

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Detekce očí a jejich jednotlivých částí ve videosekvencích
Detecting the Eyes and their Parts in Video Sequences

2015

Bc. Jakub Vlk

Zadání diplomové práce

Student: **Bc. Jakub Vlk**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Detekce očí a jejich jednotlivých částí ve videosekvencích**
Detecting the Eyes and their Parts in Video Sequences

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem diplomové práce je vytvořit software pro detekci očí a jednotlivých jejich částí v obrázcích získaných kamerou. Software je vytvářen s představou jeho využití v automobilu pro stanovení míry únavy řidiče. Obrázky, s nimiž bude software pracovat, jsou ve stupních šedi. Jsou pořízené za běžného nebo případně infračerveného osvětlení.

V diplomové práci proveďte následující:

1. Navrhněte vhodnou metodu nalezení oblasti očí.
2. Navrhněte vhodnou metodu detekce očních víček.
3. Navrhněte vhodnou metodu pro nalezení zornice a duhovky.
4. Navržené řešení realizujte v C/C++.
5. Řešení řádně experimentálně prověřte.

Způsob a výsledky řešení popište v textové části práce.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

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

Vedoucí diplomové práce: **doc. Dr. Ing. Eduard Sojka**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Dr. Ing. Eduard Sojka
vedoucí katedry

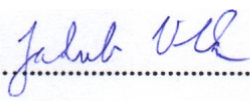


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

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

V Ostravě dne: 6.5.2015



.....

Poděkování

Rád bych velmi poděkoval Doc. Dr. Ing. Eduardu Sojkovi za odbornou pomoc a konzultaci při vytváření této diplomové práce. Dále bych chtěl poděkovat rodině a kamarádům za podporu při studiu.

Abstrakt

Tato diplomová práce se zaměřuje na navržení metod a vytvoření softwaru pro detekci očních částí v obrázcích získaných kamerou. Konkrétně se jedná o nalezení očních oblastí, detekce duhovky, detekce zornice a nalezení očních víček. Obrázky se kterými software pracuje jsou ve stupních šedi a byly pořízeny za běžného, případně infračerveného osvětlení. Motivací pro tuto práci bylo předpokládané použití detekce v systému pro sledování řidiče a míry jeho unavenosti, a prací tak přispět ke zvýšení bezpečnosti na cestách a dálnicích. V textu popisují metody, které jsou již známé, dále metody, které jsem modifikoval a i metody, které jsem sám navrhl. Výsledkem této diplomové práce je implementace detekce očních částí v jazyce C++, podrobný popis použitých technik a detailní experimentální část, ve které je vyhodnocena úspěšnost a časová náročnost na více než 1 500 obrázcích.

Klíčová slova

oblasti očí, střed oka, duhovka, zornice, oční víčka, detekce

Abstract

The thesis is focused on designing the methods and creating the software for detecting different parts of eyes in the images taken by a camera. This is done by finding the specific area of eye, which is followed by detecting the iris, locating the pupil, and identifying the eye lids. The system works with the grayscale images that can be taken using both natural and infrared light. My primary motivation for choosing this topic was the anticipated use of this detection system for monitoring the drivers and measuring their drowsiness while they are driving. This should help to increase the safety on roads and motorways. In my work, I describe the methods that are already known, as well as the methods I have modified, and the methods that were created by myself. The final practical outcome of this diploma thesis is the implementation of the methods that can recognize various parts of eye. The implementation is done in C++. The work also includes a detailed description of the techniques that were used. It also contains a thorough experimental evaluation including efficiency and time complexity, which was evaluated for over 1500 images.

Key words

area of the eyes, center of the eye, iris, pupil, eyelids, detection

Seznam použitých symbolů, zkratk a termínů

OpenCV	Open Source Computer Vision
FMM	Fast Marching Method
AdaBoost	Adaptive Boosting

Obsah

1 Úvod.....	1
1.1 Systémy pro sledování míry ospalosti řidiče.....	1
1.2 Struktura diplomové práce.....	1
2 Popis současného stavu v dané oblasti.....	3
3 Detekce obličeje, očí a jejich jednotlivých částí.....	5
3.1 Nalezení obličeje.....	5
3.1.1 Detekce obličeje pomocí kaskád slabých klasifikátorů.....	5
3.2 Detekce očních oblastí.....	11
3.3 Odebrání reflexe z očí.....	12
3.3.1 Odebrání šumu.....	13
3.3.2 Sobelův operátor.....	13
3.3.3 Binarizace obrazu.....	14
3.3.4 Odstranění reflexe světelného zdroje.....	15
3.4 Nalezení středu oka.....	16
3.4.1 Detekce středu duhovky s korekcí stínů v oční oblasti.....	17
3.4.2 Lokalizace středu oka pomocí gradientů.....	23
3.5 Nalezení duhovky.....	26
3.6 Nalezení zornice.....	28
3.7 Detekce polohy očních víček.....	29
4 Experimentální část.....	31
4.1 Metodika testu středu očí.....	32
4.1.1 Test detekce středu duhovky s korekcí stínů v oční oblasti.....	33
4.1.2 Test lokalizace středu oka pomocí gradientů.....	35
4.1.3 Porovnání metod pro detekci středu oka.....	37
4.2 Test velikosti průměru duhovky.....	39
4.3 Test detekce polohy očních víček.....	42
4.4 Časová složitost.....	44
4.5 Test na videu v reálném čase.....	45
5 Závěr.....	48
6 Literatura.....	50

Seznam obrázků

Obrázek 1 - Příklady jak mohou vypadat Haarovy vlastnosti.....	6
Obrázek 2 - Aplikace odlišných Haarových vlastností na obličej.....	7
Obrázek 3 - Znázornění kaskády. Počáteční klasifikátor může rychle eliminovat nevhodnou oblast pro obličej. Daný region se již nebude testovat, a ušetří se tak výpočetní výkon.....	8
Obrázek 4 - Vlevo hodnoty intenzity obrazu. Vpravo hodnoty integrálního obrazu.....	9
Obrázek 5 - Součet všech pixelů v obdélníku D můžeme spočítat jako $4 + 1 - (2 + 3)$	10
Obrázek 6 - Špatně detekovaný obličej.....	10
Obrázek 7 - Špatná detekce očních oblastí.....	11
Obrázek 8 - Mylně detekovaná oční oblast. Větší oční oblast na pravé straně obličeje zasahuje přes jeho polovinu. Považujeme ji tedy za nekorektní.....	12
Obrázek 9 - Vlevo derivace ve vertikálním směru. Uprostřed derivace v horizontálním směru. Vpravo výsledný gradient.....	14
Obrázek 10 - Princip metody image inpainting.....	15
Obrázek 11 - Vlevo oko s odrazem světla. Uprostřed maska s reflexí. Vpravo oko s odstraněným odrazem světla.....	16
Obrázek 12 - Vlevo obrázek před vynásobením intenzity pixelů. Vpravo obrázek po násobení intenzity.....	18
Obrázek 13 - Vlevo binární obrázek získaný pomocí binarizační Otsuovy metody. Vpravo všechny nalezené kontury společně s jejich ohraničujícími boxy.....	19
Obrázek 14 - Vlevo obrázek na vstupu. Uprostřed nalezené a očíslované kontury. Vpravo nalezený střed oka.....	20
Obrázek 15 - Vlevo binární obrázek s nalezenou elipsou a jejím středem. Vpravo obrázek oka a jeho střed.....	21
Obrázek 16 - Špatná detekce středu oka v důsledku stínu ve vnitřním rohu oční oblasti.....	21
Obrázek 17 - Binární obrázek znázorňující, zda je potřebná korekce středu oka.....	22
Obrázek 18 - Nahoře vlevo oko se špatnou detekcí středu. Nahoře uprostřed binární obrázek s viditelnou elipsou a jejím středem. Nahoře vpravo možné kontury s ohraničujícími boxy a střed s elipsou. Dole vlevo již opravený binární obrázek. Dole uprostřed finální určení středu oka.....	23
Obrázek 19 - Vlevo neořezaná oční oblast. Vpravo oblast již po ořezání.....	24
Obrázek 20 - Vlevo kruh na černém pozadí. Uprostřed velikost gradientu. Vpravo směry gradientu.....	24
Obrázek 21 - Vlevo nekorektní střed kruhu. Vpravo správný střed kruhu. Vektory d_i a g_i mají stejný směr.....	25
Obrázek 22 - Vlevo oko na vstupu. Uprostřed místo s nejpočetnějším průnikem gradientů. Vpravo oko se středem.....	26
Obrázek 23 - Vlevo obrázek oka na vstupu. Uprostřed derivace ve směru osy x a y . Vpravo derivace jen ve směru osy x	26
Obrázek 24 - Vytyčené oblasti, na kterých se zkoumají derivace ve směru osy x	27
Obrázek 25 - Vlevo vstupní obrázek oka. Uprostřed derivace ve směru osy x . Vpravo získaný obrys duhovky.....	28
Obrázek 26 - Vlevo případ, kdy nelze jednoznačně rozeznat zornici. Vpravo již zornici můžeme pozorovat.....	28
Obrázek 27 - Vlevo oko na vstupu s vynásobenou intenzitou pixelů. Uprostřed výsledek nalezení zornice na stejném obrázku. Vpravo nalezená zornice na původním obrázku oka.....	29
Obrázek 28 - Vlevo oko na vstupu. Uprostřed oko po vynásobení intenzity pixelů. Vpravo binarizovaný obraz pomocí Otsuovy metody.....	30
Obrázek 29 - Vlevo oko na vstupu. Uprostřed binarizovaný obraz. Vpravo výsledné oko se zznačenou polohou horního a spodního víčka.....	30

Obrázek 30 - Příklady tváří z BioID databáze.....	31
Obrázek 31 - Schéma očí, skutečného a předpokládaného středu a vzdáleností mezi nimi.....	32
Obrázek 32 - Obličej, u kterých byla detekce středu oka korektně.....	33
Obrázek 33 - Obličej, u kterých algoritmus mylně vyhodnotil pozice středu oka.....	34
Obrázek 34 - Přesnost a normalizovaná chyba u metody detekce středu duhovky s korekcí stínů v oční oblasti.....	35
Obrázek 35: Výběr úspěšných detekcí středů očí pomocí metody lokalizace středu oka pomocí gradientů.....	36
Obrázek 36 - Neúspěšná detekce středu oka při použití metody lokalizace středu oka pomocí gradientů. Jedná se zejména o případy, kdy osoba přivírá oči nebo je má dokonce zavřené či má brýle s výrazným odleskem světla.	36
Obrázek 37 - Přesnost a normalizovaná chyba u metody lokalizace středu oka pomocí gradientů .	37
Obrázek 38 - Metoda 1 reprezentuje postup popsáný v kapitole 3.4.1, tedy detekci středu duhovky s korekcí stínů v oční oblasti. Metoda 2 představuje lokalizaci středu oka pomocí gradientů, která byla popsána v kapitole 3.4.2.....	39
Obrázek 39 - Výběr úspěšných detekcí duhovky.....	40
Obrázek 40 - Příklady obrázků, u kterých detekce duhovky a její velikosti neproběhla úspěšně.	41
Obrázek 41 - Přesnost a normalizovaná chyba u metody detekce duhovky.....	41
Obrázek 42 - Případy úspěšné detekce polohy očních víček.....	43
Obrázek 43 - Vzorek obrázků neúspěšných detekcí očních víček.....	43
Obrázek 44 - Přesnost a normalizovaná chyba u metody detekce očních víček.....	44
Obrázek 45 - Časová složitost jednotlivých metod v milisekundách.....	45
Obrázek 46 - Příklady správných detekcí z videa v reálném čase.....	46
Obrázek 47 - Příklady nekorektních detekcí z videa v reálném čase.....	47

1 Úvod

Tato diplomová práce se zaměřuje na navržení metod a vytvoření softwaru pro detekci očí a jejich jednotlivých částí. Konkrétně se jedná o detekci očních oblastí, lokalizaci duhovky a zornice a nalezení polohy očních víček. Součástí diplomové práce je i experimentální část, ve které je vyhodnocena efektivita a časová náročnost. Podnětem pro téma této práce je možnost využití představených metod v systémech, které monitorují řidiče a míru jeho ospalosti.

1.1 Systémy pro sledování míry ospalosti řidiče

Tyto systémy pomáhají předejít nehodám, které jsou způsobeny unaveností řidiče. Upozorní jej zvukovým, vibračním či vizuálním znamením, které indikuje, že by měla osoba za volantem v blízké době zastavit. Některé systémy dokonce nabídnou polohu s nejbližším občerstvením, kde je možno zakoupit kávu. Jako první tento systém představilo v roce 2007 Volvo pod názvem Driver Alert Control [1]. Tento systém je určen mimo město a monitoruje pozemní značení na vozovce, které srovnává s pohyby, jenž provedl řidič volantem. Šofér je upozorněn, pokud auto nesleduje vozovku rovnoměrně. V dnešní době, kdy výkon malých počítačů dovoluje vyhodnocovat složitý obraz v reálném čase, nabízí většina velkých automobilek obdobný systém ve výbavě. Namátkou je to třeba Audi, BMW či Mazda.

1.2 Struktura diplomové práce

Diplomová práce se skládá z pěti částí. První kapitola přináší uvedení do problému detekce očních částí, představuje strukturu této práce a ukazuje současné využití v systémech pro monitorování řidiče. V druhé kapitole budou popsány některé ostatní postupy, které se zabývají detekcí očních částí. Jsou v ní uvedeny metody ostatních autorů, a to jak pro detekci očních oblastí, tak i pro nalezení duhovky a zornice nebo zjištění polohy očních víček. V kapitole tři popíše metodu, která vede k detekci obličeje a metodu s jejíž pomocí lze nalézt oční oblasti. Dále představím postup, jak lze eliminovat odraz světla z duhovky či zornice. Poté prezentuji metody k nalezení středu oka, detekci duhovky a zornice. Jako poslední uvedu postup, který umožní lokalizaci očních víček.

Metoda pro rozpoznání obličeje a očních oblastí využívá Haarových kaskádových klasifikátorů. Podrobně popíše princip této metody, vysvětlím ho a na obrázcích ukáži, co si představit pod pojmem Haarova vlastnost. Kapitola o odstranění reflexe z duhovky a zornice představí postup, který pomocí techniky známé pod názvem image inpainting odstraní

nežádoucí odraz světla z oka či brýlí. Pro nalezení středu očí popíši dvě metody s rozličným přístupem k problému. První využívá binarizace obrazu a následné nalezení kontury odpovídající oblasti duhovky a zornice. Druhá metoda hledá v obraze místo, kam směřuje nejvíce vektorů gradientu. Dále pomocí upravené Houghovy transformace najdu kružnice, které nejlépe kopírují poloměrem a pozicí duhovku a zornici oka. Na závěr této kapitoly představím postup, který využívá binarizaci obrazu pomocí tzv. Otsuovy metody a hledání kontur v obraze. Tyto kroky povedou k lokalizaci víček v oční oblasti.

V kapitole čtyři zevrubně slovně a graficky doložím přesnost metod na vzorku více než 1500 obrázků. Měření přesnosti provedu u obou metod lokalizace středu oka, u metody pro lokalizaci duhovky a taktéž u metody pro nalezení polohy očních víček. Na závěr této kapitoly představím časovou náročnost výše uvedených metod a doložím přesnost na mnou nahraném videu. V poslední kapitole číslo pět shrnu čeho jsem docílil v této diplomové práci a naznačím postup, který by mohl vést k vylepšení některých metod. Na závěr se taktéž zamyslím nad budoucností systémů pro sledování řidiče a úrovně jeho unavenosti.

2 Popis současného stavu v dané oblasti

V dnešní době se pro lokalizaci středu oka používá několik technik. Některé používají stojany na hlavu a některé omezují pohyb hlavy na minimum. Další možností je využití více kamer [2] nebo infračerveného osvětlení [3], [4]. Tyto techniky jsou často velmi přesné, ale také nákladné. Nežádá se, když se používají pro komerční využití v systémech pro sledování pohybu očí. Použití běžné kamery je výrazně levnější a dá se s tímto systémem lépe manipulovat. Přístup k detekci obličeje by se dal dále rozdělit na detekci založenou na vzhledu a na detekci založenou na rysech. První zmíněná využívá datové sady obrázků k natrénování systému pro rozpoznávání očí [5], [6]. Metody založené na rysech se snaží využít znalost oka k odvození jeho znaků, jako např. tvar, barvu či symetrii. Pro představu Dobeš využívá v [7] Houghovu transformaci a symetrii rysů využívá Yang a kol. v [8].

Pro detekci duhovky se používají postupy s rozličným přístupem k tomuto problému. Jedno z nejznámějších pojetí segmentace duhovky představil Daugman [9]. Další známé řešení navrhl Wildes [10]. Ten používá k nalezení duhovky detekci hran a Houghovu transformaci. Mnoho prací také pojednává o řešení tohoto problému pomocí metod založených na kruhovém modelu [11], [12], [13]. V [11] Liu vylepšil Wildesovu metodu o redukci šumu, čímž dosáhl lepší univerzálnosti řešení. V [14], [15] a [16] navrhuje segmentaci duhovky pomocí detekce okluze. V poslední době se také mnoho autorů zabývá problematikou detekce duhovky na neideálních obrázcích [17], [18], [19], [20].

Detekce zornice je velmi úzce spjata s identifikací duhovky. Pokud totiž chceme identifikovat osobu pomocí duhovky, nestačí duhovku jen najít, ale oddělit ji taky od zornice, aby se dále mohla zkoumat. Daugman v [21] použil pro detekci zornice a hranic duhovky integro-diferenciální operátor. Další známé řešení, které jsem již zmínil u detekce duhovky, navrhl Wildes [10] a lze ho použít i u detekce zornice. Tento postup využívá Houghovy transformace k nalezení kontur duhovky a zornice. Kheirolahy a kol. [22] využil optimalizované mapování barev k zvýraznění zornice. Jejich metoda dosahuje úspěšnosti 98% na databázi obrázků UBIRIS. Mi Young Nam [23] vylepšuje techniku použití detekce hran tak, že zavádí dynamický práh pro detekci hrany. Další možností je kombinace Houghovy transformace a Haarových vlnek [24]. Za zmínku také určitě stojí metoda popsaná v [25], kterou navrhl Arvacheh. Přináší aktivní model kontury, který zvýrazňuje hranici zornice.

Pro identifikaci očních víček v obraze se využívá několika technik. Jednu z těch, které jsou již dlouho známy, vypracoval v roce 2000 Tian a kolektiv [26]. V tomto článku rozlišují mezi stavem, kdy je oko zavřené a kdy otevřené. Pokud je nalezena duhovka, tak je oko považováno za otevřené a hledají se střední body na očních víčkách. V [27] se hledají rohy očí pomocí detekce hran. Rohy jsou detekovány jako místa, které mají vysokou míru zakřivení hran. Pro každé oko jsou to čtyři rohy: levý a pravý roh oka a body, kde se protíná duhovka s očními víčky. Tuto metodu poté ještě zdokonalil Yuille a kol. [28]. Z podstaty metody ale vyplývá, že pokud má osoba zešíroka otevřené oči, tak víčka nepřekrývají duhovku a metoda tak selže.

3 Detekce obličeje, očí a jejich jednotlivých částí

Detekce obličeje a očí se často používá k identifikaci osob. Nicméně v posledních letech našla uplatnění i jinde. Lze ji využít k vyhodnocení zda je člověk unaven, či zda třeba i nespí. V následujících počátečních kapitolách postupně popíši detekci obličeje a detekci oční oblasti, k čemuž použiji Haarových kaskád. Následně přistoupím k detekci středu zornice, kde představím dvě metody s rozličným přístupem k tomuto problému. První metoda využívá automatickou binarizaci a hledání kontury odpovídající oblasti s duhovkou a zornicí, kdežto druhá metoda pracuje s matematickou funkcí, u které maximum koresponduje s pozicí s nejpočetnějším průnikem vektorů gradientu, a tudíž středu oka, neboli středu duhovky a zornice. Dále představím upravenou Houghovu transformaci, s jejíž pomocí detekuji polohu a průměr duhovky a následně i zornici. Nakonec přistoupím k detekci polohy očních víček, u které použiji prahování a hledání kontur.

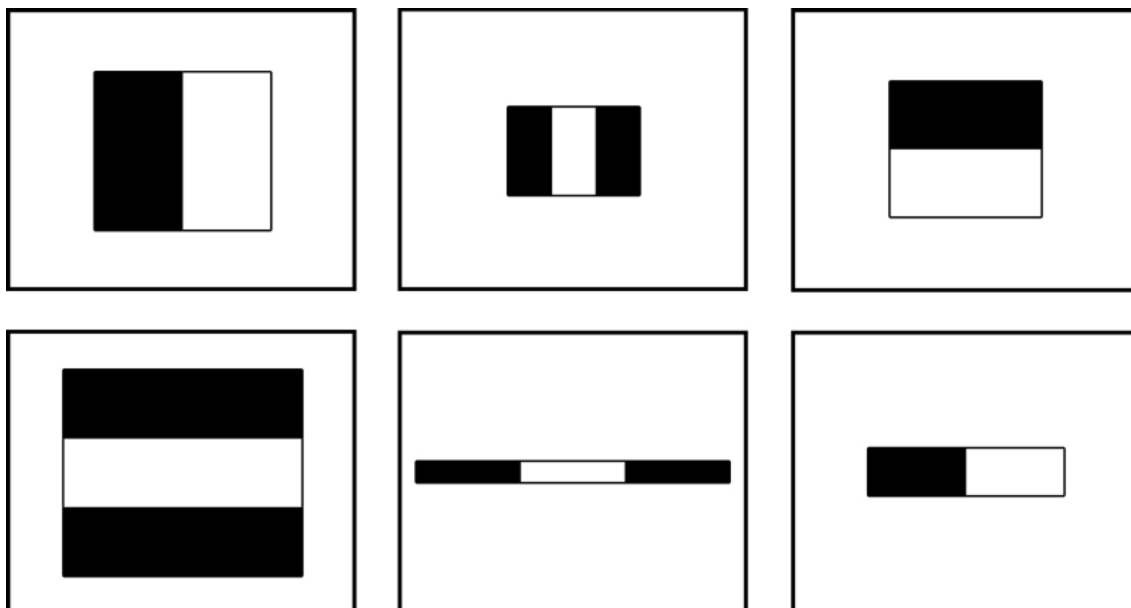
3.1 Nalezení obličeje

Nalezení obličeje v obraze bylo a stále je velmi důležitým odvětvím ve zpracování obrazu. V dnešní době je možno najít ve videu obličeje v reálném čase, a dokonce takto i provést identifikaci osob. Následné samotné využití se již skoro nabízí samo. Jednak se dá zvýšit bezpečnost například na letištích, či v jiných přísně střežených objektech, ale dá se také využít k nalezení osob. Jeho využití najdeme také u méně vážných témat, jako například detekce obličeje při focení, s čímž se dnes můžeme setkat již téměř u všech fotoaparátů a nebo u sociálních služeb, kde například Facebook dokáže pomoci s označením osob na fotografiích tak, že vytvoří obdélníky kolem potencionálních obličejů. Může se zdát, že nalezení obličeje nemá přímou souvislost s detekcí očních částí, opak je však pravdou. Pokud zdárně lokalizujeme obličej a určíme tedy oblast zájmu, můžeme značně zredukovat velikost obrazu, ve kterém budeme následovně hledat oči a jejich části, čímž značně zvýšíme efektivitu postupu.

3.1.1 Detekce obličeje pomocí kaskád slabých klasifikátorů

Velmi univerzální a úspěšnou metodou pro nalezení obličeje je použití Haarových kaskádových klasifikátorů (Haar feature-based cascade classifiers) [29]. Tuto metodu navrhli a představili v roce 2001 Paul Viola a Michael Jones, což přineslo rapidní navýšení používání detekce obličejů, jak u kamer a fotoaparátů, tak i u webových aplikací.

K nalezení obličeje v obraze se používá náročné vyhledávání pomocí posuvného okénka, nebo taky skenovacího okna, u kterého se mění velikost, poměry stran a pozice. Díky typickým stínům, které se nacházejí na lidském obličeji můžeme použít tzv. Haar-like features (obr. 1), které reprezentují rozdíl v intenzitě černobílého obrázku mezi dvěma nebo více přilehlými obdélníkovými oblastmi v obraze.

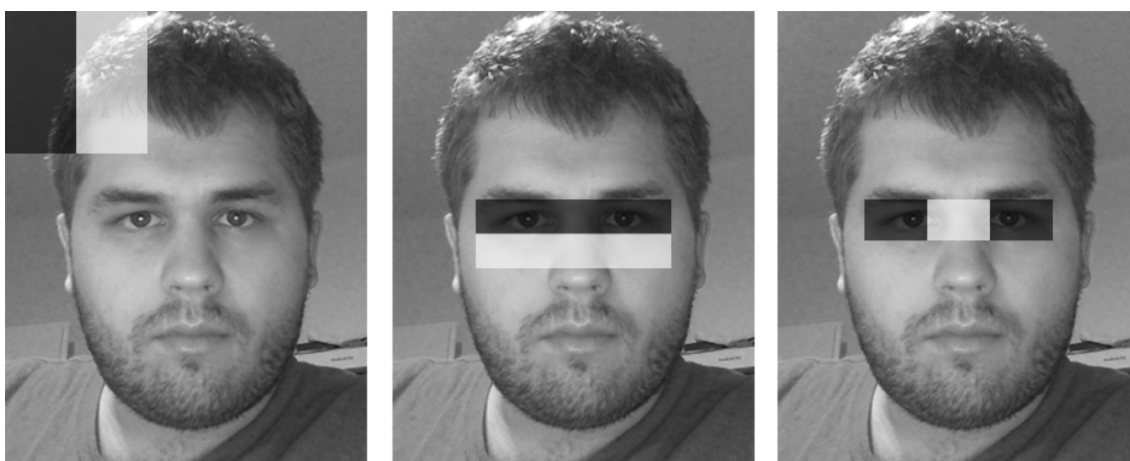


Obrázek 1 - Příklady jak mohou vypadat Haarovy vlastnosti.

Odečtením součtu pixelů v bílých obdélnících od součtu pixelů v černých obdélnících vypočteme hodnotu Haarovy vlastnosti. Pokud je výsledná hodnota v souladu s naučenou mezí, můžeme říci, zda testovaná část obrázku splňuje danou vlastnost. Slabý klasifikátor $h(x, f, p, \theta)$ se skládá z vlastnosti f , prahu θ a polarity p určující směr nerovnosti:

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{když } pf(x) < p\theta \\ 0 & \text{jinak} \end{cases}, \quad (3.1)$$

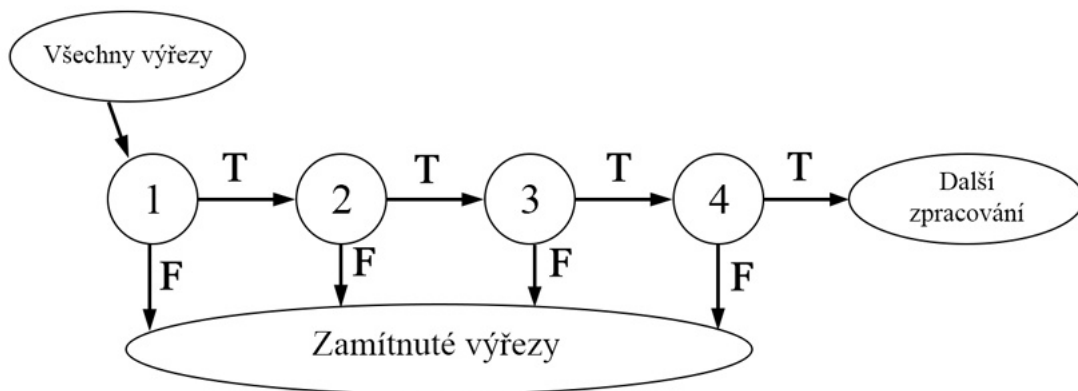
kde x je výřez z původního obrázku o velikosti 24×24 pixelů.



Obrázek 2 - Aplikace odlišných Haarových vlastností na obličejí.

Na obrázku 2 můžeme vidět aplikaci různých Haarových vlastností na obličejí. Za povšimnutí stojí různé velikosti a rotace jednotlivých vlastností. Na levém obrázku můžeme vidět vlastnost, jejíž hodnota by pravděpodobně nesplňovala naučenou mez, a tudíž by byla považována za nekorektní, kdežto na prostředním a pravém obrázku vidíme vlastnosti, které by pravděpodobně byly označeny za platné.

Obrázek 2 nám taktéž přibližuje, jak pracuje vyhledávací algoritmus. Posuvné okénko se postupně pohybuje celým obrázkem, testuje určitou vlastnost, a když dokončí jeden průchod, tak okénko změní velikost a celý proces se opakuje od začátku. Takovéto hledání by bylo velmi náročné, a proto musí detektor rychle rozhodnout, zda daný region obsahuje tvář, či nikoliv. Detektor používá kaskádu (obr. 3), což je speciální uspořádání klasifikátorů, které zajišťuje, že daný klasifikátor je vyhodnocován na dané oblasti pouze tehdy, pokud všechny předchozí klasifikátory byly vyhodnoceny pozitivně. Pokud zhodnotí, že v dané oblasti obličej není, tak se nadále daným regionem už nezabývá a netestuje na něm jiné vlastnosti, což samozřejmě značně zrychluje tento postup. Pokud si není jistý, tak testuje oblast dále.



Obrázek 3 - Znáznornění kaskády. Počáteční klasifikátor může rychle eliminovat nevhodnou oblast pro obličej. Daný region se již nebude testovat, a ušetří se tak výpočetní výkon.

Každá fáze kaskády je tzv. silný klasifikátor, ve smyslu, že může obdržet velmi velkou míru odmítnutí kombinací slabých klasifikátorů. Slabý klasifikátor disponuje kvalitou detekce jen o něco větší, než náhodné hádání. Pokud ale zkombinujeme dostatečné množství rozhodnutí na základě slabých klasifikátorů, tak vytvoříme silný klasifikátor. Tento proces kombinace několika slabých klasifikátorů do jednoho silného se nazývá boosting. Naučení takového klasifikátoru může být dosaženo pomocí nějaké variace AdaBoost učicího se algoritmu. Velmi pěkné video, kde je vizualizovaný celý proces vyhledání obličeje najdeme na [30].

U každé vlastnosti se počítá několikrát součet pixelů v obraze, což znamená, že neustále opakujeme stejné operace, které jsou při svém vysokém množství výpočetně náročné. Stejně tak jako například před cyklem for si ukládáme nějakou hodnotu dopředu, aby se nemusela počítat při každé iteraci cyklu, tak i tady je zavedena tabulka, která obsahuje předpočítané součty intenzit pixelů, což představuje tzv. integrální obraz (v angličtině se můžeme setkat i s pojmem Summed Area Table). Zelená matice na obrázku 4 reprezentuje původní obrázek a modrá matice reprezentuje integrální obraz.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	3	6	10
6	14	24	36
15	33	54	78
28	60	96	136

Obrázek 4 - Vlevo hodnoty intenzity obrazu. Vpravo hodnoty integrálního obrazu.

Jakýkoliv prvek v modré matici je tvořen součtem prvků v zelené matici, které jsou umístěny vlevo a nad od tohoto prvku. K součtu samozřejmě patří i prvek se stejnými souřadnicemi v zelené matici. Výše popsany postup je vyjádřen následující rovnicí:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (3.2)$$

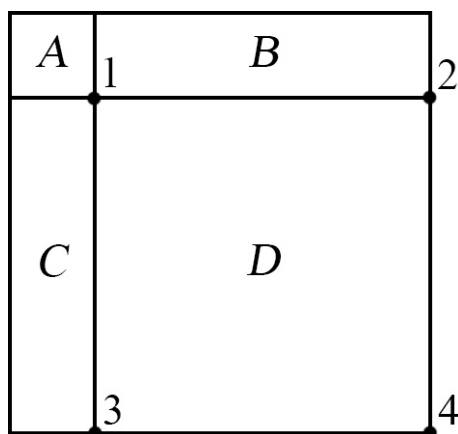
kde $ii(x, y)$ je integrální obraz a $i(x, y)$ je originální obraz. Integrální obraz může být vypočítán i jedním průchodem obrázkem:

$$s(x, y) = s(x, y-1) + i(x, y), \quad (3.3)$$

$$ii(x, y) = ii(x-1, y) + s(x, y), \quad (3.4)$$

kde $s(x, y)$ je kumulovaný součet hodnot v řádku, $s(x, -1) = 0$ a $ii(-1, y) = 0$.

Na obrázku 4 vlevo jsou tmavě zelenou barvou zvýrazněny hodnoty, u kterých chceme znát součet. Při běžném postupu bychom museli přečíst všech devět hodnot intenzit obrázku, a poté je sečíst. Pokud ale použijeme integrální obraz, tak nám stačí pracovat jen se čtyřmi hodnotami, kterými jsou rohové pixely integrálního obrazu. Součet pravého horního (10) a levého spodního (28) rohu odečteme od součtu levého horního (1) a pravého spodního rohu (136), což se rovná součtu všech intenzit v dané části obrazu, tedy 99. Na obrázku 5 můžeme vidět jednodušší příklad integrálního obrazu. Hodnota integrálního obrazu na pozici 1 je součtem všech pixelů v obdélníku A . Hodnota v místě 2 je součtem A a B , na pozici 3 to je $A + C$ a na pozici 4 to je $A + B + C + D$. Součet všech intenzit pixelů v obdélníku D spočítáme jako $4 + 1 - (2 + 3)$. Použitím Integrálního obrazu zredukujeme složitost z $O(n^2)$ na $O(1)$.



Obrázek 5 - Součet všech pixelů v obdélníku D můžeme spočítat jako $4 + 1 - (2 + 3)$.

V mé práci jsem použil knihovnu OpenCV [31], kde jsou již některé předtrénované klasifikátory v XML souborech. Například pro obličej, oči či ústa. Před samotnou detekcí obličeje je nutné obrázek optimalizovat. Jednak je nezbytné převést obrázek do odstínů šedi a dále je vhodné provést vyrovnání kontrastu. Podle zadání diplomové práce jsem aplikaci uzpůsobil pro detekci jednoho obličeje. Tato potřeba vychází z praxe a lze se s ní setkat například i u biomedicínských aplikací. Na obrázku 6 vidíme správně rozeznatý obličej. Obraz je již černobílý a má vyrovnaný kontrast.

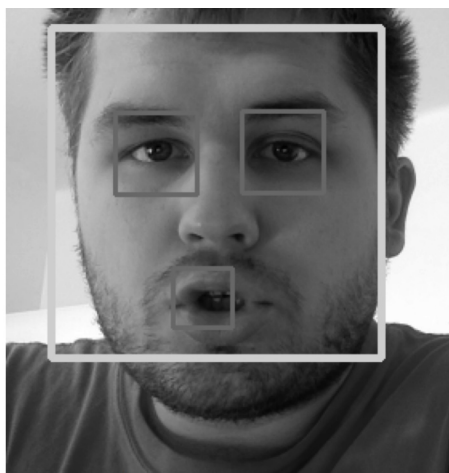


Obrázek 6 - Správně detekovaný obličej.

3.2 Detekce očních oblastí

Dalším neméně důležitým krokem je nalézt oblast, kde se nacházejí oči. Pro jejich nalezení jsem použil stejně jako u obličeje Haarových kaskádových klasifikátorů. Rozdíl tkví v tom, že tentokrát použijeme předtrénované klasifikátory, které slouží k nalezení očních oblastí. Tato technika nedetekuje jednu oční oblast s dvěma očima, ale každou oční oblast zvlášť, což je výhodné vzhledem k ušetřenému výpočetnímu výkonu a minimalizaci oblasti hledání při následovné detekci zornice, či očních víček.

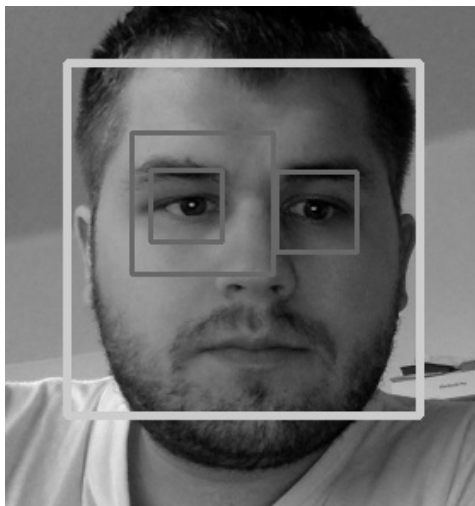
Jak můžeme vidět na obrázku 7, tak může dojít k situaci, kdy se oblast mylně vyhodnotí jako oční a dojde tak k falešné detekci. Dále popíši několik postupů, jakým způsobem je eliminovat. Pokud si horizontálně rozdělíme obličej na dvě poloviny, tak můžeme říci, že oční oblasti jsou nekorektně detekované, pokud se nacházejí ve spodní polovině obličeje. Dále odstraníme oční oblasti, které svoji horizontální polohou neodpovídají korektnímu umístění v obličeji. Pokud jedna z hran detekované oční oblasti leží mimo obličej, můžeme říci, že taková detekce není platná.



Obrázek 7 - Špatná detekce očních oblastí.

Následovně rozdělíme svisle obličej na dva shodně velké obdélníky. Nejprve vezmeme v potaz levé oko. Toto oko má střed v levém obdélníku. Pokud jeho pravá hrana překročí pravou hranu obdélníku, tedy středovou osu obličeje, můžeme konstatovat, že taková oční oblast je

nekorektní (obr. 8). Pro pravé oko je postup obdobný. Pravé oko má střed v pravé polovině obličeje a pokud jeho levá hrana zasahuje přes půl obličeje, můžeme považovat tuto oční oblast taktéž za neplatnou.



Obrázek 8 - Mylně detekovaná oční oblast. Větší oční oblast na pravé straně obličeje zasahuje přes jeho polovinu. Považujeme ji tedy za nekorektní.

Při použití detekce očních oblastí pomocí Haarových vlastností se také stává, že obdržíme dvě a více očních oblastí s podobným středem, a tudíž se tyto regiony překrývají. Obě detekce jsou ve své podstatě správné a korektně zobrazují oční oblast. Proto si musíme zvolit jednu, kterou dále použijeme. Nejlepším postupem je myslet dopředu a vybrat tu oblast, na které budou následující detekce očních částí nejméně výpočetně náročné. Proto pokud nastala výše popsaná situace, volil jsem takovou oční oblast, která má nejmenší obsah, a tudíž bude pro kroky, které dále popíší v této práci, výkonnostně nejméně náročná.

3.3 Odebrání reflexe z očí

Před samotnou detekcí duhovky a zornice, je vhodné upravit obraz do podoby optimální pro jejich nalezení. Proměnlivé světelné podmínky jsou vždy velkým problémem a je třeba se pokusit alespoň částečně eliminovat jejich důsledky. Často se například stává, že pokud sedí osoba ve dne naproti oknu, překryje velkou část duhovky a zornice odraz světla z okna. Obdobně pokud se subjekt nachází v noci před monitorem nebo u jiného zdroje světla, tak se v oblasti středu oka může vytvořit reflexe. Dále v textu popíší kroky, kterými lze odraz odebrat.

Těmito kroky jsou postupně odebrání šumu, použití Sobelova operátoru, prahování obrazu a použití metody image inpainting. Tímto postupem si radikálně vylepšíme vstupní podmínky pro následné rozeznávání částí oka.

3.3.1 Odebrání šumu

Pro začátek odstraníme z obrázku šum a detaily, čehož můžeme dosáhnout pomocí Gaussova filtru. Jedná se o nízko frekvenční filtr, který redukuje vysoké frekvence. V následujícím kroku použijeme Sobelův operátor k získání gradientu obrázku, což je směrová změna v intenzitě nebo barvě obrazu.

3.3.2 Sobelův operátor

Sobelův operátor [32], neboli taky Sobelův filtr, se používá pro detekci hran v obraze, kde se využívají dvě parciální derivace. Jedna ve směru osy x a druhá ve směru osy y . Ty se v diskrétním obraze aproximují konvoluční maticí. Matice s vyšším stupněm přináší větší odolnost vůči šumu, ale na druhou stranu nejsou detekované hrany natolik ostré, protože dojde k jejich rozmazání. Nejčastěji se používají dvě konvoluční jádra o velikosti 3×3 , kde jedna z matic je otočená o 90 stupňů (3.5). Často se můžeme setkat s pojmem gradient, což je vektor jehož první složka je tvořena onou derivací ve směru osy x a druhá složka derivací ve směru osy y .

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix} \quad (3.5)$$

Na obrázku 9 můžeme vidět derivaci v horizontálním a vertikálním směru, stejně tak jako její celkovou velikost, kterou spočítáme podle vzorce (3.6). Výsledná velikost bude největší v místech, kde je nejvyšší změna intenzity pixelů, což odpovídá místům, kde se mohou nacházet hrany. Naopak nízká hodnota svědčí o místech, kde je obrázek monotónní. Řekněme, že derivace ve směru osy x pro daný bod označíme jako f_x a ve směru osy y jako f_y . Poté se velikost derivace vypočítá jako:

$$|\nabla f(x, y)| = \sqrt{f_x(x, y)^2 + f_y(x, y)^2} \quad (3.6)$$

a její směr jako:

$$\Theta = \arctg \left(\frac{f_y(x, y)}{f_x(x, y)} \right). \quad (3.7)$$



Obrázek 9 - Vlevo derivace ve vertikálním směru. Uprostřed derivace v horizontálním směru. Vpravo výsledný gradient.

3.3.3 Binarizace obrazu

K úspěšnému odebrání reflexe z očí budeme potřebovat masku, ve které budou oblasti s ostrou reflexí světla. Proto přistoupíme k binarizaci obrazu. Nejprve si ale vysvětlíme, co si pod pojmem binarizace představít. Jedná se o segmentaci obrazu na základě intenzity pixelu. Nejčastěji rozdíl mezi objektem a pozadím obrázku. K rozeznání chtěných pixelů, provedeme porovnání intenzity každého pixelu a dané meze. Jednotlivým pixelům nastavíme specifickou intenzitu a tím je jednoznačně identifikujeme. To dosáhneme takzvaným prahováním obrazu (3.8), které zajišťuje, že pokud je intenzita pixelu na vstupním obrázku $f(x, y)$ větší než daná mezní hodnota T , je nastavena intenzita výstupnímu binárnímu obrázku $g(x, y)$ na 255. V opačném případě je nastavena na 0.

$$g(x, y) = \begin{cases} 255 & \text{když } f(x, y) > T \\ 0 & \text{jinak} \end{cases} \quad (3.8)$$

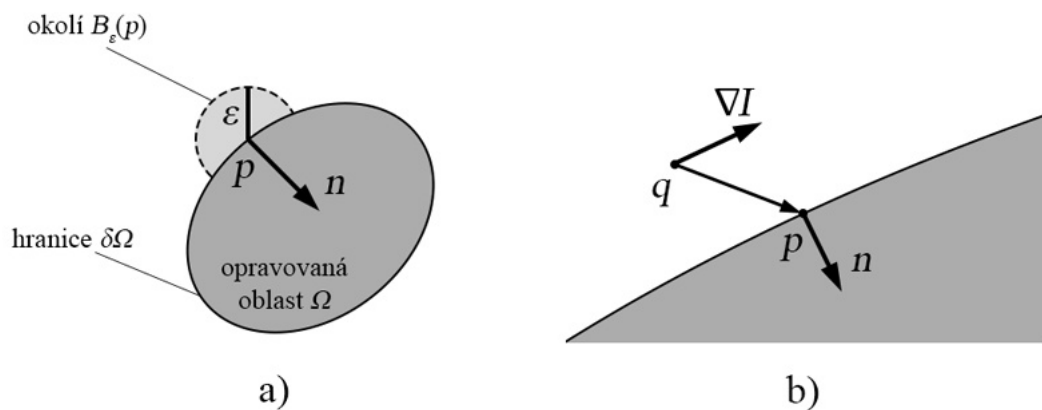
3.3.4 Odstranění reflexe světelného zdroje

Posledním krokem je odstranění odrazu světla ze zornice a duhovky. K tomu použijeme techniku, která se jmenuje image inpainting. Ta na vstupu přebírá původní obraz a masku, kterou jsme získali v minulé kapitole. Tato metoda se běžně používá pro restaurování

digitalizovaných fotografií, u nichž dokáže opravit malé poškozené části. Pokud tuto myšlenku převedeme na můj případ, tak můžeme říci, že odraz světla, který se vytváří ve formě shluku pixelů s vysokou intenzitou v místě zornice a duhovky, je vlastně ono poškození, které je třeba odstranit. Princip je celkem prostý. Nechtěné pixely nahradíme pixely sousedními.

Metoda image inpainting pracuje tak, že nejprve postupně vyplňuje pixely na okrajích oblasti, která se má opravit. Blízké okolí jednoho pixelu nahradí normalizovaným váženým součtem všech známých pixelů v okolí. Více váhy je přiřazeno pixelům ležícím poblíž bodu, blízko normály hranice a těm pixelům nacházejícím se na hranicích obrysu. Pro lepší pochopení algoritmu zvažme obrázek 10. V tomto obrázku chceme nahradit bod p , který je na hranici $\delta\Omega$ oblasti Ω , která má být opravena. Řekněme, že okolí $B_\varepsilon(p)$ bodu p je dáno velikostí ε (obr. 10(a)). Nahrazení p by pak mělo být určeno hodnotami známých obrazových bodů v jeho blízkosti, tj. v $B_\varepsilon(p)$. Jako dostatečně malé ε považujeme aproximaci prvního řádu $I_q(p)$ obrazu v bodě p , vzhledem k obrazu $I(q)$ a gradientu $\nabla I(q)$ bodu q (obr. 10(b)):

$$I_q(p) = I(q) + \nabla I(q)(p - q). \quad (3.9)$$



Obrázek 10 - Princip metody image inpainting.

Dále nahradíme bod p funkcí všech bodů q v $B_\varepsilon(p)$ součtem odhadů všech bodů q , vážené normalizovanou váhovou funkcí $w(p, q)$:

$$I(p) = \frac{\sum_{q \in B_\epsilon(p)} w(p, q) [I(q) + \nabla I(q)(p - q)]}{\sum_{q \in B_\epsilon(p)} w(p, q)}. \quad (3.10)$$

Váhová funkce $w(p, q)$ je navržena tak, že inpainting bodu p propaguje hodnoty šedé, stejně tak jako ostré detaily obrazu v okolí $B_\epsilon(p)$. V mém řešení jsem konkrétně použil algoritmus na základě článku [33]. Ten rozšířil původní metodu inpaintingu tak, že ve chvíli, kdy je pixel opraven, tak se algoritmus posune na další pixel používajíc Fast Marching Method. FMM zajistí, že pixely kolem známých pixelů jsou obarveny první. Na obrázku 11 vidíme téměř celý proces odebrání odrazu světla z duhovky a zornice. Vlevo můžeme spatřit oko před redukcí reflexe. V oblasti zornice je patrný odraz světla. Na prostředním obrázku můžeme vidět vizualizovanou velikost gradientu. Vpravo je již výsledné oko bez jakékoliv reflexe. S minimální ztrátou kvality obrázku jsme tudíž úspěšně odebrali odraz světla ze zornice.



Obrázek 11 - Vlevo oko s odrazem světla. Uprostřed maska s reflexí. Vpravo oko s odstraněným odrazem světla.

3.4 Nalezení středu oka

Nyní, když máme oční oblasti bez rušivých odlesků, můžeme přistoupit k nalezení středu oka. Tato část byla obzvlášť náročná, protože mnoho postupů, které lze najít v člácích, pracuje s kamerami, které jsou umístěny přímo na hlavě a prakticky se u nich nemění světelné podmínky. V mém řešení používám běžnou kameru, tudíž můj postup musel být více univerzální. Podrobně jsem nastudoval a implementoval dvě metody. První se pokouší najít střed pomocí binarizace obrazu a nalezení kontury v oblasti duhovky a zornice [34]. Druhá metoda [35] používá k nalezení středu směry gradientu v obrázku.

3.4.1 Detekce středu duhovky s korekcí stínů v oční oblasti

Tato metoda je popsána v článku [34]. Nicméně jsem z praxe zjistil, že ne všechny kroky, které jsou v něm uvedené se pro mé řešení dají použít, a proto jsem ji musel upravit. Zejména potom identifikaci kontury odpovídající poloze duhovky. Postup bude následující:

- vynásobení intenzity obrazu
- binarizace obrazu
- nalezení kontury, která odpovídá pozici duhovky
- oprava středu oka v případě špatné pozice středu způsobené stínem v oční oblasti

Jako vstupní obraz pro tuto metodu je vyžadován co nejmenší černobílý výřez oblasti kolem oka. Samozřejmě můžeme použít jako vstup oční oblast, o níž jsem psal v kapitole 3.2. Vzhledem k tomu, že dále budeme obraz binarizovat, je důležité dodržet předzpracování obrazu ve formě odstranění reflexe, které jsem popsal v kapitole 3.3. Dále také provedeme redukci šumu v obraze. V kapitole 3.3.1 jsem popisoval jeho snížení pomocí Gaussova filtru. V tomto případě jsem se ale rozhodl pro mediánový filtr, protože jeden z následujících kroků bude zaměřen na rozpoznání hran, a tento filtr je o něco lepší pro jejich zachování a současně dokáže velmi slušně redukovat šum. Mediánový filtr pracuje tak, že prochází každý pixel v obraze a nahradí ho mediánem okolí sousedních pixelů, čímž docílí odstranění šumu. Hrany zachovává, protože mediánová hodnota je skutečnou hodnotou jednoho z pixelů okolí zpracovávaného bodu. To znamená, že pokud narazí na hranu v obrázku, tak nevytvoří nový pixel s nerealistickou hodnotou.

Vycházíme-li z toho, že zornice a duhovka jsou mnohem tmavší než bělmo a okolí očních víček, můžeme tyto tmavé oblasti zvýraznit, pokud vynásobíme intenzitu jednotlivých pixelů konstantou. Tím zůstanou výrazně tmavé části stále tmavé a světlé části se stanou čistě bílými (3.11).

$$I'(x, y) = kI(x, y) \quad (3.11)$$

$I(x, y)$ je intenzita pixelu se souřadnicemi (x, y) a k je konstanta, kterou se násobí intenzita obrazu. $I'(x, y)$ je intenzita pixelu již po vynásobení konstantou. Pokud uvažujeme práci

s černobílým obrazem, kde je oborem hodnot množina $\{0, \dots, 255\}$, pak je třeba dát pozor na výslednou hodnotu $I'(x, y)$. Hodnota po vynásobení intenzity pixelu nesmí překročit pravou mez množiny, tedy 255. Oční oblast před a po násobení intenzity můžeme vidět na obrázku číslo 12.



Obrázek 12 - Vlevo obrázek před vynásobením intenzity pixelů. Vpravo obrázek po násobení intenzity.

Následně přikročíme k binarizaci obrazu. Na rozdíl od postupu, který jsem popsal v kapitole 3.3.3, použijeme tzv. Otsuovu metodu [36]. Jedná se o binarizační algoritmus, s jehož pomocí můžeme automatizovaně konvertovat černobílý obraz do jeho binární podoby. Vzhledem k tomu, že tato metoda hledá mez, která odděluje v obraze popředí a pozadí, funguje nejlépe na obrázcích, které mají tzv. bi-modální histogram. Bi-modální histogram je takový histogram, který má dva výrazné vrcholy. Můžeme tedy říci, že takový obraz obsahuje dvě skupiny bodů s různým rozsahem hodnot intenzit. Jelikož se ale tyto rozsahy mohou překrývat, snaží se metoda Otsu nalézt optimální práh na základě výpočtu rozptylu. Vnitřní rozptyl je dán jako:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t), \quad (3.12)$$

kde

$$q_1(t) = \sum_{i=0}^{t-1} P(i) \quad \text{a} \quad q_2(t) = \sum_{i=t+1}^{N-1} P(i) \quad (3.13)$$

jsou váhy, P reprezentuje histogram obrázku, $[0, N-1]$ je rozsah intenzit a t je treshold, neboli práh. Vnitřní rozptyl je vážená suma rozptylu všech pixelů v pozadí $\sigma_1^2(t)$ a v popředí

$\sigma_2^2(t)$. Pro získání korektního bi-modálního histogramu je třeba klást velký důraz na odstranění šumu, které jsem popsal výše. Pro obrázky, které takový histogram nemají, není výsledná binarizace natolik efektivní. Binární obraz získaný použitím Otsuovy metody můžeme vidět na obrázku 13 vlevo.



Obrázek 13 - Vlevo binární obrázek získaný pomocí binarizační Otsuovy metody. Vpravo všechny nalezené kontury společně s jejich ohraničujícími boxy.

Následně přistoupíme k nalezení kontur v binárním obraze. V ideálním případě bychom detekovali jeden obrys, a tím by byl útvar v okolí středu oka. Bohužel často tomu tak není a musíme přistoupit k určité filtraci výsledků. Jak je možno vidět na obrázku 13 vpravo, může nastat situace, kdy nalezneme jednak vhodný obrys v okolí duhovky, ale také obrys detekovaného obočí. Autoři článku [34] navrhli následující postup.

Uvažujme obdélník, který nejtěsněji obepíná zkoumanou konturu a pojmenujme ho ohraničující box. Pokud je výška ohraničujícího boxu větší nebo rovna třem polovinám jeho šířky, tak tuto konturu nepovažujeme za vhodnou, protože hledáme obrys, který má ohraničující box obdélníkového tvaru. Dle autorů článku by stačilo již jen vybrat dvě kontury s největším obsahem a rozhodnout, která z nich se nachází v oblasti duhovky a zornice. Nicméně z praxe po naimplementování mohou říci, že tento postup by nebyl dostatečný. Proto jsem musel metodu následovně upravit. Najdeme konturu s největším obsahem a předpokládáme, že to je první vhodný kandidát. Dále budeme hledat druhého vhodného kandidáta. Zavedeme minimální šířku a výšku kontury a přidáme pravidlo, že druhá kontura nemůže ležet žádnou svou částí v kontuře první, čímž odebereme obrysy, které překrývají prvního kandidáta. Tímto postupem najdeme druhou největší konturu, která by mohla ležet v oblasti zornice a duhovky a zároveň splňuje výše popsaná pravidla.

Na obrázku 14 uprostřed můžeme vidět příklad nalezení více kontur s ohraničujícími boxy. Kontura s číslem 1 představuje oblast obočí. Kontury 2 a 3 potom oblast v okolí duhovky a zornice. Podle výše popsaného postupu algoritmus vybere jako prvního kandidáta kontura s číslem 2, protože má největší obsah. Kontura číslo 3 bude ignorována, protože leží v první kontuře, a tudíž algoritmus zvolí, jako druhého kandidáta na střed oka, konturu s číslem 1.



Obrázek 14 - Vlevo obrázek na vstupu. Uprostřed nalezené a očíslované kontury. Vpravo nalezený střed oka.

Následně musíme určit, který ze dvou kandidátů na střed oka je ten správný. Opět jsem se musel odchýlit od výše zmíněného článku, protože popsaný postup nebyl dostatečně přesný. Pokud leží středy kontur dostatečně blízko sebe, vyberu konturu, která má menší průměrnou intenzitu (3.14). Průměr se počítá jen u těch pixelů, které mají intenzitu menší než konstanta T . Pokud je T dostatečně velké, tak se do průměru nepočítají případné zbytkové odlesky nacházející se v oblasti zornice. $I(x, y)$ je intenzita pixelu se souřadnicemi (x, y) a N je počet všech pixelů kontury, které byly vzaty v potaz.

$$\text{průměr} = \frac{\sum_{I(x, y) < T}^N I(x, y)}{N}, \text{ kde } I(x, y) < T \quad (3.14)$$

Předpokládejme, že počátek souřadnicového systému leží v levém horním rohu obrazu. Pokud středy dvou finálních kontur neleží dostatečně blízko vedle sebe, tak vybereme konturu s větší y souřadnicí. Pro tento postup jsem se rozhodl, protože nejčastěji docházelo k případům, kdy mezi dvěma kandidáty byly reálný střed oka a část obočí, nebo rohová část oka. Proto

pokud jsou středy kontur podobné, mohou předpokládat, že jde o rohovou část oka a lze spoléhat na to, že oko se zornicí a duhovkou bude tmavší. Pokud středy těchto kandidátů neleží poblíž, mohou se domnívat, že jedna z kontur leží v oblasti obočí, a proto vyberu tu s vyšší y souřadnicí, což lépe odpovídá středu oka. V posledním kroku přikročíme k proložení nejvhodnější kontury elipsou. Střed této elipsy, potom můžeme považovat za střed duhovky, a tedy i zornice (obr. 15).



Obrázek 15 - Vlevo binární obrázek s nalezenou elipsou a jejím středem. Vpravo obrázek oka a jeho střed.

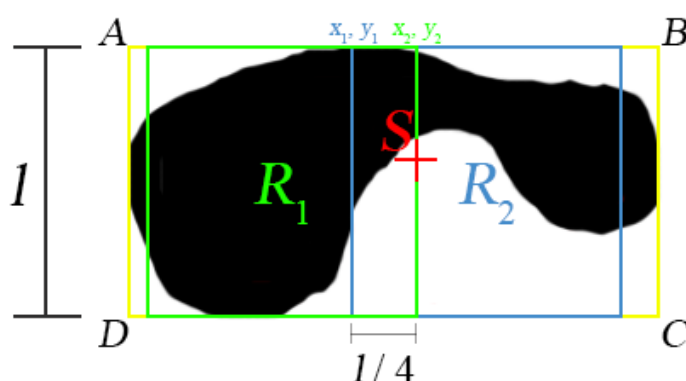
V článku [34] je dále i uvedeno vylepšení algoritmu, které značně zvyšuje jeho přesnost, a proto ho zde i popíši. Častým problémem u detekce středu oka jsou řasy, které mohou vytvořit shluk a jevit se tak na obraze jako tmavé místo. Zejména v rozích očí, kde se díky jejich tvaru často objevuje i stín. Stává se tak, že útvar v binárním obraze nemá pravidelný tvar a v něm nalezená elipsa neodpovídá středu oka (obr. 16).



Obrázek 16 - Špatná detekce středu oka v důsledku stínu ve vnitřním rohu oční oblasti.

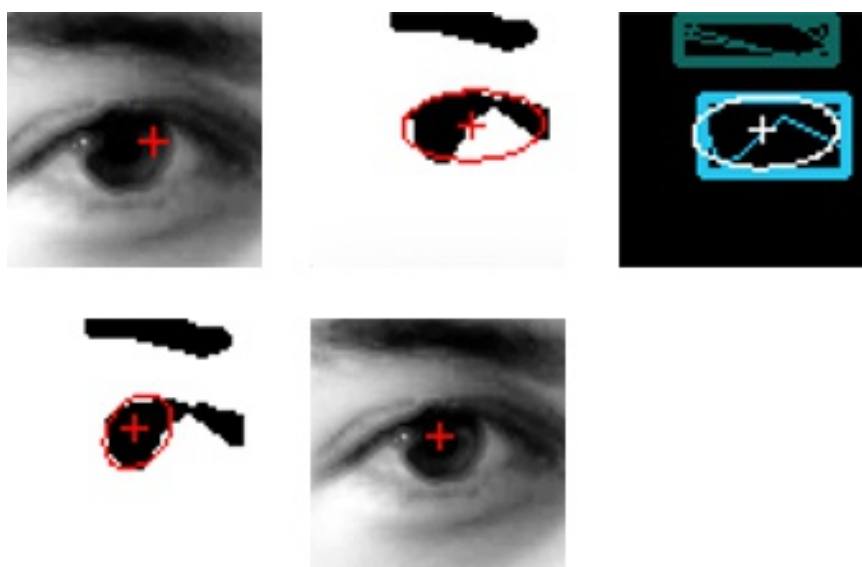
Na obrázku 16 můžeme vidět případ, kdy došlo ke špatné detekci středu zornice, a to z důvodu, že ve vnitřním rohu oka se nachází stín, který následně znehodnotí přesnost binarizace a výsledný střed proložené elipsy tak neodpovídá skutečnému středu zornice. Proto

po každé detekci středu oka provedeme kontrolu, zdali je nutné přistoupit k jeho korekci. Pro kontrolu využijeme dva pomocné čtverce, které můžeme vidět na obrázku 17. Tento obrázek představuje obdobnou situaci jaká je vyobrazena na obrázku 16 – jedná se tedy o binární obraz duhovky, zornice a stínu, který vznikl na vnitřním okraji oka. S je střed nalezené elipsy a obdélník $ABCD$ je ohraničující box. Dále vidíme dva výše zmíněné čtverce R_1 (zelený) a R_2 (modrý), které mají délku strany l , což je délka mezi body A a D , tedy výška obdélníku $ABCD$. Souřadnice levého horního vrcholu čtverce R_2 označíme jako x_1, y_1 , kde $x_1 = S_x - l/4$ a $y_1 = A_y$. Souřadnice pravého horního vrcholu čtverce R_1 označíme jako x_2, y_2 , kde $x_2 = S_x$ a $y_2 = A_y$.



Obrázek 17 - Binární obrázek znázorňující, zda je potřebná korekce středu oka.

Pokud je počet černých pixelů ve čtverci R_1 větší než ve čtverci R_2 je potřeba upravit střed. Na obrázku 18 můžeme vidět příklad nekorektní detekce středu a jeho následnou úpravu. Dále popíši postup úpravy středu oka. Nejprve postupně zkoumáme všechny body kontury, a pokud daný bod splní podmínku, že jeho souřadnice x je menší než součet $A_x + l$, tak ho považujeme za korektní bod kontury. To znamená, že pokud bod kontury leží v horizontální vzdálenosti od ohraničujícího boxu menší než je jeho výška l , považujeme bod této kontury za správný. Tímto postupem získáme novou konturu, kterou proložíme elipsou, jež bude mít jiný výsledný střed než původní elipsa. Na obrázku 18 dole vlevo můžeme vidět novou elipsu s novým středem. Jak lze vidět na stejném obrázku dole vpravo, tato korekce proběhla úspěšně a značně vylepšila předchozí výsledek.



Obrázek 18 - Nahoře vlevo oko se špatnou detekcí středu. Nahoře uprostřed binární obrázek s viditelnou elipsou a jejím středem. Nahoře vpravo možné kontury s ohraničujícími boxy a střed s elipsou. Dole vlevo již opravený binární obrázek. Dole uprostřed finální určení středu oka.

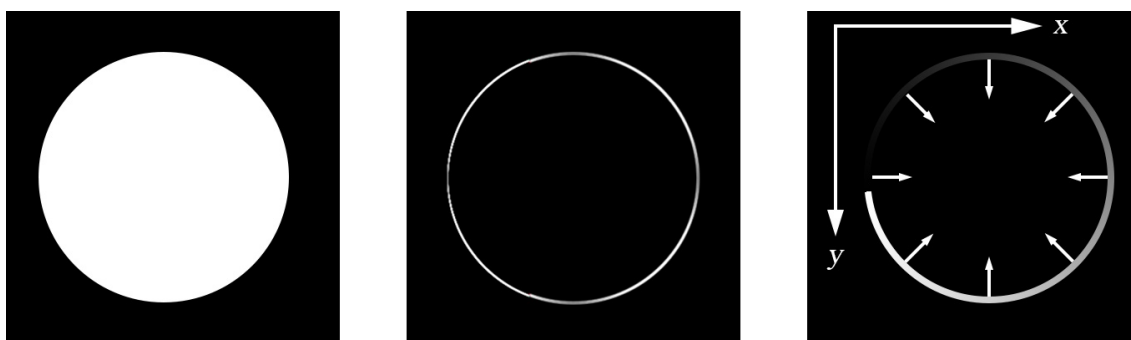
3.4.2 Lokalizace středu oka pomocí gradientů

Tato metoda se značně odlišuje od metody popsané v předchozí kapitole. Používá totiž gradient pro lokalizaci středu oka. Představuje matematickou funkci, jejíž maximum odpovídá pozici, ve které se protíná nejvíce vektorů gradientu. Tato pozice by tudíž měla odpovídat středu oka. Metoda je taktéž rezistentní vůči změně velikosti, pozice, kontrastu a rozličnému osvětlení v obraze. Jelikož dále popsaný algoritmus je výpočetně složitý, musel jsem podniknout kroky, které by ho urychlily. Jedna z možností je oblast očí zmenšit. Tento postup je účinný, ale při zmenšení obrazu dochází ke zvětšení nepřesnosti u lokalizace středu oka. Druhou možností je oblast očí ořezat (obr. 19). Pro mé řešení bylo vhodné ořezat výšku obrazu o cca 10% a šířku o cca 20%. Tento postup byl možný, protože detekované oční oblasti mají dostatečnou velikost a kolem samotné duhovky a zornice se nalézá ještě dostatek prostoru, takže touto operací nebyla nijak oční oblast znehodnocena. Výhoda tohoto postupu je, že zmenšením výšky podstatně eliminujeme rozsah obočí v obraze, což vede k menšímu počtu falešných detekcí.



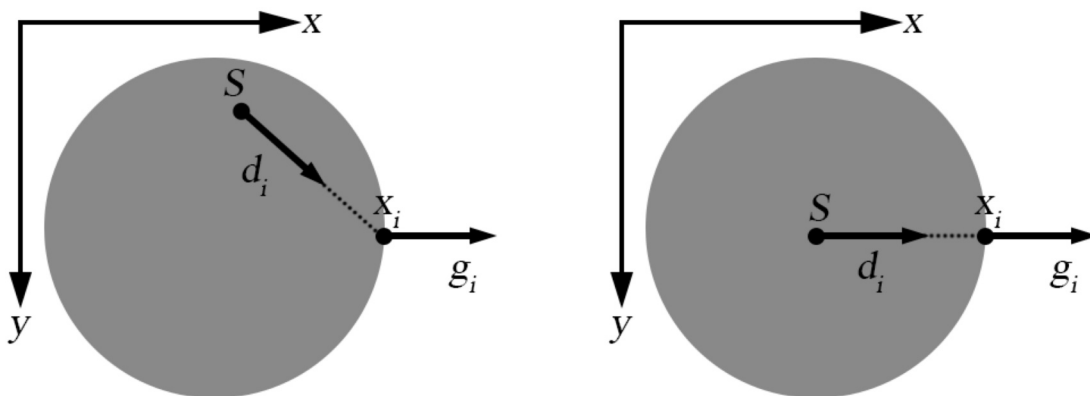
Obrázek 19 - Vlevo neořezaná oční oblast. Vpravo oblast již po ořezání.

Stejně tak jako u metody popisované v kapitole 3.3.1, musíme i nyní přistoupit k redukci šumu. U této metody jsem použil Gaussův filtr. Dále jsem použil násobení intenzity pro zachování pouze tmavých částí v obraze. Poté jsem detekoval hrany v obraze pomocí Sobelova operátoru. Tento postup jsem již popsal v kapitole 3.3.2.



Obrázek 20 - Vlevo kruh na černém pozadí. Uprostřed velikost gradientu. Vpravo směry gradientu.

Na obrázku 20 vlevo můžeme vidět bílý kruh na černém pozadí. Pokud bychom spočítali v každém bodě tohoto obrázku velikost gradientu, výsledek bude obdobný jako prostřední obrázek. Pro metodu popsanou v článku [35] je ale nejdůležitější směr gradientu, který ilustruje obrázek vpravo a ukazuje rozdílné směry vektorů gradientu, které všechny směřují do středu kružnice. Matematicky můžeme rozpoznat střed kruhu pomocí zkoumání vektorového pole gradientu obrázku (obr. 21).



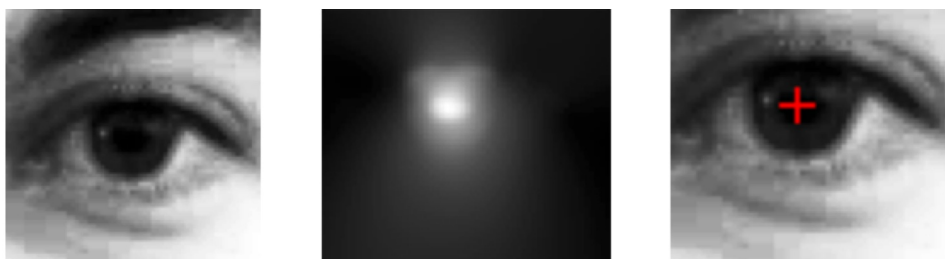
Obrázek 21 - Vlevo nekorektní střed kruhu. Vpravo správný střed kruhu. Vektory d_i a g_i mají stejný směr.

Řekněme, že S je možný střed a g_i je vektor gradientu na pozici x_i . Potom vektor d_i , který vznikl jako rozdíl mezi x_i a S , pokud nebudeme brát v potaz znaménka, by měl mít stejný směr jako vektor gradientu g_i . Proto počítáme skalární součin mezi normalizovaným vektorem d_i a normalizovaným vektorem gradientu g_i . Skalární součin počítáme pro každý bod v obraze, jenž má velikost gradient vektoru větší než nula. Střed s nejvíce průniky vektorů gradientu považujeme za střed oka:

$$S = \arg \max \left\{ \frac{1}{N} \sum_{i=1}^N (d_i^T \cdot g_i)^2 \right\}, \quad (3.15)$$

$$d_i = \frac{x_i - S}{\|x_i - S\|_2}. \quad (3.16)$$

Výsledek můžeme vidět na obrázku 22, kde vlevo vidíme oční oblast na vstupu. Uprostřed je vizuální znázornění průniků vektorů gradientu a vpravo je již výsledný obrázek se zaznačeným středem duhovky.



Obrázek 22 - Vlevo oko na vstupu. Uprostřed místo s nejpočetnějším průnikem gradientů. Vpravo oko se středem.

3.5 Nalezení duhovky

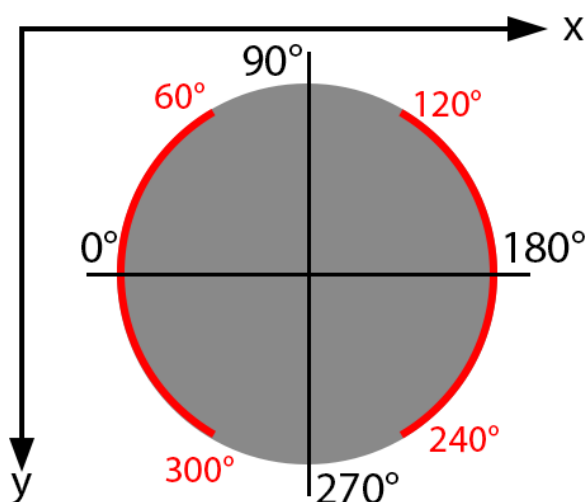
Nalezení duhovky je velmi komplexním problémem. Řada článků, které jsem studoval, používá nějakou podobu předzpracování obrazu a poté Houghovu transformaci kruhu. Tento způsob jsem testoval a došel jsem k závěru, že pro mé řešení je ho vhodné upravit. Využil jsem nalezeného středu oka, o čemž pojednává kapitola 3.4, a modifikoval jsem Houghovu transformaci tak, že na základě středu a poloměrů hledá kružnici s nejvyšším průměrnou velikostí vektoru gradientu.

Před samotným nalezením duhovky jsem opět přistoupil k optimalizaci obrazu a redukoval jsem šum pomocí Gaussova filtru. Následně jsem vynásobil intenzitu celého obrazu, abych zvýraznil tvar duhovky oproti kůži a bělmu v oční oblasti. Tuto techniku jsem popsal v kapitole 3.4.1. Poté jsem spočítal derivaci ve směru osy x pro každý bod v obraze. Na obrázku 23 uprostřed můžeme vidět velikost vektoru gradientu. Když si podrobněji prohlédneme obrázek, tak zjistíme, že se v něm nacházejí dvě výrazné části. Jednak je to obočí a za druhé duhovka s tmavou částí očních víček.



Obrázek 23 - Vlevo obrázek oka na vstupu. Uprostřed derivace ve směru osy x a y . Vpravo derivace jen ve směru osy x .

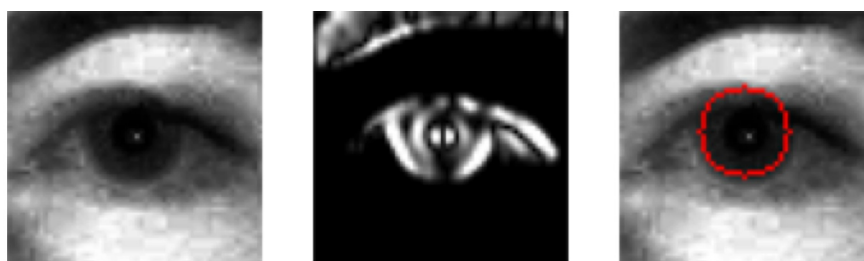
Jelikož je horní okraj víčka velmi výrazný, rozhodl jsem se použít jen derivaci ve směru osy x , což zaručí, že nejvýraznější budou hrany, u kterých se mění intenzita v horizontálním směru. To je v našem případě levý a pravý oblouk duhovky (obr. 23 vpravo). Nenastane tak situace, že by intenzita horního víčka byla natolik velká, aby v jeho oblasti byla detekována kružnice. Dále určíme minimální a maximální poloměr kružnice, díky čemuž značně omezíme počet řešení a dosáhneme i vyšší výpočetní rychlosti.



Obrázek 24 - Vytyčené oblasti, na kterých se zkoumají derivace ve směru osy x .

Samotné nalezení duhovky probíhá tak, že sečteme hodnoty všech derivací ve směru osy x , které leží na vytyčených částech pravého a levého oblouku kružnice (obr. 24). Jednak se tímto řešením vyhneme oblasti víček a za druhé dosáhneme vyšší efektivity. Nyní tedy máme seznam součtů derivací pro každý poloměr, který je ale třeba ještě zprůměrovat, tedy vydělit počtem započítaných derivací. Za kružnici, která nejlépe kopíruje pozici i poloměr duhovky považujeme tu, která má následně nejvyšší průměrnou hodnotu.

Dále jsem tento algoritmus rozšířil o možnost, že se kružnice nehledá pouze ze středu, ale je možné zadat hodnotu, která určí kolik pixelů kolem středu se má taktéž testovat na nalezení duhovky. Pokud námi předtím určený střed není úplně přesný, zadáním tzv. kernelu a jeho dostatečné hodnoty, můžeme i tak najít přesný poloměr kružnice, která kopíruje tvar duhovky. Výsledek postupu popsaného v této kapitole můžeme vidět na obrázku 25.



Obrázek 25 - Vlevo vstupní obrázek oka. Uprostřed derivace ve směru osy x . Vpravo získaný obrys duhovky.

3.6 Nalezení zornice

Stejně tak jako v předchozích kapitolách i nyní přistoupíme k předzpracování obrazu. Pro redukci šumu použijeme Gaussův filtr, jehož použití jsem popsal v kapitole 3.3.1. Pro zvýraznění tmavých částí oka použijeme násobení intenzity pixelů obrázku. Tento postup jsem popsal v kapitole 3.4.1. V ideálním případě bychom postupovali obdobně jako u detekce duhovky. Vypočítali derivaci ve směru osy x pro každý bod v obraze a hledali kružnici s nejvyšším průměrem součtu derivací, která by měla zároveň výrazně menší poloměr než kružnice kopírující duhovku. Jak ovšem můžeme vypořadovat z obrázku 26, nemůžeme se spoléhat na to, že zornice bude vždy natolik výrazná, abychom ji mohli tímto způsobem detekovat. Vlevo na stejném obrázku můžeme jasně vidět duhovku a kousek horního víčka. Avšak změna intenzity v oblasti zornice je natolik malá, že obrys zornice nejde téměř vidět. Kdežto vpravo na stejném obrázku můžeme zornici jednoznačně rozeznat. Pokud výše zmíněné vezmeme v potaz, musíme přijít s jiným řešením.



Obrázek 26 - Vlevo případ, kdy nelze jednoznačně rozeznat zornici. Vpravo již zornici můžeme pozorovat.

Dále budu pracovat pouze s obrazem se zesílenou intenzitou (obr. 27 vlevo). Kružnici, jejíž střed jsem zjistili v kapitole 3.4, a která ideálně kopíruje zornici, najdeme tak, že vypočteme průměrnou hodnotu obsahu intenzity v mezikruží, kde vnitřní kružnice má určený minimální poloměr a vnější se postupně zvětšuje až do maximálního radiusu. Všechny takto získané mezikruží poté vyšetříme a vybereme to, které má nejmenší průměrnou intenzitu. Vnější kružnice z tohoto mezikruží je kružnicí, která ideálně kopíruje zornici oka. Na obrázku 27 můžeme vidět oko po násobení intenzity pixelů a výslednou detekci zornice.



Obrázek 27 - Vlevo oko na vstupu s vynásobenou intenzitou pixelů. Uprostřed výsledek nalezení zornice na stejném obrázku. Vpravo nalezená zornice na původním obrázku oka.

3.7 Detekce polohy očních víček

Posledním praktickým úkolem této práce je detekce polohy očních víček, která se dále dá využít například u systému detekující ospalost řidiče. Tyto systémy dokážou monitorovat frekvenci mrkání, zda řidič nepřivírá oči, či zda dokonce nespí. Pro detekci polohy spodního a vrchního víčka jsem postupoval následovně. Předzpracoval jsem obraz, použil násobení intenzity pixelu, binarizoval obraz pomocí Otsuovy metody a našel konturu odpovídající prostoru mezi vrchní a spodní hranou víček.

Prvním krokem je předzpracovat obraz (obr. 28). Stejně tak jako v kapitole 3.3.1 jsem odebral šum pomocí Gaussova filtru a vynásobil intenzitu obrazu podle vzorce (3.11). Následně jsem binarizoval obraz pomocí Otsuovy metody, kterou jsem popsal v kapitole 3.4.1. Takto získaný obrázek nám dovolí vyhledat konturu, která by odpovídala svým tvarem vnitřní části oka mezi vrchním a spodním víčkem.



Obrázek 28 - Vlevo oko na vstupu. Uprostřed oko po vynásobení intenzity pixelů. Vpravo binarizovaný obraz pomocí Otsuovy metody.

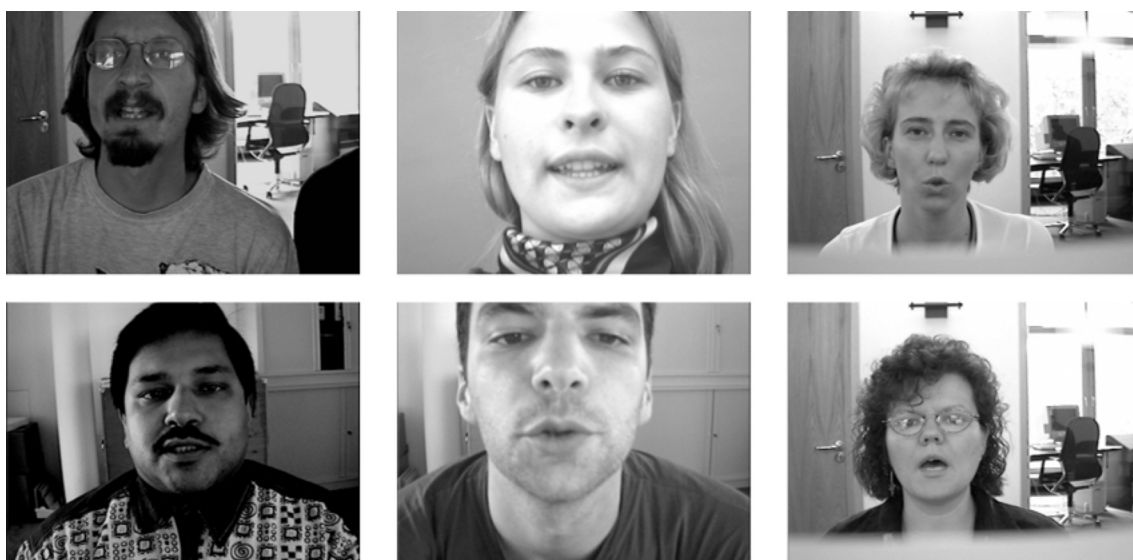
Jak můžeme vidět na obrázku 28, obrys vrchního víčka je dobře zřetelný. Ovšem spodní víčko splývá s bělmem, a tudíž není příliš výrazné. Proto jsem použil pro určení polohy dolního víčka bod na spodní části duhovky. Samozřejmě se stává, že nalezneme více kontur, a tudíž musíme vybrat tu správnou. Zavedl jsem tedy minimální šířku a výšku kontury a předpokládám, že by tato kontura měla mít podobný střed jako střed obrázku. Tímto způsobem vybereme správnou konturu, která nejlépe kopíruje oblast mezi spodním a horním víčkem. U této kontury nalezneme ohraničující box. Jeho vrchní hrana potom vyznačuje oblast horního víčka a jeho spodní hrana označuje oblast dolního víčka.



Obrázek 29 - Vlevo oko na vstupu. Uprostřed binarizovaný obraz. Vpravo výsledné oko se zaznačenou polohou horního a spodního víčka.

4 Experimentální část

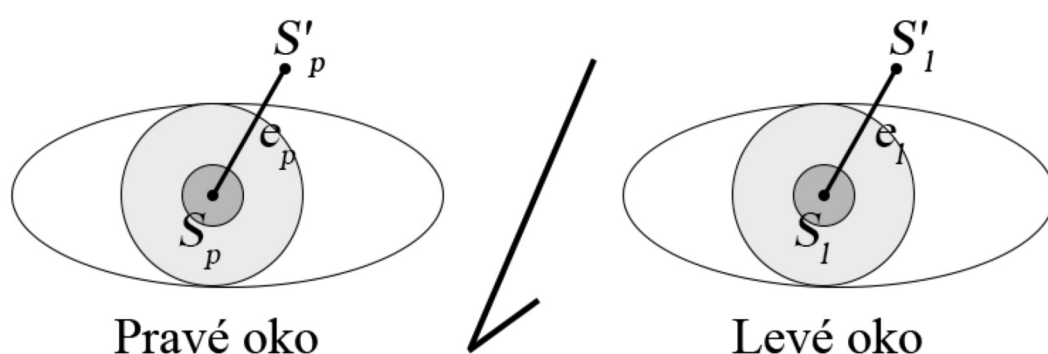
V této kapitole popíší sérii testů, které jsem provedl pro určení efektivnosti algoritmů popsanych v předchozích kapitolách. Nejprve vyhodnotím přesnost nalezení středu oka, poté korektní detekci duhovky, správnost polohy očních víček a na závěr provedu vyhodnocení časové složitosti metod. K testování jsem použil BioID databázi tváří ve verzi 1.2 [37]. Tato datová sada obsahuje 1 521 černobílých obrázků tváří 23 různých osob (obr. 30). Každý z obrázků s rozlišením 384×286 pixelů ukazuje čelní pohled na osobu, u které se mění pozice i postoj. Některé subjekty mají brýle a některým dokonce padají vlasy do oblasti očí. U několika jedinců jsou oči plně skryty za odrazem brýlí, případně mají úplně zavřené oči, nebo hlavu otočenou od kamery, či jsou silně zastíněni. Snímky byly pořízeny v různých prostředích a v rozdílných částech dne, a tudíž demonstrují rozličné světelné podmínky. Pokud vezmeme v potaz výše popsané, můžeme říci, že BioID databáze je jedna z těch, které mají největší rozsah podmínek, za kterých byly jednotlivé fotky pořízeny, a proto práce s ní reflektuje realistické podmínky v okolním světě. Samotná databáze neobsahuje pouze obrázky tváří, ale ke každému snímku je přiložen i textový soubor, ve kterém je zaznamenána pozice pravého a levého středu oka.



Obrázek 30 - Příklady tváří z BioID databáze.

4.1 Metodika testu středu očí

Na stránkách BioID databáze [37] doporučují pro vyhodnocení efektivnosti určení středu oka postup, který je naznačen na obrázku 31. Nejprve je třeba najít středy očí metodou navrženou v kapitole 3.4 a změřit vzdálenosti mezi vlastním výsledkem a manuálně zadanými hodnotami poskytnutými v BioID databázi. Tímto postupem obdržíme dvě vzdálenosti, ze kterých vybereme tu větší a vydělíme ji vzdáleností mezi skutečnými středy očí, čímž se měření stane nezávislé na velikosti obličeje.



Obrázek 31 - Schéma očí, skutečného a předpokládaného středu a vzdáleností mezi nimi.

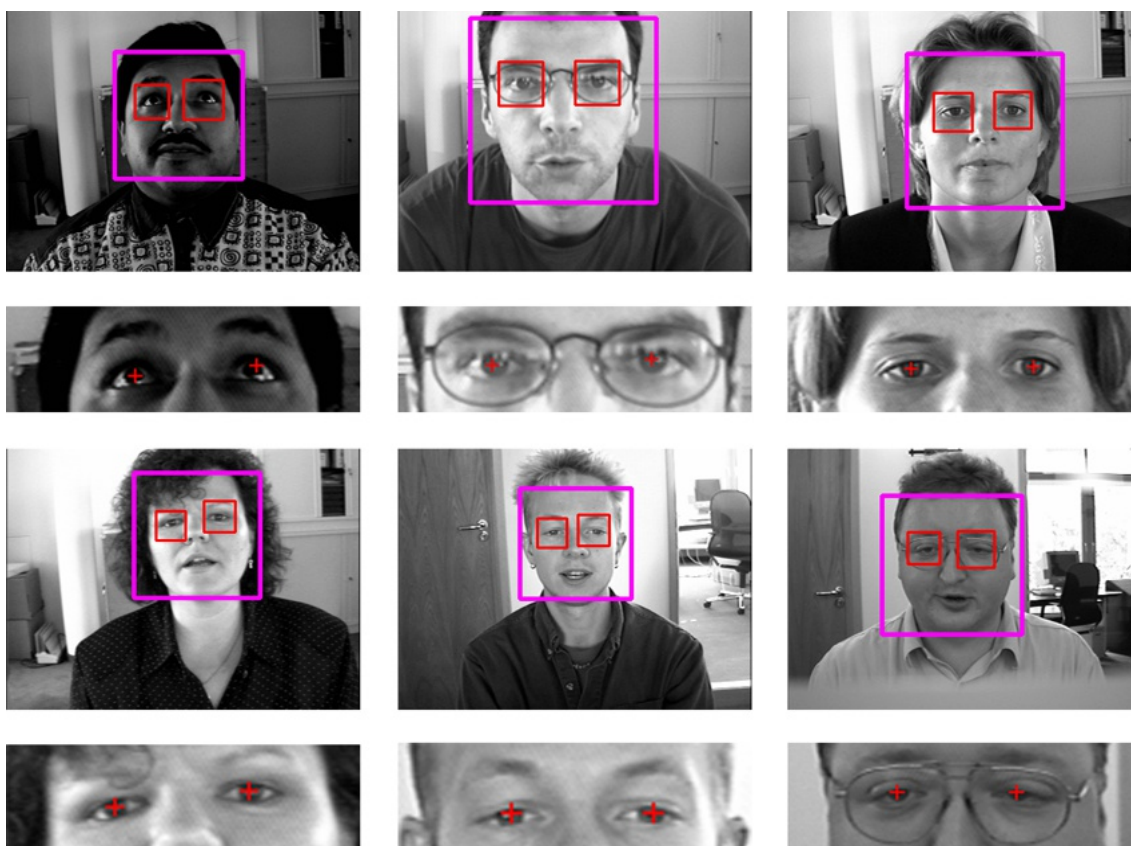
Výše zmíněný postup představil Oliver Jesorsky a kolektiv [38] a je definován jako:

$$e \leq \frac{1}{d} \max(e_l, e_p), \quad (4.1)$$

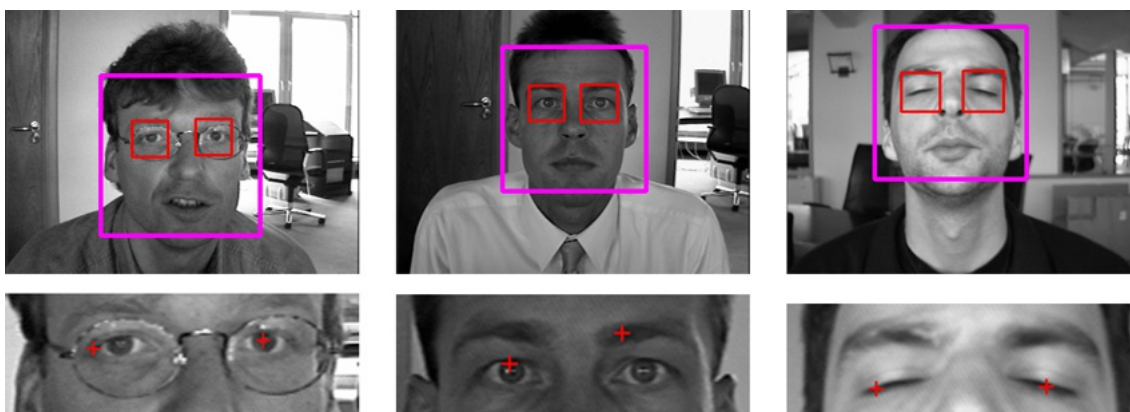
kde e_p , e_l jsou vzdálenosti mezi předpokládanými středy S'_p , S'_l a skutečnými středy S_p , S_l pravého a levého oka a d je vzdálenost mezi skutečnými středy S_p a S_l . Díky výše zmíněným skutečnostem můžeme určit rozptyl nepřesnosti na základě velikosti e . Pokud je $e \leq 0,25$, pak nepřesnost je zhruba rovna vzdálenosti mezi středem oka a rohem oka. Pokud je $e \leq 0,10$, pak nepřesnost odpovídá cca průměru duhovky, a pokud je $e \leq 0,05$, pak je velikost nepřesnosti zhruba rovna průměru zornice.

4.1.1 Test detekce středu duhovky s korekcí stínů v oční oblasti

Tato metoda, která byla popsána v kapitole 3.4.1, byla testována na výše zmíněné BioID databázi a některé z výsledků můžeme vidět na obrázku 32 a 33. Na prvním zmíněném obrázku vidíme sérii obličejů, u které byla detekce úspěšná. Tato metoda dokáže najít střed oka nejen na obrázcích bez rušivých elementů, ale také i na obličejích s brýlemi, u kterých se neodráží zdroj světla v oblasti duhovky a zornice. Dále můžeme pozorovat, že metoda je rezistentní vůči zhoršeným světelným podmínkám a proti změně pozice osoby před kamerou. Naopak na obrázku 33 se nachází výběr obličejů, u kterých proběhla detekce nekorektně. Jsou to zejména situace, kdy má osoba zavřené oči, nebo se v oblasti duhovky nachází silný odlesk světla, který způsobí, že výsledný výběr kontury odpovídající středu oka je nesprávný. Další možný důvod pro selhání algoritmu představuje kombinace odlesku světla v zornici a výrazné obočí. Pokud nastane tato situace, kontura v oblasti zornice má příliš malý obsah, a tudíž ji metoda nepovažuje za správnou.

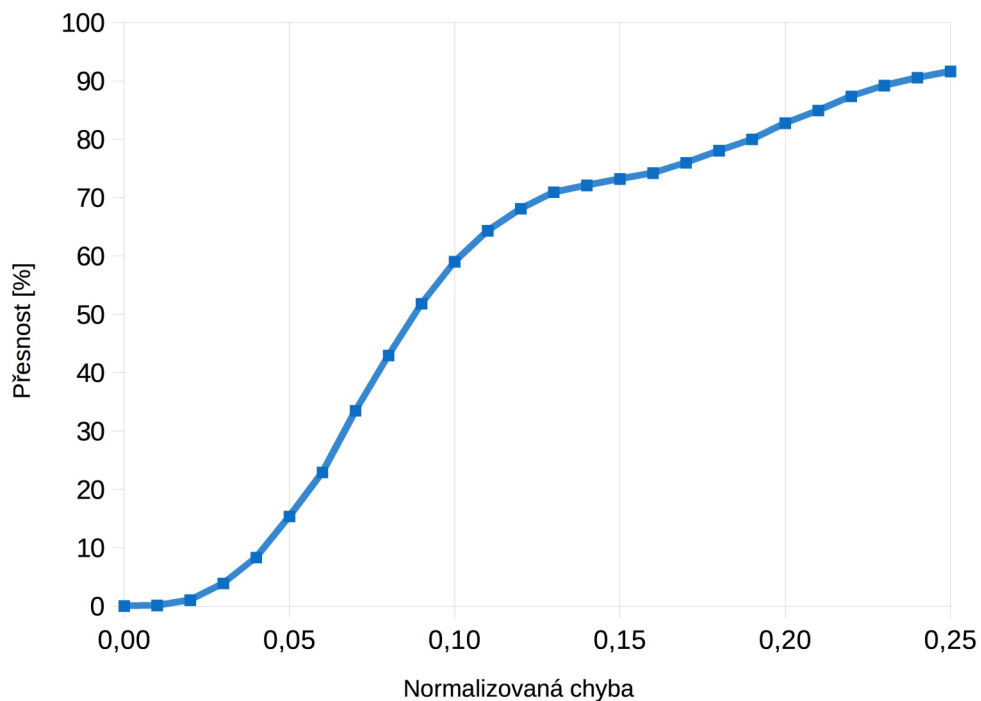


Obrázek 32 - Obličej, u kterých byla detekce středu oka korektně.



Obrázek 33 - Obličeje, u kterých algoritmus mylně vyhodnotil pozice středu oka.

Graf na obrázku 34 ukazuje úspěšnost algoritmu popsaného v kapitole 3.4.1 pro různě veliké e . Jak bylo uvedeno v předchozí kapitole, e je normalizovaná vzdálenost mezi skutečným a předpokládaným středem duhovky. V 15% případů byla velikost $e \leq 0,05$. To znamená, že pouze u 15% obličejů byla vzdálenost domnělého a skutečného středu oka rovna průměru zornice nebo menší. V 60% potom bylo $e \leq 0,10$, což se rovná průměru duhovky. Další význačné hodnoty jsou pak $e \leq 0,15$ v 73%, $e \leq 0,20$ v 83% a $e \leq 0,25$ v 92%. Pokud bychom si zvolili jako hranici úspěšnosti a neúspěšnosti odchylku o velikosti $e \leq 0,9$, která odpovídá odchylce o velikosti, jež je menší než průměr duhovky, tak bychom mohli říci, že metoda detekce středu duhovky s korekcí stínů v oční oblasti je v 52% úspěšná.



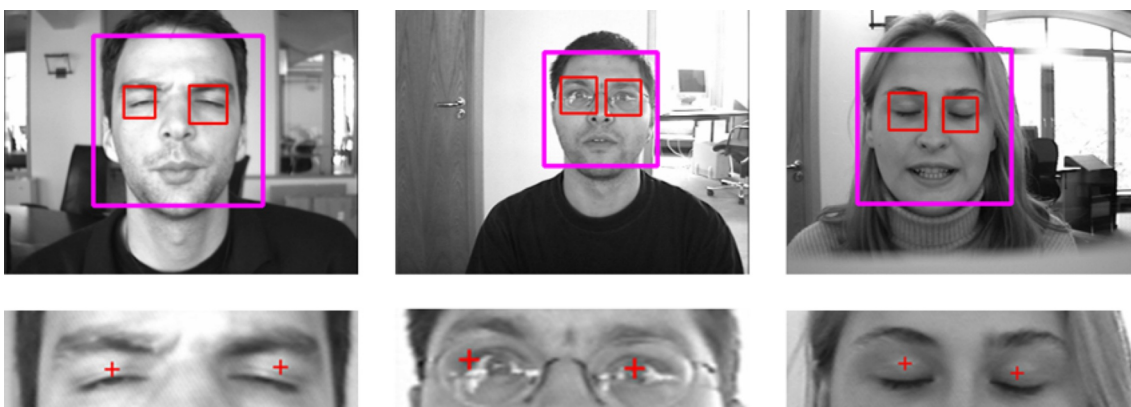
Obrázek 34 - Přesnost a normalizovaná chyba u metody detekce středu duhovky s korekcí stínů v oční oblasti.

4.1.2 Test lokalizace středu oka pomocí gradientů

Obrázek 35 ukazuje výběr ze správných detekcí středu duhovky metodou lokalizace středu oka pomocí gradientů, která je popsána v kapitole 3.4.2. Jak můžeme vidět, tak tento postup je účinný nejen při frontální poloze obličeje, ale taktéž i na pootočeném obličeji. Stejně tak dokáže detekovat střed duhovky, pokud osoba nosí brýle nebo má lehce přivřené oči. Můžeme říci, že navržený postup je všestranný a dokáže rozeznat střed oka i za zhoršených světelných podmínek. Naopak tato metoda selhává v rozpoznávání středu oka u osob se zavřenými očima (obr. 36), nebo pokud má subjekt brýle, které silně reflektují světlo. Pokud se v oblasti duhovky vyskytuje odlesk světelného zdroje, dochází k tomu, že pixely v tomto místě nepřispívají k výslednému součtu průniků vektorů gradientu, a tudíž je možné, že výsledný střed oka bude detekovaný v jiné, nekorektní oblasti.

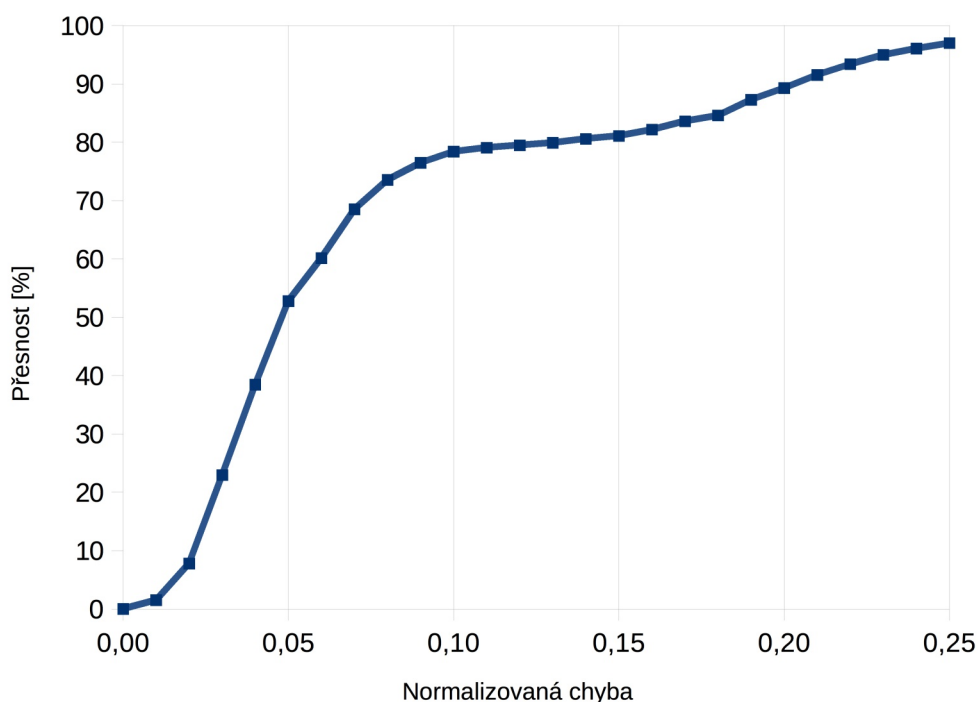


Obrázek 35: Výběr úspěšných detekcí středů očí pomocí metody lokalizace středu oka pomocí gradientů.



Obrázek 36 - Neúspěšná detekce středu oka při použití metody lokalizace středu oka pomocí gradientů. Jedná se zejména o případy, kdy osoba přivírá oči nebo je má dokonce zavřené či má brýle s výrazným odleskem světla.

Na obrázku 37 můžeme vidět graf, který ukazuje rozdílnou úspěšnost detekce středu oka na základě rovnice (4.1), pro různě veliké e . Jak lze vyčíst z grafu pro $e \leq 0,05$ je úspěšnost 53%, což znamená, že více než polovina detekovaných středu očí má odchylku do velikosti průměru zornice. Pro $e \leq 0,10$ bylo detekováno 78% očních středů, což značí, že více jak $\frac{3}{4}$ detekovaných středů má odchylku menší než je průměr duhovky. Dalšími význačnými hodnotami jsou $e \leq 0,15$ s úspěšností 81%, $e \leq 0,20$ s úspěšností 89% a $e \leq 0,25$ s 96%. Pro určení, zda je tato metoda úspěšná, či neúspěšná jsem pro lepší porovnání výsledků zvolil, stejně jako v minulé kapitole, hodnotu $e \leq 0,9$, což značí odchylku o velikosti menší než je průměr duhovky. V tomto případě byla metoda korektní v 76% případů, což je o 24% lepší výsledek než u metody popsané v kapitole 4.1.1.



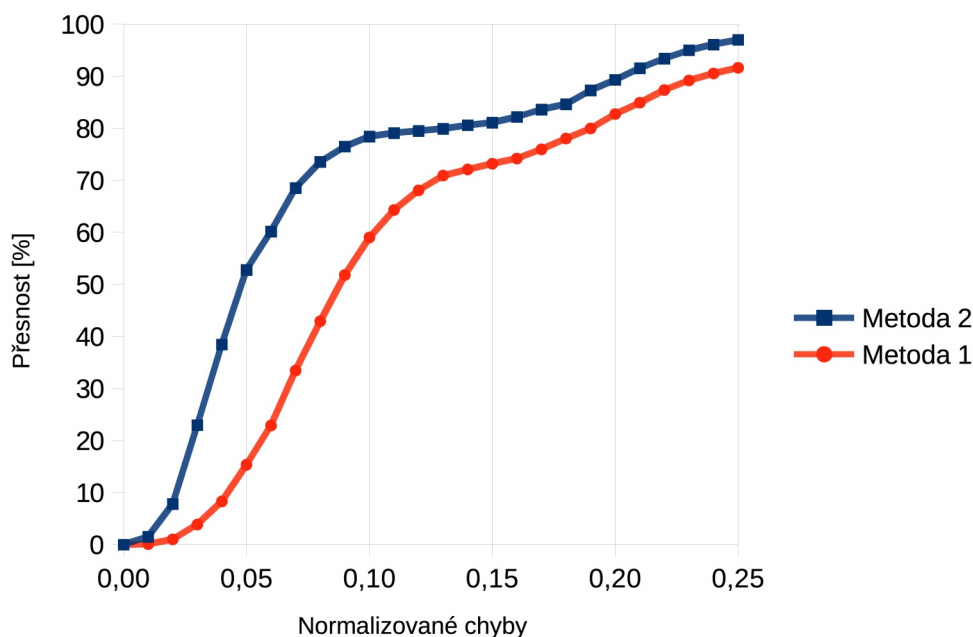
Obrázek 37 - Přesnost a normalizovaná chyba u metody lokalizace středu oka pomocí gradientů .

4.1.3 Porovnání metod pro detekci středu oka

Na obrázku 38 můžeme vidět graf, který reprezentuje přesnost jak metody detekce středu duhovky s korekcí stínů v oční oblasti, tak i metody lokalizace středu oka pomocí gradientů.

Z grafu můžeme vyčíst, že druhá zmíněná metoda dosahuje větší přesnosti v celém rozsahu normalizované chyby e . Rozdíl je zejména významný pro odchylku $e \leq 0,05$, která svou velikostí odpovídá průměru zornice. Pro první představenou metodu je pro toto e úspěšnost pouze 15%, kdežto pro druhou je to 53%. Pro tuto hodnotu e je tedy druhá metoda o celých 38% úspěšnější. Pro hodnotu $e \leq 0,15$ je druhá metoda úspěšnější o 8% a konečně pro hodnotu $e \leq 0,25$ je druhá metoda úspěšnější o 5%. Jak je patrné ze zmíněných dat a z grafu, tak první metoda se procentuálně přesností přibližuje k metodě druhé se zvětšováním velikosti normalizované chyby e .

Obě metody úspěšně detekují střed očí, pokud je osoba v různé vzdálenosti od kamery nebo je od ní pootočená. Taktéž jsou obě metody úspěšné, pokud má subjekt brýle a v oblasti duhovky a zornice se nenachází ostrý odraz ze světelného zdroje. Pokud je ovšem odraz výrazný a ani postup pro odebrání reflexe z očí popsany v kapitole 3.3 ho neeliminuje, obě metody vykazují značně zhoršenou schopnost přesné detekce středu oka. Navíc první popsaná metoda trpí nepřesností pokud má testovaná osoba výrazné obočí a v oblasti duhovky se nachází nějaký rušivý element, jako například i slabý odraz světla, který znehodnotí hledanou konturu a algoritmus dá poté přednost kontuře, která má větší obsah, ale nekoresponduje s oblastí obočí. Pokud bychom vzali v potaz hranici $e = 0,09$, kterou jsem stanovil pro obecné určení zda je metoda úspěšná, či neúspěšná, tak metoda lokalizace středu oka pomocí gradientů je oproti metodě detekce středu duhovky s korekcí stínů v oční oblasti úspěšnější o 24%.



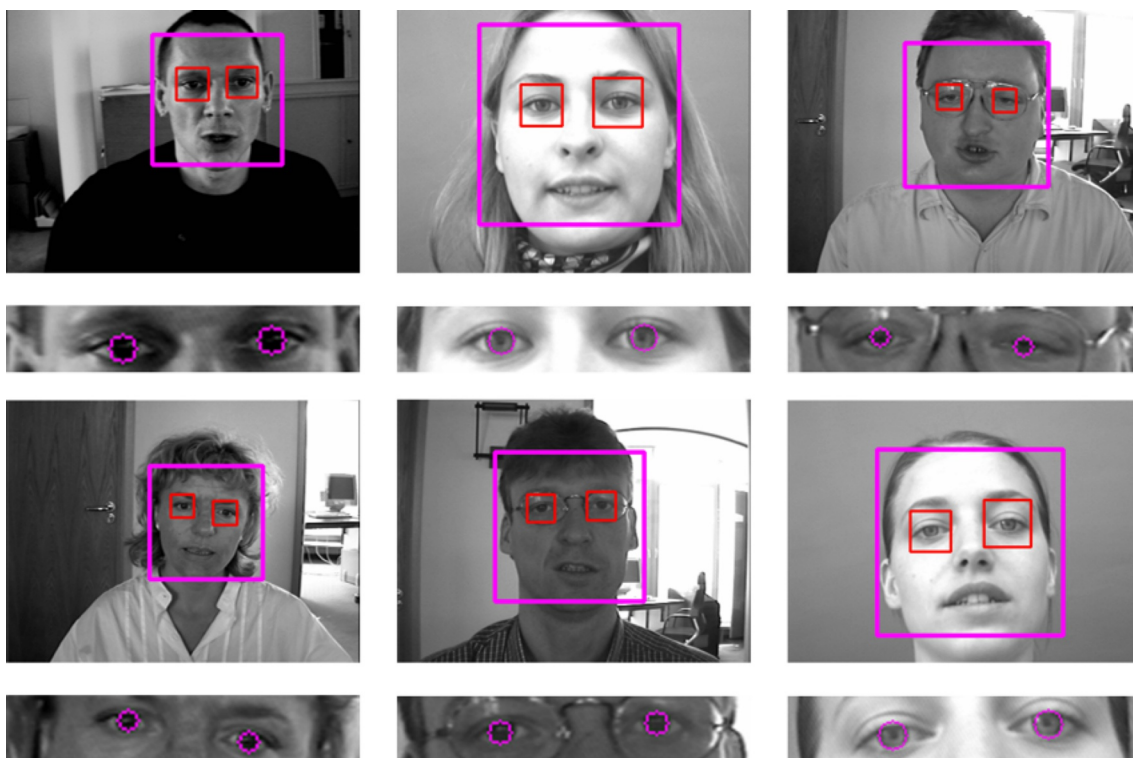
Obrázek 38 - Metoda 1 reprezentuje postup popsany v kapitole 3.4.1, tedy detekci středu duhovky s korekcí stínů v oční oblasti. Metoda 2 představuje lokalizaci středu oka pomocí gradientů, která byla popsána v kapitole 3.4.2.

4.2 Test velikosti průměru duhovky

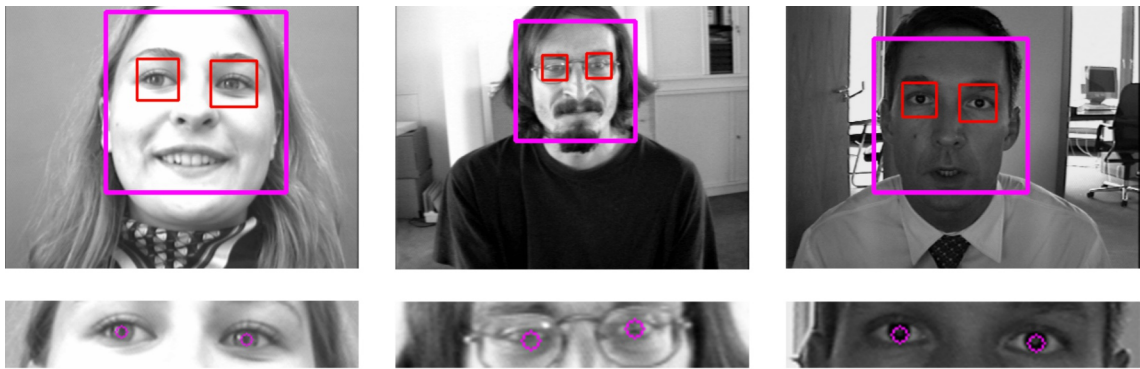
Pro ověření správnosti mého postupu určení polohy a průměru duhovky jsem taktéž použil BioID databázi. Ta ovšem neobsahuje informaci o správné velikosti duhovky. Musel jsem tudíž naprogramovat pomocnou aplikaci, kterou jsem rozšířil využitelnost této databáze. Program dovoluje načíst obrázek, ruční určení bodu na okraji duhovky a zaznamenání polohy očních víček, což využiji v následující kapitole. Následně se tyto údaje zapiší do textového souboru. Využil jsem tedy informaci o bodu na okraji duhovky a souřadnici středu oka poskytnutou BioID databází a vypočetl poloměr duhovky, proti kterému jsem testoval moje řešení popsané v kapitole 3.5. I přes naprogramování pomocné aplikace, je tento způsob relativně časově náročný, a proto jsem zpracoval vzorek náhodně 100 vybraných obrázků, u kterých proběhla detekce středu oka s $e < 0,05$, což jsou případy, u nichž byla odchylka menší než je cca průměr zornice. Zůstal jsem u používání normalizované chyby podle vzorce (4.1), kde ovšem nyní e_p , e_l představují poloměry pravé, respektive levé duhovky.

Na obrázku 39 můžeme vidět výběr ze sady obličejů, u kterých byla úspěšně provedena detekce duhovky. Jak je patrné z testovacích obrázků, metoda dokáže lokalizovat duhovku nejen

na obličejích bez rušivých elementů, ale také v případě pokud má subjekt brýle nebo i za zhoršených světelných podmínkách. Dále je algoritmus úspěšný i v případě, že osoba je potočená od kamery, a nebo se od ní nachází v různých vzdálenostech. Naopak na obrázku 40 můžeme vidět situaci, kdy metoda neúspěšně našla kružnice kopírující duhovku. Jedná se zejména o případy, kdy algoritmus detekoval duhovku v oblasti zornice, nebo v situaci, kdy brýle a odlesk v nich, znemožnil úspěšnou činnost metody. Další příčinou může být nepřesné určení středu oka, což zapříčiní, že nalezená kružnice nereflektuje se skutečnou polohou duhovky.

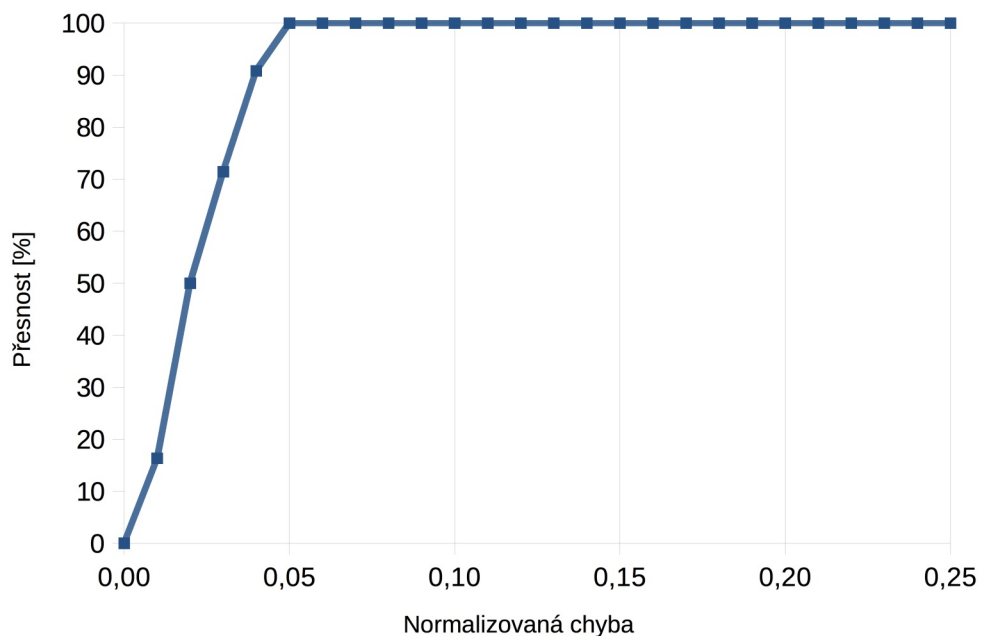


Obrázek 39 - Výběr úspěšných detekcí duhovky.



Obrázek 40 - Příklady obrázků, u kterých detekce duhovky a její velikosti neproběhla úspěšně.

V grafu na obrázku 41 můžeme vidět, že algoritmus při $e \leq 0,05$ byl úspěšný ve 100%. To znamená, že metoda detekovala ve všech případech poloměr s odchylkou menší nebo rovnou průměru zornice. Pro výsledné určení, v kolika procentech metoda byla, či nebyla úspěšná, jsem zvolil hranici $e \leq 0,04$, která odpovídá velikosti menší než je průměr zornice. U mnoha obrázků z testované sady velikost zornice odpovídá pouze několika pixelům, a tudíž je tato hranice dostatečně přesná. Podle výše zmíněného můžeme tedy říci, že metoda pro detekci duhovky má úspěšnost v 91%.

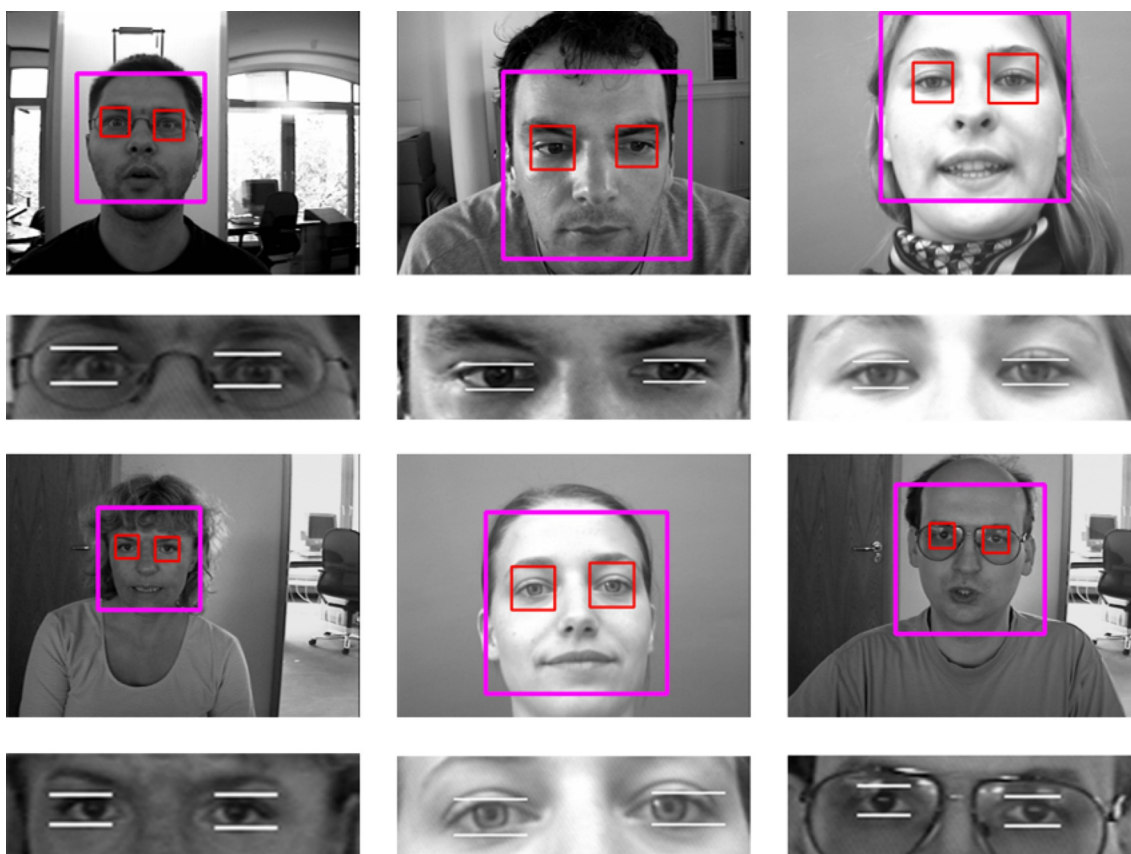


Obrázek 41 - Přesnost a normalizovaná chyba u metody detekce duhovky.

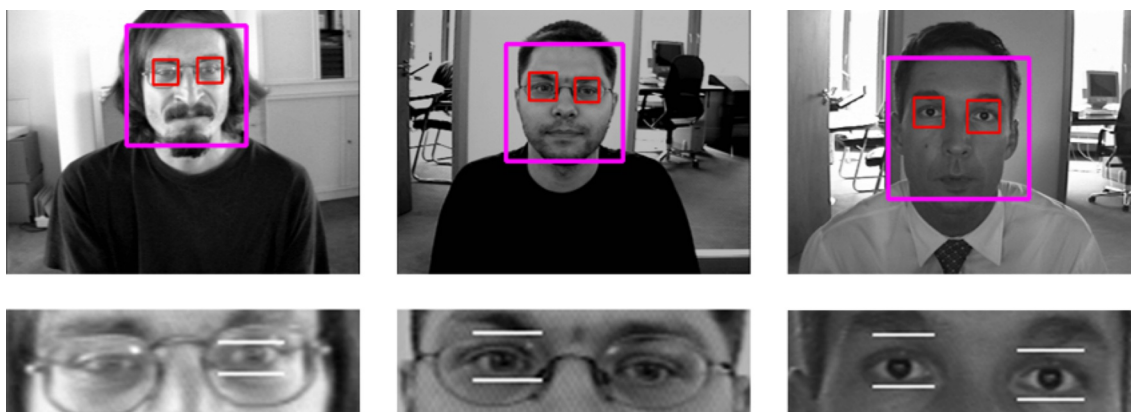
4.3 Test detekce polohy očních víček

Stejně jako v minulé kapitole, i při testování polohy očních víček, jsem použil vzorek 100 obrázků z BioID databáze obličejů. Pomocí programu popsaného v předchozí kapitole jsem ručně zaznamenával polohu spodního a horního víčka pro obě oči, kterou jsem dále porovnal s výsledky metody, kterou jsem představil v kapitole 3.7. Z podstaty oné metody vyplývá, že jsem porovnával pouze vertikální souřadnici, která odpovídá bodu na horním či spodním víčku. Pro zachování homogenity testovacích metod jsem i nyní pracoval s normalizovanou chybou podle vzorce (4.1), kde jsem změnil význam e_p a e_i , které nyní reprezentují vzdálenost domnělé souřadnice y horního, respektive dolního víčka od skutečné souřadnice, kterou jsem ručně zaznamenal ve výše zmíněném programu.

Jak můžeme vyzorovat z obrázku, který reprezentuje výběr ze vzorku korektních detekcí víček, mnou navržená metoda si poradí s proměnnou polohou obličeje a její vzdáleností od kamery. Taktéž je úspěšná v případě, že subjekt má nasazené brýle, ve kterých se nenachází odraz světla ze světelného zdroje. Obrázek 43 je výběrem nepřesných detekcí. Zejména se jedná o případy, kdy má osoba přivřené nebo zavřené oči. Dalším faktorem, se kterým si metoda často neporadí je případ, kdy má subjekt brýle, přes které se světlo láme takovým způsobem, že znehodnotí oblast za nimi a detekovaná kontura poté neodpovídá oblasti mezi horním a spodním víčkem oka.



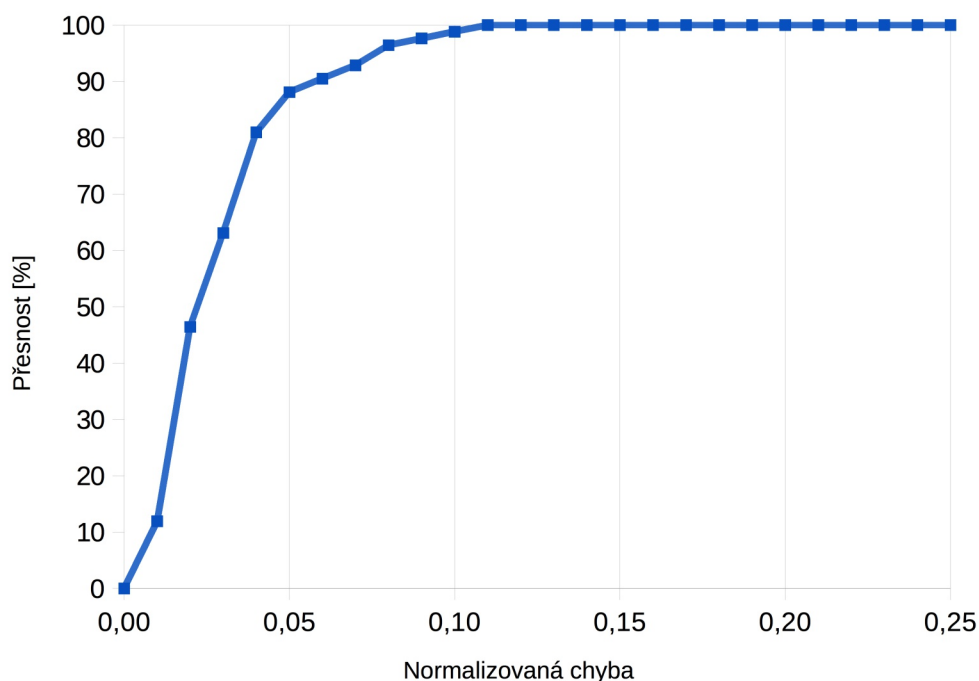
Obrázek 42 - Případy úspěšné detekce polohy očních víček.



Obrázek 43 - Vzorek obrázků neúspěšných detekcí očních víček.

Graf na obrázku 44 vypovídá o přesnosti představené metody pro určení polohy očních víček. Odchylka o velikosti zornice nebo menší, tedy $e \leq 0,05$, nastane v 88% případů. Přesnost 99% je pro $e \leq 0,10$, tedy pro velikost průměru duhovky. Odchylka o velikosti $e = 0,11$ potom

obsáhne 100% testovaných obrázků. Pro konečné určení úspěšnosti považuji za dostatečně velké $e = 0,06$. Jedná se tedy o odchylku jen o něco větší než je průměr zornice. Takto velická nebo menší normalizovaná chyba nastala v 90% případů.



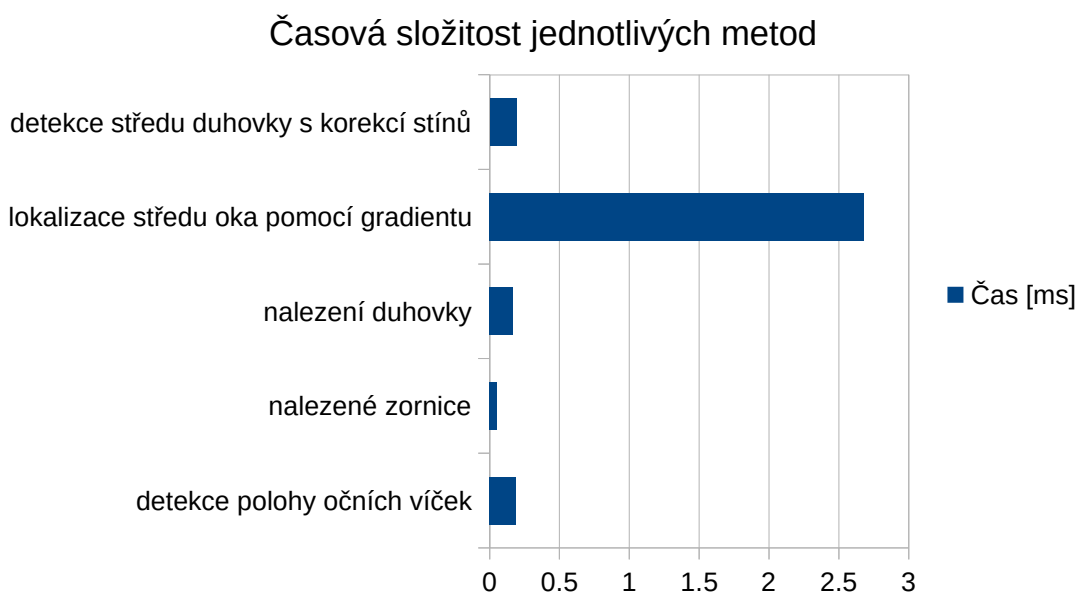
Obrázek 44 - Přesnost a normalizovaná chyba u metody detekce očních víček.

4.4 Časová složitost

V této kapitole představím počítač, na kterém jsem pracoval a popíši časovou složitost jednotlivých metod. Postupně to bude detekce středu duhovky s korekcí stínů v oční oblasti, lokalizace středu oka pomocí gradientu, detekce duhovky, lokalizace zornice a jako poslední detekce polohy očních víček. Pro vyhodnocení časové složitosti, jsem stejně jako u testování efektivnosti představených metod, použil BioID databázi s 1 521 obrázky obličejů s rozlišením 384×286 . Počítač na kterém jsem pracoval je Apple MacBook Pro s procesorem Intel Core i5 s taktom 2,4 GHz a s pamětí RAM o velikosti 4 GB.

Na 1 521 obrázcích z BioID databáze bylo úspěšně detekováno 1 469 obličejů a 2 510 očí. Dále uvedené časové údaje jsou tedy získané na takto velikém vzorku. Je třeba zmínit, že čas jsem měřil pouze pro jednotlivé metody na již detekovaných očních oblastech. Výška a šířka

takové oblasti měla průměrně hodnotu 34 pixelů. Pro metodu detekce středu duhovky s korekcí stínů v oční oblasti je celkový čas zpracování 0,5 s, tudíž průměrný čas činí 0,193 ms. Pro lokalizace středu oka pomocí gradientu je souhrnný čas zpracování 6,7 s a průměrný čas na jeden obrázek má hodnotu 2,68 ms. Můžeme tedy říci, že první metoda pracuje rychleji, ale jak je patrné z kapitoly 3.4.1 její přesnost nedosahuje tolika úspěšných hodnot jako metoda druhá. Nalezení duhovky vyžaduje pro zpracování 2 510 vzorků očí celkový čas 0,4 s, což je průměrně 0,169 ms. Pro nalezené zornice činí souhrnný čas 0,13 s, což se rovná průměrnému času na jeden snímek 0,052 ms. A nakonec celkový čas pro detekci polohy očních víček u 2 510 očí činil 0,48 s, což znamená, že průměrný čas pro nalezení dvou očních víček na jednom oku je 0,190 ms. Pokud zhodnotíme hodnoty průměrných časů, můžeme říci, že navržené metody jsou vhodné pro hledání očních částí v reálném čase.

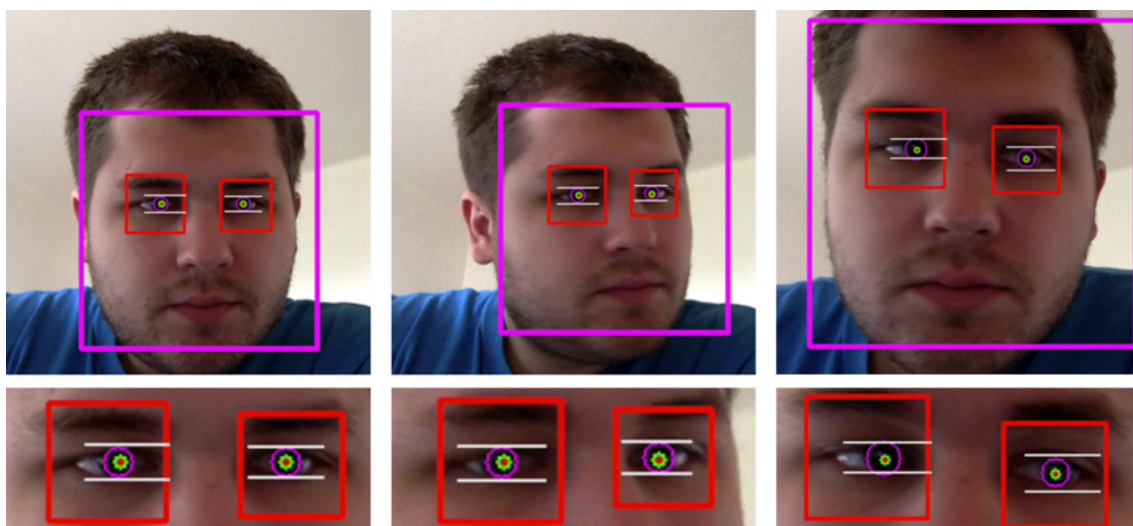


Obrázek 45 - Časová složitost jednotlivých metod v milisekundách.

4.5 Test na videu v reálném čase

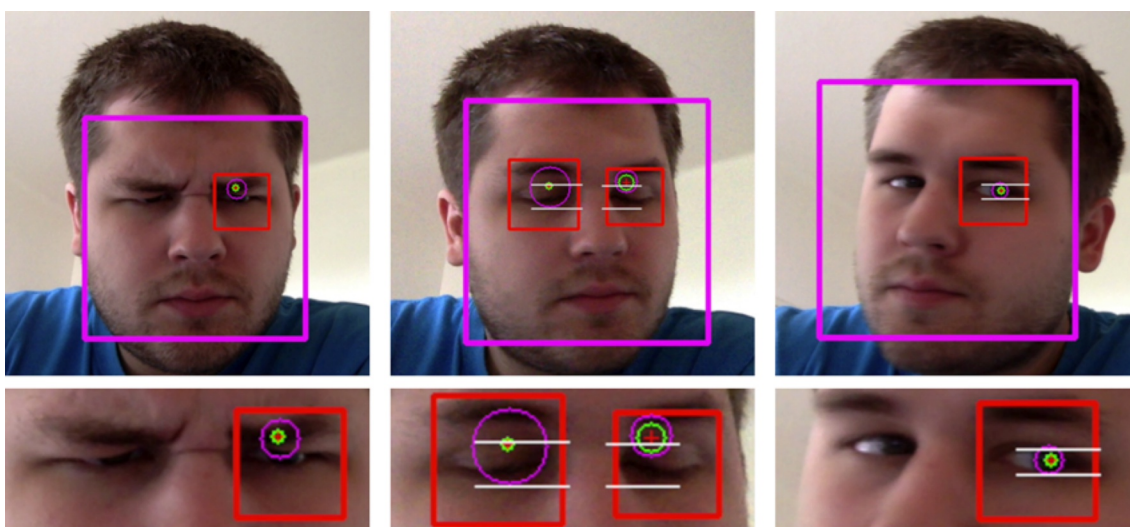
V poslední kapitole experimentální části popíši, jak jsem natočil krátké video a zhodnotím, jak korektně proběhla detekce očních částí v reálném čase. Video jsem natáčel přes vestavěnou kameru stejného notebooku, který jsem použil pro vyhodnocení časové složitosti, tedy Apple MacBook Pro. Tento model má ve výbavě HD kameru, tedy kameru s rozlišením 1280 × 720

pixelů. Video však bylo pořízeno s rozlišením 640×360 . Ve videosekvenci, která byla nahrána za běžného osvětlení, je zachycena detekce obličeje, očních oblastí, středu očí, duhovky, zornice a očních víček. Měnil jsem pozici před kamerou, otáčel jsem hlavou a střídal jsem výrazy ve tváři. Celkový čas videa je 35 sekund a skládá se z 247 snímků.



Obrázek 46 - Příklady správných detekcí z videa v reálném čase.

Pro lepší popsání úspěšnosti jsem připravil obrázek 46 a obrázek 47. Na prvním zmíněném obrázku můžeme vidět vybrané snímky z videa, u kterých byla detekce úspěšná. Jak můžeme vyzorovat, tak i pootočená poloha subjektu neovlivní výsledek detekce očních částí. Na obrázku 47 můžeme naopak spatřit případy, kdy byla detekce neúplná nebo špatná. Na obličeji vlevo a uprostřed můžeme vidět případy, kdy přivřené, či zcela zavřené oči, způsobily špatnou detekci. Na obličeji vpravo pak můžeme vidět případ, kdy extrémní rotace od kamery zapříčinila, že nedošlo k detekci jednoho oka, a tudíž ani k následovné detekci duhovky nebo očních víček.



Obrázek 47 - Příklady nekorektních detekcí z videa v reálném čase.

Na tomto 35 sekundovém videu, které se skládá z 247 snímků, jsem chtěl ukázat, že metody, které jsem zpracoval nebo navrhl, jsou vhodné pro detekci očních částí, zornice, duhovky či očních víček. V tabulce 4.1 jsou uvedeny přesnosti jednotlivých metod pro detekci očních částí. Na 247 snímcích bylo detekováno 225 obličejů, což znamená, že přesnost detekce obličeje dosáhla 91%. Na 225 obličejích bylo detekováno 190 párů očních oblastí. Za správnou detekci očních regionů jsem považoval pouze situaci, kdy byly detekovány obě oční oblasti. Úspěšnost detekce očí je tedy 84%. Na 190 správně detekovaných očních regionech proběhla korektně detekce u 84% středů očí. Pro nalezení středu jsem použil metodu lokalizace středu pomocí gradientů, kterou jsem popsal v kapitole 3.4.2. Duhovka byla lokalizovaná s úspěšností 93% a zornice s úspěšností 86%. Poloha očních víček byla vyhodnocena správně v 88% případech.

	Obličej	Oční oblasti	Středů očí	Duhovky	Zornice	Oční víčka
Počet detekcí	225	190	159	176	163	168
Přesnost	91,09%	84,44%	83,68%	92,63%	85,79%	88,42%

Tabulka 4.1: Přesnost detekce částí očí v experimentálním videu.

5 Závěr

Tématem diplomové práce bylo navrhnout a implementovat metody, které lze použít pro detekci očních částí. Zejména se pak jedná o nalezení očních oblastí, duhovky a zornice oka či lokalizaci očních víček. Dále jsem tyto metody podrobně experimentálně otestoval a stručně představil, v jakém odvětví by mohla diplomová práce nalézt uplatnění.

V rámci této práce jsem splnil výše zmíněné požadavky a podrobně jsem je popsal. Lokalizoval jsem obličej v obraze pomocí Haarových kaskádových klasifikátorů a zevrubně jsem popsal a na obrázcích ukázal, na jakém principu tento postup funguje. Dále jsem představil metodu, kterou lze odebrat odraz světelného zdroje z duhovky a zornice. Tato metoda využívá binarizaci obrazu a techniku, která se jmenuje image inpainting, která se běžně používá pro restaurování fotografií v digitálním formátu. Prezentoval jsem dvě metody, které dokáží lokalizovat střed oka a fungují na odlišném principu. První využívá binarizaci obrazu pomocí Otsuovy metody a hledání kontur v oblasti zornice a oka, kdežto druhá hledá místo v obraze, kam směřuje nejvíce vektorů gradientu. Následně jsem pomocí upravené Houghovy transformace provedl detekci duhovky a zornice a určil kružnici, ideálně kopírující jejich polohu a poloměr. Nakonec jsem pomocí binarizace obrazu a hledání kontury, která se nachází mezi horním a spodním víčkem, určil polohu očních víček.

Všechny výše uvedené postupy a metody jsem implementoval v jazyce C++ a přiložil k této diplomové práci na DVD. Dále jsem detailně otestoval funkčnost metod na vzorku 1 500 obrázků z BioID databáze. Každý obrázek měl rozlišení 384×286 pixelů a vystupovalo na nich 23 různých osob, jejichž fotky byly pořízeny v rozdílných časových obdobích, takže reprezentují reálné podmínky světa. Ukázal jsem, že první představená metoda pro nalezení středu oka není příliš vhodná pro reálné použití, ale že druhá metoda má přesnost 76%, tudíž s určitým zpřesněním by se dala použít do běžného praxe. Dále jsem prezentoval úspěšnost detekce duhovky, která dosáhla přesnosti 99%, což je velmi pozitivní výsledek. Jako poslední jsem určil přesnost lokalizace očních víček, která dosáhla hodnoty 92% a taktéž bych si pro ni dokázal představit využití v praxi. U všech výše zmíněných metod jsem zjistil výpočetní čas a doložil, že jsou vhodné pro detekci očních částí v reálném čase.

Systémy pro monitorování řidiče a sledování míry jeho unavenosti se čím dál více prosazují do dnešních aut. S rychlým růstem výkonu mobilních čipů lze již dnes s nejnovějším hardwarem podrobně monitorovat nejen situaci mimo auto, ale taktéž stav řidiče automobilu.

Vzhledem k tomu, že výkon dnešní techniky stále roste a ceny zůstávají příznivé, dokážu si představit, že tyto systémy nebudou součástí jen aut vyšší a vyšší střední třídy, ale stanou se nedílnou součástí téměř každého auta.

Ruku v ruce s výkonem jde i přesnost metod pro detekci očních částí. S rostoucím počtem jader grafických čipů, na které by se dala přesměrovat většina výkonu, roste i přesnost a úspěšnost detekce, nemluvě o možnosti využití kamer s vyšším počtem snímků za sekundu. Jsem si vědom, že například při detekci středu oka pomocí zkoumání pole vektorů gradientu, by bylo možné tento algoritmus distribuovat na více jader a tím ho značně urychlit. Tato myšlenka, může sloužit, jako námět mým následovníkům, kteří eventuálně nastudovali tuto diplomovou práci.

6 Literatura

- [1] VOLVO CAR CORPORATION. *Driver Alert Control (DAC)* [online]. [cit. 2015-03-02]. Dostupné z: <http://support.volvocars.com/uk/cars/Pages/owners-manual.aspx?mc=Y555&my=2015&sw=14w20&article=2e82f6fc0d1139c2c0a801e800329d4e>
- [2] PARK, Kang Ryoung. *A Real-Time Gaze Position Estimation Method Based on a 3-D Eye Model*. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics). 2007, vol. 37, issue 1, s. 199-212. DOI: 10.1109/tsmcb.2006.883426.
- [3] SHIH, S.-W. a J. LIU. *A Novel Approach to 3-D Gaze Tracking Using Stereo Cameras*. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics). 2004, vol. 34, issue 1, s. 234-245. DOI: 10.1109/tsmcb.2003.811128.
- [4] VILLANUEVA, A. a R. CABEZA. *A Novel Gaze Estimation System With One Calibration Point*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2008, vol. 38, issue 4, s. 1123-1138. DOI: 10.1109/TSMCB.2008.926606. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4567549>
- [5] PENTLAND, MOGHADDAM a STARNER. *View-based and modular eigenspaces for face recognition*. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94. IEEE Comput. Soc. Press, 1994, s. 84-91. DOI: 10.1109/CVPR.1994.323814. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=323814>
- [6] CASTRILLÓN, M., O. DÉNIZ, C. GUERRA a M. HERNÁNDEZ. *ENCARA2: Real-time detection of multiple faces at different resolutions in video streams*. Journal of Visual Communication and Image Representation. 2007, vol. 18, issue 2, s. 130-140. DOI: 10.1016/j.jvcir.2006.11.004. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1047320306000782>
- [7] DOBEŠ, M., J. MARTINEK, D. SKOUPIL, Z. DOBEŠOVÁ a J. POSPÍŠIL. *Human eye localization using the modified Hough transform*. Optik - International Journal for Light and Electron Optics. 2006, vol. 117, issue 10, s. 468-473. DOI: 10.1016/j.ijleo.2005.11.008. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0030402606000088>
- [8] PENG YANG, BO DU, SHIGUANG SHAN a WEN GAO. *A novel pupil localization method based on gaboreye model and radial symmetry operator*. 2004 International Conference on Image Processing, 2004. ICIP '04. IEEE, 2004, s. 67-70. DOI: 10.1109/ICIP.2004.1418691. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1418691>
- [9] DAUGMAN, John. *Statistical Richness of Visual Phase Information: Update on Recognizing Persons by Iris Patterns*. International Journal of Computer Vision. vol. 45, issue 1, s. 25-38. DOI: 10.1023/A:1012365806338. Dostupné z: <http://link.springer.com/10.1023/A:1012365806338>
- [10] WILDES, R.P. *Iris recognition: an emerging biometric technology*. Proceedings of the IEEE. vol. 85, issue 9, s. 1348-1363. DOI: 10.1109/5.628669. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=628669>
- [11] XIAOMEI LIU, K.W. BOWYER a P.J. FLYNN. *Experiments with an Improved Iris Segmentation Algorithm*. Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05). IEEE, 2005, s. 118-123. DOI: 10.1109/AUTOID.2005.21. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1544411>
- [12] DAL HO CHO, KANG RYOUNG PARK a DAE WOONG RHEE. *Real-Time Iris Localization for Iris Recognition in Cellular Phone*. Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05). IEEE, 2005, s. 254-259. DOI: 10.1109/SNPD-SAWN.2005.62. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1434897>
- [13] XINHUA FENG, CHI FANG, XIAOQING DING a YOUSHOU WU. *Iris Localization with Dual Coarse-to-fine Strategy*. 18th International Conference on Pattern Recognition (ICPR'06). IEEE, 2006, s. 553-556. DOI: 10.1109/ICPR.2006.725. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1699901>

- [14] KONG, W.K. a D. ZHANG. *Accurate iris segmentation based on novel reflection and eyelash detection model*. Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No.01EX489). IEEE, 2001, s. 263-266. DOI: 10.1109/ISIMP.2001.925384. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=925384>
- [15] JUNZHOU HUANG, YUNHONG WANG, TIENIU TAN a JIALI CUI. *A new iris segmentation method for recognition*. Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. IEEE, 2004, 554-557 Vol.3. DOI: 10.1109/ICPR.2004.1334589. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1334589>
- [16] ZHAOFENG HE, TIENIU TAN, ZHENAN SUN a XIANCHAO QIU. *Robust eyelid, eyelash and shadow localization for iris recognition*. 2008 15th IEEE International Conference on Image Processing. IEEE, 2008, s. 265-268. DOI: 10.1109/ICIP.2008.4711742. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4711742>
- [17] DAUGMAN, John. *New Methods in Iris Recognition*. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics). 2007, vol. 37, issue 5, s. 1167-1175. DOI: 10.1109/TSMCB.2007.903540. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4305270>
- [18] PROENÇA, H. a L.A. ALEXANDRE. *Iris segmentation methodology for non-cooperative recognition*. IEE Proceedings - Vision, Image, and Signal Processing. 2006, vol. 153, issue 2, s. 199-. DOI: 10.1049/ip-vis:20050213. Dostupné z: http://digital-library.theiet.org/content/journals/10.1049/ip-vis_20050213
- [19] DORAIRAJ, V., N.A. SCHMID a G. FAHMY. *Performance evaluation of non-ideal iris based recognition system implementing global ICA encoding*. IEEE International Conference on Image Processing 2005. IEEE, 2005, III-285. DOI: 10.1109/ICIP.2005.1530384. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1530384>
- [20] RAKSHIT, Soumyadip. *Pupil Shape Description Using Fourier Series*. Signal Processing Applications for Public Security and Forensics, 2007. SAFE '07. IEEE Workshop on. S.l.: [s.n.], 2007. DOI: 10.1109/ICIP.2005.1530384.
- [21] DAUGMAN, J. *How iris recognition works*. Proceedings. International Conference on Image Processing. IEEE, 2002, I-33-I-36. DOI: 10.1109/ICIP.2002.1037952. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1037952>
- [22] KHEIROLAHY, Rasoul, Hossein EBRAHIMNEZHAD a Mohammad Hossein SEDAAGHI. *Robust pupil boundary detection by optimized color mapping for iris recognition*. 2009 14th International CSI Computer Conference. IEEE, 2009, s. 170-175. DOI: 10.1109/CSICC.2009.5349260. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5349260>
- [23] MI YOUNG NAM, Phill Kyu RHEE a Chan Suk LEE. *A real time pupil location system using multiple illuminations clustering under varying environment*. 2009 Digest of Technical Papers International Conference on Consumer Electronics. IEEE, 2009, s. 1-2. DOI: 10.1109/ICCE.2009.5012398. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5012398>
- [24] CUI, Jiali, Yunhong WANG, Tieniu TAN, Li MA a Zhenan SUN. *An Iris Recognition Algorithm Using Local Extreme Points*. s. 442. DOI: 10.1007/978-3-540-25948-0_61. Dostupné z: http://link.springer.com/10.1007/978-3-540-25948-0_61
- [25] ARVACHEH, E.M. a H.R. TIZHOOSH. *IRIS Segmentation: Detecting Pupil, Limbus and Eyelids*. 2006 International Conference on Image Processing. IEEE, 2006, s. 2453-2456. DOI: 10.1109/ICIP.2006.312773. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4107064>
- [26] YING-LI TIAN, T. KANADE a J.F. COHN. *Dual-state parametric eye tracking*. Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580). IEEE Comput. Soc, 2000, s. 110-115. DOI: 10.1109/AFGR.2000.840620. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=840620>
- [27] KIN-MAN LAM a HONG YAN. *An Improved Method for Locating and Extracting the Eye in Human Face Images*. Proceedings of 13th International Conference on Pattern Recognition. IEEE, 1996, s. 411-. DOI: 10.1109/ICPR.1996.546980. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=546980>

- [28] YUILLE, Alan L., Peter W. HALLINAN a David S. COHEN. *Feature extraction from faces using deformable templates*. International Journal of Computer Vision. 1992, vol. 8, issue 2, s. 99-111. DOI: 10.1007/BF00127169. Dostupné z: <http://link.springer.com/10.1007/BF00127169>
- [29] VIOLA, P. a M. JONES. *Robust real-time face detection*. Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. 2001. DOI: 10.1109/iccv.2001.937709.
- [30] *OpenCV Face Detection: Visualized* [online]. © 2015 [cit. 2015-01-02]. Dostupné z: <https://vimeo.com/12774628>
- [31] *OpenCV* [online]. © Copyright 2015 [cit. 2014-10-02]. Dostupné z: <http://opencv.org/>
- [32] SOBEL, Irwin. *An Isotropic 3 3 Image Gradient Operator*. č. 2014. Dostupné z: http://www.researchgate.net/publication/239398674_An_Isotropic_3_3_Image_Gradient_Operator
- [33] TELEA, Alexandru. *An Image Inpainting Technique Based on the Fast Marching Method*. Journal of Graphics Tools. 2004, vol. 9, issue 1, s. 23-34. DOI: 10.1080/10867651.2004.10487596.
- [34] RANA, Sohel, Abdul AWAL a Zahidul ISLAM. *Eye Detection And Iris Center Tracking With Eyelashes Occlusion Correction*. Dostupné z: <http://www.acit2k.org/ACIT/2013Proceedings/146.pdf>
- [35] TIMM, Fabian a Erhardt BARTH. *ACCURATE EYE CENTRE LOCALISATION BY MEANS OF GRADIENTS*. Proceedings of the International Conference on Computer Vision Theory and Applications. SciTePress - Science and Technology Publications, 2011-3-5, s. 125-130. DOI: 10.5220/0003326101250130. Dostupné z: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0003326101250130>
- [36] OTSU, Nobuyuki. *A Threshold Selection Method from Gray-Level Histograms*. IEEE Transactions on Systems, Man, and Cybernetics. 1979, vol. 9, issue 1, s. 62-66. DOI: 10.1109/tsmc.1979.4310076.
- [37] *BioID Face Database* [online]. © 2015 [cit. 2015-04-01]. Dostupné z: <https://www.bioid.com/About/BioID-Face-Database>
- [38] JESORSKY, Oliver, Klaus J. KIRCHBERG a Robert W. FRISCHHOLZ. *Robust Face Detection Using the Hausdorff Distance*. <https://www.facedetection.com/downloads/AVBPA01Bio>.