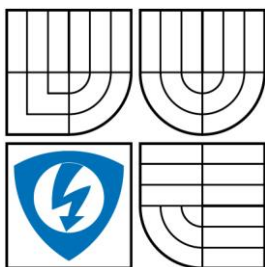


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION**

**IDENTIFIKACE OSOB POMOCÍ BIPEDÁLNÍ LOKOMOCE**  
PERSON IDENTIFICATION BY MEANS OF BIPEDAL LOCOMOTION

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

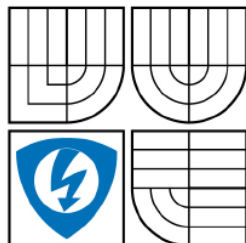
**AUTOR PRÁCE**  
AUTHOR

**Bc. JAKUB KRZYŽANEK**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. Karel Horák, Ph.D.**

BRNO 2010



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí  
techniky

# Diplomová práce

magisterský navazující studijní obor  
**Kybernetika, automatizace a měření**

**Student:** Bc. Krzyžanek Jakub

**ID:** 78565

**Ročník:** 2

**Akademický rok:** 2009/2010

**NÁZEV TÉMATU:**

## Identifikace osob pomocí bipedální lokomoce

### POKYNY PRO VYPRACOVÁNÍ

Nastudujte problematiku bipedální lokomoce a navrhnete a implementujete metodu identifikace osob pomocí charakteristických rysů pohybu. První část práce tvoří robustní segmentační algoritmus pro detekci profilu osoby v komplexní scéně, druhou část pak lokalizace kloubů dolních končetin a klasifikace osoby na základě jejich trajektorií. Výstupem práce je vizuální systém říditelný pomocí navrženého uživatelského rozhraní pracující v reálném čase rozpoznávající identitu určených osob.

### DOPORUŽENÁ LITERATURA:

- [1] HLAVÁČ, Václav, SEDLÁČEK, Miloš. Zpracování signálů a obrazů. Praha : ČVUT, 2005. 255 s. ISBN 80-01-03110-1.
  - [2] SCHLESINGER, Michail I., HLAVÁČ, Václav. Deset přednášek z teorie statistického a strukturního rozpoznávání. Praha : ČVUT, 1999. 521 s. ISBN 80-01-01998-5.
  - [3] SONKA, Milan, HLAVAC, Vaclav, BOYLE, Roger. Image Processing, Analysis and Machine Vision. 3rd edition. Toronto : Thomson, 2008. 829 s. ISBN 978-0-495-08252-1.
  - [4] SVOBODA, Tomas, KYBIC, Jan, HLAVAC, Vaclav. Image Processing, Analysis and Machine Vision : A MATLAB Companion. Toronto : Thomson, 2008. 255 s. ISBN 978-0-495-29595-2.
  - [5] VERNON, David. Machine Vision : Automated Visual Inspection and Robot Vision. Hemel Hempstead : Prentice Hall International (UK) Ltd., 1991. 260 s. ISBN 0-13-543398-3.
- Maximální počet registrací: 9999

**Termín zadání:** 8. 2. 2010

**Termín odevzdání:** 24. 5. 2010

**Vedoucí práce:** Ing. Karel Horák, Ph.D.

**Konzultanti semestrální práce:**

**Prof. Ing. Pavel Jura, CSc.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor semestrální práce nesmí při vytváření semestrální 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 § 152 trestního zákona č. 140/1961 Sb.

Vysoké Učení Technické v Brně

Fakulta elektrotechniky a komunikačních technologií

Ústav automatizace a měřicí techniky

## **Identifikace osob pomocí bipedální lokomoce**

Diplomová práce

Obor: Kybernetika, automatizace a měření

Student: Bc. Jakub Krzyžanek

Vedoucí práce: Ing. Karel Horák, Ph.D.

**Abstrakt:** Práce se zabývá metodami pro nalezení člověka při chůzi v sekvenci snímků a následným hledáním důležitých bodů pro porovnání průběhu jejich pohybu a identifikaci osob. Pro vyhledání siluety člověka jsou použity shlukové metody *k-means* a *mean shift*. Předtím však je pro nalezení přibližného místa hledaného člověka použita metoda rozdílových snímků, konkrétně estimace modelu prostředí, která nám zmenší pole hledání a zkrátí čas segmentace shlukových metod. Práce se při hledání důležitých bodů soustředí na tři místa: střed hlavy a klouby kotníků obou noh. Následně jsou tyto body znázorněny na původní sekvenci snímků a porovnány se skutečnými místy výskytu středu hlavy a kotníků, které uživatel předem vyznačil. Práce se taky zabývá porovnáním získaných průběhů pohybů těchto bodů a identifikaci osob, kterým patří snímaná chůze. Ke konci jsou zkoumány problematické situace, které se vyskytly v průběhu řešení práce.

Výsledkem práce je algoritmus, který v sekvenci snímků (resp. videa) najde pohybujícího se člověka a vyhledá důležité body (střed hlavy a kotníky) pro porovnání a identifikaci osoby, které patří snímaná chůze.

**Klíčová slova:** rozpoznávání osob, bipedální lokomoce, chůze, segmentace obrazu, identifikace člověka, shlukové metody.

Brno University of Technology

The Faculty of Electrical Engineering and Communication

Department of Control, Measurement and Instrumentation

## **Person identification by means of bipedal locomotion**

Thesis

Specialisation of study: Cybernetics, Control and Measurement

Student: Bc. Jakub Krzyžanek

Supervisor: Ing. Karel Horák, Ph.D.

**Abstract:** The aim of this thesis is to recognize a walking person in a sequence of images by defining his or her reference points to compare the course of their movement and then to identify the scanned person. Methods „k-means“ and „mean shift“ are used to obtain the silhouette of the person. However “environment model estimation” method is used here before those mentioned above. It is a type of a difference method and it helps to specify the scanning area and shortens the time of segmentation. During the search for the reference points the thesis focuses on three areas: the centre of the head and both ankle joints. Those points are later determined on the previous image sequence and compared with the real locations of the centre of the head and ankle joints marked by the user. The thesis also focuses on comparing the movement courses of those points and tries to identify the people whose walks are being scanned. Problematic situations which occurred during the whole process are analyzed in the end.

The result of the thesis is an algorithm which can locate a moving person in an image sequence (video) and determine the reference points (centre of the head and ankles) to compare them and identify the scanned person.

**Key words:** person identification, gait recognition, bipedal locomotion, image segmentation, human recognition, clustering method.

### **Bibliografická citace**

KRZYŽANEK, J. *Identifikace osob pomocí bipedální lokomoce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 65s. Vedoucí diplomové práce Ing. Karel Horák, Ph.D.

## **Prohlášení**

„Prohlašuji, že svou diplomovou práci na téma *Identifikace osob pomocí bipedální lokomoce* jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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 § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: 24. května 2010

.....

podpis autora

## **Poděkování**

Děkuji vedoucímu diplomové práce Ing. Karlu Horákovi Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Dále bych chtěl poděkovat Ing. Ondřeji Pražákovi za poskytnutí nahrávek s jeho chůzí a chůzí jeho kolegů.

V Brně dne: 24. května 2010

.....

podpis autora

## OBSAH

<b>1. Úvod .....</b>	<b>9</b>
<b>2. Pohyby v kloubech člověka při chůzi.....</b>	<b>10</b>
<b>3. Rozvržení scény .....</b>	<b>13</b>
<b>4. Segmentace .....</b>	<b>15</b>
4.1 Možnost předzpracování .....	15
4.2 Metody segmentace sekvence snímků .....	17
4.2.1 Optický tok .....	17
4.2.2 Detekce významných bodů .....	18
4.2.3 Rozdílové metody .....	19
4.2.4 Estimace prostředí .....	19
4.2.5 Shlukové metody.....	21
4.2.5.1 Metoda k-means.....	22
4.2.5.2 Metoda mean shift.....	26
4.2.6 Detekce stínů.....	30
<b>5. Segmentace v programu Matlab .....</b>	<b>34</b>
5.1 Vzhled programu.....	34
5.2 Hlavní cyklus rozpoznávání .....	35
5.3 Segmentace snímku .....	36
5.4 Porovnání metod.....	42
<b>6. Nalezení důležitých bodů v siluetě .....</b>	<b>45</b>
6.1 Distanční transformace siluety.....	45
6.2 Nalezení regionálních maxim .....	46
6.3 Nalezení kotníků a hlavy .....	47
6.4 Záznam pozice nohou .....	48
6.5 Interpolace bodů kotníků (situace zakrývající se nohou).....	49
6.6 Porovnání nalezených bodů se skutečnými .....	50

<b>7.</b>	<b>Rozpoznávání osob pomocí získaných průběhů .....</b>	<b>52</b>
7.1	Volba získání dat pro porovnávání.....	52
7.2	Standardizace dat .....	53
7.3	Porovnávání průběhů a identifikace osob .....	54
<b>8.</b>	<b>Problémy při získávání obrysu .....</b>	<b>56</b>
8.1	Nesprávné nasnímání scény .....	56
8.2	Nevhodný výběr oblečení a scény snímání.....	58
8.3	Problémy metod.....	59
8.4	Problémy při hledání kloubů .....	60
<b>9.</b>	<b>Závěr .....</b>	<b>62</b>
<b>10.</b>	<b>Literatura .....</b>	<b>64</b>



## Seznam obrázků:

OBRÁZEK 2.1: ZOBRAZENÍ LIDSKÉ CHŮZE (JEDNOSTRANNÁ OPORA) [1].....	10
OBRÁZEK 2.2: ZNÁZORNĚNÍ OHYBU (FLEXE A EXTENZE) KLOUBU KYČELNÍHO [1].....	12
OBRÁZEK 2.3: ZNÁZORNĚNÍ OHYBU KLOUBU KOLENNÍHO [1].....	12
OBRÁZEK 2.4: ZNÁZORNĚNÍ OHYBU KLOUBU HLEZENNÍHO [1].....	12
OBRÁZEK 3.1: ROZLOŽENÍ SCÉNY A ZOBRAZENÍ JEDNOTLIVÝCH VZDÁLENOSTÍ.....	13
OBRÁZEK 3.2: RŮZNÉ ZORNÉ ÚHLY PRO RŮZNÉ OHNISKOVÉ VZDÁLENOSTI [17].....	14
OBRÁZEK 4.1: CHARAKTERISTIKA TRANSFORMACE JASOVÉ STUPNICE.....	15
OBRÁZEK 4.2: POROVNÁNÍ OBRÁZKŮ PŘED A PO JASOVÉ TRANSFORMACI.....	16
OBRÁZEK 4.3: POROVNÁNÍ ROZDÍLOVÉHO SNÍMKU PŘED A PO TRANSFORMACI JASOVÉ STUPNICE.....	17
OBRÁZEK 4.4: ZNÁZORNĚNÍ POSTUPU ESTIMACE PROSTŘEDÍ (POUŽITÝ PRÁH $\varepsilon = 35$ )	20
OBRÁZEK 4.5: ROZDĚLENÍ OBRÁZKU NA BODY PŘEDSTAVUJÍCÍ VÝSKYT BAREV [8]....	21
OBRÁZEK 4.6: NÁZORNÉ PŘEDVEDENÍ ALGORITMU POMOCÍ OBRÁZKŮ: A) UMÍSTĚNÍ K VZORKŮ DO PROSTORU. B) + C) PŘÍRAZOVÁNÍ BODŮ DO JEDNOTLIVÝCH STŘEDŮ D) KONEČNÉ ROZPOLOŽENÍ SHLUKŮ A K NIM PŘÍRAZENÝCH BODŮ [6] .....	24
OBRÁZEK 4.7: SEGMENTACE OBJEKTU POMOCÍ K-MEANS S ODLIŠNÝM POČTEM SHLUKŮ .....	25
OBRÁZEK 4.8: GRAFICKÁ UKÁZKA VÝPOČTU MAXIMÁLNÍ HUSTOTY SHLUKŮ [7].....	27
OBRÁZEK 4.9: ROZDĚLENÍ BODŮ V OBRÁZKU NA SHLUKY [7].....	28
OBRÁZEK 4.10: ROZDĚLENÍ OBRÁZKU NA OBJEKTY POMOCÍ MEAN SHIFT (VELIKOST OKOLÍ = 35).....	29
OBRÁZEK 4.11: ROZDĚLENÍ NA OBJEKTY POMOCÍ 5-ROZMĚRNÉHO MEAN SHIFT (VEL.OK. = 35).....	29
OBRÁZEK 4.12: BLOKOVÉ SCHÉMA ALGORITMU PRO DETEKCI STÍNŮ .....	30
OBRÁZEK 4.13: ZNÁZORNĚNÍ DETEKCE STÍNŮ KROK PO KROKU.....	33
OBRÁZEK 5.1: UŽIVATELSKÉ ROZHRANÍ PROGRAMU .....	34
OBRÁZEK 5.2: BLOKOVÉ SCHÉMA HLAVNÍHO CYKLU PROGRAMU .....	35
OBRÁZEK 5.3: A) PRŮBĚH ZMĚNY SOUČTU BODŮ OBJEKTU, B) ZNÁZORNĚNÍ POZIC CHŮZE PŘI MIN. A MAX. SOUČTU BODŮ .....	36
OBRÁZEK 5.4: BLOKOVÉ SCHÉMA SEGMENTACE SNÍMKU .....	37
OBRÁZEK 5.5: DETEKOVANÉ STÍNY (BÍLÁ BARVA) PŘI CHŮZI.....	38
OBRÁZEK 5.6: PŘÍKLAD SEGMENTACE OBRÁZKU METODOU K-MEANS (POČET SHLUKŮ = 5) V RŮZNÝCH POLOHÁCH CHŮZE.....	38
OBRÁZEK 5.7: PŘÍKLAD SEGMENTACE OBRÁZKU METODOU MEAN SHIFT (VELIKOST OKOLÍ = 40) .....	39

OBRÁZEK 5.8: ODSTRANĚNÍ NEVHODNÝCH SHLUKŮ BODŮ .....	40
OBRÁZEK 5.9: OBRÁZEK CHŮZE OBJEKTU PO SEGMENTACI.....	40
OBRÁZEK 5.10: UKÁZKA MORFOLOGICKÉ OPERACE NA SILUETĚ A NA JEDNODUCHÉM OBRÁZKU .....	41
OBRÁZEK 5.11: SNÍMKY SILUETY CHŮZE OBJEKTU ZAHRNUTY V JEDNOM OBRÁZKU...42	
OBRÁZEK 5.12: POROVNÁNÍ VÝSLEDKŮ SEGMENTACE NA JEDNOM Z OBRÁZKŮ RŮZNÝMI METODAMI: A) PŮVODNÍ OBRÁZEK, B) ROZDÍLOVÉ SNÍMKY, C) ROZDÍLOVÉ SNÍMKY S TRANSFORMACÍ JASOVÉ STUPNICE, D) METODA K-MEAN S TRANSFORMACE JASOVÉ STUPNICE, E) METODA K-MEAN S TRANSFORMACE JASOVÉ STUPNICE A DETEKČÍ STÍNŮ, F) METODA MEAN SHIFT S TRANSFORMACE JASOVÉ STUPNICE, G) METODA MEAN SHIFT S TRANSFORMACE JASOVÉ STUPNICE A DETEKČÍ STÍNŮ .....	44
OBRÁZEK 6.1: BLOKOVÉ SCHÉMA HLEDÁNÍ DŮLEŽITÝCH BODŮ V SILUETĚ .....	45
OBRÁZEK 6.2: A) SILUETA, B) DISTANČNÍ TRANSFORMACE, C) REGIONÁLNÍ MAXIMA .46	
OBRÁZEK 6.3: OBLASTI HLEDÁNÍ KOTNÍKŮ.....	47
OBRÁZEK 6.4: OBRÁZEK ZNÁZORŇUJÍCÍ VERTIKÁLNÍ AMPLITUDOVOU PROJEKCI V PROSTORU OHRANIČENÉM ČERVENOU ČAROU.....	49
OBRÁZEK 6.5: UKÁZKA INTERPOLACE PRŮBĚHU POZICE KOTNÍKU .....	50
OBRÁZEK 6.6: OBRÁZEK S OHRANIČENÍM OBJEKTU A VYZNAČENÍM MÍST KOTNÍKŮ A HLAVY .....	50
OBRÁZEK 6.7: UKÁZKA POROVNÁNÍ NALEZENÝCH A OZNAČENÝCH MÍST KOTNÍKŮ A HLAVY .....	51
OBRÁZEK 7.1: MASKY PRO ZÍSKÁNÍ PRŮBĚHU ZE SILUETY [12] .....	52
OBRÁZEK 8.1: ROZMAZÁNÍ ČÁSTI TĚLA PŘI DLOUHÉ UZÁVĚRCE FOTOAPARÁTU .....	56
OBRÁZEK 8.2: POROVNÁNÍ METOD PŘI NEVHODNÉ BARVĚ POZADÍ.....	57
OBRÁZEK 8.3: POHYB JINÝCH OBJEKTŮ VE SCÉNĚ NESOUVISEJÍCÍCH S HLEDANÝM OBJEKTEM .....	58
OBRÁZEK 8.4: VÝSLEDEK ROZDÍLOVÉHO SNÍMKU PŘI PODOBNÉ BARVĚ POZADÍ A OBLEČENÍ.....	59
OBRÁZEK 8.5: NEVHODNĚ NASTAVENÉ PARAMETRY METOD A) MEAN SHIFT, B) K- MEANS C) DETEKCE STÍNŮ .....	60

## 1. Úvod

Vymyslelo se už velké množství biometrických metod rozpoznávání člověka, z nichž nejznámější jsou: otisky prstů, charakteristika písma, rozpoznávání podle oční sítnice, podle obličeje apod. [15] Všechny tyto metody však mají jednu společnou negativní vlastnost. Jsou to invazivní metody nebo metody na velmi malou vzdálenost. Proto je snaha vymyslet účinný algoritmus, který by mohl detekovat konkrétní osobu na větší vzdálenost (konkrétněji jednotky až desítky metrů), a proto se v této práci zabývám identifikací osob na základě bipedální chůze, která může čerpat ze sekvence snímků zhotovených na větší vzdálenost.

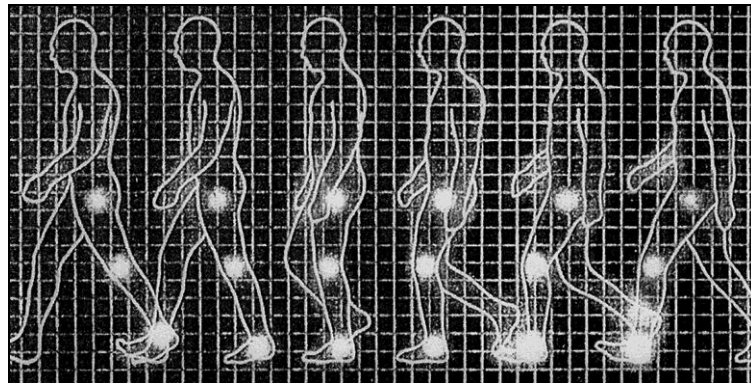
Další aspekt hrající do karet této metodě je fakt, že v poslední době se do veřejných prostorů instaluje velké množství průmyslových a soukromých kamer, které snímají nespočet lidí. Pak vykradeli dobře zamaskovaný zloděj banku nebo jinou nemovitost a nafilmují ho přitom bezpečnostní kamery, může být následně vystopován zjištěním dynamiky chůze z dalších kamer, například v jiném městě, daleko od kriminálního činu.

Cílem této práce je tedy navrhnout vhodný algoritmus, který ze sekvence snímků chůze člověka získá siluetu hledaného objektu. Z té pak vyhledá rozmístění kloubů (resp. kotníků a hlavy) člověka, s jejichž pohybů a úhlů mezi nimi svírajícími můžeme vytvořit průběh a porovnávat ho s předem získanou databází.

Práce je rozvržena podle kapitol následovně: v druhé je popsána problematika chůze z fyziologického hlediska, třetí kapitola popisuje rozvržení scény pro snímání chůze fotoaparátem nebo kamerou. Ve čtvrté jsou pak charakterizovány metody pro segmentaci a předzpracování obrazu použité nebo vyzkoušené v této práci. V páté kapitole je popsán program vytvořený pro vizualizaci a zpracování sekvence snímků a následně i předvedeny příklady znázorňující výsledky segmentace. V šesté je představena metoda pro nalezení důležitých bodů (kotníků a hlavy) pro pozdější porovnávání chůze. Sedmá kapitola naznačuje metodiku porovnávání průběhů a identifikace osob. Poslední osmá kapitola pojednává o problémech vyskytujících se v souvislosti se snímáním, předzpracováním a segmentací obrazu.

## 2. Pohyby v kloubech člověka při chůzi

„Podle Hněvkovského je chůze rytmický pohyb vykonávaný dolními končetinami a provázený souhyby všech částí těla. Charakteristické při chůzi je to, že každý krok má jeden okamžik, kdy obě nohy spočívají na zemi (dvojitá opora). Mezi oběma okamžiky dvojitá opora spočívá tělo na jedné končetině a druhá končetina přitom kmitá (osciluje) dopředu.“ [1]



**Obrázek 2.1:** Zobrazení lidské chůze (jednostranná opora) [1]

Chůze je vysoce automatizovaný pohybový projev, jehož charakter závisí na struktuře těla, jeho proporcích a hmotnosti právě tak, jako na kvalitě proprioceptivní informace z periferie a na kvalitě regulačních centrálně nervových mechanismů. Přesto, že je chůze vrozená, každý jedinec se v průběhu svého vývoje učí chodit sám a vytváří si osobní charakter chůze související s držením těla. Nejsou proto na světě dva jedinci, kteří by měli absolutně identickou chůzi. Jde o tak složitý pohyb, že nemáme přesnou metodu, která by dokonale vystihla všechny její složky. [1]

Rozlišujeme různé druhy chůze: vpřed, vzad, stranou (přisunem nebo překračováním), po schodech (nahoru a dolů), v terénu, při překračování překážek, při vstupu a výstupu (po schodech, do dopravních prostředků apod.). [1]

Pro rozpoznání člověka je důležité pozorovat určité pohyby při chůzi. Můžou to být například tyto momenty [1]:

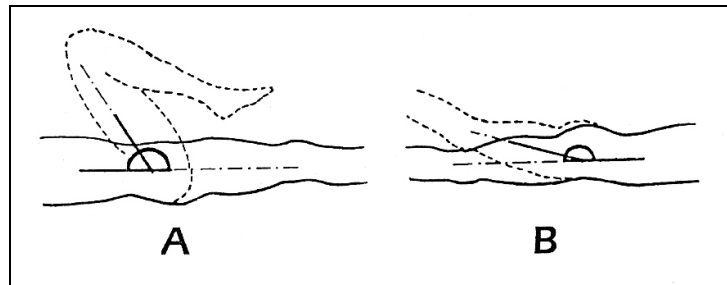
- Rytmus, pravidelnost chůze – porucha rytmu může být např. při bolestech, kdy zdravá končetina dělá rychlejší krok než postižená.
- Délka kroku – příčinou nesouměrné délky kroku může být délka končetiny, bolesti, rozsah kloubní pohyblivosti, svalová síla apod.
- Osově postavení dolní končetiny – vliv na zatěžování nosných kloubů.
- Postavení nohy a její odvíjení od podložky – první dotyk podložky je patou, dále se přenáší váha přes zevní okraj nohy na špičku (jednostranná opora).
- Pohyb těžiště – švihovou fází druhé nohy se posune těžiště těla vpřed a tím i váha těla.
- Souhyby horních končetin, hlavy a trupu – pohyb paží je doprovázen mírnou rotací trupu. Při chůzi se střídá levá paže s pravou nohou a opačně.
- Svalová aktivita – snížením svalové síly se výrazně projeví charakter chůze (omezený pohyb v kloubech, kontraktury – zkrácení svalů).
- Stabilita při chůzi – je schopnost udržovat rovnováhu, přizpůsobovat se povrchu terénu, změnám rychlosti chůze a překonávání překážek.
- Používání pomůcek – pomůcky jako třeba hůl, berle, chodítko, dlaha, ortéza, protéza, ortopedická obuv.

Z pohledu poznávání pohybu při chůzi je dobré znát úhly maximálních ohybů kloubů. Protože poloha jednotlivých částí těla pro poznávání dynamiky chůze je závislá spíše na dolní polovině trupu, zaměřím se na popsání maximálního ohnutí těchto kloubů:

- Kloub kyčelní [1]: flexe<sup>1</sup> – rozsah pohybu je až 130 stupňů s ohnutým kolenem a 90 stupňů s nataženým kolenem (*Obrázek 2.2A*).  
Extenze<sup>2</sup> – rozsah pohybu je do 15 stupňů (*Obrázek 2.2B*).

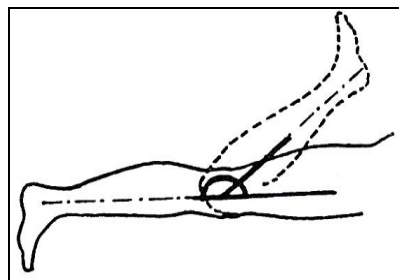
<sup>1</sup> Ohýbání - ve smyslu zkracování dané části těla.

<sup>2</sup> Zpětný pohyb k flexi, natahování - ve smyslu prodlužování dané části těla.



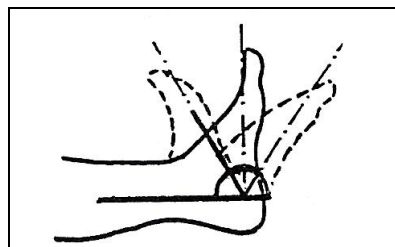
**Obrázek 2.2:** Znárodnění ohybu (flexe a extenze) kloubu kyčelního [1]

- Kloub kolenní [1]: flexe – rozsah pohybu je do 130 až 150 stupňů podle rozvoje svalstva (Obrázek 2.3).  
Extenze – výchozí nulová pozice (někdy do 10 stupňů).



**Obrázek 2.3:** Znárodnění ohybu kloubu kolenního [1]

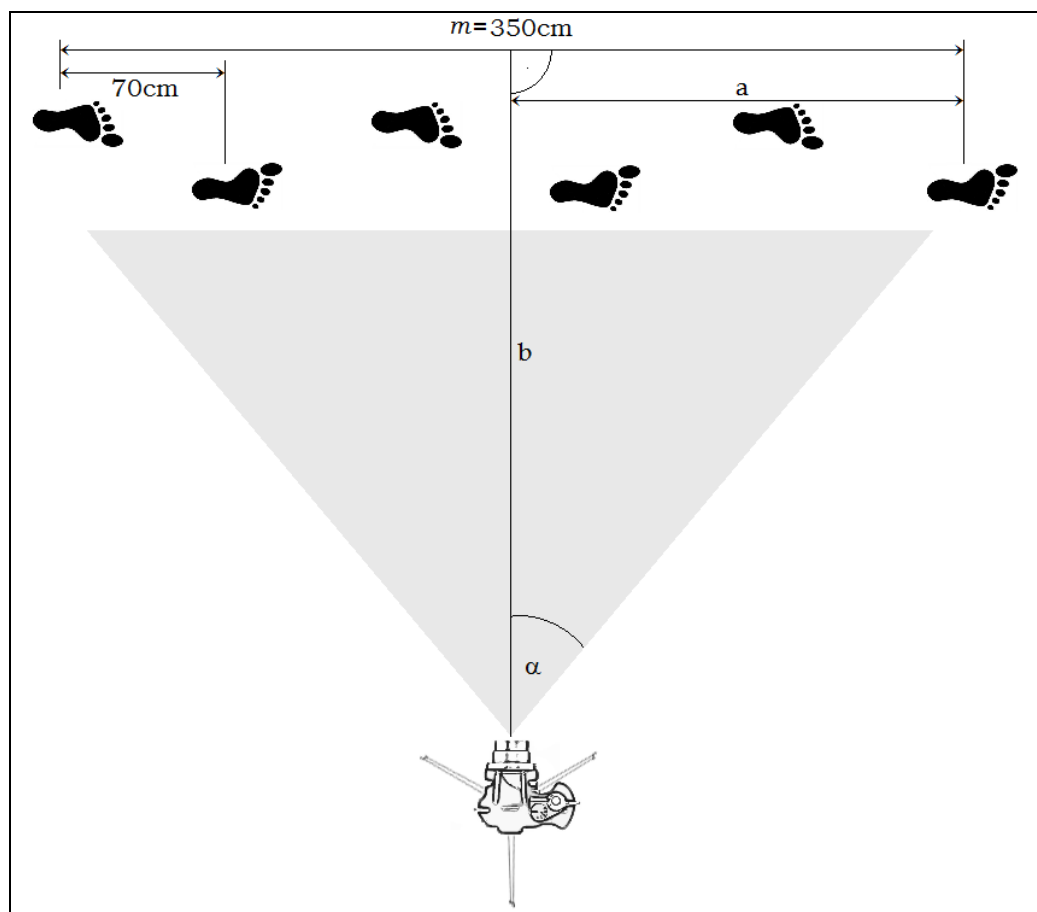
- Kloub hlezenní [1]: flexe – rozsah pohybu je do 50 stupňů (Obrázek 2.4).  
extenze – rozsah pohybu je do 15 až 20 stupňů.



**Obrázek 2.4:** Znárodnění ohybu kloubu hlezenního [1]

### 3. Rozvržení scény

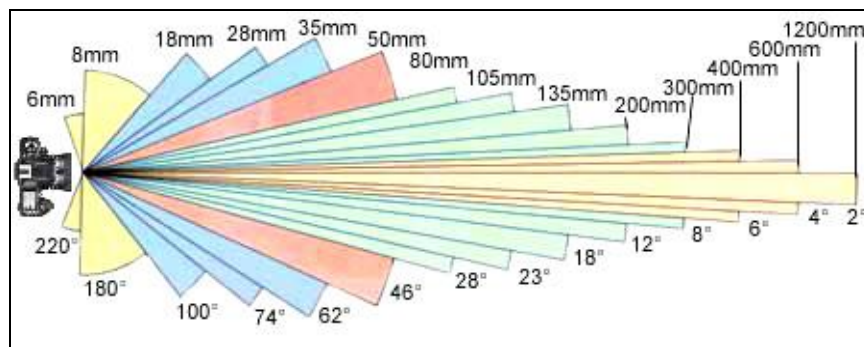
Průměrná délka jednoho kroku dospělého člověka je cca 70 cm. V naší práci budeme předpokládat, že pro rozpoznávání člověka nám bude stačit nasnímat kompletní dva kroky za sebou. Jelikož se může stát, že začneme snímat chůzi v polovině načatého kroku, musíme nasnímat minimálně tři kroky. Protože čím je člověk větší, má delší nohy a tím taky delší krok, a proto budeme předpokládat, že pro nasnímání potřebného počtu kroků nám bude stačit nasnímat pět kroků dospělého člověka. Což znamená minimální délku snímané scény:  $5 \cdot 70 \text{ cm} = 350 \text{ cm}$ .



**Obrázek 3.1:** Rozložení scény a zobrazení jednotlivých vzdáleností.

V dnešní době je možno zakoupit velmi mnoho různých fotoaparátů, které disponují nahráváním videa. Tyto fotoaparáty a kamery se liší velikostí snímacího

čipu, ohniskovou vzdáleností, rychlosti uzávěrky, velikostí clony a mnoho dalšími parametry. Pokud bychom chtěli snímat chůzi člověka konkrétním přístrojem, který jsme pro tento účel zakoupili, musíme si o něm zjistit určité specifikace. Grafické znázornění různého zorného úhlu pro různé ohniskové vzdálenosti můžeme vidět na obrázku 3.2.



**Obrázek 3.2:** Různé zorné úhly pro různé ohniskové vzdálenosti [17]

Informace potřebné pro určení vzdálenosti snímání od připravené trajektorie chůze člověka jsou ohnisková vzdálenost  $f$  a velikost snímacího čipu  $d$  (úhlopříčka) [18], z nichž vypočítáme zorný úhel  $\omega$  pomocí vzorečku:

$$\operatorname{tg} \frac{\omega}{2} = \frac{d}{2f} \rightarrow \omega = 2 \cdot \operatorname{arctg} \frac{d}{2f} \dots \alpha = \frac{\omega}{2} \quad (1)$$

ze kterého vypočítáme vzdálenost  $b$  od trajektorie chůze  $m$  pomocí goniometrické funkce:

$$m = 2 \cdot a \rightarrow b = a \cdot \operatorname{cotg} \alpha \quad (2)$$

Pokud jsme správně provedli výpočet a získali tak vzdálenost, můžeme začít natáčet videa z chůzí různých lidí. Musíme však nastavit délku snímání jednoho snímku optimálně vůči nasvětlení scény tak, aby se jednotlivé části objektu při pohybu nerozmazávaly. Vytvoříme tím ideální podmínky pro pozdější segmentaci obrázku.



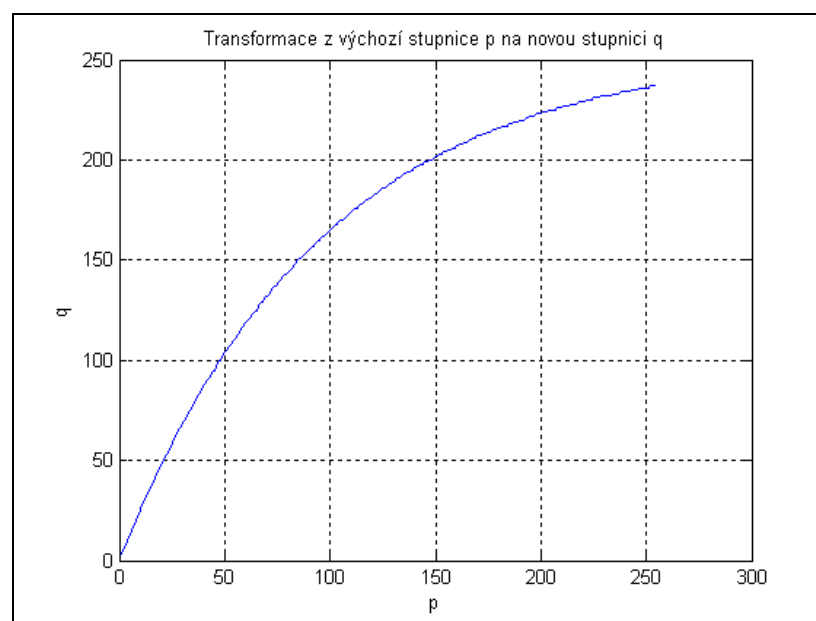
## 4. Segmentace

### 4.1 Možnost předzpracování

Při snímání pozadí s pohybujícím se objektem se v obrazu mohou vyskytovat jasně temnější místa, kde rozdíl mezi objektem a pozadím bude velmi malý. Naopak v jiných jasnějších místech se může stát, že rozdíl mezi objektem a pozadím je velký. Pro rozpoznání objektu v obou místech potřebujeme odlišnou optimální velikost prahu, proto jsem zkusil metodu transformace jasové stupnice jako předzpracování před segmentací obrazu rozdílovými snímky nebo estimací prostředí, kde velmi záleží na správném určení prahu. Tvar charakteristiky jasové transformace jsem určil empiricky jako rovnicí, ze které vypočteme nové hodnoty jasové stupnice  $q$  ze staré jasové stupnice  $p$ :

$$q = 255 \cdot \left( 1 - e^{-\frac{p}{96}} \right) \quad (3)$$

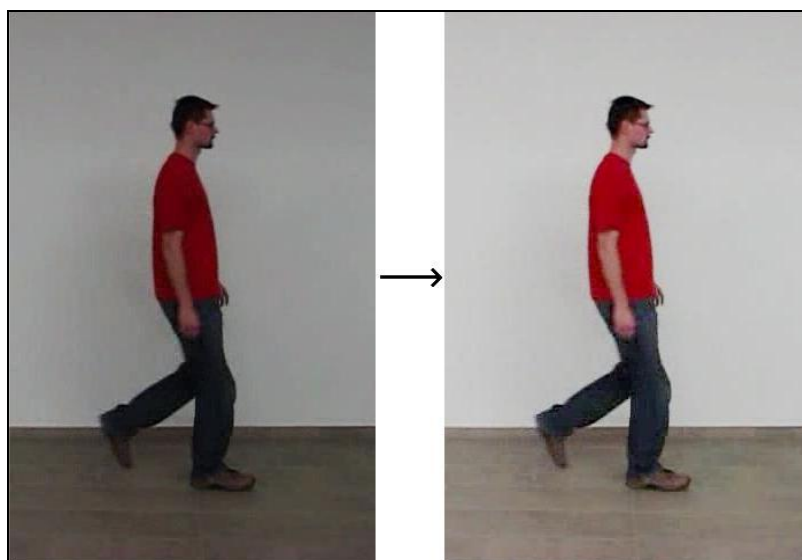
Průběh této transformace jasové stupnice můžeme znázorněn na *obrázku 4.1*.



**Obrázek 4.1:** Charakteristika transformace jasové stupnice

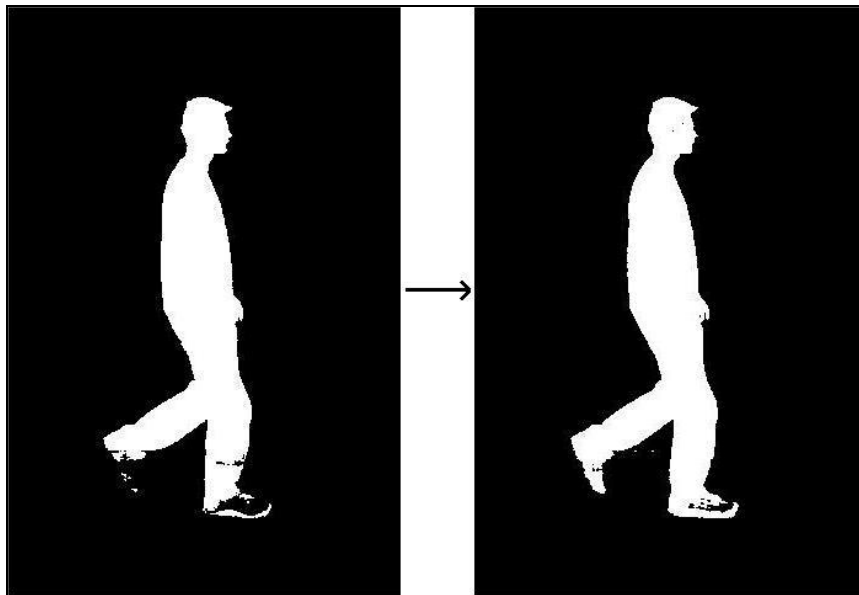
U snímků bez transformace jasové stupnice se na rozdílových obrázcích objevuje při použití nižšího prahu nechtěný šum (tvořený např. malými jasovými změnami prostředí, stínem objektu) a při použití vyššího prahu se zase ztrácejí některé části objektu (při malém rozdílu hodnoty jasu barev mezi objektem a pozadím).

Při porovnání snímků se starou a novou jasovou stupnicí na *obrázku 4.2* si můžeme povšimnout míst, kde je v levém snímku v některých místech velmi malý rozdíl mezi pozadím a objektem (například v oblasti bot) a v pravém po transformaci se tento rozdíl zvětšil. Přitom v jasnějších oblastech se rozdíl taky zvětšil, ale méně než v temnějších oblastech.



**Obrázek 4.2:** Porovnání obrázků před a po jasové transformaci

Tímto krokem jsme dosáhli toho, že při segmentaci metodou estimace modelu prostředí můžeme nastavit konstantní práh, který je méně náchylný na šum a lépe rozpozná objekt od pozadí ve tmavších místech. Porovnání segmentace pomocí rozdílového snímku s transformací a bez můžeme shlédnout na *obrázku 4.3*.



**Obrázek 4.3:** Porovnání rozdílového snímku před a po transformaci jasové stupnice

## 4.2 Metody segmentace sekvence snímků

Před popsáním pohybujícího se objektu musíme najít jeho hranice nebo nějaké vztažné body, které by byly rozmístěny na jeho těle a přesně by označovaly souřadnice umístění v prostoru snímání kamery. Pro tento úkol existují algoritmy s různou rychlostí výpočtu a výsledným výstupním obrazem, z nichž nejznámější jsou:

- Optický tok
- Detekce významných bodů
- Rozdílové metody
  - Estimace modelu prostředí
- Shlukové metody
  - Metoda k-means
  - Metoda mean shift

### 4.2.1 Optický tok

Metoda výpočtu optického toku zachycuje všechny změny obrazu za časový interval  $dt$ . Je založena na hledání stejných jasových hodnot v jistém okolí výskytu

pohybu objektu ve dvou po sobě jdoucích obrazech. Po správném nalezení souvisejících bodů pak každému ve výsledném obrazu odpovídá dvojrozměrný vektor rychlosti, vypovídající o směru a velikosti rychlosti pohybu v daném místě obrazu. [2]

#### 4.2.2 Detekce významných bodů

Tuto metodu lze použít v případech, kdy časové intervaly mezi snímky nelze považovat za velmi malé vzhledem k časovým konstantám určujícím pohyb v obrazu. Aplikační postup metody se skládá ze dvou kroků. Prvním je samotné nalezení významných bodů obrazu (např. hraniční body, jasově odlišné plochy apod.) a druhým je nalezení korespondence těchto bodů mezi jednotlivými obrazy. [2]

Ve všech obrazech posloupnosti (sekvence) je třeba nalézt taková místa, která jsou nějakým způsobem významná, tj. jsou co nejméně podobná svému okolí, představují vrcholy či hranice objektů. Pro tento účel lze použít hranové detektory nebo Moravcův operátor, který určuje významnost bodu na základě co největší odlišnosti malé části obrazu od jejího okolí a je definován takto [2]:

$$f(i, j) = \frac{1}{8} \sum_{k=i-1}^{k=i+1} \sum_{e=j-1}^{e=j+1} |g(k, e) - g(i, j)| \quad (4)$$

Tato rovnice popisuje výpočet specifické hodnoty Moravcova operátoru pro každý pixel v obrazu tak, že pomocí čtvercového okna předem zadané velikosti  $N \times N$  (v případě rovnice 4. velikosti okna  $3 \times 3$ ) vypočítáme součet rozdílů jasů sousedních pixelů s prostředním pixelem. Náročnost výpočtu se tedy bude kvadraticky zvětšovat z velikostí okna a velikostí obrazu. Je taky zjevné, že Moravcův operátor není invariantní vůči rotaci a je velmi náchylný na šum. Tyto nedostatky do určité míry odstraňuje Harris/Plesseyův operátor, který k nalezení významných bodů využívá algoritmus založený na výpočtu gradientu. Operátor je sice invariantní vůči natočení, ale bohužel je taky velmi citlivý na šum a je výpočetně velmi náročný kvůli výpočtu konvoluce s Gaussovským oknem, kterou potřebuje pro nalezení významných bodů v obrazu [14]. Výpočet nalezení významných bodů je pouze polovina úlohy. Druhou polovinu představuje určení korespondence těchto bodů mezi jednotlivými snímky a určením pohybu objektu. Tato část se provádí postupným srovnáváním dvojic všech

nalezených bodů ve dvou po sobě následujících snímcích a určováním nejpravděpodobnější věrohodnosti korespondence.

Metody optický tok a detekce významných bodů jsou velmi náročné na výpočet, protože musí přepočítávat hodnoty okolí každého bodu ve snímku. Proto taky jsou pomalé, a tedy nepoužitelné v real-time operacích.

#### 4.2.3 Rozdílové metody

Metoda rozdílových snímků, založená na zjištění rozdílů obrazů pořízených v různých časových okamžicích, je jednodušší na výpočet a tudíž rychlejší. Při snímání nepohybující se kamerou dynamického pohybu snímáme obrazy  $f_1(x, y)$ ,  $f_2(x, y)$  atd. Pokud rozdíl dvou po sobě jdoucích snímků je větší než předem nastavený práh  $\varepsilon$ , dáme na příslušnou pozici snímku hodnotu 1. V odlišném případě zaneseme na toto místo hodnotu 0. Výsledný binární obraz lze pak definovat [2]:

$$d(x, y) = \begin{cases} 0 & |f_1(x, y) - f_2(x, y)| < \varepsilon \\ 1 & |f_1(x, y) - f_2(x, y)| \geq \varepsilon \end{cases} \quad (5)$$

Rozdílová metoda může být:

- Jednosměrná (výsledný binární snímek je tvořen pouze kladnými změnami jasových hodnot oproti následujícímu snímku)
- Obousměrná (výsledný binární snímek je tvořen kladnými i zápornými body v absolutní hodnotě rozdílu dvou po sobě následujících snímků) [2]

Ačkoli rozdílové snímky jsou jednoduché na výpočet a dávají informaci o pohybu, mají několik vad. Jednou z nich je například situace, kdy se jednobarevný objekt pohybuje pomalu v určitém směru. Výsledný rozdílový snímek zachytí jenom malý objem přední a zadní části objektu. Tyto části (protože jsou velmi malé) však mylně můžeme označit za šum prostředí, a v obrázku nám nevynikne žádný pohyb.

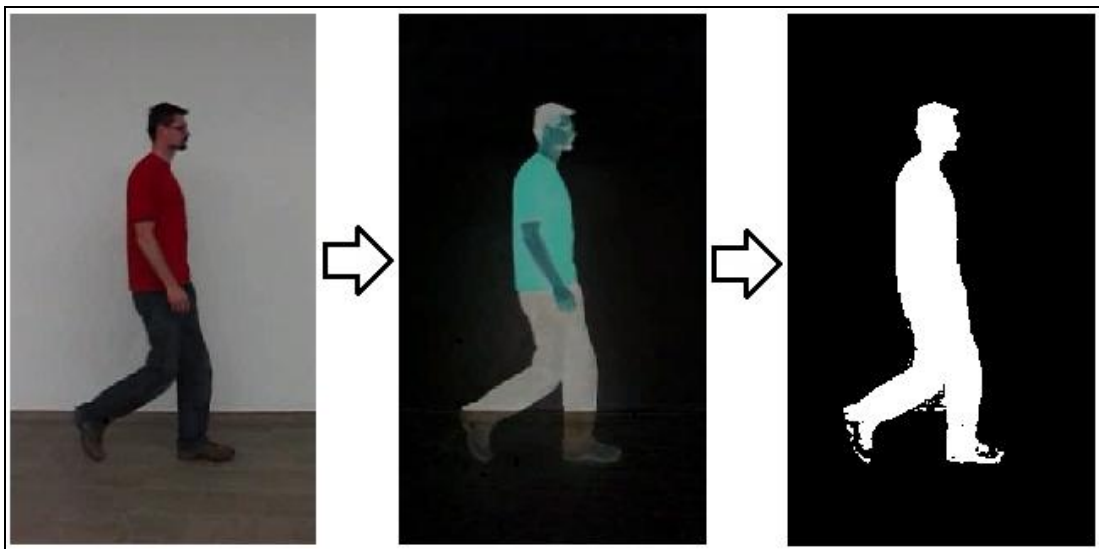
#### 4.2.4 Estimace prostředí

V tomto případě je lepší použít metodu estimace modelu, tzn. sestavení modelu pozadí z několika po sobě následujících snímků, na kterých není žádný pohyb, nebo označit první snímek modelem. Ve druhém případě se už na prvním snímku může

vyskytnout pohyb objektu nebo také nechtěný šum. Ve výsledném obrázku by se vyskytly objekty, které chybně označíme jako pohyb. Proto je vhodná první z vyjmenovaných situací - vypočítat průměrný snímek. Označíme-li model  $b(x, y)$ , můžeme pro každý obrazový bod snímku  $f(x, y)$  zapsat rovnici [2]:

$$d(x, y) = \begin{cases} 0 & |f(x, y) - b(x, y)| < \varepsilon \\ 1 & |f(x, y) - b(x, y)| \geq \varepsilon \end{cases} \quad (6)$$

Výsledný binární obraz pak znázorňuje všechny body (včetně šumu a stínu) jejichž rozdíl mezi aktuálním snímkem a vytvořeným modelem je větší než předem stanovený práh  $\varepsilon$ . Vytvoření rozdílu snímku a modelu prostředí je znázorněno na *obrázku 4.4*, kde také vidíme výsledný vyprahovaný obrázek s nalezenými místy pohybu v obraze.



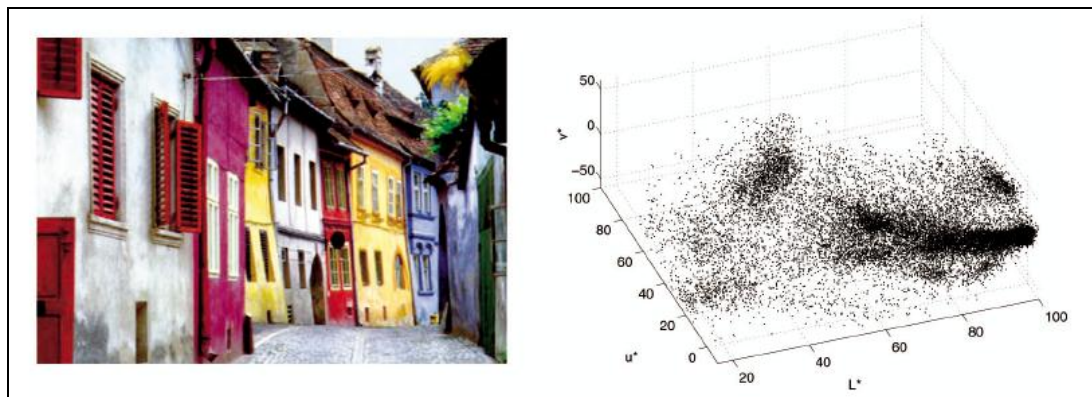
**Obrázek 4.4:** Znáornění postupu estimace prostředí (použitý práh  $\varepsilon = 35$ )

Na rozdíl od předem popsaných rozdílových metod umožňuje nejenom pozorovat krajní místa objektu kde došlo k pohybu, ale také celý hledaný objekt bez míst, ve kterých už se objekt nenachází, a které by jako pohybující se součást objektu označila předchozí metoda rozdílových snímků.

#### 4.2.5 Shlukové metody

Hlavním záměrem shlukových metod je rozdělení objektů, které jsou si něčím podobné, do několika skupin. Pak tyto skupiny seskupují objekty, které jsou podobné mezi sebou a jsou odlišné od objektů v jiných skupinách. Při zaměření na náš problém jsou těmito objekty jednotlivé body v obrazu s odlišnými barvami a různou vzdáleností od bodů se stejnou barvou, a skupiny jsou shluky těchto bodů odlišující se právě barvou, popřípadě jasnou pixelu nebo vzdáleností od pixelu stejné barvy.

Cílem těchto metod je tedy odlišit body s podobnými vlastnostmi od bodů od nich se lišících pomocí kritéria, které zajistí optimální oddělení těchto bodů do shluků s hlediska potřeb uživatele (v našem případě je třeba odlišit pohybující se objekt od pozadí). Na *obrázku 4.5* můžeme pozorovat rozložení levého obrázku na body představující výskyt jednotlivých barev. Pod pojmem shluk si pak můžeme představit nahuštění bodů na pravém obrázku, což na původním levém obrázku je ilustrováno pixely se stejnou nebo podobnou barvou.



**Obrázek 4.5:** Rozdělení obrázku na body představující výskyt barev [8]

Algoritmy shlukových metod můžeme klasifikovat do několika tříd [3]:

- Exkluzivní shlukování
- Překrývající se shlukování
- Hierarchické shlukování
- Pravděpodobnostní shlukování

Při řešení problémů jsem se zaměřil na jednodušší algoritmy z třídy exkluzivního shlukování, především na metodu *k-means* a *mean shift*. Použitím těchto metod docílím rozdělení obrazu do několika shluků, které vyjadřují podobné jasové hodnoty v barevných odstínech obrazu nebo podobné jasové hodnoty v šedotónovém obrazu a vzdálenost mezi těmito body. Těmito shluky odliším jednotlivé objekty ve snímané scéně.

#### 4.2.5.1 Metoda *k-means*

Algoritmus *k-means* [10] je jednoduchá iterační metoda pro klasifikaci vstupních bodů do předem určeného počtu  $k$  shluků. Prvně definujeme  $k$  středů (*centroids*), jeden pro každý shluk. Tyto středy by měly být logicky rozmístěny na odlišných místech co nejdál od sebe, protože různé rozmístění počátečních středů zapříčiní různé výsledky. Dalším krokem je přiřadit každý bod, patřící k vstupním datům, do nejbližšího středu. Po rozdělení všech bodů do shluků musíme přepočítat těžiště všech středů. Tím se pro některé body změní nejbližší střed a tak se musíme vrátit o krok dozadu, kde přidělujeme body středům, s tím rozdílem, že máme k dispozici nové souřadnice  $k$  středů. Tento algoritmus se pak opakuje až do té doby, kdy se už souřadnice středů nepohybují. V konečném důsledku se tato metoda snaží minimalizovat cílovou funkci [3]:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i(j) - c_j\|^2 \quad (7)$$

kde  $\|x_i(j) - c_j\|^2$  je vzdálenost mezi vstupními body  $x_i(j)$  a středy  $c_j$ ,  $n$  je počet vstupních bodů a  $k$  je počet středů.

Algoritmus můžeme zapsat pomocí následujících kroků [3]:

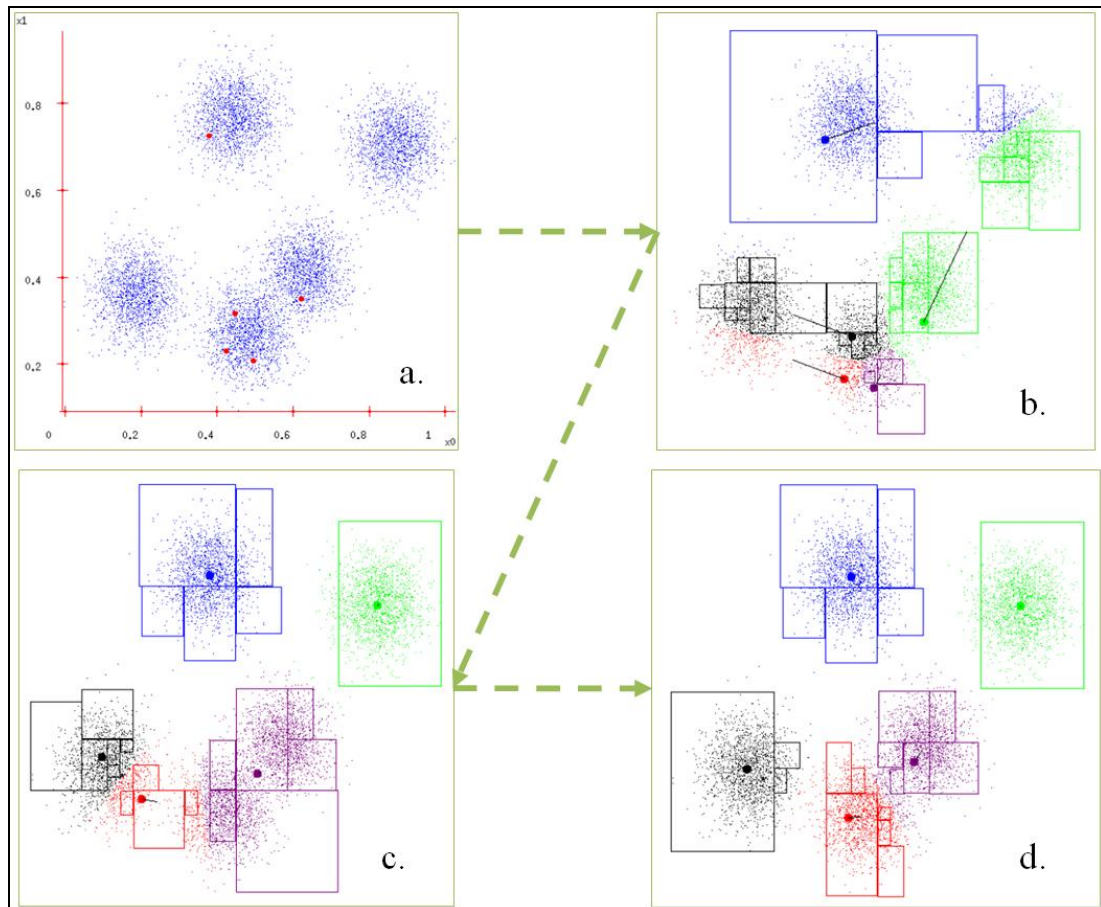
- I. Umístit  $k$  vzorků do prostoru reprezentujícího objekt. Tyto body představují iniciační skupinu středů.
- II. Přiřadit každý bod objektu do skupiny bodů nejbližze středů.
- III. Jsou-li přiřazeny všechny body, přepočítat pozici  $k$  středů.
- IV. Opakovat kroky II a III tak dlouho, doku se jednotlivá těžiště nebudou přemísťovat.



Algoritmus k-means je vždy konečný, ale jeho výsledek nemusí být optimální z hlediska globálního minima funkce. Je totiž velmi citlivý na náhodné rozmístění počátečních středů shluků. Tento problém může být vyřešen vícenásobným spuštěním algoritmu. [3]

Názorný příklad algoritmu si ukažme podrobně na základě *obrázku 4.6.*:

- Nejprve si pro *Obrázek 4.6a* určíme počet shluků  $k$  a rozmístíme náhodně počáteční středy  $m_1, m_2, \dots, m_k$ .
- Do doby, kdy se jednotlivé středy přestanou pohybovat děláme smyčku:
  - Přiřadíme každý bod do nejbližšího středu (příkaz znázorňuje přiřazení různých barev pro každý bod na *obrázku 4.6b-d*).
  - Pro  $i$  od 1 do  $k$  provádíme: (na *obrázku 4.6b,c* je to znázorněno černými čarami představujícími pohyb středů)
    - Nahradíme střed  $m_i$  novým středem, který vypočítáme ze vzdáleností bodů nově přiřazených ke starým středům.
  - Konec cyklu.
- Konec smyčky (středy se ustálily v jednom místě - *Obrázek 4.6d*).



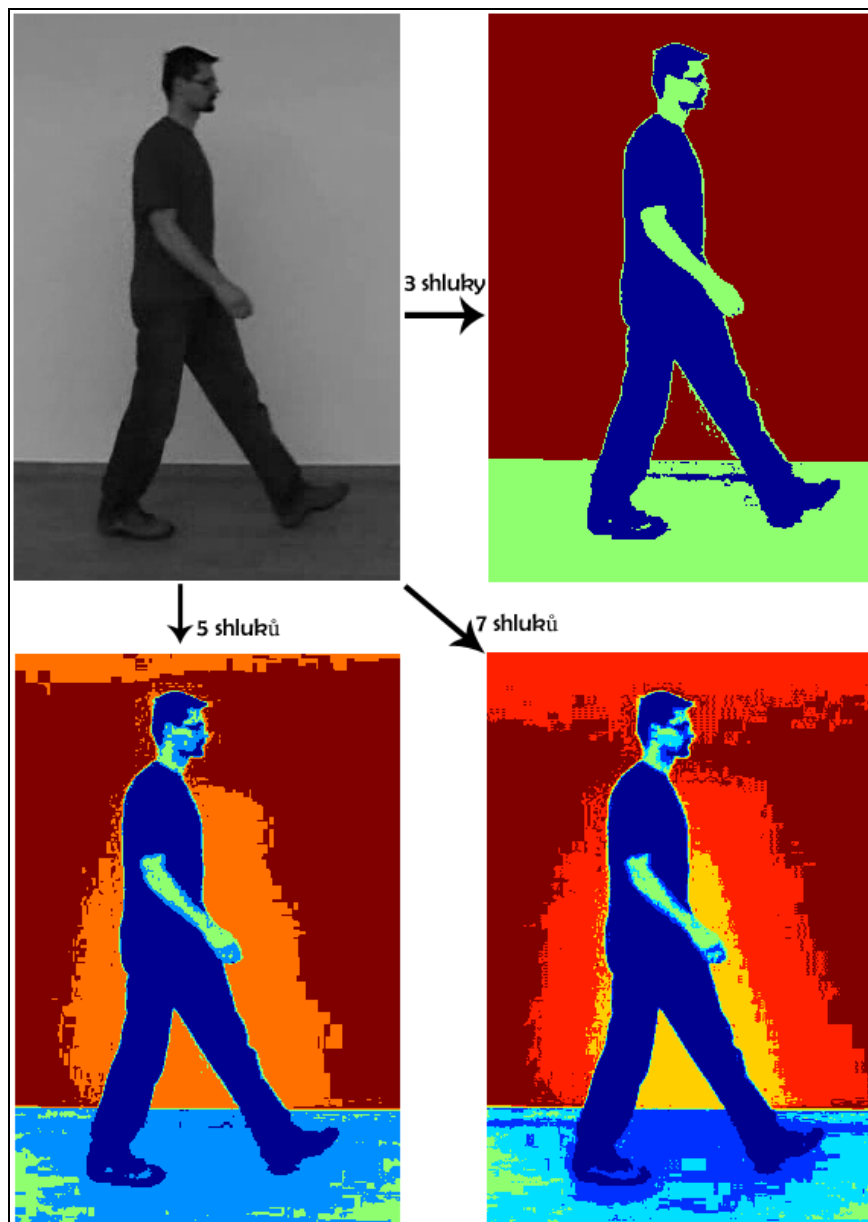
**Obrázek 4.6:** Názorné předvedení algoritmu pomocí obrázků: a) umístění  $k$  vzorků do prostoru. b) + c) přiřazování bodů do jednotlivých středů d) konečné rozpořování shluků a k nim přiřazených bodů [6]

Nejllepší možné rozpoznání všech shluků v obrazu a přiřazení k nim bodů však závisí na několika aspektech, z nichž nejdůležitější jsou [3]:

- Inicializace počátečního rozmístění středů
  - na tomto nastavení závisí konečný počet iteračních kroků a tedy délka výpočtu (čas výpočtu),
  - nejpoužívanější způsob inicializace je náhodné rozmístění, které ne vždy vede k nejlepšímu výsledku rozdělení bodů do shluků.
- Nastavení počtu shluků  $k$ 
  - většinou nastavuje uživatel, ale může být nastaven automaticky, například po zjištění všech lokálních maxim v histogramu.

- Použité metrice pro měření vzdáleností bodů od středů.

Tato metoda nám poskytne rozdělení  $k$  shluků bodů se středovými body v těžištích. V mém programu se snímek s objektem nejprve převede na šedotónový a teprve pak se posílá na vstup do funkce *k-means*, díky které jsou nalezeny výsledné shluky a k nim přiřazené body. Snímky segmentovaného objektu ve falešných barvách, které představují jednotlivé shluky bodů, znázorňuje *obrázek 4.7*.



**Obrázek 4.7:** Segmentace objektu pomocí *k-means* s odlišným počtem shluků

Na tomto obrázku je vidět počáteční obrázek (šedotónový) a výsledné segmentované snímky při výběru počtu shluků  $k=3,5,7$ . Při použití 3 shluků algoritmus přiřadil pozadí jeden shluk, člověku druhý a podlaze spolu s pokožkou člověka třetí shluk. Problém stejného shluku pokožky a podlahy se dá vyřešit tím, že rozdělíme objekt na dolní a horní část přičemž člověka budeme hledat v obou částech nezávisle na sobě. V případě, kdy by měl člověk oblečené kraťasy, by algoritmus spojil nezahalené nohy s podlahou, což by mělo za následek špatné rozpoznání objektu od pozadí. Stejně problémy se spojením pokožky a podlahy se vyskytují na snímcích s 5 a 7 shluky. S rostoucím počtem shluků  $k$  je obrázek rozdělován na více shluků s menším objemem bodů, tedy v průměrně se celková vzdálenost bodů od jednotlivých středů se zmenšuje, což způsobuje například ubývání počtu bodů v noze na levé straně snímků a jejich přechod do shluku podlahy. Proto se domnívám, že optimálním řešením vzhledem k pozdějšímu nalezení objektu v těchto snímcích je segmentace s počtem shluků  $k=5$ .

#### 4.2.5.2 Metoda mean shift

*Mean shift* algoritmus je neparametrická shluková metoda, která nepotřebuje počáteční znalost počtu shluků. Metoda je založena na odhadu hustoty jádra (*kernel*). Mámeli  $n$  bodů dat  $x_i, i=1, \dots, n$ ,  $d$ -rozměrného prostoru  $R^d$ , pak vícerozměrný odhad hustoty jádra (*kernel*) s jádrem  $K(x)$  a poloměrem okna  $h$  je [7]:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (8)$$

pro radiálně symetrickou funkci jádra (*kernelu*) můžeme zvolit profil [7]:

$$K(x) = c_{k,d} k(\|x\|^2) \quad (9)$$

kde  $c_{k,d}$  je normalizační konstanta, která zaručuje konvergenci funkce  $K(x)$  do 1.

Po dosazení rovnice (9) do (8) a vypočítání gradientu funkce  $\nabla f(x) = 0$  [7]:

$$\begin{aligned}\nabla f(x) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^n (x_i - x) g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[ \sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \right] \left[ \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x \right]\end{aligned}\quad (10)$$

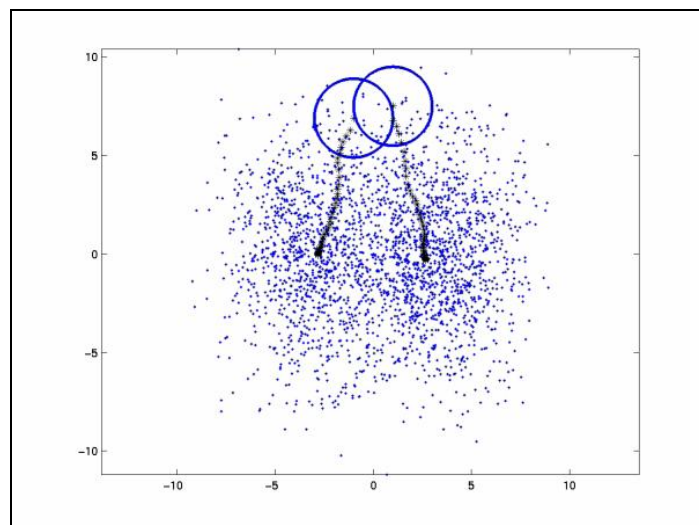
kde  $g(s) = -k'(s)$ . První termín v rovnici je proporcionální k odhadu hustoty v  $x$  počítaná s jádrem  $G(x) = c_{g,d} g(\|x\|^2)$  a druhý termín [7]:

$$m_h(x) = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x \quad (11)$$

je *mean shift* vektor, který vždy míří směrem k maximálnímu růstu hustoty. Postup při algoritmu obsahuje tyto výpočty [7]:

- výpočet *mean shift* vektoru  $m_h(x^t)$ ,
- přepočítání okna:  $x^{t+1} = x^t + m_h(x^t)$ .

Tyto dva výpočty se provádějí až do doby konvergence a zajištění, že gradient funkce hustoty je nulový. Bod, ve kterém se výpočet zastavil, je pak lokální maximum hustoty shluku bodů. Názorná ukázka tohoto výpočtu je na *obrázku 4.8*.

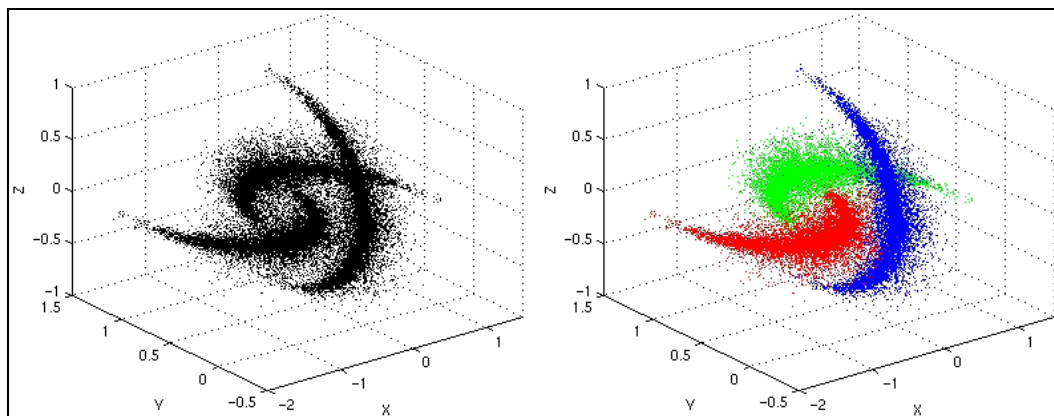


**Obrázek 4.8:** Grafická ukázka výpočtu maximální hustoty shluků [7]

Postup výpočtu pro praktické použití pomocí algoritmu *mean shift* a rozdělení bodů v obrazu na shluky můžeme shrnout v těchto krocích:

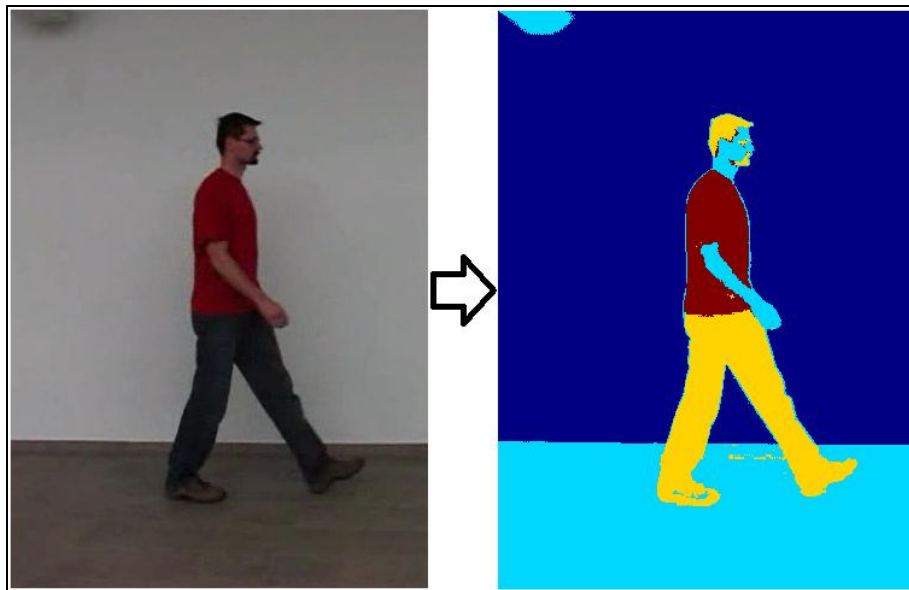
- I. Pro každý bod použijeme algoritmus *mean shift* a zapamatujeme si lokální maximum, ve kterém skončil.
- II. Shlukujeme vzorky, které skončily ve stejných lokálních maximech s určitou tolerancí.

Oblasti tvořené počátečním umístěním bodů, které skončily ve stejných maximech, nazýváme *basin of attraction*, a body které skončily ve stejné *basin of attraction* jsou přiřazeny do stejného shluku. Ukázka rozdělení bodů obrázku na shluky s použitím metody *mean shift* je znázorněna na *obrázku 4.9*. [7]

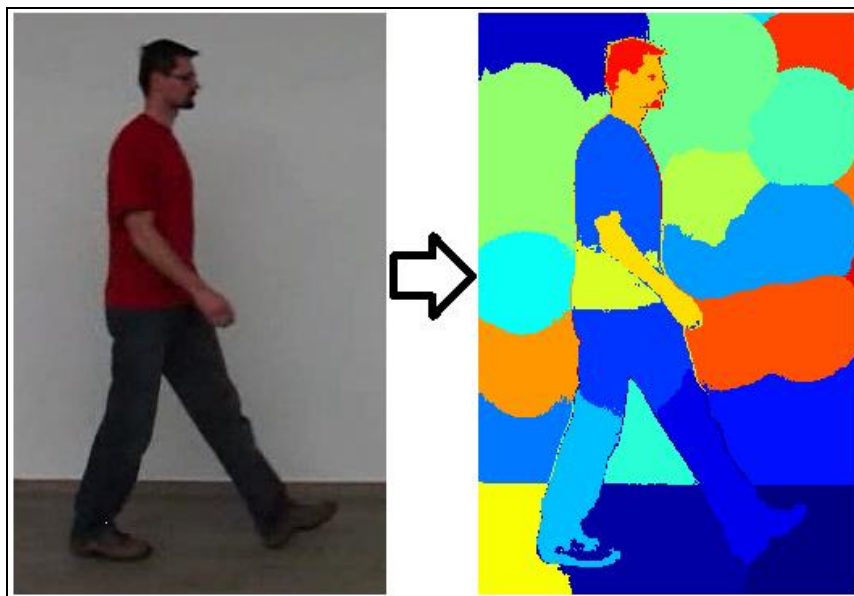


**Obrázek 4.9:** Rozdělení bodů v obrázku na shluky [7]

Při použití pro naše účely budeme od sebe rozdělovat jednotlivé objekty lišící se barvou a hledat tak ve snímku námi hledaný objekt (v našem případě člověka), který na sobě bude mít oblečení barevně odlišné od pozadí. Proto vstupem metody *mean shift* bude intenzita barevných kanálů RGB každého bodu v obrazu. Vzhledem k tomu, že máme k dispozici ještě dva další údaje (souřadnice bodů), můžeme je také použít a přivést na vstup metody *mean shift* spolu se třemi barevnými kanály RGB. Výslednou segmentaci obrázku s objektem můžeme shlédnout na *obrázku 4.10* a *obrázku 4.11*.



**Obrázek 4.10:** Rozdělení obrázku na objekty pomocí *mean shift* (velikost okolí = 35)



**Obrázek 4.11:** Rozdělení na objekty pomocí 5-rozměrného *mean shift* (vel.ok. = 35)

V případě snímku segmentovaného metodou 5-rozměrného *mean shift* jsou jasně rozlišitelné okrouhlé shluky způsobené přidáním k třem počátečním atributům (barvám *rgb*) dvou dalších atributů - souřadnic bodů v obrazu. Velikost shluků, a

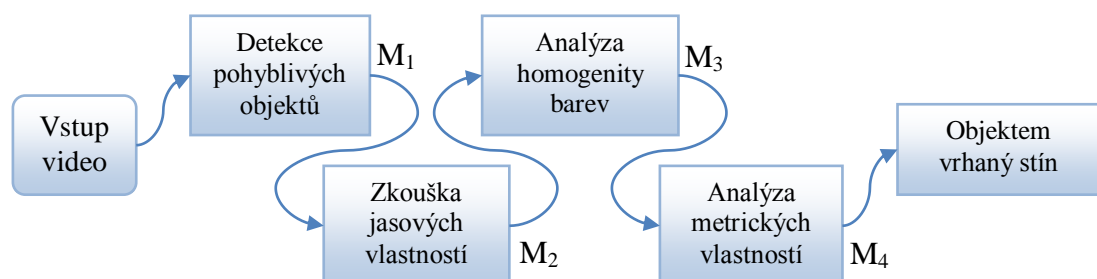
tedy množství bodů v nich obsažených, je limitován právě přidanými atributy souřadnic a taky parametru pro metodu *mean shift* - velikosti okolí, která v tomto případě je  $VO = 35$ . Při segmentaci 3-rozměrnou metodou *mean shift* s atributy barev *rgb* je výsledný snímek podobný výsledku segmentace pomocí *k-means*. Ačkoliv je rozdělení objektů a pozadí přesnější v porovnání z *k-means*, je metoda *mean shift* pomalejší. Proto podle mého názoru je metoda *k-means* k rychlému vyhledání člověka v obraze vhodnější.

#### 4.2.6 Detekce stínů

Vrhání stínu tělesem může zapříčinit spojování objektů, deformaci jejich tvarů nebo jiné obrazové ztráty. Z těchto důvodů je detekce stínů potřebná pro přesnou detekci objektů v pohybující se scéně. Popsaná metoda je založena na fotometrické invariančnosti *rgb* barevného modelu. Algoritmus postupně navrhuje kandidáty na stín z jasových vlastností obrazu, pak z barevných vlastností, a nakonec odstraní nevhodné kandidáty pomocí metrických příznaků. Metoda se zaměřuje na normalizované podobě barevného modelu definovaného následovně:

$$r = \frac{R}{R+G+B}; g = \frac{G}{R+G+B}; b = \frac{B}{R+G+B} \quad (12)$$

kde  $r+g+b=1$ . Tento *rgb* model je necitlivý na orientaci povrchu, směr a intenzitu osvětlení. Blokové schéma algoritmu můžeme shlédnout na *obrázku 4.12*.



**Obrázek 4.12:** Blokové schéma algoritmu pro detekci stínů



Postup detekce je rozdělen na čtyři kroky:

I. Detekce pohyblivých objektů.

Předpokládejme pohyb objektu ve snímané scéně. Máme-li pak k dispozici její pozadí  $B(x)$  a snímek s objektem  $I(x)$ , provedeme pro zjištění počátečních odhadovaných bodů stínu jejich odečtení, a výsledek porovnáme s předem danou hodnotou  $2 * \sigma(x)$  podle rovnice [9]:

$$F(x) = \begin{cases} 1 & |I(x) - B(x)| \geq 2 * \sigma(x) \\ 0 & \text{jinak} \end{cases} \quad (13)$$

kde  $\sigma(x)$  je střední hodnota jasového zkreslení pro pixel na pozici  $x$  a počítá se následovně [9]:

$$\sigma(x, i) = \max(\sigma_{\min}, \alpha |I(x, i) - B(x)| + (1 - \alpha) \sigma(x, i - 1)) \quad (14)$$

kde minimum zkreslení  $\sigma_{\min}$  je zaveden jako prahový šum, abychom předešli klesnutí  $\sigma$  pod kritickou hodnotu, kde by výsledkem hledání byl téměř každý bod. Pro zjednodušení výpočtu jsem však nastavil  $\sigma$  na konstantní hodnotu. Výsledný vyprahovaný snímek označíme jako matici bodů  $M_1$ . [9]

II. Vyřazení nesprávných bodů podle jasového koeficientu.

V dřívějších studiích bylo prokázáno, že mezi normálně nasvětleným pixelem a stejným pixelem ve stínu může být linearita. Proto ve II. kroku se díky poměru pozadí a scény s objektem snažíme zmenšit počet kandidátů na stín tak, že odstraníme body, které nespĺňují požadavky jasů kladené rovnicí [9]:

$$\forall p(x) \in M_1 \text{ if } \left( \alpha < \frac{B(x)}{I(x)} < \beta \right) \Rightarrow M_2 = M_2 \cup p(x) \quad (15)$$

Výsledný obraz  $M_2$  je složen s vyhovujících bodů předešlé rovnice a bodů, které patří do obrazu  $M_1$ .

III. Analýza homogenity barev.

V tomto kroku se provádí korelace mezi vstupními pixely scény s objektem a výstupními pixely pozadí v normalizovaném prostoru barev  $rgb$ . Následně jsou jako kandidáti na stín vybrány pixely, které mají větší barevnou korelaci než předem určený práh  $H_T$ . Předpokládáme, že pixely  $x \in (r, g, b)$  patří pozadí a pixely  $x \in (r', g', b')$  jsou součástí scény popředí s objektem. Vypočítáme koeficienty  $\alpha, \beta, \gamma$  podle rovnice [9]:

$$\alpha = 1 - \frac{|r - r'|}{r + r'}; \quad \beta = 1 - \frac{|g - g'|}{g + g'}; \quad \gamma = 1 - \frac{|b - b'|}{b + b'} \quad (16)$$

a korelaci barev,  $H$ , vypočítáme jako  $H(r, g, b) = \alpha * \beta * \gamma$ . Podle vztahu [9]:

$$\forall p \in M_2 \text{ if } (H > H_T) \Rightarrow M_3 = M_3 \cup p \quad (17)$$

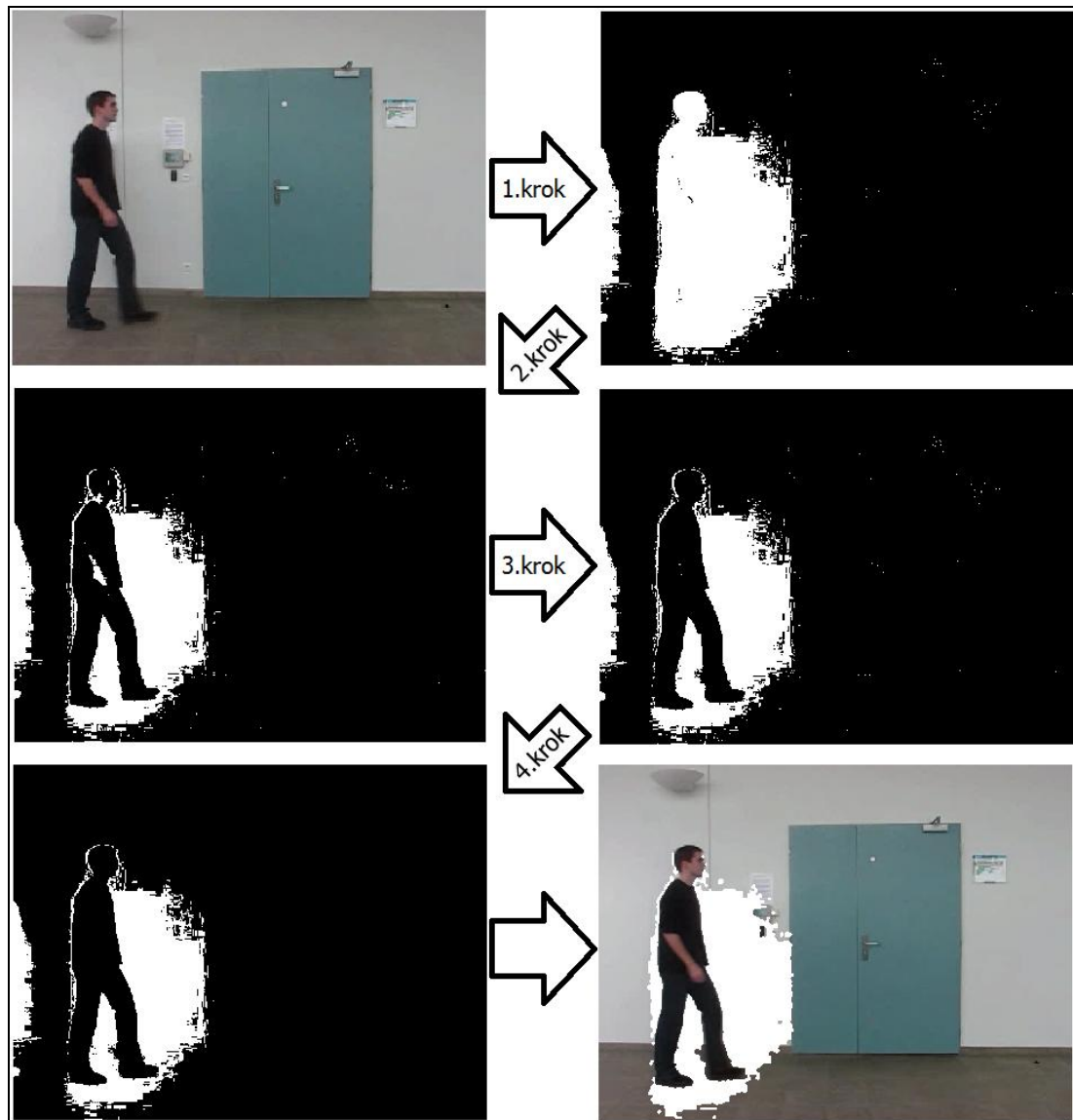
pak noví kandidáti na stín ( $M_3$ ) jsou složeni z bodů, které mají větší barevnou korelaci ( $H$ ) než práh  $H_T$  a současně patří do předchozí skupiny bodů  $M_2$ .

#### IV. Odstranění geometricky nevyhovujících objektů.

Poslední krok vykonává proces na bázi porovnávání metrických příznaků pro odstranění falešných kandidátů na stín. Předpokládejme, že  $\{R(i)\}$  jsou segmentované oblasti kandidátů na stín z kroků II a III. Pak  $C(i)$  je obvod oblastí a  $L(i)$  je hranice mezi kandidáty a detekovaným objektem. Koeficient vzniklý poměrem těchto dvou příznaků bude menší pro skutečný stín, proto závislost můžeme zapsat ve tvaru [9]:

$$\forall R \in \{R(i)\} \text{ if } \left( \frac{L(i)}{C(i)} < L_T \right) \Rightarrow M_4 = M_4 \cup R \quad (18)$$

Kde  $L_T$  je práh rozhodující zda-li oblasti z  $\{R(i)\}$  budou součástí výsledného snímku vykreslujícího stíny vrhané hledaným objektem. Na *obrázku 4.13* je celá metoda znázorněna pomocí obrázků od snímané scény až po nalezení stínů vrhaných objektem, kde bílá barva představuje kandidáty bodů na stín. Použité konstanty jsou  $\sigma = 10$ ;  $\alpha = 1, \beta = 5$ ;  $H_T = 0,94$ ;  $L_T = 0,5$ .

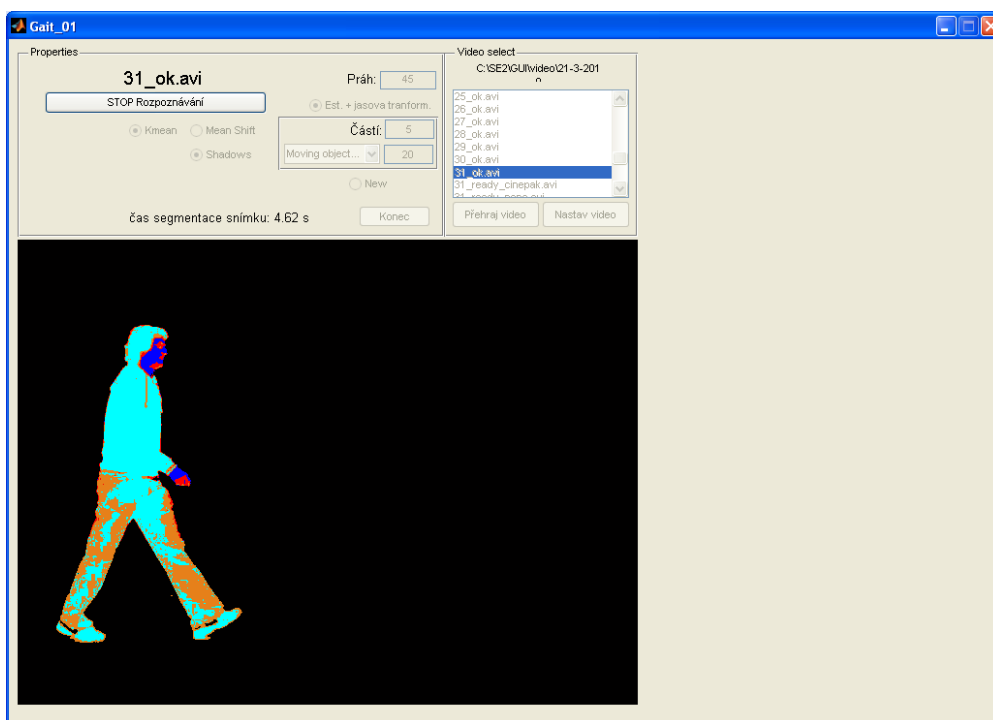


*Obrázek 4.13: Znáznornění detekce stínů krok po kroku*

## 5. Segmentace v programu Matlab

### 5.1 Vzhled programu

Jelikož pro detekci objektu ve snímané scéně je možno použít více metod, vytvořil jsem pro zrychlení detekce a posuzování účinnosti metod program s uživatelským rozhraním v Matlabu, který s videa s chůzí člověka detekuje objekt kteroukoliv metodou podle nastavení uživatele. Vzhled uživatelského rozhraní programu je znázorněn na *obrázku 5.1*. Je zde vidět část vlastností (v levé části na hoře), kde můžeme nastavovat předem popsané metody a různě je kombinovat, a část výběru videa, ve kterém je tabulka pro prohledávání složek a tlačítka pro nastavení a přehrávání videa. Pro nastavení videa pro segmentaci je potřeba ho vybrat ze seznamu a potvrdit tlačítkem „Nastav video“. Segmentace videa se spouští tlačítkem „START Rozpoznávání“. V dolní části je místo pro zobrazení aktuálního segmentovaného snímku a pro pozdější zpětné nahlížení na všechny snímky rozčleněné algoritmem.

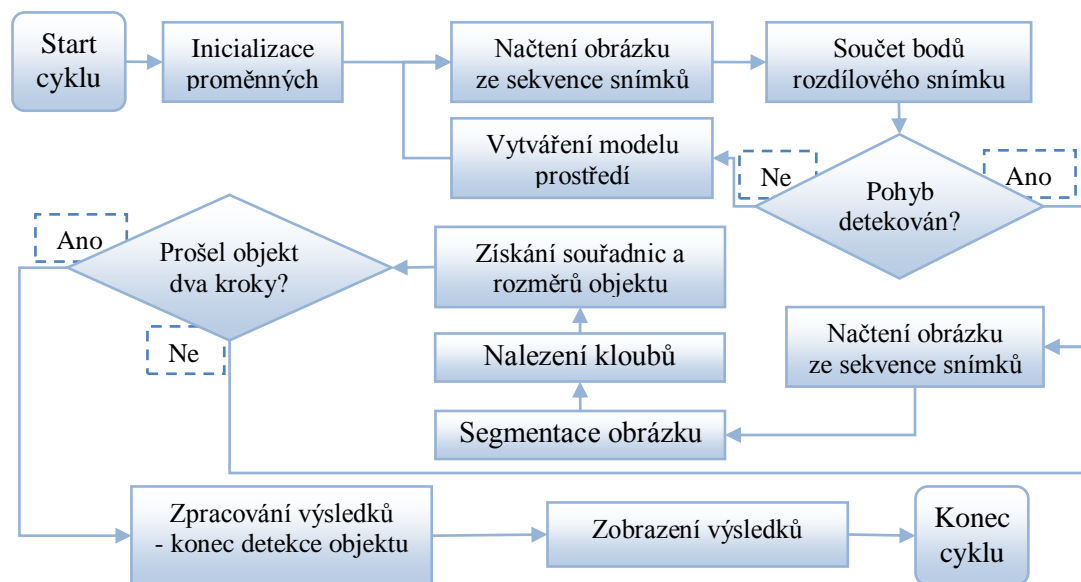


*Obrázek 5.1: Uživatelské rozhraní programu*

## 5.2 Hlavní cyklus rozpoznávání

Na začátku rozpoznávacího algoritmu se inicializují potřebné proměnné k bezchybnému průběhu programu. Pak se načte snímek z videa a vytvoří se rozdílový snímek z modelu prostředí a z načteného obrázku. Z tohoto rozdílového snímku se vypočte součet jasových hodnot, podle kterého se rozhoduje detekce pohybu. Jestli je součet menší než práh  $h$  (vypočten vzorcem  $12 \cdot \text{výška} \cdot \text{šířka}$  snímku), zprůměrně se aktuální snímek s modelem prostředí, a tím z části odstraníme šum vytvořený při snímání videa.

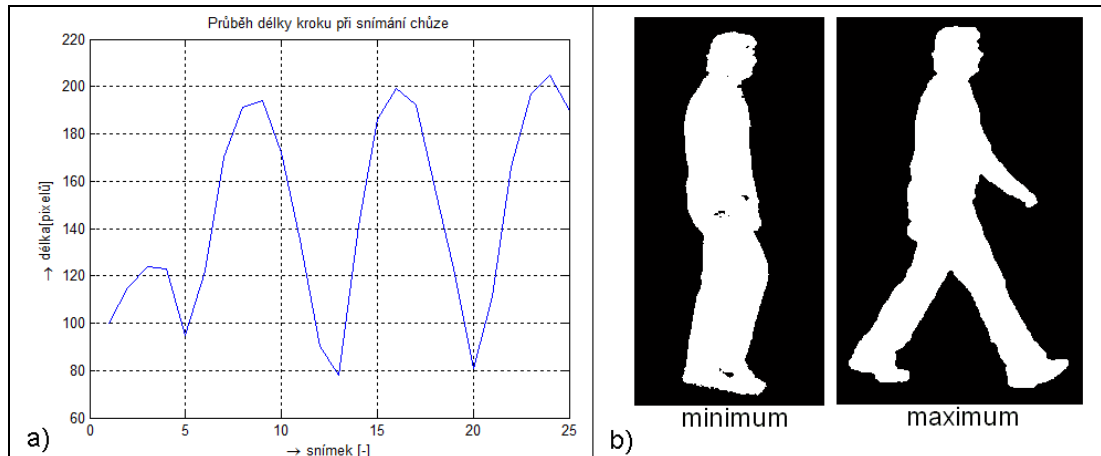
Jestli je součet větší než práh  $h$ , model prostředí se uloží a program se přesune do další části, kde jsou snímány obrázky s pohybujícím se objektem. Nejprve se načte následující snímek, který spolu s modelem vstoupí do algoritmu segmentace a na výstupu vznikne obrázek požadovaného objektu, ve kterém vyhledáme důležité body potřebné k identifikaci. Následně získáme souřadnice související z pozicí objektu ve snímané scéně a taky hodnotu délky objektu, ze které po vynesení do grafu snímků vzniknou maxima a minima, určující v jakém úseku chůze se člověk nachází.



**Obrázek 5.2:** Blokové schéma hlavního cyklu programu

Přičemž dva po sobě nacházející se maxima znamenají jeden krok. Jak je vidět na obrázku 5.3b, je největší hodnota délky objektu v situaci, kdy má člověk při chůzi

nohy od sebe, naopak má-li nohy při sobě, je hodnota délky objektu nejmenší. Takto snímáme a segmentujeme snímky do doby, kdy algoritmus rozpozná dva kroky, pak přejde do další části programu.



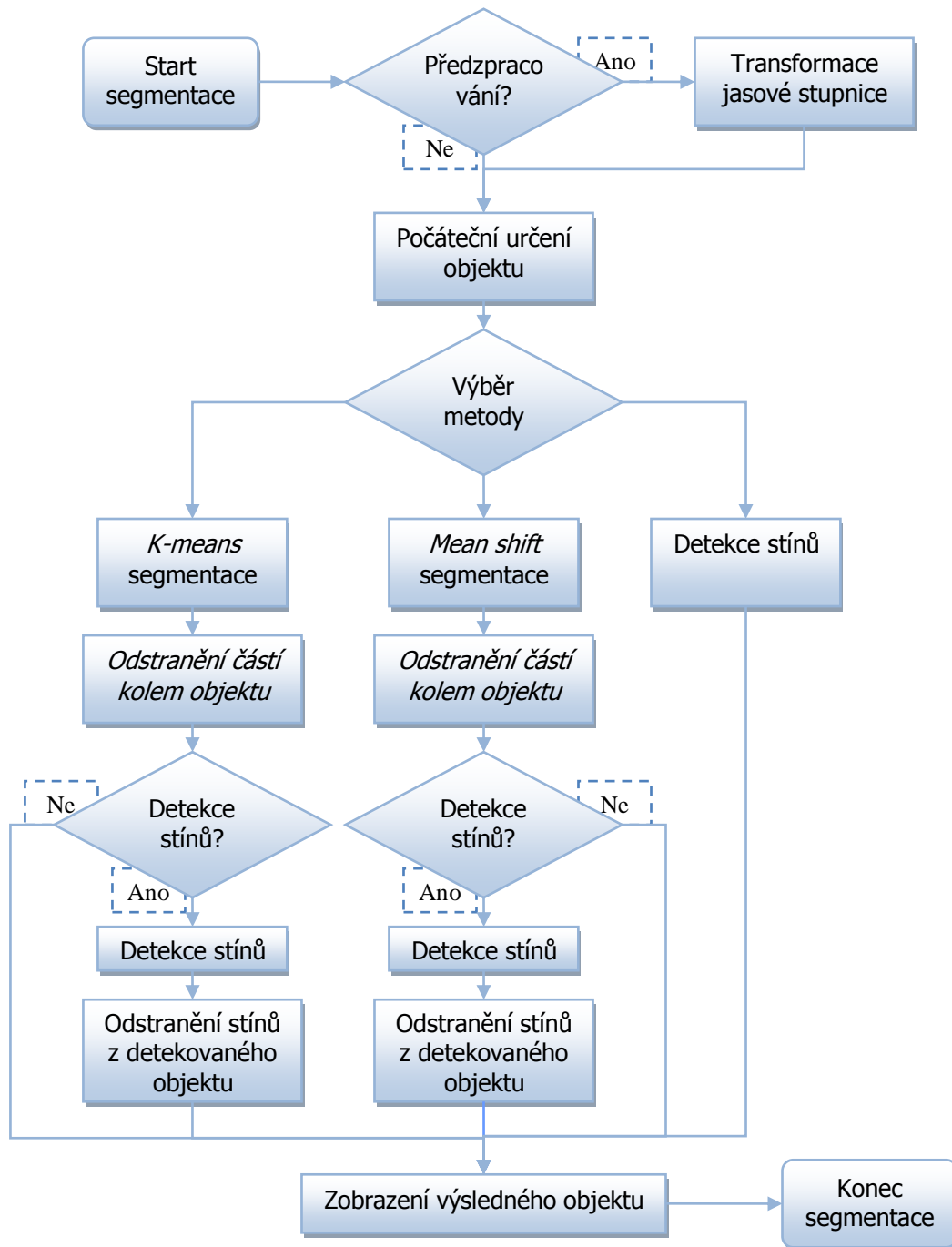
**Obrázek 5.3:** a) Průběh změny součtu bodů objektu, b) znázornění pozic chůze při min. a max. součtu bodů

Na konci rozpoznávacího procesu se obrázky s objektem uloží a následně vykreslí do zobrazovacího okna.

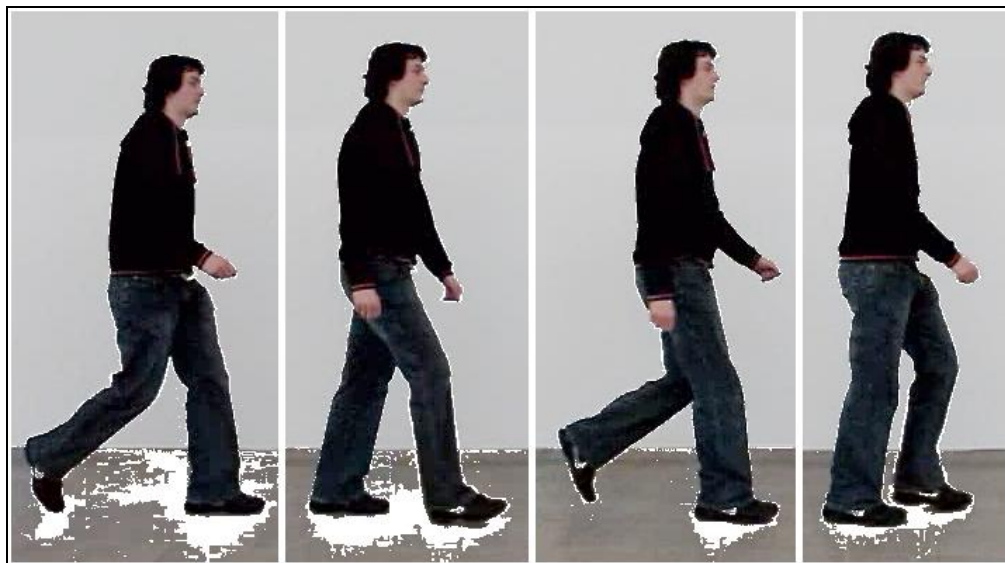
### 5.3 Segmentace snímku

Nejdůležitější částí předchozího cyklu je segmentace snímku, která má za úkol rozdělit snímek na objekt a pozadí. Blokové schéma je znázorněno na *obrázku 5.4*. Na počátku můžeme zvolit předzpracování obrázku formou transformace jasové stupnice. Pak algoritmus provede počáteční určení místa, kde se může nacházet hledaný objekt. Způsob vymezení místa je následující: vytvoření diferenciálního snímku a vyprahování → odstranění nejmenších objektů → morfologické roztažení zbylých objektů → ponechání největšího objektu.

Následně poté si můžeme vybrat mezi shlukovými segmentačními metodami *k-means* a *mean shift* nebo pouhou detekci stínů. Při výběru poslední možnosti se provede vyhledání stínů v celém obrazu a následné zobrazení. Příklad nalezení stínů můžeme vidět na *obrázku 5.5*.

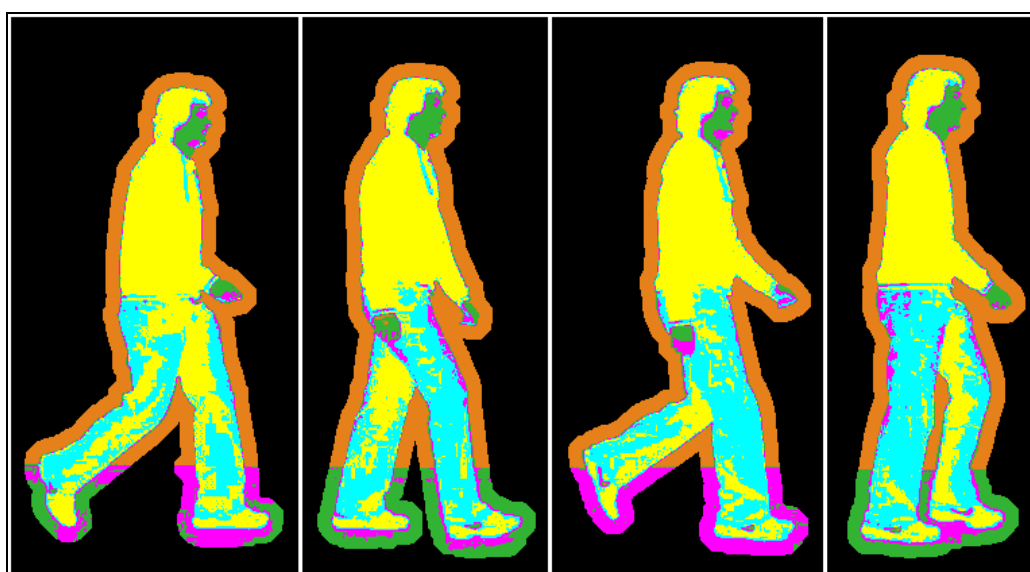


Obrázek 5.4: Blokové schéma segmentace snímku



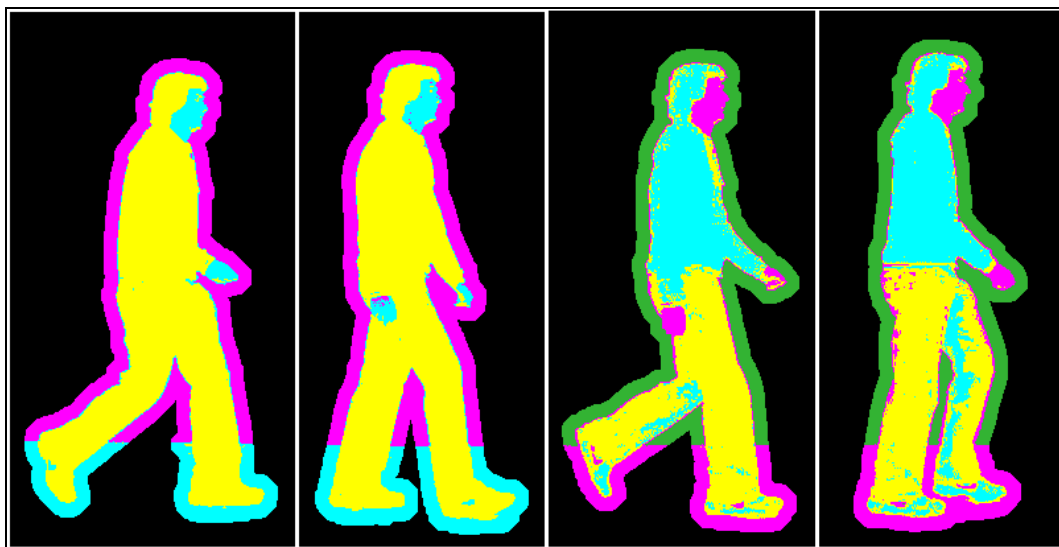
*Obrázek 5.5: Detekované stíny (bílá barva) při chůzi*

Při výběru ostatních dvou možností se počátečně určené místo nejprve segmentuje podle zvolené metody (mean shift nebo k-means, jež jsou popsány v kapitole 4), a pak jsou ze zbylé části odstraněny přebytečné body ležící kolem objektu. *Obrázek 5.6* a *Obrázek 5.7* představují různé polohy chůze člověka po segmentaci, kde každá barva představuje jeden shluk.



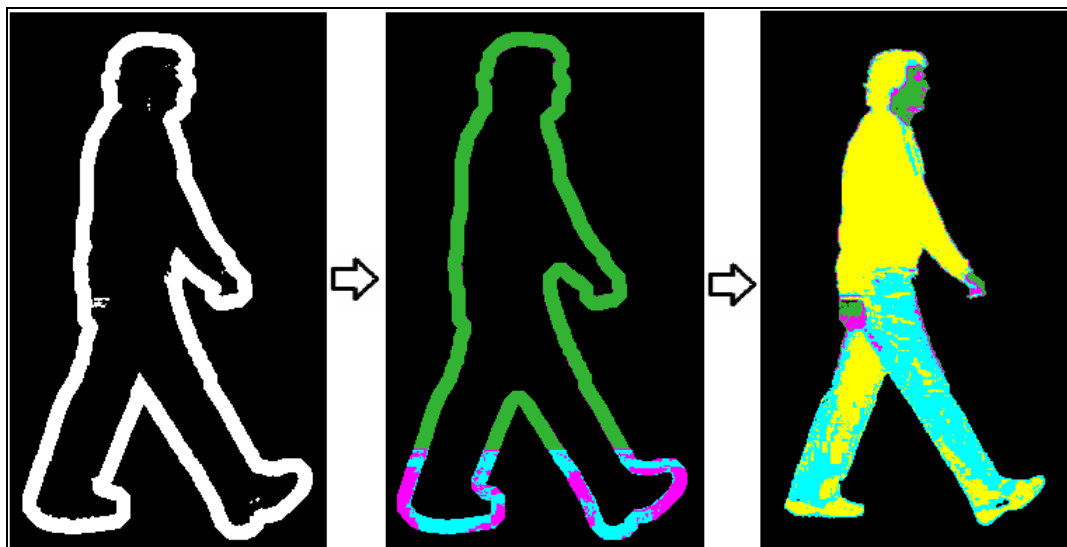
*Obrázek 5.6: Příklad segmentace obrázku metodou k-means (počet shluků = 5) v různých polohách chůze*





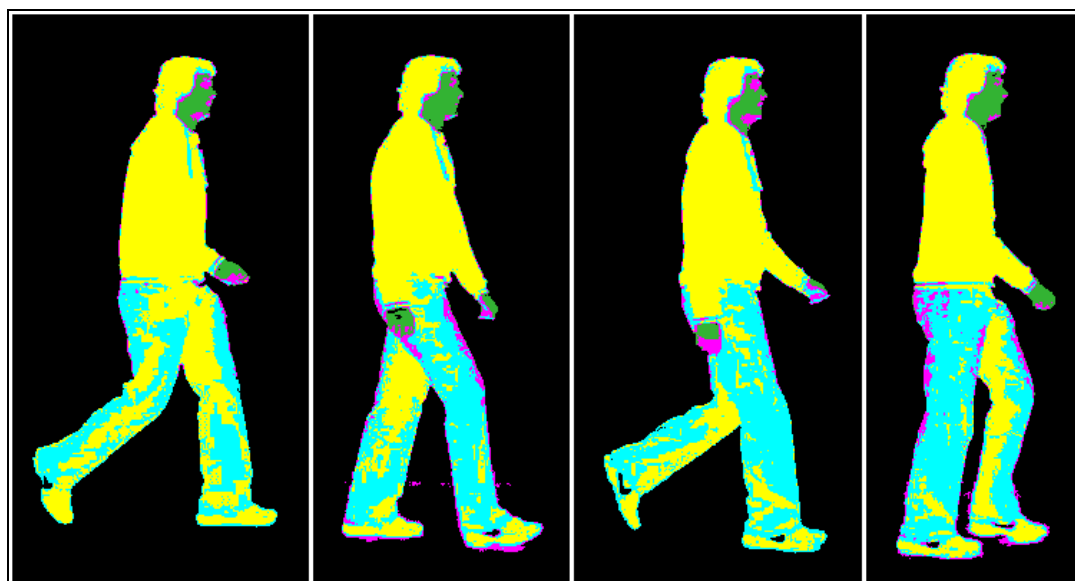
**Obrázek 5.7:** Příklad segmentace obrázku metodou mean shift (velikost okolí = 40)

Při vytváření počátečního určení místa, kde se může nacházet hledaný objekt, jsme prováděli morfologickou dilataci s maticí o rozměrech  $N \times N$ , která nám objekty nacházející se ve snímku rozšířila o daný počet bodů. Následně budeme předpokládat, že právě v těchto bodech je malá pravděpodobnost výskytu hledaného objektu. Proto provedeme rozdíl snímků s matematickou morfologií a bez ní, a tím nám vznikne pásmo, ve kterém po jeho přiložení na segmentovaný snímek zbudou shluky, které nepatří k objektu. Spočítáme proto množství bodů jednoho shluku nalézajícího se ve vybraném pásmu kolem objektu a porovnáme tuto hodnotu s množstvím bodů shluku uprostřed objektu. Je-li bodů v pásmu více než uprostřed objektu, shluk odstraníme. Při zkouškách programu jsem zjistil, že v mnoha případech algoritmus ohodnotí pokožku člověka stejným shlukem jako zem, proto jsem rozdělil objekt na dolní část a horní část a vybírání nesprávných shluků se provádí odděleně v obou částech. Zamezí se tím odstranění shluku, který znázorňuje pokožku člověka a zároveň zem. Na *obrázku 5.8.* vpravo je vidět fialový pás ve stejné rovině oddělující horní a dolní část objektu.



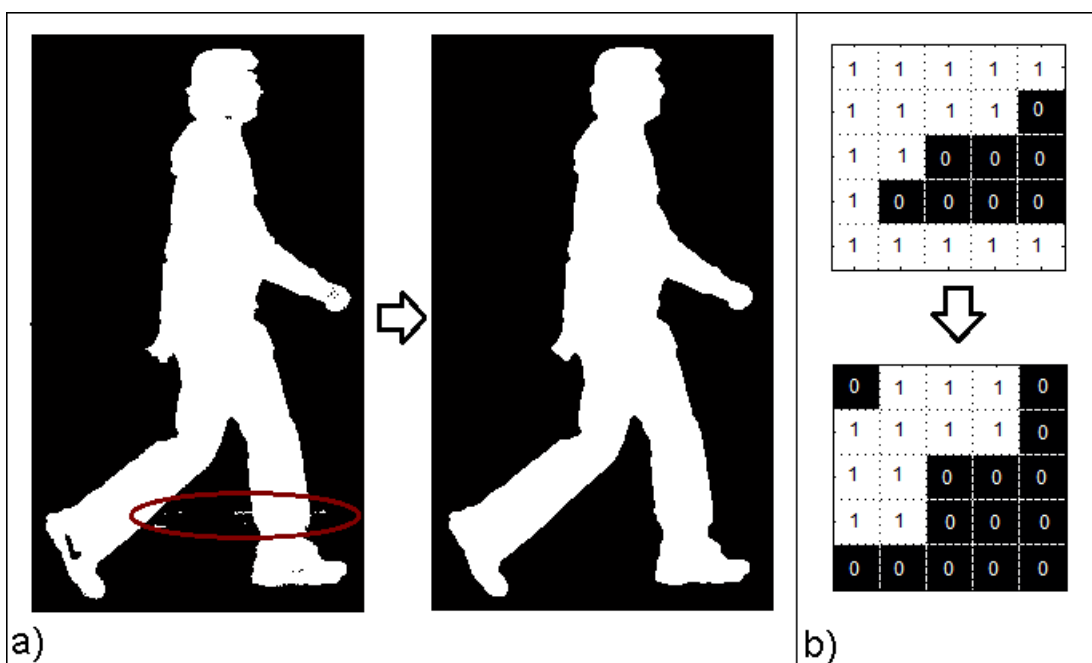
*Obrázek 5.8: Odstranění nevhodných shluků bodů*

Po této operaci máme možnost vybrat si, jestli budeme chtít detekovat a odstranit z výsledného obrázku stín. Pokud zvolíme tuto cestu, algoritmus detekuje stíny a následně je odstraní z obrázku s objektem. Výsledný segmentovaný snímek s odstraněním přebytečných částí a stínů znázorňuje *obrázek 5.9*.



*Obrázek 5.9: Obrázek chůze objektu po segmentaci*

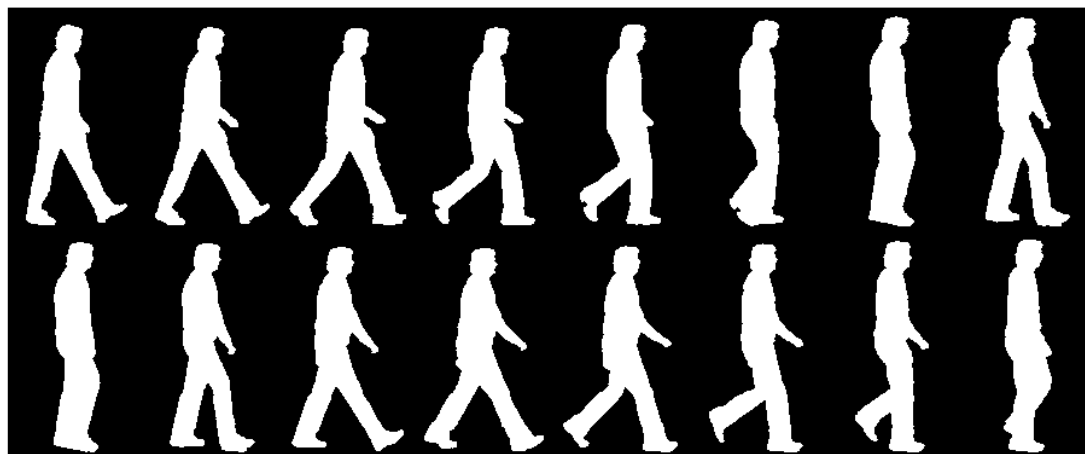
Na některých předcházejících segmentovaných snímcích vidíme kolem siluety vidět tenké podlouhlé čáry, které je potřeba odstranit. Tento úkon provedeme pomocí morfologické operace „majority“, která je definována tak, že pro každý bod v binárním obrázku (siluetě) kontroluje kolik sousedních pixelů je 1. Pokud jich je 5 a více, pak nastaví pixel na 1. V opačném případě nastaví pixel na 0. Na *obrázku 5.10b* je jednoduchý příklad operace s maticí o rozměrech 5x5. Na *obrázku 5.10a* je pak znázorněna operace se siluetou člověka po segmentaci, kde červeně označená oblast znázorňuje nežádoucí šum, který popisovaná morfologická operace odstraní.



**Obrázek 5.10:** Ukázka morfologické operace na siluetě a na jednoduchém obrázku

Tento proces segmentace se opakuje se všemi obrázky do doby, kdy program pozná, že člověk udělal dva kroky. Poté se cyklus rozpoznávání objektu z videa se segmentací ukončí a provede se zpracování těchto výsledků. Do toho patří ořezání z celkového obrázku a vložení chůze objektu do jedné proměnné (do jednoho celku), operace pro získání důležitých bodů hledaného objektu (kotníků a hlavy) a zobrazení celé chůze uživateli, s vykreslením ohraničení znázorňujícím výsledek segmentace a

vyznačením nalezených míst. Z těchto výsledků můžeme například sestavit všechny pózy obrysu člověka do jednoho snímku, jak je to znázorněno na *obrázku 5.11*.



*Obrázek 5.11: Snímky siluety chůze objektu zahrnutý v jednom obrázku*

#### 5.4 Porovnání metod

Všechny metody směřují k jednomu cíli: získat ostře ohraničenou siluetu objektu bez znetvoření šumem. Každá má však svou vlastní cestu. Bohužel žádná není stoprocentní (což vidíme na předchozích obrázcích). Můžeme je však kombinovat a získat tak co nejučinnější a co nejrychlejší algoritmus. V *tabulce 5.1* je porovnání použitých metod a jejich kombinací. Hodnocení je provedeno formou průměrného času segmentace jednoho snímku v sekundách. Je jasné, že nejjednodušší metoda bude mít nejlepší čas segmentace. Bohužel kvalita segmentace je nepřímo úměrná rychlosti, a proto jsou pro pozdější hledání důležitých míst v objektu použitelné metody označené v tabulce kurzivou. Je zde vidět, že metoda *k-means* je rychlejší než *mean shift*. Metoda *k-means* je v programu navržena pro segmentaci šedotónového obrázku, proto ztrácí jednu rozpoznávací informaci, kdežto metoda *mean shift* je navržena pro segmentaci barevného obrázku, a tak má jistou výhodu oproti metodě *k-means*, ale bohužel ztrácí na rychlosti výpočtu.

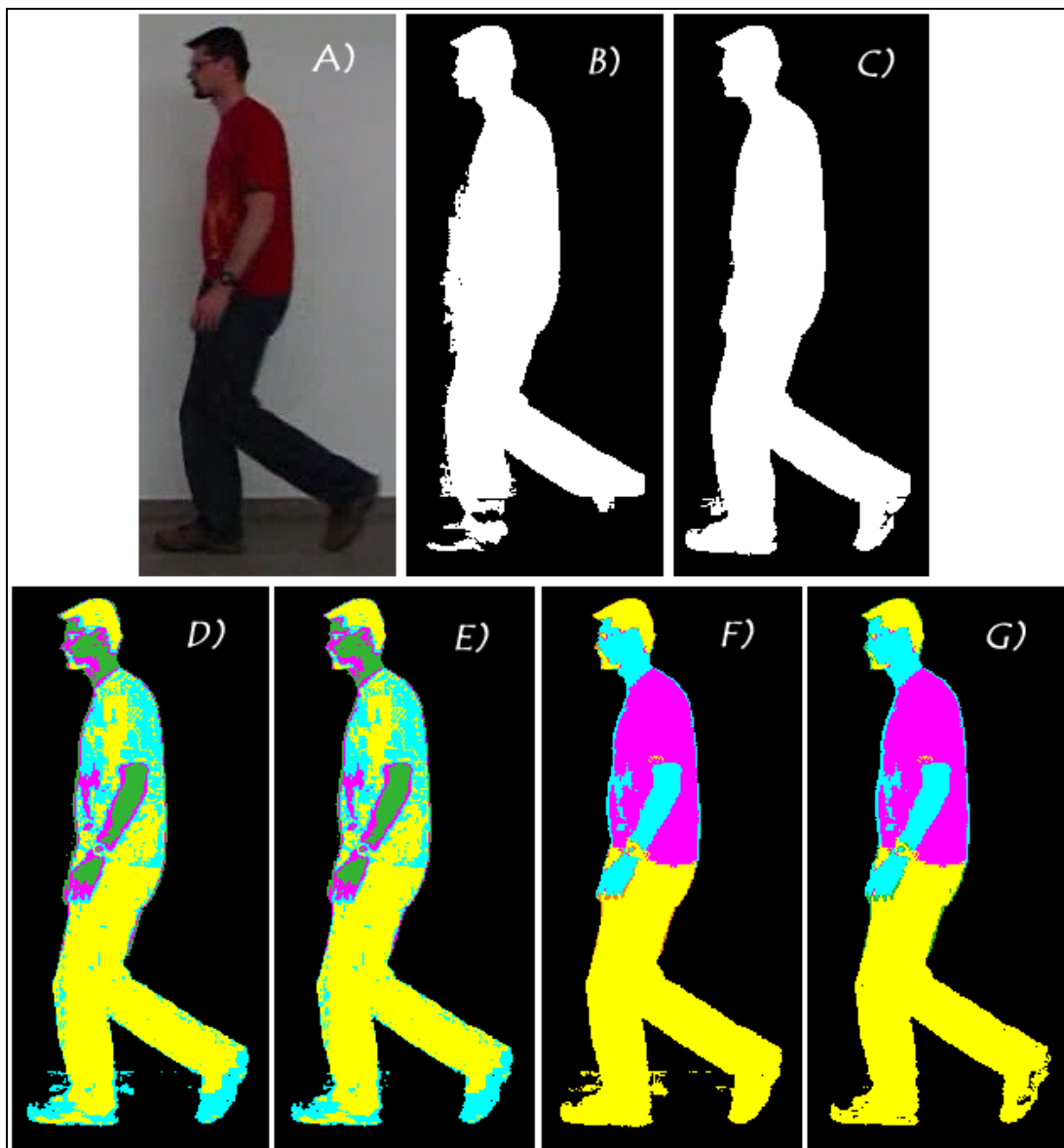
**Tabulka 5.1:** Porovnání rychlosti a kvality rozpoznání různých metod

Použité metody	Průměrný čas segmentace jednoho snímku
Rozdílové snímky	0,11 s
Rozdílové snímky s transformací jasové stupnice	0,46 s
<i>Metoda kmean</i>	0,44 s
<i>Metoda mean shift</i>	3,92 s
<i>Metoda kmean s detekcí stínů a transformací jasové stupnice</i>	1,89 s
<i>Metoda mean shift s detekcí stínů a transformací jasové stupnice</i>	11,02 s

Použitý počítač: emulace Windows XP 32-bit, RAM:1GB, Proc.: 2,87MHz

Na *obrázku 5.12* je pak porovnání kombinací těchto metod pomocí jednoho ze sekvence obrázků, na kterém jsou rozdíly nejlépe viditelné. Na *obrázku 5.12B* je znázorněna segmentace pomocí rozdílových snímků s *prahem = 45*, a pak s přidanou transformací jasové stupnice taky s *prahem = 45* – *obrázek 5.12C*. Na *obrázku 5.12D* je zobrazena segmentace pomocí *k-means* (*s počtem shluků = 5*) s přidanou transformací jasové stupnice, a následně i detekcí a odstraněním stínů na *obrázku 5.12E*. Dále pak *obrázek 5.12F* představuje segmentaci pomocí metody *mean shift* (*velikost okolí shluku = 33*) s transformací jasové stupnice, a následně *obrázek 5.12G* ukazuje stejnou konfiguraci metod s dodáním detekce a odstranění stínů.

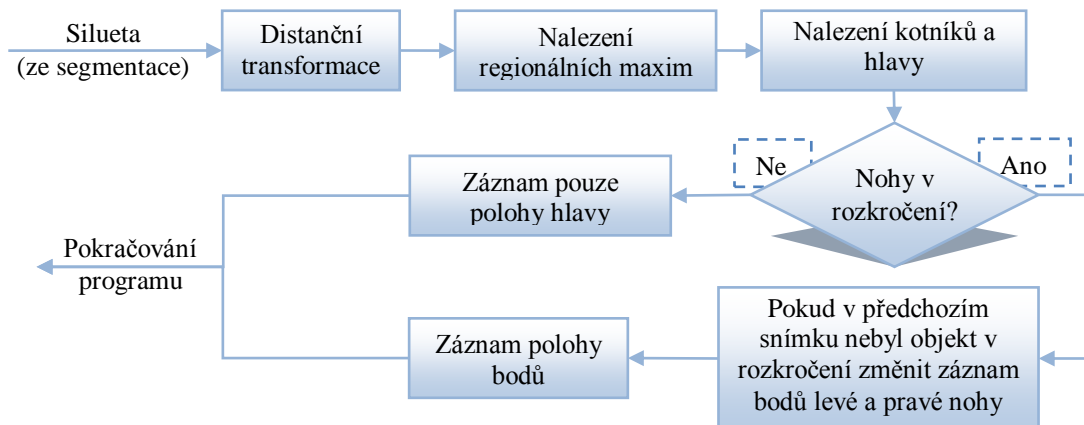
Tento obrázek je názornou demonstrací předchozí tabulky a můžeme si v něm povšimnout kvality rozpoznání objektu.



**Obrázek 5.12:** Porovnání výsledků segmentace na jednom z obrázků různými metodami: A) původní obrázek, B) rozdílové snímky, C) rozdílové snímky s transformací jasové stupnice, D) metoda k-mean s transformace jasové stupnice, E) metoda k-mean s transformace jasové stupnice a detekcí stínů, F) metoda mean shift s transformace jasové stupnice, G) metoda mean shift s transformace jasové stupnice a detekcí stínů

## 6. Nalezení důležitých bodů v siluetě

Při identifikaci nemůžeme porovnávat chůzi siluety každého člověka, tzn. každý bod v siluetě ve všech snímcích, protože by trvalo velmi dlouho než bychom porovnali pouhé dvě osoby. Proto v pohybující se siluetě musíme najít a zaznamenávat důležité body (např. klouby), které po vynesení do grafů pak můžeme porovnávat a určovat míru podobnosti s jinými průběhy. Na *obrázku 6.1* můžeme shlédnout blokové schéma algoritmu pro nalezení těchto bodů v jednotlivých siluetách. Vstupem do této metody je silueta, kterou získáme ze segmentace, a výstupem jsou pak jednotlivé body hlavy a kotníků.



**Obrázek 6.1:** Blokové schéma hledání důležitých bodů v siluetě

### 6.1 Distanční transformace siluety

Distanční transformace je metoda pro výpočet nejmenších vzdáleností od nenulových bodů v binárním obrázku. Pokud definujeme vstupní binární obraz  $P$  a výstupní obraz  $D_P$  celočíselných hodnot (tzn. distanční mapa) pak můžeme napsat, že výstupní obraz  $D_P$  je dán vztahem [11]:

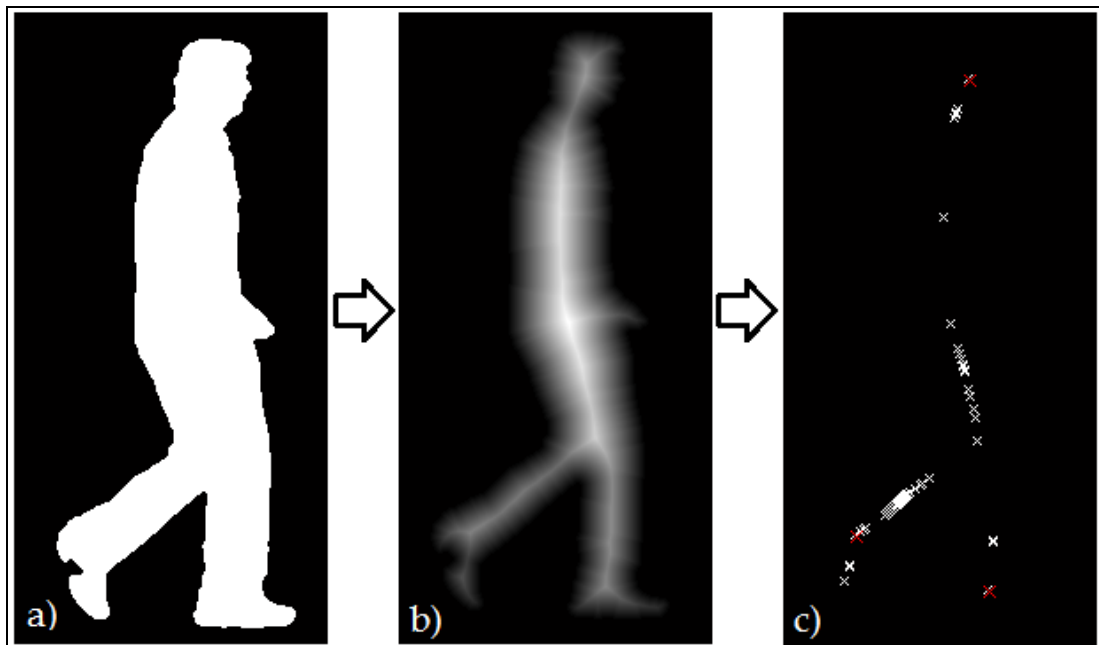
$$D_P(p) = \min_{q \in P} \|p - q\| \quad (19)$$

V prostoru  $\mathcal{R}^n$  jsou nejčastěji používané metriky pro aproximaci vzdálenosti bodů  $P$  a  $Q$  Euklidovská, Manhattan a Chessboard. Pro výpočet vzdálenosti však

používám první zmiňovanou, tj. Euklidovskou, pro kterou platí vztah pro výpočet vzdálenosti bodů  $P$  a  $Q$  [11]:

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + \dots + (p_n - q_n)^2} \quad (20)$$

Na obrázku 6.2b je ukázka výpočtu distanční mapy  $D_P$  pro siluetu z obrázku 6.2a. Můžeme na ní vidět nejjasnější místa s největší vzdáleností od nenulových bodů (okrajů).



Obrázek 6.2: a) Silueta, b) distanční transformace, c) regionální maxima

## 6.2 Nalezení regionálních maxim

Pohled na obrázek s distanční mapou, na které vidíme hřeben bodů s největší vzdáleností překrývající místa kostí člověka, mě přivedl k myšlence, abych použil funkci v programu Matlab `imregionalmax()` pro nalezení všech regionálních maxim v této distanční mapě. Tato funkce funguje tak, že v libovolně velké matici nebo obrázku najde pozici všech regionálních maxim. Například pro matici  $A$ :

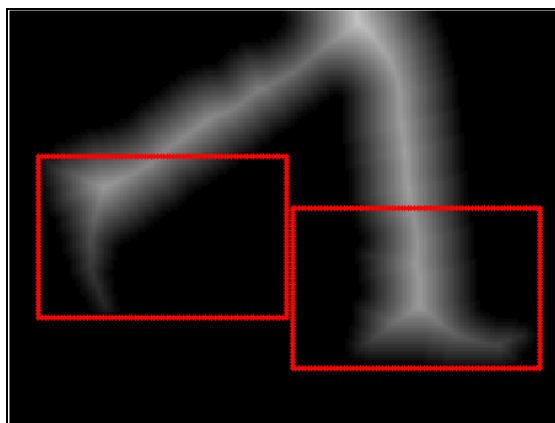


$$A = \begin{bmatrix} 10 & 10 & 20 & 10 & 0 & 0 \\ 20 & 30 & 30 & 10 & 10 & 0 \\ 10 & 10 & 10 & 10 & 10 & 10 \\ 20 & 10 & 10 & 20 & 20 & 10 \\ 10 & 20 & 30 & 45 & 20 & 10 \\ 10 & 20 & 20 & 20 & 20 & 10 \end{bmatrix} \quad \text{našel maxima: } B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Podobně byly nalezeny maxima pomocí této funkce v distanční mapě  $D_P$ , což můžeme vidět na *obrázku 6.2c*, kde tyto maxima představují bílé křížky.

### 6.3 Nalezení kotníků a hlavy

Funkce našla velké množství regionálních maxim, ze kterých bychom chtěli vybrat pouze ty, které se nacházejí na místě kotníků a uprostřed hlavy. Protože distanční vzdálenosti v místě kotníku se na všechny strany zmenšují, rozhodl jsem se použít filtraci s maskou  $M$  pouze v bodech regionálních maxim a pouze v oblastech, kde je výskyt kotníků nejpravděpodobnější. Tzn. od počátku každé nohy až po  $\frac{1}{7}$  celkové délky těla člověka. Tyto oblasti jsou zaznamenány jako červený čtverec na *obrázku 6.3*.



**Obrázek 6.3:** Oblasti hledání kotníků

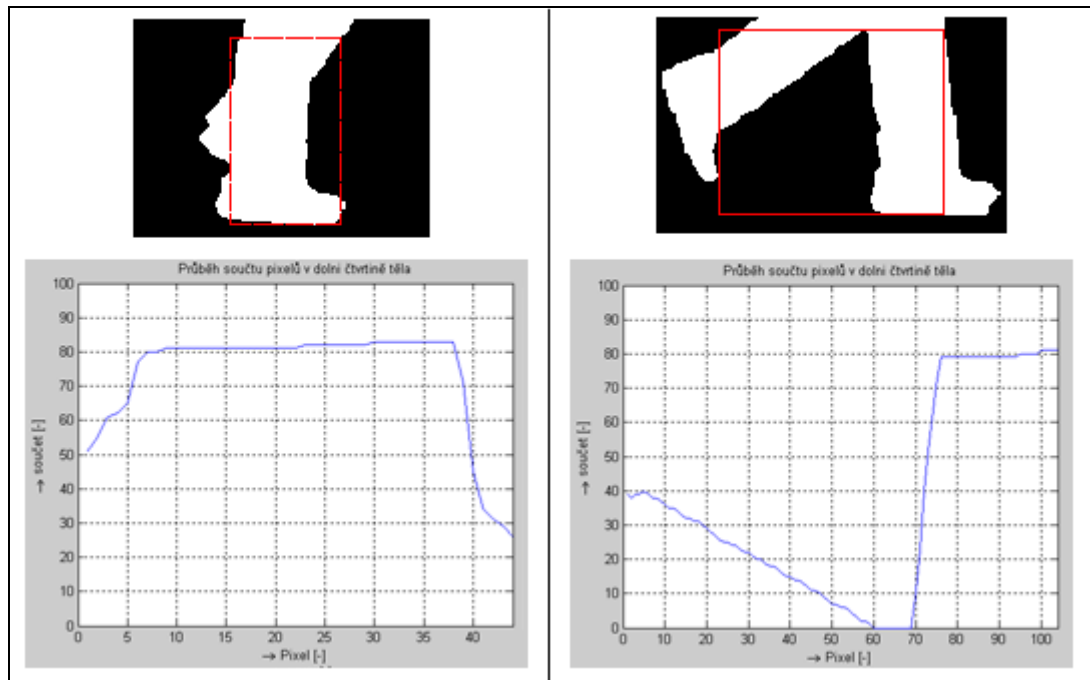
Použitá maska: 
$$M = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Tato maska má za cíl zviditelnit maxima, kolem kterých se nenacházejí jiná regionální maxima a distanční vzdálenosti kolem nich se ze všech stran zmenšují. Bohužel se taky zviditelnily místa kolem prsty, kde je obdobná situace jako u kotníků, ale hodnoty distančních vzdáleností jsou menší. Proto jsem přidal kritérium, pro které platí, že body (po filtraci maskou  $M$ ), které jsou menší než  $\frac{4}{5}$  maximální hodnoty v oblasti hledání, se odstraní. Výsledný bod, který označíme jako hledaný kotník, pak bude nejnižší ležící bod v hledané oblasti. Na *obrázku 6.2c* jsou výsledné nalezené body označeny červeným křížkem.

#### 6.4 Záznam pozice nohou

Protože výše popsaná metoda k vyhledávání kotníku v siluetě může fungovat jenom při viditelném oddělení nohou (tj. v rozkročení), musíme tedy během snímání určovat, jestli jsou nohy v rozkročení nebo se zakrývají. K tomu používám vyhledávání minima součtu pixelů ve vertikální ose v předem určené oblasti, ve které předpokládám, že se nachází nohy. Tato oblast a graf součtu pixelů jsou znázorněny na *obrázku 6.4*. Oblast se nachází v dolní čtvrtině těla a po bocích je ohraničena  $\frac{1}{6}$  aktuální šířky těla. Boční omezení je zavedeno z důvodu sporadicky nepřesné segmentace nohou nebo taky z důvodu, že by výška v místě palců byla menší než minimum potřebné k rozhodnutí o tom, jestli jsou nohy v rozkročení nebo se zakrývají (v programu je nastaveno  $\min = \frac{\text{výška siluety}}{60}$ ).

Pokud tedy program zjistí, že se nohy zakrývají, přestane ukládat hodnoty polohy kotníků do proměnné a převrátí zapisování těchto hodnot z pravého kotníku do levého a naopak. Pokud zjistí, že jsou nohy v rozkročení, začne hodnoty zase ukládat.

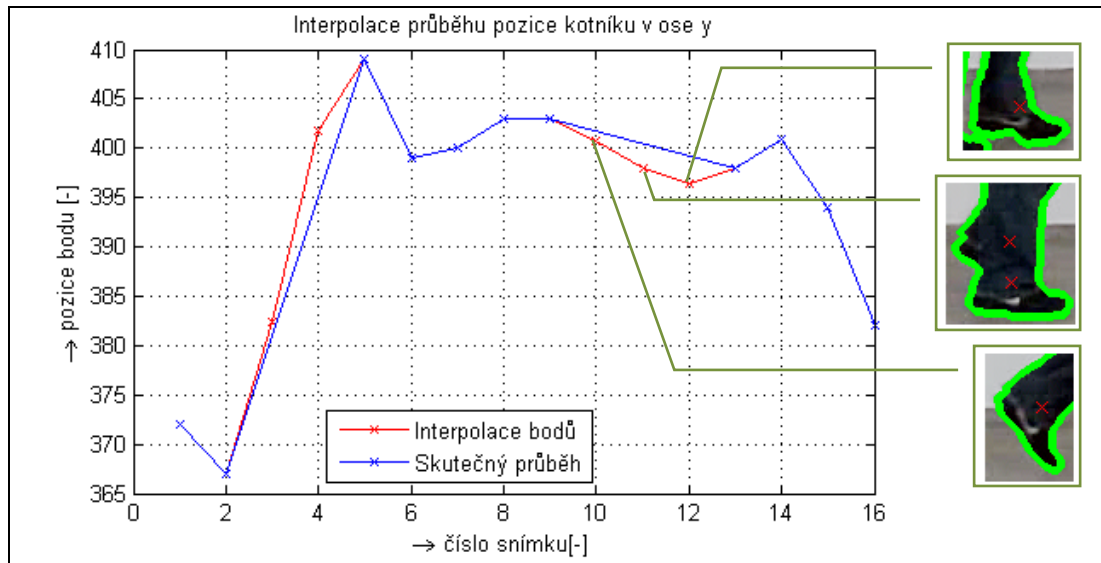


**Obrázek 6.4:** Obrázek znázorňující vertikální amplitudovou projekci v prostoru ohraničeném červenou čarou

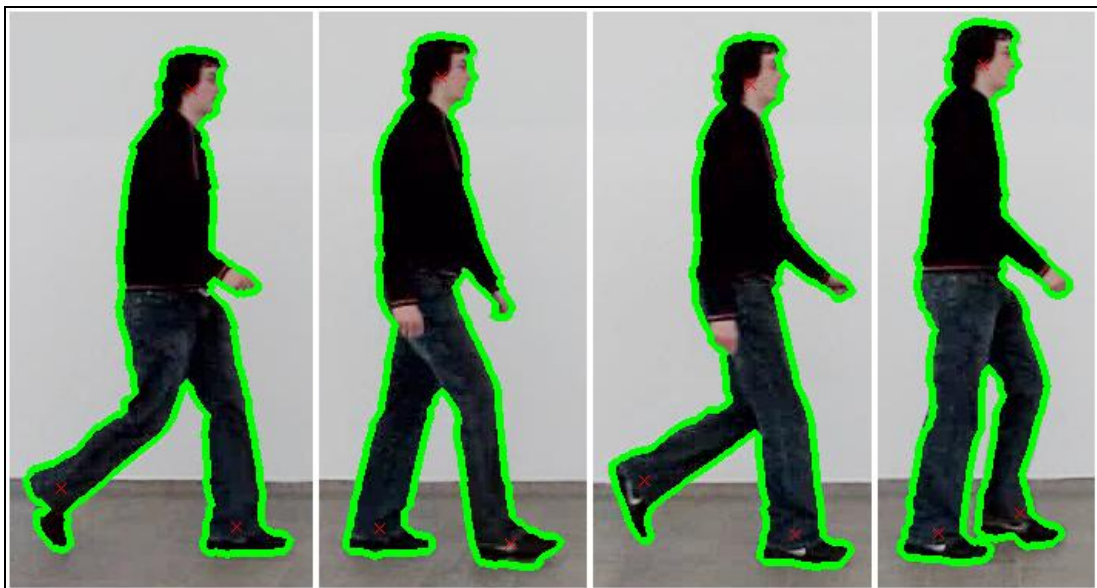
### 6.5 Interpolace bodů kotníků (situace zakrývajících se nohou)

V průběhu nalézání kotníků jsme v některých snímcích neukládali pozici kotníků. Tyto hodnoty musíme nějakým způsobem uhádnout nebo interpolovat, tzn. nalézt přibližné hodnoty těchto míst ze známého průběhu pozic kotníků z jiných snímků. Úkonu nalezení těchto bodů dosáhneme pomocí interpolace kubickým splinem funkcí *interp1()* v Matlabu. Na *obrázku 6.5* můžeme shlédnout ukázkou nalezeného průběhu pohybu kotníku v jednotlivých snímcích v ose *x* a interpolovaných bodů, které jsem doplnil obrázky nohou s nalezenými body kotníků označenými červeným křížkem.

Na *obrázku 6.6* je pak znázorněn výsledný obrázek s ohraničením hledaného objektu (zelená čára), a v něm nalezených míst kotníků a hlavy (červený křížek).



Obrázek 6.5: Ukázka interpolace průběhu pozice kotníku

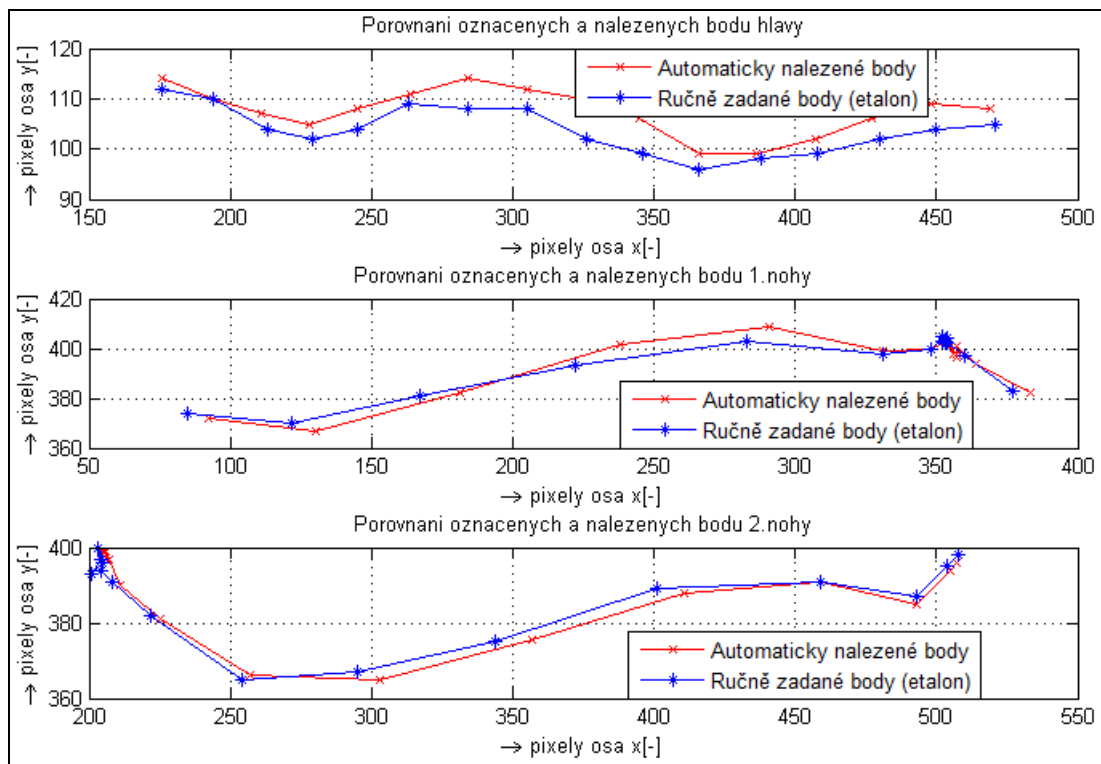


Obrázek 6.6: Obrázek s ohraničením objektu a vyznačením míst kotníků a hlavy

## 6.6 Porovnání nalezených bodů se skutečnými

Pro účel porovnání správného nalezení kotníků a hlavy jsem do programu implementoval funkci, pomocí které můžeme v průběhu segmentace označit

skutečná místa výskytu kotníků a středu hlavy. Získané pozice těchto míst můžeme dále porovnávat s nalezenými. Porovnání je znázorněno na *obrázku 6.7*, kde můžeme vidět tři srovnávací grafy: v každém po dva průběhy, ze kterých jeden (modrý) znázorňuje průběh bodů označených uživatelem a druhý (červený) automaticky nalezené programem. V tomto případě jsou rozdíly minimální, konkrétně jednotky pixelů. Kvalita nalezení však závisí na spoustě faktorů, kterými jsou např. nastavení parametrů používaných metod, výběr pozadí snímání a oblečení objektu atd. Tyto vlastnosti jsou popsány v kapitole 8.



**Obrázek 6.7:** Ukázka porovnání nalezených a označených míst kotníků a hlavy

## 7. Rozpoznávání osob pomocí získaných průběhů

Chceme-li rozpoznat konkrétní osobu pohybující se ve snímané scéně, potřebujeme nějakým způsobem porovnat její pohyby s předem připravenými srovnávacími daty. A protože srovnávání celého snímku nebo celého nalezeného objektu by bylo komplikované a nikam by nevedlo, je potřeba pohyby objektu popsat nějakým jednodušším způsobem. Proto jsem se zaměřil na vyhledávání kloubů v objektu a sledování jejich trajektorie. Následně z nich získám průběh, a tyto průběhy mohu spolu porovnávat a určovat, o kterou osobu se jedná.

### 7.1 Volba získání dat pro porovnávání

Získávání průběhů ze sekvence snímků zobrazujících chůzi člověka však můžeme provádět několika způsoby, z nichž některé jsou:

- Změna množství bodů v siluetě při použití masky

Jednou z možností pro vytvoření příznaků k rozpoznávání charakteristiky chůze je vytvořit masku a v ní počítat součet bodů jednotlivých siluet člověka. Různé masky jsou zobrazeny na *obrázku 7.1*.



**Obrázek 7.1:** Masky pro získání průběhu ze siluety [12]

Pro metodu musíme mít vytvořenou siluetu člověka, která musí být vycentrovaná na střed. Tzn. pokud máme binární obrázek (nula = pozadí, jedna = pohybující se objekt), pak při pohybu člověka (pohybující se snímky) musí být silueta pořád ve středu obrázku. Pak přiložíme zvolenou masku k pohybující se siluetě (součin siluety s maskou), a vypočteme součet všech jedniček ve vzniklém obrazu. Vzniklý průběh součtu bodů můžeme dále použít k porovnávání a identifikaci.

– Pohyby kloubů

Dalším způsobem k vytvoření potřebného průběhu pro porovnávání je najít v hledaném objektu umístění kloubů a sledovat jejich pohyb. Můžeme pak vytvořit průběh těchto pohybů nebo pohyby kombinovat, například počítat rozdíl umístění kotníků levé a pravé nohy.

– Změny úhlů vytvořené mezi nalezenými body

Jiný postup jak vytvořit průběh pro porovnávání je snímání úhlů vytvořené pomocí úseček spojujících nalezené klouby na těle hledaného objektu.

Nalezení umístění kloubů na těle objektu popisuje kapitola 6. Budeme tedy předpokládat, že máme vypočítané průběhy pohybu kloubů v obrazu.

## 7.2 Standardizace dat

Při snímání chůze různých lidí v různém prostředí není umístění nalezených kloubů, vztahující se k hornímu levému rohu obrázku jednotná, a je třeba je upravit tak, aby všechny byly souměřitelné. Jedním ze způsobů jak toho docílit, je standardizace těchto dat. Pokud mámě naměřené umístění kloubů v obrazu  $z$ , můžeme vypočítat standardizované hodnoty podle rovnice [16]:

$$x = \frac{z - \bar{z}}{s} \quad (21)$$

Přičemž  $\bar{z}$  je střední hodnota:

$$\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i \quad (22)$$

a  $s$  je směrodatná odchylka:

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})^2} \quad (23)$$

Standardizovaná data  $x$  nyní mají střední hodnotu rovnou 0 a rozptyl 1.

### 7.3 Porovnávání průběhů a identifikace osob

Po standardizaci je nutné vymezit počet kroků pro porovnávání, a také musíme v datech určit počátek a konec tohoto počtu kroků. Tento úkon byl již vyřešen přímo při snímání, neboť pomocí maximálních hodnot délky kroku zjišťujeme rozkročení a tímto určujeme, kolik snímaný objekt udělal kroků (pokud program detekuje tři maxima, znamená to dva plné kroky). A pokud nasnímáme celé dva kroky, přestaneme zaznamenávat a program přejde k dalšímu kroku. Tato funkce je blíže popsána v kapitole 5.2.

Jedním ze způsobů vyjádření podobnosti dvou průběhů je metrika vycházející z geometrického modelu dat. Při použití pro dva 2-rozměrné průběhy  $(r,s)$  stejné délky je definována euklidovská vzdálenost [16]:

$$\delta(r,s) = \sqrt{\sum_{i=1}^p (x_{ri} - x_{si})^2} \quad (24)$$

Hodnota  $\delta(r,s)$  je větší pro průběhy, které se sobě nerovnjají. Tzn. čím je tato hodnota menší, tím je větší podobnost dvou průběhů. A pokud  $\delta(r,s)=0$ , znamená to, že signály  $r$  a  $s$  jsou identické.

Před vypočítáním euklidovské vzdálenosti dvou signálů musíme zajistit jejich stejnou velikost (stejný počet vzorků). Odlišné množství bodů v průběhu může být způsobeno jak rozdílnou rychlostí snímání (způsobeno parametrem fotoaparátu vyjadřujícím počet snímků za sekundu), tak například rychlostí chůze člověka. Chceme-li tedy porovnat dva různé průběhy s odlišným počtem vzorků pomocí rovnice uvedené výše, musíme interpolací nahradit průběh s menším počtem vzorků průběhem, který je stejného tvaru, akorát je rozdělen na více vzorků, konkrétně tolik kolik vzorků má druhý průběh. Interpolace je provedena kubickým splinem funkcí v Matlabu *interp1()*.



**Tabulka 7.1:** Porovnání průběhů pohybů bodů chůzí různých osob

porovnání dvou chůzí		porovnání podle části těla		
		hlavy	levé nohy	pravé nohy
<b>Michal (chůze 2)</b>	Michal (chůze 1)	3.51	0.90	1.76
<b>Michal (chůze 2)</b>	Jakub (chůze 1)	2.29	3.06	1.30
<b>Michal (chůze 2)</b>	Adam (chůze 1)	3.16	6.16	5.55
<b>Jakub (chůze 2)</b>	Michal (chůze 1)	5.25	5.49	6.52
<b>Jakub (chůze 2)</b>	Jakub (chůze 1)	4.97	5.17	6.46
<b>Jakub (chůze 2)</b>	Adam (chůze 1)	5.71	6.49	2.47
<b>Adam (chůze 2)</b>	Michal (chůze 1)	2.48	6.41	6.21
<b>Adam (chůze 2)</b>	Jakub (chůze 1)	1.42	5.65	6.17
<b>Adam (chůze 2)</b>	Adam (chůze 1)	2.40	1.41	2.32

V *tabulce 7.1* je v prvních dvou sloupcích uvedeno jaké dvě chůze osob se porovnávaly. V pravých třech sloupcích jsou hodnoty vypočítané podle rovnice 24 vždy pro dva porovnávané průběhy pro určitý druh umístění důležitých bodů v nalezeném pohybujiícím se objektu (člověku). Porovná se podle bodů nalezených na hlavě a na levé a pravé noze.

Z tabulky můžeme usoudit, že nejlepší rozpoznávací příznak, který správně identifikoval všechny tři chůze osob (tzn. měl nejmenší hodnotu  $\delta(r,s)$  - nejvíce shodné průběhy), je porovnání podle průběhu pohybu levé nohy. Vždy nejmenší hodnoty jsou v tabulce označeny světle modrým pozadím. Je však vidět, že výsledek není jednoznačný. Pro zlepšení rozpoznávání by bylo nutné najít příznak nebo kombinaci příznaků, které by jasně určily osobu, která byla natočena.

## 8. Problémy při získávání obrysu

Bezchybná segmentace obrazu a běh programu jsou bohužel ve většině případů doménou laboratorního zkoušení programu. Existují tedy situace, které tento program nezvládá. Jedny z nejzávažnějších jsou:

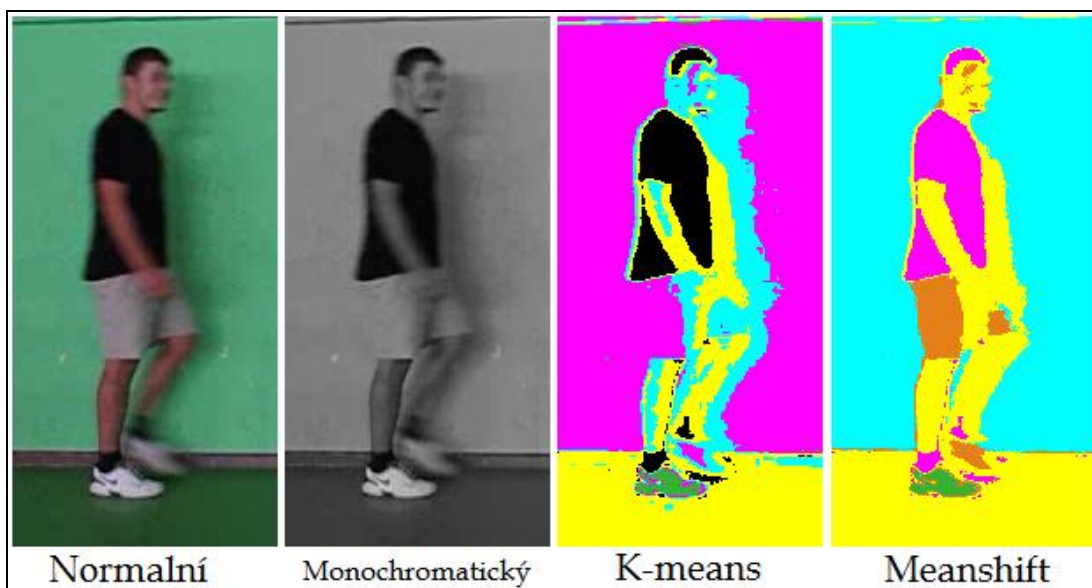
### 8.1 Nesprávné nasnímání scény

**Pomalá uzávěrka fotoaparátu** – jak se zmiňuji v kapitole 3., pro správné nasnímání scény je potřeba vhodně nastavit parametry fotoaparátu. Je proto důležité nastavit je tak, aby při snímání chůze výsledné video zachycovalo ostré detaily pohybu nohou a rukou i při poměrně rychlé chůzi. Aby se tak stalo, musíme nastavit co nejmenší uzávěrku fotoaparátu a tím zkrátit dobu snímání jednoho obrázku. Bohužel je to na úkor jasů scény, která tím pádem musí být co nejlépe nasvícená. Rozmazání nohy z důvodu dlouhého snímání obrázku můžeme shlédnout na *obrázku 8.1*.



**Obrázek 8.1:** Rozmazání části těla při dlouhé uzávěrce fotoaparátu

**Nevhodná barva pozadí** – algoritmus *k-means* v programu je navržen tak, že nejprve obrázek s objektem přetransformuje na šedotónový obrázek a pak teprve rozpoznává a rozděluje jednotlivé body do shluků. Bohužel některé odlišné barvy mohou po přechodu na šedotónový obrázek vypadat úplně stejně a tím ztrácíme povědomí o hranicích objektu. Podobná situace je znázorněna na *obrázku 8.2*, kde se po přechodu slévá barva kraťasů s pozadím, a tak metoda *k-means* je vyhodnotí jako pozadí. Jinak je na tom metoda *mean shift*, která operuje s *rgb* barvami bodů.



**Obrázek 8.2:** Porovnání metod při nevhodné barvě pozadí

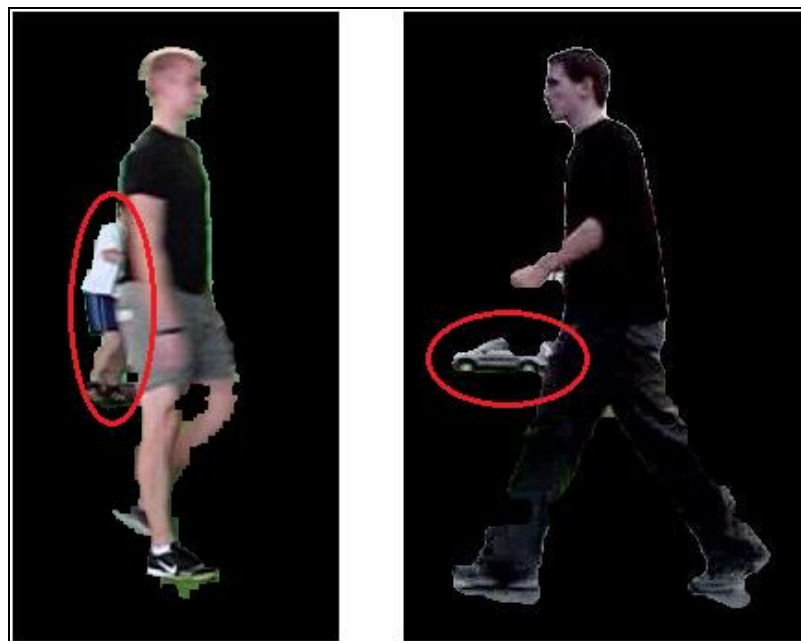
Kvůli nevhodné barvě pozadí se objevuje ještě jeden problém. V případě pozadí bílé barvy, která má hodnoty v barevném obrázku  $rgb=(255,255,255)$ , nebo černé, která má  $rgb=(0,0,0)$  (hodnoty nemusí být 255 nebo 0, ale velmi se těmito hodnotám přibližují), jsou hodnoty rozdílových bodů pozadí a objektu, oblečeného jinak než je barva pozadí, o mnoho větší než kdyby pozadí mělo nějaký barevný nádech. Příčina je v *rgb* hodnotách barev pozadí a objektu. Jak je vidět na předchozím obrázku máme zelené pozadí s hodnotami  $rgb_p=(90,180,120)$  a například hodnoty barvy pleti objektu  $rgb_o=(140,110,110)$ . Součet jejich absolutního rozdílu bude menší:

$|rgb_p - rgb_o| = 130$ , než kdyby pozadí bylo bílé ( $rgb_{bílá} = [240, 240, 240]$ ):

$|rgb_p - rgb_o| = 360$ . Proto by se optimální barva pozadí měla blížit bílé.

## 8.2 Nevhodný výběr oblečení a scény snímání

**Pohyb jiného většího objektu, nebo objektu, který se pohybuje za člověkem, ale je sním na obrázku hraničně spojen** – Na některých zkoušených sekvencích se vyskytlo pohybující se cizí těleso, které změnilo tvar siluety člověka (příklad na *obrázku 8.3*), nebo se objevilo jako další snímáný objekt. Tento jev se vyskytl v důsledku toho, že program je primárně navržen na scénu, kde se pohybuje pouze člověk.



**Obrázek 8.3:** Pohyb jiných objektů ve scéně nesouvisejících s hledaným objektem

**Velmi podobná barva objektu a pozadí** – Metoda rozdílových snímků tvoří jednu z hlavních rozpoznávacích algoritmů. A jelikož porovnává jasovou hodnotu dvou snímků (modelu a obrázku s pohybem), musí být jas bodů pohybujícího se objektu odlišný od pozadí. Pokud je tedy podobný nebo stejný, program tyto body ohodnotí jako pozadí. Obdobné je to z metodami *mean shift* a *k-means*, které shlukují

podobné barvy či podobné jasové hodnoty černobílého snímku. Jestli je barva pozadí stejná nebo velmi podobná barvě oblečení, ohodnotí tyto metody objekt jako shluk barev pozadí. Je to podobné do situace, ve které se voják maskuje tak, aby ho nikdo nezpozoroval ani tehdy, když se na něj dívá. A proto je pochopitelné, že si s tím neví rady žádná z navrhovaných metod. Tento jev je znázorněn na *obrázku 8.4*.

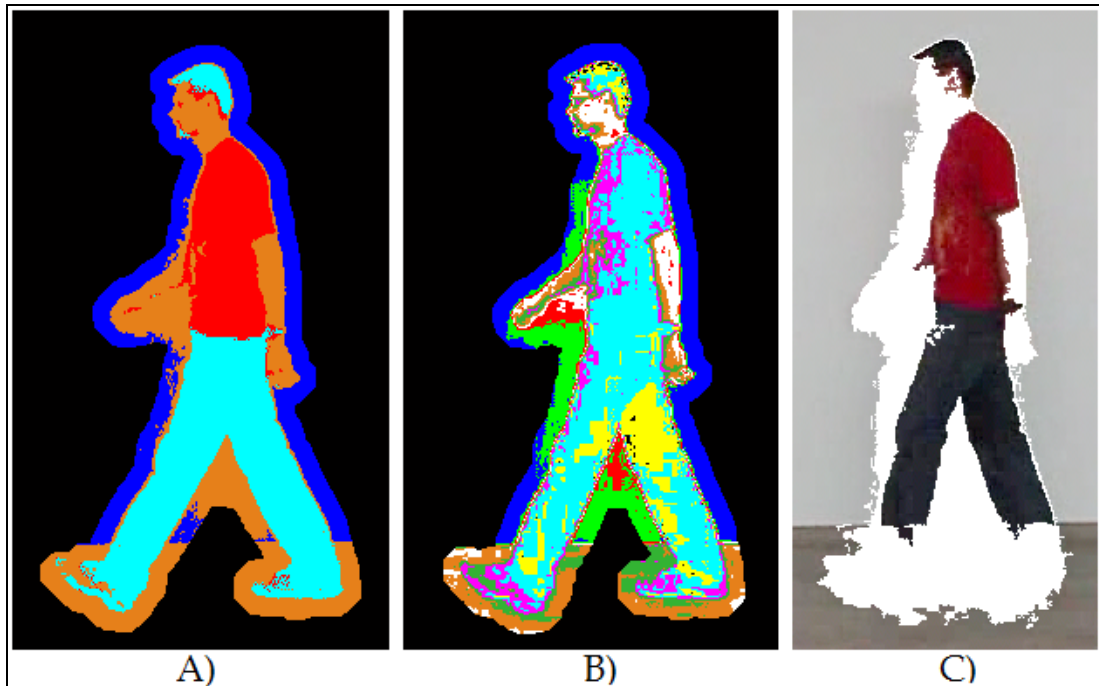


**Obrázek 8.4:** Výsledek rozdílového snímku při podobné barvě pozadí a oblečení

### 8.3 Problémy metod

Program je navržen tak, aby se co nejvíc přizpůsobil požadavkům uživatele. To znamená, že ten kdo s programem pracuje, musí pro správné rozpoznání člověka v sekvenci snímků vhodně nastavit parametry metod, které chce použít. Tento fakt je asi největším problémem, pokud se uživatel neorientuje v algoritmech metod a neví, co určitý parametr způsobuje. Nevhodné nastavení může například rozčlenit objekt na příliš mnoho shluků, přičemž později nebudeme moci rozhodnout, co je pozadí a co objekt (*Obrázek 8.5B*). Nebo nepřiměřeně nastavit detekci stínů tak, že stíny budou pokrývat velkou část těla člověka, která je následně programem vymazána (*Obrázek*

8.5C). Při vysoké hodnotě parametru velikosti okolí v metodě *mean shift* můžeme dosáhnout započtení do objektu bodů patřících z části k pozadí (Obrázek 8.5A).



**Obrázek 8.5:** Nevhodně nastavené parametry metod A) *mean shift*, B) *k-means*  
C) detekce stínů

Proto je velmi důležité být obeznámen v těchto metodách segmentace a vědět, jaký důsledek má zvětšení nebo zmenšení různých parametrů.

#### 8.4 Problémy při hledání kloubů

Protože hledání umístění kloubů hledaného objektu počítá s dobře provedenou segmentací a vytvořením přesné siluety, je většina problémů, které by se v této části mohly vyskytnout, spojena právě s nalezenou siluetou, která přesně odpovídá výskytu pohybujícího se člověka v sekvenci snímků. Ale při algoritmu nalézání umístění důležitých bodů (kloubů) pro porovnávání mohou nastat i jiné peripetie, které nezávisí na kvalitní siluetě, a na kterých závisí pozdější identifikace. Jsou to například tyto problémy:

– Brzké nalezení rozkročení

Tento problém může nastat v situaci, když se při rozkročení najde minimum v nesprávné chvíli. Dojde tak k záměně ukládání bodů levé a pravé nohy a celý průběh se tím rozladí. Tím pádem nalezené body určují úplně jiná místa, ve kterých by se ve skutečnosti měly nacházet klouby kotníků.

– Nepřesná interpolace

Problém vzniká v situaci, když musíme interpolovat více bodů vedle sebe při nízkém počtu všech nalezených bodů v průběhu. Taky nastává, pokud na okraji průběhu známe jeden nebo dva body a musíme nalézt další tři nebo více bodů v průběhu. Výsledné interpolované body leží sice v blízkosti skutečného místa kloubů, ale tato vzdálenost je příliš velká pro pozdější hledání podobného průběhu.

– Nalezení jiného maxima

Tento problém vlastně závisí na přesném nalezení siluety. Nastává při výskytu větších deformací siluety v místech výskytu hlavy a kotníků. Nalezené maximum pak neodpovídá lokálnímu maximu, kde by se ve skutečnosti mělo nacházet místo kotníků nebo středu hlavy.

Problémy se vyskytují v každé části algoritmu, proto je cesta ke konečné fázi porovnávání průběhů a identifikaci osob velmi dlouhá a ne všechny metody jí překonají bez překážek.

## 9. Závěr

Práce se zabývá možnostmi identifikace osob pomocí průběhů pohybů bodů nalezených ze segmentovaného obrazu, které vyjadřují charakteristiku chůze. Algoritmus pracuje s nahrávkami chůze snímané z profilu.

Nejprve byly prozkoumány pohyby kloubů člověka pro zlepšení vědomí o jeho pohybu. Následně bylo navrženo rozložení snímací scény. V další části práce bylo prozkoumáno několik algoritmů (popsaných v kapitole 4), které správně segmentují obrázek a najdou siluetu člověka v sekvenci snímků. Nejlepší výsledky však vykazují metody *k-means* a *mean shift* ve spolupráci s předzpracováním (transformace jasové stupnice) a následným nalezením a odstraněním stínů. Názorné ukázky segmentovaných obrázků jsou ke zhlédnutí v kapitole 5.3. Výpočty shlukových metod jsou urychleny tím, že se nejprve najde přibližné místo výskytu hledaného objektu pomocí rozdílové metody estimace modelu prostředí. Po získání siluety jsou nalezeny důležité body (v našem případě střed hlavy a klouby kotníků obou noh), jejichž pohyb je v sekvenci snímků zaznamenáván. V práci bylo uskutečněno i porovnání nalezeného a skutečného (vyznačeného uživatelem) průběhu pohybu těchto bodů. Jak je vidět na *obrázku 6.7* se tyto průběhy liší minimálně. Následně byla vyzkoušena identifikace chůze různých osob na základě porovnávání dvou průběhů. Z výsledků můžeme usoudit, že lepší rozpoznávací vlastnosti mají průběhy pohybu noh, konkrétně levé nohy. Výsledky identifikace však nejsou jednoznačné. Pro zlepšení rozpoznávání by bylo nutné najít příznak nebo kombinaci příznaků, které by přesněji určovaly osobu, která byla natočená. V poslední části jsou vysvětleny všechny problémy, které se vyskytly při řešení práce.

Při řešení bylo zjištěno, že pro správné rozčlenění snímku na pozadí a hledaný objekt, je důležité korektně nastavit parametry fotoaparátu nebo kamery a dobře rozmístit scénu, a taky optimálně nastavit parametry použité metody segmentace vzhledem k natočené scéně.

Přestože byla hledána rychlá segmentační metoda pro nalezení objektu, je celková doba výpočtu algoritmu v programu Matlab pomalá. Východiskem by bylo



přepsání algoritmu pomocí programovacího jazyka C++/C při použití knihovny OpenCV.

Výsledkem práce je program, který v sekvenci snímků představující chůzi nalezne siluetu člověka, vyhledá umístění vztažných bodů (středu hlavy a kloubů kotníků) a uloží je na disk. Získané průběhy pak můžeme porovnávat s jinými průběhy charakterizujícími chůzi různých osob. Výstupem jsou pak uložené průběhy pohybů těchto bodů. Vytvořená funkce *porovnat()* porovná dva zadané průběhy a určí jejich podobnost. Program lze spustit v Matlabu a nachází se na přiloženém DVD spolu s vytvořenými nahrávkami chůze různých osob.

## 10. Literatura

- [1] Haladová E., Nechvátalová L., *Vyšetřovací metody hybného systému*, vydal Institut pro další vzdělání pracovníků ve zdravotnictví Brno, 1997, 137 s., ISBN 80-7013-237-X
- [2] Horák K.: *Dynamické obrazy, Přednáška MPOV* [přednáška online], UAMT FEKT VUT Brno, 2008, dostupná z :  
<<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html>>  
[cit. 2010-5-7]
- [3] Matteucci M., *K-Means Clustering (a tutorial on clustering algorithms)* [tutoriál online], dostupné z:  
<[http://home.dei.polimi.it/matteucc/Clustering/tutorial\\_html/kmeans.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html)>  
[cit. 2009-12-11]
- [4] Hlaváč V., Sedláček M. *Zpracování signálů a obrazů*. Praha : ČVUT, 2005. 255 s. ISBN 80-01-03110-1.
- [5] Doubek P., *Mean-Shift segmentace* [online], CMP FEL CVUT Praha, říjen 2007, dostupné z: [http://cmp.felk.cvut.cz/cmp/courses/ZS1/Cviceni/cv4/mean\\_shift.pdf](http://cmp.felk.cvut.cz/cmp/courses/ZS1/Cviceni/cv4/mean_shift.pdf)  
[cit. 2009-5-16]
- [6] Moore W.A., *K-means and Hierarchical Clustering* [online], 2001, dostupná z:  
<<http://www.autonlab.org/tutorials/kmeans11.pdf>> [cit. 2009-12-11]
- [7] Tuzel O., Meer P., *Mean Shift Clustering*, dostupná z:  
<[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/TUZEL1/Mean\\_shift.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TUZEL1/Mean_shift.pdf)> [cit. 2009-12-11]
- [8] Comaniciu D., Meer P., *Mean shift: A robust approach toward feature space analysis*. IEEE Trans. Pattern Anal. Machine Intell., 24:603–619, 2002.  
Dostupná z: <<http://www.caip.rutgers.edu/riul/research/papers/pdf/mnshft.pdf>>  
[cit. 2009-5-16]
- [9] Chen B., Chen D., *Shadow Detection Based on rgb Color Model*, Department of Computer Science of Huaqiao University, 2006.
- [10] MacQueen J. B., *"Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical*

*Statistics and Probability*", 1967, Berkeley, University of California Press,  
1:281-297

- [11] Horák K.: *Detekce a klasifikace objektů, Přednáška MAPV* [přednáška online], UAMT FEKT VUT Brno, 2008, dostupná z:  
<<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MAPV/mapv.html>>  
[cit. 2010-5-7]
- [12] J. P. Foster, M. S. Nixon, A. Prugel-Bennett, *NewArea Based Metrics for Gait Recognition*, University of Southampton, Southampton, 2001, SO17 1BJ, UK, Dostupná z: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.8280>>  
[cit. 2010-5-15]
- [13] Havlík, J., *Klasifikace pohybu části lidského těla v prostoru*. Praha, leden 2008. 87 s. Disertační práce na Fakultě elektrotechnické Českého vysokého učení technického v Praze na katedře teorie obvodů. Školitel: Prof. Ing. Jan Uhlíř, CSc. Dostupná z [cit. 2010-5-15]:  
<[http://amber.feld.cvut.cz/user/xhavlikj/documents/2008\\_01.pdf](http://amber.feld.cvut.cz/user/xhavlikj/documents/2008_01.pdf)>
- [14] Parks, D., Gravel, J. P. *Harris/Plessey Operator* [publikace online]. Dostupná z: <<http://www.cim.mcgill.ca/~dparks/CornerDetector/harris.htm>> [cit. 2010-5-15]
- [15] Dražanský, M. *Přehled biometrických systémů a testování jejich spolehlivosti*. In Kongres Bezpečnosti sítí. Praha, 11.4.2007. Dostupné z: <[http://data.security-portal.cz/clanky/113/odborne\\_prednasky/Prezentace.pdf](http://data.security-portal.cz/clanky/113/odborne_prednasky/Prezentace.pdf)> [cit. 2010-5-15]
- [16] Kelbel, J., Šilhán, D. *Shluková analýza* [článek online]. Dostupná z: <<http://staff.utia.cas.cz/nagy/skola/Projekty/Classification/ShlukovaAnalyza.pdf>> [cit. 2010-5-15]
- [17] Macenauer, A., *Základní charakteristiky objektivů I - Lek. 8* [seriál online]. Dostupná z: <<http://www.fotoaparar.cz/article/5007/1>> [cit. 2010-5-15]
- [18] Pihan, R., *Objektivy, jak vybrat a používat* [článek online]. 10.1.2006. Dostupná z: <[http://www.fotografovani.cz/art/tech\\_vybirame\\_jak/rom\\_lenses1.html](http://www.fotografovani.cz/art/tech_vybirame_jak/rom_lenses1.html)> [cit. 2010-5-15]