta se zabývá změnou velikosti zornice. S využitím psychologických postupů se pozoruje vnímání určitých podmětů a emocí pomocí změny velikosti pupily. Pro tyto účely je tedy třeba sledovat i změnu plochy zornice.

Video-okulografie využívá v zásadě dva přístupy pořizování dat. Vždy je třeba nalézt v záznamu kamery zornici a na základě její polohy či velikosti probíhá další zpracování. V prvním případě se využívá zařízení na principu brýlí umístěných na hlavě sledované osoby, například na obrázku 2.1, které obsahují kameru zaznamenávající oblast oka. Měření probíhá buď kamerou umístěnou proti směru pohledu, mimo zorné pole oka, nebo s využitím speciálního polopropustného skla brýlí (s odrazem především IR záření) a umístění kamery ve směru pohledu oka namířené na sklo brýlí. Druhou možností je řešení znázorněné na obrázku 2.2, kde je kamera umístěna v oblasti, kterou pozorovaná osoba sleduje. V tomto případě je třeba provést nejprve detekci oblasti oka, druhá část detekce zornice, pro určení kam se subjekt dívá, je však stejná.



Obrázek 2.1. Měřicí brýle Tobii Pro Glasses 2 [24].

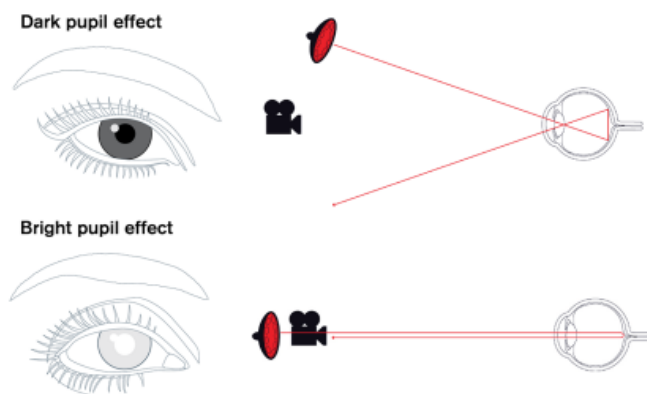


Obrázek 2.2. Distanční tracker Tobii Pro X3-120 [24].

V našem případě je cílem, zabývat se konkrétně detekcí zornice v oblasti oka, neboť testovací záznamy jsou pořízeny pomocí brýlí, které zaznamenávají přímo oblast oka. První náhled na problematiku detekce pupily v obraze přináší myšlenku detekce černého

kruhu na snímku, který je uvnitř další kružnice. Nabízí se tedy metody hledání hran a detekce příslušného vzoru. Další metodou by mohlo být využití učících se algoritmů.

Při detekci se využívá charakteristické vlastnosti lidského oka [24]. Oko se osvětluje infračerveným světlem, které je pro člověka neviditelné, v kameře jsou však zřejmé charakteristické vlastnosti odrazivosti oka. V případě, že dopad světla je přibližně v optické ose oka, zornice je výrazně světlejší než okolí. V případě, kdy však infračervené světlo dopadá ze směru mimo optickou osu, dochází naopak ke ztmavnutí zornice. Tento jev je demonstrován na schématu 2.3.



Obrázek 2.3. Odrazivost infračerveného přisvícení pupily [24].

2.2 Přístupy detekce zornice

K detekci pupily se někdy využívá odrazivosti obrazce sestaveného z infračervených diod, které při kolmém dopadu světla na zornici vytváří světlá místa. Konkrétně v tomto případě popsaném v práci [3], se jedná o kružnici tvořenou několika úzce směřovanými zdroji. Detekce odraženého zdroje určí přibližnou polohu a následně se, při této znalosti, hledá hrana zornice na dalším rozhraní. K nalezení hran se využívá struktury tenzorové analýzy a následného prahování, je zde tedy určitá závislost algoritmu na okolním osvětlení.

Autoři článku [2] využívají tmavé odrazivosti zornice, kdy je k detekci využit Cannyho hranový detektor. Následně s využitím dilatace se převede snímek na binární obraz s vyfiltrovanými malými plochami a spojitými většími objekty, mezi kterými by se měla nacházet zornice. V binárním obraze se poté pro každý bod příslušející sledovaným plochám spočítá vzdálenost od okraje této plochy a tak vznikne ve středu zornice bod s největší hodnotou v celém snímku. Hodnota v této pozici zároveň udává poloměr zornice.

Jiným přístupem se zabývá článek [19], kde je popsána detekce zornice na základě histogramu celého snímku. Předpokládá se, že zornice bude mít černou barvu a proto se v histogramu označí oblast na tmavém konci, kterou budeme považovat za zornici (očekáváme přibližně stejné obsažení všech tmavých složek, výraznější změna již není zornice). Následně prohledáváme snímek v obou směrech a na základě hodnot z histogramu určíme začátek a konec zornice.

Přístup využití hledání výrazné změny jasu v obou směrech snímku využívají i autoři článku [21]. Zde je však zmíněna i často využívaná metoda využití Houghovy transformace pro hledání kružnic. Konkrétně se provádí hledání kružnic na šedotónovém snímku ovšem pouze na body, které byly předem určeny hranovou detekcí. Jako zornice je následně označena v prostoru Houghovy transformace výrazná špička mezi nízkými hodnotami jasu pro danou kružnici.

Houghova transformace se využívá k detekci v mnoha modifikacích, další implementaci popisuje článek [18]. Opět na snímek, který je předzpracován nalezením hran, aplikujeme Houghovu transformaci, která se omezuje na určité rozmezí poloměrů. S využitím této transformace pro každý hranový bod, na základě hranového gradientu, přiřadí svoji váhu k určitému středu, podle právě uvažovaného poloměru.

Další zpracování jsou založena na segmentaci na charakteristické oblasti. Následně se mezi segmenty hledá zornice. Tento postup využívají autoři článku [12]. Vstupem je v jejich případě snímek s nalezeným obličejem. Tento snímek se převede do odstínů šedi, neboť barvy nemají při detekci významný vliv a navíc se sníží výpočetní náročnost na další operace. Pomocí k-means se provede segmentace na zvolený počet oblastí s podobnou barvou. Předpokládá se, že mezi segmenty budou výrazně znázorněny zornice. Následně se provede prahování na binární obrázek s prahem zvoleným experimentálně (závislé na okolním prostředí). Pro testování je použit průměr jasu celého obrázku, který je však úspěšný pouze někdy. Získaný snímek vymaskujeme kruhovým jádrem, a pokud nalezneme shluk o určité velikosti se stejnými hodnotami, označíme ho za zornici.

Jiný přístup, který též využívá segmentaci popisuje práce [11]. Vstupní obrázek je převeden na binární snímek s předpokladem, že zornice bude nejtmaší část snímku. Binární snímek se rozdělí na jednotlivé oddělené tmavé segmenty. U těchto segmentů se provede nalezení elipsy. Parametry elipsy se detekují sledováním průběhu hrany segmentu v souvislosti se středem (těžištěm) segmentu. Jako kritérium pro nejlepší elipsu se využije rozdíl skutečné plochy segmentu a vypočítané plochy podle získaných parametrů. Za zornici je potom zvolena elipsa s nejnižší hodnotou tohoto kritéria.

Následující přístup využívá učící algoritmus jehož užití je popsáno v práci [30]. Pro redukcí oblastí zájmu se využije Viola-Jones detektor, který využívá příznaky Haarovy báze. Na základě těchto příznaků jsou tedy v obraze označeny oblasti potenciálně obsahující zornici. Na nalezenou oblast se dále aplikuje k-means na dva shluky, který by měl oddělit zornici od okolí. Tak je označena přibližná poloha. Chybí tedy již jen nalézt parametry výsledné elipsy příslušející zornici. Pro určení těchto parametrů se ve zkoumané oblasti detekují hrany pomocí Cannyho hranového detektoru a následně metodou RANSAC na výběr z hranových bodů, aplikací minimalizační funkce na elipsu, provede nalezení parametrů zornice.

Viola-Jones detektor je využit i v jiném případě. V článku [4] jej však autoři využívají k detekci obličeje. Ten se potom rozdělí vertikálně i horizontálně na polovinu a pro detekci se vybere vždy čtvrtina snímku, která obsahuje oko. Pro detekci oblasti oka, se využije takzvaná kumulativní distribuční funkce (CDF), která vznikne integrací histogramu, tedy v diskrétním případě jde o sumu. V daném případě bylo zjištěno, že pokud je hodnota CDF menší než určitý práh, jedná se o oblast zornice. Pro nalezení středu zornice je třeba odstranit malé detekované regiony mimo zornici, k tomu se využije eroze. Ve zbylé oblasti potom nalezneme střed, který je středem zornice.

Nakonec zmíníme přístup využití detekce hran, který využívají autoři článku [27]. K detekci využívají hranový detektor označovaný jako EDPF [26], který ve snímku detekuje hrany, které jsou rozděleny na hranové segmenty. Pro každý nalezený segment se spočítá míra kruhovitosti segmentu na základě směru gradientu v hranových bodech.

Následuje fitování elips do hranových elementů. Kandidáti na pupilu se získají kombinacemi získaných elips. Výběr správného výsledku je nakonec proveden na základě kombinace parametrů: chyby fitování elipsy, excentricity a poměru délky segmentů a obvodu výsledné elipsy.

Kapitola 3

Teorie použitých metod

Tato kapitola popisuje teoretický základ některých metod využívaných při detekcích objektů zájmu v obraze. Tyto metody jsou následně s různými modifikacemi implementace využity v další části práce při konkrétní detekci zornice v obraze.

3.1 Detekce hran

Hranové detektory [10] slouží k lokálnímu předzpracování snímků. Hraný jsou místa s největšími změnami jasu. Obecně tedy ve snímku daném spojitou funkcí hledáme derivace dané funkce, abychom získali gradient (3.1) a tím hodnotu potenciálního hranového bodu. (Výpočet gradientu obsahuje f funkci jasu snímku, x, y parametry funkce f pozice v rovině.) Takto však ze snímku opět dostaneme dvojrozměrný snímek. Je tedy třeba zvolit práh, od jaké hodnoty se již jedná o hranu. Případně je možné další zpracování, pokud například hledáme pouze takové objekty, aby byly všechny jejich hrany uzavřenými křivkami. Dalším zjednodušením je, že náš snímek není analogovou funkcí, ale diskretním obrázkem s omezeným počtem pixelů.

$$|\text{grad } f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (3.1)$$

V určitých případech je též možné hledat hrany pouze určitého profilu podle toho jaký je průběh přechodu kolem maximální změny.

Pro výpočet hran v diskretním světě využíváme takzvaných hranových masek, které využijeme ke dvojrozměrné konvoluci se vstupním snímkem. Tímto postupem dostaneme již zmíněný obrázek potenciálních hran snímku.

3.2 Houghova transformace

Houghova transformace [22] provede zobrazení jednoduchých objektů do n -rozměrného prostoru. Objekty (obvykle přímky, kružnice) popíšeme pomocí co nejmenšího počtu (n) parametrů. Hledané objekty musí být v daném obrázku parametry jednoznačně definovatelné.

Vytvoříme tedy prostor všech přípustných parametrů a následně ve snímku pro všechny kombinace přípustných parametrů hledáme, zda takový objekt existuje. Pomocí hodnotící funkce přiřadíme danému objektu kvalitu nalezeného objektu. Jako hodnotící funkce se často využívá počet hranových bodů ležících na příslušném objektu.

Posledním krokem je nalézt ve vytvořeném prostoru hledané objekty. Nejlepší objekty jsou potom maxima hodnotící funkce v prostoru parametrů.

V souvislosti s detekcí zornice se obvykle využívá Houghova transformace, která hledá kružnice. S využitím parametrické rovnice kružnice (3.2) kde $[a, b]$ je střed kružnice a r je poloměr vidíme, že je potřeba vytvořit tří dimenzionální prostor parametrů (θ je úhel

od $-\pi$ do π který je parametrem rovnice). Aplikací rovnic na každý hranový bod potom přičítáme příslušnost kružnici s danými parametry.

$$\begin{aligned}x &= a + r \cos(\theta) \\y &= b + r \sin(\theta)\end{aligned}\tag{3.2}$$

3.3 Segmentace

Segmentace [8] je úloha, která se zabývá rozdělením snímku podle určitých vlastností snímku v dané oblasti. V našem případě nás bude zajímat segmentace s ohledem na podobnost v barvě (jasu), případně polohu.

3.3.1 K-means

K-means [8] je metoda, která dělí vstupní data na k shluků tak, že minimalizuje kvadratickou chybu. Algoritmus probíhá iterativně a je závislý na vstupním stavu. Vstupní stav udávají počáteční středy regionů. V každém kroku se, na základě minimální vzdálenosti, rozřadí všechna data ke středům a středy se z nově vzniklých shluků přepočítají. Algoritmus končí, pokud již nedochází ke změnám mezi shluky, případně omezeným počtem iterací. Jako souřadnice vstupních dat je možné v případě zpracování obrazu brát například jas nebo jas v kombinaci s polohou. Implementace v našem případě dělí snímek pouze podle jasu (byla provedena s využitím [1]).

3.3.2 Mean-shift

Mean-shift segmentace [6, 22] vyhledává shluky v n -rozměrném prostoru, který je tvořen souřadnicemi polohy ve snímku a jasovými složkami. Slouží k nalezení lokálních maxim a tedy rozdělení na segmenty s podobnou barvou a polohou (segmenty jsou souvislé).

K nalezení shluku se využije na každý pixel obrázku hledání váženého průměru těžiště v okolí daného bodu, přičemž se hledá jeho lokální maximum. Pro nalezení se využívá kruhově symetrická funkce, která má název Epanečnikův kernel (3.3), kde c je konstanta upravující jádro na součet do hodnoty jedna a x jsou souřadnice bodů (poloha a barva).

$$K(x) = \begin{cases} c(1 - \|x\|)^2 & \text{pro } \|x\| \leq 1 \\ 0 & \text{jinak} \end{cases}\tag{3.3}$$

Pro souřadnice, které leží uvnitř Epanečnikova kernelu, dostaneme (3.4) (kde x je souřadnice pro kterou výpočet provádíme a x_i jsou jednotlivé souřadnice uvnitř jádra), čímž vypočteme požadované těžiště.

$$f(x) = \frac{1}{n} \sum_{i=1}^n (x - x_i)\tag{3.4}$$

Míru citlivosti na změnu souřadnice (jasu či polohy) udává jediný parametr, kterým je velikost jádra. Upravujeme tedy hodnotu, kdy je příslušná souřadnice uvnitř jádra či mimo. Tedy upravením výpočtu normy příslušným koeficientem $1/h$ zvětšíme či zmenšíme okolí. h upravuje velikost jádra. V případě, že chceme rozlišit citlivost na barvu a na polohu použijeme při výpočtu normy rozdílný koeficient pro souřadnice polohy a rozdílný koeficient pro souřadnice jasu.

Výpočtem prvního těžiště pro každý bod snímku algoritmus nekončí. Iterativním způsobem opakujeme výpočet vždy s aktuální hodnotou těžiště, dokud změna není menší než příslušný práh $1/1000$ tím hledání ukončíme. Následně spojíme oblasti, které

dokonvergovaly ke stejnému těžišti s tolerancí stejnou jako při splnění podmínky, zda je příslušná souřadnice uvnitř jádra. Tyto oblasti jsou našimi hledanými segmenty. (Implementace byla provedena s využitím [23].)

3.4 Učící algoritmy

V této části se budeme zabývat takzvaným učením s učitelem. Cílem je na základě dostupných klasifikovaných dat natrénovat algoritmus, který je schopen klasifikovat předložená data do skupin podle trénovací množiny.

3.4.1 Kaskádní detektor

Detektor pracující s příznaky předzpracovaného šedotónového obrázku. Kaskádní detektor (podle implementace MATLAB - Computer Vision toolboxu [13]) pracuje na základě kaskádního klasifikátoru, který je složen z jednoduchých klasifikátorů metodou Boostování. Jednotlivé fáze jsou označovány jako *stage*. Takto složený klasifikátor je následně velmi odolný. Jednotlivé stage jsou pouze detektory určující, zda je sada příznaků pozitivní či negativní.

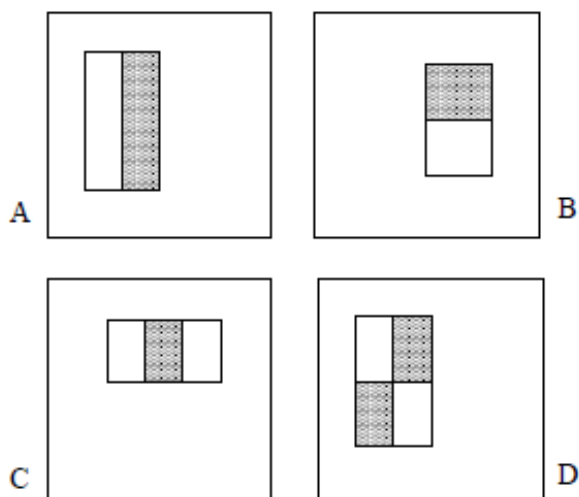
Naučený detektor při klasifikaci prochází plovoucím oknem zkoumaný snímek. Na základě jednotlivých fází buď potvrdí možnost výskytu požadovaného objektu a pokračuje další stage, nebo výskyt zamítne a pokračuje se dalším oknem. V případě, že všechny stage potvrdí výskyt příznaku, je oblast označena jako hledaný objekt. Tento přístup označí ve snímku všechny oblasti s výskytem požadovaného objektu.

Pro správné naučení klasifikátoru je tedy třeba dostatečný počet obrázků objektu, tak značný počet negativních obrázků, které jsou tvořeny většinou obvyklým pozadím hledaného objektu. Negativní obrázky jsou velmi důležité, neboť finální detektor v prvních fázích zamítá jednoznačně negativní oblasti.

Detektor může využívat různé druhy příznaků. Jednou z možností je Haarova báze funkce (*Haar*), Další je histogram orientovaných gradientů (*HOG*). Třetí možností je potom lokální binární vzor (*LBP*).

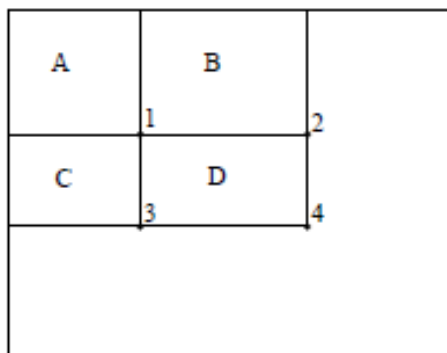
3.4.1.1 Haar

Detektor, který využívá Haarovu bázi se označuje obvykle jako Viola-Jones detektor [28]. Jak naznačuje obecný popis metody detektoru, Haarovy příznaky se využijí k předzpracování obrazu, pro následné využití upravené metody AdaBoost. Použité příznaky jsou znázorněny 3.1. Z tvaru příznaků je zřejmé, že nejcharakterističtější se projeví hrany.



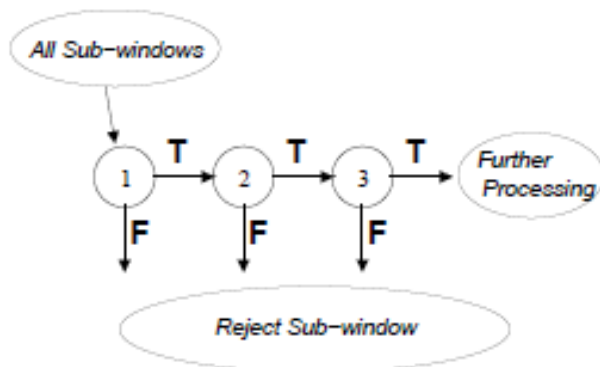
Obrázek 3.1. Haar příznaky - masky, které se využijí k výpočtu hodnoty příznaku pro příslušný obdélník snímku, rozdílné barvy znázorněného příznaku vyjadřují pixely které se přičítají a které odečítají [28].

Pro snadný výpočet hodnoty příznaku se využívá takzvaný integrální obrázek 3.2, ve kterém platí, že hodnota v daném bodě je rovna součtu všech pixelů obdélníku od levého horního rohu po příslušný bod (v 3.2 plocha *A* má součet v bodě 1). Jedná se o řádkový kumulativní součet, následovaný sloupcovým kumulativním součtem. V takovémto obrázku snadno nalezneme hodnotu příslušného obdélníku.



Obrázek 3.2. Integrální snímek (hodnota obdélníku *D* je rovna hodnotám v očíslovaných bodech podle rovnice $(1 + 4) - (2 + 3)$) [28].

Na základě velkého množství příznaků se využije metoda AdaBoost pro sestavení jednotlivých fází detektoru. Jednotlivé jednoduché klasifikátory (stage) hledají několik málo příznaků, aby se minimalizoval výpočetní výkon a nepočítaly se všechny u všech oken. Vytvoří se tedy jednoduché rozhodování s určitým prahem, zda je příznak obsažen v příslušném objektu či nikoliv a případně se přidá do finálního klasifikátoru. Defaultní metoda AdaBoost se snaží minimalizovat chybu falešných detekcí, zde se však minimalizují špatné negativní detekce vzhledem k několika fázím detektoru. Klasifikace následně probíhá podle modelu na 3.3.



Obrázek 3.3. Schéma detektoru [28] („All Sub-windows“ – všechna testovaná okna snímku; T – potvrzení detekce, F – zamítnutí detekce; „Reject Sub-window“ – zamítnutí okna; „Further Processing“ – další zpracování).

■ 3.4.1.2 LBP

Lokální binární vzor (LBP) je rotačně a šedotónově invariantní (různý rozsah mezi minimálními a maximálními hodnotami snímku). Pro popis (LBP), jak je popsán v konkrétní implementaci [17], je třeba definovat kruhově symetrické okolí. Kruhově symetrické okolí přísluší bodu, pro který počítáme vzor a má určitý poloměr a počet bodů, které nás zajímají. Jedná se tedy o body, které jsou symetricky, se stejnou vzdáleností, rozmístěny na kružnici o daném poloměru kolem středu (zkoumaný bod). Zajímají nás jasové hodnoty obrázku v daných bodech.

Pro hodnotu vzoru v daném bodě potom platí rovnice (3.6), kde P je počet bodů okolí a R je poloměr okolí. g_p jsou potom jasové hodnoty bodů okolí a g_c je jasová hodnota středu. V rovnici se dále využívá funkce (3.5).

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (3.5)$$

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (3.6)$$

Tak získáme binární vzor, ale není rotačně invariantní. Rotační invariance dosáhneme rotací získané binární řady tak, aby začínala největším možným počtem nul. Z takto získaných hodnot, převedením binární řady na číslo, bychom již mohli udělat histogram a získali bychom příznak pro určité okno. Další modifikace spočívá ve změně hodnot pro řetězce, které obsahují více než dvě změny posloupnosti z nul na jedničky nebo naopak. Pokud je těchto změn více, získá hodnotu o řád vyšší, než je maximální hodnota původního řetězce.

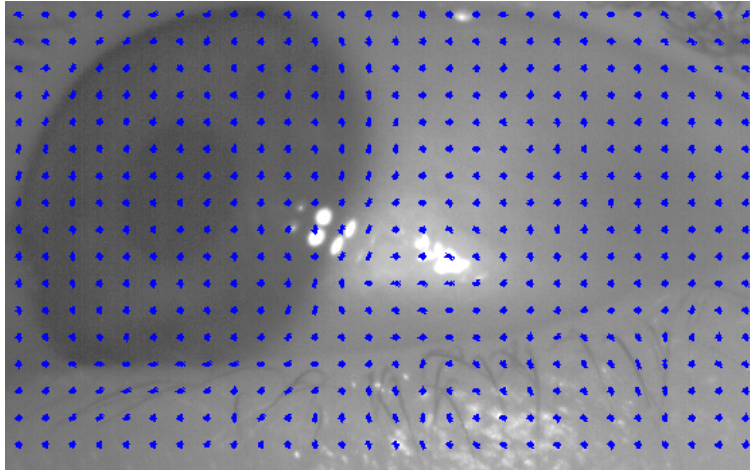
$$VAR_{P,R} = \frac{1}{P} \sum_{p=0}^{P-1} (g_p - \mu)^2, \quad kde \quad \mu = \frac{1}{P} \sum_{p=0}^{P-1} g_p \quad (3.7)$$

Další úpravou je výpočet variance (3.7). Každá hodnota vzoru popsaného v předchozím odstavci se vydělí touto variancí. Na takto získané příznaky se aplikuje algoritmus učení a detekce, podobně jako je popsáno v předchozí části.

■ 3.4.1.3 HOG

Histogram orientovaných gradientů [5]. Jak již název napovídá, jedná se o využití vzoru podle četnosti výskytu gradientů v daném okně snímku. Pro univerzálnost na rozdílné

vstupní snímky je třeba provést normalizaci jasových složek vstupního obrázku. Následuje výpočet gradientu pro zvolenou velikost bloku. Pro daný počet sledovaných směrů gradientu se přiřadí hodnoty gradientu příslušející jednotlivým směrům v bloku. Hodnoty je třeba normalizovat. Histogram takto získaných hodnot pro celé okno se použije jako hledaný příznak. Na obrázku 3.4 jsou modře nakresleny normované hodnoty gradientů pro příslušný blok. Velikost modrého objektu v příslušném směru je dána velikostí gradientu v daném směru. Zbytek učícího a detekčního algoritmu je obdobný jako v předchozích případech, pouze s touto definicí vzoru.



Obrázek 3.4. HOG příznaky.

■ 3.4.2 Neuronová síť

Neuronová síť [29], je učící algoritmus založený na principu biologických neuronů. Matematicky můžeme popsat funkci neuronu jako váhovaný součet vstupních parametrů, který generuje výstupní stav neuronu (přenosovou funkci). Výstup se použije pro účely klasifikace, zda je objekt popsán sadou parametrů, hledaným objektem. Jako konkrétní implementaci jsme využili MATLAB – Neural Network Toolbox [14].

■ 3.5 Fitování kružnice, elipsy

Pro detekci zornice očekáváme hledání kružnice, případně elipsy. Potřebujeme tedy pravidlo pro určení jak je daný objekt podobný tomuto tvaru. Dalším využitím tohoto kroku je získání parametrů detekovaného objektu.

■ 3.5.1 Kritérium podobnosti kružnici, podobnost

Intuitivní možností je porovnávání vzdálenosti okrajových bodů od středu objektu. Nejokrouhlejší objekt je v takovém případě s nejvyrovnanější vzdáleností od středu. Jako střed jsme zvolili těžiště objektu tedy průměr poloh všech bodů příslušejících danému segmentu. Potom je kritériem kruhovitosti (3.8) kde n je počet bodů obvodu, r_i jsou jednotlivé vzdálenosti bodů od těžiště a \bar{r} je průměrná vzdálenost od těžiště.

$$K = \frac{1}{n} \sum_{i=1}^n |r_i - \bar{r}| \quad (3.8)$$

Nejvíce kruhový objekt je potom ten, který má nejnížší hodnotu kritéria. Jedná se totiž o součet rozdílů od ideálního kruhu vydělený počtem prvků obvodu. Toto kritérium předpokládá vyrovnaný průběh vzdáleností, elipsa tedy nebude mít tak příznivé

hodnoty jako kružnice, v některých částech práce však pracujeme se zjednodušeným modelem zornice jako kružnicí. Pro eliminaci vlivu této nedokonalosti jsme v některých částech penalizovali kritérium rozdílem nejmenší a největší vzdálenosti od středu, která v případě vyrovnaného kruhu je téměř nulová, proto kritérium nezmění, pokud však jde o protáhlý element dojde k velké penalizaci.

■ 3.5.2 Kritérium podobnosti kružnici, poměr plochy a obvodu

Další možností pro hodnocení kruhovitosti je porovnání na základě vzorce (3.9) kde S je plocha objektu a O je obvod daného objektu [9].

$$K = \frac{4S\pi}{O^2} \quad (3.9)$$

$$O = 2\pi r, \quad S = \pi r^2, \quad K = \frac{\pi r^2}{2\pi r} \frac{4\pi}{2\pi r} = \frac{4\pi^2 r^2}{4\pi^2 r^2} = 1 \quad (3.10)$$

Nejvíce kruhovitý objekt má potom kritérium nejbližší 1 pro ideální kruh totiž platí (3.10), kde r je poloměr. Toto kritérium je však hodně citlivé na malá vyboulení kružnice, která způsobí rychlý růst obvodu, plocha se však příliš nemění.

■ 3.5.3 Fitování elipsy

Pro určení parametrů regionu a odhad parametrů vepsané elipsy jsme využili funkci MATLAB – Image Processing Toolboxu. Jedná se se o funkci *regionprops* [15], která měří vlastnosti regionů. Konkrétně jsme využili parametrů určení těžiště objektu, jako středu elipsy. Dále pak velikosti poloos elipsy, která má stejný druhý centrální moment jako zkoumaný region a nakonec natočení takto získané elipsy.

Kapitola 4

Praktická část

Kapitola se zabývá popisem zpracovávaných dat, která jsou využita k demonstraci funkce jednotlivých algoritmů a zároveň jsou referenčními daty pro porovnání. Dále je zde provedena charakteristika vstupních dat a na jejich základě vytvořena metodika hodnocení výsledků. Následuje popis způsobu určení věrohodnosti pro výběr z několika kandidátů na zornici.

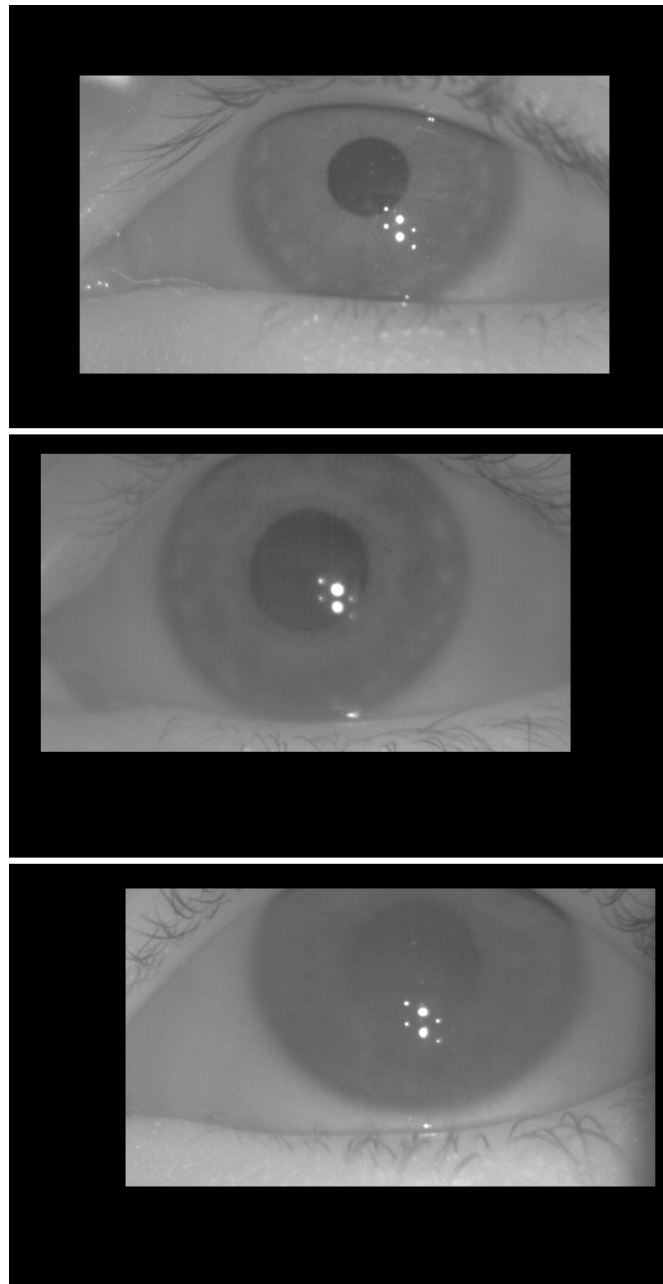
4.1 Použitá data

Jako referenční data jsou využity výsledky měření přístrojem I4Tracking® od firmy Medicton Group, který zpracovává zaznamenané snímky pomocí softwarového řešení EyeGaze SDK od Firmy Eyedea Recognition. Z výstupů tohoto software získáme pozici a rozměry zornice.

Brýle, které slouží k pořizování záznamu oka, jsou na obrázku 4.1. Tracker využívá metody záznamu oka odrazem od polopropustného skla brýlí. Kamera na druhé straně potom zaznamenává scénu. Ukázka snímků z kamery zaznamenávající oko, které se následně využijí ke zpracování, jsou na obrázku 4.2, kde jednotlivý snímek má rozlišení $752 \times 480 \text{ px}$.



Obrázek 4.1. Brýle pro záznam oka [16].

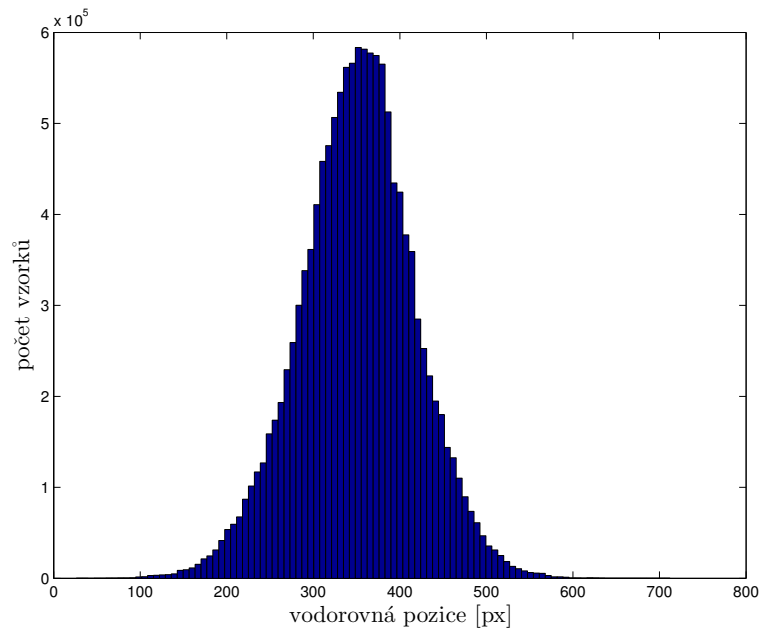


Obrázek 4.2. Zdrojové obrázky z kamery shora srovnané od nejlepšího k detekci až dole obtížně detekovatelné.

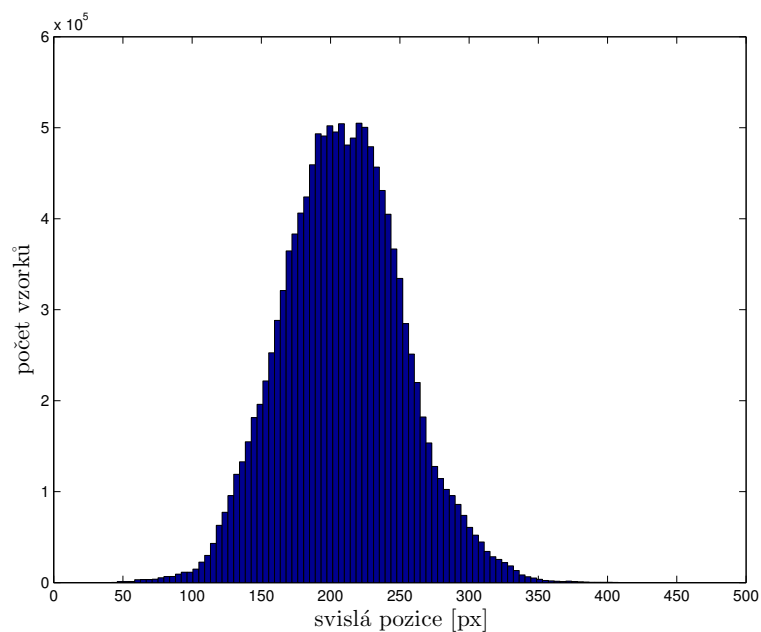
Zaznamenané snímky využívá nejen referenční metoda, ale jsou zároveň i vstupními daty pro testování algoritmů popsanych v následujících kapitolách.

4.2 Analýza dat

Dostupná data referenční metody obsahují 13 457 584 snímků, z 343 měření. Detekované parametry vykazují značnou závislost, kterou je možné pozorovat na histogramech velikosti parametrů na obrázcích 4.3, 4.4, 4.5 a 4.6.

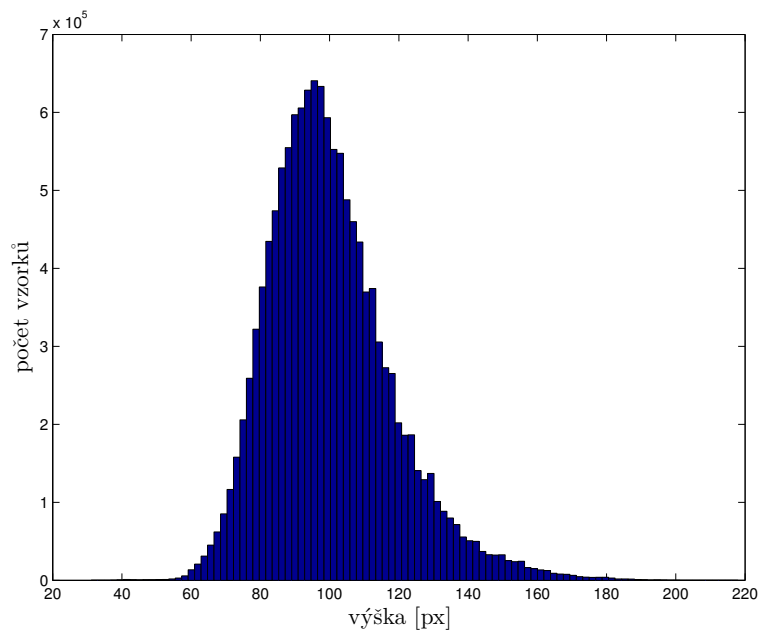


Obrázek 4.3. Histogram četností pozic zornice ve vodorovném směru.

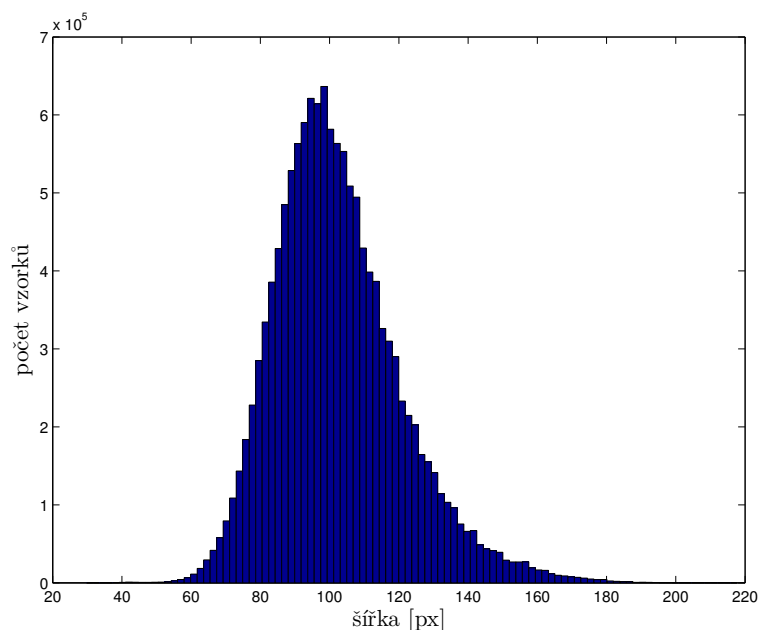


Obrázek 4.4. Histogram četností pozic zornice ve svislém směru.

Pozice zornice značnou měrou, podle očekávání, směřuje ke středu snímku, jak potvrzují histogramy četnosti na obrázcích 4.3 a 4.4. Průměrná hodnota polohy je ve vodorovném směru $350,8 \text{ px}$ a ve svislém směru $209,0 \text{ px}$ při velikosti vstupního obrázku $752 \times 480 \text{ px}$.

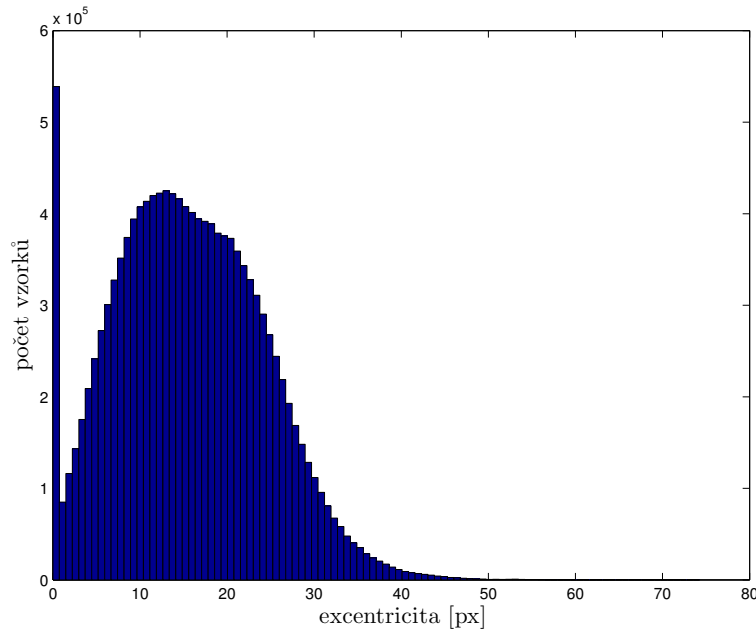


Obrázek 4.5. Histogram četností výšky elipsy zornice.



Obrázek 4.6. Histogram četností šířky elipsy zornice.

Z histogramů na obrázcích 4.5 a 4.6, můžeme vyčíst četnosti rozměrů zornice. Jedná se tedy o elipsy. Průměrná velikost je dána výškou $100,2 \text{ px}$ a šířkou $102,4 \text{ px}$. Fakt, že jsou oba histogramy velmi podobné, nám naznačuje, že se jedná spíše o speciální případ elipsy, kterou je kružnice. Tuto tendenci můžeme dále potvrdit z histogramu excentricity sledovaných elips 4.7, jenž má průměrnou hodnotu $15,6 \text{ px}$ což je vzhledem k velikosti poloos elipsy kolem 50 pixelů poměrně velká podobnost kružnici. Charakter kružnice podporuje i vysoká špička v histogramu u hodnoty 0 px .



Obrázek 4.7. Histogram četností excentricity elipsy zornice.

Na základě těchto dostupných vstupních dat byly zvoleny tři přístupy zpracování pro implementaci. Jednotlivé metody jsou popsány ve třech následujících kapitolách. První, v dostupných materiálech často využívanou metodu, s využitím hranové detekce. Druhou s využitím přímého učení příznaků obrázku a třetí s využitím segmentace a následné klasifikace jednotlivých objektů.

Pro testování našich algoritmů jsme provedli na všech snímcích odstranění černého rámečku. Raným testováním jsme odhalili problém s odrazy infračervených led diod v záznamu, které slouží k přisvícení scény a zvýraznění zornice (popsáno v části 2.1). Tyto odrazy měly velký vliv na funkci algoritmů, vzhledem k extrémním hodnotám jasu v těchto bodech. Proto byly tyto body nahrazeny průměrnou hodnotou okolních pixelů. Body byly detekovány jako oblasti s maximálním jasnem ve snímku. Následně bylo okolí získané dilatací o 5 pixelů nahrazeno průměrem ohraničujícím tuto oblast získané dilatací o 6 pixelů (hodnoty určeny experimentálně, body jsou vzhledem k umístění při záznamu přibližně stejně velké). Pro testování algoritmů jsme provedli výběr přibližně 100 000 snímků rovnoměrně ze všech záznamů.

4.3 Metodika hodnocení úspěšnosti

Úkolem je mimo jiné porovnat námi implementované přístupy řešení detekce zornice s detekcí pomocí referenční metody EyeGaze SDK. Pro jisté určení správnosti je nutná kontrola pohledem člověka, aby se vyloučila případná nedokonalost referenčního algoritmu. Tento postup byl při vývoji aplikován pro ladění jednotlivých metod. V dokumentu dále vždy dáváme příklad pro vizuální kontrolu. Pro kontrolu většího množství zpracovaných snímků se však spokojíme s myšlenkou, že detekce pomocí referenční metody je správná. Parametry detekované zornice porovnááme s výstupem referenční metody, která má však podle dostupných snímků též určité nedokonalosti, čímž může růst naše výsledná nepřesnost.

Nejjednodušší kontrolou, kterou je možné aplikovat, i v případě, kdy nalezneme pouze kružnici místo referenční elipsy, je detekce, zda vzdálenost nalezeného středu od refe-

renčního je menší, než velikost referenční elipsy. Tento parametr tedy zvolíme pro vyřazení špatných detekcí, jejichž střed leží mimo předpokládanou zornici. Pokud je tato vzdálenost menší považujeme detekci za úspěšnou.

Dalším krokem je zhodnotit kvalitu detekce. Pro hodnocení kvality tedy využijeme, jako jedno z kritérií, vzdálenost středů referenčního a nalezeného objektu. Tímto parametrem určíme úspěšnost algoritmu s maximální vzdáleností středů 20% velikosti referenční elipsy. Jako další parametr, pro porovnání kvality jednotlivých metod, využijeme porovnání plochy s referenční elipsou. Hodnotíme tedy rozdíl plochy v procentech z plochy referenční elipsy. Přesnost plochy je důležitá při využití detekce v pupilometrii, kdy nás nezajímá poloha, ale změna velikosti zornice. Tyto parametry jsou voleny s ohledem na skutečnost, že referenční metodu považujeme za přesnou správnou detekci, přestože z detekovaných snímků je někdy vizuální kontrolou zřejmé, že detekce není úplně přesná a proto hodnocení rozdílu těchto parametrů musíme brát s ohledem na velikost zornice.

Nakonec budeme sledovat i dobu zpracování, která je též důležitým kritériem vzhledem k velkému množství zpracovávaných dat (vzorkovací frekvence kamery 150 snímků za sekundu).

4.4 Určení věrohodnosti řešení

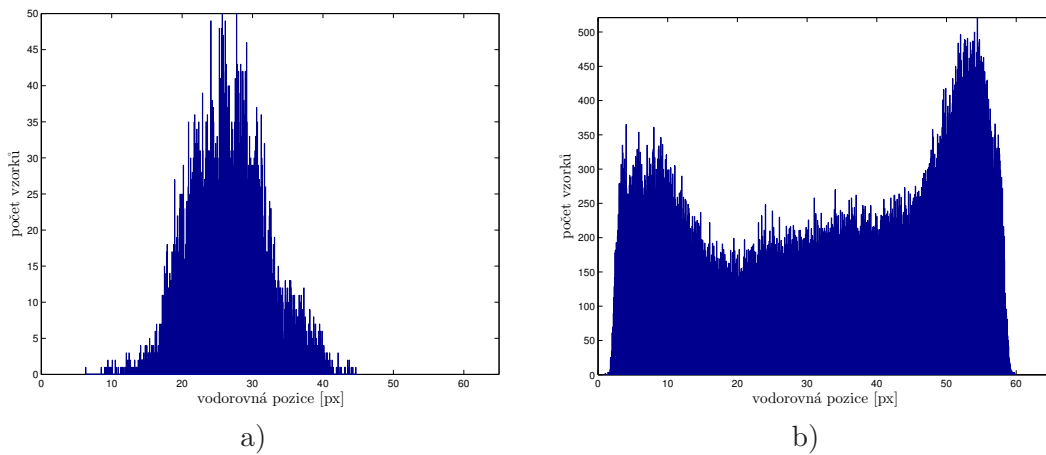
Některé z metod zpracování vrací několik potenciálních kandidátů na pupilu, případně dělí snímek pouze na určité regiony, ze kterých je třeba zvolit nejlépe odpovídající objekt. Pro tyto účely je třeba využít charakteristických znaků zornice ve snímku, k určení věrohodnosti detekce. Předpokládáme tedy vytvoření klasifikátoru, který bude využívat charakteristické parametry zornice. V prvním kroku je třeba nalézt parametry, které jsou rozdílné pro zornici a okolní prostředí. Následně sestavíme klasifikátor s využitím těchto hodnot.

Pro vytvoření určité statistiky, co platí pro pupilu a co pro části okolí je třeba snímky rozdělit na nějaké regiony. K tomuto účelu jsme využili algoritmus, který je též jednou z testovaných metod pro detekci. Jedná se o mean-shift jehož použití je podrobněji popsáno v kapitole 7.

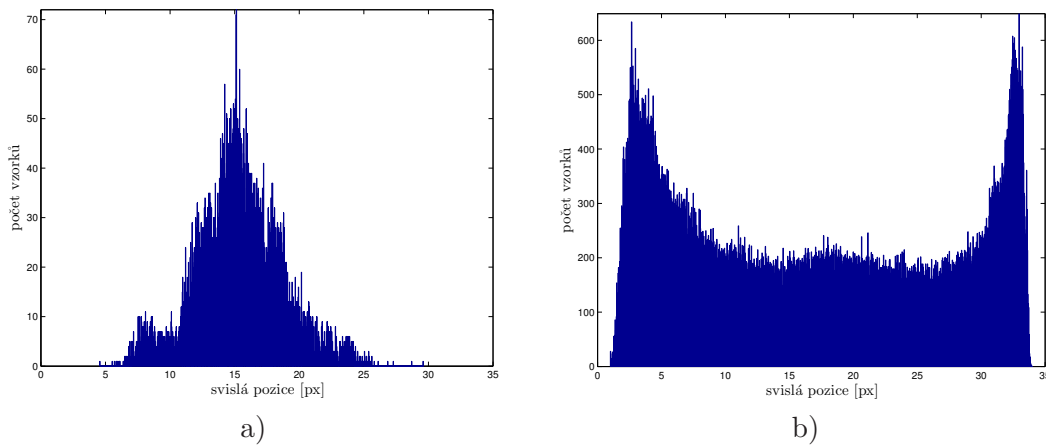
Zde se tedy spokojíme s tím, že jsme rozdělili snímek na regiony podle jasu a polohy. Pro jednotlivé regiony jsme si spočítali následující charakteristiky (v důsledku zpracování na zmenšeném snímku jsou i parametry popsané ve statistice poměrně zmenšeny vzhledem k původní velikosti snímku).

První z parametrů je poloha ve snímku, kterou můžeme pozorovat na histogramech 4.8 ve vodorovném směru a 4.9 ve směru svislém. Na levých histogramech jsou pozice segmentů příslušejících zornici, vpravo pak pozice segmentů které nejsou zornicí. Z tvaru grafů můžeme usuzovat, že je vysoká pravděpodobnost pro odlišení segmentů, které jsou zornicí, na základě tendence, že je obvykle uprostřed snímku.

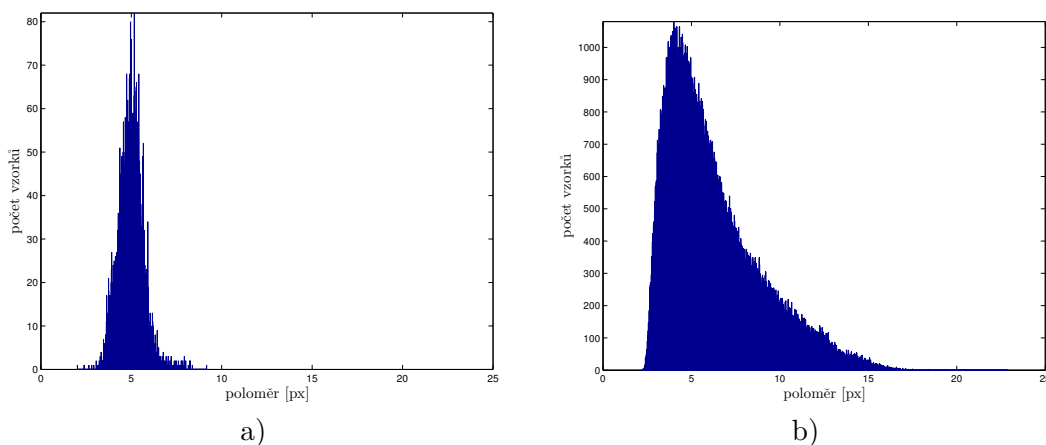
Další charakteristikou je „poloměr“, který dostaneme jako průměrnou vzdálenost krajních bodů regionu od jejich těžiště. Porovnání tohoto parametru je znázorněné na histogramech 4.10. Opět vlevo je histogram pro segmenty příslušející zornici a vpravo pro ostatní segmenty. Z grafu je patrné, že je velké množství segmentů, které nepříslušejí zornici a mají podobnou velikost jako zornice, tento parametr však může eliminovat příliš velké poloměry.



Obrázek 4.8. Histogramy četností pozice zornice ve vodorovném směru, porovnání pozice zornice (a) a pozic ostatních regionů (b).



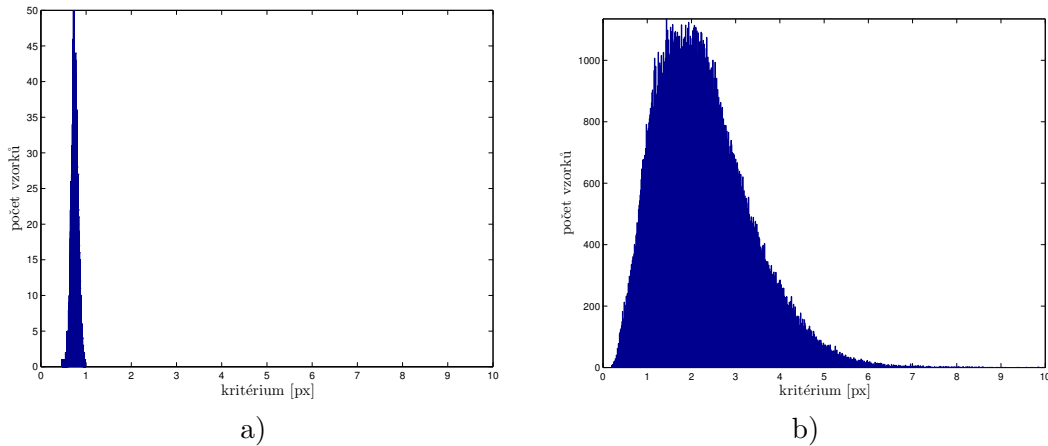
Obrázek 4.9. Histogramy četností pozice zornice ve svislém směru, porovnání pozice zornice (a) a pozic ostatních regionů (b).



Obrázek 4.10. Histogramy četností poloměrů, porovnání poloměrů zornice (a) a poloměrů ostatních regionů (b).

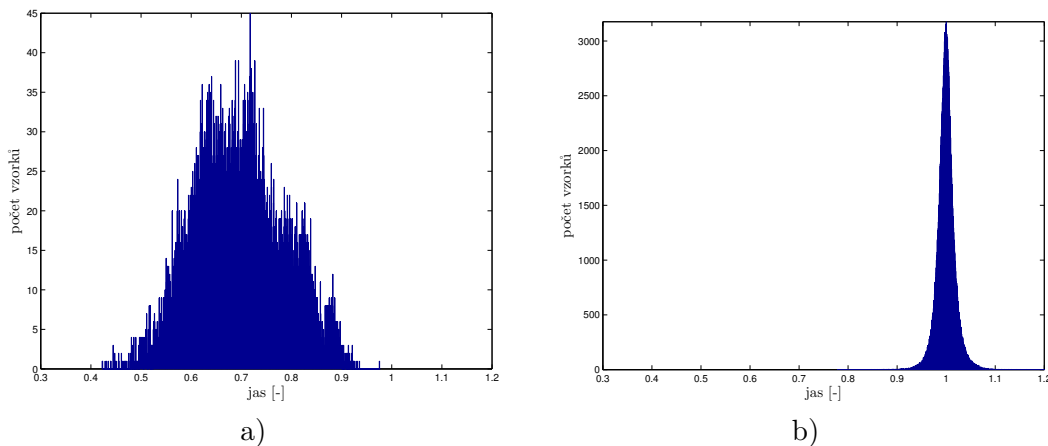
Následuje parametr popisující míru odchýlení od kruhového tvaru charakterizovanou podle pravidla popsaného v části 3.5.1. Tento parametr je pro porovnání znázorněn na histogramech 4.11. Podobně jako u kritéria poloměru se parametry zornice překrývají

s částí, kdy se nejedná o segmenty zornice. Opět je zde ale velká část segmentů, která má hodnoty mimo tuto oblast a je možné je pomocí tohoto kritéria vyřadit.



Obrázek 4.11. Histogramy četností hodnot kritéria kruhovitosti, porovnání kritéria zornice (a) a ostatních regionů (b).

Posledním zmíněným poměrně výrazným parametrem je fakt, že pupila je ve většině případů tmavší než duhovka. Abychom omezili vliv různého osvětlení snímku, za parametr zvolíme poměr průměrné hodnoty jasu zornice k průměrné hodnotě jasu zbylého snímku. Tento parametr je porovnán na histogramech 4.12. Zde je nejvýraznější tendence oddělení segmentů, které jsou zornicí od segmentů, které zornici neobsahují. Tato charakteristika tedy naznačuje, že nejjednodušším způsobem výběru zornice z kandidátů je výběr podle poměru jasu objektu vzhledem k okolí.



Obrázek 4.12. Histogramy četností poměru jasu segmentu vzhledem k okolí, porovnání pro segmenty zornice (a) a ostatních regionů (b).

Ze statistiky dostupné na předchozích histogramech (4.8, 4.9, 4.10, 4.11 a 4.12) je zřejmé, že parametry segmentů mají ve většině případů výrazné hranice mezi zornicemi a ostatními segmenty, nebo mohou oddělit alespoň část negativních případů. Tato skutečnost umožňuje využít předpoklad užití pravděpodobnosti hodnoty daného parametru jako metody k určení věrohodnosti, zda je zvolená oblast zornice či nikoliv.

Na základě předchozího zjištění jsme provedli natrénování klasifikátorů pro různá vstupní data s využitím neuronové sítě. Vzhledem k testování několika metod s různými předpoklady a možnostmi, jsme vytvořili samostatný klasifikátor pouze na kružnice, který se využívá na zmenšených snímcích, kde zanedbáváme rozdílné poloosy elipsy. Druhou možností je potom natrénovaný klasifikátor na elipsy podle referenčního řešení

(negativní případy jsou však generovány s využitím negativních kandidátů jednotlivých metod).

Jako klasifikátor jsme využili implementaci neuronové sítě v prostředí MATLAB (Neural Network Toolbox) [14]. Konkrétně využíváme dopřednou neuronovou síť s trénovací metodou „Scaled conjugate gradient backpropagation“. Klasifikátor má za úkol určit pravděpodobnost s jakou se jedná o zornici a s jakou o okolí.

Pro využití v metodě hledání kandidátů pomocí kaskádního detektoru jsme jako pozitivní příklady využily správné parametry zornice podle referenčního řešení. Negativní příklady jsme generovali jako neúspěšné kandidáty generované touto metodou při porovnání s referencí. Za parametry pro učení neuronové sítě jsme zvolili (na základě poznatků ze statistiky výše): polohu středu, velikosti poloos, poměr průměrného jasu, oblasti vzhledem k okolí a velikost plochy. Zde jsme konkrétně využili neuronovou síť s vnitřní vrstvou se 6 neurony a výstupní vrstvou se dvěma neurony pro pozitivní a negativní detekce.

Další případ, kde je třeba volit mezi potenciálními kandidáty je metoda segmentace. V tomto případě jsme pro trénování klasifikátoru provedli segmentaci snímků pomocí mean-shift, jak je podrobněji popsáno v kapitole 7. Segmenty jsme rozdělili do skupin, které přísluší zornici a které ne (podle referenčního řešení, zda se překrývaly většinou plochy s referenční zornicí či nikoliv). Tyto regiony jsme využili jako trénovací data pro dva různé klasifikátory.

První z klasifikátorů má za úkol detekovat zornici jako kružnici. Proto jsme ze segmentů získali následující parametry pro natrénování neuronové sítě: polohu středu, poloměr, kritérium kruhovitosti podle 3.5.1 (proč jsme volili tuto variantu podrobněji v kapitole 7) a poměr průměrného jasu uvnitř oblasti vzhledem k okolí. Tento klasifikátor má obdobnou strukturu jako předchozí, pouze skrytá vrstva obsahuje 5 neuronů.

Druhý z klasifikátorů využitý u metody segmentace má za úkol detekovat mezi segmenty správnou elipsu. Proto jsme jako vstupní parametry volili: polohu středu, velikosti poloos, úhel natočení, rozdíl vypočítané polohy z parametrů elipsy od skutečné plochy, průměrný jas oblasti, poměr jasu oblasti k okolí a poměr velikosti poloos (parametry získané fitováním elipsy do segmentů). Klasifikátor má opět obdobnou strukturu jako oba předchozí, liší se pouze tím, že skrytá vrstva obsahuje 9 neuronů.

Kapitola 5

Metoda na principu hranové detekce

Mezi přístupy popsány v dostupných člancích vykazoval pěkně dokumentované výsledky článek [27]. Proto jsme implementovali detekci hran podle autorů tohoto článku. Jedná se o využití hranového detektoru označovaného jako EDPF [26, 25].

Následuje popis algoritmu podle [27, 26, 25]. Nejdříve se provede filtrování Gaussovským jádrem (5×5 a $\sigma = 1$). Dále detektor využívá jednoduchý operátor k detekci hran ve svislém a vodorovném směru. Využije se Prewittův operátor. Z těchto hodnot se spočítá následně mapa gradientu $G = \sqrt{G_x^2 + G_y^2}$ (kde G_x jsou hodnoty v horizontálním směru obdobně pak hodnoty ve G_y ve vertikálním směru). Odlišnost od běžných detektorů nastává až v dalším kroku, kdy se provádí hledání hran tak, aby měly pouze šířku jednoho pixelu a byly rozděleny do souvislých částí.

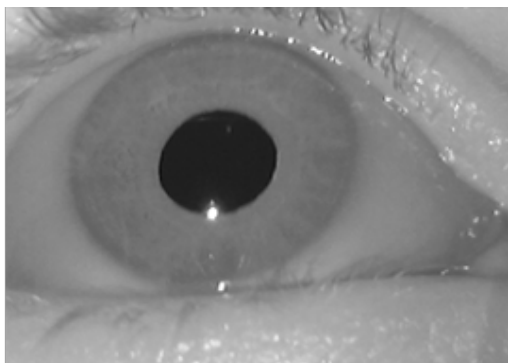
Pro tyto účely se hledají opěrné body, které se volí jako lokální maxima v bodech, kde je horizontální gradient větší než vertikální, tedy zkoumáme zda body nad a pod mají nižší hodnotu než zkoumaný pixel, pokud ano, označíme bod jako opěrný. Pokud převládá vertikální gradient nad horizontálním, zkoumáme obdobně body vlevo a vpravo od tohoto pixelu.

Následuje chytré trasování, které slouží k propojení bodů získaných v předchozím kroku. Spojování probíhá tak, že se postupuje přes jednotlivé opěrné body a pokud ještě bod není přidán mezi hrany, přidá se mezi hrany. Pokud zde převládá větší gradient v horizontálním směru, prohledává se okolí tří pixelu vlevo a vpravo. Pokud však převládá vertikální gradient, je postup obdobný, pouze zkoumáme okolí nahoře a dole. Z tohoto opěrného bodu se vydáme oběma zkoumanými směry do okolí s největší hodnotou gradientu. Z následujícího bodu pokračujeme vždy do horizontálního, či vertikálního okolí podle převládající hodnoty gradientu. Volíme takový směr, který má maximální hodnotu v mapě gradientu. Mezi hrany přidáváme body do té doby, dokud nenarazíme na bod, který je již přidán, případně bod s nulovým gradientem. V tom případě pokračujeme dalším opěrným bodem, pokud jsme již všechny nepřidali.

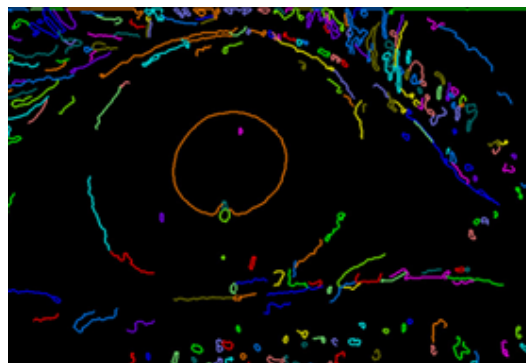
Posledním krokem je vyřídění nežádoucích hranových elementů na základě Helmholtzova principu [26]. Tento krok vyřazuje body na základě délky segmentu a minimální hodnoty gradientu při porovnávání s celým snímkem.

Pro výběr kruhovitých segmentů k dalšímu zpracování, se na základě článku [27] využije na jednotlivé segmenty elementů výpočet směru gradientu (pomocí funkce \arctan z gradientu v horizontálním a vertikálním směru). Tyto hodnoty se naškálují do osmi směrů a pomocí výpočtu entropie se zkoumá vyrovnanost výskytu jednotlivých směrů. Jako kruhové objekty jsou považovány segmenty, které mají nejvyrovnanější výskyt směrů.

Na obrázku 5.1 je vstupní snímek, který využívají k detekci ve zdrojovém článku. Následuje obrázek 5.2, kde jsou barevně vyznačeny hranové elementy. Pro porovnání na obrázku 5.3 je náš vstupní obraz a na snímku 5.4 pak detekce hran. Z tohoto porovnání je zřejmé, že detekce na našich vstupních datech nefunguje podle očekávání. Získáme mnoho kratších hranových elementů, které však obvykle nezvýrazní hledanou zornici. Tento jev je pravděpodobně důsledkem méně ostrých snímků a menších rozdílů v od-



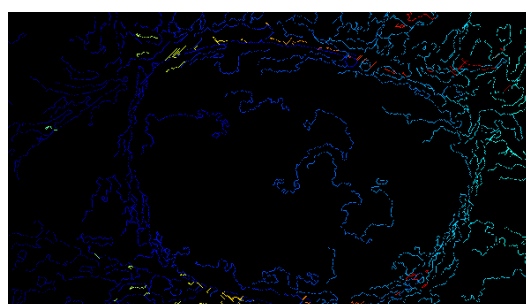
Obrázek 5.1. Vstupní obrázek (předlohy)[27].



Obrázek 5.2. Detekované hrany (předlohy)[27].



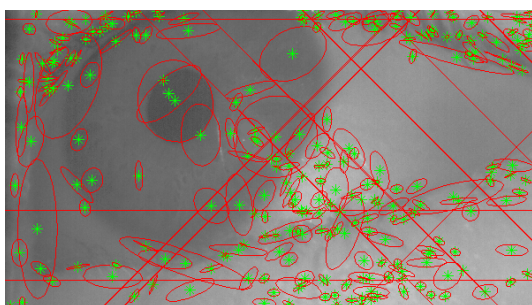
Obrázek 5.3. Vstupní obrázek našeho zpracování.



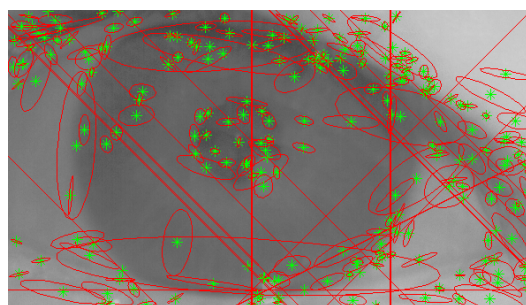
Obrázek 5.4. Detekované hrany na našich vstupních datech.

razivosti zornice a zbytku oka v důsledku přisvícení. Nedostatečný kontrast způsobí, že hrany nejsou bezpečně detekovány.

Hranová detekce funguje v této implementaci na ostrých obrázcích s velkým kontrastem a není tedy pro naše účely dostatečně robustní. Na obrázku 5.5 jsou vykresleny elipsy vepsané nejvíce kruhovitým elementům (250 elementů). Na tomto obrázku je možné vidět několik elementů, které by bylo možné k detekci použít. Většina testovaných snímků však odpovídala spíše situaci na druhém příkladu 5.6. Tento příklad neobsahuje elementy příslušející požadované zornici.



Obrázek 5.5. Obrázek s vykreslením elips nejpravděpodobnějších hran, pozitivní příklad.



Obrázek 5.6. Obrázek s vykreslením elips nejpravděpodobnějších hran, negativní příklad.

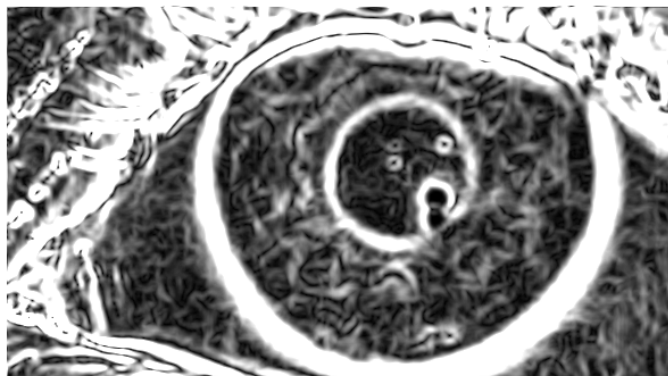
Vzhledem k zjištěným problémům následovala úprava algoritmu s využitím kruhové hranové masky. V původní verzi se využívá hledání hran pouze rozdíly ve vertikálním a horizontálním směru snímku. Tato úprava však rotuje maskou ve všech směrech, pro

detekci hran s různým natočením (měla by se lépe postihnout celá kružnice). Ztratíme tak však možnost trasování hran na základě vertikálního a horizontálního směru gradientu.

Abychom zároveň při detekci hran postihli více směrů rotací masky, ale též podpořili detekci rozmazaných snímků zvolili jsme větší velikost masky, než 3×3 jako v originálním případě. Použili jsme tedy filtr v podobě (5.1), který snižuje vliv bodů se vzdáleností kolmou na výpočet gradientu. Filtr jsme rotovali s krokem $\pi/25$ od $-\pi$ do π jako hodnotu gradientu v požadovaném bodě jsme potom volili maximum ze snímků získaných všemi filtry.

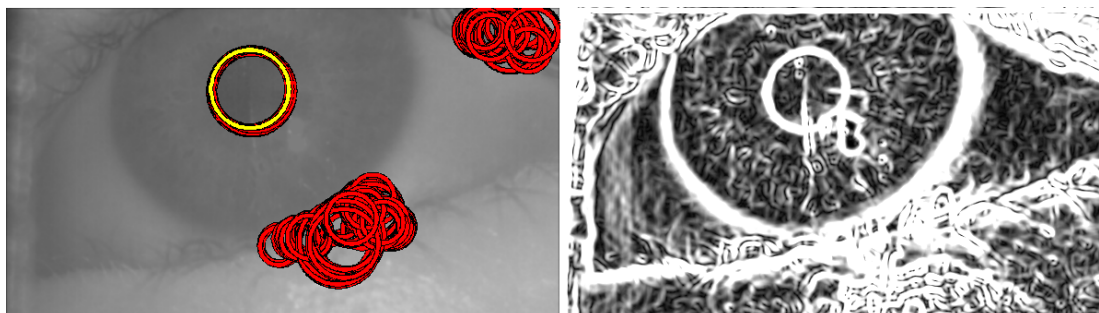
$$\begin{bmatrix} 0.306 & 0.611 & 0.917 & 1.222 & 1.528 & 1.833 & 1.528 & 1.222 & 0.917 & 0.611 & 0.306 \\ 0.306 & 0.611 & 0.917 & 1.222 & 1.528 & 1.833 & 1.528 & 1.222 & 0.917 & 0.611 & 0.306 \\ 0.306 & 0.611 & 0.917 & 1.222 & 1.528 & 1.833 & 1.528 & 1.222 & 0.917 & 0.611 & 0.306 \\ 0.306 & 0.611 & 0.917 & 1.222 & 1.528 & 1.833 & 1.528 & 1.222 & 0.917 & 0.611 & 0.306 \\ 0.306 & 0.611 & 0.917 & 1.222 & 1.528 & 1.833 & 1.528 & 1.222 & 0.917 & 0.611 & 0.306 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.306 & -0.611 & -0.917 & -1.222 & -1.528 & -1.833 & -1.528 & -1.222 & -0.917 & -0.611 & -0.306 \\ -0.306 & -0.611 & -0.917 & -1.222 & -1.528 & -1.833 & -1.528 & -1.222 & -0.917 & -0.611 & -0.306 \\ -0.306 & -0.611 & -0.917 & -1.222 & -1.528 & -1.833 & -1.528 & -1.222 & -0.917 & -0.611 & -0.306 \\ -0.306 & -0.611 & -0.917 & -1.222 & -1.528 & -1.833 & -1.528 & -1.222 & -0.917 & -0.611 & -0.306 \\ -0.306 & -0.611 & -0.917 & -1.222 & -1.528 & -1.833 & -1.528 & -1.222 & -0.917 & -0.611 & -0.306 \end{bmatrix} \quad (5.1)$$

Rotace masky zajistila poměrně značné zvýraznění hran, které je vidět na snímku 5.7. Značně se však zvýšila náročnost výpočtu hran. Snímky však vizuálně nasvědčovaly lepší detekci zornice.

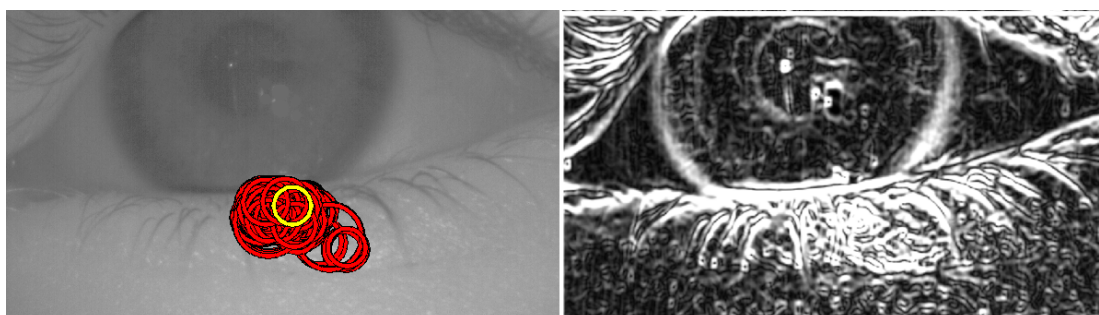


Obrázek 5.7. Hrany zvýrazněny rotační maskou.

Pro ověření úspěšnosti detekce jsme provedli automatickou detekci kružnic pomocí Houghovy transformace. Při testování detekované kružnice závisely na správnosti nalezených hran a vhodně zvoleném sledovaném poloměru. Provedli jsme test pro rozmezí poloměrů kružnic 20 až 100 *px* což by podle statistiky v části 4.2 mělo být dostatečné. Výsledky tohoto experimentu jsou zobrazeny na obrázcích 5.8 a 5.9, kde je vykresleno 1000 kružnic, které jsou podle Houghovy transformace nejvýraznější. První snímek je správná detekce, druhý potom ukazuje špatnou detekci.



Obrázek 5.8. Obrázek s vykreslením kružnic získaných Houghovou transformací (žlutě nejvýraznější kružnice, červeně zbylé detekované), pozitivní příklad.



Obrázek 5.9. Obrázek s vykreslením kružnic získaných Houghovou transformací (žlutě nejvýraznější kružnice, červeně zbylé detekované), negativní příklad.

Upravenou metodou s rotačním filtrem jsme otestovali část vstupních dat, kdy nás zajímalo, kolik testovaných snímků bude mít mezi maximálními hodnotami Houghovy transformace správné detekce (dalším předpokládaným krokem by bylo využití dat analyzovaných v kapitole 4). Doba zpracování jednoho snímku je průměrně 14,61 sekund, již to bylo odrazujícím faktorem od této metody vzhledem k době, kterou trvá referenční řešení (0,10 s). Střed některé z kružnic se však nacházel uvnitř zornice referenčního řešení pouze ve 47,06 %. Proto jsme dále vývoj této metody ukončili (výsledky v tabulce 5.1).

Rotační maska sice přinese zvýraznění hran i u rozmazanějších snímků neřeší však problém, kdy je barva zornice příliš podobná duhovce, kde selhávala předchozí popsána hranová detekce. Metoda by pravděpodobně byla schopna detekovat zornici, je zde ovšem důležité správně nastavit parametry pro aktuální světelné podmínky a sledovanou osobu, případně nastavení použité kamery. Požadavek je však, aby tyto parametry nebylo nutné nastavovat, případně aby se nastavily automaticky. Dalším problémem je poměrně dlouhé zpracování snímku způsobené rotační maskou při hledání hran a využití Houghovy transformace s velkým rozsahem poloměrů.

	detekce hran + Houghova transformace	EyeGaze SDK
Úspěšnost	47,06 %	100 %
Průměrná doba zpracování	14,61 s	0,10 s

Tabulka 5.1. Výsledky hranové detekce vzhledem k referenční metodě, kterou považujeme za správnou (porovnává se pouze, zda mezi nalezenými kandidáty existuje správná detekce, nejedná se tedy o algoritmus s finálními výsledky).

Kapitola 6

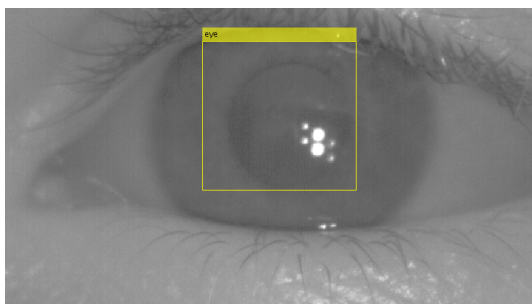
Metoda učení na základě vzoru v obraze

Učící algoritmy se obvykle v oblasti zpracovávané problematiky využívají k detekci obličejů v záznamech případně detekci oka jako takového. Využívají se však i k detekci zornice jako referenční řešení, EyeGaze SDK nebo [30], které sloužilo jako předloha pro naši implementaci.

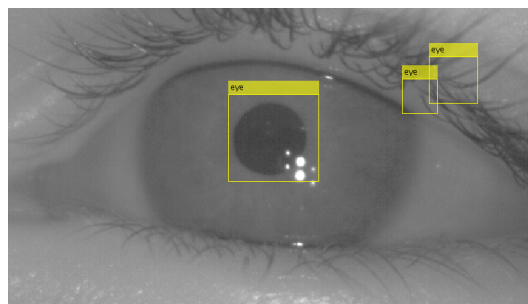
Pro otestování funkce učícího algoritmu jsme použili detektor implementovaný v knihovně Matlabu (Computer Vision Toolbox) [13], podrobněji popsáno v kapitole 3.4.1. Jedná se o učící metodu, založenou na metodě Boosting. K detekci je využíváno vzorů Haar, LBP nebo HOG.

Učení je poměrně zdlouhavou záležitostí, nalezení shody ve snímku však je potom poměrně rychlé. Pro správnou detekci je však nutné velké množství správně i špatně detekovaných, anotovaných snímků. Pro učení byly využity detekované snímky referenčního řešení, ze kterých jsme vybrali obdélníkovou část, která obsahuje zornici. Ze zbylé části snímku jsme generovali negativní příklady.

Natréovaný klasifikátor jsme spustili na testovací množině dat. Jak můžeme vidět z ukázkových snímků 6.1 a 6.2, obvykle dojde k nalezení zornice i v případě, kdy pupila není příliš zřetelná. Naopak, i když je podle lidských smyslů zřetelně oddělitelná, detektor vytvořil falešné detekce jinde.



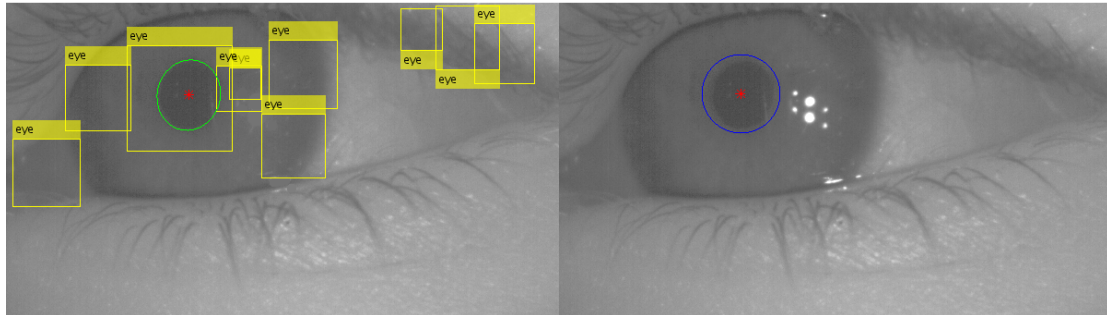
Obrázek 6.1. Správná detekce zornice (žlutě označená oblast detekované zornice).



Obrázek 6.2. Detekce více kandidátů (žlutě označená oblast detekované zornice).

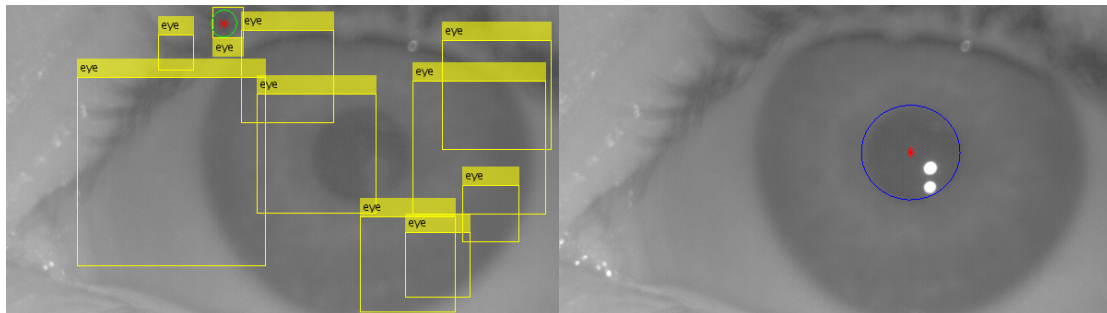
Před detekcí, stejně jako při učení, byly odstraněny odrazy infračerveného přisvětlení pro eliminaci vlivu těchto odrazů na průběh algoritmu, kdy není vždy zaručeno, že tyto odrazy se nacházejí v blízkosti zornice. Pro detekované oblasti následuje aplikace k-means segmentace na dva shluky, z nichž jeden s nižší hodnotou je požadovaná zornice a druhý je okolí, tedy duhovka. Následně do tmavšího regionu byla vepsána elipsa pomocí charakteristických parametrů objektu 3.5.3.

Vzhledem k tomu že detekcí v jednom snímku může být více, je nutné rozhodnout se pro jednoho kandidáta. Z tohoto důvodu byla za pupilu označena elipsa s nejtmaším průměrem jasu.



Obrázek 6.3. Porovnání kaskádního detektoru s refererencí EyeGaze SDK, pozitivní příklad. Vlevo Kaskádní detektor, žlutě nalezené oblasti detektorem, zeleně finální zornice. Vpravo referenční detekce modře. Středý detekovaných zornic červeně.

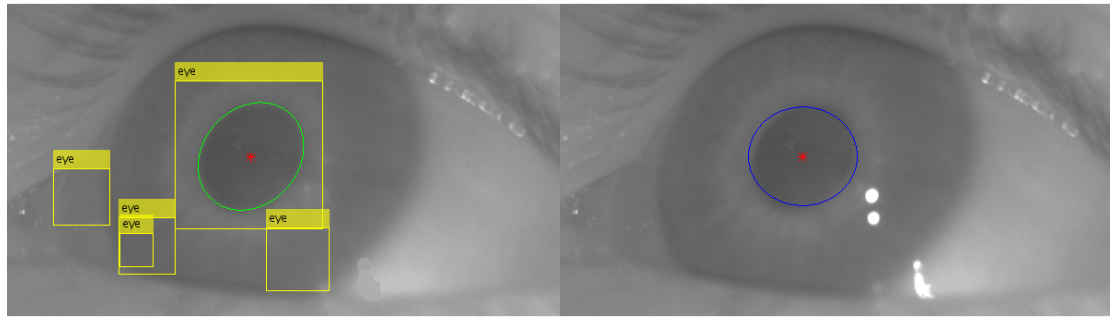
Na snímku 6.3 můžeme pozorovat porovnání detekované zornice s referenčním řešením. Zde je dokonce zřejmé, že detekovaná elipsa lépe odpovídá hledané zornici.



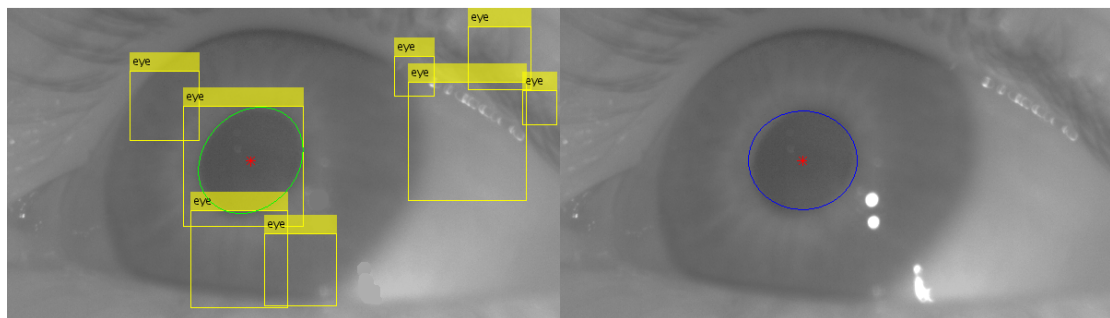
Obrázek 6.4. Porovnání kaskádního detektoru s refererencí EyeGaze SDK, negativní příklad. Vlevo Kaskádní detektor, žlutě nalezené oblasti detektorem, zeleně finální zornice. Vpravo referenční detekce modře. Středý detekovaných zornic červeně.

Na snímku 6.4 můžeme naopak pozorovat špatnou detekci. Zde jsou detekovány sice části kruhových elementů, v žádné oblasti se však nenachází celá pupila a proto dojde ke špatnému výběru oblasti příslušející zornici.

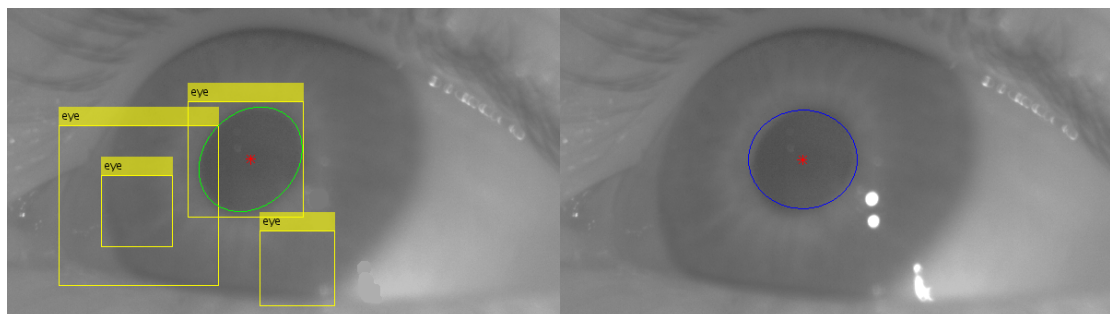
Odlíšné chování detektorů s různým vzorem je potom na obrázcích 6.5, kde jsou provedeny detekce na stejném snímku. Je zde patrné, že přestože všechny metody detekovaly správně, metoda která má největší úspěšnost (využití LBP) generuje méně, ale kvalitních kandidátů. Další dvě generují mnohem více falešných detekcí, proto mohou být jejich výsledky horší.



HOG



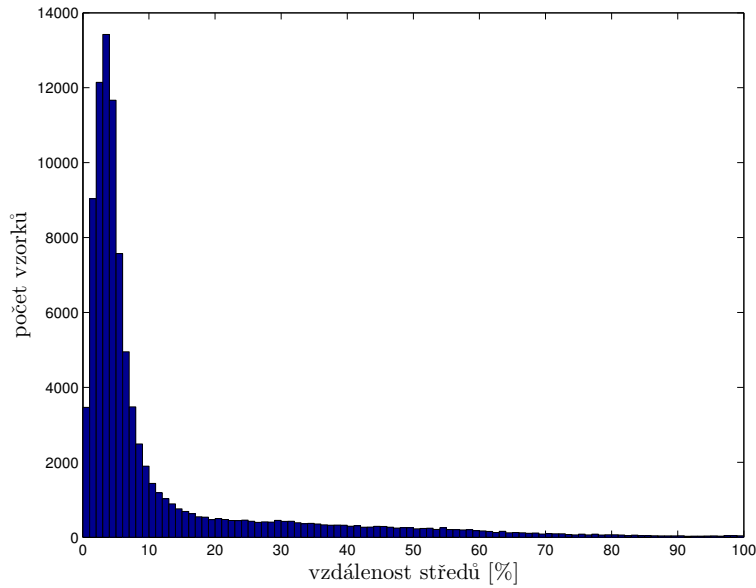
Haar



LBP

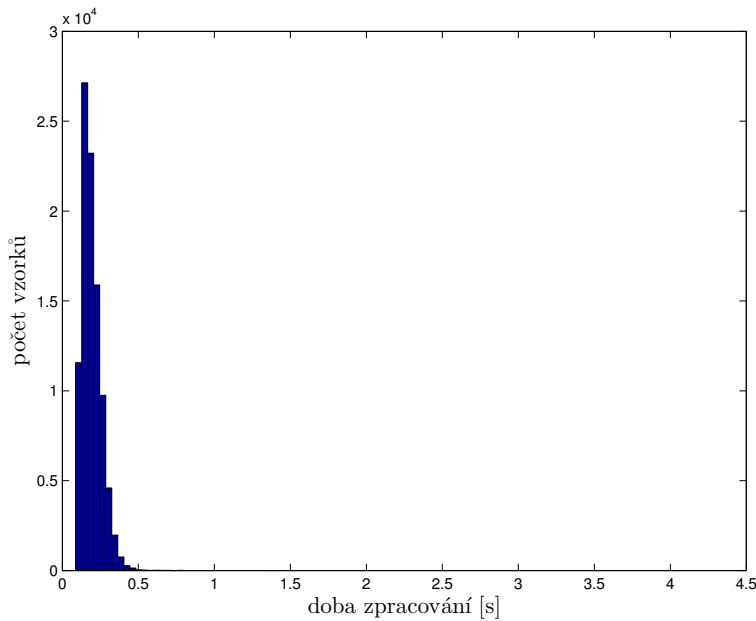
Obrázek 6.5. Porovnání detekcí metod na stejném snímku.

Při porovnání s referenční metodou Eye Gaze SDK, jsme pro porovnání s implementací s nejuspěšnějším vzorem (LBP) dosáhli úspěšnosti 98,44%. Úspěšné znamená, že střed nalezené elipsy je v menší vzdálenosti od středu referenční elipsy, než je velikost referenční zornice. Přesnost této vzdálenosti můžeme zhodnotit na histogramu 6.6, kde průměrná hodnota vzdálenosti je 11,50 % velikosti zornice. Porovnání dosažených parametrů jednotlivých metod je v tabulce 6.1 (porovnání parametrů všech metod v grafech je přehledně v příloze B).



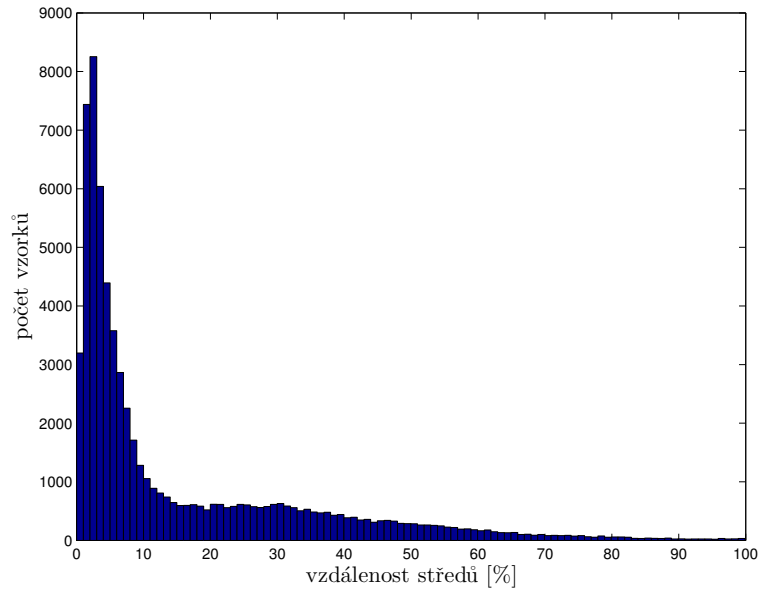
Obrázek 6.6. Histogram vzdáleností poloh středů (v procentech velikosti zornice) vzhledem k referenčnímu zpracování pro detektor s LBP.

Pozitivní na této metodě je i krátká doba zpracování. Doba zpracování je zřejmá z histogramu 6.7, kdy průměrná doba zpracování je 0,19 s.



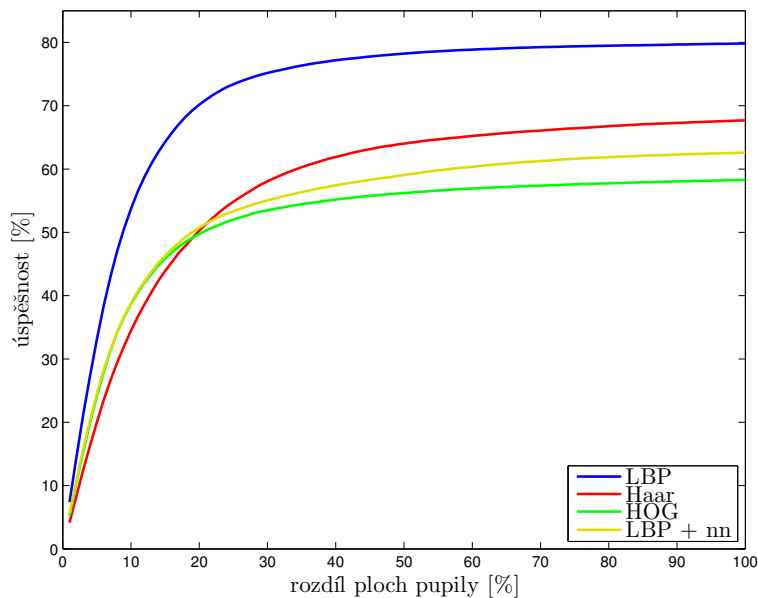
Obrázek 6.7. Histogram četností délky zpracování detektorem s LBP.

Další možnost vylepšení měla spočívat ve volbě kandidátů na zornici s využitím naučené neuronové sítě. Využili jsme síť popsanou v části 4.4. Neuronovou síť jsme aplikovali na detektor se vzorem s nejlepšími dosavadními výsledky, tedy lokálním binárním vzorem. Předpokládané zlepšení se projevilo v mírném nárůstu úspěšnosti, přesnost však není lepší než v případě pouhé volby na základě jasu. Úspěšné detekce jsme dosáhli v 99,08 % případech, vzdálenost středů od reference potom můžeme pozorovat v grafu 6.8, kde průměrná hodnota je 15,68 %.



Obrázek 6.8. Histogram vzdáleností poloh středů (v procentech velikosti zornice) vzhledem k referenčnímu zpracování pro detektor s LBP a klasifikací neuronovou sítí.

V tabulce 6.1 můžeme porovnat vliv využití jednotlivých vzorů na hodnocení výsledné detekce. Je zde navíc řádek hodnot, kolik výsledků padne do tolerance 20 % velikosti zornice, který umožňuje určit kvalitu detekce pro eyetracking. Dalším faktorem je zde skutečnost, že referenční detekce nemusí být naprosto přesná, jak je například vidět na obrázku 6.3. Pro pupilometrii je též důležitá plocha detekované zornice. Na grafu 6.9 můžeme proto vidět porovnání tolerance velikosti plochy v závislosti na úspěšnosti (průměrný rozdíl středů menší než 20 % velikosti zornice).



Obrázek 6.9. Graf zobrazuje % úspěšnosti detekce zornice (maximální povolená vzdálenost středů je 20 % velikosti) v závislosti na povoleném rozdílu velikosti plochy opět v %. Křivky pro jednotlivé detektory jsou pojmenovány podle jejich vzoru, speciálně LBP + nn je detektor s využitím klasifikace neuronovou sítí.

Pomalý nárůst křivek v grafu 6.9 mají za následek pravděpodobně případy jako na obrázku 6.10. Zde dojde ke správné detekci oblasti zornice, fitování elipsy však



Obrázek 6.10. Špatně vepsaná elipsa. (Porovnání kaskádního detektoru s referencí Eye-Gaze SDK. Vlevo Kaskádní detektor, žlutě nalezené oblasti detektorem, zeleně finální zornice. Vpravo referenční detekce modře. Středů detekovaných zornic červeně.)

	HOG	Haar	LBP	LBP + nn
Správná detekce	98,30 %	90,52 %	98,44 %	99,08 %
Průměrný rozdíl středů	18,07 %	15,81 %	11,50 %	15,68 %
Rozdíl středů menší než 20 %	63,60 %	70,66 %	82,03 %	70,55 %
Délka detekce	0,44 s	0,80 s	0,19 s	0,22 s

Tabulka 6.1. Porovnání metod detekce různými vzory (označení sloupců podle vzoru uží-tého v detektoru, speciálně LBP + nn je detektor s využitím klasifikace neuronovou sítí).

neproběhne správně. To způsobí větší vzdálenost středů a zároveň velký rozdíl v ploše detekované a předpokládané zornice.

Metoda využití kaskádního detektoru se zdá být podle tabulky výsledků 6.1 velmi úspěšná. Při využití všech vzorů dostaneme vysoké počty detekovaných zornic, nej-lépe však dopadlo využití lokálního binárního vzoru. Při porovnání s ostatními je jeho detekce nejrychlejší. Také průměrný rozdíl středů detekované a originální elipsy je nejmenší. Detekce s využitím klasifikace neuronovou sítí měla sice více odhalených zornic, ale její přesnost není tak vysoká, proto za lepší považujeme výběr podle jasů. Nej-lépe vychází i rozdíl středů s tolerancí 20 % velikosti zornice, který je splněn v 82,03 %. Při porovnání času s referenční detekcí, která trvá průměrně 0,10 sekundy je také po-měrně dobrá, neboť naše implementace je v prostředí MATLAB, ale referenční řešení je v jazyce C (tedy napsáno na nižší úrovni - rychlejší).

Kapitola 7

Metoda založená na segmentaci a následné klasifikaci

V důsledku velké náročnosti výpočtu hledání hran a následné detekci kružnice za pomoci Houghovy transformace vedlo k nalezení metody, která by kroky výpočetně zjednodušila, případně spojila. Z tohoto důvodu jsme vyzkoušeli přístup segmentace snímku na významné oblasti, na jejichž hrany použijeme kritérium podobnosti s kružnicí případně elipsou. Přístup segmentace je též často využíván v různých modifikacích 2.2 k řešení problému detekce zornice.

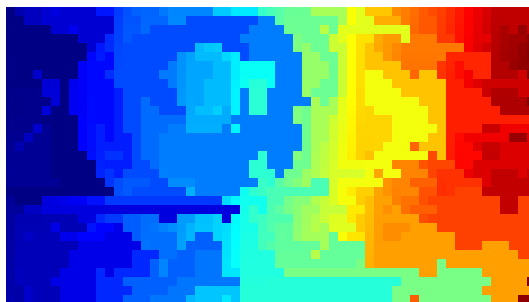
Pro segmentaci snímku na oddělené oblasti se nabízejí metoda mean-shift a k-means které rozdělí snímek na oblasti s podobnou barvou. Hlavním rozdílem je že k-means má přesně daný počet oblastí na které se snímek má rozdělit, přičemž mean-shift užívá pouze parametry omezující míru rozdílů mezi jednotlivými oblastmi. Proto jsme se rozhodli využít metody mean-shift, která je podrobněji popsána v části 3.3.2.

7.1 Mean-shift segmentace

Vzhledem k vysoké výpočetní náročnosti se segmentace provádí na převzorkovaném obrázku na nižší rozlišení. Po zpracování snímku touto metodou získáme segmentovaný obrázek, pro původní rozlišení $338 \times 600 \text{ px}$ převzorkovaném na desetinovou velikost, například obrázek 7.1 a výsledek segmentace potom 7.2.



Obrázek 7.1. Zmenšený obrázek pro segmentace.

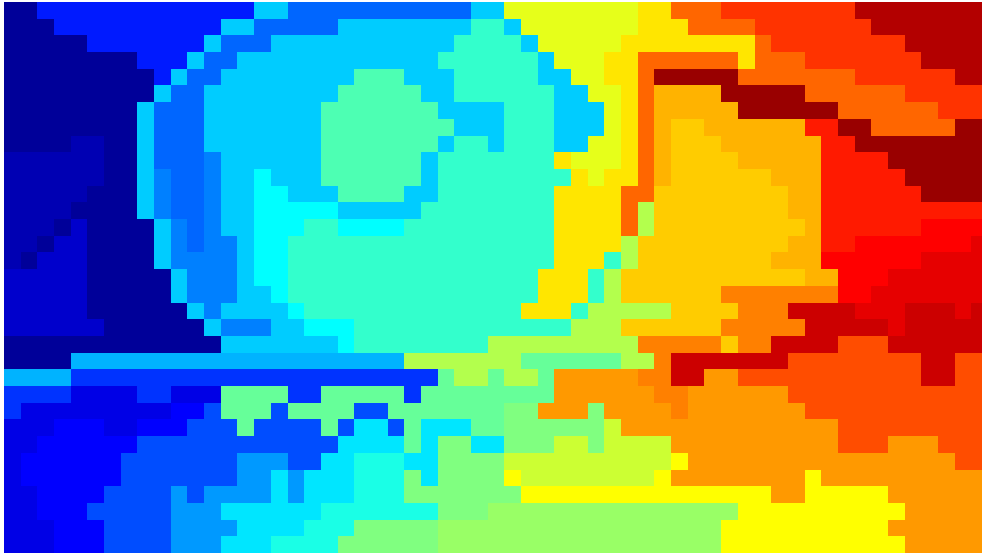


Obrázek 7.2. Výsledek segmentace.

Zpracovávané obrázky mají tři složky barvy (červená, zelená, modrá) obraz je však pouze v odstínech šedi. Proto pro snížení výpočetní náročnosti převedeme obrázek na šedotónový, pouze jedna složka jasů. Výstupy segmentace se v zásadě nemění, dojde však k úspoře výpočetního času, neboť prostor pro segmentaci se sníží o dvě souřadnice.

7.2 Další zpracování

Dalším krokem je odstranění příliš malých segmentů. V našem případě jsme zvolili velikost segmentu minimálně 20 pixelů. Odstraněním menších ploch získáme obrázek 7.3.

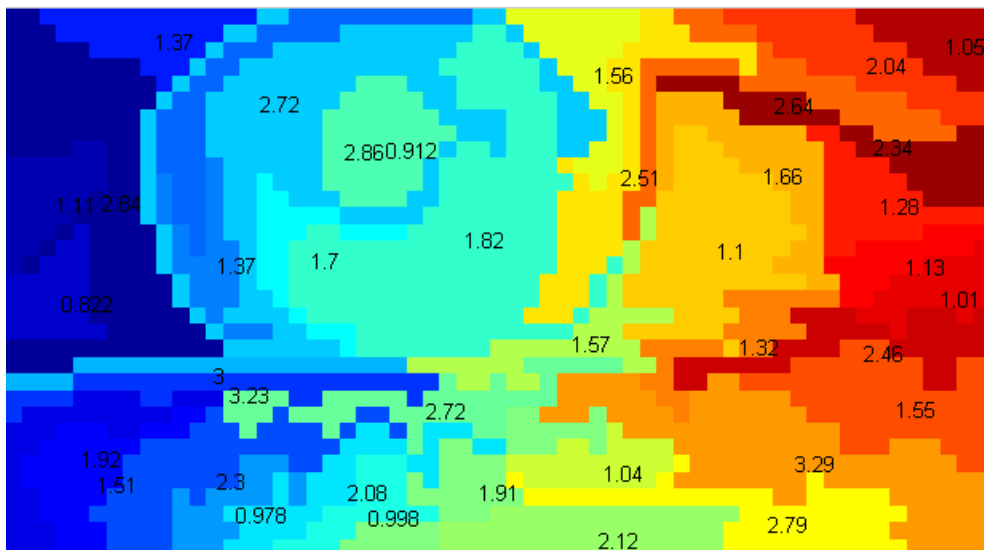


Obrázek 7.3. Segmentace s minimální velikostí ploch.

V takto rozděleném obrázku je třeba nalézt objekty, které jsou nejvíce kruhové. Předpokládáme totiž, že zornice a duhovka jsou nejkruhovější objekty.

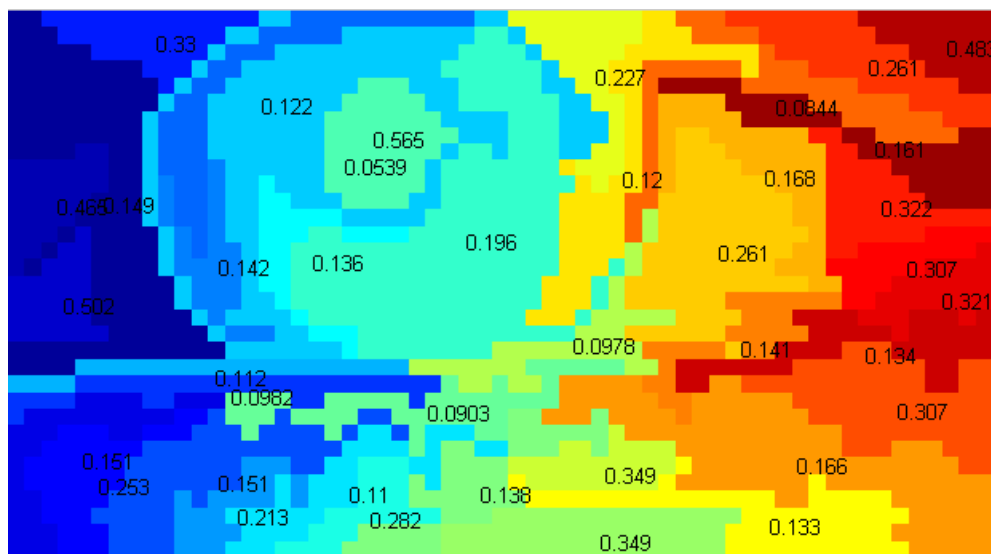
7.2.1 Míra kruhovitosti

Při využití kritéria popsaného 3.5.1. Získáme pro segmentovaný snímek 7.4, kde jsou hodnoty kritéria vypsány vždy u příslušného středu objektu. Nejlepší kruh je segment s nejnižší hodnotou kritéria (hledáme tedy minimum).



Obrázek 7.4. Segmentace s hodnotami kritéria kruhovitosti (kritérium je napsáno u těžiště segmentu).

Ukázka hodnot využití jiného kritéria 3.5.2, je na obrázku 7.5, kde jsou hodnoty kritéria vypsány vždy u příslušného středu objektu. Nejlepší kruh je potom segment, který má hodnotu nejbližší 1 (hledáme tedy maximum).



Obrázek 7.5. Segmentace s hodnotami kritéria kruhovitosti (kritérium je napsáno u těžiště segmentu).

Při detekci kruhovitosti jsou dále vyplňovány díry v objektech, aby se kruhy detekovaly správně, v případě že oblasti nejsou jednoduše souvislé, kritéria nevrací správné výsledky. Detekci jsme zjednodušili pouze na hledání kružnice, přestože zornice může být i elipsa, neboť poloosy elipsy jsou téměř stejné. Tento jev můžeme demonstrovat na podobnosti histogramů 4.5 a 4.6. Dalším důvodem je též, že parametry jsou částečně zdeformovány zmenšením obrázku.

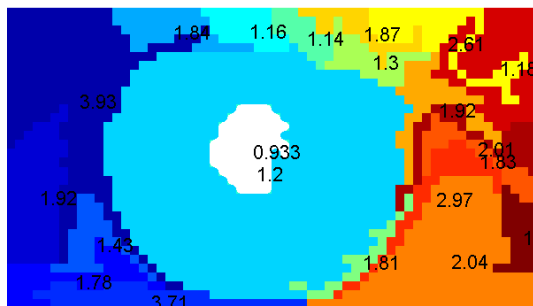
Detekce kruhových regionů bývá obvykle poměrně úspěšná. Osvědčilo se především kritérium s kontrolou odchylek od ideálního poloměru. Kritérium, kde se využívá poměru plochy a obvodu objektu se příliš neosvědčilo. Pravděpodobně vlivem malých nedokonalostí rovnoměrnosti obvodu, které příliš nezmění plochu, ale obvod mnohonásobně naroste. Především tento vliv přisuzujeme výrazným odrazům, které jsme eliminovali průměrem z okolí, segmentace je však často oddělí mimo zornici, přestože jde o oblast pupily.

Pro dosažení lepších výsledků je třeba zvýšit citlivost na barevné změny a následně provést spojování sousedních segmentů v případě, že dojde ke zlepšení kritéria kruhovitosti. V naší konkrétní implementaci jsme jako kompromis výpočetní náročnosti a kvality výstupu volili spojování maximálně 4 segmentů. Jako předzpracování jsme dále užili průměrování filtrem o velikosti 10 px . V metodě mean-shift se jako citlivost na polohu využívá koeficient $1/10$ citlivost na barvu je potom mnohem větší a to $1/2$. Zbývá vyřešit problém nalezení zornice mezi kružnicemi. Předpokládáme, že se jedná o menší ze dvou soustředných kružnic, které jsou tvořeny zornicí a duhovkou. Hledání dvou soustředných nejvíce kruhovitých objektů se však příliš neosvědčilo, proto bylo využito neuronové sítě, postup je podrobněji popsán v následující části.

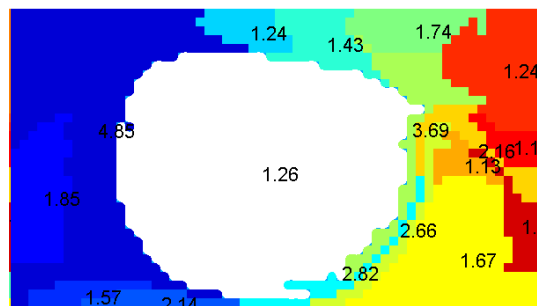
7.2.2 Volba segmentu příslušejícího zornici

Jestliže jsme rozdělili obrázek, ve kterém chceme detekovat zornici na několik oblastí, přičemž předpokládáme, že nějaký z objektů je hledaná zornice. Zornice nemusí být vždy pomocí kritéria kruhovitosti přesně odhalena. Chtěli bychom však nějakým způsobem klasifikovat, s jakou pravděpodobností je určitý segment zornicí. Proto je třeba zjistit výstupní charakteristiky měření, ze správně detekované polohy zornice. Tento krok je podrobněji popsán v části 4.4.

Pomocí neuronové sítě tak odhalíme velkou část zornic například 7.6. Někdy však nedojde ke správné segmentaci a nalezneme místo zornice duhovku 7.7, proto je třeba provést další zpracování na výřezu této části. Naopak i v případě, kdy zornici odhalíme, hrana není příliš přesná, neboť detekce proběhla na zmenšeném obrázku a tedy máme menší rozlišení detekované zornice.

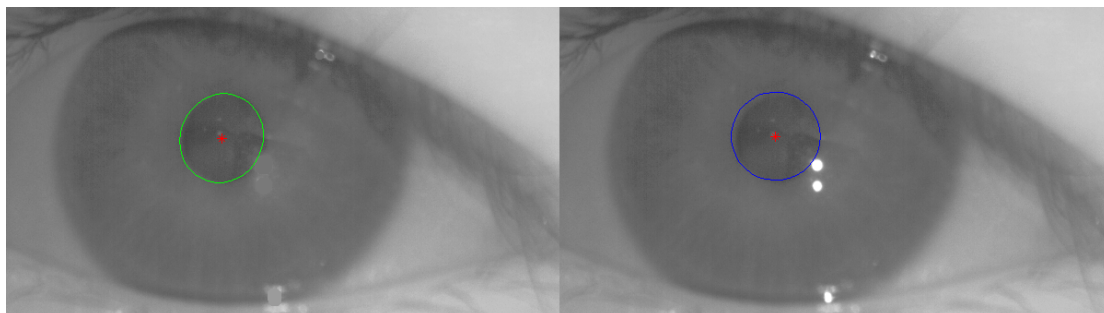


Obrázek 7.6. Správně detekovaná zornice (bílá oblast).



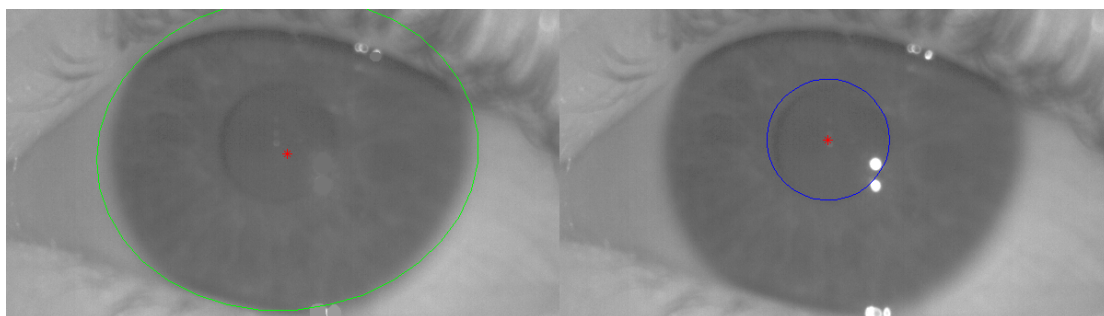
Obrázek 7.7. Špatně detekovaná zornice (zornice má být bílá oblast, zde jde evidentně o duhovku).

Následuje tedy podobný postup jako v případě využití metody kaskádního detektoru. Na oblast, ve které předpokládáme, že se nachází zornice (nyní již v původním obrázku), ať se již jedná o detekci zornice, či celé duhovky užijeme metodu k-means. Tato metoda rozdělí oblast na dva shluky a do tmavší oblasti vepíšeme elipsu, která je naší hledanou zornicí.



Obrázek 7.8. Ukázka detekce zornice. (Porovnání výstupu segmentace s referencí Eye-Gaze SDK. Vlevo zeleně detekovaná zornice. Vpravo referenční detekce modře. Střed detekovaných zornic červeně.)

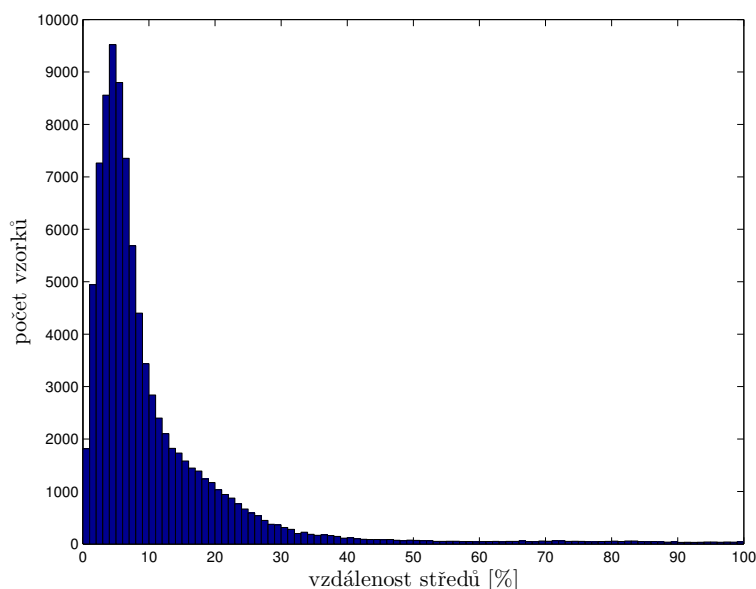
Obvykle se podaří zornici odhalit, příklad na obrázku 7.8. Ani v případě, kdy dojde k nalezení duhovky nemusí být střed nutně příliš odchylen, důkaz na obrázku 7.9. Tento případ však není možné využít k pupilometrii, kdy plocha rozhodně neodpovídá zornici.



Obrázek 7.9. Ukázka detekce duhovky místo zornice. (Porovnání výstupu segmentace s referencí EyeGaze SDK. Vlevo zeleně detekovaná zornice. Vpravo referenční detekce modře. Středů detekovaných zornic červeně.)

Jinou metodou výběru segmentů může být přímá aplikace detekce elipsy jako zornice mezi nalezenými segmenty. Do každého nalezeného segmentu místo výpočtu kritéria kruhovitosti vepíšeme elipsu a využijeme naučené neuronové sítě na hledání zornic jako elipsy popsaného v části 4.4. Tuto metodu aplikujeme přímo na nalezené segmenty, které nyní slučujeme podle toho zda se zvýší pravděpodobnost určená neuronovou sítí. U této metody očekáváme nižší přesnost vzhledem k detekci zornice na zmenšeném obrázku mezi segmenty.

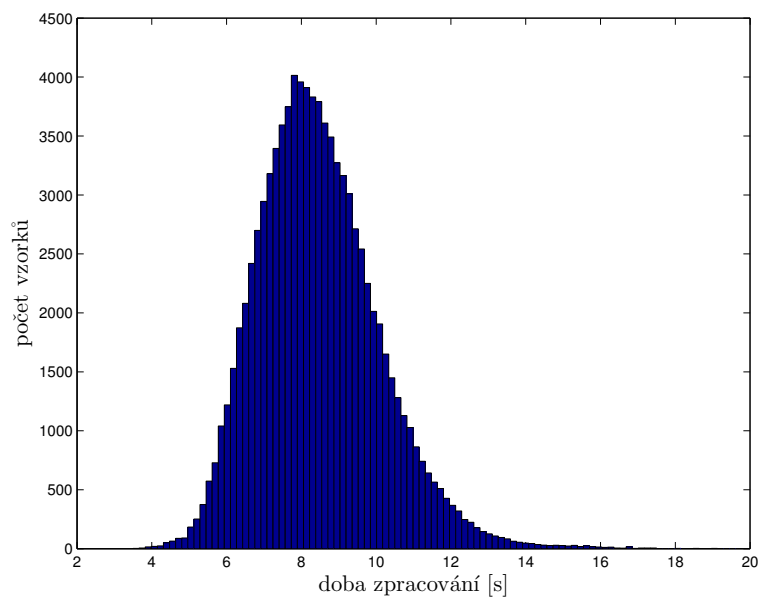
Lépe ze dvou popsaných metod dopadla detekce s využitím hledání kružnic. Porovnání metod v tabulce 7.1. Úspěšnější z metod odhalila zornici v 98,61 % případů s průměrnou vzdáleností středů 10,85 % (podrobněji vzdálenosti středů dokumentuje histogram 7.10).



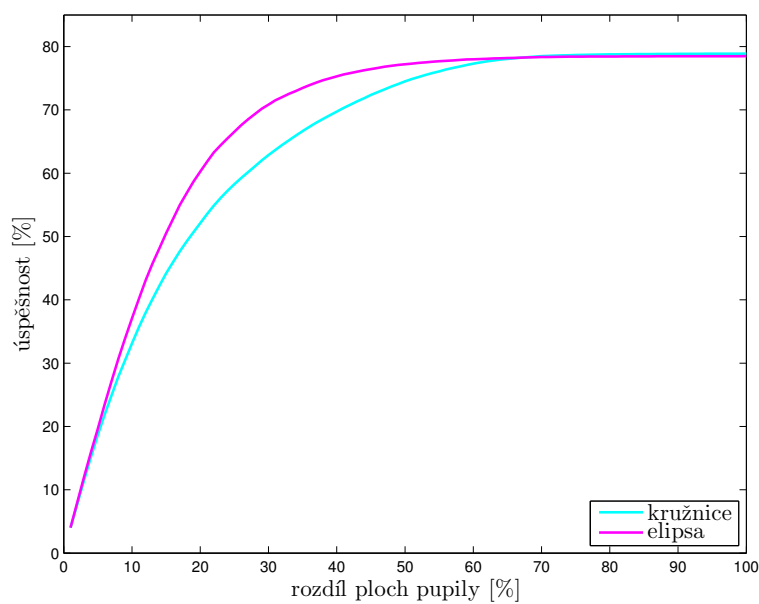
Obrázek 7.10. Histogram vzdáleností poloh středů (v procentech velikosti zornice) vzhledem k referenčnímu zpracování pro segmentační metodu s detekcí kružnice.

Nevýhodou segmentační metody je doba zpracování, která je u výhodnější implementace průměrně 8,45 s (podrobněji opět čas znázorňuje histogram 7.11).

Dalším hodnotícím kritériem kvality detekce jsme volili rozdíl ploch. Konkrétně pro detekce, které mají vzdálenost středu od reference o méně než 20 % velikosti zornice hodnotíme rozdíl plochy v procentech z plochy zornice. Tuto závislost můžeme pozorovat v grafu 7.12. Z tohoto grafu je zřejmé, že detekce elipsy má větší strmou nárůstu křivky a proto detekce s ohledem na plochu je přesnější v tomto případě. Tato metoda je však ze všech dosud testovaných časově nejnáročnější, neboť zmíněné slučování segmentů a určování pravděpodobnosti každého z nich je výpočetně náročné. Proto za lepší metodu z hlediska použití považujeme využití detekce kružnice mezi segmenty.



Obrázek 7.11. Histogram četností délky zpracování segmentační metodou s detekcí kružnice.



Obrázek 7.12. Graf zobrazuje % úspěšnosti detekce zornice (maximální povolená vzdálenost středů je 20 % velikosti) v závislosti na povoleném rozdílu velikosti plochy opět v %.

Porovnání využití detekce kružnice (kružnice) a detekce elipsy (elipsa).

Na závěr tedy srovnání parametrů obou testovaných metod využívajících segmentaci vzhledem k referenčnímu řešení je v tabulce 7.1.

	Segmentace kružnice	Segmentace elipsa	EyeGaze SDK
Správná detekce	98,61 %	96,43 %	100,00 %
Průměrný rozdíl středů	10,85 %	11,54 %	–
Rozdíl středů menší než 20 %	86,00 %	85,31 %	–
Délka detekce	8,45 s	26,43 s	0,10 s

Tabulka 7.1. Porovnání metod detekce segmentací. Segmentace - kružnice, využívá kritérium kruhovitosti pro nalezení potenciálního objektu a následně pomocí k-means hledá finální tvar. Segmentace - elipsa, hledá mezi segmenty na základě naučené neuronové sítě nejlepší zornici.

Kapitola 8

Závěr

Práce obsahuje popis některých metod využívaných k automatické detekci pupily v obraze. Následně jsou demonstrovány tři metody různého přístupu detekce a jejich implementace s několika modifikacemi. Implementace metod pro detekci by měla být za pomoci nástroje Matlab Coder převeditelná na funkce v jazyce C. Funkce použité z Matlab toolboxů podporují podle dokumentace převod tímto nástrojem včetně využití natrénovaných klasifikátorů a detektorů.

Předchozí tři kapitoly popisují testované metody detekce zornice. Nyní následuje závěrečné zhodnocení úspěšnosti jednotlivých algoritmů a jejich charakteristik včetně porovnání s referenční metodou EyeGaze SDK.

První metoda se zabývá nalezením zornice pomocí detekce hran a následného hledání kruhovitěho objektu, v obecném případě elipsy. Tato metoda podle prvních předpokladů měla být výhodná hlavně z důvodu, že není třeba rozsáhlých vstupních dat pro učící algoritmy, ale měla by být schopna přímé detekce.

První metoda však na zdrojových vstupních datech nevykazovala požadované vlastnosti. Implementace detekce hran podle článku [27] detekovala požadované hrany pouze na snímcích s vysokým kontrastem, problémem byly též rozmazané snímky. Metodu detekce hran jsme proto upravili o využití rotačního filtru pro detekci hran ve více směrech, v ideálním případě by však bylo možné dosáhnout pouze detekce ve 47,06% případů. Metoda tedy není dostatečně robustní.

Další metodou je segmentace, která by podle prvotních předpokladů též neměla potřebovat vstupní data pro učení. Předpokladem bylo nalezení kruhovitých segmentů, kde bude jasně vystupovat oblast zornice. Tento předpoklad se úplně nepotvrdil a bylo třeba využít učících algoritmů pro výběr správné oblasti zornice. Nejlepší se tedy ukázala implementace s detekcí kružnice a následně vepsání elipsy do kontur detekované oblasti. Výsledkem je správná detekce v 98,61% případů. Detekce neodhalí vždy zornici, někdy je detekována duhovka 7.9. Tato metoda má též velkou nevýhodu, kterou je doba zpracování v průměru 8,04 s, což je v porovnání s referenční metodou i další testovanou metodou velmi vysoká hodnota.

Jako nejlepší metoda se ukázalo využití kaskádního detektoru s využitím Haarovy báze, histogramu orientovaných gradientů nebo lokálního binárního vzoru. Nejlépe dopadla tato metoda s lokálním binárním vzorem a považujeme ji tak za celkově nejlepší metodu, které bylo dosaženo. Výsledné parametry dosažené touto metodou sice nedosahují přesnosti umístění středu jako metoda segmentace. Pokud však vezmeme v úvahu další parametry kvality detekce a to především plochu, kaskádní detektor je v tomto ohledu lepší. Pozitivní je i kratší doba detekce.

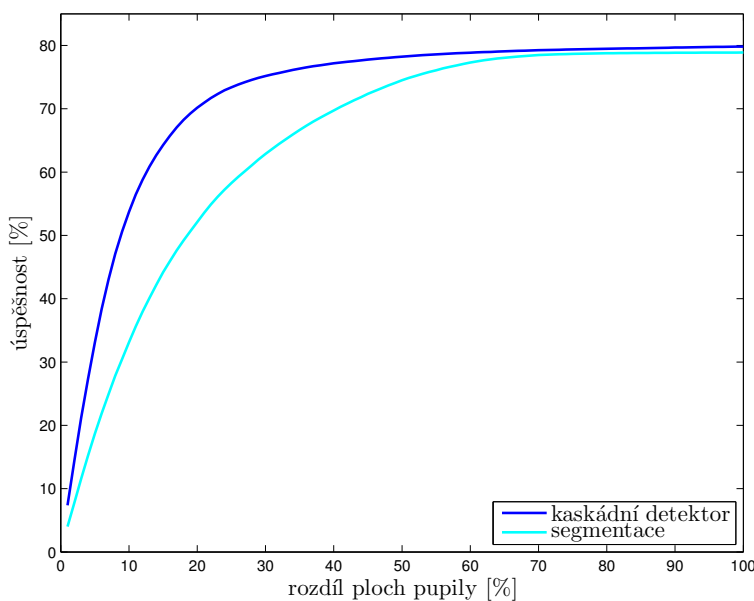
Rozdíl plochy skutečné zornice a naší detekované není takovým problémem pro eye-tracking, pokud detekujeme přibližně správně střed, problém však nastává v případě využití pro pupilometrii. Porovnání úspěšnosti metod vzhledem k rozdílu plochy detekované a referenční zornice je zobrazen v grafu 8.1. Zde vidíme srovnání dvou nejlepších metod, lepší je tedy využití metody kaskádního detektoru. I lepší z metod však vykazuje v některých případech poměrně velké rozdíly plochy. To je dáno pravděpodobně

nesprávným fitováním elipsy (ukázka na obrázku 6.10). Zlepšením fitování elipsy by tedy došlo k lepší detekci správné plochy a pravděpodobně by se snížila i odchylka středu některých detekcí.

Porovnání těchto nejúspěšnějších metod je v tabulce 8.1 (porovnání všech metod testovaných v dokumentu včetně grafů parametrů je shrnuto na jednom místě v příloze B). Jako nejúspěšnější tedy hodnotíme využití kaskádního detektoru vzhledem k nejlepším výsledkům kvality detekce. Rozdíl středu od referenčního řešení je v 82,03 % menší než 20 % velikosti zornice a nejlépe vychází i doba zpracování řádově srovnatelná s referenční metodou. Využití segmentace nevykazuje též špatné výsledky, hlavní nevýhodou této metody je však doba zpracování.

	Hranová detekce	Kaskádní detektor	Segmentace	EyeGaze SDK
Správná detekce	47,06 %	98,44%	98,61 %	100,00 %
Průměrný rozdíl středů	—	11,50 %	10,85 %	—
Rozdíl středů menší než 20 %	—	82,03 %	86,00 %	—
Délka detekce	14,61 s	0,19 s	8,46 s	0,10 s

Tabulka 8.1. Porovnání nejlepších modifikací jednotlivých metod detekce. (Kaskádní detektor s LBP a segmentace s detekcí kružnice.)



Obrázek 8.1. Graf zobrazuje % úspěšnosti detekce zornice (maximální povolená vzdálenost středů je 20 % velikosti) v závislosti na povoleném rozdílu velikosti plochy opět v %. Porovnání využití metody kaskádního detektoru s LBP a segmentace s detekcí kružnice.

Detekce zornice popisovaná v této práci se zabývá odděleně jednotlivými snímky. Tento postup může způsobit značné rozdíly výsledků vzhledem k referenční metodě. Referenční metoda provádí filtrování detekcí v čase, což je umožněno kontinuálním záznamem videa se snímkovací frekvencí 150 snímků za sekundu. V takovém případě můžeme předpokládat, že parametry zornice po sobě následujících snímků se příliš nezmění a pomocí tohoto kritéria penalizovat určité kandidáty na detekci.

Dalším krokem ve vývoji této automatické detekce by tedy měla být filtrace v čase, která by mohla značně pozitivně ovlivnit výsledky detekce. V případě tohoto postupu by mohlo dojít ke kombinaci kaskádního detektoru a segmentace, která by se prováděla pouze na určitém okolí předchozí zornice a nebyla by tolik časově náročná. Tímto krokem by též mohlo dojít ke zlepšení vepsání elipsy, případně změnit současnou detekci elipsy za dokonalejší způsob určení elipsy.

Literatura

- [1] *Segmentace*.
https://cw.fel.cvut.cz/wiki/courses/a4m33dzo/mat/2014_segmentace.
- [2] *Iris Detection matlab*.
<http://matlabsproj.blogspot.cz/2012/06/iris-detection-matlab.html>.
- [3] ALONSO-FERNANDEZ, Fernando a Josef BIGUN. *Iris Pupil Detection by Structure Tensor Analysis*.
<http://www2.hh.se/staff/josef/public/publications/fernandez11linkoping.pdf>.
- [4] ASADIFARD, Mansour a Jamshid SHANBEZADEH. *Automatic Adaptive Center of Pupil Detection Using Face Detection and CDF Analysis*.
http://www.iaeng.org/publication/IMECS2010/IMECS2010_pp130-133.pdf.
- [5] DALAL, N. a B. TRIGGS. Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005. s. 886-893 vol. 1. ISSN 1063-6919. Dostupné na DOI 10.1109/CVPR.2005.177.
- [6] DOUBEK, Petr. *Mean-Shift segmentace*.
<http://cmp.felk.cvut.cz/cmp/courses/ZS1/Cviceni/cv4/meanshift.pdf>.
- [7] DUCHOWSKI, Andrew. *Eye Tracking Methodology, Theory and Practice*. Second Edition vyd. Springer-Verlag London, 2007. ISBN 978-1-84628-608-7.
- [8] FISHER, R. B., K. DAWSON-HOWE, A. FITZGIBBON, C. ROBERTSON a E. TRUCCO. *Dictionary of Computer Vision and Image Processing*. John Wiley & Sons, Ltd, 2006. ISBN 9780470016305. Dostupné na DOI 10.1002/0470016302.
- [9] GREGOR, Michal. *Detektory objektů v obraze a jejich realizace*.
https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=9096.
- [10] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. 3 vyd. Česká technika - nakladatelství ČVUT, 2009. ISBN 978-80-01-04442-1.
- [11] JAVADI, Amir-Homayoun, Zahra HAKIMI, Morteza BARATI, Vincent WALSH a Lili TCHEANG. *SET: a pupil detection method using sinusoidal approximation*. Dostupné na DOI 10.3389/fneng.2015.00004.
- [12] KULKARNI, Surashree a Sagar GALA. *A Simple Algorithm for Eye Detection and Cursor Control*.
<http://www.ijcta.com/documents/volumes/vol4issue6/ijcta2013040602.pdf>.
- [13] MATHWORKS. *Train a Cascade Object Detector*.
<http://www.mathworks.com/help/vision/ug/train-a-cascade-object-detector.html>.
- [14] MATHWORKS. *Classify Patterns with a Neural Network*.
<http://www.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html>.
- [15] MATHWORKS. *regionprops*.
<http://www.mathworks.com/help/images/ref/regionprops.html>.

- [16] MEDICTON GROUP S.R.O. *I4Tracking®*.
<http://www.i4tracking.cz/>.
- [17] OJALA, T., M. PIETIKAINEN a T. MAENPAA. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Jul, 2002, ročník 24, č. 7, s. 971-987. ISSN 0162-8828. Dostupné na DOI 10.1109/TPAMI.2002.1017623.
- [18] QIN, H., X. WANG, M. LIANG a W. YAN. A novel pupil detection algorithm for infrared eye image. In: *Signal Processing, Communication and Computing (ICSPCC), 2013 IEEE International Conference on*. 2013. s. 1-5. Dostupné na DOI 10.1109/ICSPCC.2013.6664050.
- [19] ROSELIN, Vanaja, E. CHIRCHI a L. M. WAGHMARE. *Feature Extraction and Pupil Detection Algorithm Used for Iris Biometric Authentication System*.
http://www.sersc.org/journals/IJSIP/vol6_no6/14.pdf.
- [20] SIROIS, Sylvain a Julie BRISSON. Pupillometry. *WIREs Cogn Sci 2014*. Nov, 2014, ročník 5, č. 6, s. 679-692. Dostupné na DOI 10.1002/wcs.1323.
- [21] SOLTANY, Milad, Saeid TOOSI ZADEH a Hamid-Reza POURREZA. *Fast and Accurate Pupil Positioning Algorithm using Circular Hough Transform and Gray Projection*.
<http://www.ipcsit.com/vol5/102-ICCCM2011-C094.pdf>.
- [22] SONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image Processing, Analysis, and Machine Vision*. 3 vyd. Thomson Learning, 2008. ISBN 978-0-495-08252-1.
- [23] SVOBODA, Tomáš, Jan KYBIC a Václav HLAVÁČ. *Image Processing, Analysis, and Machine Vision: A MATLAB Companion*.
<http://visionbook.felk.cvut.cz/>.
- [24] *Tobiipro*.
<http://www.tobiipro.com/>.
- [25] TOPAL, Cihan a Cuneyt AKINLAR. Edge Drawing: A combined real-time edge and segment detector. *Journal of Visual Communication and Image Representation*. 2012, ročník 23, č. 6, s. 862 - 872. ISSN 1047-3203. Dostupné na DOI <http://dx.doi.org/10.1016/j.jvcir.2012.05.004>.
- [26] TOPAL, Cihan a Cuneyt AKINLARA. *EDPF: A Realtime Parameter-Free Edge Segment Detector with a False Detection Control*.
<http://ceng.ankadolu.edu.tr/CV/EDPF/>.
- [27] TOPAL, Cihan a Cuneyt AKINLARA. *An Adaptive Algorithm for Precise Pupil Boundary Detection Using the Entropy of Contour Gradients*.
<http://ceng.ankadolu.edu.tr/cv/eyetracking/download/PupilDetection.pdf>.
- [28] VIOLA, P. a M. JONES. Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. 2001. s. I-511-I-518 vol.1. ISSN 1063-6919. Dostupné na DOI 10.1109/CVPR.2001.990517.
- [29] VOLNÁ, Eva. *NEURONOVÉ SÍŤE 1*.
http://www1.osu.cz/~volna/Neuronove_site_skripta.pdf.
- [30] ŚWIRSKI, Lech, Andreas BULLING a Neil DODGSON. *Robust real-time pupil tracking in highly off-axis images*.
<http://www.cl.cam.ac.uk/research/rainbow/projects/pupiltracking/files/>.

Příloha A

Softwarová příloha

Algoritmy byly implementovány v prostředí MATLAB. Skripty a funkce demonstrující jednotlivé algoritmy jsou přiloženy na optickém disku. Detailní popis jednotlivých skriptů a funkcí je popsán v jejich nápovědě.

Soubor *README.txt* obsahuje popis uspořádání a formát vstupních dat a trénování učících algoritmů. Dále jen spouštěcí skripty pro jednotlivé algoritmy, kterými jsou:

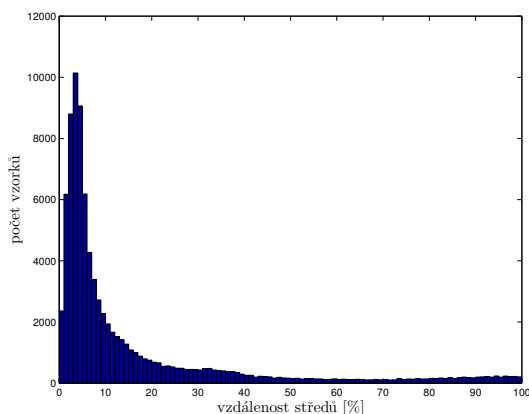
- *Hrany_paper.m* – detekce hran podle článku [27]
- *Hrany_hough.m* – detekce hran s využitím rotačního filtru a Houghovy transformace
- *Cascade_detector_col.m* – detekce s využitím kaskádního detektoru a výběr kandidáta podle barvy
- *Cascade_detector_nn.m* – detekce s využitím kaskádního detektoru a výběr kandidáta pomocí neuronové sítě
- *Segmentace_kruznice.m* – segmentace s detekcí kruhovitých objektů
- *Segmentace_elipsa.m* – segmentace s přímou detekcí elipsy

Příloha B

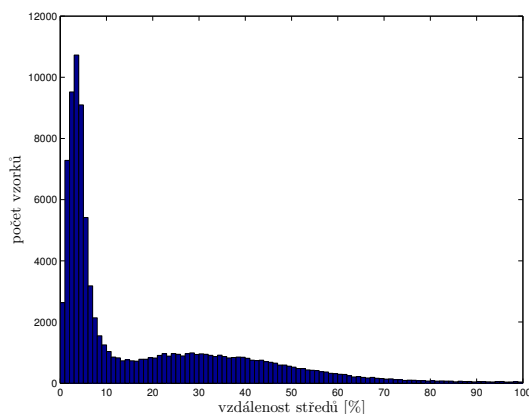
Porovnání parametrů všech metod

V této příloze jsou seskupeny histogramy a graf hodnocených parametrů všech testovaných metod pro přehledné porovnání výsledků. Následuje tabulka porovnávající všechny dosažené výsledné parametry. V popisech k jednotlivým grafům je využito následujících označení. Pro kaskádní detektor se využívá označení podle použitého vzoru detektoru (Haar, HOG, LBP), kde LBP + nn je využití neuronové sítě pro výběr kandidátů. Pro metody segmentace je potom využíváno označení kružnice pro využití výběru segmentů s hledáním kružnice a elipsa pro hledání elipsy.

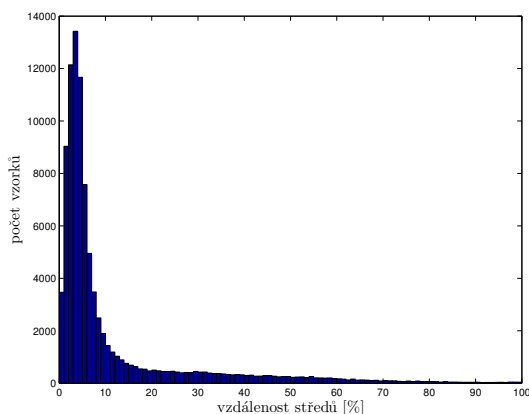
B.1 Středy



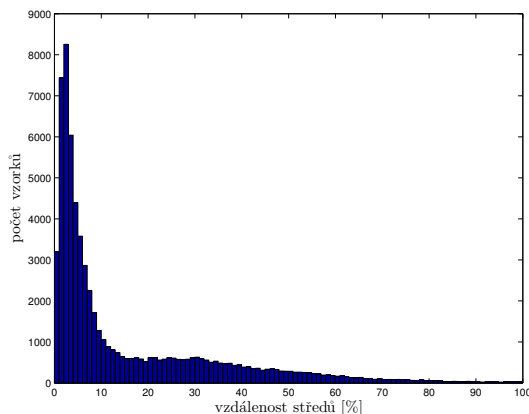
Obrázek B.1. Rozdíl středů Haar.



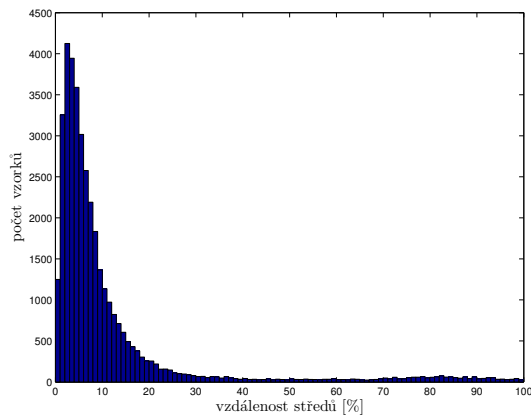
Obrázek B.2. Rozdíl středů HOG.



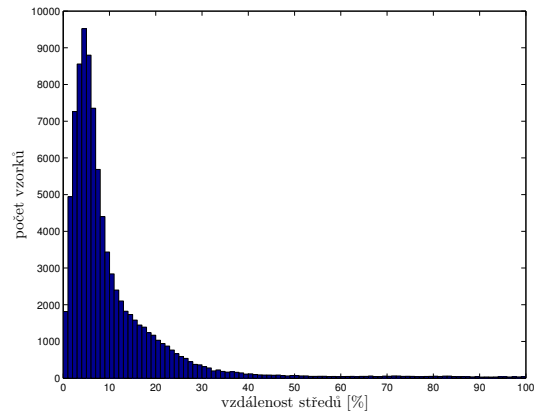
Obrázek B.3. Rozdíl středů LBP.



Obrázek B.4. Rozdíl středů LBP + nn.

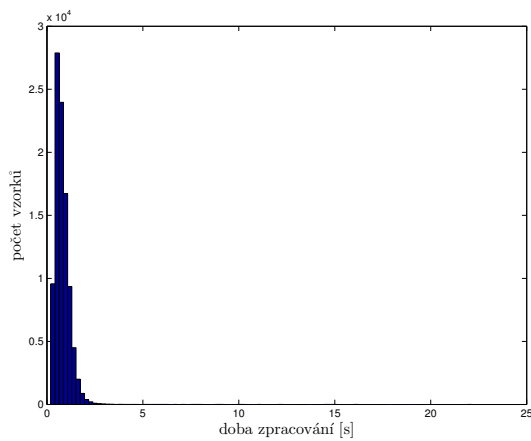


Obrázek B.5. Rozdíl středů segmentace s detekcí elipsy.

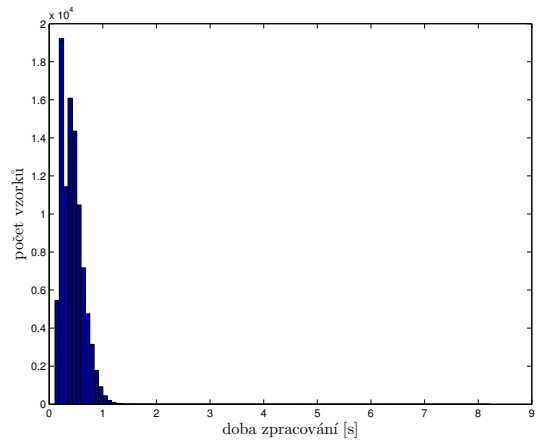


Obrázek B.6. Rozdíl středů segmentace s detekcí kružnice.

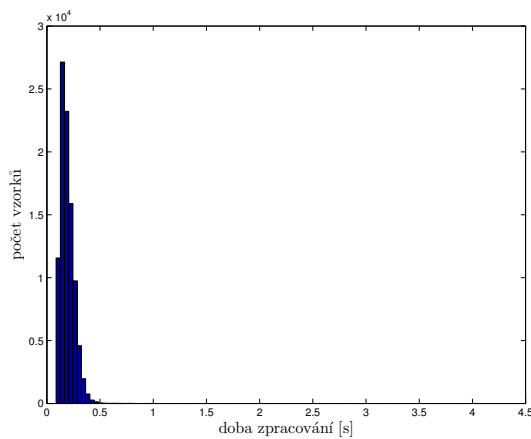
B.2 Doba zpracování



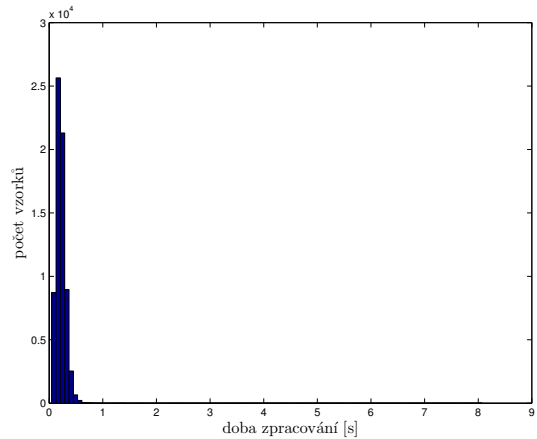
Obrázek B.7. Doba zpracování Haar.



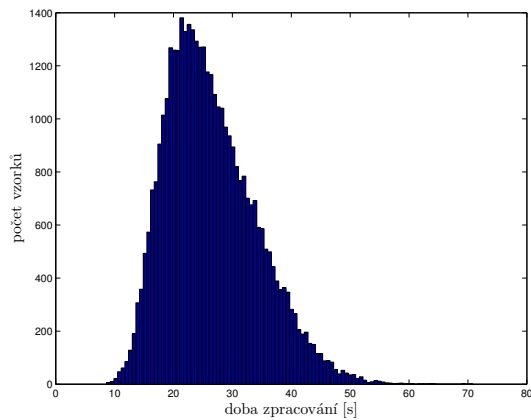
Obrázek B.8. Doba zpracování HOG.



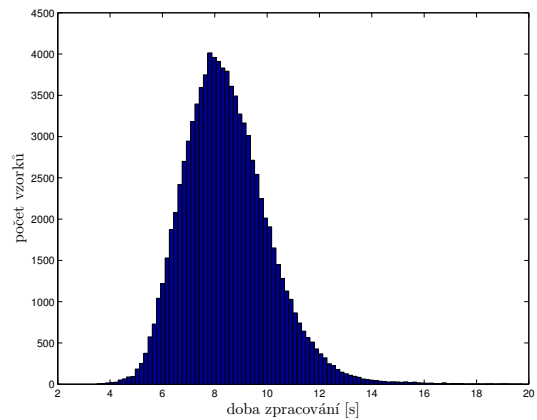
Obrázek B.9. Doba zpracování LBP.



Obrázek B.10. Doba zpracování LBP + nn.

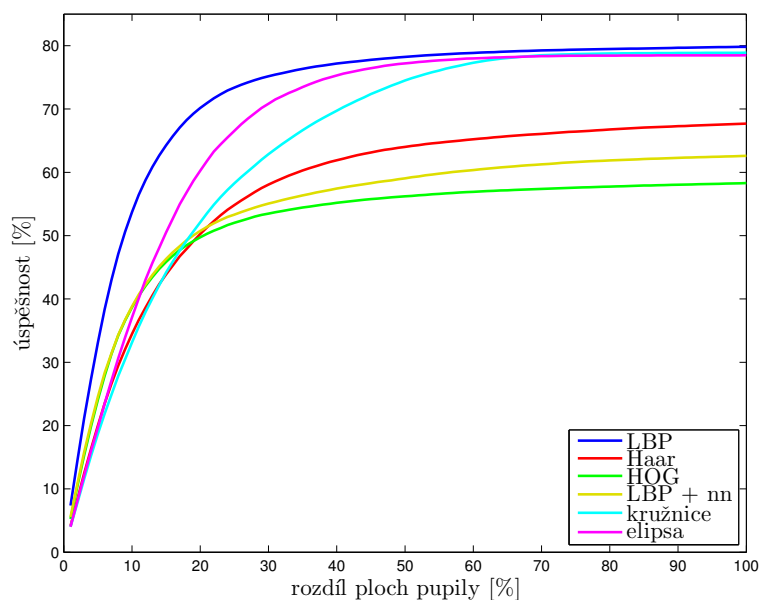


Obrázek B.11. Doba zpracování segmentace s detekcí elipsy.



Obrázek B.12. Doba zpracování segmentace s detekcí kružnice.

B.3 Kvalita detekce plochy



Obrázek B.13. Graf zobrazuje % úspěšnosti detekce zornice (maximální povolená vzdálenost středů je 20 % velikosti) v závislosti na povoleném rozdílu velikosti plochy opět v %.

B.4 Tabulka zkoumaných parametrů

	Hranová detekce	Kaskádní detektor HOG	Kaskádní detektor Haar	Kaskádní detektor LBP	Kaskádní detektor LBP + nn	Segmentace kružnice	Segmentace elipsa	EyeGaze SDK
Správná detekce	47,06 %	98,30 %	90,52 %	98,44 %	99,08 %	98,61 %	96,43 %	100,00 %
Průměrný rozdíl středů	—	18,07 %	15,81 %	11,50 %	15,68 %	10,85 %	11,54 %	—
Rozdíl středů menší než 20 %	—	63,60 %	70,66 %	82,03 %	70,55 %	86,00 %	85,31 %	—
Délka detekce	14,61 s	0,44 s	0,80 s	0,19 s	0,22 s	8,46 s	26,43 s	0,10 s

Tabulka B.1. Porovnání parametrů metod detekce.