

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE



Vladislav Trnka

Sledování pohybu očí ve VR headsetu XTAL Eye Tracking in VR headset XTAL

Vedoucí práce:	Ing. Antonín Pošusta, Ph. D.
Garant za katedru:	prof. Dr. Ing. Jan Kybic
Studijní program:	Otevřená informatika
Obor:	Informatika a počítačové vědy

Praha
květen 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Trnka** Jméno: **Vladislav** Osobní číslo: **466025**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra kybernetiky**
Studijní program: **Otevřená informatika**
Studijní obor: **Informatika a počítačové vědy**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Sledování pohybu očí ve VR headsetu XTAL

Název bakalářské práce anglicky:

Eye Tracking in VR Headset XTAL

Pokyny pro vypracování:

Cílem bakalářské práce bude implementovat sledování očí a jeho zpřístupnění v zařízení XTAL, vyvinutém a vyrobeném firmou VRgineers. Vlivem unikátnosti zařízení XTAL z pohledu umístění kamer, velikosti displejů a použití specializovaných free-form čoček, není možné použít dostupná řešení.

Práce bude rozdělena na tři fáze, které na sebe postupně navazují.

První fáze bude detekování oční zornice v obrazu z kamer uvnitř headsetu. V této fázi bude vyzkoušeno naivní rozpoznání zornice pomocí jejich barevných vlastností a také natrénování rozpoznávací neuronové sítě na stejný úkol. Budou porovnány jejich výhody a nevýhody – hlavně z pohledu přesnosti a výpočetní náročnosti. Zároveň bude vytvořen obrazový dataset, který bude použit na trénování.

V druhé fázi budou získaná data použita pro určení objektu uživatelského zájmu. Bude vyzkoušeno více způsobů kalibrace, např. různé možnosti rozmístění kalibračních bodů. Z možných kalibrací poté bude vybrána jedna, která dosahuje nejlepších výsledků dle rozličných kritérií, např. přesnosti nebo časové náročnosti kalibrace.

Třetí fáze bude napsání jednoduché demo aplikace, která využije získaná data a zajímavým způsobem je zobrazí uživateli.

Seznam doporučené literatury:

- [1] Joseph Redmon, Ali Farhadi - YOLOv3: An Incremental Improvement - 2018 - arXiv
- [2] Pieter Blignaut - Mapping the Pupil-Glint Vector to Gaze Coordinates in a Simple Video-Based Eye Tracker - 2014 - Journal of Eye Movement Research, 7(1)
- [3] Arantxa Villanueva, Victoria Ponz, Laura Sesma-Sanchez, Mikel Ariz, Sonia Porta, and Rafael Cabeza - Hybrid method based on topography for robust detection of iris center and eye corners - 2013 - ACM Transactions on Multimedia Computing, Communications, and Applications, Volume 9 Issue 4
- [4] Fuhl, W., Hospach, D., Kübler, T., Rosenstiel, W., Bringmann, O., & Kasneci, E. - Ways of improving the precision of eye tracking data: Controlling the influence of dirt and dust on pupil - 2017 - Journal of Eye Movement Research, 10(3)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Antonín Pošusta, Ph.D., VRgineers, Inc., Praha

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

prof. Dr. Ing. Jan Kybic, algoritmy pro biomedicínské zobrazování FEL

Datum zadání bakalářské práce: **06.01.2020**

Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce: **30.09.2021**

Ing. Antonín Pošusta, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Tomáš Svoboda, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

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

V Praze,

.....

Poděkování

Chtěl bych poděkovat mému vedoucímu práce za věcné konzultace, cenné rady při psaní a nadšení pro celý projekt. Dále bych chtěl poděkovat mé sestře, která mi je vždy velkým vzorem v tom, jak dokáže efektivně pracovat a motivovat mě k úspěšnému dopsání celé práce.

Abstrakt

Bakalářská práce se zabývá tématem sledování očí ve virtuální realitě a jeho zpřístupněním v zařízení XTAL, vyvinutém a vyrobeném firmou VRgineers. Vlivem unikátnosti zařízení XTAL z pohledu umístění kamer, velikosti displejů a použití specializovaných free-form čoček není možné použít již dostupná řešení pro daný problém.

Práce je rozdělená do čtyř hlavních částí. První shrnuje aktuální stav VR a podrobně představuje brýle XTAL. V druhé je popsán automatický způsob na rozpoznávání lidské zornice z obrazových dat. K tomu je využito řešení za pomoci neuronových sítí. Třetí část se zabývá využitím těchto dat a jejich převedením v informaci, kam se uživatel dívá. Čtvrtá, a poslední, část představuje jednoduchou aplikaci, která ukazuje funkčnost celého řešení.

Klíčová slova

Sledování očí, konvoluční neuronové sítě, bodová kalibrace, XTAL, virtuální realita

Abstract

The main goal of this bachelor thesis is to implement eye-tracking system for virtual reality headset called XTAL. XTAL is a unique device thanks to its placement of cameras, display dimensions and usage of a special free-form lenses. Because of that no existing commercial solution can be used.

Bachelor thesis is divided into four parts. The first part is description of current VR state and in-depth look to XTAL headset. The second part is dedicated to automated solution for finding human eye pupil from image data. The third part is focused on converting pupil data to human gaze. And the last section is a demo application, that easily shows how eye tracking works.

Key words

Eye-tracking, convolution neural networks, point calibration, XTAL, virtual reality

Obsah

SEZNAM ZKRATEK A JMEN	10
SEZNAM OBRÁZKŮ	12
SEZNAM GRAFŮ	13
SEZNAM ROVNIC	14
SEZNAM TABULEK	15
SEZNAM PŘÍLOH	15
1 ÚVOD	16
2 VIRTUÁLNÍ REALITA A BRÝLE XTAL	17
2.1 Krátká historie VR.....	17
2.2 Motivace ke sledování očí.....	17
2.3 Brýle XTAL.....	18
2.3.1 Princip fungování brýlí XTAL.....	19
2.3.2 Infračervené kamery v brýlích.....	19
2.4 Použité technologie v rámci BP.....	19
3 ČÁST PRVNÍ – ROZPOZNÁVÁNÍ ZORNICE	20
3.1 Digitální reprezentace obrazu.....	20
3.2 Filtrování obrazu.....	20
3.2.1 Jas a kontrast.....	21
3.2.2 Prahování (thresholding).....	21
3.2.3 Konvoluce.....	21
3.2.4 Rozmazání (blurring).....	22
3.2.5 Detekce hran.....	22
3.3 Další důležité pojmy.....	22
3.3.1 Klasifikace, lokalizace a segmentace sémantická a instanční.....	22
3.3.2 NMS – Non maxima suppression.....	24
3.4 Metrika hodnocení přesnosti detekce.....	24
3.4.1 IoU – Intersection over union.....	24
3.4.2 Precision a recall.....	25
3.4.3 TP, TN, FP a FN v rámci počítačového vidění.....	25
3.4.4 AP – average precision.....	26
3.4.5 mAP – mean average precision.....	26
3.5 Lokalizace skleněné čočky.....	27
3.5.1 Bilineární interpolace.....	27
3.5.2 Použití interpolace k výřezu.....	28
3.6 Lokalizace zornice.....	28
3.6.1 Lidské oko.....	28
3.6.2 Hledání zornice bez CNN.....	29
3.6.3 Hledání zornice za použití CNN.....	30
3.6.4 Výsledky první části.....	34

3.6.5 Diskuze k první části	35
4 ČÁST DRUHÁ – ZJIŠTĚNÍ BODU UŽIVATELOVA ZÁJMU	37
4.1 Přehled metod určených k zjištění bodu zájmu	37
4.1.1 Metody založené na regresi	37
4.1.2 Metody založené na 3D modelu oka	37
4.2 Metody řešení u jiných náhlavních souprav	38
4.3 Výběr metody	39
4.4 Polynomiální modely	39
4.5 Chybová metrika	40
4.6 Postup měření	40
4.7 Nehomogenní lineární optimalizace	42
4.8 Výsledek měření	42
4.8.1 Výsledek měření x	43
4.8.2 Výsledek měření y	44
4.8.3 Výsledek měření dohromady	45
4.9 Výsledky druhé části	47
4.9.1 Vizualní úhel	47
4.10 Diskuze k druhé části	48
5 DEMO PROJEKT	49
6 ZÁVĚR	52
7 REFERENCE	53

Seznam zkratek a jmen

VR	Virtuální realita Virtual reality
HMD	Náhlavní souprava Head mounted display
XTAL	Brýle pro virtuální realitu vyrobené firmou VRgineers VR headset made by VRgineers company
CPU	Procesor Central processing unit
GPU	Grafický procesor Graphics processing unit
RAM	Počítačová paměť Random Access Memory
ET	Sledování očí Eye-tracking
BP	Bakalářská práce Bachelor thesis
C++	Programovací jazyk C++ programming language
SDK	Podpůrné nástroje pro vývoj Software development kit
CNN	Konvoluční neuronové síť Convolution neural network
FPS	Snímky za sekundu Frames per second
CUDA	Technologie firmy NVIDIA pro výpočty na grafické kartě Technology for computing on GPU
OpenCV	Programová knihovna na zpracování obrazu Software library for image processing
Darknet	Programová knihovna na interferenci konvolučních neuronových sítí Software library for CNN interference
IPD	Vzdálenost očí měřená od zornice k zornici Interpupillary distance
FOV	Šířka zorného pole Field of view
RGB, HSV, YUV	Barevné prostory Color space
NMS	Potlačení nemaxima Non-maximum Suppression
IoU	Průnik nad sjednocením Intersection over Union

AP	Průměrná přesnost Average precision
mAP	Průměrná přesnost přes více tříd Mean average precision
MAE	Střední absolutní chyba Mean average error
MSE	Střední kvadratická odchylka Mean squared error
RMSE	Odmocněná střední kvadratická chyba Root mean squared error
MSER	Detektor oblastí Blob detector - Maximally stable extremal regions
PASCAL VOC	Všeobecný obrazový dataset Multipurpose image dataset
MS COCO	Všeobecný obrazový dataset Multipurpose image dataset
VRGeyes	Dataset obrazů očí Eye images dataset
OpenEDS	Dataset obrazů očí Eye images dataset
GitHub	Prostor pro sdílení programového kódu Online service for code sharing
SSD, YOLO, R-CNN	Příklady konvolučních neuronových sítí Examples of CNN
FOVE, HTC, StarVR	Firmy vyrábějící VR náhlavní soupravy Companies that make VR headsets
Tobii	Firma zabývající se sledováním očí Company which is specialized od eye-tracking

Seznam obrázků

<i>Obrázek 1 – Vzhled brýlí XTAL, dostupný z https://vrgineers.com/xtal/.....</i>	<i>18</i>
<i>Obrázek 2 – Grafická vysvětlení pojmů ohledně detekce, dostupné z [37]</i>	<i>23</i>
<i>Obrázek 3 – Vizuální příklad NMS, dostupné z [12]</i>	<i>24</i>
<i>Obrázek 4 – Grafická reprezentace IoU, dostupné z [12].....</i>	<i>24</i>
<i>Obrázek 5 – Závislost přesnosti a výtěžnosti na nastavení prahu se dá vyjádřit také graficky. Dostupné z [33].....</i>	<i>26</i>
<i>Obrázek 6 – Ukázka z vlastního videa roi_video.mkv.....</i>	<i>27</i>
<i>Obrázek 7 – Ukázka označení zornice a diod v datasetu. Obrázek pochází od uživatele s nasazenými dioptrickými brýlemi. Lze tedy pozorovat přidání nových odlesků od diod.</i>	<i>33</i>
<i>Obrázek 8 – Příklad správné detekce za pomoci metody MSER.....</i>	<i>36</i>
<i>Obrázek 9 – Příklad nedetekované zornice. Lze vidět splývání zornice a duhovky do jedné plochy. Obzvláště v krajních pozicích je tento jev velmi častý. Pouze metody YOLOv3-Tiny se vstupním rozlišení 192 a 256 pixelů jsou schopny detekovat i tyto krajní případy.</i>	<i>36</i>
<i>Obrázek 10 – Zobrazení kalibračních obrazců ve variantě od 5 do 9 bodů. Lze vidět, že obrazec s body je zde použit dvakrát – jednou v rozestavení do kříže a jednou do plusu.</i>	<i>41</i>
<i>Obrázek 11 – Zobrazení kalibračních obrazců ve variantě od 8 do 11 bodů. S přibývajícím počtem kalibračních bodů by se měla zlepšovat dosažitelná odchylka.....</i>	<i>41</i>
<i>Obrázek 12 – Zobrazení kalibračních obrazců v nižším počtu než 5. S největší pravděpodobností budou mít velkou chybu, jsou zde pouze pro zajímavost.....</i>	<i>41</i>
<i>Obrázek 13 - Grafický popis vizuálního úhlu pro lepší čtenářovu představu. Dostupné z [36].....</i>	<i>47</i>
<i>Obrázek 14 – Úvodní scéna celé testovací aplikace. Ta obsahuje opici Suzanne a za ní čtvercovou mřížku. Ta slouží k přesnějšímu vyhodnocení predikované pozice. Uživatel má totiž šanci zaměřit se na výrazný bod a sledovat, jestli se na ten samý bod posune i diamant.</i>	<i>50</i>
<i>Obrázek 15 – Ukázka z kalibrace levého oka. Lze vidět červenou tečku, která určuje detekovaný střed oka. Dále lze vidět 4 modré křížky, které označují již zkalibrované body.</i>	<i>50</i>
<i>Obrázek 16 – Další ukázka z průběhu kalibrace. V daném stavu má již uživatel kalibraci provedenou, a proto je zde 7 modrých křížků. Lze vidět, že křížky tvoří podobný obrazec jako je kalibrační obrazec. To je samozřejmě správně.</i>	<i>51</i>
<i>Obrázek 17 – Ukázka kalibračního bodu zobrazeného jako diamant. V pozadí lze vidět zatemněná původní scéna.</i>	<i>51</i>
<i>Obrázek 18 – Jak již bylo řečeno, tak existuje knihovna, která řeší vše ohledně eye-trackingu. Tato knihovna se dá vzít a použít i v jiných projektech než pouze ve výše uvedeném. Zde se nachází obraz z jejího úspěšného integrování do aplikace s vesmírnou lodí. Lze zde vidět technologii foveačního vykreslování. Při té se oblast uživatelského zájmu vyrenderuje s vyšší kvalitou obrazu než zbytek scény. Ušetří se výkon a celkový dojem ze scény je lepší.</i>	<i>51</i>

Seznam grafů

- Graf 1 – Ukázka grafu tréninku sítě YOLOv3-Tiny se vstupním rozlišením 256x256px. Lze vidět, že červená křivka mAP roste do iterace č.3000 a následně už pouze minimálně. Graf tréninku dalších rozlišení vypadá velmi podobně, pouze se červená křivka zastaví níže – 86 % pro rozlišení 192x192px a pouhých 44 % pro rozlišení 96x96px..... 34*
- Graf 2 – Porovnání naměřené chyby v rámci x souřadnice. Chyba je rozdělena jak přes kalibrační obrazce, tak přes polynomy. Zápis polynomu je v následujícím tvaru. Každý člen daného polynomu je popsán dvojicí čísel, což jsou stupně polynomu u x a y souřadnice. Absolutní člen je přidán automaticky. 43*
- Graf 3 - Pokračování grafu výše, který byl rozdělený na dvě části kvůli lepší přehlednosti. Zde jsou vidět pouze kalibrační obrazce s nižším počtem bodů než je 5. Lze vidět, že výsledky jsou výrazně horší než v případě s pěti a více body. Je třeba upozornit, že tento graf má jinak nastavené měřítko v ose y. 44*
- Graf 4 – Porovnání rozptylu chyby v rámci y souřadnice. I zde se mezi nejlepší kalibrační obrazce řadí sedmibodová a pětibodová do kříže. Celková dosažitelná chyba je však mnohem horší než v případě souřadnice x. U sedmibodové lze v rámci x souřadnice dosáhnout chyby kolem 0,07, zato zde se chyba pohybuje na úrovni 0,23. Tedy více než třikrát více. 44*
- Graf 5 - I v případě y souřadnice byl graf rozdělen na dvě části kvůli lepší přehlednosti. Situace je zde velmi podobná jako v případě souřadnice x. Čtyřbodová již nedosahuje dostatečných výsledků a dvoubodová stále dokáže predikovat, i přes velice malý počet vstupních dat. 45*
- Graf 6 - Výsledné hodnoty chyby dohromady přes souřadnice x a y. Modrou čarou jsou označeny kombinace, které jsou nejlepší v rámci pětibodové kalibrace. Dosahují průměrných hodnot chyby kolem 0,028. Oranžovou barvou jsou označeny nejlepší modely v rámci sedmibodové kalibrace. Jejich průměrná chyba je nižší – kolem 0,024. 46*
- Graf 7 – Vizualizace chyby jednotlivých měřených bodů. Červené tečky označují pravdu, zelené tečky predikci a modrá čára je jejich spojnice. Delší modrá čára označuje větší vzdálenost, a tedy i chybu. 46*

Seznam rovnic

<i>Rovnice 1 – Úprava jasů a kontrastu.....</i>	<i>21</i>
<i>Rovnice 2 – Gamma korekce</i>	<i>21</i>
<i>Rovnice 3 - Prahovací rovnice.....</i>	<i>21</i>
<i>Rovnice 4 – Spojitá konvoluce dvou funkcí.....</i>	<i>21</i>
<i>Rovnice 5 - Diskrétní konvoluce dvou funkcí</i>	<i>22</i>
<i>Rovnice 6 - Příklad 3x3px Gaussovského filtru</i>	<i>22</i>
<i>Rovnice 7 - Příklad jádra detekujícího hrany v ose x</i>	<i>22</i>
<i>Rovnice 8 - Příklad jádra detekujícího hrany v ose y</i>	<i>22</i>
<i>Rovnice 9 – Výpočet hodnoty IoU.....</i>	<i>24</i>
<i>Rovnice 10 - Accuracy.....</i>	<i>25</i>
<i>Rovnice 11 – Přesnost (precision).....</i>	<i>25</i>
<i>Rovnice 12 - Výťažnost (recall)</i>	<i>25</i>
<i>Rovnice 13 - Average precision</i>	<i>26</i>
<i>Rovnice 14 - Mean average precision</i>	<i>27</i>
<i>Rovnice 15 - Maticový zápis bilineární interpolace</i>	<i>28</i>
<i>Rovnice 16 - Výpočet hodnoty za pomoci bilineární interpolace.....</i>	<i>28</i>
<i>Rovnice 17 - Zápis polynomu pomocí sumy.....</i>	<i>40</i>
<i>Rovnice 18 – Zápis polynomu se dvěma vstupními proměnnými.....</i>	<i>40</i>
<i>Rovnice 19 – Výpočet chyby MAE.....</i>	<i>40</i>
<i>Rovnice 20 – Výpočet chyby MSE</i>	<i>40</i>
<i>Rovnice 21 – Výpočet chyby RMSE.....</i>	<i>40</i>
<i>Rovnice 22 – Chyba optimalizovaná v rámci metody nejmenších čtverců</i>	<i>42</i>
<i>Rovnice 23 – Příklad rovnice pro výpočet koeficientů polynomu $f(x,y) = \alpha * x * x + \beta * x + \gamma * x * y + \delta$.....</i>	<i>42</i>
<i>Rovnice 24 – Výpočet hodnoty vizuálního úhlu</i>	<i>47</i>

Seznam tabulek

<i>Tabulka 1 – Popis jednotlivých tříd rozdělení v rámci počítačového vidění</i>	<i>26</i>
<i>Tabulka 2 – Porovnání jednotlivých vyzkoušených metod na procesoru Intel i7-7700K. Lze vidět, že pouze CNN modely dosahují dostatečných výsledků mAP. Zároveň se potvrdilo, že CPU není vhodný prostředek pro interferenci CNN. Dokonce i přes všechny CPU optimalizace, které knihovna OpenCV obsahuje.</i>	<i>35</i>
<i>Tabulka 3 – Porovnání rychlosti interference GPU na dvou implementacích – OpenCV a darknet. Zde je vidět obrovské zrychlení v porovnání s CPU verzí, kde časy kolem 2ms jsou již dostačující. Na rozdíl od CPU implementace, kde OpenCV bylo lepší než darknet, se zde role obrací. Darknet je rychlejší při menších nárocích na prostředky. Je to dáno především tím, že darknet je zaměřen pouze na sítě typu YOLO a přináší tedy mnohá vylepšení.</i>	<i>36</i>
<i>Tabulka 4 - Dosažená chyba souřadnice x</i>	<i>48</i>
<i>Tabulka 5 - Dosažená chyba souřadnice y</i>	<i>48</i>
<i>Tabulka 6 - Porovnání hodnot přesnosti s dalšími headsety</i>	<i>48</i>

Seznam příloh

Příloha 1 - Video zobrazující posun čočky v rámci obrazu	<i>roi_video.mkv</i>
Příloha 2 - Video zobrazující detekci metody založené na prahování	<i>thresholding.mp4</i>
Příloha 3 - Video zobrazující detekci metody založené na MSER	<i>mser.mp4</i>
Příloha 4 - Video zobrazující detekci metody YOLO s rozlišením 96px	<i>yolo_96.mp4</i>
Příloha 5 - Video zobrazující detekci metody YOLO s rozlišením 256px	<i>yolo_256.mp4</i>
Příloha 6 - Zdrojový kód aplikace	<i>code.zip</i>

1 Úvod

Virtuální realita (VR) je na vzestupu. Od jejího znovuobjevení v roce 2013 stále vznikají nové náhlavní soupravy (headsety), které posouvají hranice uživatelského zážitku dál, než bylo dříve vůbec myslitelné. Uživatelský zážitek se vzhledem k roku jejich vzniku neuvěřitelně zlepšil a stále se vyvíjí. VR má před sebou slibnou budoucnost.

Česko-americká firma VRgineers vyrábí brýle s nejširším zorným polem na trhu. V porovnání s ostatními výrobci se vyznačuje především tím, že k zobrazení používá dva displeje místo jednoho. Právě to, společně se speciálně vyvinutými čočkami, umožňuje dosáhnout hodnot zorného pole až na hranici lidského zraku.

Brýle pro virtuální realitu jsou složité zařízení. Je to mistrovské dílo od návrhu hardwaru po napsaný software. Obsahují mnoho částí, které jsou spolu propojené a vytváří tak onen neopakovatelný zážitek. Všechny brýle pro VR obsahují základní součásti jako sledování pohybu hlavy nebo rychlý obraz. Avšak to, co odlišuje běžné brýle od těch nejlepších, je precizní provedení daných modulů nebo dokonce i přidání dalších, které nabízejí nové možnosti.

Právě přidáním jednoho nadstandardního modulu se bude zabývat tato bakalářská práce. Jedná se o modul, který umožňuje sledovat, kam se uživatel dívá. Tato skutečnost obrovským způsobem rozšiřuje možnosti, které takový VR systém dokáže nabídnout. Ať už se jedná o nový způsob interakce s VR, rozšířené možnosti vykreslování nebo využití pro lékařský výzkum.

BP bude rozdělena na čtyři hlavní části. V první části bude popsána historie virtuální reality a její aktuální stav. Dále budou podrobněji prozkoumány brýle XTAL, s hlavním zaměřením na technické řešení pro sledování očí.

Druhá část bude pojednávat o způsobu rozpoznávání lidské zornice z obrazu. Základním předpokladem pro funkční eye-tracking je právě informace o poloze zornice. Při špatné detekci zornice nebude nikdy celý projekt fungovat. Proto bude pro rozpoznávání použito několik různých způsobů, které budou porovnány dle výsledné přesnosti i rychlosti.

Třetí část bude plynule navazovat na část druhou. Jejím úkolem bude použít naměřená data o lidské zornici a převést je do informací vyjadřujících směr lidského pohledu. V této kapitole budou prozkoumány možné způsoby, které daný problém řeší. Zároveň bude nahlédnuto na to, jak daný problém řeší jiné headsety na trhu. Zkoumané způsoby budou porovnány na základě předem definovaných kritérií, mezi která patří rychlost kalibrace nebo přesnost.

Poslední část bude věnována ukázkové aplikaci. V ní bude možnost vyzkoušet, jak eye-tracking funguje. Součástí představené aplikace bude také knihovna, která umožní jednoduché využití této technologie pro další aplikace.

Celá bakalářská práce bude řešena na obrazových příkladech, avšak finální výsledek má fungovat hlavně na videu, tedy postupném sledu obrazů. Mnoho sekcí bude proto doplněno o nahraná videa pro lepší představu.

2 Virtuální realita a brýle XTAL

2.1 Krátká historie VR

První pokusy o virtuální realitu pomocí náhlavních displejů lze datovat již do 18. století, kdy se používaly náhlavní displeje se slotem na dvojici obrázků. Každý z obrázků byl zaznamenán z mírně posunutého místa, což následně v brýlích vytvořilo prostorový obraz. [1]

Další období vývoje VR je pak shodné s vývojem počítačů. Již v roce 1961 existoval systém na přenášení obrazu z kamer do náhlavní soustavy displejů. Obraz se pak otáčel v závislosti na uživatelské hlavě. Samozřejmě vše v rámci technických možností té doby. [2]

Druhé období VR se datuje do poslední dekády minulého století, a to společně s nástupem herních konzolí. V roce 1993 společnost SEGA oznámila vydání VR brýlí pro svoji konzoli SEGA Genesis. I přesto, že výrobek nebyl nikdy dodán na trh, obsahoval již všechno, co se v dnešní době očekává od headsetu. Byl to headset s displejem, čočkami, sledováním pohybu hlavy a zvukem.

Ve stejném roce byla splněna další nutná podmínka pro funkční VR. Bez výkonu schopného plně vykreslit 3D scénu nemají headsety moc smysl. Roku 1993 se na trh dostala hra s názvem DOOM, která byla první veřejně známou 3D hrou. Celá geometrie prostoru však byla počítána pomocí CPU, který na to není určený. Výkon tou dobou ještě nebyl dostatečný.

Desetiletí uzavírá Nintendo se svým produktem jménem Virtual Boy. Ten byl uveden na trh v roce 1995 a obsahoval náhlavní soupravu a ovladač do ruky. S tehdejší cenou US\$179,95 se již dá mluvit o běžně dostupném VR. Výrobek však nebyl nikdy pozitivně přijat, a proto byl po necelém roce stažen z prodeje. Za největší problémy se považovalo použití pouze monochromatického displeje, žádné sledování pozice hlavy, a hlavně výrazné pocity nevolnosti při hraní. [3]

Celé první desetiletí 21. století se pak nedělo nic, čekalo se na prvního průkopníka. Rapidní rozvoj výkonu počítačů a pokrok na poli displejů sliboval nové možnosti. Těchto možností využila v roce 2013 firma Oculus, která uvedla na trh svůj headset s názvem Rift. Obsahoval vše důležité pro provozuschopný VR systém – tedy především sledování pozice hlavy, barevný displej – avšak s použitím moderních technologií. Displeje tou dobou měly malé rozlišení, takže uživatel viděl v brýlích pixelovou mřížku, ale přesto bylo jasné, že nové období VR právě nastalo.

V tomto období, již tedy třetím, se nacházíme v roce 2020. Na trh postupně přišly další firmy se svými headsety. Mezi nejznámější produkty patří, mimo již zmíněný Oculus Rift, také HTC Vive nebo VR kit pro herní konzoli PS4 od Sony. V dnešní době je již základní VR plně dostupná a vyladěná technologie. Důkazem toho může být nový Oculus Quest z roku 2019, který ke svému běhu nepotřebuje dokonce ani PC a vystačí si s výkonem integrovaným do brýlí.

Vývoj však neustává a stále je možné věci zlepšovat, zrychlovat a rozšiřovat jejich možnosti. Jedním z možných vylepšení je i přidání sledování pohybu očí, tedy téma, kterým se zabývá tato bakalářská práce.

2.2 Motivace ke sledování očí

Touha znát informace, kam se člověk dívá, je s námi již dlouho. Už dávno v 19. století probíhaly první pokusy o záznam pohybu přes soustavu zrcadel. Tehdy se výzkumníci snažili přijít na to, jak lidské oko funguje. Mělo to své výsledky, neboť byly objeveny např. fixace a sakády. Tedy sledy rychlých pohybů a následných zastavení na bodě zájmu. [4]

Možné způsoby využití získaných dat se postupem času rozvíjejí. Jedním z dnes nejčastějších způsobů využití je marketing. Uživatel se ukáže grafický návrh a sleduje se jeho odezva na různé části návrhu. Zkoumá se, které části ho zaujaly jako první nebo které naopak úplně přeskočil. Další z nepřeborných možností využití je zkoumání zdravotního stavu uživatele. Dnešní systémy již dokáží odhalit např. ADHD nebo Parkinsonovu nemoc. [5]

Všechny výše jmenované činnosti lze zkoumat jak v reálném prostředí, tak i v prostředí virtuálním. Tedy že se např. daný grafický návrh nemusí tisknout na papír, ale rovnou se ukáže uživateli na displeji.

V dnešní době se proto stává virtuální svět pro firmy mnohem oblíbenější. Může za to především klesající cena zařízení a celkové náklady s tím spojené. Důkazem toho je Ikea, která má téměř celý svůj produktový katalog vykreslený na počítači namísto tradičního focení. [6] Mimo nižší ceny a lepšího výsledku to přináší i bonusy ve formě 3D modelu každého výrobku do jejich aplikace návrháře domácnosti.

VR se tedy stává v daném využití pro firmy mnohem důležitější. Poslední částí do celé skládačky je pak zisk samotných dat ze sledování očí, neboť vše ostatní je již připravené. Jak kvalitní VR brýle, 3D scény určené na zkoumání, tak speciální systémy určené na vyhodnocování naměřených dat.

Pro HMD (náhlavní soupravu) pak existuje ještě jedno specifické využití – foveační vykreslování. Technologie sloužící ke snížení potřebného výkonu PC a nároků na datovou propustnost, při stejné nebo dokonce i lepší výsledné kvalitě obrazu. Člověk totiž v reálném světě zaměřuje svůj pohled pouze na kousek prostoru. Ten vymezený kousek vidí ostře a zbytek pak spíše rozmazaně. Této vlastnosti lze využít i ve VR, kdy se hlavní oblast uživatelova zájmu vykreslí s vyšší kvalitou a zbytek s mnohem nižší kvalitou. [7]

2.3 Brýle XTAL

Brýle XTAL, vyvíjené česko-americkou společností VRgineers od roku 2016, jsou náhlavní souprava pro vnímání virtuální reality. Jejich největší odlišností vzhledem ke konkurenci je výrazně ostřejší obraz. Toho je dosaženo díky OLED displejům nové generace s hustěji umístěnými pixely. Jejich další výraznou vlastností je možnost sledovat pohyb rukou bez nutnosti držet ovladače v ruce.



Obrázek 1 – Vzhled brýlí XTAL, dostupný z <https://vrgineers.com/xtal/>

Brýle jsou z konstrukčního hlediska velmi složité a obsahují spoustu součástí. Pro potřeby sledování oka stačí znát pouze vybraná část, avšak o to podrobněji. Z konstrukce vychází některé vlastnosti, které je třeba zohlednit při návrhu samotného algoritmu.

2.3.1 Princip fungování brýlí XTAL

Základní princip je jednoduchý - uživatel sleduje displej v přední části brýlí. Jelikož by obraz byl nepřírozně rovný, je třeba ještě mezi uživatele a displej vložit skleněnou deformující čočku. Ta společně se specializovaným softwarem zajistí, že uživatel vnímá obraz přirozeně.

Brýle mají tuto část zdvojenou, aby každé oko bylo nezávislé. V brýlích jsou tedy dva displeje a dvě čočky, které zároveň musí řešit přirozené spojení dvou displejů do jednoho, bez viditelného rámečku uprostřed zorného pole. Jelikož každý uživatel má jiné parametry obličeje, ať už díky pohlaví, výšce nebo rase, je potřeba, aby byla celá soustava nastavitelná. Nesprávné nastavení nutí daného uživatele šilhat nebo jinak dlouhodobě namáhat své oči. Správné nastavení záleží hlavně na tom, jak má uživatel daleko od sebe zornice (z angl. interpupillary distance, ipd).

První nastavení spočívá v horizontálním posunu celé čočky, aby člověk měl oko přesně uprostřed. Zde má totiž čočka nejlepší optické vlastnosti. Druhá možnost přizpůsobení brýlí je předozadní posun displejů. To slouží převážně k zaostření. Oba tyto posuny budou v rámci práce důležité.

2.3.2 Infračervené kamery v brýlích

Pro sledování pozice oka je do brýlí přidán speciální hardware. Jedná se o dvě infračervené kamery, kde každá snímá jedno oko. Ty jsou fyzicky umístěné mezi displejem a čočkou. Z konstrukčních důvodů je jedna zrcadlově otočená a je nutno s tím počítat. Například obraz softwarově otočit, nebo provádět celý algoritmus zvlášť pro každé oko.

Kamery mohou pracovat ve dvou módech. První je zaměřen na vyšší rozlišení obrazu při nižší snímkové frekvenci a druhý mód funguje přesně naopak. Kamera je tedy schopna dodávat snímky v rozlišení 1280x800 pixelů při frekvenci 120 Hz. Druhý mód má pak poloviční rozlišení při dvojnásobné frekvenci.

2.4 Použité technologie v rámci BP

V rámci celé práce budou použity následující technologie a knihovny. Hlavním programovacím jazykem je C++, především díky svému vysokému výkonu, který je pro VR potřeba. C++ umožňuje nativní přístup k hardwaru – zde ke kamerám. Doprovodným jazykem je Python díky jednodušší práci s ním a velkému množství knihoven.

Na zpracování obrazu bude využita open-source knihovna OpenCV, která je vyvíjena již od roku 2000. Její hlavní výhodou je vysoká míra optimalizace, početná komunita a fórum plné užitečných odpovědí. Společně s nativní podporou pro konvoluční neuronové sítě je OpenCV ideální knihovna na zpracování obrazu. Díky vystavenému rozhraní pro Python lze zkoušet části kódu v Pythonu a poté je velice rychle, již otestované, přepsat do C++. [8]

V Pythonu budou dále použity standardní nástroje pro vědeckou činnost. Jmenovitě Pandas a Numpy pro uložení dat. Matplotlib a Seaborn jako vizualizační nástroje. Další použité knihovny budou popsány dále v textu.

3 Část první – rozpoznávání zornice

První částí celého systému je rozpoznávání zornice. V této části bude vysvětleno, jak se ve vstupním obrazu najde zornice. Bude se tak dít postupným zmenšováním výřezu, kde se zornice může nacházet. Některé úpravy budou pevně dané pro všechny snímky, zatímco jiné úpravy budou mírně odlišné pro každý snímek. V této části bude také uveden obrazový dataset určený pro učení neuronových sítí.

3.1 Digitální reprezentace obrazu

Existují dva přístupy k uložení obrazu do paměti počítače. První se jmenuje vektorový a druhý rastrový. Vektorový ukládá pouze matematické informace o obsahu obrazu, které poté zpětně zobrazí. Jeho výhodou je hlavně možnost přibližování obrazu bez ztráty kvality, naopak hlavní nevýhodou jsou nároky na vyšší výpočetní výkon pro zobrazení.

Druhým používaným přístupem je rastrová grafika. Ta se dá nejjednodušeji popsat jako obdélníková mřížka, kde každé pole mřížky je vyplněno hodnotou. Jedno pole se nazývá pixel. Pro černobílý obraz stačí pouze jedna mřížka, která udává světlost obrazu. Typická hodnota nula je kompletně černá a vysoké číslo značí kompletně bílou. Jak vysoké to číslo bude, záleží na požadované přesnosti a paměťové náročnosti. Typickým příkladem může být jednobajtový rozsah 0-255, schopný rozeznat 256 stupňů šedi. Další možností může být plovoucí rozsah mezi 0 a 1, který má teoreticky neomezené rozlišení. Pro uložení jedné hodnoty jsou potřeba 4 bajty. [9]

Rozšířením celého konceptu a přidáním více mřížek lze vytvořit barevný obraz. Existuje mnoho formátů, jak mřížky kombinovat pro vytvoření barvy. Mezi nejpoužívanější zástupce patří RGB barevný prostor, dále pak HSV nebo YUV. Každý barevný prostor řeší výsledné míchání barev jiným způsobem, avšak jedno mají společné. Pro černobílý obraz stačí jedna mřížka, zato pro barevný jsou potřeba 3. Přidáním další mřížky je možné rozšířit obraz o průhledný (tzv. alfa) kanál.

Zde je třeba uvést, že knihovna OpenCV používá ve výchozím stavu formát BGR. Ten je stejný jako RGB, avšak má vyměněný první a poslední kanál. Důvodem, proč se nepoužívá standardní model RGB, je, že v začátcích vývoje knihovny, tedy na přelomu tisíciletí, byl formát BGR považován jako standard. A od té doby to tak zůstalo. [10]

Pro tuto bakalářskou práci je důležitý rastrový černobílý model s rozsahem 0 až 255. Je to z důvodu, že tato data přichází z infračervených kamer. Přidání barevných kamer by sice v mnohém ulehčilo práci s obrazem, například filtrování pomocí barvy, ale není to možné. V brýlích by muselo být přisvětlení viditelným světlem, které by ovšem negativně ovlivnilo uživatelův zážitek.

3.2 Filtrování obrazu

K získání důležitých informací z obrazu je potřeba umět obraz modifikovat. Níže jsou vypsány možné úpravy, které lze na obraz aplikovat. Rozhodně to nejsou všechny, jedná se pouze o ty nejzákladnější a zde používané. Daným operacím se říká filtr. Knihovna OpenCV má na svých stránkách velmi přehledně zpracované všechny operace, důvody jejich použití a změny, které provádějí. Jelikož se jedná o fundamentální základy zpracování obrazu, budou všechny následující podkapitoly odkazovat na [9] nebo tutoriálové stránky OpenCV [11].

Filtrování se typicky provádí přes celý obraz, případně na zvoleném výřezu. Méně častou operací je přístup k jednotlivým pixelům a nastavení jejich hodnot přímo.

Filtrování se dá rozdělit na dvě kategorie, kde první bere v potaz pouze hodnotu jednoho pixelu, zatímco druhý přístup se zaměřuje i na blízké okolí daného pixelu. Oba dva přístupy jsou validní a používají se pro jiný účel. Např. detekce hran by neměla vůbec smysl, kdyby se zaměřila pouze na jednotlivé pixely.

3.2.1 Jas a kontrast

Mezi nejzákladnější filtry se počítá úprava jasu a kontrastu. Jas je v případě černobílého obrazu vnímán jako světlost a kontrast jako rozdíl mezi tmavou a světlou na malé ploše.

Úprava jasu a kontrastu se provádí buďto lineárně, nebo nelineárně. Při lineární úpravě je výsledná hodnota každého pixelu určena lineární funkcí s parametry alfa a beta. Alfa ovládá kontrast, kde hodnota vyšší než jedna kontrast zvyšuje a naopak. Beta naopak ovládá jas, kde kladné hodnoty jas zvyšují a záporné snižují. Zde je potřeba myslet na vnitřní reprezentaci hodnot a nenechat je přetéct.

$$g(x) = \alpha * f(x) + \beta \quad (1)$$

Rovnice 1 – Úprava jasu a kontrastu

Nelineární změny v jasu se nazývají gamma korekce. Zde se výsledná pixelová hodnota vypočítá z exponenciální funkce s gammou jako parametrem.

$$g(x) = f(x)^\gamma \quad (2)$$

Rovnice 2 – Gamma korekce

3.2.2 Prahování (thresholding)

Další z technik používaných na každý pixel zvlášť je prahování. Prahování je filtr, který z původně černobílého obrazu udělá binární obraz, tedy černo-bílý. Prahování má pouze jeden parametr, a to prahovou hodnotu. Celý obraz je porovnán vůči této hodnotě, a když je vyšší, tak se nastaví hodnota 255 a 0 v případě opačném.

$$f(x) = \begin{cases} 255, & x \geq p \\ 0, & x < p \end{cases} \quad (3)$$

Rovnice 3 - Prahovací rovnice

3.2.3 Konvoluce

Dosud se všechny filtry aplikovaly pouze na zkoumaný bod a neřešily jeho blízké okolí. Pro zkoumání bodu v závislosti na blízkém okolí je třeba zavést pojem konvoluce.

Konvoluce je obecný matematický pojem z teorie zpracování signálů. Konvoluce vyjadřuje, jak jedna funkce reaguje a mění svůj tvar v závislosti na funkci jiné. Výsledkem je třetí funkce definovaná pomocí integrálu.

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha) d\alpha \quad (4)$$

Rovnice 4 – Spojitá konvoluce dvou funkcí

Pro teorii zpracování obrazu není potřeba řešit spojitý integrál, stačí diskrétní. To vychází z faktu, že obraz je v počítači zapsán taky diskrétně. Diskrétní konvoluce má stejnou vlastnost jako její spojitá verze. Tedy vyjadřuje, jak funkce f reaguje na funkci g . Funkcí f se bere celý obraz a funkcí g zkoumaný filtr. Funkce g je nazývána konvoluční jádro.

$$(f * g)(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b g(s, t) * f(x - s, y - t) \quad (5)$$

Rovnice 5 - Diskrétní konvoluce dvou funkcí

Konvoluční jádro definuje, jaký filtr se na obraz použije. Definuje totiž násobící koeficienty, které se mají aplikovat na blízké okolí pixelu. Ty mohou být kladné, záporné i nulové. Hodnota výsledného pixelu je součet hodnot přes celé jádro. Čím větší jádro, tím větší okolí je pokryto. Pro praktické využití se osvědčily velikosti jader 3x3px, 5x5px nebo 7x7px.

Všechny následující filtry budou vnitřně používat konvoluci, pouze budou měnit jádro.

3.2.4 Rozmazání (blurring)

V mnoha případech je třeba obraz mírně rozmazat. Může se to zdát jako nežádoucí krok díky ztrátě informací, avšak opak je pravdou. Mírným rozmazáním obrazu zaniknou drobné nepřesnosti obrazu, zato významné části zůstanou.

Existuje mnoho filtrů, které obraz rozmazávají. Jako nejzákladnější lze považovat průměr okolí, kdy se všechny pixelové hodnoty v blízkém okolí sečtou a následně vydělí jejich počtem. Takový filtr se jmenuje *box filtr*.

Lepších výsledků se dosahuje za použití Gaussovského filtru. Ten v rámci jádra nepřisuzuje každému pixelu stejnou váhu, ale vypočte jí z Gaussovského rozložení.

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Rovnice 6 - Příklad 3x3px Gaussovského filtru

3.2.5 Detekce hran

Detekce hran je technika, kdy se v obrazu najdou všechny hrany. Koeficienty jádra mají matematický základ v derivaci a v hledání extrémů. Jádro má takový tvar, aby dokázalo najít prudké změny v intenzitě. Čím rychlejší přechod ze světlé do tmavé části obrazu a naopak, tím výraznější hrana. Některá jádra hledají změny pouze v jedné ose. Pro kompletní detekci hran je třeba ji provést dvakrát – jednou v x ose a podruhé v y ose.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Rovnice 7 - Příklad jádra detekujícího hrany v ose x

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Rovnice 8 - Příklad jádra detekujícího hrany v ose y

3.3 Další důležité pojmy

3.3.1 Klasifikace, lokalizace a segmentace sémantická a instanční

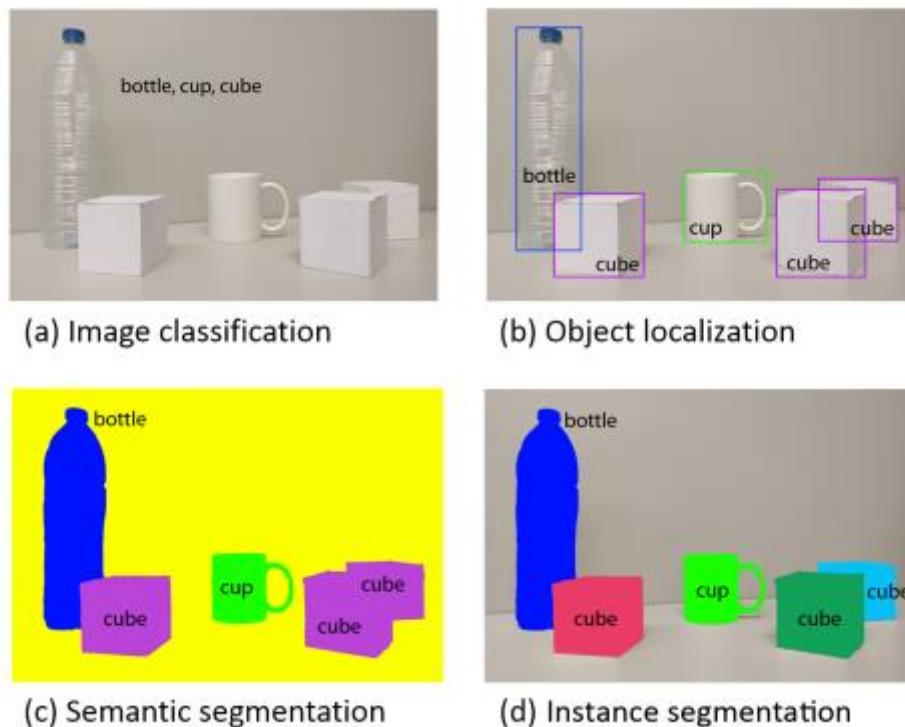
Všechny 4 pojmy (klasifikace, lokalizace a segmentace sémantická a instanční) souvisí se získáváním informací z obrazu. Každý však obsahuje jinak přesnou informaci. Klasifikace se soustředí na pojmenování toho, co se nachází na obrazu. Může to být například složitější věta

jako „Pes ležící na pohovce.“, ale i jednoduché zařazení do předem definovaných tříd – pes, pohovka.

Lokalizace je zpřesňující krok, který v daném obraze určuje, kde přesně se daný objekt nachází. Ten objekt je určen ohraničujícím obdélníkovým výřezem.

Segmentace dále zpřesňuje informaci a pozici objektu určuje v rámci pixelové přesnosti. Existuje tedy maska, která přesně pokrývá daný objekt. Segmentace se dělí na dva poddruhy, kde první – sémantický – nedokáže rozlišit mezi různými instancemi daného objektu. Tedy dva psy vedle sebe rozpozná správně jako psy, ale ve výsledné masce z nich udělá jednoho. Instanční segmentace tento problém řeší a rozpozná oba dva psy správně.

Všechny 4 pojmy jsou důležité a představují postupný vývoj v oblasti rozpoznání obrazu. Nejdříve byly vyřešeny jednodušší problémy a postupně se přidávaly další. Každý jednodušší krok je nutným předpokladem ke složitějšímu kroku. Protože např. přesná segmentace objektu je zbytečná, pokud klasifikátor objekt místo psa označil za kočku.



Obrázek 2 – Grafická vysvětlení pojmů ohledně detekce, dostupné z [37]

3.3.2 NMS – Non maxima suppression

Non-maxima suppression se dá do češtiny přeložit jako potlačení nemaxima. Při práci s detektory se dost často stane, že jeden objekt v obraze bude detekován vícekrát. Každá z těchto detekcí bude mít trochu jinou pravděpodobnostní hodnotu, jejich ohraničující obdélník bude posunutý o pár pixelů, ale bude se jednat vždy o ten samý objekt. NMS je způsob, jak z predikcí vybrat pouze ty důležité, a naopak opakující se zahodit. [9]



Obrázek 3 – Vizuální příklad NMS, dostupné z [12]

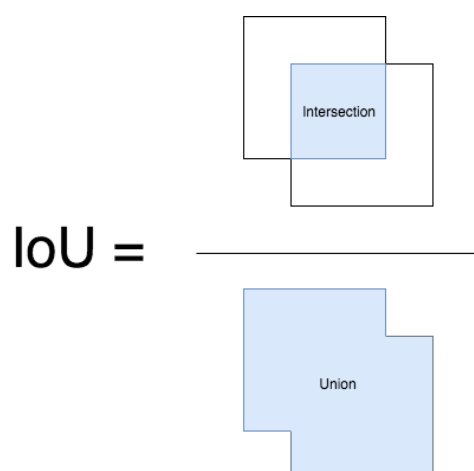
3.4 Metrika hodnocení přesnosti detekce

3.4.1 IoU – Intersection over union

Průnik nad sjednocením je způsob, jak zjistit přesnost lokalizace. Ten porovnává přesnost na základě porovnání detekce a pravdy. Může se jednat o detekci ve formě obdélníku, polygonu nebo v pixelové segmentaci, jelikož princip je stejný. Vypočítá se poměr plochy průniku a sjednocení těchto dvou obrazců. Čím více se výsledný poměr blíží číslu 1, tím více je detekce přesnější.

$$IoU(P, T) = \frac{plocha(P \cap T)}{plocha(P \cup T)} \quad (9)$$

Rovnice 9 – Výpočet hodnoty IoU



Obrázek 4 – Grafická reprezentace IoU, dostupné z [12]

3.4.2 Precision a recall

Precision a recall jsou dva pojmy z teorie strojového učení a pravděpodobnosti (česky přesnost a výtěžnost). [13]

Uvažujme binární detektor, který zařazuje objekty do dvou tříd – splňující a nesplňující zvolený požadavek. Vzhledem k predikci existují 4 možné výsledky zařazení dle pravdivé a predikované třídy. Pokud je objekt správně zařazen, tak se jedná o true positive – TP – nebo true negative – TN. Naopak při špatné klasifikaci je objekt buďto false positive – FP – nebo false negative – FN.

Za pomoci těchto 4 zařazení lze vypočítat 3 různé metriky o kvalitě prediktoru. Přesnost určuje, s jakou pravděpodobností je pozitivně klasifikovaný prvek doopravdy pozitivní. Výtěžnost určuje pravděpodobnost, s jakou dokáže klasifikátor najít pozitivní prvek. Poslední metrikou je accuracy, která říká, jak celkově je klasifikátor přesný. Ta může být občas nepřesná, pokud dataset obsahuje výrazně více objektů z jedné třídy než z druhé. Poté by klasifikátor mohl jednoduše určit všechno do početnější třídy a mít vysokou accuracy, i když vlastně nic neumí. Projeví se to například na výtěžnosti, která bude nula.

Pro různé úlohy je každá metrika jinak důležitá. Například při sestavování spamového filtru na e-mail je důležitá výtěžnost. Pro uživatele je lepší, aby se k němu dostaly všechny důležité emaily a občas i nějaký spam, než aby nechodily žádné spamy, ale zároveň ani důležité emaily.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

Rovnice 10 - Accuracy

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

Rovnice 11 – Přesnost (precision)

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

Rovnice 12 - Výtěžnost (recall)

3.4.3 TP, TN, FP a FN v rámci počítačového vidění

Jak již bylo řečeno, tak rozdělení je binární a pracuje tedy pouze na dvou vstupních třídách – pozitivní a negativní. Avšak ve světě rozpoznávání neexistuje nic jako pravdivě pozitivní a negativní třída. Je tedy třeba všechny detekce nějak rozdělit. K tomu se používá kombinace IoU s nastaveným prahem a zároveň správné zařazení do třídy. Obě dvě podmínky musí být splněny, avšak prahování má vyšší přednost v případě zařazení do třídy. To je vidět hlavně u negativní skupiny. Vše vysvětlí Tabulka č. 1.

Zde je třeba říci, že kategorie true negative v rámci rozpoznávání nedává smysl. Jedná se totiž o všechny nenaleznuté lokalizace prázdných objektů. A takových je v jednom obraze nekonečně mnoho, jelikož lze vytvořit nekonečně mnoho různých obdélníků, které nic neobsahují – tedy obsahují pouze pozadí.

		Pravda	
		Positive	Negative
Predikce	Positive	TP správně zařazená třída a IoU vyšší než práh	FP špatně zařazená třída a IoU vyšší než práh
	Negative	FN IoU menší než práh a objekt zde má být	TN IoU menší než práh a objekt zde nemá být

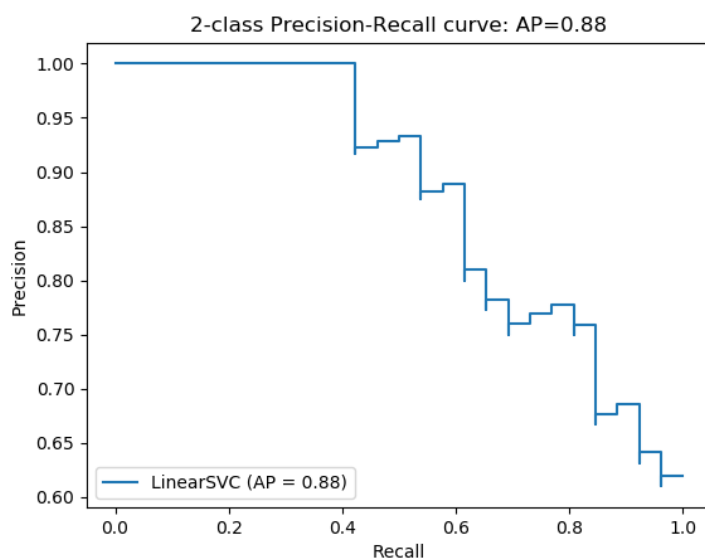
Tabulka 1 – Popis jednotlivých tříd rozdělení v rámci počítačového vidění

3.4.4 AP – average precision

Hodnota přesnosti a výtěžnosti se mění s nastaveným prahem. Pro určení použitelnosti klasifikátoru přes různé hodnoty prahů se používá metrika vážené přesnosti. Ta vytvoří průměr hodnot přes určený počet prahů a vyjádří přesnost klasifikátoru jako jedno číslo. Nárůst mezi jednotlivými výtěžnostmi se poté použije jako váha.

$$AP = \sum_n (R_n - R_{n-1})P_n \quad (13)$$

Rovnice 13 - Average precision



Obrázek 5 – Závislost přesnosti a výtěžnosti na nastavení prahu se dá vyjádřit také graficky. Dostupné z [33]

3.4.5 mAP – mean average precision

Poslední podkapitola, která se bude zabývat vysvětlením pojmů ohledně přesnosti, se nazývá mAP. Žádná z předchozích metrik nebyla dostatečná pro kvalitní popsání detektoru. Jelikož kvalitní detektor dělá mnoho věcí, tak je všechny potřeba zohlednit. Jedná se o kvalitu klasifikace

společně s detekcí přes všechny třídy. Výše uvedené kapitoly postupně přidávaly do metriky některé vlastnosti, ale teprve až mAP je všechny integruje do sebe. Výsledná definice mAP je tedy průměr AP přes všechny třídy. Hodnoty mAP nabývají mezi 0 a 1, kde 1 je nejlepší možný výsledek.

Jelikož tato BP pracuje v detekci pouze s jednou třídou, tak je možné mAP a AP mezi sebou zaměňovat.

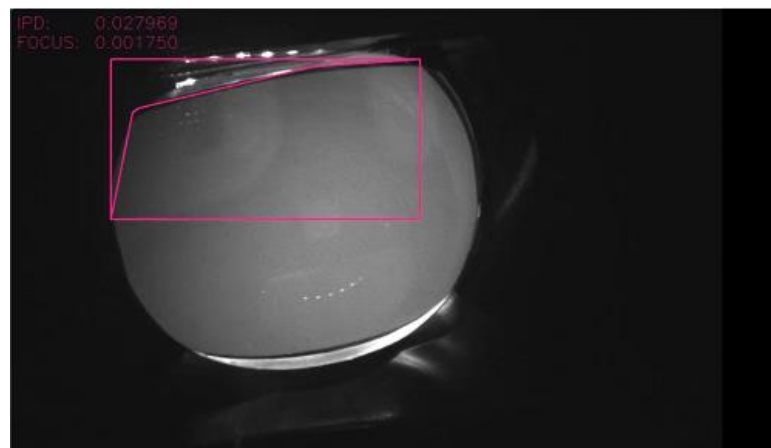
$$mAP = \frac{1}{|\text{classes}|} \sum_{c \in \text{classes}} \text{precision}(c) \quad (14)$$

Rovnice 14 - Mean average precision

3.5 Lokalizace skleněné čočky

Existují dvě nastavení brýlí, která ovlivňují pozici čočky v obraze. Ty byly již popsány v části o brýlích XTAL. Ve zkratce se jedná o horizontální posun čočky a předozadní posun displejů. Je zajištěno, že čočka bude vždy vidět, i přes všechny možné kombinace posunu. Výše uvedené posuny jsou zaznamenány ve video příloze č. 1 s názvem *roi_video.mkv*. Zároveň je zde i vidět výsledné řešení pomocí zde uvedené metody.

Pro lokalizaci čočky je možno využít faktu, že oba dva posuny lze z brýlí vyčíst. Tyto údaje jsou v lineární škále, proto je možno využít bilineární interpolaci k výpočtu pozici čočky.



Obrázek 6 – Ukázka z vlastního videa *roi_video.mkv*

3.5.1 Bilineární interpolace

Lineární interpolace je způsob, jak dopočítat neznáme hodnoty v zadaném rozsahu dvou hodnot – krajních bodů. Výsledná hodnota je vážený průměr podle vzdálenosti od krajních bodů. Všechny možné hodnoty lze najít na úsečce spojující krajní body.

Bilineární interpolace je rozšíření lineární interpolace do 2D. Hodnoty se nehledají pouze na úsečce, nýbrž na celé ploše. Tu určují 4 krajní body – $P_{11} = (x_1, y_1)$, $P_{12} = (x_1, y_2)$, $P_{21} = (x_2, y_1)$ a $P_{22} = (x_2, y_2)$.

Ze všech bodů a funkčních hodnot se sestaví maticová rovnice o 4 neznámých. Ty se použijí k samotné interpolaci pouhým dosazením do vzorce. (15)

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1 * y_1 \\ 1 & x_1 & y_2 & x_1 * y_2 \\ 1 & x_2 & y_1 & x_2 * y_1 \\ 1 & x_2 & y_2 & x_2 * y_2 \end{bmatrix} * \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} f(P_{11}) \\ f(P_{12}) \\ f(P_{21}) \\ f(P_{22}) \end{bmatrix} \quad (15)$$

Rovnice 15 - Maticový zápis bilineární interpolace

$$f(x, y) \approx \alpha_1 + \alpha_2 * x + \alpha_3 * y + \alpha_4 * x * y \quad (16)$$

Rovnice 16 - Výpočet hodnoty za pomoci bilineární interpolace

3.5.2 Použití interpolace k výřezu

V rámci brýlí XTAL jsou 4 krajní body dány následujícími kombinacemi – displej přiblížený a čočka vlevo, displej přiblížený a čočka vpravo, displej oddálený a čočka vlevo, displej oddálený a čočka vpravo.

Je potřeba zachytit tvar celé čočky v rámci obrazu. Tedy získat pozice bodů, které čočku ohraničují. Čím více bodů, tím lepší segmentace, ale vyšší paměťové nároky. Hlavně je důležité odstranit všechny možné odlesky, které by v další části mohly způsobovat nepřesnosti.

Pro segmentační popis čočky vychází nejlépe 8 bodů. Pro každý bod je vytvořen vlastní bilineární interpolátor, jelikož každý bod se posouvá jinak. Je to proto, že střed kamery je posunutý od středu obrazu.

Již zmíněných 8 bodů bylo ručně označeno v rámci 4 krajních kombinací. Za pomoci těchto naměřených dat je již možné interpolovat. Výsledek interpolace a přidání segmentační masky lze vidět také ve video příloze č. 1.

Po úspěšné lokalizaci čočky se vstupní obraz zmenšil. Jak moc, záleží hlavně na velikosti hodnoty přiblížení. Změna nastavení ipd velikost výřezu téměř nezmění. Při největším přiblížení se obraz z původních 1280x800px zmenšil na 660x350px. Při nejmenším přiblížení má finální výřez rozměry 557x290px.

3.6 Lokalizace zornice

Po úspěšné lokalizaci čočky je možné začít hledat zornici. Pro segmentování byly vyzkoušeny dvě metody, kde jedna používá konvoluční neuronové sítě (convolution neural networks, CNN) a druhá ne. Výsledek obou metod bude srovnán v rámci přesnosti segmentace a dalších kritérií.

3.6.1 Lidské oko

Lidské oko je párový orgán, který umožňuje lidem vnímat svět. Nachází se v horní části obličeje blízko mozku, aby bylo fyzické spojení mezi mozkiem a okem co nejkratší, a tedy vizuální informace předána co nejrychleji. Velikost celého oka dospělého člověka je průměrně 24 mm a nezáleží na rase ani pohlaví. Oko dětí dosáhne plné velikosti již ve dvanácti letech. [14]

Lidské oko se skládá z mnoha částí. Důležité pro tuto práci jsou části, které jsou vidět. Jedná se o zornici, duhovku a bělmo. Uprostřed oka se nachází zornice. Jejím úkolem je regulace množství dopadajícího světla na sítnici uvnitř oka. Regulace probíhá pomocí změny velikosti zornice, kdy malá zornice propustí méně světla než velká. Zornice má černou barvu a kruhový tvar.

Další viditelnou částí oka je duhovka. Je to oblast kruhového tvaru, která obepíná zornici. Duhovka má u lidí různé barvy, ty však vlivem infračerveného obrazu nelze vidět. Třetí viditelnou částí oka je bělmo. To je bílé a vyplňuje zbytek prostoru oka.

Poslední částí, která se týká oka, je oční víčko. To je výběžek kůže, který je schopen oko zakrýt – mrknout. Při mrknutí dochází k čistění oka od nečistot a jeho zvlhčení. Mrkání probíhá reflexivně, ale člověk je schopen mrknout i svévolně. Průměrně člověk mrkne 5x až 15x za minutu, avšak při soustředěné práci (například na počítači nebo právě v brýlích pro VR) se frekvence může zmenšit. Délka jednoho mrknutí je méně než půl sekundy. [15]

Při vývoji algoritmu je důležité na mrknutí nezapomenout, jelikož ve chvíli mrknutí není vidět zornice, dochází tedy k okluzi sledovaného předmětu. Je ovšem možné na něj nahlížet. i z pozitivního jiného hlediska. Mrknutí může sloužit jako způsob interakce s VR.

Oko pracuje ve dvou režimech, které se střídají. Jedná se o sakády a fixace. Sakáda je rychlý pohyb oka do chtěné pozice a fixace je přesný opak, tedy když se oko zaměří na nějaký objekt a nehýbe se. V rámci sakád dokáže oko vyvinout rychlost až $900^\circ/\text{s}$. Převáděno na čas uvnitř kamer, tak oko dokáže přesunout z levé části displeje do pravé za méně než 30ms. Ten stejný čas platí pro přejezd z horní části displeje do dolní.

3.6.2 Hledání zornice bez CNN

První metoda k detekci zornice nepoužívá CNN, ale pouze se snaží za pomoci běžných operací s obrazem získat polohu zornice. Zornice je v obraze vidět jako černá tečka. Celé řešení tedy bude spočívat v detekci černé plochy.

3.6.2.1 Adaptivní prahování

První řešení používá ke své práci adaptivní prahování. Využívá faktu, že zornice je temná plocha v obraze a snaží se na základě této informace zornici najít. Prvně je obraz vyfiltrován mediánovým filtrem. To zajistí odstranění šumu z obrazu. Zároveň mediánový filtr dokáže zvýraznit zornici.

V ideálním případě by šlo nastavit jeden práh, který by zornici vždy našel. Jelikož se však hodnota černé liší s každým obrazem, tak to není možné. Dokonce i v rámci jednoho obrazu má zornice trochu jiné hodnoty. Pro správné nastavení prahu byl tedy nastaven algoritmus, který se snaží ideální prahovou hodnotu vyčíst z obrazu.

Pro správné nastavení prahu je vytvořen histogram všech hodnot v obraze. Zde se typicky objeví jedna hodnota, která svým počtem převažuje nad ostatními. K této hodnotě je poté přičten malý epsilon pro lepší detekci. Je to z důvodu výše uvedeného, tedy že hodnoty uvnitř zornice se mírně liší.

Po provedení tohoto prahování vznikne černo-bílý obraz, kde bílá obsahuje místa s možnou zornicí a černá naopak pozadí. Pro lepší filtrování je přidána ještě morfologická operace otevření. Ta v daném obraze dokáže rozpojit dvě kontury, které se mírně dotýkají. Je to hlavně z důvodu, že na obraze je více ploch černých a takto nalezená zornice má tendence s nimi splýnout.

Posledním krokem je najít všech vzniklých kontur v obraze a jejich filtrování. Typicky v obraze vznikne výsledných kontur víc, proto je třeba najít tu, která je s největší pravděpodobností zornice. Tento finální filtr využívá vlastnosti, že zornice je kruhová. Hledá konturu s nejlepším poměrem dvou ploch – plochy opsané kružnice a plochy kontury. Čím blíže je tento poměr hodnotě jedna, tím větší pravděpodobnost, že kontura je kruhová.

3.6.2.2 MSER

Hledání zornice se dá obecněji popsat jako hledání souvislé plochy, čemuž se v angličtině říká *blob detection*. V této kategorii existuje mnoho způsobů, kde většina z nich byla vyzkoušena. Pouze jediná metoda přinesla výsledky, které se již dají dále použít. Zbylé metody nedokázaly

detekovat zornici vůbec, případně jenom v ideálních podmínkách – tedy když člověk kouká přímo před sebe a hranice mezi zornicí a bělmem jsou velmi výrazné.

Použitelná metoda se jmenuje MSER (Maximally stable extremal regions) a dá se zjednodušeně popsat následovně. Metoda funguje na principu postupného prahování obrazu. Při každém prahu se uloží, které pixely jsou pod prahem a které nad. Za extrémní region je prohlášena oblast, která vydrží víceméně neměnná přes více prahů. Po provedení detekce bylo provedeno stejné filtrování jako v případě kapitoly výše. Tedy z mnoha nalezených oblastí byla vybrána ta jedna, která je s největší pravděpodobností zornice. [16]

3.6.3 Hledání zornice za použití CNN

Druhou metodou k hledání zornice bude využití konvolučních neuronových sítí (CNN). Princip neuronových sítí je znám již z minulého století, kdy probíhaly první pokusy. Kvůli nedostatečnému výkonu tehdejších počítačů byly sítě velmi omezené. Klasická plně propojená neuronová síť si také uměla pouze těžko poradit s obrazem na vstupu. [17]

Toto omezení bylo překonáno za pomoci nových vrstev sítě, které dokáží ze vstupního obrazu vybrat pouze důležitá data a zbytek zahodit. Ta nejdůležitější vrstva využívá ke své práci již zmíněnou konvoluci. Podle ní také vzniklo celé pojmenování – konvoluční neuronové sítě. Konvoluční vrstvy umožňují aplikovat na obraz různé filtry – např. detekci hran – které lépe zachycují důležitá data v obrazu.

Od svého znovuobjevení, kolem roku 2010, získávají CNN na veliké popularitě. Téměř všechny operace s obrazem, které byly dříve řešeny konvenčními metodami, jsou dnes nahrazovány za použití CNN. Práce s nimi je mnohem pohodlnější a dosahuje se lepších výsledků. Dříve nemožné nebo velmi těžké úlohy lze nyní řešit relativně snadno. Hlavně v oblasti klasifikace obrazu, detekce a segmentace CNN vynikají.

Práce s CNN však není úplně jednoduchá a má své omezení. Největším problémem, stejně jako všech metod strojového učení, je potřeba datasetu. Tedy označených obrazů se vzorovými daty. Ten je potřeba buďto najít již hotový, nebo takový vytvořit. To znamená sehnat obrazová data a do nich ručně doplnit hledaná data. Další nevýhodou je relativní neznámost celého učení. CNN fungují jako černá skříňka a v případě neúspěchu je těžké zjistit, co je špatně. Poslední nevýhodou je nutnost učení, které v případě složitějších problémů může trvat i dny na výkonných grafických kartách.

3.6.3.1 Porovnání dostupných CNN

V současné době (březen 2020) je na světě mnoho různých CNN. Liší se uspořádáním vrstev, jejich vzájemným propojením, použitím jiné aktivační funkce a mnohým dalším. Všechny CNN se dají zařadit do dvou kategorií, které definují jejich vlastnosti. Jsou buďto jednofázové nebo dvoufázové – *one-stage*, *two-stage*. Dvoufázové se vyznačují vyšší přesností za cenu nižší rychlosti a jednofázové naopak.

Dvoufázové fungují tak, že prvně najdou oblasti, kde je pravděpodobné, že se nachází objekt. Na dané oblasti poté aplikují CNN a získají výsledek. Neznámějším členem dvoufázových CNN je síť s názvem R-CNN. Je to jedna z prvních, která umožňovala detekci pozice objektů v obraze. Z toho její celý název Region-CNN. Pochází z roku 2013 a postupně dostala mnohá vylepšení zlepšující, jak přesnost, tak i rychlost. Nové verze nesly příznačně jméno Fast R-CNN a následně Faster R-CNN. Poslední příspěvek do rodiny se nazývá Mask R-CNN a pochází z roku 2017. Jak již název napovídá, tak síť umí i segmentaci obrazu. [18]

Jednofázové sítě fungují jinak. Aplikují CNN rovnou na celý obraz a z výstupu určí důležité informace. Hlavní výhoda je jasná, průchod CNN, tedy to náročné na výpočetní výkon, se dělá pouze jednou. Mezi nevýhody pak patří nižší přesnost nebo i nedetekování objektů. Hlavně při velikém počtu objektů v obraze je tato nevýhoda patrná. Mezi zástupce této kategorie patří dvě sítě – SSD a YOLO. Obě dvě dávají svoji příslušnost velmi jasně najevo již v názvu, neboť SSD znamená Single Shot Detector a YOLO poté You Only Look Once. První verze sítě YOLO vyšla v roce 2015 a byla postupně vylepšována až do roku 2018, kdy vyšla YOLOv3. V roce 2020 vyšla nová generace s názvem YOLOv4, která již nijak nesouvisí s původními autory. Síť YOLO existuje i v jedné další variantě pojmenované Tiny-YOLO. Již z názvu je patrné, že síť je malá. Dosahuje tedy horších detekčních vlastností při ještě vyšší rychlosti. [19] [20]

Nové modely sítí vznikají téměř každým dnem a posouvají hranice, jak v přesnosti, tak v rychlosti. Na tomto lze demonstrovat, že použití CNN v rámci rozpoznávání obrazu je nyní obrovským tématem.

3.6.3.2 MS COCO Dataset

Společně s rozvojem CNN se rozvíjí i jejich nedílná součást – trénovací datasety. Jedním z prvních veřejně dostupných datasetů je PASCAL VOC, který sloužil jako podklad pro veřejnou soutěž. V té době bylo standardem uvádět dosaženou lokalizační přesnost sítě právě na tomto datasetu.

V roce 2005 vyšla první verze, která obsahovala 1578 obrazů a na nich označené 4 třídy – jízdní kola, auta, motorky a lidi. Postupem času se dataset rozšiřoval a zlepšoval. V roce 2007 se zvětšil počet tříd na 20 a na tomto číslu již zůstal. V roce 2009 přibyla druhá kategorie zabývající se segmentací obrazu. Poslední verze vyšla roku 2012, která obsahovala přes 11000 obrazů a na nich 27450 lokalizovaných objektů a skoro 7000 segmentací. [21]

Po ukončení PASCAL VOC soutěže se v roce 2015 objevil nový veřejně dostupný dataset, který převzal štafetu a se stejným cílem pokračuje dál. Jedná se o MS COCO dataset – common objects in context. Jednalo se o 200000 obrazů, na kterých byla vyznačena segmentace do 80 tříd. Obrovský nárůst v rozsahu vzhledem k PASCALu. [22]

Podobně jako PASCAL VOC, tak i MS COCO organizuje rozpoznávací soutěže. První proběhla v roce 2015 a poté každý další rok. Narozdíl od PASCALu se dataset již nevylepší, vše se trénuje, zkouší a vyhodnocuje na stejném datasetu. Pouze se mění pravidla soutěže, kdy v roce 2018 byla odstraněna podpora pro lokalizaci a soutěží se jenom v segmentaci.

Existuje celkem 12 metrik, které porovnávají přesnost detekce uvnitř MS COCO datasetu. Každá se zaměřuje na něco trochu jiného, např. pouze na malé objekty nebo naopak pouze na velké. Hlavní používanou metrikou je mAP přes více prahovacích hodnot – konkrétně 10 hodnot v rozmezí 0,5 až 0,95 po 0,05 krocích. Je to velká změna oproti metrice PASCALu, kde byl nastaven pouze jeden práh IoU, a to 0,5. Hlavní rozdíl těchto dvou metrik je, že pro dobrý výsledek v soutěži PASCAL stačí detekovat objekty docela nepřesně, zatímco v MS COCO je potřeba detekovat mnohem přesněji.

Pod záštitou MS COCO existují i další soutěže, například rozpoznávání lidské pózy nebo pojmenování obrazu. Tyto další soutěže mají svoje datasety a svoje metriky.

3.6.3.3 Výběr CNN

Není možné natrénovat všechny dostupné sítě a následně z nich vybrat nejlepšího kandidáta. Je nutné vybrat jednu, případně dvě, dle daných požadavků na síť. Požadavky této práce jsou následující, vysoká rychlost při nízkém vytížení GPU a CPU. Sledování očí v rámci brýlí je pouze

doplňek, nemůže si tedy pro sebe vzít všechny výpočetní výkon. Ten musí jít hlavně do zpracování 3D scény a vykreslování. Dalším požadavkem je možnost interference v C++. V současnosti je hlavním jazykem pro strojové učení Python, a tedy většina sítí má implementaci v něm.

V ideálním případě bude tedy síť dostatečně rychlá, aby zvládala zpracovat dva proudy dat (levé a pravé oko), kde každý proud má 120 snímků za sekundu (FPS). Toto vše při celkovém vyžití GPU pod 10 %. Zní to jako velmi náročný úkol, avšak některé vlastnosti úkolu celou práci zlehčují. Například hledání pouze jedné detekční třídy – oko – zmenšuje náročnost celého úkolu. Také znalost, že na obraze nikdy nebude více než jedno oko, zlehčuje celou práci. Není potřeba vůbec provádět NMS, stačí vzít pouze region s nejvyšší pravděpodobností.

Pro další postup byla nakonec zvolena síť YOLOv3-Tiny. Plnohodnotné modely byly vyřazeny kvůli nedostatečné rychlosti. Faster R-CNN se pohybuje v řádu jednotek FPS a YOLOv3 v řádu nižších desítek FPS. To vše při plném vyžití GPU. Odlehčený model dosahuje již dostatečných hodnot pro toto zadání. Na grafice vyšší střední třídy, NVIDIA RTX 2070, je schopen běžet kolem 370 FPS. S vyšší rychlostí ovšem klesá dosažených výsledků mAP na MS COCO datasetu, kde YOLOv3-Tiny dosahuje pouhých 33 %. [23] [19] [24]

Dalšími důvody pro zvolení zrovna této sítě je i její výborná podpora. Jedná se o jednu z neznámějších sítí, a proto se dá najít spousta návodů, jak síť natrénovat. Zároveň je celý framework napsán v C++, což umožňuje snadné použití v produkčním kódu.¹

3.6.3.4 Obrazový dataset

Jak již bylo řečeno, tak metody strojového učení potřebují ke své práci dataset, na kterém se učí. V rámci této práce vznikl potřebný dataset s názvem VRGeyes. Ten ve své aktuální verzi obsahuje přes 7000 obrazů a jejich segmentací. Segmentovány byly dvě třídy – zornice a diody. Zornice je samozřejmě to hlavní, co se v obraze hledá, ale diody mohou být také užitečné. Ač v rámci této práce nejsou využity, je dobré mít data připravena do budoucnosti.

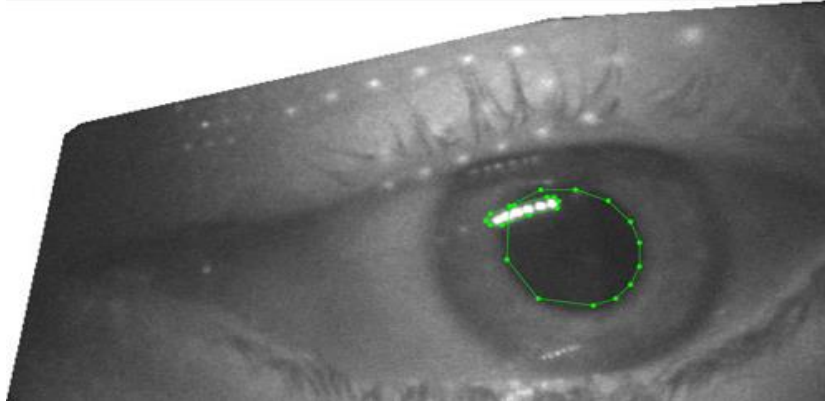
Naměřená data pochází celkem od 4 lidí, kdy každý dodal do datasetu asi 300 obrazů levého oka a 300 obrazů pravého oka. Dva z participantů měli nasazené dioptrické brýle, neboť ty způsobují změnu kvality obrazu, hlavně zhoršují ostrost. Další nevýhodou brýlí je přidání, a dokonce i znásobení odlesků.

Vzhledem k tomu, že výsledné obrazy vypadají téměř stejně, je pro tyto účely vybraný vzorek 4 lidí dostačující. Do budoucna by bylo účelné do pozorování zařadit další účastníky a rozšířit tak získání dat.

Dataset obsahuje ještě jednu kategorii obrazů, která obsahuje pouze záběry z prázdného HMD. Jedná se tedy o data, kde není zornice. Taková data jsou důležitá, aby se síť naučila i situace, kdy zornice přítomná není.

¹ Práce na této BP začaly již před více než rokem (březen 2019), kdy bylo YOLO nejmodernější řešení. I dnes je to moderní řešení, avšak situace už je přece jenom trochu jiná. Objevily se nové sítě, které dosahují lepších výsledků. Dokonce existují i segmentační sítě pracující v reálném čase, jako Yolact nebo CenterMask. Zároveň přibyla podpora CUDA akcelerace interference do OpenCV. To vše by ovlivnilo výběr sítě, když by měl proběhnout v době psaní textu.

Pro zlepšení detekce a snížení přeučtenosti sítě bude k tréninku použit ještě jeden dataset. Jedná se o veřejný dataset s názvem OpenEDS od firmy Facebook. Ten obsahuje přes 12000 segmentovaných obrazů zornice, které byly pořízeny pomocí infračervené kamery uvnitř HMD. Tedy přesně to, čím se zabývá tato BP. Avšak jejich kamera je mnohem detailnější a obrazy přesnější. Nelze proto použít pro učení pouze tento dataset.



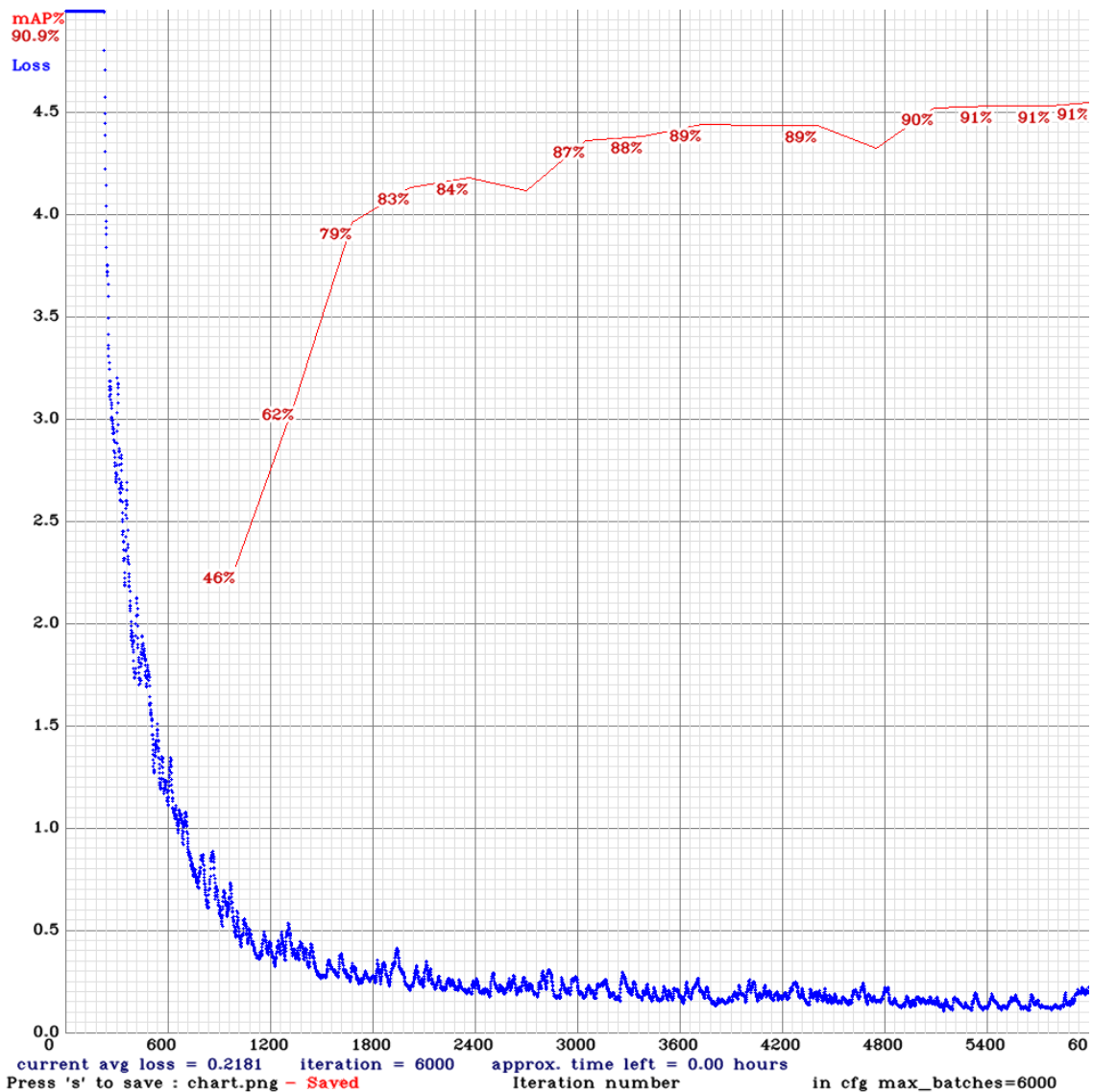
Obrázek 7 – Ukázka označení zornice a diod v datasetu. Obrázek pochází od uživatele s nasazenými dioptrickými brýlemi. Lze tedy pozorovat přidání nových odlesků od diod.

3.6.3.5 Učení CNN

Referenční implementace od autora sítě YOLO se nachází na GitHubu. Existuje mnohem lepší implementace, která původně vznikla jako odnož té referenční. Ta opravuje mnohé chyby, přidává nové vlastnosti a tím zlepšuje detekci a mnohé další. Bude tedy použita tato verze. [24]

Po úspěšné kompilaci se přikročilo k samotnému trénování. Sít YOLO umožňuje měnit vstupní velikost obrazu. Čím větší vstup, tím lepší (a pomalejší) detekce. Byly natrénovány celkem 3 sítě, kde každá měla jinou vstupní velikost. Konfigurační soubory byly upraveny dle návodu na stránkách sítě. Jedná se především o nastavení počtu tříd, které detektor má hledat. Od toho se odvíjí počet filtrů, které se mají aplikovat v konvolučních vrstvách. Zároveň od počtu tříd se odvíjí i minimální počet iterací, po které se má síť trénovat. Bylo zvoleno celkem 6000 iterací pro každou síť. Trénink trval na jedné GPU NVIDIA 1080 Ti v rámci minut. Sít se vstupním rozlišením 96x96px byla natrénovaná asi za 10 minut. Sít s nejvyšším rozlišením, tedy 256x256px, do hodiny. Tímto bych chtěl poděkovat katedře kybernetiky za poskytnutí GPU serverů.

Z Grafu č. 1 lze vidět, že 6000 iterací je dostatečných. Červená křivka mAP roste relativně strmě zhruba do iterace č. 3000 a od té doby už stoupá pouze minimálně, případně už je rovnoběžná. Pokud by byl potřeba lepší výsledek, tak delší trénink ho nezajistí.



Graf 1 – Ukázka grafu tréninku sítě YOLOv3-Tiny se vstupním rozlišením 256x256px. Lze vidět, že červená křivka mAP roste do iterace č.3000 a následně už pouze minimálně. Graf tréninku dalších rozlišení vypadá velmi podobně, pouze se červená křivka zastaví níže – 86 % pro rozlišení 192x192px a pouhých 44 % pro rozlišení 96x96px.

3.6.4 Výsledky první části

Jak již bylo řečeno, tak byly použity celkem tři modely ke zjištění zornice. Jeden byl velice naivní a používal pouze adaptivní thresholding. Druhý model využíval ke své práci jako detektor oblastí MSER. Třetí model tvořila CNN síť s názvem YOLOv3-Tiny. Všechny výsledky jsou zaneseny v Tabulce č. 2.

Pro vypočtení AP byla použita testovací množina 1000 obrazů z datasetu VRGeyes. Jedná se o rozdělení datasetu v poměru 85:15 jako trénovací a testovací data. K vypočtení AP byl použit python kód, který lze najít v referenci č. 25.

Jak již bylo řečeno, tak při rozpoznávání očí je důležitá nejenom přesnost, ale i rychlost detekce. Měření rychlosti detektoru probíhalo na stroji s procesorem Intel i7-7700K a grafické kartě NVIDIA 1080 Ti. V systému bylo nainstalováno 16 GB RAM. Pro interferenci CNN byly vyzkoušeny dvě implementace – darknet a OpenCV.

3.6.5 Diskuze k první části

Z Tabulky č. 2 lze vidět, že obecná pravidla pro práci s obrazem a detekci stále platí. Lepší a přesnější detekční metody potřebují mnohem více výpočetního výkonu. Další pravidlo, že interference CNN je možná pouze přes GPU se také potvrdilo. Při přesunu práce z CPU na GPU totiž výrazně klesne doba zpracování. Zároveň využití GPU se zvedne pouze nepatrně.

Je vidět, že první dvě zmíněné metody – tedy prahování a MSER – nedosahují uspokojivých výsledků. Hodnota AP menší než 1 % pro prahování značí, že tato metoda je naprosto nevhodná. Bylo by potřeba ještě mnoho zlepšení, aby dosáhla alespoň použitelných výsledků. Metoda MSER je na tom lépe – mAP kolem 10 % značí, že metoda má potenciál být užitečná. Při lepším nastavení jejich parametrů by bylo možné zvednout hodnotu mAP určitě výše.

Jasným vítězem je YOLO. Jenom to potvrzuje teorii, že CNN jsou velmi dobrým nástrojem na zpracování obrazu a hledání informací v něm. Hodnoty mAP mnohonásobně překonávají konvenční metody. Nejmenší testované rozlišení – tedy 96x96px – dosahuje hodnot mAP kolem 40 % a pro lepší detekci je třeba zvýšit vstupní rozlišení.

Po zvýšení rozlišení na 192 nebo 256 se očekávatelně zvednou hodnoty mAP, ale i potřebný čas na interferenci. Hodnoty mAP kolem 90 % jsou již perfektní. Čas interference kolem 2ms na GPU je dostatečný. Jedná se přibližně o 500 FPS. To je více, než jsou kamery schopny generovat.

Ze všech výše vypsanych metod byla nakonec zvolena ta nejpřesnější – tedy Yolov3-Tiny s rozlišením 256x256. Všechny metody byly vyzkoušeny na testovacím záběru z kamer a výsledky jsou vidět v příložených videích pojmenovaných podle jednotlivých metod.

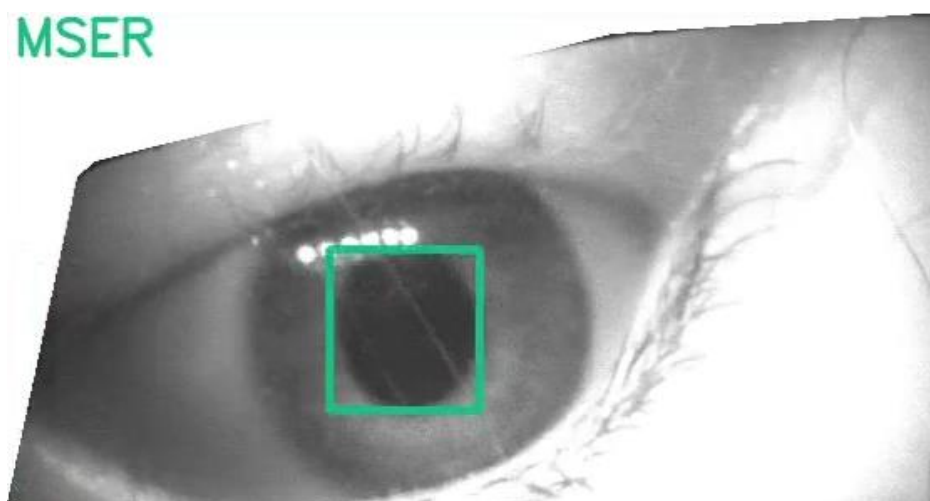
Metoda	mAP [%]	Čas OpenCV CPU [ms]	využití CPU [%]	Čas darknet CPU [ms]	využití CPU [%]
Prahování	0,52	6,00	5	-	-
MSER	9,81	9,50	7	-	-
YOLOv3-Tiny 96px	44,00	5,50	40	12,00	75,00
YOLOv3-Tiny 192px	87,00	7,80	49	20,00	80,00
YOLOv3-Tiny 256px	91,00	11,00	52	32,00	80,00

Tabulka 2 – Porovnání jednotlivých vyzkoušených metod na procesoru Intel i7-7700K. Lze vidět, že pouze CNN modely dosahují dostatečných výsledků mAP. Zároveň se potvrdilo, že CPU není vhodný prostředek pro interferenci CNN. Dokonce i přes všechny CPU optimalizace, které knihovna OpenCV obsahuje.

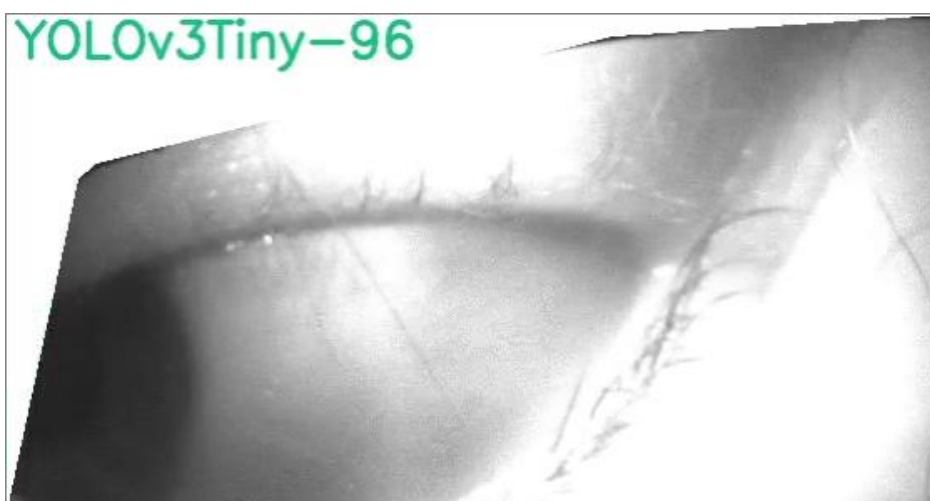
Metoda	mAP [%]	Čas OpenCV GPU [ms]	využití GPU [%]	Čas darknet GPU [ms]	využití GPU [%]
YOLOv3-Tiny 96px	44,00	2,5	12,00	1,5	5,00
YOLOv3-Tiny 192px	87,00	2,8	13,00	1,8	6,00
YOLOv3-Tiny 256px	91,00	3	14,00	2,2	6,00

Tabulka 3 – Porovnání rychlosti inference GPU na dvou implementacích – OpenCV a darknet. Zde je vidět obrovské zrychlení v porovnání s CPU verzí, kde časy kolem 2ms jsou již dostačující. Na rozdíl od CPU implementace, kde OpenCV bylo lepší než darknet, se zde role obrací. Darknet je rychlejší při menších nárocích na prostředky. Je to dáno především tím, že darknet je zaměřen pouze na sítě typu YOLO a přináší tedy mnohá vylepšení.

Byla vyzkoušena i rychlost a zatížení při použití NVIDIA 2080 Ti. Ta nabízí oproti kartě 1080 Ti hrubý výkon vyšší zhruba o 20 %. 2080 Ti byla zhruba stejně rychlá při menším zatížení GPU. Tato síť je totiž již tak malá, že nelze plně využít celé GPU. Přidávání dalších výpočetních jednotek tedy výkon nezvyšuje. Pro ještě rychlejší časy je třeba zvýšit výkon jedné výpočetní jednotky např. přetaktováním.



Obrázek 8 – Příklad správné detekce za pomoci metody MSER



Obrázek 9 – Příklad nedetekované zornice. Lze vidět splývání zornice a duhovky do jedné plochy. Obzvláště v krajních pozicích je tento jev velmi častý. Pouze metody YOLOv3-Tiny se vstupním rozlišením 192 a 256 pixelů jsou schopny detekovat i tyto krajní případy.

4 Část druhá – zjištění bodu uživatelova zájmu

Druhá část této bakalářské práce se bude zabývat zjištěním, kam se uživatel dívá. Využije k tomu data získaná v první části, která zpracuje a převede v oblast zájmu. V této kapitole budou porovnány dvě metody, které mohou sloužit pro převod informací z prostoru souřadnic kamery do souřadnic displeje. Dále budou vypsány různé metody, které jsou použity pro řešení daného problému u jiných náhlavních souprav zařízení.

4.1 Přehled metod určených k zjištění bodu zájmu

Výborný přehled všech dostupných metod, které zjišťují bod uživatelova zájmu, se nachází v článku. [25] Pochází sice již z roku 2010, avšak obsahuje obecné znalosti, které se nemění. Článek rozděluje metody na dva hlavní druhy. Obě dvě metody ke své funkci potřebují detekované informace z oka, tedy hlavně zornici a diody. Jejich rozdíl spočívá v tom, jak s informacemi následně pracují.

4.1.1 Metody založené na regresi

První druh se zabývá metodami založenými na regresi. Ty předpokládají, že existuje přímé mapování mezi pozicí zornice v obraze a pozicí uživatelova zájmu. Toto mapování je typicky vyjádřeno polynomiálním modelem nebo pomocí neuronových sítí. Možnosti se však neomezují pouze na mapování zornice, neboť existují i metody, které mapují pozici odrazu diod nebo vektor zornice a diod.

Pro regresní metody je typicky nutná kalibrace, při které se najdou správné parametry do modelu. Kalibrace je provedena pomocí několika bodů, které jsou postupně uživateli zobrazeny a zaznamenány jejich pozice. Ze znalosti těchto pozic se vypočtou správné parametry do modelu.

4.1.2 Metody založené na 3D modelu oka

Druhou velkou kapitolou, kterou článek zmiňuje, jsou metody založené na 3D modelování oka. Pozice zájmu lze jednoduše vyčíst z modelu, jako průnik vektoru pohledu a předmětu ve scéně. Model využívá faktu, že všechny lidské oči mají podobný tvar a velikost. Jejich parametry jsou tedy předem známy. Ani tato metoda se neobejde bez detekce důležitých částí oka, tedy např. zornice.

Pro kompletní konstrukci modelu je však zapotřebí znalost celého snímacího i snímaného systému – tedy pozice kamery, světla a hlavy. Pozice stacionárních objektů se dá zjistit předem pomocí kalibrace. Avšak pozice hlavy se neustále mění, a proto je potřeba s tím počítat, a ve výsledném modelu pozici znovu vyhodnocovat. Může se tak stát například pomocí externího snímání hlavy. Případně nasazením celého snímacího zařízení přímo na uživatelovu hlavu, což způsobí nezávislost celého řešení na pozici hlavy. Naštěstí toto omezení není u VR nutno řešit, jelikož je celá soustava neměnná vůči hlavě.

Ani toto řešení se neobejde bez kalibrace. Ta jde odstranit přidáním více diod a kamer do systému. V článku [26] jsou uvedeny možné kombinace, které vedou k odstranění kalibrace. V případě jedné kamery a jedné diody nelze spolehlivě vyřešit bezkalibrační metodu bez přidání dalších informací o systému, jako např. pozici hlavy nebo vzdálenost oka od kamery. V případě jedné kamery a více diod již lze vytvořit a spočítat matematický model, který bude zkalibrován pouze jednou pro každého uživatele. Při této kalibraci se najdou 3 specifické parametry oka daného uživatele. Tyto 3 zmíněné parametry jsou velice podobné pro všechny lidi a je možné je dosadit předem, udělat tedy metodu plně bezkalibrační. Poslední kombinací je poté více kamer

a více diod, kdy je metoda již plně bezkalibrační. Samozřejmě s rostoucím počtem kamer a světel roste přesnost celého řešení.

Požadavky každé ze dvou výše uvedených metod určují jejich použití. Regresní metody nepotřebují mnoho vybavení a dost často si vystačí pouze se základní webkamerou. Z tohoto důvodu se proto hodí převážně na domácí použití. Nevýhodou je pak nutnost kalibrace. Zato modely založené na 3D modelu oka kalibraci uživatele většinou nepotřebují. Zato vyžadují více měřicího vybavení a jeho správnou kalibraci. Z toho důvodu se spíše hodí do firemní nebo akademické sféry, která má prostředky na zajištění daného vybavení.

4.2 Metody řešení u jiných náhlavních souprav

Následující kapitola bude pojednávat o historii a současnosti eye-trackingu ve světě VR. Tedy jaké náhlavní soupravy danou funkci nabízí, jak u nich probíhá kalibrace, ale také jaké jsou výstupní informace. Znalost těchto informací umožní lépe se připravit na daný problém a předem odhalit možné designové chyby.

Historie eye-trackingu ve VR se datuje již do roku 2015, kdy společnost pojmenovaná FOVE vydala na kickstarteru svůj první headset umožňující sledování očí. Po úspěšné kampani se rozběhly první dodávky brýlí v následujícím roce. FOVE využívá jednu infračervenou kameru pro každé oko. Pro zprovoznění eye-trackingu je potřeba bodová kalibrace před každým použitím. V České republice se tyto brýle nedají sehnat. [27]

Známou firmou, která se obecně zabývá sledováním očí, je firma Tobii. Ta nabízí různé zařízení pro sledování uživatelského zájmu. Jedná se například o přídatný modul k PC nebo klasické brýle vybavené sledováním oka. S tím, jak se firma zaměřuje pouze na eye-tracking, tak má velice dobře vyvinuté podpůrné softwary pro následné zobrazení a vyhodnocení naměřených dat. Dalším produktem bylo eye-tracking řešení integrovatelné do brýlí pro virtuální realitu. Stalo se tak v roce 2018 společně s oznámením nových brýlí StarVR. Na webových stránkách těchto brýlí se však velmi záhy objevilo oznámení o pozastavení programu pro vývojáře, které se od té doby nezměnilo. Tyto brýle také nejdou běžně koupit. [28] [29]

Tobii to se svou technologií nevzdalo, neboť se brzy objevila v dalších brýlích. Jedná se o HTC Vive Pro Eye oznámené v lednu roku 2019. Tyto brýle jsou nyní běžně dostupné a dají se koupit za cenu kolem 35 000 Kč. Použití eye-trackingu vyžaduje od uživatele projít kalibračním procesem, který se skládá ze tří částí. V první části je zajištěno, že uživatel má správně nasazený headset ve vertikálním směru. Ve druhé části je správně nastaveno uživatelské IPD. Třetí část spočívá v bodové kalibraci. Při té je uživateli postupně zobrazeno 5 bodů, rozmístěných do kříže. S největší pravděpodobností není měřena pouze pozice kalibračního bodu, ale také pohyb k němu. Tato kalibrace stačí provést pouze jednou pro každého uživatele.

Tobii také nabízí rozsáhlé informace týkající se eye-trackingu uvnitř VR. Dokumentace obsahuje vysvětlení, jak se měří vizuální úhly, jak se mění přesnost detekce se změnou pohledu, ale třeba i doporučení, jak správně navrhovat rozhraní, aby bylo přístupné pro ovládání očima.

Poslední firma, která nabízí headsety s eye-trackingem, se jmenuje Varjo. První verze jejich headsetu vyšla na začátku roku 2019 a druhá revize na konci toho samého roku. Používají svoji technologii pojmenovanou 20/20 Eye Tracker™. Mají dva módy kalibrace rozdělených podle časové náročnosti. Mód s názvem rychlý je standardní pěti bodová kalibrace. Tu je nutné provést před každým použitím. Jejich headset má také problémy s posunem náhlavní soupravy na hlavě. V jejich vývojářské dokumentaci se přímo píše, že data mohou být v takovém případě nekvalitní. Cena jednoho headsetu je 6000 EUR. [30]

Na trhu se tedy v aktuální době nachází dva funkční headsety nabízející eye-tracking. V něčem se shodují a v něčem se liší. Mezi shodu patří např. snímání oka v infračerveném světle. Oba ke své práci potřebují kalibraci, která je pětibodová. Velkým rozdílem pak je, že u HTC stačí kalibrace jednou pro každého uživatele, zatímco u VARJA je jí potřeba před každým použitím. Obě dvě řešení mají problém, kdy brýle nejsou ideálně nasazené na uživatelské hlavě. Snaží se to řešit naváděním uživatele k posunu brýlí do ideální pozice. Důvodem by mohlo být malé zorné pole kamer a následné vyjždění oka mimo jejich záběr. Dalším velkým rozdílem je také cena, kde HTC je výrazně levnější. Informace, které jsou zpřístupněny uživateli, se téměř neliší. Jedná se hlavně o 3D vektor s orientací v prostoru, ale také podpůrné informace o oku, jako např. velikost zornice nebo jestli uživatel zrovna mrknul.

4.3 Výběr metody

Jak již bylo řečeno, existují dva hlavní přístupy k zjištění bodu uživatelského zájmu – pomocí regrese nebo za pomoci 3D modelování oka. Obě dvě metody jsou v tomto případě použitelné. Regresní metodu lze použít téměř vždy a v případě 3D modelování odpadá jeden z hlavních problémů – sledování pozice hlavy.

Nakonec byl zvolen přístup pomocí regrese. Jedná se o metodu, která je mnohem jednodušší na implementaci a případné odhalení chyb. Zároveň je mnohem větší šance úspěchu, neboť při správné kalibraci systém bude fungovat a odpustí mnohé chyby. To se však nedá říci o 3D modelu, který ke své práci potřebuje mnohem více dat, která ovšem mohou být špatná. Chyby se poté nasčítají a výsledný model nebude fungovat. Nevýhoda regresní metody je nutnost recalibrace před každým použitím. Bude se jednat o velmi podobné řešení jako nabízí Varjo.

V rámci této práce bude prozkoumáno více možností kalibrace. Ty se budou měnit použitým počtem kalibračních bodů a použitým polynomem. Závěrečné porovnání proběhne na základě dvou kritérií, a to přesnosti detekce a časové náročnosti kalibrace.

4.4 Polynomiální modely

Pro výběr testovaných polynomiálních modelů posloužila tabulka z následujícího článku. [31] Daný článek se sice zabývá mapováním celého vektoru oka a zornice, ale pro základní přehled je to dostačující. Důležité je, aby polynom dokázal zachytit pohyb oka, který není lineární, ale kulový. Polynomiální model bude rozdělený pro každou výslednou souřadnici zvlášť. Bude se tedy hledat nejlepší polynom pouze pro souřadnici x a druhý pro souřadnici y . To umožní mnohem flexibilnější řešení.

Polynom, nebo také mnohočlen, je výraz v následujícím tvaru (17). Polynom je tvořen součtem několika členů, kde každý člen je tvořen vstupní proměnnou x , exponentem vstupní proměnné a koeficientem daného členu a_i . Stupeň polynomu je označení určené číslem nejvyššího exponentu. Polynomu prvního stupně se říká polynom lineární, druhého stupně kvadratický a polynomu třetího stupně poté polynom kubický.

Existuje i polynom, který má více vstupních proměnných. Tento polynom je výraz v tomto tvaru (18) a stupeň je určen nejvyšším součtem exponentů u a y v rámci jednoho členu. Pro tuto BP je důležitý hlavně mnohočlen se dvěma vstupními proměnnými, kde každá určuje jednu souřadnici v obraze.

V praxi se osvědčilo používat polynomy s malým stupněm. Mají sice větší chybu v rámci interpolace, ale dokáží lépe zachytit celkový prostor. Polynomy s vysokým stupněm mají totiž tendence přeučít se na vstupní data.

$$p(x) = \sum_{i=0}^n a_i x^i \quad (17)$$

Rovnice 17 - Zápis polynomu pomocí sumy

$$p(x, y) = \sum_{i=0}^n \sum_{j=0}^n a_{ij} x^i y^j \quad (18)$$

Rovnice 18 – Zápis polynomu se dvěma vstupními proměnnými

4.5 Chybová metrika

Pro správné vyhodnocení výsledných modelů je třeba použít dobrou metriku. Chyba bude určována nejprve nezávisle v rámci souřadnic, jelikož je každá vypočtena nezávisle na sobě. Celková chyba řešení poté musí souřadnice spojit. Pro obě dvě varianty platí, že chyba je vzdálenost mezi predikovaným bodem a bodem pravdy.

Mezi standardní metriky patří střední absolutní chyba, střední kvadratická chyba a odmocněná střední kvadratická chyba, anglicky mean average error (MAE), mean squared error (MSE) a root mean square error (RMSE).

Všechny tři fungují na principu porovnání dvojic hodnot, kde jedna z nich je predikovaná a druhá naměřená. Jejich rozdíl spočívá v tom, jak s daným rozdílem dále pracují. Kvadratické modely umocňují chybu na druhou. To má za následek, že větší chyby mají v celkovém výsledku mnohem větší váhu. Pro lepší představu výsledných dat se však nehodí vědět informaci o chybě v druhé mocnině, ale v první. Z tohoto důvodu bude jako hlavní metrika použita MAE a RMSE.

$$MAE(x, y) = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (19)$$

Rovnice 19 – Výpočet chyby MAE

$$MSE(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (20)$$

Rovnice 20 – Výpočet chyby MSE

$$RMSE(x, y) = \sqrt{MSE(x, y)} \quad (21)$$

Rovnice 21 – Výpočet chyby RMSE

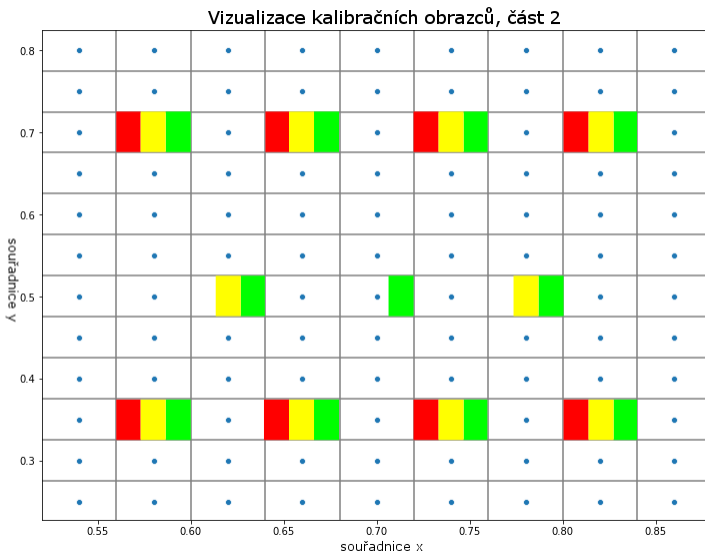
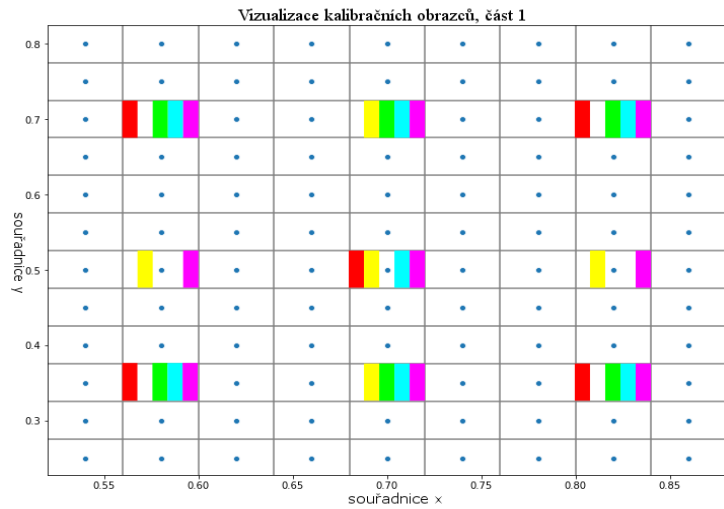
4.6 Postup měření

Celkem se testovacího měření zúčastnilo deset osob. Měření spočívalo v postupném zobrazení 108 bodů uvedených v nákresech níže jako modré tečky. Měření nepokrývá celý prostor displeje, ale pouze jeho výřez. Některé oblasti displeje totiž nejdou přímo zaměřit okem, ale jsou zde pouze pro zlepšení periferního vidění.

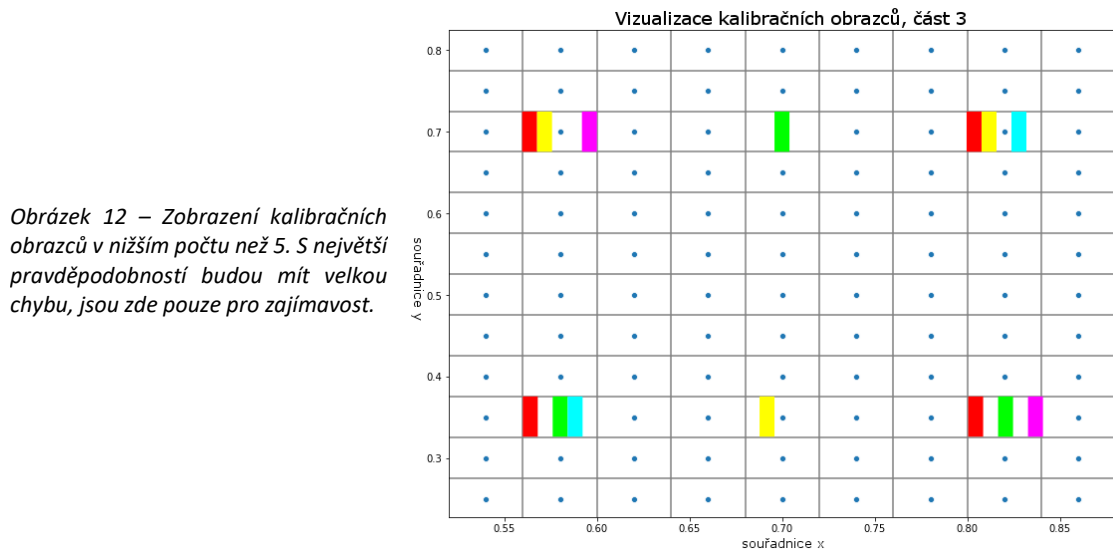
Pro správné zaměření bodu a eliminaci chyb byl každý bod zaměřen vícekrát a zprůměrován. Po každém zaměřeném bodu dostal uživatel krátký čas, aby mohl přejít očima na další bod a neovlivňoval tím jeho pozici. Měření jednoho uživatele při tomto nastavení trvá kolem 5 minut.

Z výše uvedené množiny bodů byla poté vybrána menší část, která sloužila jako referenční body pro kalibraci. Vybrané body se různí podle jejich počtu ale i polohy. Všechny kalibrační množiny jsou vyznačeny v nákresu.

Obrázek 10 – Zobrazení kalibračních obrazců ve variantě od 5 do 9 bodů. Lze vidět, že obrazec s body je zde použit dvakrát – jednou v rozestavení do kříže a jednou do plusu.



Obrázek 11 – Zobrazení kalibračních obrazců ve variantě od 8 do 11 bodů. S přibývajícím počtem kalibračních bodů by se měla zlepšovat dosažitelná odchylka.



Obrázek 12 – Zobrazení kalibračních obrazců v nižším počtu než 5. S největší pravděpodobností budou mít velkou chybu, jsou zde pouze pro zajímavost.

4.7 Nehomogenní lineární optimalizace

Zvolené polynomy a kalibrační body je třeba spojit, neboť samy o sobě nemají žádnou informaci. Polynom dokáže predikovat data až teprve ve chvíli, kdy má určené koeficienty. K jejich výpočtu právě slouží kalibrační body, které dodají potřebná data.

Řešení tohoto optimalizačního problému vede na maticovou rovnici, která má počet řádků stejný jako počet kalibračních bodů. Počet proměnných je stejný jako počet koeficientů v polynomu. Z toho vyplývají různé kombinace, které mohou vyvstat. Tyto kombinace se dají zařadit do tří kategorií, rozdělených dle porovnání počtu proměnných a rovnic. Každá kategorie bude mít svá specifika a bude se jinak řešit. Pro všechny kombinace bude předpokládáno, že v matici nejsou žádné lineárně závislé řádky ani sloupce. To je možno si dovolit díky chytrému postavení kalibračních bodů. Příklad maticové rovnice pro výpočet koeficientů polynomu $\alpha x^2 + \beta x + \gamma xy + \delta$ lze vidět v (23).

V případě, že soustava má více rovnic než neznámých, tak se jedná o soustavu přeuročenu. Pro tu neexistuje žádné přesné řešení. Rovnice se však dá řešit přibližně, kdy se hledá řešení, které minimalizuje součet rozdílů očekávané a predikované hodnoty. Takovému řešení se říká řešení ve smyslu nejmenších čtverců. (22)

Velmi podobně jako přeuročenu soustavu lze řešit i soustavu nedourčenu, tedy s menším počtem rovnic než neznámých. Zde je nekonečně mnoho řešení, které danou soustavu řeší. Je proto třeba najít řešení, které soustavu řeší nejlépe. Pro daný problém se osvědčilo řešení, jehož vektor má nejmenší normu. Tedy vzdálenost od počátku.

Poslední možná kombinace má stejný počet rovnic i neznámých. Taková rovnice má přesně jedno řešení, které bude použito.

Mohlo by se zdát, že řešení s nulovou chybou bude lepší než řešení s nějakou chybou. Avšak v daném případě, kdy celý mapovaný prostor je výrazně nelineární, je lepší zvolit řešení sice přibližné, ale zato lépe popisné. Čím více informací o celém mapovaném prostoru je zakódováno v maticové rovnici, tím lepší predikce daného prostoru. Pro další kroky tedy bude upřednostňováno řešení s menším počtem polynomiálních členů než s větším.

Nelinearita celého prostoru vzniká na mnoha místech. První místo je samotná kamera, která snímá skrze čočku a její obraz má na krajích distorzi. Další nelinearita vzniká na čočce uvnitř brýlí XTAL. Také oko, které je kulové a zornice má po něm kulový pohyb, vnáší do systému nelinearitu.

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2 \quad (22)$$

Rovnice 22 – Chyba optimalizovaná v rámci metody nejmenších čtverců

$$\begin{pmatrix} x_1^2 & x_1 & x_1 y_1 & 1 \\ x_2^2 & x_2 & x_2 y_2 & 1 \\ x_3^2 & x_3 & x_3 y_3 & 1 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} \quad (23)$$

Rovnice 23 – Příklad rovnice pro výpočet koeficientů polynomu $f(x,y) = \alpha x^2 + \beta x + \gamma xy + \delta$

4.8 Výsledek měření

Na každý dataset bylo vyzkoušeno přes 80 kombinací měřících bodů a polynomů. Výsledkem každé kombinace bylo vypočítání MAE a RMSE chyb. Pro jejich vizualizaci se nejlépe hodí

krabicový graf. Ten umožňuje velice přehledně vidět jaká je průměrná chyba, ale také jaký je její rozptyl.

Na výsledné modely lze koukat nezávisle v rámci souřadnice pouze do jisté míry. Mohou mít každá jiný polynom, avšak kalibrační obrazec musí mít stejný. To je z důvodu, aby se nemuselo kalibrovat pro každou souřadnici zvlášť. Zbytečně by se tím prodloužila doba kalibrace. Je proto výhodné si u každé souřadnice určit, které kalibrační modely dosahují dostatečných výsledků a poté najít jejich společný průnik. A na výsledném průniku najít finální kombinaci.

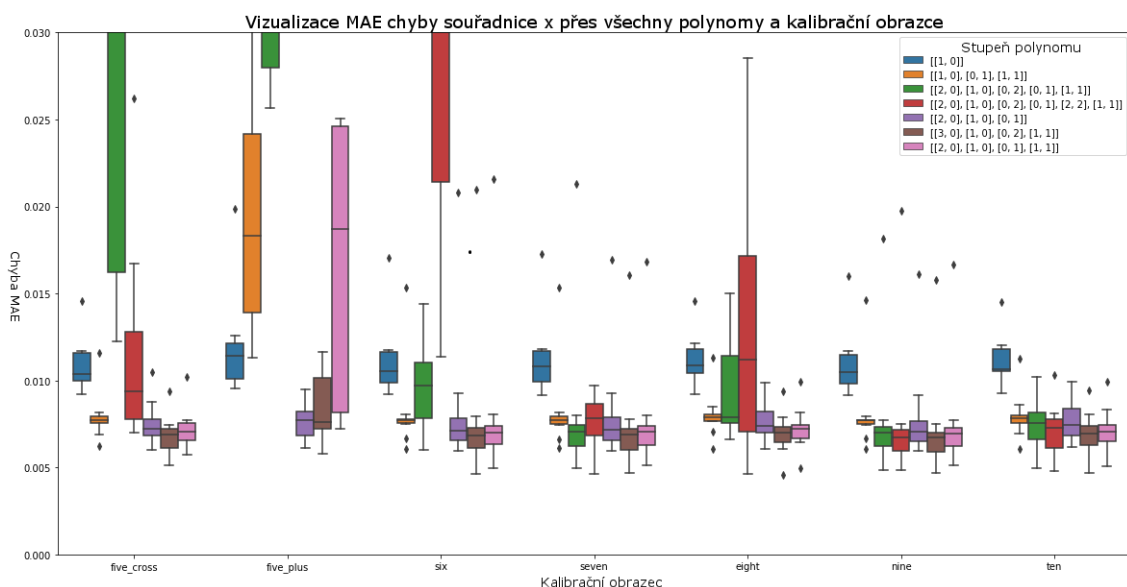
Je samozřejmě možné řešit chybu souřadnic dohromady od začátku. To by však přineslo mnohem více možných kombinací, neúměrně prodloužilo dobu vyhodnocení, a i grafy by byly nepřehledné. Zároveň by nebylo zřejmé, na kterou souřadnici je třeba se více zaměřit.

4.8.1 Výsledek měření x

V rámci x souřadnice je situace docela jednoduchá. Lze vidět, že výsledná chyba záleží hlavně na kalibračním obrazci a až poté na zvoleném polynomu. Od sedmi kalibračních bodů již všechny polynomy dosahují téměř stejných výsledků a nezlepšují se. Pouze nejjednodušší zvolený polynom, tedy lineární funkce, nabízí horší výsledek. Takto jednoduchý polynom nedokáže dobře popsat celou sledovanou soustavu.

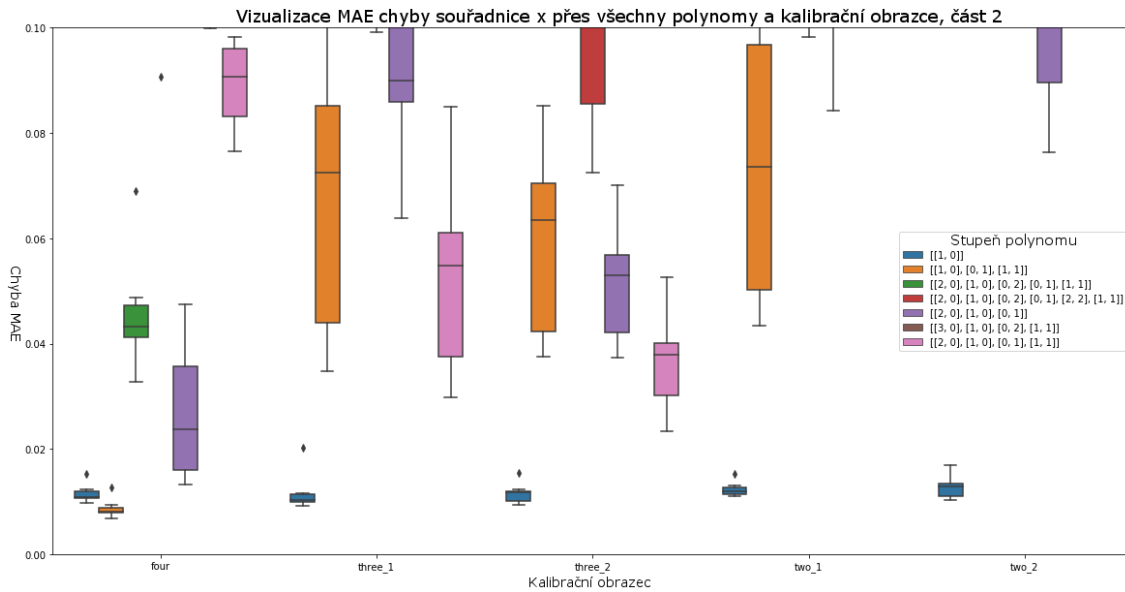
S vyšším počtem kalibračních bodů než sedm, se již chyba téměř nezlepšuje, zato doba strávená kalibrací ano. Je tedy možné tyto kalibrační obrazce vyřadit. S klesajícím počtem je důležité zvolit správný polynom, jelikož ty složitější mají větší tendence k chybám. Při správně zvoleném polynomu lze dosáhnout téměř stejné chyby jako při sedmibodové kalibraci v každém kalibračním obrazci do pěti bodů včetně.

Velice zajímavá je situace s nižším počtem kalibračních bodů než 5. Tyto modely byly přidány spíše pro ukázkou, avšak i čtyřbodový kalibrační model se ukázal jako funkční. Největší překvapení byl dvoubodový model, který i přes velmi omezený vzorek dat dokázal alespoň částečně predikovat.



Graf 2 – Porovnání naměřené chyby v rámci x souřadnice. Chyba je rozdělena jak přes kalibrační obrazce, tak přes polynomy. Zápis polynomu je v následujícím tvaru. Každý člen daného polynomu je popsán dvojicí čísel, což jsou stupně polynomu u x a y souřadnice. Absolutní člen je přidán automaticky.

Např. $[[1, 0], [0, 1], [1, 1]]$ je následující polynom – $p(x, y) = a * x + b * y + c * x * y + d$

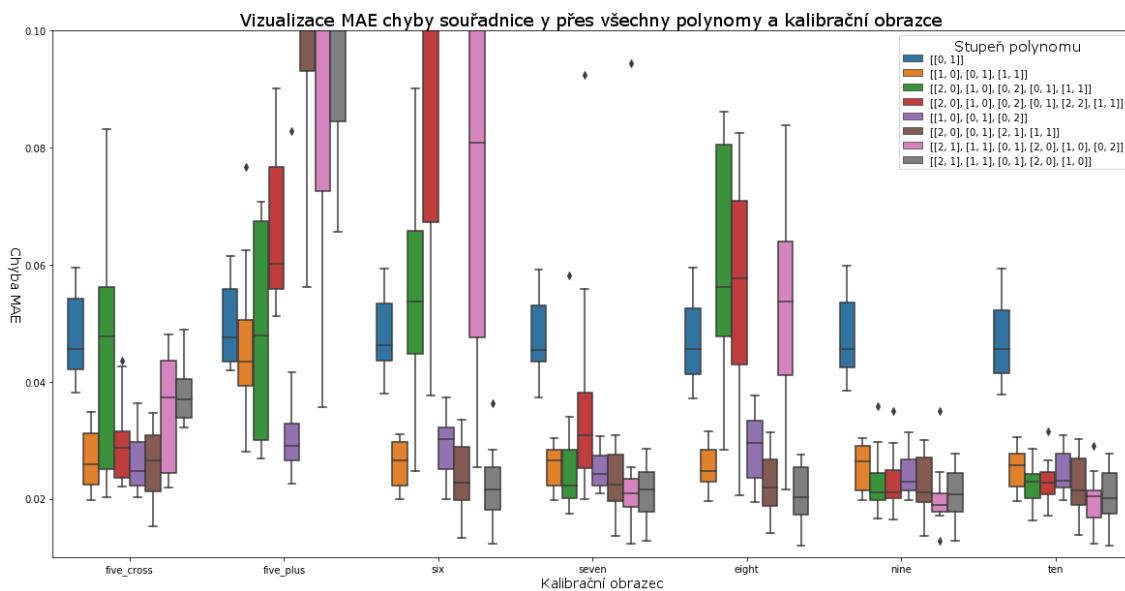


Graf 3 - Pokračování grafu výše, který byl rozdělen na dvě části kvůli lepší přehlednosti. Zde jsou vidět pouze kalibrační obrazce s nižším počtem bodů, než je 5. Lze vidět, že výsledky jsou výrazně horší než v případech s pěti a více body. Je třeba upozornit, že tento graf má jinak nastavené měřtko v ose y.

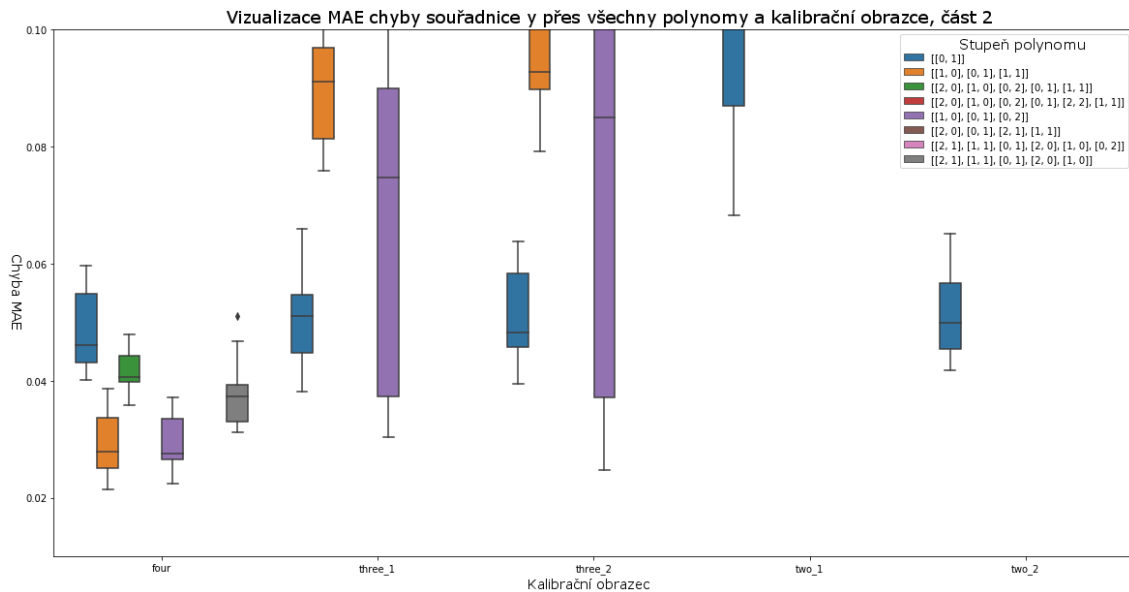
4.8.2 Výsledek měření y

V rámci souřadnice y je situace podobná jako u souřadnice x. Model se sedmi kalibračními body dosahuje velmi dobrých výsledků a se zvyšujícím počtem bodů se již chyba nezlepšuje. Lze vidět, že modely, které nemají kalibrační bod ve středu mají tendence být horší. Lze to vidět u modelu s osmi kalibračními body, který je celkově výrazně horší než model se sedmi. Tato situace nastává i u modelu se šesti body, který je horší než model s pěti. Výsledný model tedy musí obsahovat kalibrační bod uprostřed pro menší chybu.

Zároveň lze vidět, že u pěti kalibračních bodů je rozmístění do kříže mnohem lepší než rozmístění do plusu. Křížové rozmístění je pouze mírně horší než sedmibodová kalibrace.



Graf 4 – Porovnání rozptylu chyby v rámci y souřadnice. I zde se mezi nejlepší kalibrační obrazce řadí sedmibodová a pětibodová do kříže. Celková dosažitelná chyba je však mnohem horší než v případě souřadnice x. U sedmibodové lze v rámci x souřadnice dosáhnout chyby kolem 0,07, zato zde se chyba pohybuje na úrovni 0,23. Tedy více než třikrát více.



Graf 5 - 1 v případě y souřadnice byl graf rozdělen na dvě části kvůli lepší přehlednosti. Situace je zde velmi podobná jako v případě souřadnice x. Čtyřbodová již nedosahuje dostatečných výsledků a dvoubodová stále dokáže predikovat, i přes velice malý počet vstupních dat.

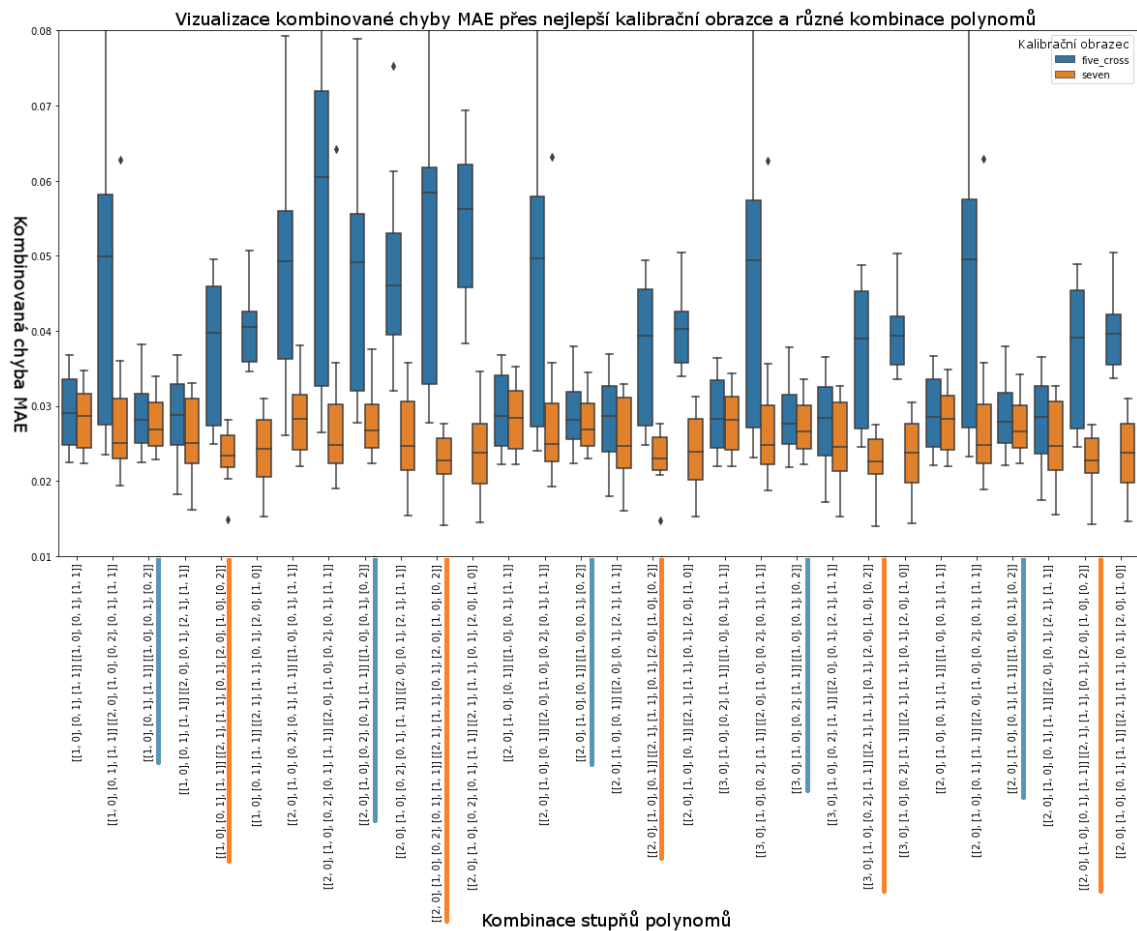
4.8.3 Výsledek měření dohromady

Jak již bylo řečeno, tak souřadnice musí mít stejný kalibrační obrazec. Jak u souřadnice x, tak u souřadnice y vyšly do užšího výběru kalibrační obrazce s pěti a sedmi body. V Grafu č. 6 jsou zobrazeny všechny možné kombinace dvojic polynomů. Lze vidět, že sedmibodová kalibrace je vždy přesnější než kalibrace pětibodová. To samozřejmě dává smysl, vzhledem k výsledkům popsaným výše.

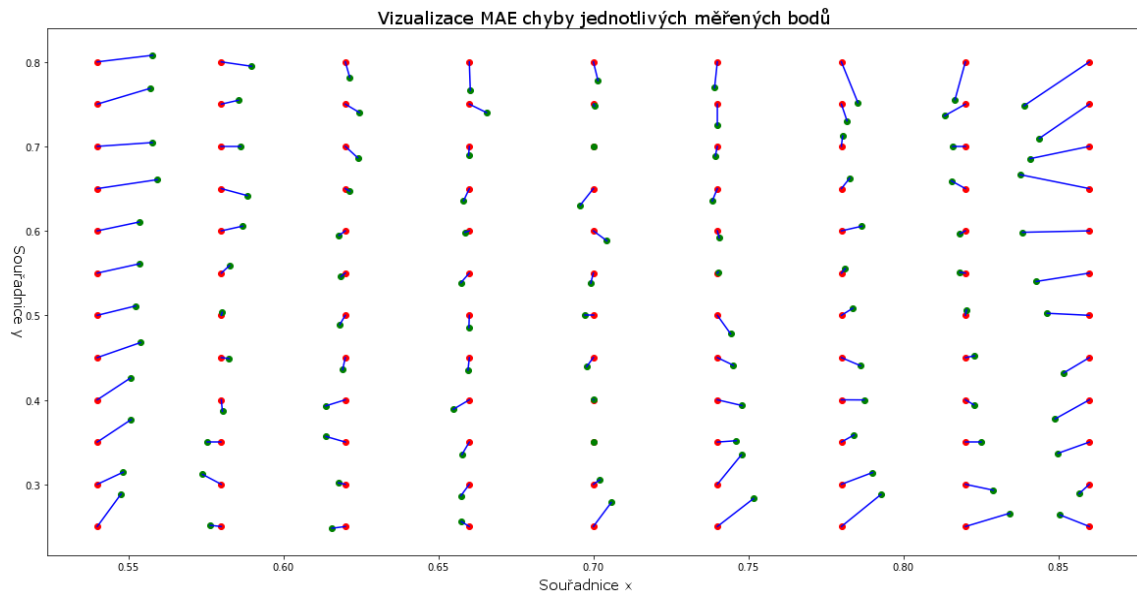
V pětibodové konfiguraci lze vybrat více kombinací polynomů, které dosahují nejlepších výsledků. Všechny je spojuje polynom pro souřadnici y, který je stejný. Je to zároveň ten stejný polynom, který dosahuje nejmenší chyby pouze v souřadnici y. Všechny tyto polynomy jsou vyznačeny v Grafu č. 7.

V sedmibodové konfiguraci je situace úplně stejná jako v případě pětibodové. Existuje množina dvojic polynomů, které dosahují téměř stejných nejlepších výsledků. Spojuje je zase to, že mají stejný polynom pro souřadnici y. Lze z toho usoudit, že není do budoucna třeba řešit všechny kombinace polynomů v rámci kalibračního obrazce. Stačí pouze vzít polynomy, které dosáhly nejmenší chyby v rámci x a y souřadnice.

Bylo empiricky změřeno, že na jeden kalibrační bod stačí 2,5 sekundy. V daném případě trvá celá pětibodová kalibrace přes obě oči 25 sekund, sedmibodová poté 34. Jelikož každý z vybraných kalibračních obrazců má své výhody, tak budou zachovány oba dva. Výsledný polynom bude vybrán tak, aby měl co nejméně členů.



Graf 6 - Výsledné hodnoty chyby dohromady přes souřadnice x a y. Modrou čarou jsou označeny kombinace, které jsou nejlepší v rámci pětibodové kalibrace. Dosahují průměrných hodnot chyby kolem 0,028. Oranžovou barvou jsou označeny nejlepší modely v rámci sedmibodové kalibrace. Jejich průměrná chyba je nižší – kolem 0,024.



Graf 7 – Vizualizace chyby jednotlivých měřených bodů. Červené tečky označují pravdu, zelené tečky predikci a modrá čára je jejich spojnice. Delší modrá čára označuje větší vzdálenost, a tedy i chybu.

Lze vidět, že uprostřed grafu je chyba velmi malá a zvětšuje se směrem ke kraji. Je to dáno především horší detekční schopností oka, když je v krajních polohách. Měření se shoduje s vizualizací z SDK firmy Tobii. Ta také ukazuje, že nejlepších výsledků dosahují uprostřed a chyba roste směrem ke kraji. Viz [35]

4.9 Výsledky druhé části

Všechny naměřené výsledky byly v souřadnicích displeje v rozsahu 0 až 1. Např. hodnota 0,07 tedy znamená odchylku o 7 % v rámci displeje.

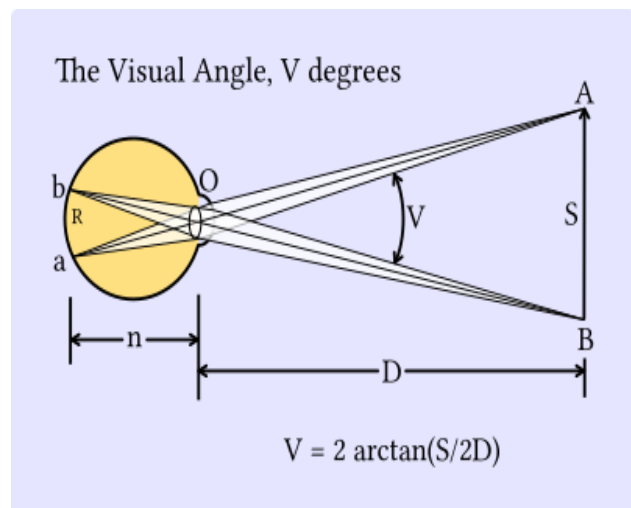
Displej u brýlí XTAL má následující parametry. Nabízí rozlišení 2560x1440 pixelů a jeho fyzické rozměry jsou 121x68 milimetrů. Jednoduchým výpočtem lze zjistit, že odchylka 1 % je asi 25 pixelů nebo přes 1mm v rámci horizontálního směru. V rámci vertikálního znamená 1 % odchylka 14 pixelů nebo asi půl milimetru.

Výsledky v rozměrech displeje jsou však těžko porovnatelné. Všechny další práce zabývající se sledováním oka zveřejňují svoji přesnost ve vizuálních úhlech. Zároveň i Tobii a další firmy zveřejňují přesnost v těchto jednotkách. Pro lepší porovnatelnost tedy budou hodnoty z rozměrů displeje převedeny do vizuálních stupňů.

4.9.1 Vizuální úhel

Vizuální úhel, anglicky visual angle, je veličina, která udává kolik stupňů z uživatelského zorného pole zabírá zvolený objekt. Je tedy závislá na velikosti daného objektu a vzdálenosti od oka. Čím větší je objekt (nebo čím je blíže), tím větší bude jeho úhlová hodnota. Dle SDK firmy Tobii zabírá palec při natažené ruce asi 2 stupně horizontálně.

Velikost daného úhlu lze velmi jednoduše vypočítat ze vzorečku (24), kde S je šířka (nebo výška) měřeného předmětu a D jeho vzdálenost od oka.



Obrázek 13 - Grafický popis vizuálního úhlu pro lepší čtenářovu představu. Dostupné z [36]

$$V = 2 \arctan\left(\frac{S}{2D}\right) \quad (24)$$

Rovnice 24 – Výpočet hodnoty vizuálního úhlu

V rámci brýlí pro VR XTAL (a všech dalších) není možné měřit úhly dle uvedeného vzorce (24). Důvodem je, že do prostoru mezi sledovaný objekt a oko se dostane skleněná čočka, která vše mění. Pro zjištění vizuálního úhlu uvnitř brýlí XTAL tedy bude použit tento výpočet. Je změřeno, že x hodnota displeje 0,55 odpovídá zhruba -40° a hodnota 0,9 poté 70° . Jedná se o rozsah, kde 0° znamená ideální střed, kam by měl uživatel koukat. Ten není shodný se středem displeje. Ze znalosti daných údajů lze velmi jednoduše vypočítat daný vizuální úhel.

Stejnou logiku lze aplikovat na získání vizuálního úhlu ve vertikální ose. Displejová hodnota 0,2 odpovídá asi -50° , hodnota 0,8 poté 50° . Vypočtené výsledky jsou uvedeny dále v Tabulce č. 4 a č. 5.

	Dosažená chyba souřadnice x			
	v souřadnicích displeje	v pixelech	v milimetrech	ve vizuálních stupních
Pětibodová kalibrace	0,007	17,92	0,847	2,12
Sedmibodová kalibrace	0,007	17,92	0,847	2,12

Tabulka 4 - Dosažená chyba souřadnice x

	Dosažená chyba souřadnice y			
	v souřadnicích displeje	v pixelech	v milimetrech	ve vizuálních stupních
Pětibodová kalibrace	0,026	37,44	1,846	4,33
Sedmibodová kalibrace	0,022	31,68	1,562	4,03

Tabulka 5 - Dosažená chyba souřadnice y

4.10 Diskuze k druhé části

Dosažené výsledky celého řešení lze vidět v Tabulce č. 6. V rámci souřadnice x je chyba minimální, kolem dvou vizuálních stupňů. To je hodnota velmi přesná, na hranici lidského vnímání.

Mírně horší výsledky jsou v rámci souřadnice y, kde je chyba kolem 4 stupňů. To už je chyba, kterou člověk má šanci rozlišit. Avšak přesto je to velmi dobrý výsledek.

Celé řešení je na hranici možné chyby. Zobrazovaný obrazec, tedy diamant, je sám veliký asi dva vizuální stupně. Každý uživatel při jeho sledování bude koukat trochu jinak, což ovlivní výsledky. Pokusy se zmenšením diamantu se neosvědčily, neboť byl až moc malý. Mohlo by možná pomoci změnit kalibrační obrazec na něco, co lépe definuje jeden bod, jako např. terč.

Ve srovnání s dalšími headsety na trhu vypadá přesnost tohoto řešení špatně. Ostatní headsety se chlubí přesností pod jeden vizuální stupeň, oproti zde navrženému řešení s chybou kolem 3 stupňů. Všechno je to dáno nejednotnou testovanou metodikou. Dosažená chyba jednoho stupně je pouze na velmi malém prostoru. V dokumentaci HTC Vive Eye Pro se přímo píše, že chyba do jednoho stupně je pouze v rámci 20 stupňů FOV. Tedy pouze v ideálním případě, kdy člověk kouká do středu čočky a displeje. [32]

Velkou výhodou brýlí XTAL je jejich velké FOV. V rámci přesnosti eye-trackingu se však projeví spíše negativně. Právě díky tomu, že v krajních pozicích výrazně klesá přesnost. Nejlépe je to vidět z Grafu č. 7 umístěného výše. Chyba uprostřed měřeného rozsahu je téměř nulová a výrazně roste do krajů. Kdyby se zmenšil rozsah, tak by výrazně klesla celková chyba. S tím by klesla i chybová hodnota vizuálního úhlu na srovnatelné hodnoty.

	Dosažená celková chyba ve vizuálních stupních
Pětibodová kalibrace	3,23°
Sedmibodová kalibrace	3,08°
FOVE	<1°
HTC Vive Eye PRO	1°
VARJO	<1°

Tabulka 6 - Porovnání hodnot přesnosti s dalšími headsety

5 Demo projekt

Další a poslední částí této BP je část zabývající se demo aplikací. Ta je velmi jednoduchá, ale stále dokáže názorně předvést, jestli dané řešení funguje.

Projekt se skládá ze dvou částí. První část je pouze knihovna, která řeší vše kolem eye trackingu. Druhá část je poté aplikace, která danou knihovnu využívá. Knihovnu je následně možné vzít a integrovat do dalších aplikací.

Po spuštění aplikace se uživateli zobrazí demo scéna. Ta obsahuje hlavu opice Suzanne, která je známá z 3D modelovacího programu Blender. Při pohybu hlavy se adekvátně mění zobrazovaná scéna jako u každé aplikace ve VR.

Program je velmi jednoduchý a má pouze 3 klávesy, které ho ovládají. První klávesa je F4. Ta slouží ke změření a správnému nastavení uživatelova IPD. Druhá klávesa – DELETE – spouští sledování očí. Klávesa END resetuje celou kalibraci.

Po zmáčknutí DELETE se inicializuje celé sledování očí. Otevře se spojení s kamerami, do paměti se načte neuronová síť, najde se správný ořez obrazu z kamer dle daného ipd a úrovně přiblížení. Tedy všechny kroky, které jsou popsány výše.

Po této spouštěcí sekvenci započne kalibrační část. Celá původní scéna se ztmavne a viditelně se začnou uživateli zobrazovat kalibrační body. Jedná se o 7 bodů na každé oko, celkem tedy 14.

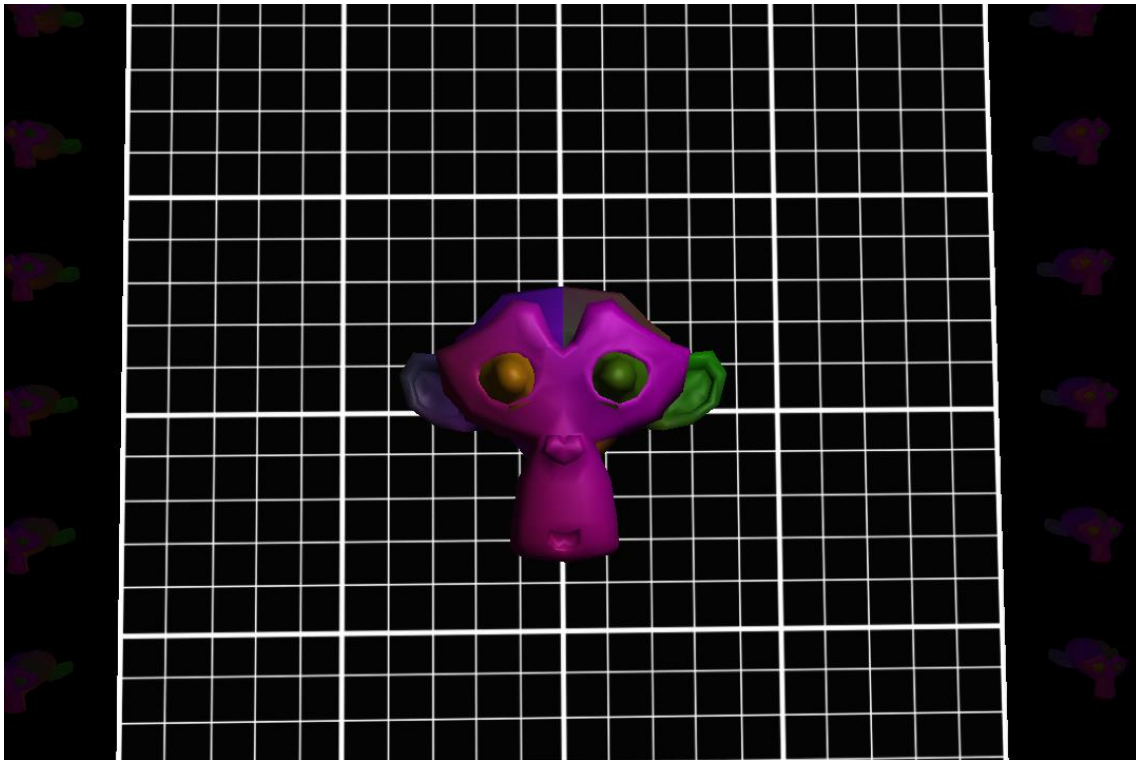
Kalibrační bod je zobrazen po 2 sekundy pro správné zaměření a odfiltrování chyb. Po každém bodu má uživatel 0,5 sekundy pro přejetí na další bod. Tyto časové hodnoty byly nastaveny podle empirického měření. Je velký rozdíl mezi uživatelem, který nikdy předtím nekalibroval a který ano. Ten, který kalibroval, by mohl mít kalibraci mnohem rychlejší, jelikož ví, co má očekávat. Hodnoty jsou nastaveny spíše střídavě, aby i uživatel bez předchozích zkušeností mohl vše pohodlně zkalibrovat.

Kalibrační bod je zobrazen jako točící se diamant, který má normálně fialovou barvu. Ta se však pro lepší uživatelovu představu změnila v šedou v případě, že se neměří. Tedy v případě změně pohledu na další kalibrační bod.

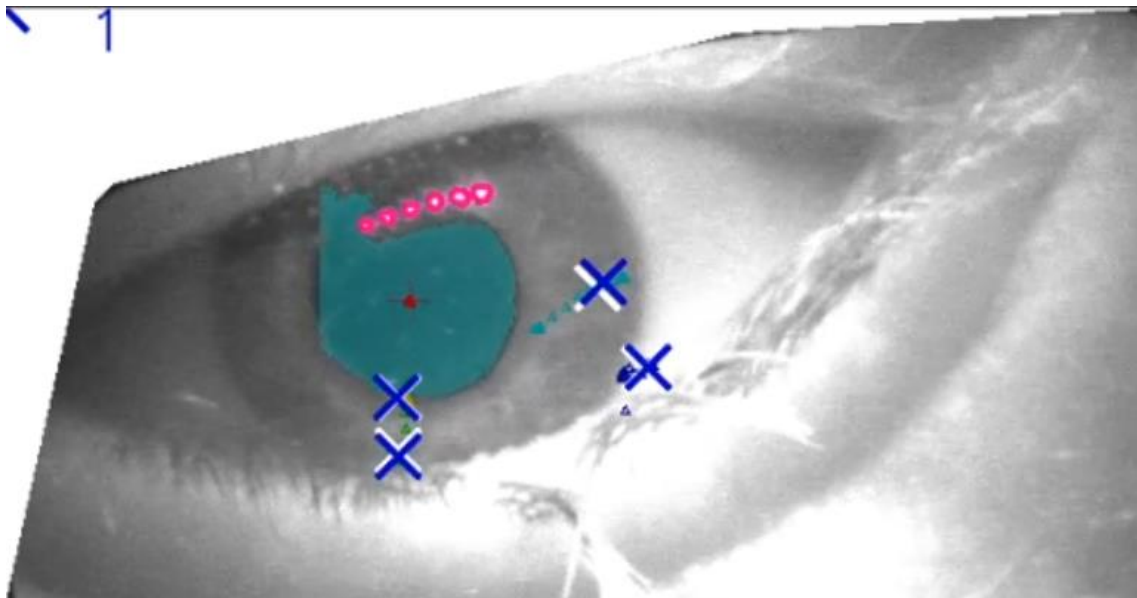
Pro lepší představu o stavu kalibrace je záběr z kamer zobrazen na monitoru. Zde je vidět, jak se uživatelovo oko hýbe a zároveň pozice bodů, které už byly nakalibrovány.

Po úspěšné kalibraci přejde aplikace do druhého módu. V něm aplikace funguje přesně naopak. V kalibračním módu uživatel sledoval kalibrační body, avšak nyní již body sledují uživatele. To znamená, že kam se uživatel podívá, tam se zobrazí bod.

Pro zlepšení přesnosti celého řešení byl přidán filtr, který zprůměruje posledních 15 hodnot a ty použije jako výslednou pozici pohledu. Zároveň je přidáno zprůměrování vertikální pozice z obou očí pro ještě přesnější data.

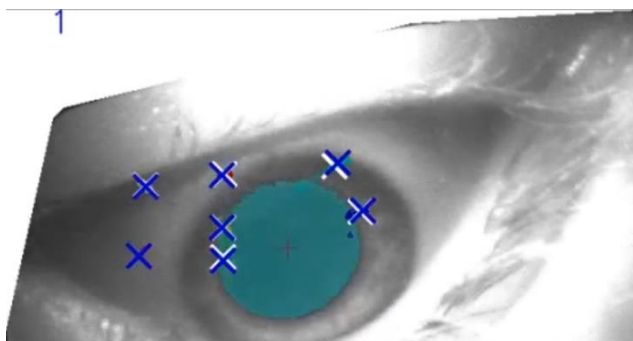


Obrázek 14 – Úvodní scéna celé testovací aplikace. Ta obsahuje opici Suzanne a za ní čtvercovou mřížku. Ta slouží k přesnějšímu vyhodnocení predikované pozice. Uživatel má totiž šanci zaměřit se na výrazný bod a sledovat, jestli se na ten samý bod posune i diamant.



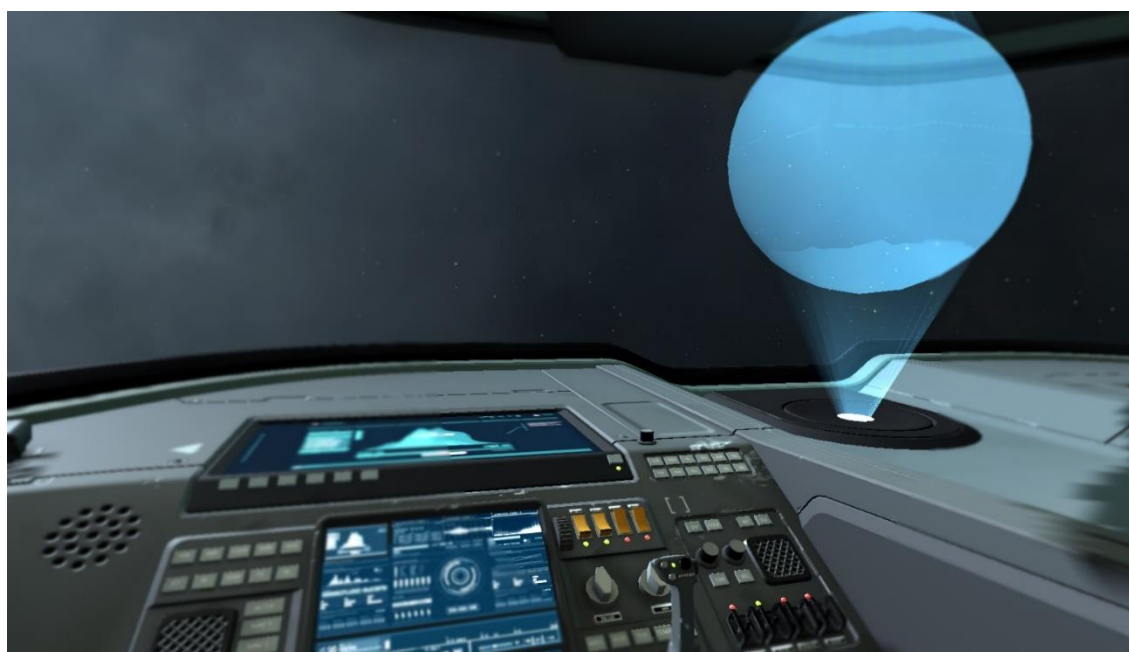
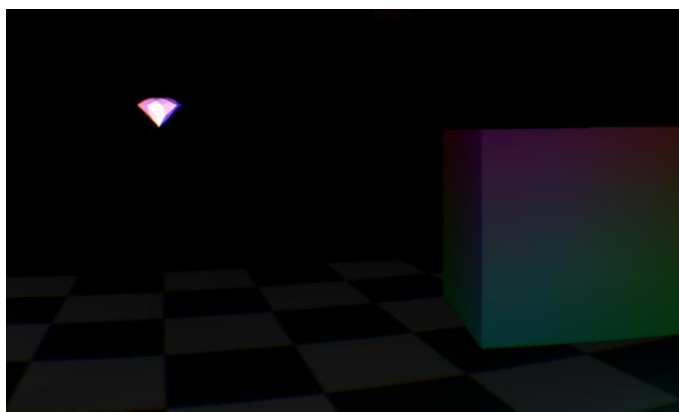
Obrázek 15 – Ukázka z kalibrace levého oka. Lze vidět červenou tečku, která určuje detekovaný střed oka. Dále lze vidět 4 modré křížky, které označují již zkalibrované body.

Ukázka pochází z novější verze aplikace, která umí i segmentovat zornici a detekovat led diody. Ty jsou zde zobrazeny růžově.



Obrázek 16 – Další ukázka z průběhu kalibrace. V daném stavu má již uživatel kalibraci provedenou, a proto je zde 7 modrých křížků. Lze vidět, že křížky tvoří podobný obrazec jako je kalibrační obrazec. To je samozřejmě správně.

Obrázek 17 – Ukázka kalibračního bodu zobrazeného jako diamant. V pozadí lze vidět zatemněná původní scéna.



Obrázek 18 – Jak již bylo řečeno, tak existuje knihovna, která řeší vše ohledně eye-trackingu. Tato knihovna se dá vzít a použít i v jiných projektech než pouze ve výše uvedeném. Zde se nachází obraz z jejího úspěšného integrování do aplikace s vesmírnou lodí. Lze zde vidět technologii foveačního vykreslování. Při té se oblast uživatelského zájmu vyrenderuje s vyšší kvalitou obrazu než zbytek scény. Ušetří se výkon a celkový dojem ze scény je lepší.

Uživatel se dívá někam do prostoru pravého kraje přístrojové desky. Velmi jasně lze vidět změnu kvality vykreslování na displeji.

6 Závěr

Virtuální realita je budoucnost. Otevírá nové možnosti a rozšiřuje ty stávající, ať už se jedná o realtime spolupráci lidí z různých koutů planety, prototypování výrobků nebo prosté hraní her. Být součástí nejlepší české firmy v daném oboru je veliká zkušenost. A vědět, že výsledek této BP se bude reálně používat po celém světě, to ještě umocňuje.

Cílem této bakalářské práce bylo navrhnout, zrealizovat a popsat řešení pro sledování očí ve virtuální realitě. Práce byla rozdělena do čtyř částí, které na sebe vzájemně navázaly. a společně tak zajistily splnění zadaného úkolu.

První část byla obecný úvod do celé problematiky sledování očí ve virtuální realitě. Tyto obě technologie se vyvíjely samostatně a mohou fungovat nezávisle. Cílem této BP bylo tyto technologie propojit. Byla tedy popsána historie jak sledování očí, tak i virtuální reality. V rámci VR byl také popsán headset XTAL, na kterém je celé řešení implementováno.

Druhá část byla zaměřena pouze na obrazové rozpoznávání. V této části bylo prozkoumáno několik způsobů, jak ve vstupním obrazu oka najít zornici. Jako nejlepší způsob se ukázalo využití konvolučních neuronových sítí (CNN), které aktuálně patří k nejmodernějšímu řešení. Řešení založené na CNN byly v detekci výrazně lepší než jiné testované metody a dokázaly si poradit i s náročnými situacemi, kdy byla zornice téměř neviditelná.

Třetí část navázala na výsledek dosažený v části druhé a dále ho rozšířila. Jejím úkolem bylo zjistit, kam se uživatel dívá, a to pouze za pomoci znalosti polohy zornice v obraze. Byly zde popsány dva hlavní obecné způsoby, které se dají k danému problému využít. Jmenovitě regresní model a model založený na 3D popisu oka a jeho okolí. Z nich byl vybrán první jmenovaný model, který byl následně rozvinut a použit.

V rámci regresního modelu bylo porovnáno mnoho způsobů regrese, které se liší jak použitým kalibračním obrazcem, tak použitým polynomem. Celkem bylo naměřeno přes 200 různých kombinací, ze kterých byly vybrány ty nejlepší a podrobeny dalšímu zkoumání. Výslednou metodu tvoří sedm kalibračních bodů a dosahuje přesnosti pod dva vizuální stupně. Na konci kapitoly bylo uvedeno srovnání s headsety, které již zkoumanou technologii sledování očí obsahují. Výsledné řešení dosáhlo srovnatelných výsledků s komerčním řešením.

Čtvrtá část byla nejkratší ze všech, ale má stejnou důležitost jako předchozí kapitoly. V této části byl popsán testovací projekt, který umožňuje ověřit, že navržené řešení funguje. Projekt se skládá ze dvou částí, kde první je knihovna starající se o všechno ohledně sledování očí a druhá část je vizualizační aplikace. Ta ke své práci využívá danou knihovnu. Zároveň je i předvedena integrace knihovny do jiných projektů.

Jednalo se o náročné zadání. K dosažení cíle bylo potřeba získat mnoho informací z různých oborů počítačových věd. I přes všechny problémy, které se při zpracování zadaného úkolu objevily, se podařilo projekt dotáhnout do konce a vytvořit řešení, které je funkční, rychlé a snadno použitelné. Sledování očí ve virtuální realitě umožňuje výrazně změnit způsob, jak se člověk ve VR cítí. Ať už se jedná o zvýšení kvality obrazu nebo interakci s prostředím pouze na základě očí. Tohle všechno je budoucnost.

Práce na zlepšení systému budou pokračovat i dále po skončení této BP. Navržené regresní řešení je již na hranici možných zlepšení. Dalším krokem může být implementace plně bezkalibrační metody, které posune uživatelův zážitek zase o kus dál. Cíl vyrobit nejlepší VR platformu současnosti stále pokračuje.

7 Reference

1. **Barnard, Dom.** History of VR - Timeline of Events and Tech Development. *Virtual speech*. [Online] <https://virtualspeech.com/blog/history-of-vr>.
2. History Of Virtual Reality. *Virtual reality society*. [Online] <https://www.vrs.org.uk/virtual-reality/history.html>.
3. **Seibert, William.** Virtual Reality Then: A Look Back at the Nintendo Virtual Boy. *Techspot*. [Online] 2017. <https://www.techspot.com/article/1085-nintendo-virtual-boy/>.
4. **A. Nicole Somani, Bayan Al Othman, MD.** Saccade. *EyeWiki*. [Online] <https://eyewiki.aao.org/Saccade>.
5. Clinical Research. *Tobii Pro*. [Online] <https://www.tobiipro.com/fields-of-use/clinical-research/>.
6. Most Pics In IKEA Catalogues Aren't Photos, They're 3D Renders. *Kotaku*. [Online] <https://kotaku.com/most-pics-in-ikea-catalogues-arent-photos-theyre-3d-re-1627713133>.
7. **Heaney, David.** Editorial: Foveated Rendering Is Essential To Consumer VR's 2nd Generation. *UploadVR*. [Online] 2019. <https://uploadvr.com/foveated-rendering-matters/>.
8. OpenCV - About. *OpenCV*. [Online] <https://opencv.org/about/>.
9. **Šonka M., Hlaváč V., Boyle R.** *Image Processing, Analysis and Machine vision, 4th edition*. Toronto, Canada : Thomson Learning, 2015. ISBN-10: 1133593607.
10. **Mallick, Satya.** Why does OpenCV use BGR color format ? *Learn OpenCV*. [Online] 2015. <https://www.learnopencv.com/why-does-opencv-use-bgr-color-format/>.
11. OpenCV Tutorials. *OpenCV documentation*. [Online] https://docs.opencv.org/master/d9/df8/tutorial_root.html.
12. **K, Sambasivarao.** Non-maximum Suppression (NMS). *towards data science*. [Online] <https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c>.
13. **Cvrček, V.** Precision a recall. *Příručka ČNK*. [Online] <https://wiki.korpus.cz/doku.php/pojmy:precision>.
14. **Hugh Davson, Edward S. Perkins.** Human eye. *ENCYCLOPÆDIA BRITANNICA*. [Online] <https://www.britannica.com/science/human-eye>.
15. **Cronin, Melissa.** Why Do We Blink So Much? Blinking Provides Mental Rest, New Research Suggests. *huffpost*. [Online] https://www.huffpost.com/entry/why-do-we-blink-so-much-mental-rest_n_2377720.
16. **Matas, Jiri and Chum, Ondrej and Urban, Martin and Pajdla, Tomas.** Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *British Machine Vision Conference*. 2002, Sv. 22, 10.
17. **Culurciello, Eugenio.** THE HISTORY OF NEURAL NETWORKS. *Dataconomy*. [Online] <https://dataconomy.com/2017/04/history-neural-networks/>.

18. **Malik, Ross Girshick and Jeff Donahue and Trevor Darrell and Jitendra.** Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv*. 2013.
19. **Farhadi, Joseph Redmon and Santosh Divvala and Ross Girshick and Ali.** You Only Look Once: Unified, Real-Time Object Detection. *arXiv*. 2015.
20. **Liao, Alexey Bochkovskiy and Chien-Yao Wang and Hong-Yuan Mark.** YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv*. 2020.
21. The PASCAL Visual Object Classes Homepage. [Online] <http://host.robots.ox.ac.uk/pascal/VOC/>.
22. **Dollár, Tsung-Yi Lin and Michael Maire and Serge Belongie and Lubomir Bourdev and Ross Girshick and James Hays and Pietro Perona and Deva Ramanan and C. Lawrence Zitnick and Piotr.** Microsoft COCO: Common Objects in Context. *arXiv*. 2014.
23. **Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun.** Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv*. 2015.
24. **AlexeyAB.** Darknet. *github*. [Online] <https://github.com/AlexeyAB/darknet>.
25. **Dan Witzner Hansen, Qiang Ji,** In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*. 2010, Sv. 32.
26. **Eizenman, E. D. Guestrin and M.** General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections. *IEEE Transactions on Biomedical Engineering*. 2006, Sv. 53, 6.
27. FOVE: The World's First Eye Tracking Virtual Reality Headset. *Kickstarter*. [Online] <https://www.kickstarter.com/projects/fove/fove-the-worlds-first-eye-tracking-virtual-reality>.
28. About. *StarVR*. [Online] <https://www.starvr.com/>.
29. StarVR Unveils Advanced Virtual Reality Headset with Integrated Tobii Eye Tracking. *tobii*. [Online] <https://www.tobii.com/group/news-media/press-releases/2018/8/starvr-unveils-advanced-virtual-reality-headset-with-integrated-tobii-eye-tracking/>.
30. Eye tracking. *VARJO Developer*. [Online] <https://developer.varjo.com/docs/native/eye-tracking>.
31. **Blignaut, Pieter.** Mapping the Pupil-Glint Vector to Gaze Coordinates in a Simple Video-Based Eye Tracker. *Journal of Eye Movement Research*. 2014.
32. Vive pro eye specs, user guide. *VIVE Developers*. [Online] <https://developer.vive.com/resources/knowledgebase/vive-pro-eye-specs-user-guide/>.
33. Precision-Recall. *scikit-learn*. [Online] https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html.
34. **Rafael Padilla, Sergio Lima Netto and Eduardo A. B. da Silva.** Survey on Performance Metrics for Object-Detection Algorithms. *International Conference on Systems, Signals and Image Processing (IWSSIP)*. 202.
35. Hardware Accuracy. *Tobii*. [Online] <https://vr.tobii.com/sdk/eye-behavior/hardware-accuracy/>.
36. Visual angle. *Wikipedia*. [Online] https://en.wikipedia.org/wiki/Visual_angle.

37. **A. Garcia-Garcia, S. Orts-Escolano, S.O. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez.** A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv*. 2017.