

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2025

Petr Dvořáček



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## SLEDOVÁNÍ POSTAVY

FIGURE DETECTION

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

**Petr Dvořáček**

### VEDOUCÍ PRÁCE

SUPERVISOR

**Ing. Miloslav Richter, Ph.D.**

**BRNO 2025**

# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Petr Dvořáček

**ID:** 240333

**Ročník:** 3

**Akademický rok:** 2024/25

**NÁZEV TÉMATU:**

## Sledování postavy

### POKYNY PRO VYPRACOVÁNÍ:

Vytvořte program, který bude sledovat cvičící postavu a zobrazovat odchylky od definovaného správného provedení.

- 1) Seznamte se s metodami vyhodnocení obrazu vhodnými pro řešení dané úlohy.
- 2) Dále použijte k řešení metody využívající obrysu postavy. Definujte sledované charakteristiky. Stanovte podmínky pro správnou práci aplikace. Vytvořte databázi vzorových a testovacích snímků a videí pro testování algoritmů.
- 3) Navrhněte a realizujte algoritmy pro sledování obrysů pohybujících se objektů. Navrhněte archivaci změřených průběhů. Navrhněte a realizujte metody pro stanovení a popis odchylek mezi průběhy. Stanovte velikost detekovaných odchylek a navrhněte vhodná řešení pro zobrazení vad postoje a stanovení doporučení pro korekci vad postoje postavy.
- 4) Navržené řešení realizujte a zhodnoťte kvalitu dosažených výsledků.

### DOPORUČENÁ LITERATURA:

Gonzalez R.C., Woods R.E.: Digital Image Processing, 4th edition, Pearson, 2017, ISBN 978-0133356724

**Termín zadání:** 10.2.2025

**Termín odevzdání:** 28.5.2025

**Vedoucí práce:** Ing. Miloslav Richter, Ph.D.

**Ing. Miroslav Jirgl, Ph.D.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Tato práce pojednává o návrhu aplikace pro sledování postavy v obraze a srovnání její levé a pravé strany podle středové osy. Tato aplikace může být použita například pro porovnání rozsahu hybnosti končetin. Obsahem počátku práce jsou použité metody pro nalezení postavy v obraze. Následně jsou probírány nutné úpravy obrazu pro co nejlepší vyhledávání a zhodnoceny chybné výsledky. Také podmínky nutné pro co nejlepší běh a výsledek programu a jsou tu vysvětleny problémy nastávající při nedodržení těchto podmínek. Poslední část dokumentu je věnována implementaci navržené aplikace pro vyhodnocení video záznamu. V závěru práce jsou diskutovány dosažené výsledky a možné budoucí inovace programu pro zlepšení vyhodnocení.

## **Klíčová slova**

Strojové vidění, detekce postavy, hybnost končetin, metoda momentů, Hu momenty, obrazové momenty, diagnostika hybnosti, MATLAB, zpracování obrazu, RGB detekce kůže, detekce kůže pomocí poměru RGB, odečítání pozadí,

## **Abstract**

This work discusses the design of an application for tracking a figure in an image and comparing its left and right sides based on the central axis. This application can be used, for example, to compare the range of motion of limbs. The initial part of the work covers the methods used for detecting the figure in the image. Subsequently, it examines the necessary image adjustments to optimize detection and evaluates potential inaccuracies in the results. Also, conditions required for the best performance and results of the program, as well as explaining the issues that arise when these conditions are not met. The last section of this document focuses on implementation of devised application for video processing. The conclusion discusses achieved results and potential future innovations to enhance the program's evaluation capabilities.

## **Keywords**

Machine vision, figure detection, limb momentum, method of moments, Hu moments, image moments, momentum diagnostics, MATLAB, image processing, RGB skin detection, RGB ratio skin detection, image subtraction

## **Bibliografická citace**

DVOŘÁČEK, Petr. *Sledování postavy*. Bakalářská práce. Miloslav RICHTER (vedoucí práce). Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2025.

# Prohlášení autora o původnosti díla

**Jméno a příjmení studenta:** *Petr Dvořáček*

**VUT ID studenta:** *240333*

**Typ práce:** *Bakalářská práce*

**Akademický rok:** *2024/25*

**Téma závěrečné práce:** *Sledování postavy*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 26.5.2025

-----  
podpis autora

## **Poděkování**

Chtěl bych poděkovat vedoucímu bakalářské práce panu Ing. Miloslavu Richterovi, PhD. za vedení práce, vstřícnost a hodnotné rady. Také bych rád poděkoval rodičům, Petrovi a Jitce Kohoutkovým za gramatickou kontrolu práce a pomoc při jejím výtisku. Na závěr bych rád poděkoval svojí přítelkyni za pomoc, kontrolu a rady při formátové úpravě textu práce.

V Brně dne: 26.5.2025

-----

podpis autora

# Obsah

<b>SEZNAM OBRÁZKŮ .....</b>	<b>9</b>
<b>ÚVOD .....</b>	<b>10</b>
<b>1. METODA MOMENTŮ .....</b>	<b>11</b>
1.1 VÝPOČET MOMENTŮ:.....	12
1.2 TVORBA BSP STROMU: .....	13
1.3 SLEDOVÁNÍ POMOCÍ MOMENTŮ .....	14
1.4 POUŽITÍ METODY MOMENTŮ .....	14
<b>2. IMPLEMENTACE V PROSTŘEDÍ MATLAB.....</b>	<b>16</b>
2.1 OBRAZ V MATLABU .....	16
2.2 STANDARTNÍ OPERACE .....	16
<b>3. RGB RATIO DETEKCE KŮŽE .....</b>	<b>17</b>
3.1 VNÍMÁNÍ KŮŽE LIDSKÝM OKEM.....	17
3.2 EXISTUJÍCÍ MODELY POMĚRŮ .....	17
3.3 KONKRÉTNÍ MODEL ZALOŽENÝ NA POMĚRECH RGB.....	18
3.4 REÁLNÉ POUŽITÍ PŘI SNÍMÁNÍ OBRAZU .....	19
<b>4. VÝVOJ SLEDOVACÍ APLIKACE .....</b>	<b>21</b>
4.1 STŘED POMOCÍ HRAN OBJEKTU .....	21
4.2 IMPLEMENTACE METODY MOMENTŮ .....	22
4.3 URČENÍ STŘEDOVÉ OSY .....	23
4.4 PŘEVŘÁCENÍ DLE STŘEDOVÉ OSY .....	24
4.5 URČENÍ ODCHYLEK .....	25
<b>5. POSTUPNÉ ZDOKONALOVÁNÍ APLIKACE .....</b>	<b>27</b>
5.1 ZPŘESŇOVÁNÍ ODCHYLKY .....	27
5.2 ZLEPŠENÍ VÝRAZNOSTI POSTAVY .....	27
5.3 DETEKCE KŮŽE.....	30
5.4 ODEČÍTÁNÍ POZADÍ.....	31
5.5 VÝPOČET MĚŘÍTKA OBRAZU .....	33
5.6 ODMĚŘOVÁNÍ ODCHYLEK.....	33
5.7 VYKRESLOVÁNÍ VYMĚŘENÝCH ODCHYLEK .....	35
<b>6. PODMÍNKY FUNKČNOSTI.....</b>	<b>38</b>
6.1 KONTRAST S POZADÍM.....	38
6.2 CIZÍ OBJEKTY .....	40
6.3 ÚHEL SNÍMÁNÍ.....	41
<b>7. IMPLEMENTACE APLIKACE PRO SNÍMÁNÍ VIDEO.....</b>	<b>43</b>
7.1 PŘEVOD APLIKACE PRO ZPRACOVÁNÍ VIDEO.....	43
7.2 ÚPRAVY APLIKACE .....	44
<b>8. ZHODNOCENÍ PRÁCE .....</b>	<b>46</b>
8.1 POUŽITÍ VHODNÝCH METOD PRO ZPRACOVÁNÍ .....	46

8.2	DEFINICE SLEDOVANÝCH CHARAKTERISTIK A PODMÍNKY PRO JEJICH SPRÁVNÉ NALEZENÍ.....	46
8.3	NÁVRH ZPRACOVÁNÍ A VYHODNOCENÍ SLEDOVACÍCH ALGORITMŮ A MOŽNOSTI UKLÁDÁNÍ SNÍMKŮ A VIDEOSEKVENCÍ. ....	47
8.4	ZHODNOCENÍ KVALITY VÝSLEDKŮ A ZPRACOVÁNÍ.....	47
<b>9.</b>	<b>ZÁVĚR.....</b>	<b>48</b>
	<b>POUŽITÁ LITERATURA.....</b>	<b>49</b>
	<b>SEZNAM SYMBOLŮ A ZKRATEK .....</b>	<b>51</b>
	<b>PŘÍLOHY .....</b>	<b>52</b>

# SEZNAM OBRÁZKŮ

1.1	Detekce postavy v obraze pomocí optického toku [1].....	11
1.2	Příklady Kvadrických povrchů [2]. .....	12
1.3	Příklad hledání objektu, telefonu, v obraze pomocí momentů .....	14
1.4	Příklad sledování ručičky otáčkoměru v autě pomocí metody momentů [6]. .....	15
2.1	Složení vrstev RGB barevného modelu ve 3D matici obrazu [7].....	16
3.1	Znázornění detekce kůže pomocí tří úrovní poměrů RGB. Dobře detekovatelný obraz [10]. .....	19
3.2	Znázornění detekce kůže pomocí tří úrovní poměrů RGB. Špatně detekovatelný obraz [12]. .....	20
4.1	Vykreslení nalezených hran a středové osy mobilního telefonu v obraze.....	21
4.2	Vykreslení nalezené osy telefonu. Vlevo nejlepší dosažený výsledek. Vpravo obvyklý výsledek. ...	22
4.3	Demonstrační vykreslení možností funkcí „ <i>get_ellipse</i> “ a „ <i>draw_ellipse</i> “ .....	23
4.4	Výpočet kalibrační středové osy postavy. Vlevo kalibrační postoj. Uprostřed cvičící postoj s nakloněnou osou. Vpravo cvičící postoj s kalibrovanou osou. ....	24
4.5	Demonstrace oddělených polovin postavy. Vlevo převrácená polovina. Vpravo nepřevrácená původní polovina.....	25
4.6	Výsledek vzájemného odečtu oddělených polovin .....	26
4.7	Výsledný obraz s vyznačením odchylek polovin postavy .....	26
5.1	Výsledný obraz s vyznačením odchylek polovin postavy .....	27
5.2	Demonstrace úpravy funkcí „ <i>imlocalbrighten</i> “. Vlevo původní obraz. Vpravo upravený obraz. ....	28
5.3	Demonstrace úpravy funkcí „ <i>imflatfield</i> “. Vlevo původní obraz. Vpravo upravený obraz. ....	29
5.4	Demonstrace úpravy funkcí „ <i>imlocalbrighten</i> “. Vlevo upravený obraz preprocessingem. Vpravo neupravený obraz preprocessingem. ....	29
5.5	Znázornění funkce MATLAB aplikace „ <i>Color Thresholder</i> “.....	30
5.6	Postava s detekovanou kůží a převedena do binárního obrazu.....	31
5.7	Detekce postavy s přítomností cizích objektů. ....	32
5.8	Binární obraz postavy po odečtení pozadí, před (vlevo) a po (vpravo) filtraci chyb odečtení obrazců. ....	33
5.9	Vysvětlení řazení a ukládání velikostí odchylek ve 2D matici.....	34
5.10	Znázornění ukládání počátečních a koncových souřadnic odchylek. ....	34
5.11	Vyznačení odchylek postavy červenou barvou. Vpravo přiblíženo pro lepší viditelnost.....	35
5.12	Zobrazení varování příliš velké odchylky při špatném postoji.....	36
5.13	Výsledný obraz s vyznačením a vypsáním odchylek postavy. Vpravo přiblíženo pro lepší čitelnost. ....	37
6.1	Demonstrace špatného kontrastu postavy s pozadím. ....	38
6.2	Demonstrace špatného výběru oblečení postavy. ....	39
6.3	Výpočet momentů a převod do binárního obrazu za přítomnosti cizích objektů na obraze bez použití preprocessingu.....	40
6.4	Výpočet momentů a převod do binárního obrazu za přítomnosti cizích objektů na obraze s použitím preprocessingu.....	41
6.5	Odchýlení osy postavy na základě úhlu snímání .....	42
7.1	Ukázkové snímky při zpracování videa v původním formátu .....	44
7.2	Ukázkové snímky při zpracování videa ve formátu „DUHA“ .....	45

# ÚVOD

Dnešní medicína je velmi pokročilá ve spoustě odvětví a dokáže téměř bezchybně diagnostikovat zdravotní problémy. Nicméně častokrát nepříjemné a zdlouhavé návštěvy zdravotních zařízení spoustu pacientů odradí a v dnešní době moderních technologií hledají nejprve diagnózu na internetu a pokoušejí se zjistit svůj stav sami, než se vydají k lékaři.

Dnešní medicína využívá velké množství moderních technologií, včetně právě strojového vidění a umělé inteligence, nicméně její nastavení a použití může být komplikované, finančně náročné, a ne vždy plně přesné, jelikož umělá inteligence pracuje na základě podobností naučených předloh a neměří tedy dostatečně přesně.

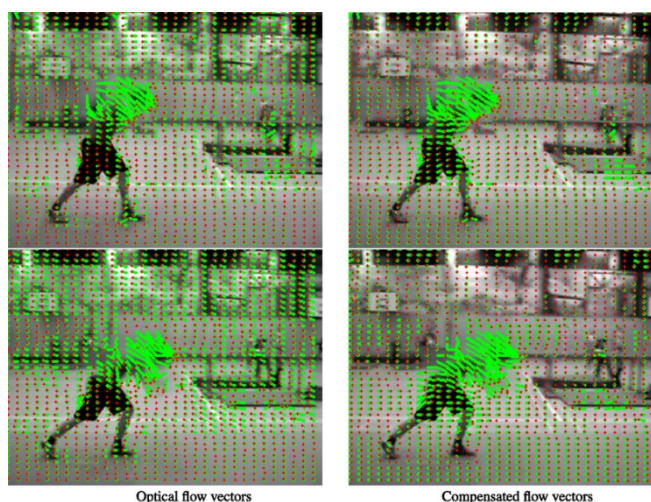
Bakalářská práce pojednává o návrhu aplikace zjištění odchylky levé strany postavy vůči pravé za pomoci strojového zpracování obrazu pixel po pixelu. Tato aplikace může být použita pro laické zdravotní a sportovní účely jako například zjištění rozsahu hybnosti končetin zrcadlově vůči sobě, nebo případně naklonění těla na stranu ze zdravotních či jiných důvodů. Nicméně výsledky aplikace nelze považovat za odborné posudky a případné zjištěné problémy je stále třeba konzultovat s kvalifikovaným lékařem.

Dokument bakalářské práce popisuje postup vývoje snímací aplikace, vysvětluje použité metody a problémy řešené při snímání a zpracování postavy na obraze a videu. Kapitola 4 je zaměřena na samotné hledání postavy v obraze a základní návrh aplikace. V 5. kapitole jsou popisovány konkrétní zdokonalení a inovace vytvořené aplikace. V 6. kapitole jsou následně popsány podmínky a doporučení pro správnou funkčnost aplikace. Předmětem 7. kapitoly je implementace a nutné úpravy navržené aplikace pro vyhodnocení videa. Práce je vypracována převážně za pomoci obecných znalostí programu MATLAB a zpracování obrazu získaných na Vysokém učení technickém v Brně v předmětu BPC-ZVS.

# 1. METODA MOMENTŮ

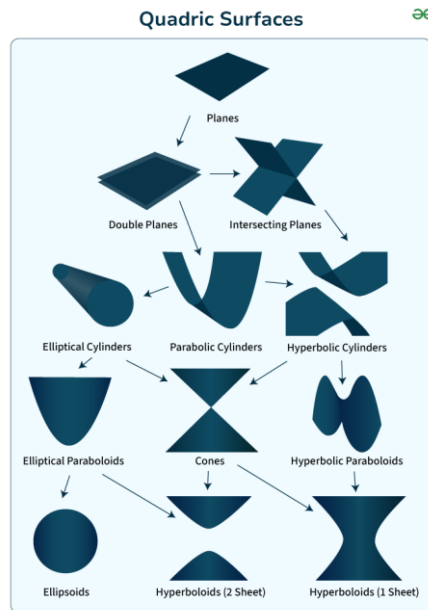
Pro odhadnutí středové osy snímaného objektu, resp. člověka je využito metody momentů. Tato metoda vysvětluje vnímání objektů jako soubor vlastností, díky kterým je možné objekty porovnávat mezi sebou, ale také odvodit jejich tvar, zkosení, zaoblení a jiné. Pomocí elipsy, která částečně opisuje tvar objektu, je možné pak určit středovou osu, natočení směru objektu a také zakřivení objektu. Je možné tuto metodu tedy používat také pro detekci objektů v obraze.

Kvůli nevýhodám detekce pomocí optického toku se začaly používat robustnější kroky založené na funkcích. Metody založené na rozdílnosti objektu a pozadí byly limitovány jednotným pozadím.



Obrázek 1.1 Detekce postavy v obraze pomocí optického toku [1].

Techniky aktivních obrysů, které zvládají sledovat deformovatelné objekty, musejí být ručně nastavovány, tedy nedetekují samy. A postupy založené na tvarových modelech používají kvadriky a super kvadriky, které jsou velmi náročné na výpočty, tedy velmi drahé [3].



Obrázek 1.2 Příklady Kvadrických povrchů [2].

Metoda momentů je použitelná na různé druhy objektů jako je pevný, kloubový, deformovatelný aj. na stacionárním pozadí s pevně daným záběrem kamery. Převedením obrazů do binární oblasti lze získat pohybující se objekt odečítáním pozadí. Nejdříve je pomocí momentů vytvořen bsp-strom, reprezentující aproximace objektu elipsami v počtu  $2^k$  pro každou vrstvu  $k$  bsp-stromu. Hierarchie stromu se poté použije ke sledování objektu snímek po snímku.

## 1.1 Výpočet momentů:

Pro výpočet momentů se používá vážený průměr úrovní jasu pixelů v šedém spektru definovaný vztahem:

$$M_{pq} = \sum_i \sum_j i^p j^q I(i, j). \quad (1.1)$$

Kde  $M_{pq}$  je obecné označení dvoudimenzionálního momentu obrazu,  $i$  a  $j$  indexy řádku a sloupce a  $I$  vstupní obraz. Momenty objektů v šedém spektru je náročné vypočítat. Proto se momenty počítají v binární oblasti, kde objekty jsou tvořeny 1 a pozadí 0, použitím zjednodušeného vzorce:

$$M_{pq} = \sum_{i,j \in Obj} i^p j^q. \quad (1.2)$$

Nulový moment pak určuje celkový počet pixelů v objektu, tj. jeho plochu. První moment po normalizaci udává souřadnice těžiště objektu:

$$\bar{x} = \frac{M_{1,0}}{M_{0,0}}, \quad (1.3)$$

$$\bar{y} = \frac{M_{0,1}}{M_{0,0}} \quad (1.4)$$

Zde jsou  $\bar{x}$  a  $\bar{y}$  normalizované souřadnice. Za pomoci druhého momentu lze pak vypočítat parametry ekvivalentní elipsy objektu:

$$\mu'_{2,0} = \frac{M_{2,0}}{M_{0,0}} - \bar{x}^2, \quad (1.5)$$

$$\mu'_{1,1} = \frac{M_{1,1}}{M_{0,0}} - \bar{x}\bar{y}, \quad (1.6)$$

$$\mu'_{0,2} = \frac{M_{0,2}}{M_{0,0}} - \bar{y}^2 \quad (1.7)$$

Kde  $\mu'_{2,0}$  je parametr ekvivalentní elipsy vypočtený z momentů obrazu. Dosazením do rovnic pak lze získat konkrétní parametry:

$$\theta = \frac{1}{2} \cdot \tan^{-1} \left( \frac{2\mu'_{1,1}}{\mu'_{2,0} - \mu'_{0,2}} \right), \quad (1.8)$$

$$l = \sqrt{8 \left( \mu'_{2,0} + \mu'_{0,2} + \sqrt{4\mu'^2_{1,1} + (\mu'_{2,0} - \mu'_{0,2})^2} \right)}, \quad (1.9)$$

$$w = \sqrt{8 \left( \mu'_{2,0} + \mu'_{0,2} - \sqrt{4\mu'^2_{1,1} + (\mu'_{2,0} - \mu'_{0,2})^2} \right)} \quad (1.10)$$

Theta  $\theta$  určuje orientaci hlavní osy,  $l$  udává délku elipsy a  $w$  šířku elipsy. Orientace objektu je možné získat z orientace hlavní osy, která je udávána od  $-\pi$  po  $+\pi$ .

## 1.2 Tvorba bsp stromu:

Každá vrstva stromu je tvořena  $2^k$  uzly, kde  $k$  je počet vrstev a každý uzel se v další vrstvě dělí na dva další. Počáteční vrstva root (kořenová) má pouze jeden uzel, tedy jednu elipsu. Větším počtem elips je možné lépe popsat tvar objektu. Předpokladem je že v případě sekvence bude každý snímek mít stejný počet vrstev stromu. Každý uzel obsahuje výše zmíněné parametry a také orientaci vedlejší osy  $d$ , která se vypočítá vztahem:

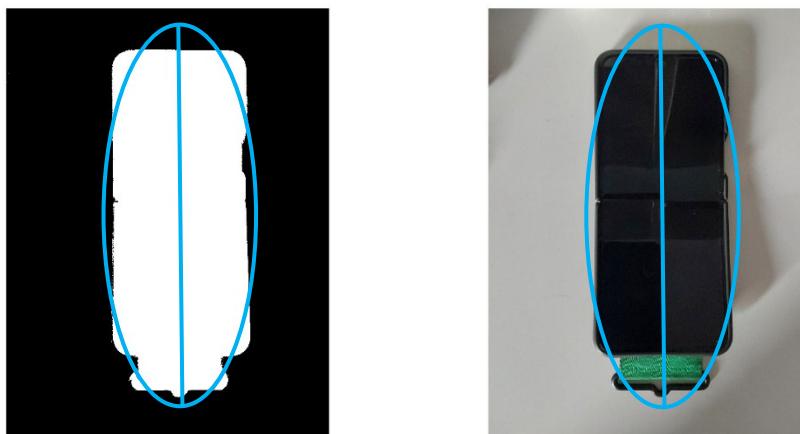
$$d = \left( \cos \left( \theta + \frac{\pi}{2} \right), \sin \left( \theta + \frac{\pi}{2} \right) \right), \quad (1.11)$$

Rozdělení uzlů pak probíhá jako rozdělení objektu podle této osy na dvě poloviny a pro tyto poloviny se opět vypočítají parametry elips. Tento princip se opakuje i pro již rozdělené části objektu. Objekt je pak definován jako  $2^k$  elips na každé úrovni stromu.

### 1.3 Sledování pomocí momentů

Pro sledování je zapotřebí uvažovat strom  $j$  a strom  $j+1$ . V kořenu se parametrizuje hlavní osa první elipsy a promítnou se k ní středy hlavních os následujících dvou elips po rozdělení. Následně se vypočítá vzdálenost těchto středů od počátku parametrizované osy a vypočtou se transformace mezi elipsami ve stromě  $j$  a jejími korespondenty ve stromě  $j+1$ . Tento postup se opakuje pro každé uzly stromů v aktuálním a následujícím snímku [3].

Nicméně pro pouhé nalezení objektu v obraze postačí první iterace stromu, tedy jedna elipsa vytvořená pomocí metody momentů, jako je vidět na obrázku 1.3, kde byl nalezen v obraze tvar mobilního telefonu.



Obrázek 1.3 Příklad hledání objektu, telefonu, v obraze pomocí momentů

### 1.4 Použití metody momentů

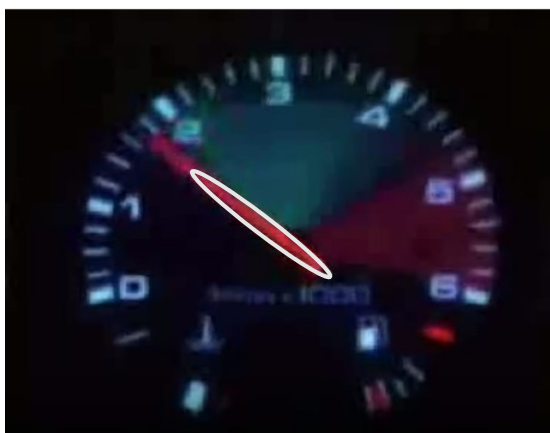
Za pomocí metody momentů je možné hledat objekty v obraze a srovnávat nalezené objekty s jinými, jelikož každý tvar objektu má poměrně jedinečné seskupení všech momentů. Momenty slouží tedy jako identifikační číslo každého tvaru.

Dle momentů lze tedy identifikovat jeden nebo i více objektů na obraze. Pro identifikaci více objektů je samozřejmě potřeba si najít užší oblast hledání každého z objektů tzv. ROI (region of interest), a v té oblasti určovat momenty hledaného objektu. Poté lze určit momenty všech objektů. Dle momentů je možné pak nalezené objekty

srovnávat se zadaným objektem a případně vyhodnotit shodnost zadaného a nalezeného objektu.

Další použití metody momentů může být hledání konkrétního tvaru nebo objektu ve skupině tvarů, resp. objektů. V tomto případě jsou srovnávány momenty objektů spolu navzájem a je hledán objekt s momenty, které se výrazně liší od ostatních, například čtverec ve skupině kruhů [4].

Také lze sledovat objekt v obraze, kdy za pomoci metody momentů lze určovat směr natočení, velikost, nebo třeba jak asymetrický objekt je. Lze tím sledovat kupříkladu buňky pod mikroskopem, jejich pohyb, tvar a natočení, nebo číst rychlost auta na analogovém rychlostním budíku, kde po správné normalizaci lze natočení ručičky přiřadit k reálné rychlosti vozidla [5].



Obrázek 1.4 Příklad sledování ručičky otáčkoměru v autě pomocí metody momentů [6].

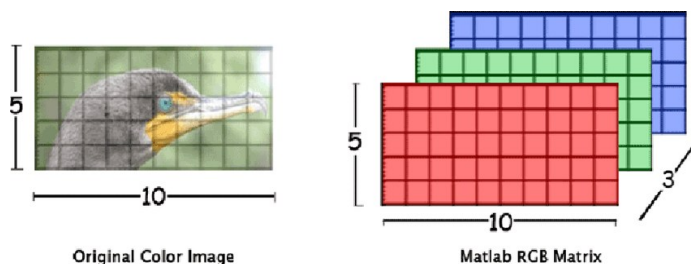
## 2. IMPLEMENTACE V PROSTŘEDÍ MATLAB

Pro návrh sledovací aplikace bylo využito prostředí MATLAB, jelikož jsou v něm implementovány matematické, obrazové a vektorové funkce ve standardních knihovnách a nebylo tedy nutné instalovat dodatečné knihovny funkcí. V těchto základních knihovnách se nachází mnoho funkcí pro běžné operace s vektory a maticemi, ale i morfologické operace právě s maticemi obrazů a operace pro úpravu a práci s obrazem.

Je možné tu tvořit specifické funkce pro potřebu aplikace, které knihovny neposkytují. MATLAB disponuje přehledným seznamem proměnných, kde lze snadno pro potřeby debugingu dohledat načtené nebo vypočtené hodnoty.

### 2.1 Obraz v MATLABU

Obraz je brán jako 3D matice pixelů s hodnotou jasové úrovně obvykle nejvyšší hodnoty 256 ve formátu uint8. Čím větší hodnota, tím jasnější pixel. Třetí rozměr matice obsahuje tři navzájem se překrývající vrstvy, z nichž každá představuje jednu barvu RGB barevného modelu.



Obrázek 2.1 Složení vrstev RGB barevného modelu ve 3D matici obrazu [7].

### 2.2 Standartní operace

Standartní MATLAB knihovny umožňují načítat obrázky pomocí funkcí do maticových proměnných a upravovat je neboli provádět preprocessing, například odstranění šumu, normalizace jasových úrovní nebo převedení do binárního obrazu.

Také morfologické operace jsou ve standardních knihovnách dostupné. Je možné v binárním obraze odstranit místa která nejsou podstatná, nebo naopak zvýraznit tvary, které jsou důležité. Lze i libovolně převést obraz na jiný datový typ, nebo barevný model dle potřeby a provádět různé transformace obrazu [8][9].

### 3. RGB RATIO DETEKCE KŮŽE

Vzhledem k tomu, že kůže není dostatečně kontrastní vůči bílé stěně a v domácích podmínkách se bude jen těžko hledat dostatečně tmavé, či dokonce černé pozadí, je nutné provádět detekci kůže rukou a obličejem. Je mnoho způsobů, které jsou ovšem náročné, zbytečně složité, nebo vyžadují použití dalších nástrojů.

Hledání kožních pixelů na obrázku pomocí RGB Ratio neboli poměru hodnot barev Red, Green a Blue, je poměrně jednoduché jak pro implementaci v kódu, tak i pro následné použití. Tato metoda není sice z nejpřesnějších, a není možné ji srovnávat s vyhledáváním pomocí AI nebo jiných nástrojů, nicméně, pro toto použití by měla být postačující. Velkým jejím problémem může být přítomnost předmětů v obraze majících stejnou, nebo hodně podobnou barvu k barvě kůže.

#### 3.1 Vnímání kůže lidským okem

V lidském oku jsou jednotlivé barvy vnímány světlo citlivými fotoreceptory, tzv. čípky a tyčinkami. Sítnice je tvořena ze šesti vrstev, z nichž poslední je složena z gangliových buněk. Velká část z nich se nazývá malé trpasličí buňky, které jsou aktivovány červenou barvou na okrajích a deaktivovány zelenou barvou ve středu. Menší skupina, tzv. koniocelulární buňky jsou aktivovány modrou barvou na okrajích a deaktivovány červenou nebo zelenou ve středu.

Lidským okem je možno velmi dobře vnímat barvu kůže. Jak můžeme chápat z předešlého odstavce, je to ovšem jen kombinace barev RGB, jakožto aktivace a deaktivace buněk sítnice těmito jednotlivými barvami. Snímání kůže pomocí snímání obrazu by bylo možné tedy vztahy mezi těmito barvami, jako je tomu právě v oční sítnici. Poměr mezi právě RGB barvami je tedy velmi důležitý právě pro snímání lidské kůže zpracováním obrazu.

#### 3.2 Existující modely poměrů

Mezi nejpobulárnější modely patří Kovac et al., Swift, a Saleh. Všechny tři jsou poměrně jednoduché, jak je i vidět na jejich pravidlech srovnání hodnot mezi barvami RGB.

Pravidla Kovac et al.– pixel je kožní pixel pokud:

- 1)  $R > 95$  a  $G > 40$  a  $B > 20$  a
- 2)  $\text{Max}(R, G, B) - \text{Min}(R, G, B) > 15$  a
- 3)  $|R - G| > 15$  a
- 4)  $R > G$  a  $R > B$

Pravidla Swift – pixel není kožní pixel pokud:

- 1)  $B > R$  nebo
- 2)  $G < B$  nebo
- 3)  $G > R$  nebo
- 5)  $B < R/4$  nebo
- 6)  $B > 200$

Pravidla Saleh – pixel je kožní pixel pokud:

- 1)  $R - G > 20$  a
- 2)  $R - G < 80$

Swift a Saleh modely ovšem nejsou schopny snímat tmavší části pleti nebo žlutější odstín pleti. Proto je používán vylepšený model založený na poměru těchto hodnot RGB. Ve srovnání s modely Kovac et al., Swift a Saleh je lepší. Toto tvrzení je založeno částečně i na domněnce, že snímání kůže založené na poměrech je lepší než to založené pouze na prahování hodnot. Dalším pádným argumentem je fakt, že při ne vždy perfektních různorodých podmínkách osvětlení a stínů je použití poměrů hodnot barev méně ovlivňováno než hodnoty samotné.

### 3.3 Konkrétní model založený na poměrech RGB

Na začátku je pro zajímavost definováno šest poměrů pro zjištění oddělitelnosti kůže a tzv. nekůže, tedy pro odhad kvality. Jsou to poměry:  $R/G$ ,  $G/B$ ,  $B/R$ ,  $(R+G)/B$ ,  $(B+R)/G$ ,  $(G+B)/R$ . Pro každých šest jsou vypočteny průměrné hodnoty pro kůži a nekůži, které se srovnávají mezi sebou. Čím větší mezi hodnotami rozdíl je, tím kvalitněji půjde pixely od sebe odlišit.

V další analýze se pak hodnoty tří vybraných poměrů zaokrouhlí na jedno desetinné místo a spočítá se počet pixelů kůže a nekůže pro každý tento zaokrouhlený poměr. Počet kožních pixelů představuje pozitivní detekce kůže (TP – True positive), naopak počet nekožních pixelů negativní detekce (FP – False positive). Jejich poměrem  $TP/FP$  se získá poměr kvality (QR), jenž je požadován co největší, tedy vysoký TP a nízký FP. Nicméně ne vždy je možné dosáhnout co nejvyšší kvality, proto vznikla sada pravidel o třech úrovních jako kompromis založený na poměru kvality QR.

V první úrovni (nejdůvěryhodnější) jsou pravidla založena na QR vyšším než 0.5:

- 1)  $1.3 \leq R/G \leq 2.4$  a
- 2)  $0.8 \leq (G+B)/R \leq 1.4$  a
- 3)  $0.4 \leq B/R \leq 0.6$

Druhá úroveň je založena na poměru kvality větším než 0.2:

- 1)  $1.2 \leq R/G \leq 3.2$  a
- 2)  $0.6 \leq (G+B)/R \leq 1.6$  a
- 3)  $0.2 \leq B/R \leq 0.7$

Ve třetí úrovni (nejméně důvěryhodné) jsou pravidla definována obdobně jako ve druhé s rozdílem některých poměrů, kde je QR menší než 0.2, ale se značným počtem TP.

- 1)  $1.0 \leq R/G \leq 3.4$  a
- 2)  $0.1 \leq (G+B)/R \leq 1.9$  a
- 3)  $0.1 \leq B/R \leq 0.9$

### 3.4 Reálné použití při snímání obrazu

Použití detekce kůže v praxi stále velmi závisí na kvalitě obrazu, světla a složení barev, ačkoliv měl poměr RGB tyto aspekty částečně překonat. V případě dobrého zvolení barev pozadí a popředí ve kterém se nachází i právě kožní pixely, je možné tyto pixely poměrně dobře RGB poměrem a jeho třemi úrovněmi určit.



Fig. 1. Image from UChile dataset (hand.jpg.png).

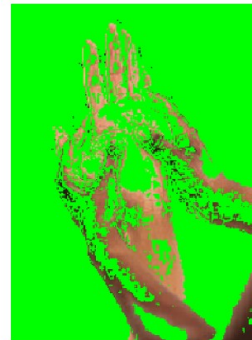


Fig. 2. Skin detection with the first level rules. Observe relatively small number of skin pixels detected.



Fig. 3. Skin detection with the second level rules.



Fig. 4. Skin detection with the third level rules.

Obrázek 3.1 Znázornění detekce kůže pomocí tří úrovní poměrů RGB. Dobře detekovatelný obraz [10].

V dobře vytvořeném obraze není detekce kůže problémem. Nejprve se použije první úroveň detekce, kdy se detekují nejvíce důvěryhodné kožní pixely, viz obrázek 3.1 vpravo nahoře. Nakonec po dokončení všech tří úrovní máme téměř celý obrazec s barvou kůže. Ve třetí úrovni už ovšem dochází k přisuzování i nekožních pixelů, což jsou právě oblasti kolem rukou na obrázku 3.1 vpravo dole.

Obraz ovšem nemusí být takto dobře oddělitelný. O čemž se dá přesvědčit v následujícím obrázku 3.2 použití detekce. Jelikož pozadí postavy na obrázku 3.2 má barvu velice podobnou barvě kůže, je detekce složitější, až téměř nefunkční [11].



Fig. 5. Image from UChile dataset (19980416\_ABC\_KeyFrame\_45882\_46004.jpg.png). Fig. 6. Skin detection with the first level rules.



Fig. 7. Skin detection with the second level rules.

Fig. 8. Skin detection with the third level rules. Observe that large number of non-skin regions are detected as skin.

Obrázek 3.2 Znázornění detekce kůže pomocí tří úrovní poměrů RGB. Špatně detekovatelný obraz [12].

## 4. VÝVOJ SLEDOVACÍ APLIKACE

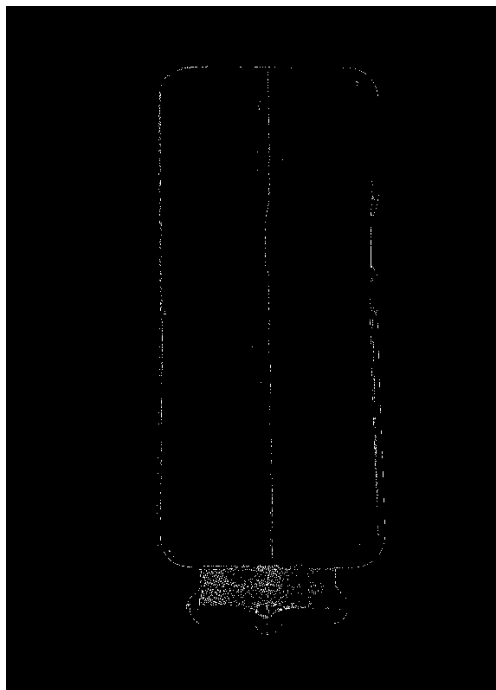
Od aplikace bylo požadováno určit postavu v obraze a získat její středovou osu. Podle této osy se měly určit odchylky pravé strany oproti levé straně.

### 4.1 Střed pomocí hran objektu

Prvotní myšlenkou bylo najít hrany objektu v obraze a za jejich pomoci získat středovou osu objektu. Nalezení hran proběhlo standardní funkcí „*edge*” s parametrem „*Sobel*“ značící název vyhledávací metody. Výpočet středové osy byl pak dopočítán jako rozdíl indexů pixelů hran na daném řádku, kterým se získala velikost objektu a polovina této velikosti se přičetla k indexu prvního z hranových pixelů na řádku.

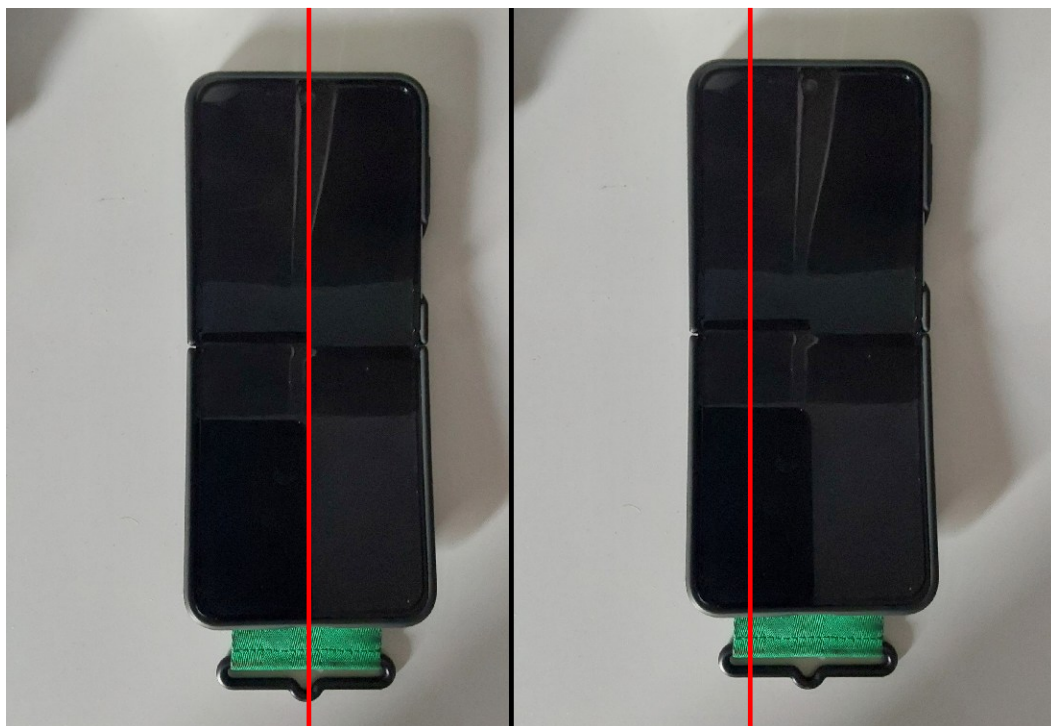
$$stred = col(1) + \frac{col(2) - col(1)}{2}, \quad (4.1)$$

Středový pixel je počítán pro každý řádek, tedy je tato rovnice vepsána do „*for*” cyklu, o délce výšky obrázku, s iteračním parametrem pro proměnnou *stred*. Proměnné *col(1)* a *col(2)* pak značí nalezený první a poslední pixel hrany na daném řádku pomocí funkce „*find*” s parametry „*first*” a „*last*”. Výsledek zlomku v rovnici (4.1) je ovšem nutné zaokrouhlovat kvůli neceločíselným výsledkům, které by nebylo možné použít jako indexové číslo pixelu.



Obrázek 4.1 Vykreslení nalezených hran a středové osy mobilního telefonu v obraze

Z nalezených pixelů středu je vypočítán průměrný index pozice pixelu a tento index použit na všech řádcích jako střed nalezeného objektu, tedy mobilního telefonu. Tím vznikne středová osa protínající telefon.



Obrázek 4.2 Vykreslení nalezené osy telefonu. Vlevo nejlepší dosažený výsledek. Vpravo obvyklý výsledek.

Ovšem tento způsob hledání osy není vždy přesný. Aby bylo dosaženo co nejlepšího výsledku, musela se při průměrování vybírat pouze oblast uvnitř objektu, aby se zamezilo započítání šumových pixelů, obr. 4.2 vlevo. V případě použití všech nalezených středových pixelů, které se ve spoustě případů nenacházejí v pomyslném středu objektu, se výsledná osa posune mimo skutečný střed objektu viz obr. 4.2 vpravo.

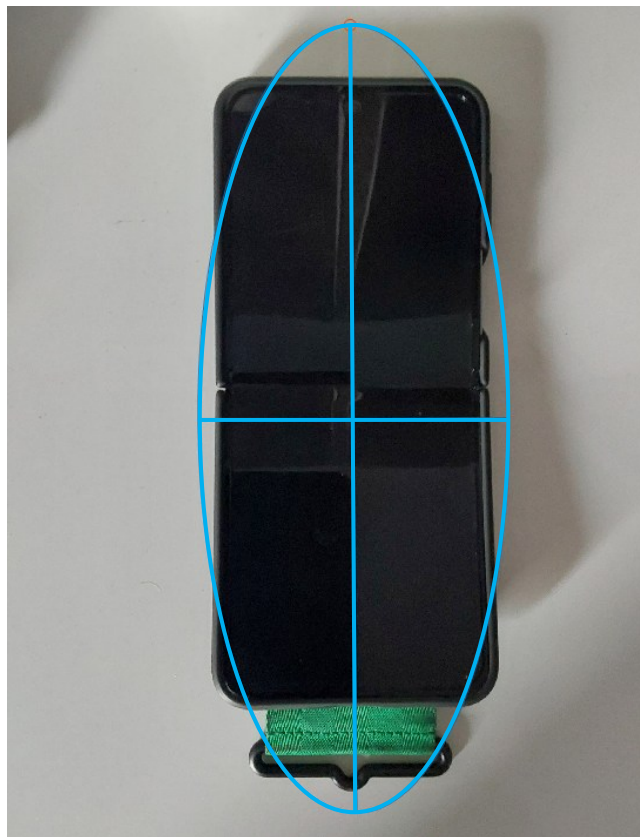
## 4.2 Implementace metody momentů

Další z využitelných metod pro nalezení středové osy je metoda momentů. Tato metoda pracuje na principu matematických rovnic popisujících tvar objektu, tzv. momenty objektu, resp. momenty tvaru. Pomocí těchto momentů lze pak vypočítat rozměry elipsy obkreslující objekt.

Metoda momentů pracuje s černobílým obrazem, ve kterém rozlišuje černé pixely jako nepodstatné a bílé pixely jako hledaný objekt. Je tedy potřeba obraz vždy před použitím funkcí převést do binárního obrazu.

Elipsa je výsledkem funkce „*get\_ellipse*“, která vypočítává momenty objektu a z nich jednoduchými poměry a vzorci získává rozměry a parametry elipsy. Poté z těchto rozměrů a parametrů i výslednou elipsu. Následně pomocí funkce „*draw\_ellipse*“ je

elipsa vykreslována do obrazu. Kromě samotné elipsy jsou vypočítávány a mohou být vykresleny i hlavní a vedlejší osa elipsy, nebo například směr elipsy, nebo přesněji řečeno směr hlavní osy, tedy její počáteční průsečík s elipsou.



Obrázek 4.3 Demonstrační vykreslení možností funkcí „*get\_ellipse*“ a „*draw\_ellipse*“

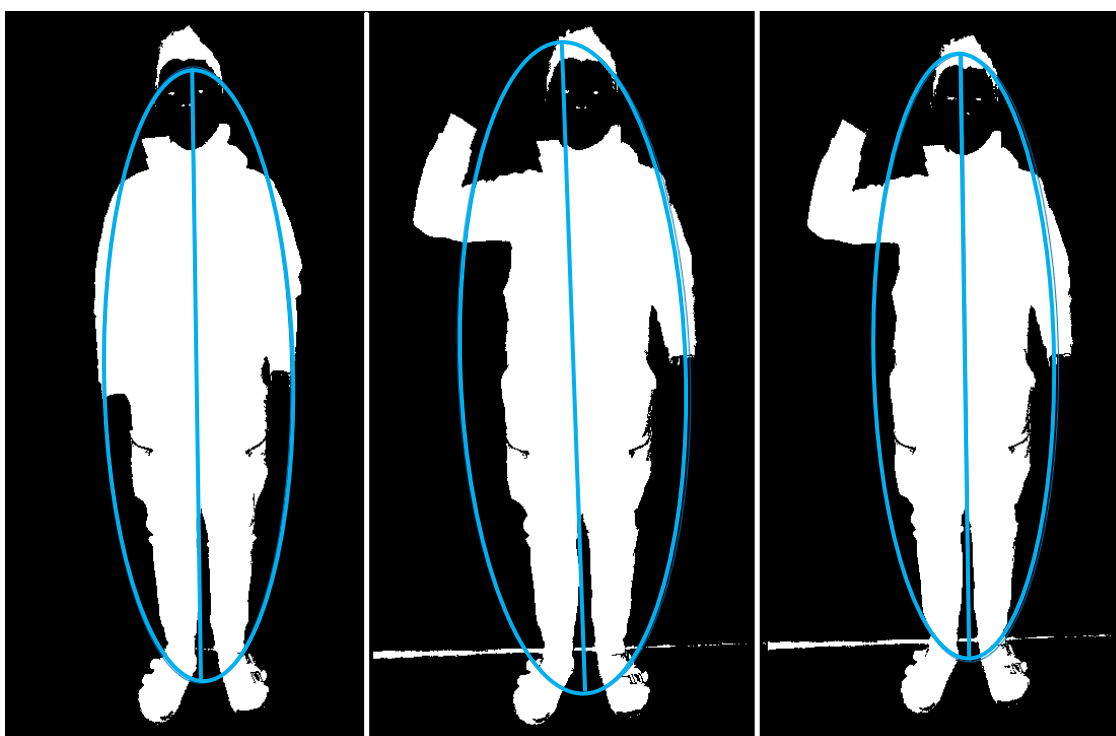
Obě tyto funkce byly zapůjčeny od Raphaël Candelier z Francouzských laboratoří Laboratoire Jean Perrin a autor je nabídnul volně ke stažení a použití. Funkce byly použity pro lehkou úsporu práce a nejsou perfektní implementací metody momentů, ale pro použití v této práci plně postačují [9].

### 4.3 Určení středové osy

Od této chvíle je už hledána v obraze postava člověka, nicméně tento popis by měl být využitelný i v případě hledání objektu. V první části aplikace probíhá kalibrace středové osy podle rovně stojící postavy, kdy se středová osa nalezne funkcí „*get\_ellipse*“ jako hlavní osa elipsy „*major*“. Pomocí funkce „*draw\_ellipse*“ se osa vypíše z výstupní struktury funkce „*get\_ellipse*“ viz obr. 4.4 vlevo. Jelikož výpis hlavní osy z funkcí jsou pouze počáteční a koncový bod této osy. To je postačující počet parametrů pro sestrojení úsečky, ovšem jsou potřeba konkrétní pixely na rádcích obrazu, kterými osa prochází. Je tedy nutné zjistit vzdálenostní rozdíl mezi těmito body a vytvořit vektor lineárně

rozložených hodnot, jehož velikost se bude rovnat právě vzdálenosti těchto dvou bodů. Tím jsou získány pixely, kterými osa prochází na každém řádku obrazu.

Ve druhé části se znovu nalezne středová osa pro nový snímek postavy, kde postava například pozvednula ruku, či se naklonila. Jelikož se tvar postavy změnil, změnili se i momenty, tedy se změnila i elipsa a hlavní osa, která se v tuto chvíli naklonila viz obr. 4.4 uprostřed. Zde se využije původní kalibrační osa. Z nové nakloněné osy je zjištěna průměrná hodnota pořadí pixelů na řádcích, kterými osa prochází. Je vytvořen nový vektor rozdílu této průměrné hodnoty a pixelů kalibrační osy. Znovu se vypočte průměr tohoto vektoru posunu a výsledná hodnota je brána jako průměrná vzdálenost, o kterou se musí každý pixel kalibrační osy posunout. Nyní je kalibrační osa ve středu postavy na novém obraze viz obr. 4.4 vpravo.



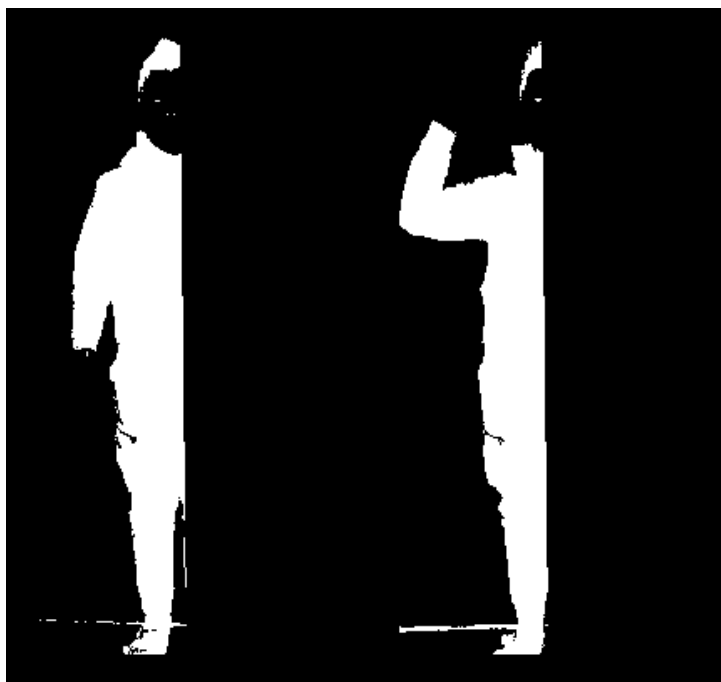
Obrázek 4.4 Výpočet kalibrační středové osy postavy. Vlevo kalibrační postoj. Uprostřed cvičící postoj s nakloněnou osou. Vpravo cvičící postoj s kalibrovanou osou.

#### 4.4 Převrácení dle středové osy

Osa nalezená posunutím kalibrační osy se uloží do vektoru, který se použije pro převrácení postavy. Převrácení poloviny objektu dle středové osy je realizováno vytvořenou funkcí „*flipOs*“, která řádek po řádku převrací pravou stranu obrazu doleva dle středové osy, tedy každý řádek dle určeného středového pixelu z osového vektoru.

Jelikož je nutné předpokládat, že postava nebude stát přesně uprostřed obrazu, musí být převrácení řádků omezeno o procentuální hodnotu šířky obrazu, aby nedocházelo

k převracení neexistujících pixelů za okrajem obrazu. Funkce pak vrací původní a převrácenou polovinu obrazu odděleně.



Obrázek 4.5 Demonstrace oddělených polovin postavy. Vlevo převrácená polovina. Vpravo nepřevrácená původní polovina.

## 4.5 Určení odchylek

Odchylka pravé strany od levé je určována odečtením těchto samostatných polovin od sebe. Výsledný obraz je v absolutní hodnotě, aby nedošlo ke ztrátě pixelů, jelikož se případné záporné hodnoty v obraze obvykle nulují. Tím jsou získány pixely, ve kterých se poloviny neshodují.



Obrázek 4.6 Výsledek vzájemného odečtu oddělených polovin

Tento rozdíl polovin je následně přičten k červené složce původního obrazu. Výsledkem je červené označení nesouhlasících částí mezi polovinami postavy.



Obrázek 4.7 Výsledný obraz s vyznačením odchylek polovin postavy

## 5. POSTUPNÉ ZDOKONALOVÁNÍ APLIKACE

### 5.1 Zpřesňování odchylky

Jak lze vidět na obrázku 4.7, odchylky polovin postavy jsou nepřesné z důvodu nedokonalého určení ideálního středu postavy a také nedokonalostí postavy, jelikož tvar postavy člověka není perfektně symetrický. Je tedy třeba toleranci odchylky upravit, což je provedeno morfologickým otevřením obrazu odchylky před vložením do původního obrazu. Toto způsobí vymizení přebytečných odchylek, jaké jsou vidět např. na noze postavy, nebo také na hlavě. Aby došlo k dostatečné změně tolerance a k vyhlazení odchylky ve všech směrech bez ostrých hran, byl jako strukturální element použit kruh s poloměrem 18 pixelů. Při zmenšování poloměru se bude vykreslovat více detailů odchylek, naopak při zvětšování se budou detaily ztrácet. Pro dosažení stejných výsledků při jiném rozlišení obrazu se bude hodnota poloměru lišit.



Obrázek 5.1 Výsledný obraz s vyznačením odchylek polovin postavy

### 5.2 Zlepšení výraznosti postavy

Pro zlepšení detekce postavy v obraze lze vstupní obraz upravit, aby byl lépe zpracovatelný použitými funkcemi. Jelikož jsou na obraze stíny a různá intenzita světla, nemusí být postava dostatečně čitelná, proto se provádí se tzv. preprocessing, kdy se obraz upravuje pro lepší nebo přesnější zpracování.

V této práci je obraz převáděn do binárního obrazu, což znamená, že světlé pixely obrazu se převedou na 1, bílou barvu, značící pozadí a tmavé pixely na 0, tedy černou barvu, označující popředí. Pro práci v aplikaci je ovšem třeba černou a bílou barvu vyměnit, takže jsou binární obrazy invertovány a pozadí je tedy černé a popředí bílé. Tento typ obrazu je potřebný pro výpočet momentů obrazu.

Aby bylo možné co nejlépe určit tmavé a světlé pixely, je potřeba tmavé pixely co nejvíce ztmavit a světlé pixely více zesvětlit a zvýraznit. Také je potřeba zesvětlit tmavá místa, jako např. stíny. Proto je v aplikaci prováděno dvěma funkcemi zesvětlení tmavých míst a světlých míst obrazu.

Pro zesvětlení světlých míst je používána funkce „*imlocalbrighten*“. Tato funkce zvyšuje kontrast obrazu a tím postavu zvýrazňuje a zlepšuje čitelnost postavy. Zároveň odstraňuje částečně stíny na pozadí.

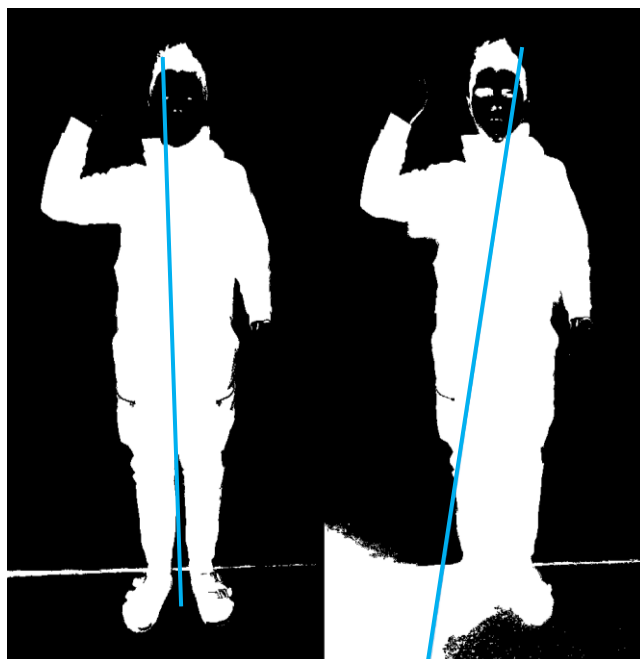


Obrázek 5.2 Demontrace úpravy funkcí „*imlocalbrighten*“. Vlevo původní obraz. Vpravo upravený obraz.

Další v pořadí je použita funkce „*imflatfield*“, která zesvětluje obraz jako celek a nemění jen kontrast. Tato funkce téměř odstraní nežádoucí stíny a postavu ještě více zvýrazní. Tato funkce se ovšem snaží vyvažovat světlá místa s tmavými. Pokud by tedy byla postava tmavá, mohla by v obraze více zaniknout. I z tohoto důvodu je použita funkce „*imlocalbrighten*“, aby postava byla více světlá a lépe ji funkce „*imflatfield*“ rozpoznala. Jelikož zvýšení kontrastu nejvíce působí na barevnou složku, která je v daném pixelu nejvíce obsáhlá, černé oblečení může například zmodrat působením venkovního světla, jak lze vidět na obrázku 5.2. V místnosti osvětlené pouze běžnými žlutými žárovkami dostane černé oblečení odstín žluté až zelené.



Obrázek 5.3 Demontrace úpravy funkcí „*imflatfield*“. Vlevo původní obraz. Vpravo upravený obraz.



Obrázek 5.4 Demontrace úpravy funkcí „*imlocalbrighten*“. Vlevo upravený obraz preprocessingem. Vpravo neupravený obraz preprocessingem.

Při použití obou těchto funkcí je pak na binárním obraze téměř pouze postava a vše ostatní je nulové. Jak lze vidět na obrázku 5.4 vpravo, postava je sice rozeznatelná, ale u jejích nohou se rozprostírá bílá skvrna, vzniklá nežádoucím stínem.

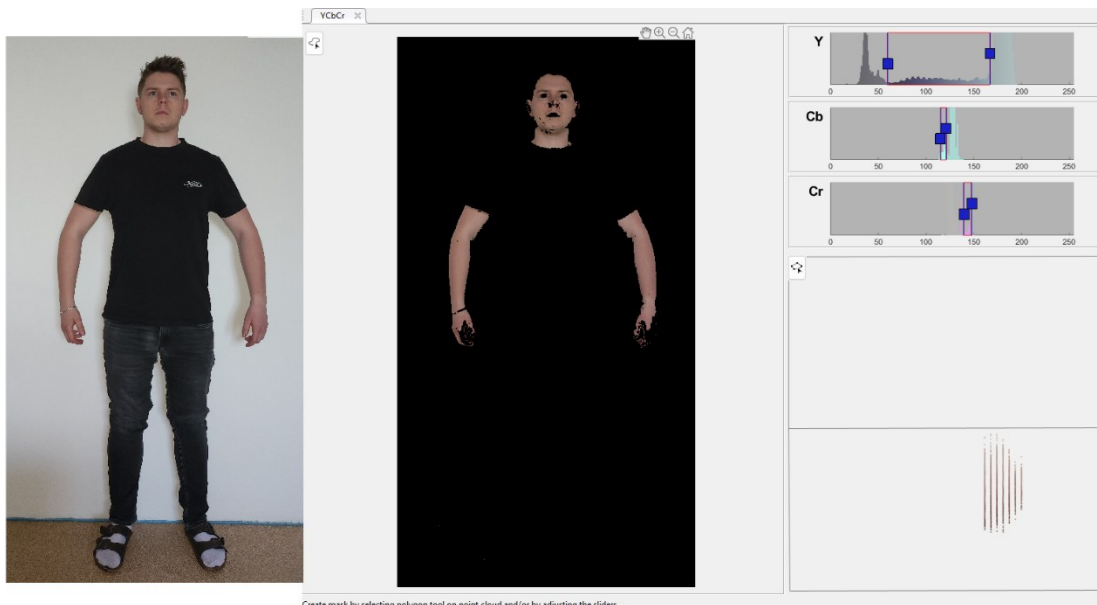
Je také vidět, že nalezená osa metodou momentů se vlivem této skvrny naklání doprava, jelikož je skvrna započítána do výsledného tvaru objektu, resp. postavy. Díky preprocessingu se podařilo skvrnu téměř plně odstranit viz obr. 5.4 vlevo.

### 5.3 Detekce kůže

Pro detekování částí postavy v barvě kůže, tedy obvykle rukou a obličeje, bylo využito interní funkce MATLABu „*color thresholder*“, pomocí kterého lze filtrovat jednotlivé barevné složky.

Kůže nejen že splývá se světlým pozadím, ale je také obtížně detekovatelná v jednotlivých barvách spektra. Z toho důvodu se vytvářejí funkce pro detekci kůže, které skládají hodnoty z různých barev spektra dohromady.

Touto funkcí tedy lze prahovat nejen barevné spektrum RGB, ale samozřejmě i ostatní barevné prostory, jako např. HSV, YCbCr, nebo  $L^*a^*b$ , ve kterých může být hledání správné kombinace barev jednodušší. Takto je možné z obrazu vytáhnout i jednu barvu, nebo barvu a její odstíny. Funkce má možnost označení konkrétního místa v obraze, pro které má automaticky prahovat barevné spektrum. Následně je možné odstíny ještě doladit ručně.



Obrázek 5.5 Znárodnění funkce MATLAB aplikace „*Color Thresholder*“.

Na obrázku 5.5 jsou poměrně dobře oddělené odstíny kůže od zbytku obrazu. Takto vytvořenou masku je možné exportovat jako funkci a použít její barevnou nebo i binární část. Tedy lze v obraze nalézt zvolenou barvu a zvýraznit ji. Ať už je prahování tvořeno v jakémkoliv barevném spektru, výsledná funkce je převedena do hodnot RGB. Tato vytažená část, tedy viditelná kůže na černém pozadí, se převede do binárního obrazu a přičte se k binárnímu obrazu postavy získaného ze vstupního obrazu. Takže získaný

binární obraz má o poznání větší váhu při výpočtu momentů a zároveň je nyní binární postava kompletní. Kde je viditelná kůže, nyní nejsou černá místa, ale celá postava je zaplněna bílými pixely.

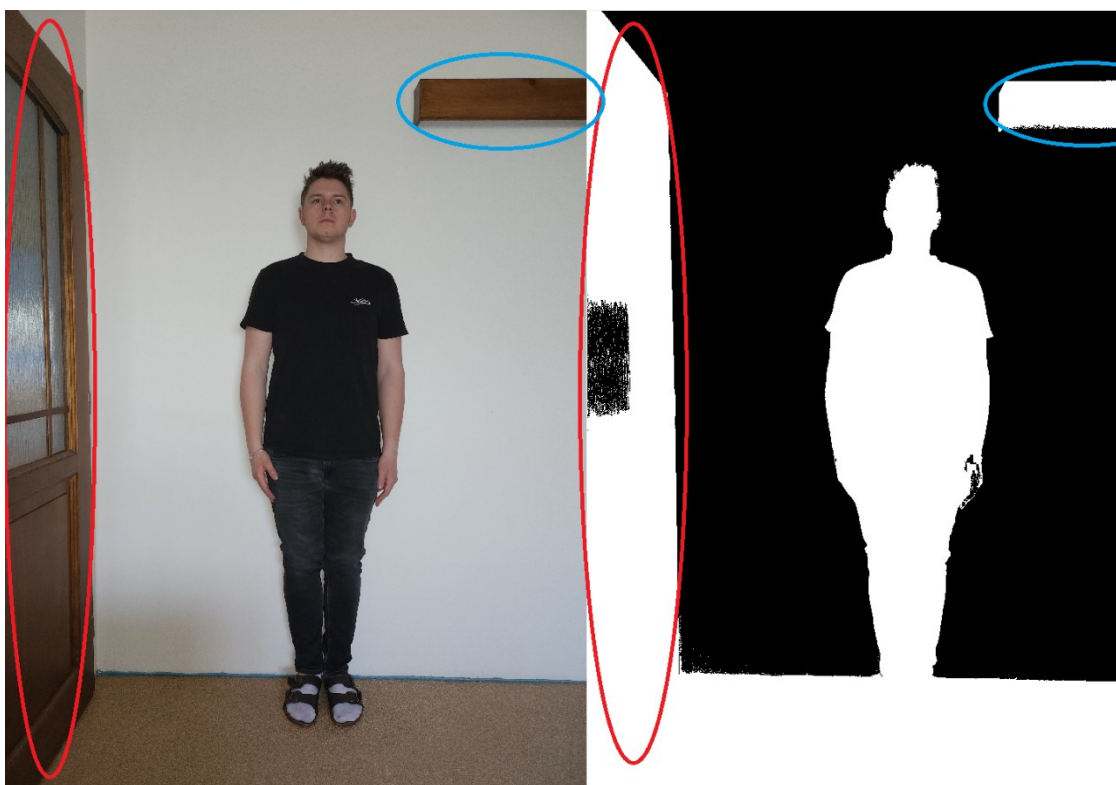


Obrázek 5.6 Postava s detekovanou kůží a převedena do binárního obrazu.

Pro potřeby aplikace toto snímání kůže postačuje, nicméně pro přesnější detekci by bylo možné využít metody RGB ratio, tedy poměrů RGB barev pro získání barvy kůže, zmíněné v úvodu této práce.

## 5.4 Odečítání pozadí

Velkým problémem snímání postavy jsou okolní objekty narušující obraz a komplikující výpočty. Jelikož program pracuje s tmavě oblečenou postavou před bílou stěnou, promítnou se do binárního obrazu také objekty okolo postavy, protože jsou taktéž tmavé vůči bílé zdi. Mohou poté ovlivňovat výpočty zvětšením počtu bílých pixelů v obraze, nebo přímo měnit váhu počtu pixelů v částech obrazu. Je nutné tyto nežádoucí obrazce eliminovat.

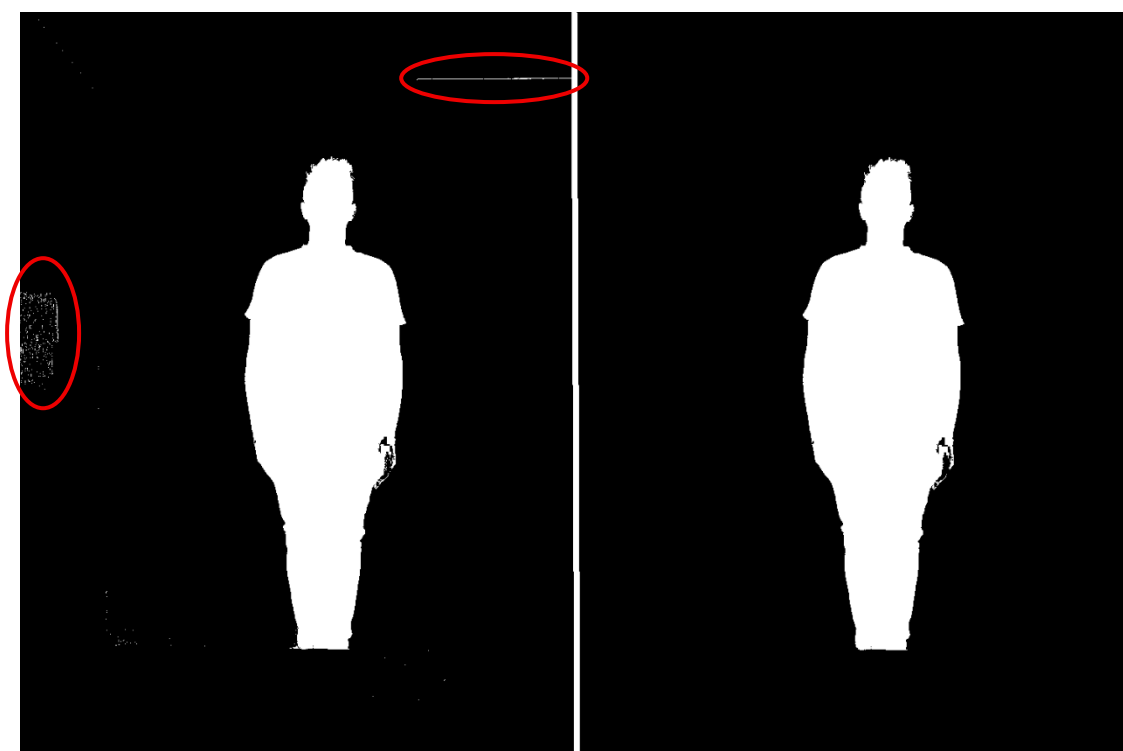


Obrázek 5.7 Detekce postavy s přítomností cizích objektů.

Jednou z možností je právě odečítání pozadí. Metoda, při níž je zdokumentováno nejprve pozadí samotné bez postavy, aby se jednoznačně určilo, co obsahuje. Následně se mohou pořizovat snímky postavy. Od těchto snímků je pozadí aritmeticky odečteno pixel po pixelu, čímž zůstane pouze postava.

Způsob odečítání pozadí ovšem není tak jednoduchý. V případě odečítání barevného obrazu by sice došlo k odečtení pozadí, ovšem při odečítání hodnot pixelů mezi sebou by došlo i k odečtení původního pozadí od pixelů postavy. To by způsobilo změnu barev postavy, nebo její narušení. I proto se toto odečítání provádí v binárním obraze. Ten by se dal potom používat i jako maska pro další odstraňování pozadí.

Pro potřeby této aplikace se pracuje pouze s odečtením na binární úrovni. Po odečtení obrazců z pozadí zůstane pouze obrys postavy, se kterým lze dále pracovat. Odečtení obrazců zpravidla nebývá přesné, a ne všechny pixely se navzájem vynulují. Je tedy dobré použít ještě nějaké morfologické operace pro odstranění těchto přebytečných chyb a vyplnění děr. V aplikaci se nejlépe osvědčila funkce „*bwareafilt*“, která filtruje v obraze chyby a šum dle velikosti. Zadaným číslem ponechá počet největších tvarů v obraze. V tomto případě jeden největší obrazec – tedy postavu. Odečítání pozadí by v tomto případě mělo sloužit spíše jako pojistka, aby se na pozadí nevyskytnul objekt větší než samotná postava.



Obrázek 5.8 Binární obraz postavy po odečtení pozadí, před (vlevo) a po (vpravo) filtraci chyb odečtení obrazců.

Tímto způsobem je tedy možné vytáhnout postavu z obrazu bez preprocessingu, tedy již není nutné obraz zesvětlovat a zvyšovat kontrast pro odstranění stínů, šumu a nežádoucích obrazců, jak je popsáno v podkapitole 5.2. Také byly eliminovány stíny vytvářené přímo postavou posunem člověka blíže k pozadí.

## 5.5 Výpočet měřítka obrazu

Pro výpočet odchylek postavy v centimetrech je potřeba znát měřítko vstupního obrazu. Z toho důvodu je pro aplikaci vytvořena funkce výpočtu měřítka. Funkce změří velikost postavy v pixelech. Následně vypočítá z dosazené reálné výšky postavy v milimetrech počet milimetrů na jeden pixel. Například funkce změří výšku postavy 2650 pixelů, její reálná výška je 182 cm, tedy 1820 mm. Měřítko je pak 0,68 mm/px.

## 5.6 Odměrování odchylek

Samotné měření odchylek postavy probíhá ve vytvořené funkci „*Odmer*“, do které se vkládá binární obraz odchylek získaný odečtením polovin postavy. Tento binární obraz s odchylkami se prochází nejprve horizontálně a poté vertikálně, pro nalezení, respektive změření horizontálních a vertikálních odchylek.

Při nalezení odchylky na daném řádku, popř. sloupci, se zapíše řádková a sloupcová souřadnice odchylky a změří se její velikost podle počtu bílých pixelů za sebou.



V tomto komplexním systému jsou tedy uloženy všechny potřebné údaje pro další výpočty a vykreslení odchylek.

## 5.7 Vykreslování vyměřených odchylek

Vykreslení probíhá ve funkci „*VykresOdchyl*“. Je to souhrn několika operací, a má čtyři vstupní argumenty. Měřítka a původní vstupní obraz, do kterého se bude odchylka vyznačovat. Dále binární obraz odchylek, jelikož funkce vyhledání odchylek je volána až v tomto kroku. A poslední z argumentů, volba požadovaných odchylek, horizontální, vertikální, nebo obě. Mezi argumenty patří také požadavek o zobrazení výsledného obrazu ihned po výpočtu, který je ovšem nepovinný a standartně je nastaven na 0, tedy že se obrázek nezobrazí.

Na začátku funkce se zavolá odměřovací funkce popsána v podkapitole 5.6 a její nalezené odchylky se přepočítají pomocí měřítka na odchylky v milimetrech. Všechny nalezené odchylky se červenou barvou vyznačí do vstupního obrazu. Toto vyznačení probíhá stejně jako na konci podkapitoly 4.4, tedy přičtením do původního obrazu. Pouze je již prováděno v jiné části kódu.



Obrázek 5.11 Vyznačení odchylek postavy červenou barvou. Vpravo přiblíženo pro lepší viditelnost.

Následně se v matici odchylek převedených na milimetry kontroluje velikost odchylek. Pokud odchylky přesáhnou hranici 30 cm, je to bráno jako špatná póza člověka na snímku a zobrazí se, že je odchylka moc velká.

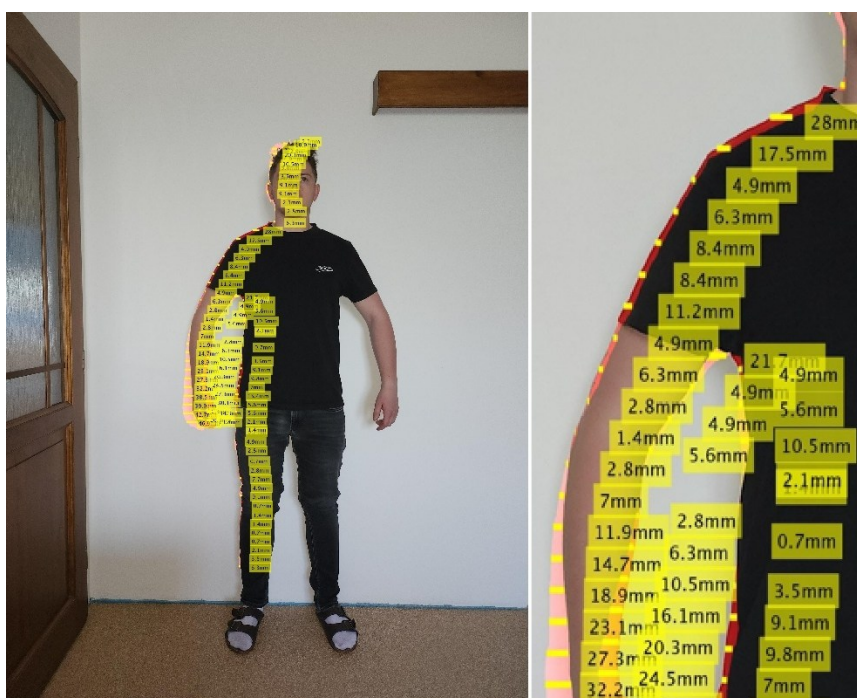


Obrázek 5.12 Zobrazení varování příliš velké odchylky při špatném postoji.

Také proběhne filtrace zbytečně velkých odchylek, jelikož je zbytečné vypisovat velikosti odchylek větších než zhruba 5 cm, protože větší odchylky jsou jednoduše již rozpoznatelné pouhým okem.

Poté se vypočítají z počátků a velikostí odchylek jejich koncové souřadnice, aby bylo možné vykreslovat čáry v místech, kde jsou měřeny odchylky, jejichž hodnoty jsou vypisované na obraze. Pro tyto výpočty je použit právě výběr, které odchylky je třeba zobrazovat, aby se vypočítaly pouze požadované odchylky a urychlil se tak chod aplikace.

V poslední řadě se vypíší do obrazu velikosti odchylek v milimetrech a vykreslí se žluté čáry v místě kde byla daná odchylka měřena.



Obrázek 5.13 Výsledný obraz s vyznačením a vypsáním odchylek postavy. Vpravo přiblíženo pro lepší čitelnost.

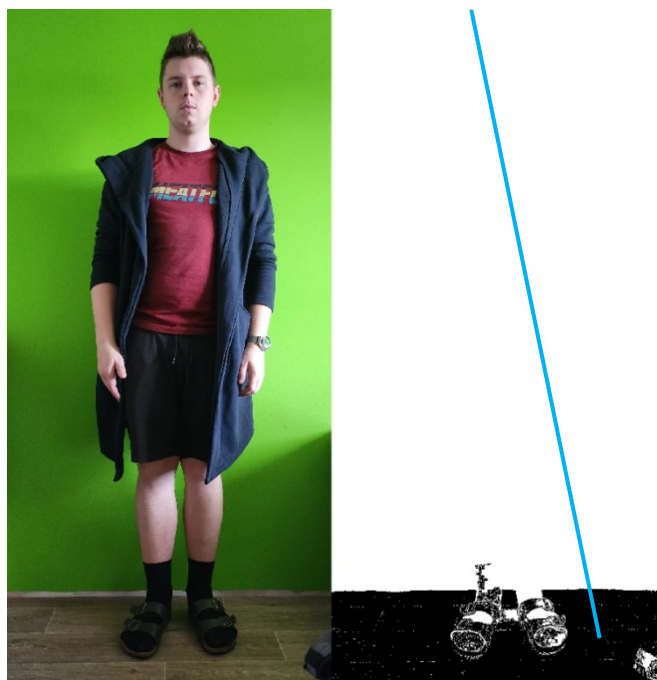
Nakonec se dle vstupního požadavku zobrazí nebo nezobrazí výsledný obraz. V každém případě se výsledek uloží do složky, kde se nachází aplikace s příponou jpg.

## 6. PODMÍNKY FUNKČNOSTI

Aplikace pracuje se základními možnostmi zpracování obrazu a je tedy stále omezena na přísně definovanou scénu snímků. Také by pro kalibraci a následné vyhodnocení mělo být použito stejné rozlišení fotoaparátu, aby nedošlo k nesprávnému určení středové osy postavy, špatnému výpočtu měřítka, nebo celkové nefunkčnosti programu z důvodu jiných rozměrů os a obrazů. Podstatná část těchto podmínek se ovšem vztahuje ke zpracování bez použití odečítání pozadí, které běh programu značně usnadňuje. Při použití odečtu pozadí některé z podmínek již nejsou relevantní.

### 6.1 Kontrast s pozadím

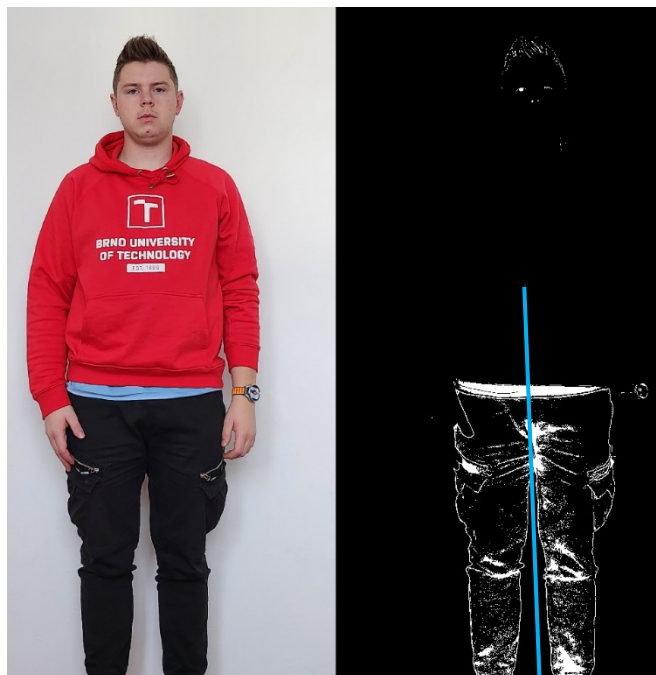
Pro správné převedení původního obrazu do binárního a následné správné zpracování, je zapotřebí vytvořit co největší kontrast mezi postavou, resp. hledaným objektem, a pozadím. Proto nejlepší volbou pro snímání je bílé pozadí, tedy v případě postavy bílá stěna. A aby bylo dosaženo největšího kontrastu, tak postava musí být oblečena do černého oblečení. Možností kontrastních kombinací je spousta, ale jejich provedení by mohlo být pro uživatele náročnější.



Obrázek 6.1 Demonstrace špatného kontrastu postavy s pozadím.

Zde na obrázku 6.1 je možné vidět špatný výběr pozadí a oblečení postavy. Pozadí, ačkoliv se zdá poměrně světlé a postava tmavá, tak po preprocessingu vstupního obrazu, kdy se funkce snaží vyrovnat celkový kontrast a jas obrazu, tak dochází k nadměrnému zesvětlení postavy, kdy se ztrácí na pohled viditelný kontrast s pozadím. Pozadí je pak

málo světlé, aby v případě binárního obrazu tmavě oblečená postava vyniknula z pozadí. Vidět jsou pouze nohy, kde i přesto že podlaha působí poněkud tmavě, tak v binárním obraze díky preprocessingu na ní částečně vynikají černé ponožky postavy. Při použití jiné funkce, např. pro změnu lokálního kontrastu, bylo dosahováno o něco lepšího, nicméně stále neuspokojivého výsledku. Postava sice byla viditelná, ovšem stále spíše její hrany než plný tvar, což je pro metodu momentů nedostačující.



Obrázek 6.2 Demontrace špatného výběru oblečení postavy.

Na obrázku 6.2 je naopak vidět opačný problém, a to nevhodný výběr oblečení pro použití s aplikací. Pozadí je velmi světlé, takže v binárním obraze se po negaci zobrazuje jako černé. Avšak červená mikina postavy, která se zdá jako dostatečně rozdílná oproti pozadí, tak složením barev nesplňuje požadavky kontrastu s bílým pozadím. Tudiž v binárním obraze naprosto zanikne v pozadí. To je způsobenou metodou měření kontrastu, jelikož aplikace pracuje právě s červenou složkou obrazu. Tím pádem v červeném spektru červená mikina nereflektuje vůči bílé stěně, protože stěna v tomto spektru také získá nádech červené.

Bohužel v tomto případě i kalhoty postavy, které jsou černé a velmi kontrastní s pozadím, téměř zanikají v pozadí, protože preprocessing vstupního obrazu se snaží vyrovnávat světelnost a kontrast celého obrazu. Tím je způsobeno nadměrné zesvětlení kalhot, a tedy i následné zaniknutí v pozadí. Proto je výběr oblečení zvolen takto. Při alternativním naprogramování by bylo možné Barvu oblečení nebo i pozadí změnit, nicméně zvolená kombinace je nejvíce univerzální. Možností by také bylo použití algoritmů výpočtu kontrastu na základě lokálních podmínek.

## 6.2 Cizí objekty

Programovou rozlišitelnost postavy ve vstupním obraze může ztížit i cizí objekt na obraze, který nepatří k postavě, ale kontrastuje s pozadím. Takovým objektem může být například stůl, nebo skříň, které jsou ve tmavém dekoru, tedy kontrastují s bílou stěnou.

Pak se takovýto objekt promítne i do černobílého obrazu a při výpočtech momentů je počítán tvar postavy i s takovýmto objektem. Momenty jsou tedy vypočítány špatně a jsou nevyhovující pro získání středu postavy. Střed by byl v tomto případě vychýlen v závislosti na velikosti cizího objektu.



Obrázek 6.3 Výpočet momentů a převod do binárního obrazu za přítomnosti cizích objektů na obraze bez použití preprocessingu.

Při nepoužití preprocessingu je výpočet momentů takřka nepoužitelný, protože se do celkového tvaru na obraze započítají i nežádoucí objekty, jako v tomto případě dveře po obou stranách postavy viz obr. 6.3. Elipsa obkreslující celý hledaný objekt nyní obkresluje celý obrázek, tedy i osa postavy není správná. Postava na obraze ani vizuálně nevyniká, ale naopak je rušena právě stíny spojenými s tmavými dveřmi. Funkce pro vyplnění děr v binárním obraze toto rušení ještě více podpoří.



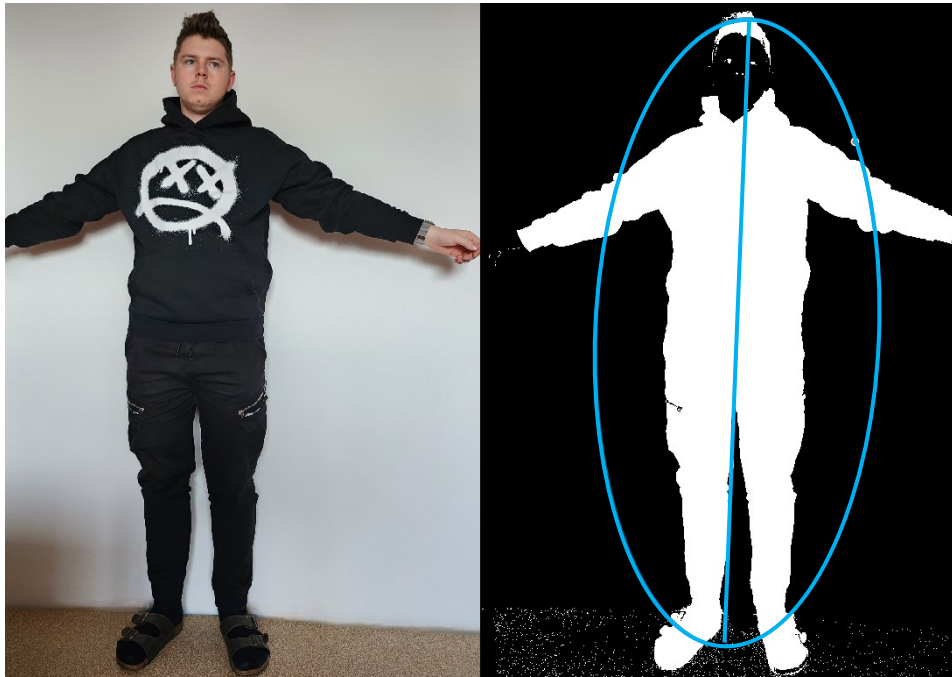
Obrázek 6.4 Výpočet momentů a převod do binárního obrazu za přítomnosti cizích objektů na obraze s použitím preprocessingu.

V případě použití vhodně zvoleného preprocessingu je možné se stínů zbavit a odstranit i dveře z binárního obrazu. Nicméně je stále viditelný tmavší obrys dveří. Tyto tmavé hrany stále ovlivňují výpočet momentů pro tvar postavy a stále je elipsa a osa postavy značně zkreslená, i přesto že vizuálně pouhým okem se obraz zdá být téměř v pořádku viz obr. 6.4.

Tyto problémy mohou nastávat při standartním snímání postavy. Nicméně v případě použití odečtu pozadí od popředí tyto problémy z většiny odpadají. Protože cizí objekty a stíny v binárním obraze vymizí a kontrast postavy s pozadím už není potřeba tak velký. Poté stačí neupravený vstupní obraz, který se odečtem pozadí zpracuje do téměř perfektního binárního obrysu postavy. Celý popis odečítání pozadí byl vysvětlen v podkapitole 5.4.

### 6.3 Úhel snímání

Důležitou podmínkou správné funkčnosti aplikace je úhel snímání postavy. Aplikace je vytvořena pro čelní snímání postavy. Pro výpočet momentů je nutný co nejmenší úhel snímání postavy, aby nedocházelo ke zvětšení tvaru postavy. Při snímání postavy ze strany je viditelný bok postavy, který se následně promítne i do binárního obrazu. Tvar postavy je pak na jedné straně výraznější a jsou tím ovlivněny hodnoty momentů. Důsledkem toho je pak vychýlena i středová osa postavy.



Obrázek 6.5 Odchýlení osy postavy na základě úhlu snímání

Ačkoliv na první pohled se zdá postava správně zaznamenána, na binárním obraze lze vidět odchýlení středové osy. Tato odchylka může být způsobena právě nasnímáním pod úhlem, kdy je vidět bok postavy, který ovlivňuje vypočtené momenty. Úhel nasnímání také ovlivňuje vizuální náklon postavy.

## 7. IMPLEMENTACE APLIKACE PRO SNÍMÁNÍ VIDEOA

Díky prostředí MATLAB lze aplikaci jednoduše implementovat i na videozáznam. Video se do prostředí nahraje jako sekvence po sobě jdoucích snímků. Společně s počtem snímků jsou tu uloženy i ostatní parametry videa. Jde tedy pouze o vyhodnocení každého snímku videa zvlášť pomocí navržené aplikace. Vyhodnocení videa by mohlo mít i širší využití nežli vyhodnocení jednoho obrazu a to např. pro správné držení těla a rovný postoj. Také by bylo možné používat pro opakování přednaučených cviků, nebo právě replikovatelnost cviku pravé a levé strany těla. Vše by bylo možné sledovat i v reálném čase. Uživatel by měl možnost okamžitě reagovat na odchylky a přizpůsobovat postoj správné pozici.

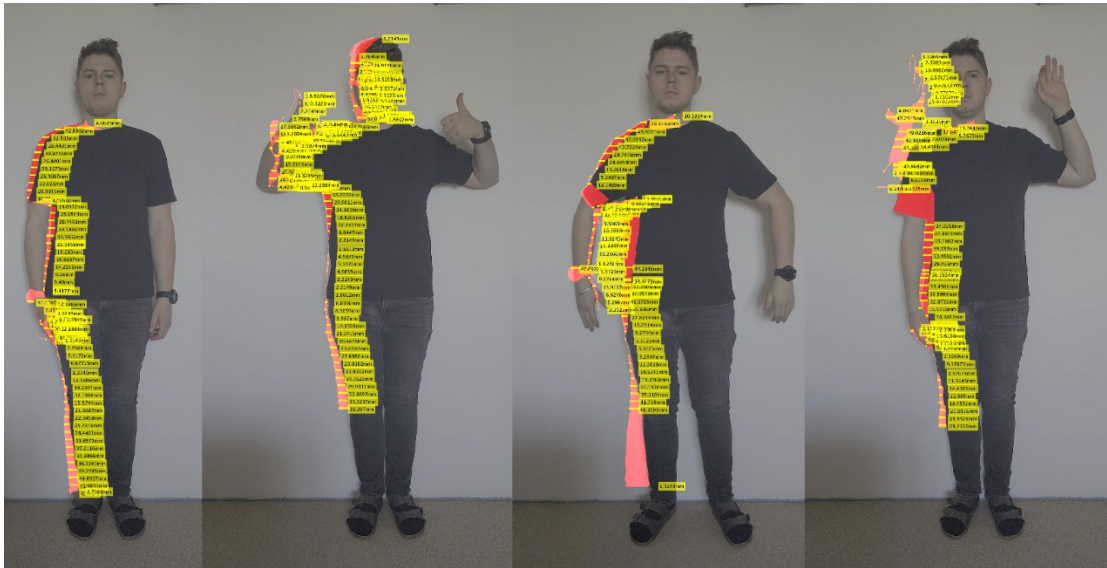
### 7.1 Převod aplikace pro zpracování videa

Jelikož se jedná o opakující se činnost zpracování snímků, je použita stejná aplikace, s pár drobnými úpravami, jako například while loop pro vyhodnocení všech snímků videa. Tento cyklus si hlídá, jestli je v sekvenci snímků platný snímek. V opačném případě se ukončí.

Na začátku se jako v původní aplikaci vytvoří středová osa, podle konfiguračního snímku s rovně stojící postavou viz podkapitola 4.3. Tato část kódu se neopakuje, provede se pouze při spuštění aplikace a nadále je středová osa uložena v proměnné.

Následně se načte i samotné video a jeho parametry a všechny snímky se za sebe uloží do proměnné. Od této chvíle je průběh aplikace uzavřen do cyklu while. Dále aplikace načte jeden z počátečních snímků, aby se uložil snímek bez postavy, kvůli odečítání pozadí. Předpokládá se, že uživatel bude operovat s aplikací sám, a tedy nebude na počátku videa před kamerou.

Po nastoupení uživatele do záběru se provádí stejný výpočet jako při vyhodnocení jednoho obrazu, ale pro každý obraz ze sekvence videa. Tedy se vyfiltruje pozadí vytvoří se osa, dle které se pak nastaví konfigurační osa a provede se překrytí polovin postavy. Pro každý snímek se vypočítá měřítko a všechny získané parametry se vloží do vykreslovací funkce. Výsledek vykreslovací funkce se nakonec zapíše do nové videosekvence a po vyhodnocení celého videa se soubor videa již mimo while loop uzavře a uloží do složky s aplikací.



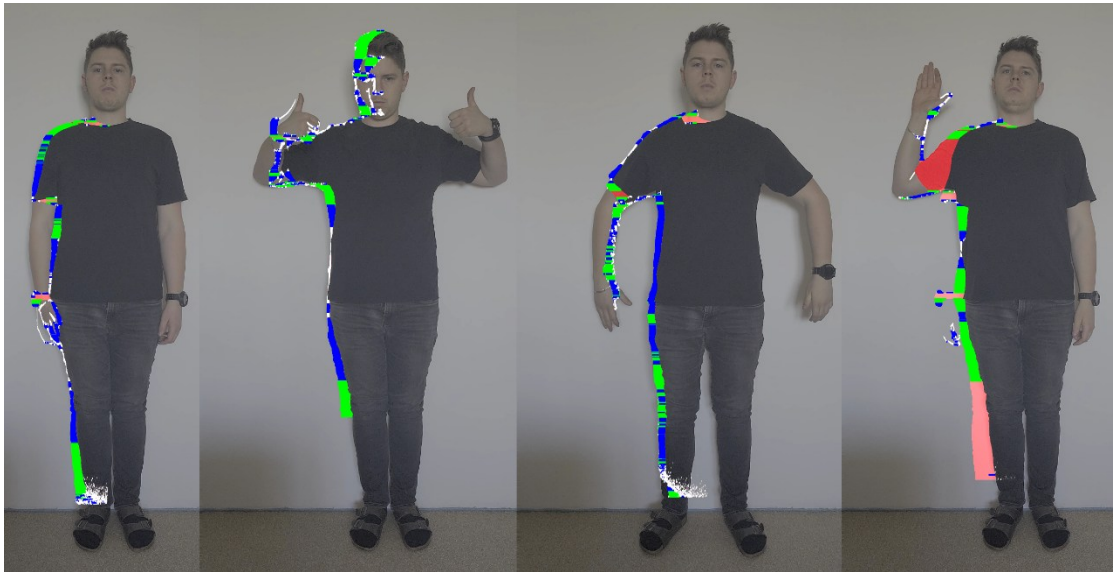
Obrázek 7.1 Ukázkové snímky při zpracování videa v původním formátu

## 7.2 Úpravy aplikace

Pro správnou funkčnost aplikace bylo nutné udělat pár změn v kódu. První a poměrně zásadní úpravou je ošetření vyhodnocení videa podmínkou, aby neprobíhaly výpočty 10 vteřin po začátku a 10 vteřin před koncem videa z důvodu pádu programu. Protože aplikace pracuje se středovou osou postavy a ve chvíli, co se člověk pohybuje do záběru nebo pryč z něj, není možné výpočet spolehlivě provést, protože postava nestojí kolmo ke kameře, je jinak natočená nebo nekompletní a program zkolabuje. Postava je kupříkladu v jednu chvíli částečně v obraze a částečně mimo něj, tedy program prakticky vyběhne ven z obrazu, což se pro něj jeví jako chyba a ukončí se. Jelikož se tvoří v aplikaci nová sekvence videa, převádí se tedy po tuto dobu vstupní video přímo na výstupní bez použití vyhodnocení.

Další úpravou je výpočet prováděný pouze každou vteřinu, tedy každý 30. snímek, což urychluje vyhodnocení celého videa. Provádět výpočet pro každý snímek v sekvenci může být zdlouhavé a pro běžného uživatele nepodstatné. Z pohledu uživatele se jedná pouze o plynulost změn odchylek ve videu a rychlé změny se jeví až nepřehledné, zatímco změny každou vteřinu jsou poněkud přehlednější a uživatel je schopen si video pohodlně zastavit v místě kde se mu odchylka bude zdát posouzení hodna.

Poslední úpravou je podmínka vykreslování pouze obrazu s platným měřítkem. Může totiž nastat situace, že se pro některý z obrazů nepodaří vypočítat měřítko a do hodnoty měřítka se uloží 0 nebo NaN, což znamená Not a Number, tedy hodnota není číslo. Program pak ukončí výpočet, protože mu chybí zásadní údaj pro vyhodnocení. Tento problém je vyřešen jednoduchou podmínkou. Zapříčiněno to může být například špatným postojem postavy, kdy program nedokáže postavu správně oddělit od pozadí.



Obrázek 7.2 Ukázkové snímky při zpracování videa ve formátu „DUHA“

Pro jednodušší posouzení je v programu implementována možnost tzv. duhového znázornění odchylky, kde bílou barvou jsou podbarveny všechny odchylky. Odchylky mezi 1 a 3 cm jsou zbarveny modře, mezi 3 a 5 cm zeleně a nad 5 cm červeně. Odchylky menší 1 cm tedy zůstanou bílé. Tento způsob vyhodnocení je vhodné používat např. pro video, aby bylo rychle rozpoznatelné, kde je odchylka velká a kde naopak malá. Následně při budoucím možném použití vyhodnocování v reálném čase by bylo podstatně jednodušší pro uživatele rozpoznávat a rychle reagovat na vykreslované odchylky.

## 8. ZHODNOCENÍ PRÁCE

Aplikace v této práci je schopna sledovat pohybující se postavu a vykreslovat její odchylky pravé strany vůči levé, ať už na statickém obraze nebo i na videu. Odchylky dokáže měřit v rádech desetin milimetrů ať už vertikálně nebo horizontálně. Aplikace je schopna pracovat s nežádoucími objekty na pozadí pomocí odečítání pozadí. Objekty se pak nepromítnou do binárního obrazu použitého pro výpočet. Problémem pro program může být například béžová podlaha, kterou lze vidět na obrázcích, protože má odstín podobný barvě kůže, takže aplikace ji snímá jako součást postavy. Také si neumí plně poradit s nohama, které jsou dosti asymetrické a jelikož není osa perfektně v jejich středu a vyhodnocení odchylek v těchto místech je nepřesné. Funkční provedení aplikace je v přílohách i s demonstračními fotografiemi a videi.

### 8.1 Použití vhodných metod pro zpracování

Pro řešení byly použity metody úpravy kontrastu, odečítání pozadí, detekce kůže, nebo stěžejní metoda momentů. Právě metoda momentů je teoreticky rozebírána v kapitole 1, prakticky pak v podkapitole 4.2. Následně je opakovaně zmiňováno použití v celé práci. Detekování kůže je vysvětleno v kapitole 3 a aplikováno v podkapitole 5.3, kde díky detekci kůže je možné sledovat kompletní postavu i s obličejem a s např. tričkem s krátkými rukávy. Úpravy obrazu jsou provázány celou prací.

### 8.2 Definice sledovaných charakteristik a podmínky pro jejich správné nalezení

Právě obrys postavy v binárním obraze je využíván pro výpočet metodou momentů. Aplikace pracuje na bázi sledování odchylek pravé a levé poloviny těla. Zaznamenává velikosti a pozice těchto odchylek a z toho tedy celkový postoj a pozici postavy člověka. Podmínky a doporučení pro správnou funkci a chod aplikace jsou definovány v kapitole 6., kde je možné např. najít požadované oblečení a pozadí pro správnou funkčnost, nebo třeba i úhel kamery, který by měl být ideálně kolmý k postavě. Velkým přínosem je odečet pozadí, který velkou část podmínek řeší. Např. stíny v pozadí, nebo právě cizí objekty, se kterými měl program původně problém. Je tím zpřesněna i metoda momentů, která není ovlivněna právě cizími objekty v binárním obraze. Vzorové, testovací a demonstrační snímky a videa se nacházejí v přílohách.

### **8.3 Návrh zpracování a vyhodnocení sledovacích algoritmů a možnosti ukládání snímků a videosekvencí.**

V práci je sepsána kompletní realizace sledovací aplikace včetně podmínek a použitých algoritmů. Výsledky jsou ukládány jako snímky, popř. videa s vykreslením odchylek a jejich velikostí, nebo pouze barevným označením malých a velkých odchylek. Při překročení 30 cm, jakožto velkého vychýlení postavy, je vypisováno hlášení o příliš velké odchylce. V obraze a videu jsou odchylky zobrazovány červeně, popisy velikostí žlutě a místa měření velikostí taktéž žlutě, popř. použitím vykreslení barvami. V kapitole 5.7 je popisována funkce pro vykreslení odchylek včetně jejich měření, které se provádí s přesností na desetiny milimetru.

### **8.4 Zhodnocení kvality výsledků a zpracování.**

Jak již bylo zmíněno, navrhnutá aplikace je součástí příloh, a její řešení je popsáno v této práci. Aplikace je schopna nalézt postavu v obraze bez nežádoucích objektů v pozadí a vyhodnotit odchylku pravé poloviny těla vůči levé s přesností na desetinu milimetru. Měřítka obrazu je vypočítáváno ze zadané výšky postavy a program sám z výšky postavy v pixelech dopočítá hodnotu měřítka. Celé vyhodnocení je vykreslováno přímo do obrazu a ukládáno jako obrázek, popř. video. Při nadměrné odchylce je hlášením na snímku uživatel upozorňován na přílišnou odchylku a měl by se tedy lépe srovnat. Je zde využito i snímání kůže, které taktéž pracuje dobře a napomáhá kvalitě zpracování pomocí metody momentů. Při zpracování videa je doba vyhodnocení nadměru dlouhá, což by bylo nutné ještě odladit případným zlepšením hledání odchylek. Aplikace je tvořena pro černě oblečenou postavu na bílém pozadí a s obtížemi si poradí s jinými barvami, nebo spíše vůbec. Nicméně z tohoto důvodu práce obsahuje podmínky pro správnou funkci. Problémem může být i nesprávně zvolená podlaha, např. v béžové barvě podobné barvě kůže. Ne vždy se podaří i správně určit odchylky nohou, protože aplikace kvůli špatné barvě podlahy není schopna provést výpočet až k chodidlům, a je nutné s tím počítat při výpočtu měřítka. Také jsou nohy různě odchýleny od osy, narozdíl od těla, takže je problém i s jejich vyhodnocením. Přílohy obsahují převážně zkušební snímky používané při vývoji aplikace a hledání chyb při snímání.

## 9. ZÁVĚR

Práce se zabývala návrhem sledovací aplikace pro účely snímání hybnosti postavy člověka za pomoci srovnání pravé a levé poloviny otočením přes středovou osu. Vykreslení odchylky a jejích hodnot se provádělo barevně včetně velikostí v milimetrech do nového obrazu, který se následně ukládal. Jsou tu objasněny všechny použité funkce a metody pro správné nalezení postavy v obraze pomocí metody momentů, odstranění nežádoucích předmětů pro výpočet pomocí odečítání pozadí, snímání kůže prahováním hodnot barevných spekter a konečné vyhodnocení odchylek této postavy.

Také práce vysvětluje použití této aplikace na video záznam a potřebné úpravy pro správný a efektivní chod aplikace při vyhodnocení odchylek ve videu a následné ukládání videa s vykreslenými odchylkami, jak ve formátu vypisování velikostí, tak i tzv. duhovém formátu. Bylo využíváno stejného programu jako pro jeden obraz. V práci jsou objasněny všechny změny nutné pro správné fungování aplikace při zpracování videa.

Program jako takový je schopen odfiltrovat nežádoucí předměty v pozadí odečtem pozadí od popředí. Podmínkou správné funkce je bílé pozadí a ideálně černé oblečení postavy pro dostatečný kontrast, nicméně program je schopen snímat i kůži, tedy snímá i obličej a není třeba ani například dlouhých rukávů. Aplikace je také poměrně přesná. Je schopna vyhodnotit odchylku na desetiny milimetru a výsledek vypisuje přímo do obrazu i s označením místa měření vypsané odchylky.

Také nabízí možnost výběru horizontálních a vertikálních odchylek, pro urychlení výpočtu programu a přehlednost při optické kontrole. Použitelný je na různé velikosti postavy díky funkci měřítka, kam stačí pouze zadat výšku snímané postavy. Pro přehlednost a opakovatelnost kódu je také více operací rozděleno do funkcí.

Program by bylo možné v budoucnu ještě urychlit, například optimalizováním hledání odchylek, jako třeba hledání odchylky v blízkosti pixelu nalezení odchylky na předešlém řádku. Nabízí se možnost dopracování hlášení, na kterou stranu se musí postava pohnout, aby se odchylka minimalizovala. Také by bylo možné „naučit“ program rozeznávat více barev a pozadí, pro větší flexibilitu použití. Lepšímu zhodnocení postojů, nebo hybnosti a lepší použitelnosti by napomohla i možnost volby srovnávací a srovnávané strany, případně kombinace obou možností srovnání stran. Jistým pokračováním práce by bylo jistě real-time provedení, aby bylo možné na odchylky okamžitě reagovat. Možná by byla implementace do jiného kódu s vytvořením vstupních parametrů, například C++, a vytvoření uživatelské aplikace, ovládané pouze vstupními parametry a využívající integrovanou kameru např. mobilního telefonu, nebo webkameru počítače.

## POUŽITÁ LITERATURA

- [1] BOUTHEMY, Patrick; JAIN, Mihir a JÉGOU, Hervé. *Improved Motion Description for Action Classification*. Online. In: Frontiers in ICT. January 2016. Dostupné z: [https://www.researchgate.net/figure/Optical-flow-field-vectors-shown-as-green-vectors-with-red-end-points-before-and-after\\_fig6\\_290181771](https://www.researchgate.net/figure/Optical-flow-field-vectors-shown-as-green-vectors-with-red-end-points-before-and-after_fig6_290181771). [cit. 2025-01-03].
- [2] *Kvadratické povrchy*. Online. In: Geeksforgeeks.org. 2024, aktualizováno 1. srpna 2024. Dostupné z: <https://media.geeksforgeeks.org/wp-content/uploads/20240731111812/Quadric-Surfaces.png>. [cit. 2025-01-03].
- [3] CARVALHO, Paulo Cezar; ROCHA, L. a VELHO, Luiz. *Image Moments-Based Structuring and Tracking of Objects*. Online. In: IEEE Xplore. Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing, Fortaleza, Brazil, 2002. Dostupné z: <https://ieeexplore.ieee.org/document/1167130>. [cit. 2025-01-03].
- [4] *10.4 Hu Moments*. Online. In: Wordpress.com. 2020. Dostupné z: <https://cvexplained.wordpress.com/2020/07/21/10-4-hu-moments/>. [cit. 2025-01-03].
- [5] CANDELIER, Raphaël. *Tracking object orientation with image moments*. Online. In: Raphael.candelier.fr. February 11th, 2016. Dostupné z: <https://raphael.candelier.fr/?blog=Image%20Moments>. [cit. 2025-01-03].
- [6] CANDELIER, Raphaël. *Tracking object orientation with image moments*. Online. In: Raphael.candelier.fr. February 11th, 2016. Dostupné z: [https://raphael.candelier.fr/Posts/Image%20Moments/370\\_gauge\\_tracked.mp4](https://raphael.candelier.fr/Posts/Image%20Moments/370_gauge_tracked.mp4). [cit. 2025-01-03].
- [7] PRATAP SINGH, Bhupendra. *The project is based on emerging field Image Processing, In this project A Graphical User Interface has been designed using the software Labwindows. which can process both type of Real time Image Processing and Also many formats of images like .JPG, .DAT, .BMP etc.* Online. In: Researchgate.net. 2015. Dostupné z: [https://www.researchgate.net/figure/Color-image-representation-and-RGB-matrix\\_fig15\\_282798184](https://www.researchgate.net/figure/Color-image-representation-and-RGB-matrix_fig15_282798184). [cit. 2025-01-03]
- [8] *MATLAB list of standart functions*. Online. Mathworks.com. 1994. Dostupné z: [https://www.mathworks.com/help/matlab/referencelist.html?type=function&stid=CRUX\\_topnav](https://www.mathworks.com/help/matlab/referencelist.html?type=function&stid=CRUX_topnav). [cit. 2025-01-05].

- [9] CANDELIER, Raphaël. *Tracking object orientation with image moments*. Online. In: Raphael.candelier.fr. February 11th, 2016. Dostupné z: <https://raphael.candelier.fr/Posts/Image%20Moments/+IM.zip>. [cit. 2025-01-03].
- [10] OMANOVIC, Samir; BUZA, Emir a BESIC, Ingmar. *RGB ratios based skin detection: Fig. 1-4*. Online. In: IEEE Xplore. Opatija, Croatia, 2014, s. 6. ISBN 978-953-233-077-9. Dostupné z: <https://ieeexplore.ieee.org/document/6859777>. [cit. 2025-05-16].
- [11] OMANOVIC, Samir; BUZA, Emir a BESIC, Ingmar. *RGB ratios based skin detection*. Online. In: *IEEE Xplore*. Opatija, Croatia, 2014, s. 6. ISBN 978-953-233-077-9. Dostupné z: <https://ieeexplore.ieee.org/document/6859777>. [cit. 2025-05-16].
- [12] OMANOVIC, Samir; BUZA, Emir a BESIC, Ingmar. *RGB ratios based skin detection: Fig. 5-8*. Online. In: IEEE Xplore. Opatija, Croatia, 2014, s. 6. ISBN 978-953-233-077-9. Dostupné z: <https://ieeexplore.ieee.org/document/6859777>. [cit. 2025-05-16].

## SEZNAM SYMBOLŮ A ZKRATEK

### Zkratky:

FEKT	Fakulta elektrotechniky a komunikačních technologií
VUT	Vysoké učení technické v Brně
Resp.	Respektive
Obj	objekt/y
Bsp	Binary spaced partitioning (Binárně dělený prostor)
tj.	to jako
tzv.	takzvaně
ROI	Region of Interest (Oblast zájmu)
3D	tří dimenzionální, trojrozměrná
uint8	8bitový unsigned integer
RGB	Red, Green, Blue model (červená, zelená, modrá)
col	column (sloupec), proměnná pro indexovou pozici na řádku
popř.	Popřípadě
NaN	Not a Number

### Symbols:

$M_{pq}$	obecné označení dvoudimenzionálního momentu obrazu
$i$	index řádku
$j$	index sloupce
$\bar{x}$	normalizovaná x souřadnice těžiště objektu
$\bar{y}$	normalizovaná y souřadnice těžiště objektu
$\mu'_{2,0}$	parametr ekvivalentní elipsy vypočtený z momentů obrazu
$\theta$	orientace hlavní osy ekvivalentní elipsy
$l$	délka ekvivalentní elipsy
$w$	šířka ekvivalentní elipsy
$d$	délka vedlejší osy ekvivalentní elipsy
$k$	počet vrstev BSP stromu
$\pi$	matematická konstanta pí, zde úhel natočení 180°

## PŘÍLOHY

Přílohami práce jsou zdrojový kód vytvořené aplikace a zdrojové kódy použitých a vytvořených funkcí. Dále jsou součástí testovací fotografie, videozáznam a výsledné fotografie a videozáznamy. Vše je na přiloženém flashdisku rozděleno do adresářů.

- / .....kořenový adresář přiloženého média
  - | BP240333.pdf..... PDF dokument práce
  - | Použité obrázky a videa..... Adresář s používanými obrázky a videi
  - | Výsledné obrázky a videa..... Adresář s vyhodnocenými obrázky a videi
  - | Programy a funkce aplikace ..... Adresář se soubory zdrojového kódu
    - | Snimani\_obrazu\_BP.m..... Main zdrojový kód zpracování obrazu
    - | Snimani\_Videa\_BP.m ..... Main zdrojový kód zpracování videa
    - | draw\_ellipse.m ..... Fce vykreslení elipsy do obrazu
    - | flipOs.m..... Fce otáčení postavy dle osy
    - | get\_ellipse.m ..... Fce získání elipsy z metody momentů
    - | Meritko.m..... Fce pro výpočet měřítka
    - | Mask.m..... Fce prahování barvy kůže
    - | OdectiPozadi.m ..... Fce odečítání pozadí
    - | Odmer.m..... Fce odměření odchylek
    - | VykresOdchyl.m ..... Fce vykreslení odchylek