

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Sledování pohybu a změny výrazu obličeje v obrazech

Analysis of Movement and Face Expression Using Images

Zadání bakalářské práce

Student: **Markéta Hrabánková**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Sledování pohybu a změny výrazu obličeje v obrazech**
Analysis of Movement and Face Expression Using Images

Jazyk vypracování: čeština

Zásady pro vypracování:

Sledování a vyhodnocování výrazu obličeje je v posledních letech velmi diskutované téma. Využití může nalézt například k ovládání počítače nebo automobilu. Tato práce se zaměřuje na analýzu pohybu a výrazů obličeje za pomoci obrazů.

1. Seznamte se s metodami pro analýzu obličeje pomocí obrazů.
2. Seznamte se s dostupnými knihovnami z této oblasti (například OpenCV, Dlib).
3. Vytvořte aplikaci, která bude sledovat pohyb nebo změny výrazu v obličeji (například detekce mrkání, otevření úst, natočení obličeje).
4. Experimentálně ověřte funkčnost, přesnost a rychlost navrženého řešení.
5. Své závěry řádně zdokumentujte v textu práce.

Seznam doporučené odborné literatury:

- [1] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1867-1874.
- [2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-511-I-518 vol.1.


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


Vedoucí bakalářské práce: **Ing. Radovan Fusek, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018




doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 23. dubna 2018

Hedvika.....

Ráda bych na tomto místě poděkovala svému vedoucímu práce panu doktoru Radovanu Fuskovi za rady a pomoc při tvorbě práce.

Abstrakt

Cílem práce bylo seznámit se s metodami pro detekci a analýzu obličeje pomocí obrazů a použít získané informace a dostupné knihovny pro implementaci sledování pohybu a výrazu tváře. V navrženém řešení byly využity funkce z knihoven OpenCV a dlib. Podstatnou částí implementace je detekce zájmových bodů obličeje a jejich analýza, díky níž zjistíme rotaci a polohu hlavy, což je následně použito pro ovládání pohybu myši pomocí natáčení hlavy.

Klíčová slova: detekce obličeje, zájmové body obličeje, Histogram Orientovaných Gradientů, pozice a natočení hlavy, Haarovy příznaky, OpenCV, dlib

Abstract

The aim of this thesis was to get familiar with face detection and analysis methods and to utilize gained information and available libraries for implementation of face movement and expression tracking. The libraries used in the suggested solution were OpenCV and dlib. The substantial part of the implementation is facial landmark detection and analysis used for finding out the head pose, which is then used for mouse movement controlling via head rotation.

Key Words: face detection, facial landmarks, Histogram of Oriented Gradient, head pose, Haar features, OpenCV, dlib

Obsah

Seznam použitých zkratek a symbolů	7
Seznam obrázků	8
Seznam tabulek	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Detekce obličeje v obraze	12
2.1 Histogram Orientovaných Gradientů	12
2.2 Metoda Viola a Jones	14
3 Zájmové body obličeje	16
3.1 Metody detekce	16
3.2 Datasety	19
4 Knihovny OpenCV a dlib	22
4.1 Detekce objektů a obličeje	22
4.2 Detekce zájmových bodů obličeje	22
5 Implementace analýzy zájmových bodů obličeje	24
5.1 Postup řešení	25
5.2 Rychlost a přesnost	32
5.3 Problémy a návrhy zlepšení	37
6 Závěr	38
Literatura	39
Přílohy	42
A Zdrojové kódy aplikace na DVD	42

Seznam použitých zkratek a symbolů

HOG	– Histogram of Oriented Gradients
SVM	– Support vector machines
RGB	– Red-Green-Blue (barevný model)
LAB	– Lightness, osa a zelená - červená, osa b modrá - žlutá (barevný model)
AdaBoost	– Adaptive Boosting
DPM	– Deformable Part Models
AAM	– Active Appearance Model
LBF	– Local Binary Features
LFPW	– Labeled Face Parts in the Wild
3D	– Three-dimensional
2D	– Two-dimensional
RAM	– Random-access memory

Seznam obrázků

1	Sestavení histogramu z gradientů jedné buňky [8].	13
2	Příznaky používané při detekci [9].	15
3	Příklady počtu zájmových bodů, které mohou být na obličeji detekovány [13][12].	16
4	Ukázka výsledku detektoru z knihovny flandmark [11].	16
5	Ukázky výsledku detektoru založeného na AAM [4].	17
6	Postupné zlepšení odhadu polohy zájmových bodů u jednotlivých kroků kaskády za použití regresních stromů [3].	18
7	Ukázky výsledku detektoru založeného na regresních stromech [3].	18
8	Velikosti lokálních oblastí daných zájmových bodů při odlišných stupních kaskády u přístupu založeném na LBF [5].	18
9	Ukázky výsledku detektoru založeného na LBF [5].	19
10	Příklady obrázků z LFPW datasetu [6].	19
11	Příklady obrázků z Helen datasetu [15].	20
12	Příklady obrázků z 300-W datasetu [13].	21
13	Detekce obličeje pomocí knihoven OpenCV (kruhy) a dlib (čtverce) [10].	22
14	Detekce zájmových bodů obličeje pomocí knihoven OpenCV a dlib [18][17].	23
15	Vyobrazení jednotlivých kroků implementace.	24
16	Příklad snímku uloženého v matici.	25
17	Ukázky detekovaného obličeje.	26
18	Ukázky detekovaných zájmových bodů při různých pozicích a výrazech obličeje. .	27
19	Gesto pro kliknutí myši - otevřená ústa.	27
20	3D model obličeje pro určení pozice a rotace hlavy [14].	28
21	Ukázky směru vektoru při různém natočení hlavy.	29
22	Ukázky výsledku analýzy zájmových bodů.	30
23	Ukázka rozdílu v úhlech rotace při použití kalibrace a bez ní.	32
24	Velikosti obdélníků použitých při testování přesnosti.	34
25	Ukázky nepovedené detekce obličeje a zájmových bodů.	35

Seznam tabulek

1	Rychlost detekce a analýzy obličeje při méně častém pohybu hlavou.	33
2	Rychlost detekce a analýzy obličeje při častějším a rychlejším pohybu hlavou. . .	33
3	Rychlost analýzy obličeje a klikání myši při detekci otevřených úst.	33
4	Výsledky měření přesnosti pohybu a kliknutí myši.	36

Seznam výpisů zdrojového kódu

1	Načtení obrazu pomocí OpenCV.	25
2	Detekce obličeje pomocí dlib.	26
3	Detekce zájmových bodů obličeje pomocí dlib.	26
4	OpenCV funkce pro získání pozice a rotace obličeje.	29
5	OpenCV funkce pro získání pozice bodu (zadaného ve výchozí 3D pozici) na 2D snímku.	31
6	Funkce použité pro ovládání myši v operačním systému Linux.	31

1 Úvod

Analýza obrazu a detekce různých objektů má dnes již mnoho využití, i když je to stále náročný úkol vzhledem k počtu různých předmětů a jejich poloh. Proto se nové metody a způsoby použití stále hledají. Konkrétně detekce člověka má široké využití při analýze obrazů z pouličních kamer a nesmíme zapomenout na autonomní vozidla, která jsou na analýze obrazu závislá, a detekce chodců je důležitou součástí jejich systému. Příklad bychom ale našli i v jednodušší oblasti jako jsou fotoaparáty, kde mohou být přímo při focení zvýrazněny na displeji postavy nebo jejich obličeje. V mnoha mobilních aplikacích pro úpravu fotografií se s nalezeným obličejem dá dále pracovat a můžete si obličej například orámovat nebo třeba přidat korunku na hlavu. Sledování pohybu obličeje a jeho analýza nám dává další možnosti využití. Po detekci důležitých částí obličeje můžeme například sledovat oči a rozpoznat, kdy člověk mrknul nebo zda má oči vůbec otevřené. Tímto problémem se zabývají například výrobci aut, aby mohli sledovat řidiče, zda náhodou neusnul za jízdy. Rozpoznání emocí podle výrazu obličeje může být další oblastí zkoumání. Jelikož podle důležitých bodů obličeje můžeme určit natočení a polohu hlavy, mohou být údaje použity pro ovládání různých přístrojů jako je například počítač nebo palubní deska automobilu. Obecně kontrola různých věcí pouze za pomoci hlavy by mohla být užitečná například pro handicapované, kteří běžného ovládání za pomoci rukou nejsou schopni nebo jim to činí velké obtíže. Předpokládám, že pro všechny by bylo ovládání výrazem obličeje minimálně vítanou změnou. Nejen z těchto důvodů si myslím, že pro analýzu obličeje z obrazu nalezneme v budoucnu čím dál tím více využití.

Tato práce se zabývá problematikou detekce obličeje a jeho zájmových bodů, včetně využití nalezených bodů pro analýzu obličeje. Budou představeny knihovny, které implementují různé metody detekce a vlastní návrh řešení implementace sledování obličeje a výrazu tváře.

První kapitola bude věnována detekci obličeje a budou podrobněji popsány dvě významné metody, které se v současnosti používají (Histogram Orientovaných Gradientů a metoda Viola a Jones). Další kapitola se bude zabývat zájmovými body obličeje, jejich obecnému popisu, algoritmům detekce a datasetům obrázků, které si při učení a testování detektorů využívají. Blíže budou popsány tyto metody: landmark, metoda, kterou popsali Kazemi a Sullivan a postupy založené na AAM a LBF. V poslední teoretické části budou představeny knihovny OpenCV a dlib především z hlediska detekce obličeje a obličejových zájmových bodů, kde bude popsáno, jaké algoritmy pro detekci tyto knihovny používají. V poslední kapitole bude představena vlastní implementace analýzy zájmových bodů a její využití pro ovládání počítače pomocí myši (posun a kliknutí levým tlačítkem). V závěru této části budou popsány výsledky testů rychlosti a přesnosti naimplementované aplikace a také některé problémy tohoto řešení včetně návrhů zlepšení.

2 Detekce obličeje v obraze

Metod pro detekci různých objektů v obraze existuje v dnešní době stále více, výzkum se rozrůstá a jsou nacházeny nové oblasti využití. Algoritmů pro detekci obličeje nalezneme spoustu a různé způsoby přístupu a detekce se hodí pro různé situace. Podle toho můžeme algoritmy rozdělit do několika základních skupin.

- Jedna ze skupin přístupů je založena na typických rysech obličeje. Tyto části jsou většinou neměnné a nejsou unikátní. Jedná se například o oči, nos nebo barvu kůže, která je velice specifická, ale nemůže být použita pro detekci osob všech skupin a je závislá na okolním osvětlení. Výhodou těchto metod je snadná implementace a výsledky jsou také dobré.
- Znalostní metoda využívá přesně daných pravidel a popsaných vlastností obličeje. Popis by měl odpovídat všem obličejům, tudíž algoritmus bude velmi složitý. U těchto metod může být využito strojového učení.
- Dále můžeme vyhledávat pomocí předem připravených obrázků s obličejí, kde daný obraz srovnáváme s uloženými vzorovými obličejí. Tento způsob je opět jednoduchý na implementaci.
- Podobným postupem je vzhledově založená metoda, kdy jsou z databáze různě vypadajících obličejů vytvořeny modely, které se použijí při porovnávání a detekci.

Pro detekci objektů včetně obličeje se ukázala vhodná metoda Histogram Orientovaných Gradientů (Histogram of Oriented Gradients – HOG) [1], jejíž podstatou je výpočet gradientů a sestavení histogramů. Tento deskriptor oproti jiným postupům dosahuje velmi dobrých výsledků. V oblasti detekce obličeje je také zásadní metoda, kterou popsali P. Viola a M. Jones [2]. V dnešní době se jedná o velice často používaný postup dosahující vysoké rychlosti. V následující části práce budou oba výše zmíněné algoritmy popsány.

2.1 Histogram Orientovaných Gradientů

Tato metoda [1] je založena na výpočtu orientovaných gradientů pro každý pixel a následném vytvoření histogramů pro danou oblast. Celý postup je rozdělen do následujících částí. První je příprava vstupního obrazu, následuje samotný výpočet gradientů, rozdělení obrazu na buňky a vytvoření histogramů, normalizace za pomoci překrývajících se bloků (několik buněk), spojení histogramů orientovaných gradientů ze všech bloků a nakonec klasifikace, zda objekt byl nebo nebyl nalezen, pomocí lineárního SVM (Support vector machines).

2.1.1 Příprava vstupního obrazu

Způsobů, jak obraz předzpracovat, bylo vyzkoušeno více, včetně změn barevných modelů na černobílý, RGB nebo LAB. Byly použity i metody pro odstranění vlivu nerovnoměrného osvětlení

a jasů. Avšak žádný postup neměl zásadní vliv na výslednou úspěšnost, což může být zapříčiněno tím, že v jednom z následujících kroků je prováděna normalizace lokálně.

2.1.2 Výpočet gradientů

Výkon detektoru Histogram Orientovaných Gradientů je citlivý na způsob výpočtu gradientů a vyzkoušeno bylo více postupů. Výpočet gradientu pro daný pixel je založen na změně intenzity barvy sousedních pixelů. Výsledný vektor má svou velikost a směr, což následně pomáhá určit hrany v obraze. Vzorce pro výpočet velikosti a úhlu jsou uvedeny níže, kde $|g|$ udává velikost gradientu (vektoru) a α určuje jeho směr (úhel).

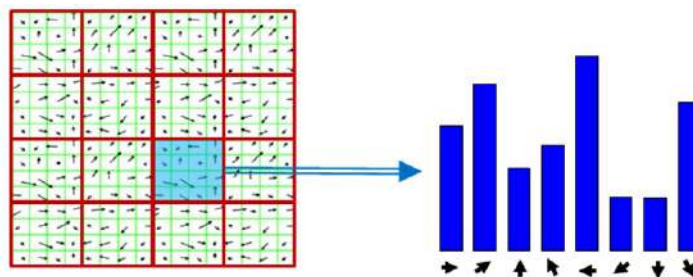
$$|g| = \sqrt{g_x^2 + g_y^2}$$

$$\alpha = \arctan\left(\frac{g_y}{g_x}\right)$$

Pokud je vstupní obraz v RGB modelu, je vektor spočítán pro každou barevnou složku zvlášť a jako výsledný je vybrán ten, který má největší velikost (změna intenzity je největší).

2.1.3 Rozdělení obrazu na buňky a výpočet histogramů

V této fázi je obraz rozdělen na síť buněk o dané velikosti (například 8x8 pixelů), přičemž mohou být čtvercové nebo kruhové, a pro každou buňku je sestaven histogram gradientů. Rozsah kanálů histogramu je 0°- 180°nebo 0°- 360°a může jich být libovolný počet. Pro detekci člověka je vhodné použít tzv. „unsigned“ gradient, tedy rozmezí 0°- 180°, a rozdělení do 9 kanálů (po 20 stupních). Tvorba samotných histogramů vypadá následovně. Vezmeme jednotlivé pixely v buňce, podle směru gradientu určíme správný kanál a vložíme do něj hodnotu, která je určena velikostí gradientu daného pixelu. Pokud je úhel gradientu na rozmezí mezi dvěma kanály, je jeho velikost poměrově rozdělena mezi dané kanály. Tímto postupem vznikne pro každou buňku histogram orientovaných gradientů, což je v podstatě vektor, jehož hodnoty jsou dány součtem velikostí gradientů pro daný úhel. Znázornění tohoto postupu můžeme vidět na obrázku 1.



Obrázek 1: Sestavení histogramu z gradientů jedné buňky [8].

2.1.4 Normalizace bloků

Jelikož velikosti gradientů jsou závislé na osvětlení, které může být odlišné v různých částech obrazu, je potřeba provést lokální normalizaci histogramů, abychom zamezili vlivu nerovnoměrného jasu. Normalizace je prováděna přes více buněk, které společně tvoří jeden blok. Pro každý blok tedy máme několik histogramů (podle toho, kolik buněk blok tvoří), které spojíme, aby vytvořily jeden větší vektor, který je následně znormalizován. Každý blok je normalizován zvlášť, ale jednotlivé bloky se vzájemně překrývají, což by se mohlo zdát zbytečné normalizovat histogramy v buňce vícekrát, ale tento postup vede k výrazně lepšímu výkonu algoritmu.

2.1.5 Výpočet výsledného vektoru

Pro všechny bloky máme získaný jeden normalizovaný vektor. Výsledný HOG deskriptor je dán vektorem, který získáme spojením vektorů ze všech bloků v daném detekčním okně. Tato oblast má velikost 64x128 a okraj okolo osoby je okolo 16 pixelů na všech čtyřech stranách. Zmenšení velikosti okraje, ať toho dosáhneme zmenšením detekčního okna nebo zvětšením osoby, má v obou případech za následek výrazný pokles výkonu.

2.1.6 Klasifikátor

Získaný deskriptor je v posledním kroku předán klasifikátoru, který určí, zda se v obraze objekt vyskytuje nebo ne. Pro klasifikaci byl využit lineární SVM (Support vector machines), což je metoda strojového učení.

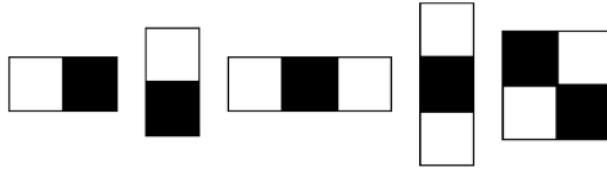
2.2 Metoda Viola a Jones

Významnou metodu pro detekci obličeje popsali P. Viola a M. Jones [2]. Přispěli novými algoritmy a především dosahují vysoké rychlosti. Při výpočtech je používán pouze černobílý obraz, tudíž je potřeba nejdříve provést převod do stupňů šedi. Samotný algoritmus se skládá ze tří hlavních částí. První je jiná reprezentace obrazu zvaná integrální obraz, která umožňuje rychlé vyhodnocování vlastností. Druhá je algoritmus učení založený na AdaBoost a třetí podstatnou částí je použití postupně složitějších klasifikátorů v kaskádové struktuře.

2.2.1 Haarovy příznaky a integrální obraz

Při detekci jsou využívány Haarovy vlnky, konkrétně při průchodu obrazem jsou testovány Haarovy příznaky. Využity jsou tři druhy příznaků, které můžete vidět na obrázku 2. Jsou to „two-rectangle feature“ (hranové), kde je vypočítán rozdíl sumy pixelů v obdélníkových oblastech, které mají stejnou velikosti a sousedí spolu vertikálně nebo horizontálně. Dalším druhem je „three-rectangle feature“ (čárové). Výpočet je proveden odečtením sum vnějších obdélníků od sumy prostředního obdélníku. Poslední je „four-rectangle feature“, u kterého spočítáme rozdíl

sum mezi diagonálními páry obdélníků. Tyto příznaky mohou být vypočítány velice rychle, pokud je obraz reprezentován jako integrální. Proto je pro tuto metodu převod velmi důležitým prvním krokem. Jednotlivé výpočty nejsou prováděny přes celý obraz, ale je vytvořeno podokno, které postupně prochází přes celý integrální obraz a nad ním jsou výpočty prováděny.



Obrázek 2: Příznaky používané při detekci [9].

2.2.2 AdaBoost

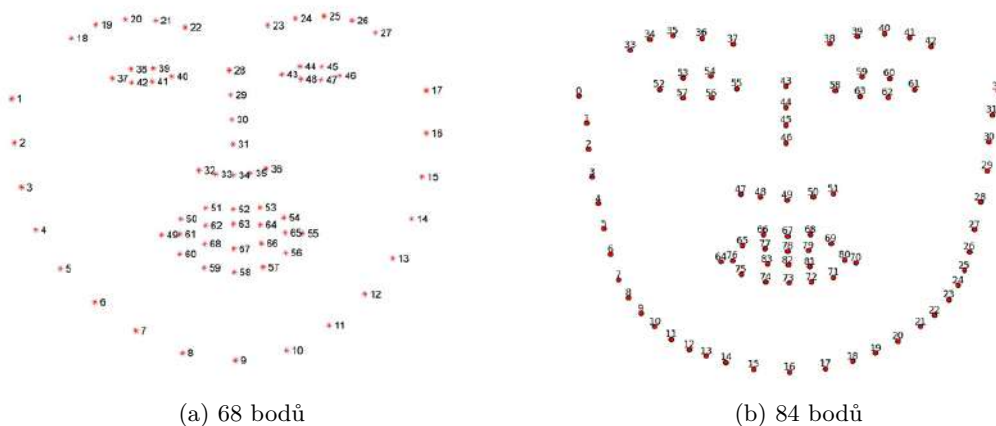
Pro snížení počtu výpočtů je využit klasifikátor AdaBoost (Adaptive Boosting). Jedná se o učící algoritmus, který z množiny klasifikátorů vybere jednoduché (slabé) a kombinuje je do složitějšího (silnějšího) klasifikátoru. Vstupem pro učení jsou pozitivní obrazy (vyskytuje se na nich obličej) a negativní obrazy (nevyskytuje se na nich obličej, obsahují pouze pozadí).

2.2.3 Kaskádová struktura

Metoda kaskády vychází z faktu, že naprostá většina podoken v obraze je negativní, tudíž je snaha vyřadit z výpočtu co nejvíce možných negativních podoken již na začátku za použití jednodušších klasifikátorů. Zatímco pozitivní podokna budou procházet postupně složitějšími klasifikátory a výpočet se tak bude soustředit především na části, které mají větší pravděpodobnost výskytu objektu. Díky tomu se rapidně zvýší rychlost celého výpočtu. Podstatou kaskády je tedy využití nejprve jednoduchých menších klasifikátorů, které jsou schopny vyřadit velké množství negativních podoken a zároveň spolehlivě rozpoznat pozitivní podokna. První (nejjednodušší) klasifikátor v kaskádě určí několik negativních podoken, která jsou rovnou zahozena a dále se s nimi již nepočítá. Naopak pozitivní výsledky jsou předány druhému klasifikátoru, který opět určí a zahodí negativní snímky a pozitivní pošle na přezkoumání třetímu klasifikátoru. Tímto způsobem se pokračuje a čím jsme v kaskádě dál, tím složitější mají klasifikátory úkol. Stupně v kaskádě jsou vytvořeny učením klasifikátorů za pomoci AdaBoost.

3 Zájmové body obličeje

Po detekci obličeje je dalším krokem určení polohy důležitých částí obličeje jako jsou oči, nos, ústa, obočí a další. Jedná se o nalezení polohy tzv. obličejových zájmových bodů. Koordináty těchto zájmových bodů lze dále využít pro sledování obličeje, kdy můžeme například určit pozici či natočení hlavy nebo výraz tváře (otevřená ústa, zavřené či otevřené oči). Metod, které řeší tuto problematiku, již bylo zveřejněno více a stále se pracuje na lepších způsobech. Každý postup je založen na trochu jiném principu a může nalézt různý počet zájmových bodů 3.



Obrázek 3: Příklady počtu zájmových bodů, které mohou být na obličeji detekovány [13][12].

3.1 Metody detekce

3.1.1 Flandmark

Jeden z detektorů důležitých bodů je implementován v knihovně pro jazyk C, která se jmenuje flandmark [11].

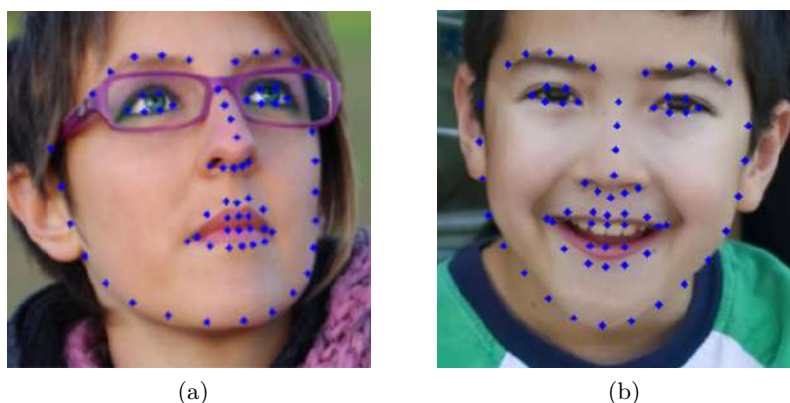


Obrázek 4: Ukázka výsledku detektoru z knihovny flandmark [11].

Detektor používá strukturovaný klasifikátor založený na DPM (Deformable Part Models), což můžeme zjednodušeně popsat tak, že předpokládáme, že objekt je tvořen z částí, nejprve je detekován celý objekt a poté pomocí jeho částí je výsledek vylepšen. Ukázkou výsledku detektoru z knihovny flandmark můžeme vidět na obrázku 4. Červený obdélník znázorňuje ohraničení získané pomocí detektoru obličeje a modrý ohraničuje vstupní data použitá pro detektor. Červené body jsou již vyobrazení samotných zájmových bodů získaných při výpočtu. Jak můžeme vidět, tento detektor zachycuje pouze základní obličejové zájmové body, oproti pozdějším detektorům jich je výrazně méně.

3.1.2 AAM

Další metoda je založena na AAM (Active Appearance Model). Jedná se o statistické modely tvaru a vzhledu, ze kterých může být vygenerovaný specifický objekt dané třídy, například obličeje. V článku "Optimization problems for fast AAM fitting in-the-wild" [4] ukázali, že jako vstup může být použita pouhá intenzita pixelů. Dosahovali podobných výsledků jako jiné metody založené na AAM ovšem bez použití složitých tvarů nebo robustních modelů vlastností. Ukázky takového detektoru jsou na obrázku 5.

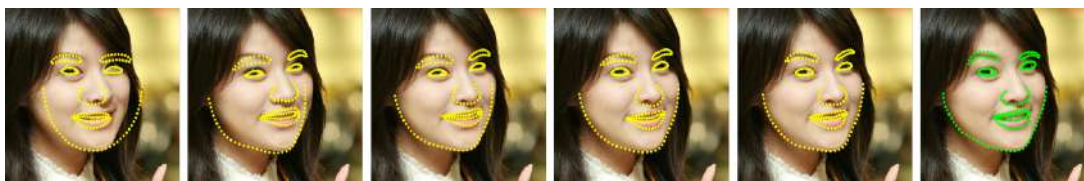


Obrázek 5: Ukázky výsledku detektoru založeného na AAM [4].

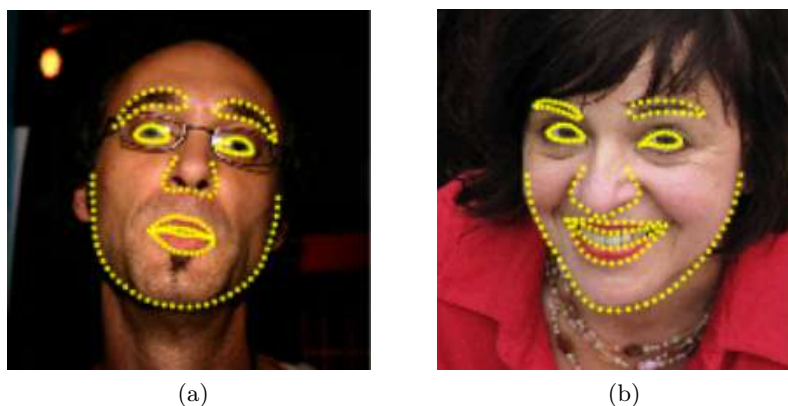
3.1.3 Kazemi a Sullivan

Jiný přístup je řešen v "One millisecond face alignment with an ensemble of regression trees"[3] a metoda zde popsaná získává kvalitní výsledky velice rychle v reálném čase. Algoritmus je založený na významných poznacích získaných z předchozích úspěšných výzkumů týkajících se nalezení důležitých částí obličeje. Podstatou je využití souboru regresních stromů, přičemž obzvláště výkonné je učení samotných regresních funkcí. Jelikož nalezení tvaru a získání vlastností je na sobě vzájemně závislé (pro odhad tvaru potřebujeme spolehlivé vlastnosti a naopak), je použit kaskádový přístup 6. Tvar není odhadován z globálně zjištěných vlastností, ale obraz je přeměněn do normalizovaného systému koordinát podle současného odhadu tvaru, vlastnosti jsou

poté získány z tohoto obrazu a využity pro obnovený odhad tvaru. Tento postup je několikrát zopakován. Ukázky výsledku můžeme vidět na obrázku 7.



Obrázek 6: Postupné zlepšení odhadu polohy zájmových bodů u jednotlivých kroků kaskády za použití regresních stromů [3].



Obrázek 7: Ukázky výsledku detektoru založeného na regresních stromech [3].

3.1.4 LBF

Následující metoda “Face alignment at 3000 fps via regressing local binary features” [5] je také založena na regresi, přesněji regresi LBF (Local Binary Features). Tvar obličeje je opět odhadován kaskádovým způsobem, kdy začneme s prvotním odhadem a v následujících fázích je tvar vylepšován 8.



Obrázek 8: Velikosti lokálních oblastí daných zájmových bodů při odlišných stupních kaskády u přístupu založeném na LBF [5].

Podstatou této práce je tzv. lokální princip pro určení polohy daného zájmového bodu v každé fázi. Nejprve je naučena funkce generující lokální vlastnosti pro všechny zájmové body zvlášť, tyto funkce jsou poté spojeny a použity při učení regrese pro konečný výstup. Díky lokálnímu přístupu je tato metoda velice rychlá a přesností se rovná ostatním postupům. Příklady výsledku jsou na obrázku 9



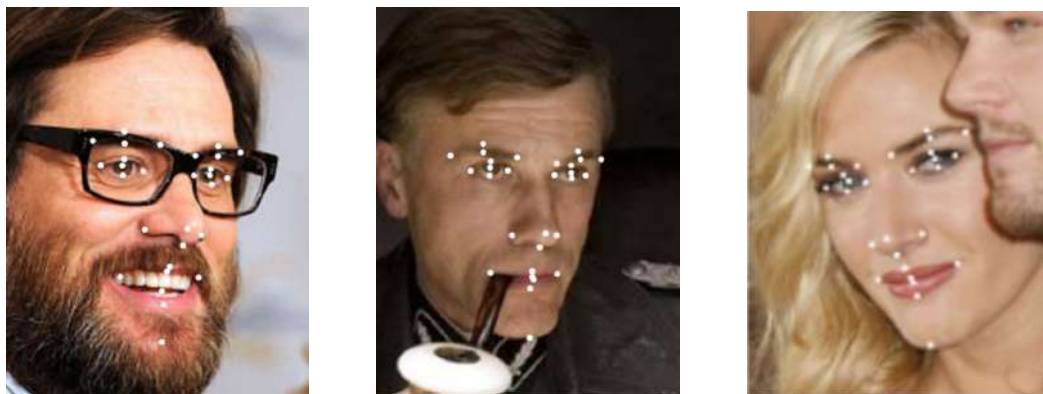
(a)

(b)

Obrázek 9: Ukázky výsledku detektoru založeného na LBF [5].

3.2 Datasetsy

Nedílnou součástí algoritmů pro detekci zájmových bodů obličeje jsou datasety obrázků, na kterých je detektor trénován a testován. Existují soubory obrázků, které byly vytvořeny v průběhu zkoumání detekce zájmových bodů a jsou následně používány i u dalších nových metod, což je vhodné nejen pro vytvoření a testování detektoru, ale také pro porovnání výsledků odlišných detektorů. Lépe si můžeme představit rozdíly v přesnosti různých přístupů, pokud jsou výsledky demonstrovány na stejných datasetech obrázků.



(a)

(b)

(c)

Obrázek 10: Příklady obrázků z LFPW datasetu [6].

3.2.1 LFPW

Labeled Face Parts in the Wild (LFPW) dataset byl poprvé zmíněn v publikaci "Localizing Parts of Faces Using a Consensus of Exemplars"[6]. Autoři vytvořili tento dataset, jelikož chtěli výsledky svého detektoru ukázat na obrázcích náročnějších pro detekci, než obsahovali tehdejší datasety. Potřebovali snímky s různými světelnými podmínkami, s odlišnými pozicemi obličeje a rušivými elementy jako jsou brýle nebo sluchátka. Původní dataset obsahoval webové odkazy na 3000 obrázků, avšak ne všechny odkazy jsou v současné době stále dostupné. Zájmových bodů bylo na každém obraze manuálně označeno 35 a příklady snímků můžeme vidět na obrázku 10.

3.2.2 Helen

Dalším datasetem je tzv. Helen dataset, který byl vytvořen autory článku "Interactive Facial Feature Localization"[7]. Opět se tvůrci snažili poskládat rozmanitější a náročnější soubor obrázků a přidali ještě další podmínku, kterou je vysoké rozlišení. Celkem obsahuje dataset 2000 obrázků k trénování a 330 k testování. Všechny byly získány z webové stránky Flickr a byly na nich označeny zájmové body (větší počet než u LFPW datasetu). Příklady jsou na obrázku 11.



(a)



(b)



(c)

Obrázek 11: Příklady obrázků z Helen datasetu [15].

3.2.3 300-W

300-W dataset byl vytvořen pro účely soutěže, kde byly na obrázcích z tohoto souboru testovány různé metody detekce zájmových bodů obličeje [13]. Jednalo se o vůbec první soutěž tohoto druhu a vítězi se stali J. Yan, Z. Lei, D. Yi, a S. Z. Li - Learn to combine multiple hypotheses for face alignment (Academia) a E. Zhou, H. Fan, Z. Cao, Y. Jiang, a Q. Yin - Facial landmark localization with coarse-to-fine convolutional network cascade (Industry). Podstatou snímků jsou velice různorodá a živelná prostředí, ve kterých byly pořízeny, a obsahují také neobvyklé objekty v různých pozicích. Celkem se jedná o 2x300 obrázků (300 z venkovního prostředí a 300 z vnitřního). Příklady z tohoto datasetu jsou na obrázku 12.



(a)



(b)

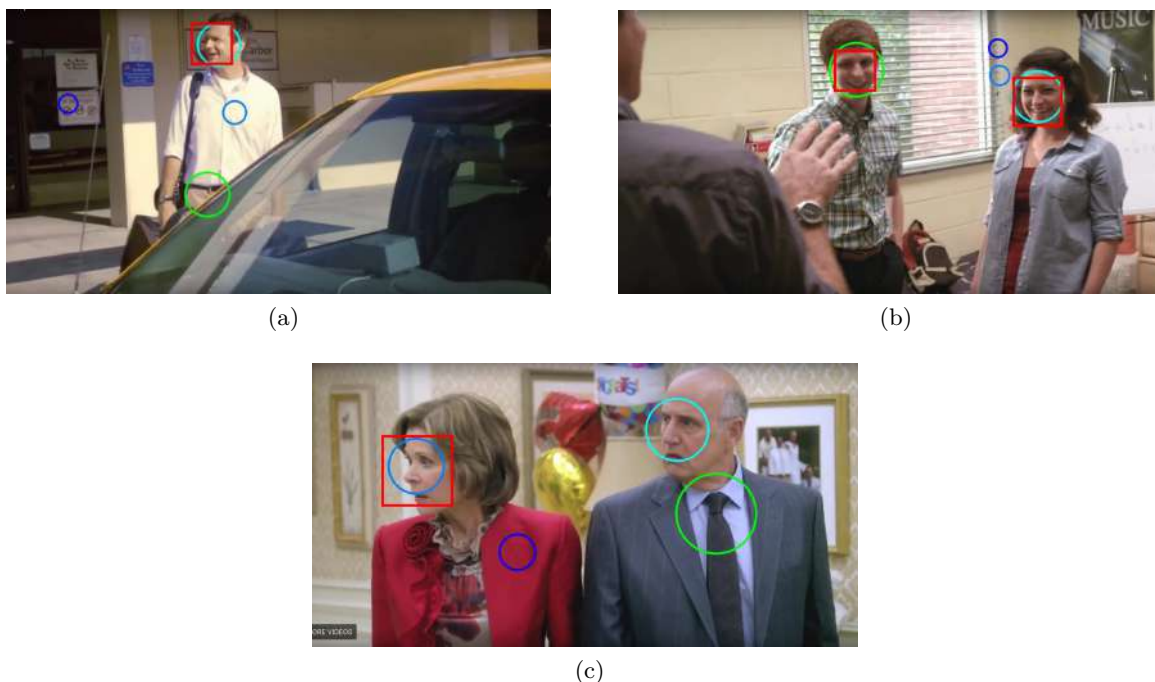
Obrázek 12: Příklady obrázků z 300-W datasetu [13].

4 Knihovny OpenCV a dlib

Pro sledování a analýzu obličeje lze použít funkce z knihoven OpenCV a dlib. Obě poskytují velice užitečné nástroje pro práci s obrazy i mnoho algoritmů strojového učení.

4.1 Detekce objektů a obličeje

Každá z knihoven používá trochu jiný přístup při řešení detekce objektů v obraze. V knihovně OpenCV je metoda založena na postupu, který popsal P. Viola a M. Jones [2], zatímco dlib zvolilo využití deskriptoru HOG [1]. V obou případech můžeme detektor natrénovat na jakýkoli objekt, který si zvolíme. Musíme poskytnout trénovací data a poté použijeme trénování, které je v knihovnách naimplementováno, čímž získáme vlastní detektor. Co se týče detekce obličeje, obě knihovny nabízí již hotový detektor, který může být snadno použit. Pokud bychom měli popsat rozdíl mezi těmito detektory, s použitím knihovny OpenCV získáme mnohem více chybně detekovaných obličejů než u knihovny dlib. Na obrázku 13 modré a zelené kruhy znázorňují detekci od OpenCV a červené čtverce od knihovny dlib. Jak lze vidět, OpenCV detekuje více míst, na kterých žádný obličej není. Obrazy jsou ukázky z videa [10] a rozdíl je tu dobře znatelný.

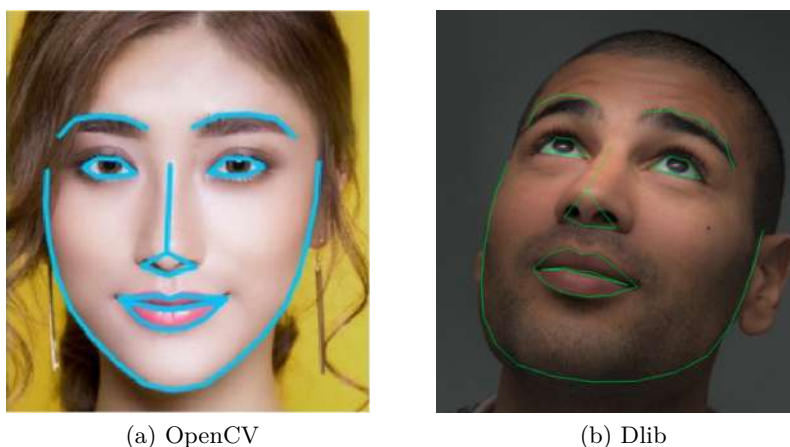


Obrázek 13: Detekce obličeje pomocí knihoven OpenCV (kruhy) a dlib (čtverce) [10].

4.2 Detekce zájmových bodů obličeje

Funkce pro nalezení zájmových bodů obličeje byla k dispozici nejdříve v knihovně dlib. Používá metodu „One millisecond face alignment with an ensemble of regression trees“ [3]. Rychlost a

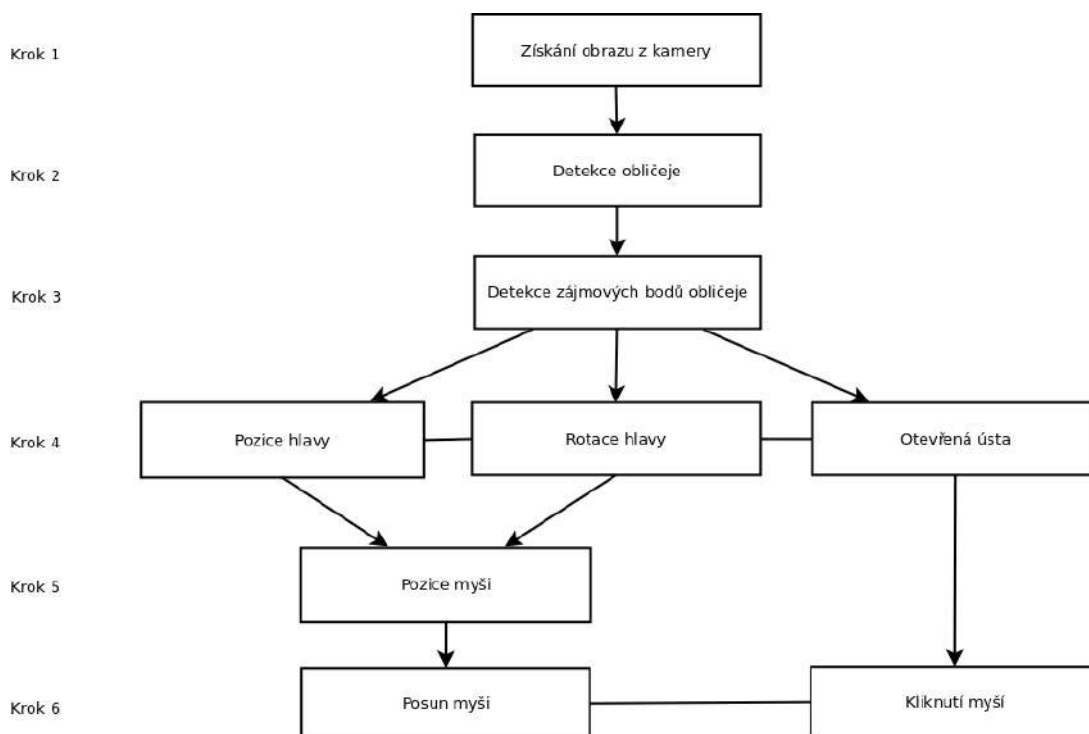
přesnost jejich implementace odpovídá výsledkům, které jsou v článku zmíněny. OpenCV nabízí implementaci založenou na třech různých metodách. Jedna z nich je stejná jako u knihovny dlib [3], další je popsána v článku “Optimization problems for fast AAM fitting in-the-wild” [4]. Třetí způsob je založen na “Face alignment at 3000 fps via regressing local binary features” [5]. Podpora tohoto problému ještě není u OpenCV kompletní, v současné době nabízí pouze jeden natrénovaný model založený na třetí metodě, pokud bychom chtěli použít předchozí dvě, musíme natrénovat detektor vlastní. Jelikož tyto funkce nenabízí OpenCV příliš dlouho, dá se v blízké době očekávat další vylepšení.



Obrázek 14: Detekce zájmových bodů obličeje pomocí knihoven OpenCV a dlib [18][17].

5 Implementace analýzy zájmových bodů obličeje

Cílem implementace vlastního programu bylo analyzovat obraz z webové kamery a zjistit podstatné informace o hlavě (poloha, natočení, otevřená ústa) pomocí detekovaných zájmových bodů obličeje a následné využití získaných informací k základnímu ovládání počítače pomocí myši (pohyb myši je určen natáčením a pohybem hlavy, kliknutí levým tlačítkem provedeme otevřením úst). Abychom mohli obličejové zájmové body analyzovat, musíme nejprve detekovat samotný obličej a dané body. K tomuto účelu jsem použila knihovnu dlib, která nabízí jak detekci obličeje tak detekci obličejových zájmových bodů. Zároveň jsem z knihoven dlib a OpenCV využila několik funkcí, které následnou analýzu usnadní. Navrhované řešení není optimální, jedná se pouze o základní implementaci, kde narazíme na spoustu problémů a návrhů jejich zlepšení. Jak konkrétně je řešení přesné či rychlé je popsáno v části 5.2, která se zabývá testováním aplikace.



Obrázek 15: Vyobrazení jednotlivých kroků implementace. V prvních třech krocích postupně získáme obraz z kamery, detekujeme obličej a zájmové body. Ve čtvrtém kroku provedeme analýzu získaných koordinát zájmových bodů a nalezneme tak polohu a natočení hlavy. Také zjistíme, zda jsou na obličeji otevřená ústa. Z vypočítané polohy a rotace určíme v pátém kroku novou pozici pro myš. V předchozích krocích jsme získali potřebné informace pro ovládání myši, tudíž v šestém a zároveň posledním kroku provedeme samotný posun myši na novou pozici a kliknutí, pokud byla detekována otevřená ústa.

5.1 Postup řešení

Při řešení implementace jsem postupovala po několika krocích znázorněných na obrázku 15. Nejprve je nutné získat obraz z kamery, poté detekovat obličej a následně zájmové body. Pokud máme koordináty důležitých bodů, můžeme s nimi dále pracovat a získat polohu hlavy, natočení či výraz tváře. Analýzou obličeje získáváme informace, které použijeme pro ovládání myši, kdy podle pozice a natočení myši pohneme a pokud detekujeme otevřená ústa, klikneme levým tlačítkem.

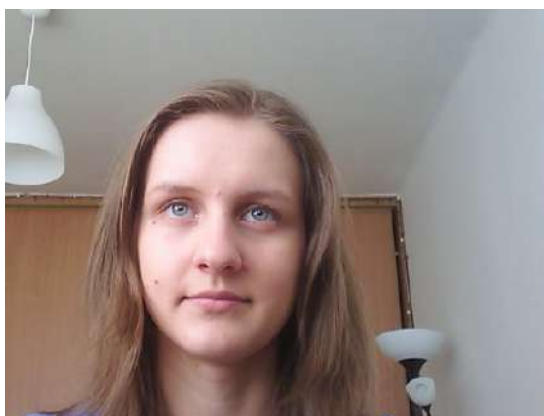
5.1.1 Zachycení obrazu z kamery

Pro získání obrazu z kamery jsem použila funkce z knihovny OpenCV. Následující část kódu 1 nejprve inicializuje spojení s výchozí kamerou, poté načte snímek z kamery a uloží ho do proměnné deklarované jako matice, což je běžná reprezentace obrázků pomocí knihovny OpenCV.

```
// Inicializace spojení s kamerou  
cv::VideoCapture capture(0);  
// Načtení aktuálního snímku  
cv::Mat cameraFrame;  
capture >> cameraFrame;
```

Výpis 1: Načtení obrazu pomocí OpenCV.

Ve vytvořené matici může být uložen například takový snímek, jako je na obrázku 16.



(a)

Obrázek 16: Příklad snímku uloženého v matici.

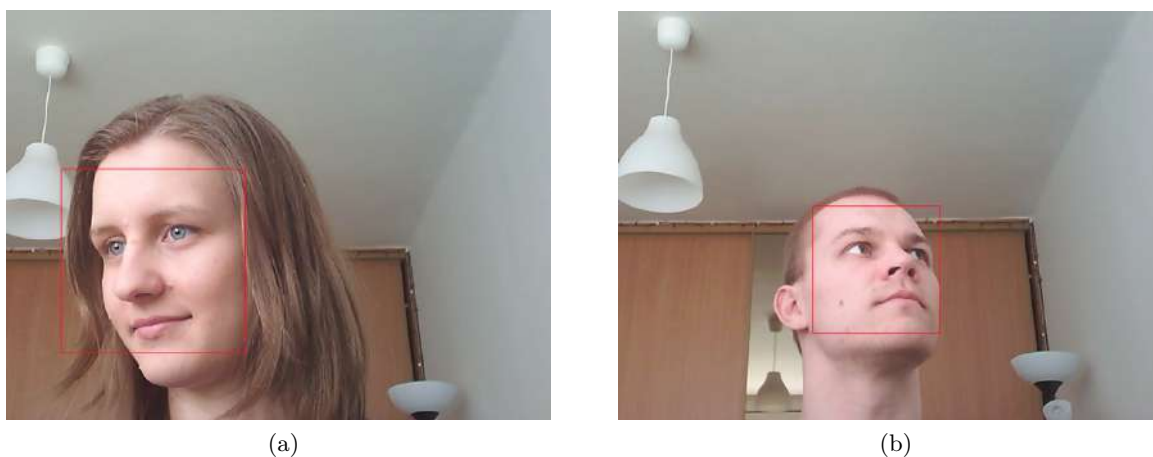
5.1.2 Detekce obličeje

Pokud již máme aktuální snímek z kamery, je potřeba na něm detekovat obličej. Pro detekci jsem použila knihovnu dlib, přesněji jejich předpřipravený detektor. Také je potřeba převést obraz z

OpenCV matice do reprezentace knihovny dlib, abychom jej mohli použít při detekci. Po použití detektoru získáme pole obličejů (ohraničení jejich polohy), které byly detektorem nalezeny 2. Příklad takového ohraničení je vidět na obrázku 17.

```
// Inicializace detektoru
dlib::frontal_face_detector detector = dlib::get_frontal_face_detector();
// Převod obrazu do dlib reprezentace
dlib::cv_image<dlib::bgr_pixel> image(cameraFrame);
// Detekování obličejů
std::vector<dlib::rectangle> faces = detector(image);
```

Výpis 2: Detekce obličeje pomocí dlib.



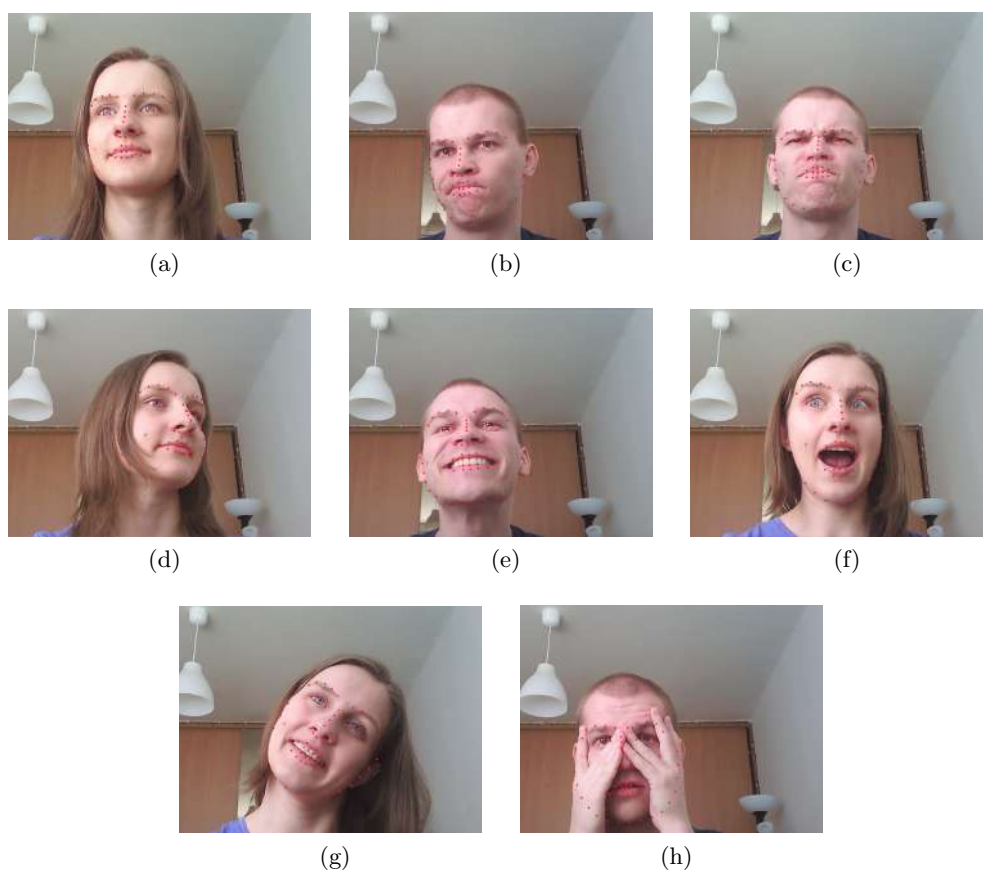
Obrázek 17: Ukázky detekovaného obličeje.

5.1.3 Nalezení zájmových bodů obličeje

V případě, že byl nějaký obličej nalezen, můžeme na něm detekovat zájmové body. Pomocí funkce z knihovny dlib získáme 68 koordinát těchto bodů. Přístupovat k nim můžeme jednoduše pomocí indexů a podle obrázku 3a můžeme odvodit, že koordináty, které ohraničují například levé oko, najdeme pod indexy 42 až 47 (při indexování od nuly) 3. Ukázky detekce zájmových bodů u odlišných pozic a výrazů obličeje jsou na obrázku 18.

```
// Získání polohy zájmových bodů obličeje
dlib::full_object_detection shape = predictor(image, faces[0]);
// Uložení koordinát vnitřního koutku levého oka
cv::Point2d point = cv::Point2d(shape.part(42).x(), shape.part(42).y());
```

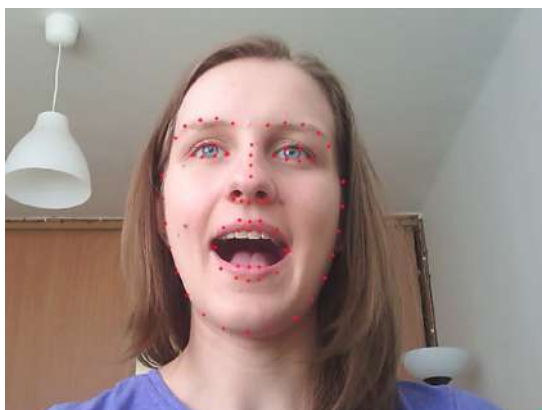
Výpis 3: Detekce zájmových bodů obličeje pomocí dlib.



Obrázek 18: Ukázky detekovaných zájmových bodů při různých pozicích a výrazech obličeje.

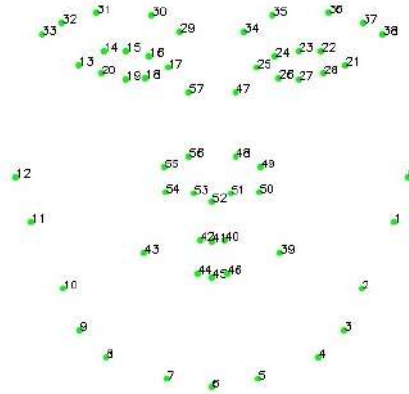
5.1.4 Analýza zájmových bodů

Pomocí koordinát zájmových bodů obličeje můžeme určit mnoho věcí. Především polohu částí obličeje jako jsou oči, obočí, nos a další. Zjistíme také vcelku jednoduše, zda jsou oči nebo ústa otevřená či zavřená.



Obrázek 19: Gesto pro kliknutí myší - otevřená ústa.

Konkrétně otevřená ústa, která v programu používám jako gesto pro kliknutí myší, poznáme tak, že vzdálenost mezi body ohraničující ústa je větší než v případě, že jsou ústa zavřená. Ukázka pozice zájmových bodů při otevřených ústech je na obrázku 19. Pro zjištění polohy a natočení hlavy nám pouze zájmové body nestačí. Musíme mít předem připravený 3D model obličeje (obrázek 20) a odpovídající body porovnáme se získanými koordináty z obrazu, díky čemuž určíme pozici a rotaci obličeje na obrazu.



Obrázek 20: 3D model obličeje pro určení pozice a rotace hlavy [14].

K výpočtu je ještě potřeba znát některé informace o kameře, jako je například ohnisková vzdálenost, avšak u tohoto programu, kde je využita webová kamera, nemá kalibrace kamery na výsledek zásadní vliv a můžeme tudíž použít přibližné hodnoty. Během výpočtu jsou využity základní vzorce pro výpočet posunutého a natočeného bodu ve světovém souřadném systému 1 a přepočtení koordinát ze světového souřadného systému do systému obrazu 2. Ve vzorcích je použito následující označení: výchozí pozice bodu ve světovém souřadném systému (U, V, W) , matice natočení (3×3) R , vektor posunu (3×1) t , pozice natočeného a posunutého bodu (X, Y, Z) , pozice bodu na snímku (x, y) , ohnisková vzdálenost na ose x f_x a ose y f_y , souřadnice optického středu c_x a c_y , měřítko s . [16]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} R & | & t \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2)$$

Celý výpočet posunu a natočení vůči kameře není jednoduchý, jelikož se nejedná o řešení lineární

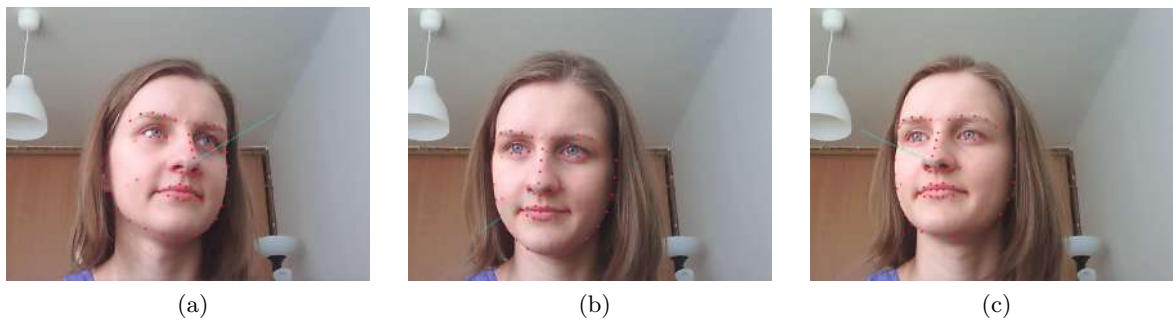
rovnice, a ve své implementaci jsem pro výpočet použila funkci z knihovny OpenCV 4. Příklady hodnot, které můžeme analýzou bodů získat jsou na obrázku 22.

```
// Získání rotace a pozice pomocí detekovaných zájmových bodů
cv::solvePnP(source_3D_points, source_image_points, cam_matrix, dist_coeffs,
             rotation_vec, translation_vec);
```

Výpis 4: OpenCV funkce pro získání pozice a rotace obličeje. Vstupními parametry jsou `source_3D_points` - body 3D modelu obličeje, `source_image_points` - pozice bodů na obrázku, `cam_matrix` - matice s parametry kamery (ohnisková vzdálenost, optický střed), `dist_coeffs` - koeficienty distorze. Výstupními parametry jsou `rotation_vec` - vektor rotace, `translation_vec` - vektor posunu.

5.1.5 Ovládání myši

Na jakou pozici bude myš posunuta nebo zda se provede kliknutí, závisí na předchozí analýze zájmových bodů. Rozhodnutí, zda se má kliknout, je dáno detekováním otevřených úst. Posun myši již tak jednoznačný není. Je mnoho způsobů, jak by mohla být vypočtená poloha a natočení obličeje použita pro určení nové pozice myši. Já jsem využila toho, že koncový bod vektoru, který je veden od obličeje kolmo do prostoru, by měl při pohybu hlavou přibližně odpovídat pohybu myši na obrazovce. Pokud pohneme hlavou doprava, koncový bod vektoru se pohne také doprava a to je směr, jakým očekáváme, že se pohne i kurzor. Vyobrazení vektoru je vidět na obrázku 21. Pro výpočty jsem použila funkci z knihovny OpenCV, která dokáže pomocí zjištěné rotace



Obrázek 21: Ukázky směru vektoru při různém natočení hlavy.

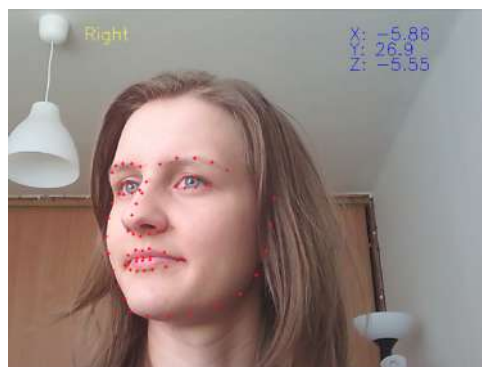
a pozice hlavy převést 3D bod, který zadáme obdobně jako 3D model obličeje, do 2D koordinát snímku 5. Jedná se o opačný proces oproti zjišťování natočení, kde použijeme 3D model a 2D zájmové body a získáme polohu a natočení. Naopak v tomto případě využijeme vypočtenou pozici a rotaci společně s 3D bodem, abychom získali otočenou a posunutou polohu daného bodu na snímku. Pomocí tohoto výpočtu zjistíme pozici vektoru (vedeného ve 3D směrem od obličeje) na 2D obrazu. Následně je potřeba převést koordináty koncového bodu ze souřadného systému snímku do systému monitoru a zjistíme tak novou pozici pro myš.



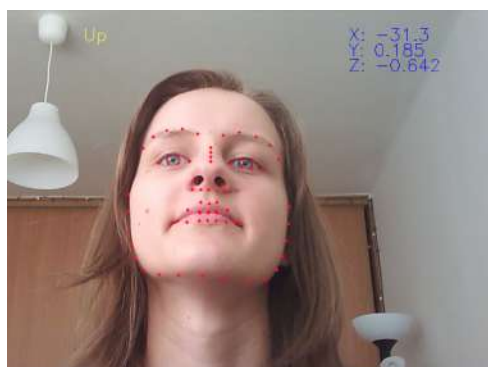
(a) Dopředu



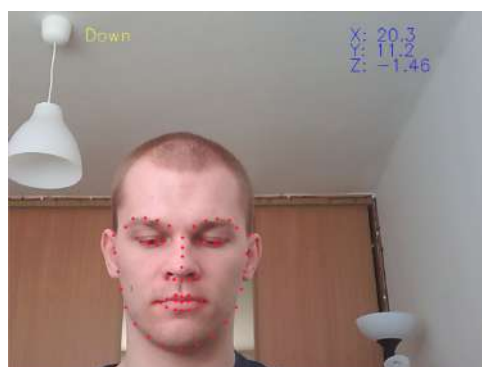
(b) Vlevo



(c) Vpravo



(d) Nahoru



(e) Dolů

Obrázek 22: Ukázky výsledku analýzy zájmových bodů, kde podle úhlů rotace kolem os x , y a z určíme, zda je hlava natočena nahoru, dolů, doprava, doleva nebo je zpřímá. Pokud je podle úhlů zjištěna nějaká rotace, je její směr napsán v levém horním rohu obrázku. Samotné úhly jsou vypsány v pravém horním rohu a můžeme si všimnout, že u obrázku 22a, kdy je pohled směřován dopředu se hodnoty úhlů pohybují kolem nuly. Zatímco u ostatních snímků se u patřičné osy mění čísla do záporných či kladných hodnot (x - nahoru a dolů, y - doleva a doprava). Pouze osa z se výrazně nemění, jelikož úhly určují naklonění hlavy doprava nebo doleva a to v těchto případech zůstává přibližně stejné.

```
// Získání pozice bodu na snímku
cv::projectPoints(source_3D_points, rotation_vec, translation_vec, cam_matrix,
    dist_coeffs, reprojected_2D_points);
```

Výpis 5: OpenCV funkce pro získání pozice bodu (zadaného ve výchozí 3D pozici) na 2D snímku. Vstupními parametry jsou `source_3D_points` - výchozí 3D body, `rotation_vec` - vektor rotace, `translation_vec` - vektor posunu, `cam_matrix` - matice s parametry kamery (ohnisková vzdálenost, optický střed), `dist_coeffs` - koeficienty distorze. Výstupním parametrem je `reprojected_2D_points` - 2D pozice bodů na snímku.

Samotné provedení kliku a posunu již nemá se zájmovými body nic společného a závisí na systému, na kterém je program spuštěn. Já jsem používala operační systém Linux a použité funkce jsou v ukázce kódu 6. Pro kliknutí myši je nejprve nutné správně nastavit událost, která se má provést a poté zavolat funkci pro její provedení. První událostí je zmáčknutí tlačítka, poté je nutné událost přenastavit na uvolnění tlačítka a zavolat opět provedení události.

```
// Posunutí kurzoru na danou pozici
XWarpPointer(display, None, root_window, 0, 0, 0, 0, x, y);
// Zaslání události k provedení
XSendEvent(display, PointerWindow, True, 0xffff, &event);
```

Výpis 6: Funkce použité pro ovládání myši v operačním systému Linux. Vstupní parametry u funkce pro posunutí kurzoru, které jsou pro naše účely důležité, jsou `display` - display, na kterém chceme myši pohybovat, `root_window` - okno, ve kterém chceme myši pohybovat, `x` a `y` - souřadnice, na které chceme kurzor posunout. U funkce pro zaslání události jsou důležité vstupní parametry `display` - display, na který chceme událost poslat, `PointerWindow` - okno, na kterém chceme, aby se událost provedla, `event` - událost, kterou chceme provést.

5.1.6 Kalibrace

Jak již bylo zmíněno v části 5.1.4, při použití webové kamery nemá kalibrace kamery zásadní vliv na výsledek detekce. Ovšem základní kalibrace výchozích hodnot podstatných pro analýzu obličeje již potřeba je, jelikož poloha a natočení jsou vázané na pozici kamery. Například pokud se díváme přímo před sebe, ale kameru máme umístěnou nad naším obličejem, bude snímek pořízený z kamery vypadat, jako bychom se koukali směrem dolů, i když jsme hlavou nepohnuli. I následnou analýzou obrazu získáme hodnoty, které budou odpovídat natočení obličeje dolů. Takové chování není žádoucí, proto je potřeba si výchozí hodnoty zjistit a uložit, abychom informace zjišťované v následující části při analýze určovali oproti výchozímu stavu. Ukázka rozdílných výsledků při použití kalibrace a bez ní je na obrázku 23.



Obrázek 23: Ukázka rozdílu v úhlech rotace při použití kalibrace a bez ní u přibližně stejného natočení hlavy. Na obrázku 23a, kde jsou úhly počítány s prvotní kalibrací se hodnoty pohybují kolem nuly, jelikož byla tato pozice určena jako výchozí. Naopak na obrázku 23b, kde při výpočtech úhlů nebyla brána v úvahu výchozí pozice, jsou hodnoty od nuly více vzdálené.

5.2 Rychlost a přesnost

V následující části práce budou ukázány výsledky získané při testování aplikace. Nejprve bude zmíněna rychlost analýzy zájmových bodů a ovládání myši. Další část bude věnována přesnosti aplikace, kdy bude testována úspěšnost kliknutí do správného prostoru ohraničeného obdélníkem a na závěr budou ukázány některé nepovedené detekce zájmových bodů obličeje. Aplikace byla testována na dvou počítačích. Jedním je stolní počítač (dále označován jako Počítač 1) s těmito parametry: procesor Intel Core i5-6400, RAM 8 GB, grafická karta GeForce GTX 750 Ti. Druhý počítač je notebook (dále označován jako Počítač 2) s parametry: procesor Intel Core i5-7200U, RAM 8 GB, grafická karta Intel HD Graphics 620 (Kaby Lake GT2).

5.2.1 Rychlost

Měření rychlosti detekce a analýzy obličeje pro následný pohyb a kliknutí myši probíhalo na obou počítačích za třech odlišných podmínek. Nejprve byla měřena rychlost analýzy a pouze pohybu myši, kdy jednou byl pohyb hlavou před kamerou spíše pomalejší a méně častý (tabulka 1) a při druhém měření byl pohyb hlavou rychlejší a různorodý (tabulka 2). Do třetího měření bylo zahrnuto i kliknutí levým tlačítkem myši, pokud byla detekována otevřená ústa (tabulka 3). U všech tří měření je v tabulce uvedena doba, za jakou je zpracováno 1000 snímků, dále doba zpracování jednoho snímku a počet snímků analyzovaných za 1 sekundu.

Výsledky u prvních dvou měření se v zásadě neliší. Jeden snímek je zpracován za 90 ms a obrazů za sekundu projde přibližně 11. Z toho je patrné, že na rychlosti a způsobu pohybu hlavy před kamerou nezáleží, jelikož snímky jsou zpracovávány jednotlivě a není podstatné, jaká pozice a výraz tváře jsou analyzovány. Hodnoty u třetího měření se podstatně neliší, pouze je patrné,

že samotné provedení kliknutí má na rychlost drobný vliv. Snímků za sekundu bylo zpracováno okolo deseti, což o jeden méně, a 1000 snímků trvalo analyzovat přibližně o 10 sekund déle.

Tabulka 1: Rychlost detekce a analýzy obličeje při méně častém pohybu hlavou.

	Méně pohybu hlavou		
	1000 snímků (s)	1 snímek (s)	Snímků za sekundu
Počítač 1	87.30	0.09	11.45
Počítač 2	91.39	0.09	10.94

Tabulka 2: Rychlost detekce a analýzy obličeje při častějším a rychlejším pohybu hlavou.

	Více pohybu hlavou		
	1000 snímků (s)	1 snímek (s)	Snímků za sekundu
Počítač 1	86.16	0.09	11.61
Počítač 2	90.81	0.09	11.01

Tabulka 3: Rychlost analýzy obličeje a klikání myši při detekci otevřených úst.

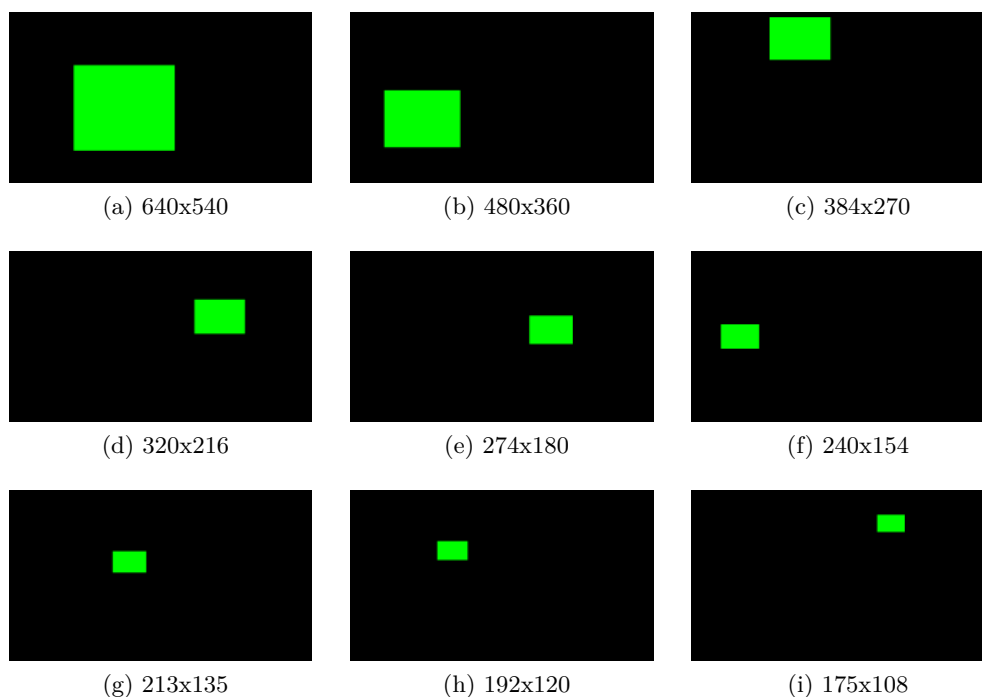
	Pohyb hlavou a klikání myši		
	1000 snímků (s)	1 snímek (s)	Snímků za sekundu
Počítač 1	96.41	0.10	10.37
Počítač 2	100.51	0.10	9.95

5.2.2 Přesnost

Kromě rychlosti implementovaného řešení byla dále zkoumána přesnost pohybu myši a celková schopnost programu kliknout na místo, na které chce uživatel. K testu byly použity stejné počítače jako u měření rychlosti. Oba mají rozlišení obrazovky 1920x1080 pixelů, přičemž Počítač 1 má velikost monitoru 23 palců a Počítač 2 14 palců. Průběh experimentu byl následující. Osoba před počítačem měla za úkol kliknout do zeleného obdélníku na obrazovce za použití pohybu hlavy a otevření úst k ovládní kurzoru. Obdélníky se objevovaly na náhodných pozicích a vygeneroval se nový pokaždé, když uživatel kliknul. Následně bylo do výsledků měření zaznamenáno, zda bylo provedeno kliknutí v daném obdélníku nebo ne.

Velikost obdélníků se postupně v průběhu měření zmenšovala. Prvotní velikost byla 640x540, což odpovídá velikosti jednoho obdélníku, pokud obrazovku rozdělíme na 2x3 obdélníky. Po 30 vygenerovaných obdélnících byla velikost změněna. Obrazovka byla pomyslně rozdělena na 3x4 obdélníků (v každém sloupci i řádku o jeden více), čímž se velikost nové sady obdélníků zmenšila na 480x360. Tímto způsobem se pokračovalo až do velikosti 175x108 pixelů (10x11 obdélníků). Postupné zmenšování obdélníků a jejich velikost oproti obrazovce jsou znázorněny na obrázku 24.

Během testu bylo měřeno, kolik kliknutí bylo provedeno ve správném prostoru a také doba trvání celého testu. V tabulce 4 jsou zaznamenány výsledné hodnoty. Pro každou velikost obdélníků je zapsána úspěšnost kliknutí, rychlost celého testu a průměrná rychlost, která byla potřeba pro posunutí myši do dané oblasti. Záznamy jsou ukázány pro dvě osoby, z čehož jedna z nich jsem já (Osoba 1) a druhá (Osoba 2) je dobrovolník, který způsob implementace programu nezná, tudíž může test provádět méně zaujatě.



Obrázek 24: Velikosti obdélníků použitých při testování přesnosti.

Z výsledků je patrné, že z počátku dosahovala Osoba 2 horších výsledků než Osoba 1, jelikož zjišťovala, jak se program chová. Ovšem čím déle v testu pokračovala, tím lépe se jí odhadoval pohyb myši a dosahovala vyšší nebo stejné úspěšnosti, i když se obdélníky zmenšovaly. Také je důležité zmínit, že Osoba 2 začala s testováním na Počítači 2, což se odráží i ve výsledcích, které jsou znatelně horší než u Počítače 1. Například u velikosti 213x135 bylo dosaženo úspěšnosti 100% a 73.33%, rozdíl je tedy 26.67% a čas méně povedeného testu je dokonce horší o 80 sekund (přibližně o 2.5 sekundy na jeden obdélník). Lepších výsledků tedy bylo u Osoby 2 dosaženo při použití Počítače 1 a úspěšnost při těchto testech neklesla pod 90% ani u nejmenších obdélníků. V případě Osoby 1 bylo u prvních čtyř velikostí dosaženo pokaždé úspěšnosti 100%, ovšem rozdíl byl v době prováděného testu. Průměrný čas na jedno kliknutí se z 2 sekund u prvního měření zvýšil na 5 sekund u čtvrtého měření. Stejně jako u Osoby 2 je vidět rozdíl v úspěšnosti na odlišných počítačích, kdy lepších výsledků bylo dosaženo u Počítače 2, který byl u Osoby 2 druhým testovaným počítačem, a úspěšnost neklesla pod 93.33%. Naopak u Počítače 1, který byl první, je nejnižší úspěšnost 83.33%. Pokud bychom se zaměřili na 100% přesnost, té bylo

dosaženo alespoň jednou osobou u všech velikostí kromě poslední, tam je nejvyšší úspěšnost 93.33%.

Z celého experimentu bylo zjištěno, že velkým problémem je, že myš se tzv. klepe a nezůstane v klidu na jednom místě, tudíž je obtížné se trefit kliknutím do menší oblasti. Především je tato vlastnost patrná, když se kurzor nachází v rozích obrazovky. Okolo středu většinou nebylo obtížné kliknout do správného místa. Kurzor se také někdy překvapivě pohnul, jakmile se uživatel snažil kliknout a otevřel ústa. S gestem otevřených úst je spojena další příčina nižší úspěšnosti, kdy docházelo k nesprávnému detekování otevřených úst a provedlo se kliknutí, aniž by to uživatel chtěl. Na některá místa na obrazovce občas nebylo ani možné myš posunout, i když se to nestávalo často, a jednalo se především o rohové oblasti.

5.2.3 Chybné detekce obličeje a zájmových bodů

Nejen analýza zájmových bodů není úplně přesná, ale samotné detekování obličeje a zájmových bodů má své rezervy. Obličej například není detekován především při natočení z profilu a odhad pozic zájmových bodů není vždy takový, jaký bychom čekali. Ukázky nepovedených detekcí jsou na obrázku 25.



Obrázek 25: Ukázky nepovedené detekce obličeje a zájmových bodů. Na obrázcích 25a a 25b nejsou vyobrazeny zájmové body, jelikož na takových snímcích nebyl detekován žádný obličej. Obrázky 25c a 25d ukazují chybný odhad pozic zájmových bodů.

Tabulka 4: Výsledky měření přesnosti pohybu a kliknutí myši. Řádky P 1 a P 2 znamenají měření na Počítači 1 a Počítači 2. Ve sloupci Úsp. [%] je úspěšnost správného kliknutí v procentech, sloupec Rych. 1 [s] znamená celkovou dobu testu a Rych. 2 [s] je průměrný čas potřebný pro kliknutí na jeden obdélník. Tabulka je rozdělena po 2 řádcích na části a každá část patří nadepsané velikosti obdélníků.

	Osoba 1			Osoba 2		
	Úsp. [%]	Rych. 1 [s]	Rych. 2 [s]	Úsp. [%]	Rych. 1 [s]	Rych. 2 [s]
Velikost obdélníků: 640x540						
P 1	100.00%	64.40	2.15	100.00%	57.24	1.91
P 2	100.00%	79.13	2.64	93.34%	67.97	2.27
Velikost obdélníků: 480x360						
P 1	100.00%	77.79	2.59	100.00%	69.82	2.33
P 2	100.00%	92.32	3.08	100.00%	112.80	3.76
Velikost obdélníků: 384x270						
P 1	100.00%	110.04	3.67	96.67%	99.76	3.33
P 2	100.00%	109.92	3.66	90.91%	156.90	5.23
Velikost obdélníků: 320x216						
P 1	100.00%	121.94	4.06	90.00%	119.32	3.98
P 2	100.00%	144.48	4.82	63.64%	190.20	6.34
Velikost obdélníků: 274x180						
P 1	100.00%	136.57	4.55	90%	144.08	4.80
P 2	96.67%	102.69	3.42	76.67%	143.25	4.78
Velikost obdélníků: 240x154						
P 1	90.00%	136.85	4.56	100.00%	137.11	4.57
P 2	96.67%	115.07	3.84	76.67%	199.46	6.65
Velikost obdélníků: 213x135						
P 1	86.67%	139.09	4.64	100.00%	121.70	4.06
P 2	93.33%	240.63	8.02	73.33%	201.74	6.72
Velikost obdélníků: 192x120						
P 1	83.33%	171.37	5.71	93.33%	227.85	7.60
P 2	100.00%	159.06	5.30	83.33%	171.75	5.73
Velikost obdélníků: 175x108						
P 1	93.33%	207.30	6.91	93.33%	179.35	5.98
P 2	93.33%	154.73	5.16	80.00%	169.28	5.64

5.3 Problémy a návrhy zlepšení

Již v předchozích kapitolách bylo zmíněno několik problémů, které mohou zhoršit výsledek detekce a analýzy. Jeden takový se týká samotné detekce zájmových bodů obličeje, kdy při vykreslování odhadované pozice se body mírně pohybují, přestože se obličej na snímku nehýbe. Následkem je například takový nedostatek, že i bez pohybu hlavou se myš na obrazovce tzv. klepe a zhoršuje se tak přesnost. Zlepšením by mohlo být použití pole pro detekované body a počítat průměr z několika snímků, aby se hodnoty příliš nelišily. Před detekcí zájmových bodů je nutné nejdříve detekovat obličej, kde je také prostor k zlepšení a mohl by být například natrénován vlastní detektor. Dalšího zlepšení bychom mohli dosáhnout použitím hodnot pro kalibraci kamery, které jsou přesné pro danou kameru a ne pouze přibližné. Pro vylepšení rychlosti by mohlo být využito vícevláknové řešení, jelikož u této implementace je zatíženo jedno jádro procesoru a ostatní využita nejsou.

6 Závěr

Cílem práce bylo seznámit se s postupy a metodami pro analýzu obličeje z obrazu a použít získané informace k vlastní implementaci sledování pohybu obličeje a změny výrazu tváře, jako je detekce otevřených úst či natočení hlavy.

V první kapitole jsme se seznámili s algoritmy pro detekci obličeje. Popsán byl deskriptor Histogram Orientovaných Gradientů, který je založen na výpočtu orientovaných gradientů pro každý pixel obrazu a sestavení histogramu, a metoda Viola a Jones, kde jsou pro detekci využívány Haarovy příznaky a kaskádová struktura klasifikátorů. Další popsané metody se zabývaly detekcí zájmových bodů obličeje. Jednalo se o landmark, kde je použit klasifikátor založený na DPM, další postup byl založen na AAM, Kazemi a Sullivan představili metodu založenou na regresních stromech a poslední přístup byl založen na LBF, kde je podstatou lokální princip. Představeny byly také LFPW, Helen a 300-W datasey obrázků, které se používají pro trénování a testování detektorů. Dále byly přiblíženy knihovny OpenCV a dlib, které byly následně využity u vlastní implementace analýzy obličeje z obrazu.

U návrhu řešení sledování obličeje byly využity obličejevé zájmové body k analýze obličeje. Výsledky analýzy byly dále využity pro ovládání myši. Pro detekci obličeje a jeho zájmových bodů bylo využito funkcí a detektorů z knihovny dlib a pro následnou analýzu zájmových bodů byly použity i funkce knihovny OpenCV. Pomocí pozic zájmových bodů bylo možné zjistit natočení a polohu hlavy, díky čemuž jsme mohli například určit, zda je obličej na obraze natočen nahoru, dolů, doprava či doleva. Informace získané z analýzy zájmových bodů byly dále použity pro posun kurzoru ve směru natočení hlavy a kliknutí levým tlačítkem myši při detekci otevřených úst. Během experimentů bylo zjištěno, že navrhované řešení umožňuje kliknutí na naprostou většinu míst na monitoru, problémem je, že kurzor se stále mírně pohybuje, tudíž kliknutí do přesně vymezené malé oblasti je obtížnější. Rychlost implementovaného řešení byla pro tento případ dostačující a na případných zlepšeních se dá dále pracovat.

Navázat na tuto práci by bylo možné vyzkoušením nebo dokonce vytvořením různých detektorů zájmových bodů obličeje a zjištěním, jaký by byl pro daný účel nejvhodnější. Další výzkum by se mohl týkat odlišných způsobů výpočtu pozice pro kurzor, aby bylo řešení přesnější. Pro zlepšení rychlosti by mohla být využita vícevláknová implementace.

Literatura

- [1] DALAL, N. a TRIGGS, B. Histograms of Oriented Gradients for Human Detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) [online]. IEEE, 2005, s. 886-893 [cit. 2018-04-22]. DOI: 10.1109/CVPR.2005.177. ISBN 0-7695-2372-2. Dostupné z: <http://ieeexplore.ieee.org/document/1467360/>
- [2] VIOLA, P. a JONES, M. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001 [online]. IEEE Comput. Soc, 2001, I-511-I-518 [cit. 2018-04-22]. DOI: 10.1109/CVPR.2001.990517. ISBN 0-7695-1272-0. Dostupné z: <http://ieeexplore.ieee.org/document/990517/s>
- [3] KAZEMI, Vahid a SULLIVAN, Josephine. One millisecond face alignment with an ensemble of regression trees. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition [online]. IEEE, 2014, 2014, s. 1867-1874 [cit. 2018-04-22]. DOI: 10.1109/CVPR.2014.241. ISBN 978-1-4799-5118-5. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909637>
- [4] TZIMIROPOULOS, Georgios a PANTIC, Maja. Optimization Problems for Fast AAM Fitting in-the-Wild. In: 2013 IEEE International Conference on Computer Vision [online]. IEEE, 2013, 2013, s. 593-600 [cit. 2018-04-22]. DOI: 10.1109/ICCV.2013.79. ISBN 978-1-4799-2840-8. Dostupné z: <http://ieeexplore.ieee.org/document/6751183/>
- [5] REN, Shaoqing, CAO, Xudong, WEI, Yichen a SUN, Jian. Face Alignment at 3000 FPS via Regressing Local Binary Features. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition [online]. IEEE, 2014, 2014, s. 1685-1692 [cit. 2018-04-22]. DOI: 10.1109/CVPR.2014.218. ISBN 978-1-4799-5118-5. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909614>
- [6] BELHUMEUR, Peter N., JACOBS, David W., KRIEGMAN, David J. a KUMAR, Neeraj. Localizing Parts of Faces Using a Consensus of Exemplars. IEEE Transactions on Pattern Analysis and Machine Intelligence [online]. 2013, 35(12), 2930-2940 [cit. 2018-04-22]. DOI: 10.1109/TPAMI.2013.23. ISSN 0162-8828. Dostupné z: <http://ieeexplore.ieee.org/document/6412675/>
- [7] LE, Vuong, BRANDT, Jonathan, LIN, Zhe, BOUDEV, Lubomir a HUANG, Thomas S. Interactive Facial Feature Localization [online]. ECCV, 2012 [cit. 2018-04-22]. Dostupné z: http://www.ifp.illinois.edu/~vuongle2/helen/eccv2012_helen_final.pdf
- [8] A Short introduction to descriptors. Gil's CV blog [online]. [cit. 2018-04-22]. Dostupné z: <https://gilscvblog.com/2013/08/18/a-short-introduction-to-descriptors/>

- [9] ResearchGate [online]. [cit. 2018-04-22]. Dostupné z: https://www.researchgate.net/figure/Basic-group-of-Haar-like-features-in-the-OpenCV-library-includes-these-five-variations__fig3__272640190
- [10] Dlib 18.6 released: Make your own object detector!. Dlib C++ Library [online]. [cit. 2018-04-22]. Dostupné z: <http://blog.dlib.net/2014/02/dlib-186-released-make-your-own-object.html>
- [11] Flandmark [online]. [cit. 2018-04-22]. Dostupné z: <http://cmp.felk.cvut.cz/~uricamic/flandmark/index.php#structclass>
- [12] 1st 3D Face Tracking in-the-wild Competition [online]. [cit. 2018-04-22]. Dostupné z: <https://ibug.doc.ic.ac.uk/resources/1st-3d-face-tracking-wild-competition/>
- [13] 300 Faces In-the-Wild Challenge (300-W), ICCV 2013 [online]. [cit. 2018-04-22]. Dostupné z: <https://ibug.doc.ic.ac.uk/resources/300-W/>
- [14] Head-pose-estimation. GitHub [online]. [cit. 2018-04-22]. Dostupné z: <https://github.com/lincolnhard/head-pose-estimation>
- [15] Helen dataset [online]. [cit. 2018-04-22]. Dostupné z: <http://www.ifp.illinois.edu/~vuongle2/helen/>
- [16] Head Pose Estimation using OpenCV and Dlib. Learn OpenCV [online]. [cit. 2018-04-22]. Dostupné z: <https://www.learnopencv.com/head-pose-estimation-using-opencv-and-dlib/>
- [17] Real-Time Face Pose Estimation. Dlib C++ Library [online]. [cit. 2018-04-22]. Dostupné z: <http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>
- [18] Facemark : Facial Landmark Detection using OpenCV. Learn OpenCV [online]. [cit. 2018-04-22]. Dostupné z: <https://www.learnopencv.com/facemark-facial-landmark-detection-using-opencv/>

Seznam příloh

A	Zdrojové kódy aplikace na DVD (Příloha na DVD)	42
---	--	----

A Zdrojové kódy aplikace na DVD

Adresářová struktura souborů na DVD je následující.

```
| readme.txt
├─ aplikace
│   └─ Predictors
│       └─ shape_predictor_68_face_landmarks.dat
│   └─ CMakeLists.txt
│   └─ GestureCalibration.cpp
│   └─ GestureCalibration.h
│   └─ LandMarkDetector.cpp
│   └─ LandMarkDetector.h
│   └─ main.cpp
│   └─ PoseDetector.cpp
│   └─ PoseDetector.h
```

V souboru `readme.txt` jsou uvedeny informace potřebné pro spuštění programu. Složka aplikace obsahuje soubory se zdrojovými kódy (soubory s příponami `.h` a `.cpp`), soubor `CMakeLists.txt` pro sestavení programu pomocí `cmake` a složku `Predictors`, ve které je soubor `shape_predictor_68_face_landmarks.dat` potřebný pro detekování zájmových bodů.