

# Čestné prohlášení

Prohlašuji, že jsem zadanou diplomovou práci zpracoval sám s přispěním vedoucího práce a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo její části se souhlasem katedry.

Datum: 12. 5. 2014

.....

Podpis

# Poděkování

Na tomto místě bych rád poděkoval panu Ing. Stanislavu Vítkovi, PhD., vedoucímu diplomové práce, který mi byl po celou dobu nápomocen, za jeho trpělivost a ochotu, za jeho užitečné rady a připomínky.

Dále bych chtěl poděkovat mé rodině a to především mým rodičům za to, že mi umožnili tuto vysokou školu studovat.

# Anotace

**Jméno:** Jaromír Hoško

**Název diplomové práce:** Bezpečnostní kamerové systémy

Cílem této diplomové práce je popsat způsoby návrhu a optimalizace bezpečnostních kamerových systémů. Teoretická část práce je zaměřena na popis vlastností a parametrů objektivů a na metody detekce objektu z kamerového záznamu. Praktickou částí práce je návrh aplikace v programu Matlab. Úkolem této aplikace je monitorování prostoru, detekce pohybu a sledování pohybujícího se objektu.

**Klíčová slova:**

Bezpečnostní kamerové systémy, detekce objektu, tracking, Matlab

# Abstract

**Name:** Jaromír Hoško

**Title of thesis:** Surveillance camera systems

The aim of this thesis is to describe the methods of design and optimization of surveillance camera systems. The theoretical part of this thesis focuses on the description of the lens characteristics and parameters and also on the object detection methods from the camera record. The practical part of this thesis is designing application in Matlab. The goal of this application is the monitoring of area, motion detection and tracking of a moving object.

**Index Terms:**

Surveillance camera systems, object detection, tracking, Matlab

# Obsah

Obsah .....	1
Seznam obrázků .....	3
1. Úvod .....	5
2. Komerové systémy .....	6
2.1 Vlastnosti kamerových systémů .....	8
2.2 Fixní kamery .....	9
2.3 PTZ Kamery .....	9
3. Vlastnosti objektivů.....	11
3.1 Ohnisková vzdálenost a zorný úhel.....	11
3.2 Měnitelný rozsah ohniskové vzdálenosti – ZOOM .....	12
3.3 Clonové číslo a světelnost objektivu.....	13
3.4 Hyperfokální vzdálenost.....	14
4. Metody detekce a klasifikace objektu .....	15
4.1 Odečtení snímků .....	16
4.2 Odečtení pozadí.....	17
4.3 Odečítání předzpracovaného pozadí.....	18
4.4 Detekce pomocí masky.....	19
4.5 Porovnání metod .....	20
4.6 Viola – Jones .....	21
4.7 Klasifikace objektu .....	23
4.7.1 Pomocí masky.....	24

5.	Návrh kamerového systému.....	25
5.1	Technické parametry .....	25
5.2	Připojení a instalace a kamery.....	26
5.3	Ovládání kamery .....	27
5.3.1	Základní ovládání.....	27
5.3.2	Ovládání přes webový prohlížeč.....	28
5.3.3	Ovládání pomocí Matlab aplikace.....	29
6.	Aplikace na monitorování prostoru .....	31
6.1	Statická kamera, pohybující se objekt .....	32
6.2	Pohybující se kamera .....	33
6.2.1	Dopočítávání pozadí .....	33
6.2.2	Načtení pozadí na začátku .....	35
6.3	Popis funkce.....	36
6.4	Popis GUI .....	38
6.5	Další možnosti aplikace.....	39
6.5.1	Určení rychlosti a vzdálenosti objektu od kamery .....	39
6.5.2	Klasifikace objektu podle barvy.....	40
7.	Závěr.....	42
8.	Seznam použité literatury a zdrojů.....	44
9.	Přílohy .....	46

# Seznam obrázků

Obr. 1 - Schéma kamerového systému.....	7
Obr. 2 - Monitorovací pracoviště [13.9] .....	8
Obr. 3 - Fixní dome kamery [13.10] .....	9
Obr. 4 - PTZ dome kamera a ovládací pult [13.9] .....	9
Obr. 5 - Vztah objektu a objektivu .....	11
Obr. 6 - Zorný úhel v závislosti na ohniskové vzdálenosti [13.14].....	12
Obr. 7 - Clona a clonové číslo [13.5] .....	13
Obr. 8 - Tabulka hyperfokálních vzdáleností .....	14
Obr. 9 - Odečítání po sobě jdoucích snímků .....	16
Obr. 10 - Odečtení pozadí.....	17
Obr. 11 - Odečítání předzpracovaného pozadí .....	18
Obr. 12 - Princip detekce pomocí masky [7] .....	19
Obr. 13 - Detekce pomocí masky .....	20
Obr. 14 - Porovnání metod na detekci objektu .....	20
Obr. 15 - Princip Viola - Jones detekce [5] .....	22
Obr. 16 - Výsledky detekce obličeje Viola - Jones metodou .....	22
Obr. 17 - Úprava obrázku .....	23
Obr. 18 - Masky pro různé objekty [11] .....	24
Obr. 19 - Kamera Vivotek PT8133 [14].....	25
Obr. 20 - Okno Installation wizard.....	26
Obr. 21 - Základní ovládání.....	27
Obr. 22 - Ovládací GUI.....	30
Obr. 23 - Monitorovaný prostor se dvěma kamerami .....	31
Obr. 24 - Výsledky při statické kameře.....	32
Obr. 25 - Princip dopočítávání pozadí.....	34
Obr. 26 - Zkreslení obrázku při dopočítávání.....	34
Obr. 27 - Načtená pozadí .....	35

Obr. 28 - Princip funkce aplikace.....	36
Obr. 29 - Zorná pole kamer při předání.....	37
Obr. 30 - GUI.....	38
Obr. 31 - Určení vzdálenosti objektu.....	39
Obr. 32 - Barevný prostor HSV [13.13].....	40
Obr. 33 - Barevné charakteristiky obrázků.....	41



# 1. Úvod

Tato práce se zabývá kamerovými systémy. Ve druhé kapitole budeme popisovat vlastnosti důležité pro návrh kamerového systému. Dále zde rozebereme druhy kamerových systémů a možnosti, které kamerové systémy nabízejí.

Protože objektiv je nedílnou součástí kamery, budou ve třetí kapitole popsány důležité vlastnosti objektivů. Bude se zde rozebírat, jaký vliv na výsledný obraz mají udávané parametry objektivů. Konkrétně se zaměříme na ohniskovou vzdálenost a zorný úhel, na clonové číslo a světelnost objektivu a na hyperfokální vzdálenost objektivu, která se využívá při ostření.

Ve čtvrté kapitole se již zaměříme na zpracování obrazu z kamery. Jsou zde popsány základní metody na detekci objektu v obraze a klasifikace detekovaného objektu. Vybrali jsme metody odečtení snímků, odečtení pozadí, odečtení upraveného pozadí a metodu detekce pomocí masky. Dále je zde porovnání těchto metod a představení některých pokročilejších metod na detekci objektu.

Práce se zabývá návrhem kamerového systému, proto v páté kapitole rozebereme základní technické parametry kamery Vivotek PT8133, kterou jsme pro návrh používali. Dále zde budou popsány způsoby ovládání této kamery.

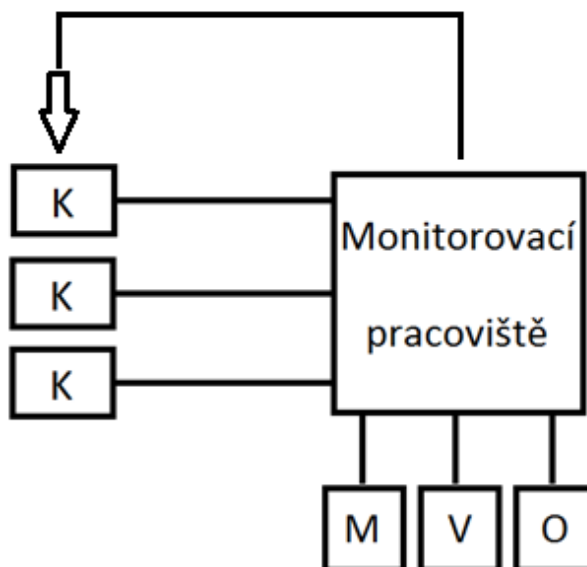
Nejdůležitější z celé práce je její praktická část. Popisu praktické části je věnována předposlední, šestá kapitola. Zabývali jsme se praktickým návrhem aplikace, která bude sledovat určitý prostor a detekovat pohyb. Na základě detekce pohybu bude aplikace posílat kameře příkazy a bude tak ovládat pohyb kamery.

## 2. Kamerané systémy

Kamerané systémy (CCTV - Closed Circuit Television, uzavřený televizní okruh) se dnes používají ve velkém měřítku a jejich popularita stále roste. Kamerané systémy jsou významným pomocníkem v boji proti kriminalitě. V dnešní době tvoří nedílnou součást ochrany života, zdraví, majetku a mají také významný preventivní účinek. Tyto systémy se dnes využívají pro monitorování nejrůznějších objektů a pozemků a k následnému uchování záznamů na záznamová zařízení. Často se využívají i k ochraně rodinných domů a efektivně doplňují elektronické zabezpečovací systémy.

Na začátek by bylo vhodné říci, co to vlastně je kamerový systém. Jedná se o uzavřený okruh několika kamer, které používáme k trvalému monitorování určitého prostoru. Součástí kamerového systému jsou kamery, které prostor sledují, monitory, na kterých uživatel vidí záběry z kamer a ovládání nebo software, který celý proces řídí. V této práci se budeme zabývat pouze digitálními kamerovými systémy, které mají oproti analogovým systémům řadu výhod. Kamery nemusíme propojovat koaxiálními kabely, ale máme na výběr z celé řady možností, jako například ethernet nebo bezdrátové propojení Wi-Fi. Z toho plynou nižší náklady na instalaci systému. Vedle snazší instalace nabízí digitální technologie i další výhody při samotném provozování kamerového systému. Digitální obraz je kvalitnější než obraz analogový, což oceníme zejména v okamžiku potřeby identifikace pachatele. Největší výhodou digitálního systému je možnost inteligentního zpracování videa. Inteligentní analýzou videa se myslí především analýza pohybujících se objektů (aplikace na počítání lidí atd.) a poplach vyvolaný analýzou obrazu (detekce nesprávného pohybu např. pád člověka nebo auto v protisměru [5.9]). Toho je možné využít v inteligentních domech, zejména

pro seniory. Instalovaný systém sám rozpozná, zda dotyčný upadl a v takovém případě je možné, aby ihned informoval lékaře.



*Obr. 1 - Schéma kamerového systému*

Na obrázku 1 je znázorněno schéma typického kamerového systému. Běžný systém může být ještě doplněn i o mikrofony a reproduktory, a téměř vždy ho doplňuje záznamové médium pro ukládání zaznamenaných dat. Obdélníky s písmenem K jsou Kamery, které posílají záznam do monitorovacího pracoviště, monitorovací pracoviště slouží k Monitorování (sledování záznamu na obrazovce), k Vyhodnocování a k Ovládání. Zpětnou vazbou je zde znázorněno, že monitorovací pracoviště reaguje na dění na scéně a pohybuje kamerami podle potřeby.

Cílem této práce by mělo být nahrazení monitorovacího pracoviště aplikací, která dokáže rozlišit, zda se na scéně pohybuje člověk a bude automaticky dávat povely kameře a ta bude reagovat natočením ve směru pohybu (kamera bude "sledovat" pohybujícího se člověka).

### 2.1 Vlastnosti kamerových systémů

CCTV systémy se vyznačují velkou variabilitou a širokými možnostmi řešení ostrahy prostoru. Mezi základní vlastnosti patří:

- Počet kamer
- Typ propojení kamer
- Různé typy kamerových zařízení  
*(např. pohyblivé, s termovizí, venkovní, noční vize atd.)*
- Počet pracovišť a určení jejich priorit  
*(které kamery a za jakých okolností může obsluha sledovat popř. ovládat)*



Obr. 2 - Monitorovací pracoviště [13.9]

Vzhledem k různorodým požadavkům uživatelů, má každý kamerový systém specifické vlastnosti. Je jasné, že s rostoucím počtem kamer je nutné systém více automatizovat, protože obsluha monitorovacího pracoviště by nestíhala kontrolovat záběry ze všech kamer a přepínat mezi nimi. Například je možné, aby se na monitoru objevoval pouze záběr, ve kterém nastala určitá změna (většinou nás zajímá pohyb člověka) a aby se statické scény nezobrazovaly. Na obrázku 2 vidíme příklad toho, jak může vypadat větší monitorovací pracoviště.

## 2.2 Fixní kamery

Fixní kamery jsou nepohyblivé kamery pro vnitřní nebo venkovní použití, s pevně daným úhlem záběru. Tento úhel záběru je přesně definován při instalaci kamery. Mohou být buď s pevným, nebo vyměnitelným objektivem. Stejně tak mohou být v klasickém nebo dome provedení s krytem. Tento typ kamer je nejlepší volbou pro instalace do prostorů, kde předpokládáme jednoznačnou trajektorii pohybu objektů. Na obrázku 3 vidíme příklad provedení několika fixních kamer od firmy NetRex.



Obr. 3 - Fixní dome kamery [13.10]

## 2.3 PTZ Kamery

PTZ kamery jsou jedny z nejuniverzálnějších kamer. Zkratka PTZ plyne z anglických slov Pan (otáčení doprava a doleva), Tilt (náklon nahoru a dolů) a Zoom (přibližování a oddalování). Jedná se tedy o kamery s ovládáním natočení náklonu a zoomu. Navíc tyto kamery umožňují rotaci 360 stupňů a pohled přímo



Obr. 4 - PTZ dome kamera a ovládací pult [13.9]

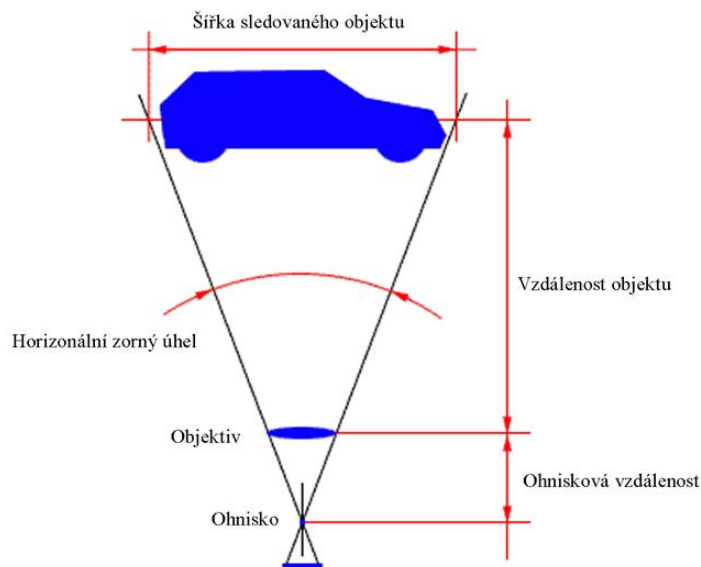
pod sebe. Tyto vlastnosti umožňují uživateli přesné zaměření a sledování důležitých míst a výborně se hodí pro sledování osoby v záběru [5.12]. Pohyb kamery je ovládán na dálku, většinou pomocí speciálního ovládacího pultu (obrázek 4) s klávesnicí a joystickem. Pult také umožňuje ukládat nastavené pozice do paměti pro rychlejší pozorování důležitých míst.

Důležitou součástí PTZ kamery je objektiv. Existují PTZ kamery s výměnnými objektivy, ale v praxi se většinou setkáváme s kamerami, které mají objektiv pevný. Různé vlastnosti objektivů budou podrobněji popsány v další kapitole. Důležité je, abychom při výběru kamery brali ohled na to, v jakém prostředí bude kamera používána, a podle toho jsme vybrali správný model.

Další důležitou vlastností nejen PTZ kamer je propojitelnost k PC. Opět zde platí, že různé kamery nám nabízejí různé možnosti. Pro připojení jedné kamery je možné použít sériový port. Nejpoužívanější způsob připojení je ethernetovým kabelem. Tuto možnost využijeme, když máme pouze jednu nebo jen několik málo kamer (při více než jedné kameře je nutno použít síťové prvky switch/router). Pro připojení více kamer lze tento způsob použít pouze s většími síťovými prvky, a proto není optimální. Lepší řešení je využít kamery, které nabízejí možnost bezdrátového připojení. V této práci jsme pracovali s jednou nebo maximálně dvěma kamerami a využívali jsme připojení k PC přímo pomocí ethernetového kabelu nebo pomocí routeru. [13.9]

## 3. Vlastnosti objektivů

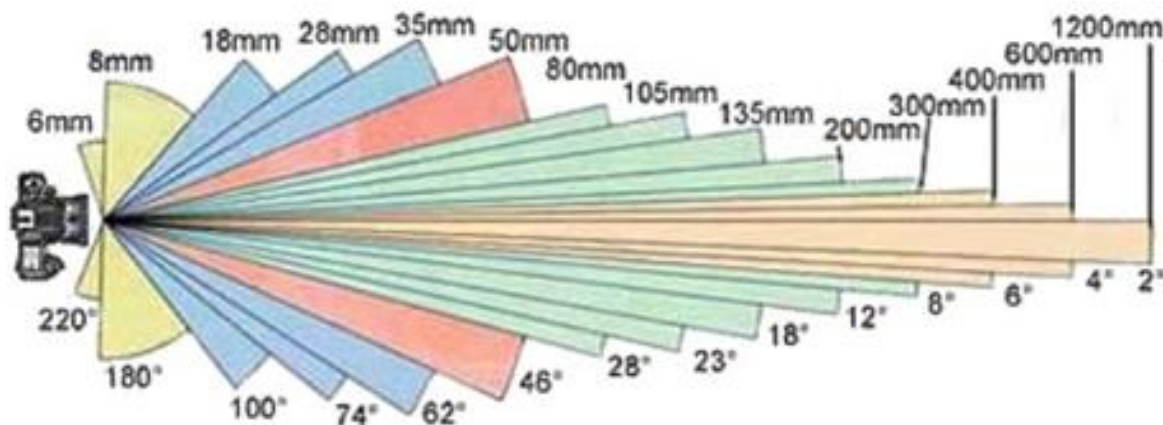
V této kapitole budeme popisovat základní vlastnosti objektivů. U objektivů nás nejvíce zajímá světelnost, ohnisková vzdálenost, diagonální úhel záběru (úhel úhlopříčky záběru) a rozměr snímacího čipu. V praxi nás, spíše než diagonální úhel záběru, bude zajímat horizontální a vertikální úhel záběru. Pro různé formáty čipu jsou úhly záběru pro stejný objektiv odlišné. Přitom platí, že čím menší je rozměr čipu, tím menší je úhel záběru stejného objektivu při stejné ohniskové vzdálenosti.



Obr. 5 - Vztah objektu a objektivu

### 3.1 Ohnisková vzdálenost a zorný úhel

Ohnisková vzdálenost je jedna z nejdůležitějších vlastností každého objektivu. Fyzikální definice ohniskové vzdálenosti: „Ohnisková vzdálenost je vzdálenost mezi optickým středem objektivu a rovinou snímače (čipu) při zaostření na nekonečno.“ Mnohem důležitější je fakt, že každé ohnisko má různý zorný úhel záběru a od toho se odvíjí jednak použití vhodného ohniska podle toho, co chceme mít v záběru a také dělení objektivů do několika kategorií. [13.14]



Obr. 6 - Zorný úhel v závislosti na ohniskové vzdálenosti [13.14]

Na obrázku 6 vidíme, jak se zorný úhel mění v závislosti na ohnisku. Žlutou barvou jsou znázorněny ohniskové vzdálenosti pro širokoúhlé objektivy nebo objektivy s tzv. rybím okem (zorný úhel cca 180°), naopak nejužší zorný úhel mají tzv. teleobjektivy (cca 10°).

$$\operatorname{tg} \frac{\alpha}{2} = \frac{u}{2f} \quad (3.1)$$

Ze vztahu 3.1 plyne, že zorný úhel ( $\alpha$ ) nezávisí pouze na ohniskové vzdálenosti ( $f$ ), ale i na velikosti snímače ( $u$ ), která je pevně daná výrobou čipu. Senzor většinou nebývá čtvercový, proto máme v horizontálním a vertikálním směru jiný zorný úhel. Na obrázku 6 jsou zobrazeny zorné úhly pro full frame senzor (velikost senzoru odpovídá velikosti filmového políčka).

### 3.2 Měnitelný rozsah ohniskové vzdálenosti – ZOOM

Optický zoom není nic jiného než změna ohniskové vzdálenosti objektivu. Zoom udává, kolikrát můžeme změnit ohniskovou vzdálenost. V praxi výrobce udává maximální zoom. Máme-li tedy objektiv s ohniskovou vzdáleností  $f = 8\text{mm}$  a zoom  $12x$ , jsme schopni dosáhnout  $f' = 96\text{mm}$ . S použitím zoomu se nám ovšem změní zorný úhel. My jsme v této práci používali kameru bez zoomu.

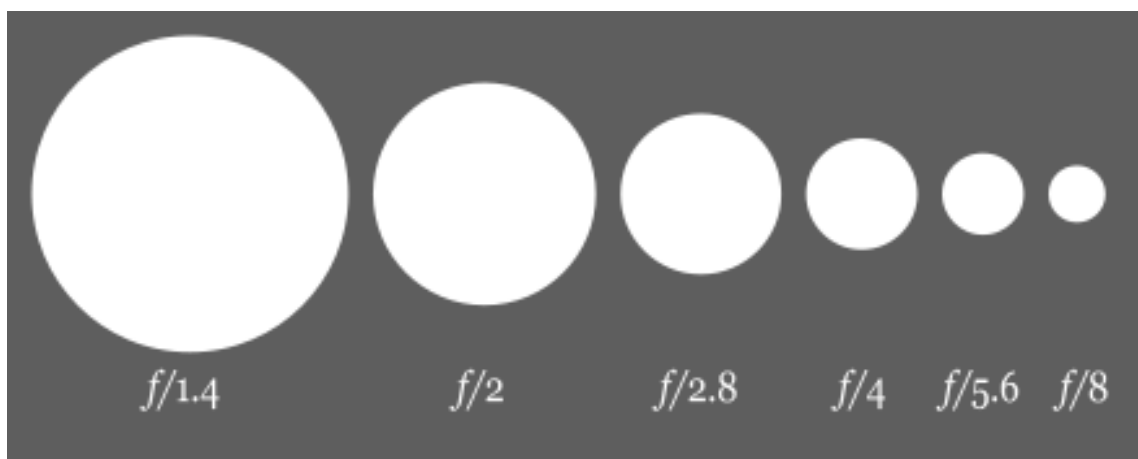


### 3.3 Clonové číslo a světelnost objektivu

Dalším důležitým parametrem objektivu je jeho světelnost. Světelností se označuje největší možný průměr clony (minimální clonové číslo  $N$ ), kterého je objektiv schopen. Zavřít clonu (zmenšit otvor) dokáže každý objektiv, ale základní funkce objektivu je doprava světla na senzor a tak možnost otevřít clonu (tj. světelnost) je zásadní. Dobrá světelnost umožňuje snímat i za špatných světelných podmínek. Definované světelnosti je každý objektiv schopen jen při zaostření na nekonečno. Při zaostření na bližší vzdálenost je proto třeba počítat s poklesem světelnosti. [13.5]

$$N = \frac{f}{D} \quad (3.2)$$

Clonové číslo udává velikost změny světelnosti. V praxi se používá normalizovaná řada clonových čísel  $N = (1; 1,4; 2; 2,8; 4; 5,6; 8; 11; 16; 22)$ , kdy změna o jedno clonové číslo znamená pokles světelnosti objektivu o  $\frac{1}{2}$  - každé číslo  $N$  v řadě je  $\sqrt{2}$ -krát větší, než číslo před ním. [13.5]



Obr. 7 - Clona a clonové číslo [13.5]

### 3.4 Hyperfokální vzdálenost

Hyperfokální vzdálenost  $h$  se definuje jako vzdálenost, od které, když je objektiv zaostřen na nekonečno, je vše až do nekonečna ostré, čili přední hranice pásma ostrosti při zaostření na nekonečno. Tato vzdálenost závisí na ohniskové vzdálenosti objektivu, nastaveném clonovém čísle a v neposlední řadě také na tom, co ještě považujeme za ostré a co už ne. Tuto vlastnost definujeme prostřednictvím průměru největšího akceptovatelného rozptylového kroužku  $c$ . Pokud bod neleží přesně v rovině, na kterou je zaostřeno, zobrazí se na filmu nikoli jako bod, ale jako kroužek. Tomuto kroužku se říká rozptylový kroužek. Pokud je tento kroužek dostatečně malý, jeví se nám stále ještě jako přijatelně ostrý bod. V praxi se používá velikost rozptylového kroužku 0.03mm. [13.5]

$$h = \frac{f^2}{Nc} \quad (3.3)$$

Pro lepší orientaci výrobce často udává tzv. hyperfokální tabulku daného objektivu. V této tabulce si vyhledáme hyperfokální vzdálenost pro danou ohniskovou vzdálenost a clonové číslo. V tabulce nalezneme hyperfokální vzdálenosti přímo v metrech.

N[-] f[mm]	1	1.4	2	2.8	4	5.6	8	11	16	22
14	6,53	4,67	3,27	2,33	1,63	1,17	0,82	0,59	0,41	0,30
17	9,63	6,88	4,82	3,44	2,41	1,72	1,20	0,88	0,60	0,44
20	13,33	9,52	6,67	4,76	3,33	2,38	1,67	1,21	0,83	0,61
24	19,20	13,71	9,60	6,86	4,80	3,43	2,40	1,75	1,20	0,87
28	26,13	18,67	13,07	9,33	6,53	4,67	3,27	2,38	1,63	1,19
35	40,83	29,17	20,42	14,58	10,21	7,29	5,10	3,71	2,55	1,86
50	83,33	59,52	41,67	29,76	20,83	14,88	10,42	7,58	5,21	3,79
70	163,33	116,67	81,67	58,33	40,83	29,17	20,42	14,85	10,21	7,42
105	367,50	262,50	183,75	131,25	91,88	65,63	45,94	33,41	22,97	16,70
135	607,50	433,93	303,75	216,96	151,88	108,48	75,94	55,23	37,97	27,61
200	1333,33	952,38	666,67	476,19	333,33	238,10	166,67	121,21	83,33	60,61
300	3000,00	2142,86	1500,00	1071,43	750,00	535,71	375,00	272,73	187,50	136,36

Obr. 8 - Tabulka hyperfokálních vzdáleností

## 4. Metody detekce a klasifikace objektu

Při zpracování záznamu z kamery je důležité detekovat v obraze objekty, které nás zajímají. V bezpečnostních systémech nás většinou zajímá detekce člověka, ale samozřejmě může jít i o jiné objekty (detekce aut v dopravních aplikacích na počítání frekvence dopravy atd.). Ve většině používaných metod se detekcí objektu myslí oddělení popředí (objekt zájmu) od pozadí. Při detekci objektu máme několik základních problémů:

- Různé variace stejného objektu

*U lidí například pohled zepředu, ze stran, různobarevné oblečení atd.*

- Různé jasové podmínky

*Stejná scéna, ale v jinou denní dobu.*

- Odlišné druhy pozadí

*Málo členité x velmi členité, uvnitř budovy x venku.*

- Různé měřítko pro stejný objekt

*Objekt blízko kamery se bude přirozeně jevit větší než stejný objekt dále.*

- Problémy s kompozicí záběru

*V záběru se nám může objevit neúplný objekt.*

- Různě vztažené pohyby

*Pohybující se kamera a nehybný člověk, statická kamera a pohybující se člověk, pohybující se kamera i člověk.*

Každá z těchto podmínek nám bude určitým způsobem ztěžovat detekci objektu, a proto musíme zvolit metodu, která bude za daných podmínek dávat nejlepší výsledky. V této kapitole rozebereme základní metody na detekci objektu v obraze.

## 4.1 Odečtení snímků

Základní metodou je odečtení dvou po sobě jdoucích snímků. Odečítáme přecházející snímek od aktuálního snímku. Výsledkem získáme rozdílový snímek, ve kterém jsou zvýrazněné změny, které v záběru nastaly. Matematický zápis postupu vidíme v rovnici 4.1. Provádíme postupné odečítání všech pixelů na souřadnicích  $x, y$  snímku  $i-1$  od snímku  $i$ .  $X$  a  $Y$  udávají velikost snímku. Výsledek odečtení vidíme na obrázku 9<sup>1</sup>. Je vidět, že pro obrázky s větší změnou dává metoda celkem pěkné výsledky, i když je to dané především dobře zvolenou scénou. Výsledný rozdílový snímek je nutno před použitím ještě dále upravovat (invertovat, prahovat a odfiltrvat šum).

$$\text{diff } x, y, i = \text{frame } x, y, i - \text{frame } x, y, i - 1 \quad (4.1)$$

Podmínky platnosti:  $0 < x \leq X, 0 < y \leq Y$



Obr. 9 - Odečítání po sobě jdoucích snímků  
Zleva: frame  $i$ , frame  $i+1$ , rozdílový snímek

Tato metoda je vhodná, pokud máme v záběru scénu s malými změnami na pozadí. Větší změny na pozadí mají na výsledek negativní vliv. Je vidět, že stín, který je na obou snímcích velmi podobný, není ve výsledném obrázku zachován. Naopak metoda není vhodná pro detekci pomalu se pohybujících objektů, protože

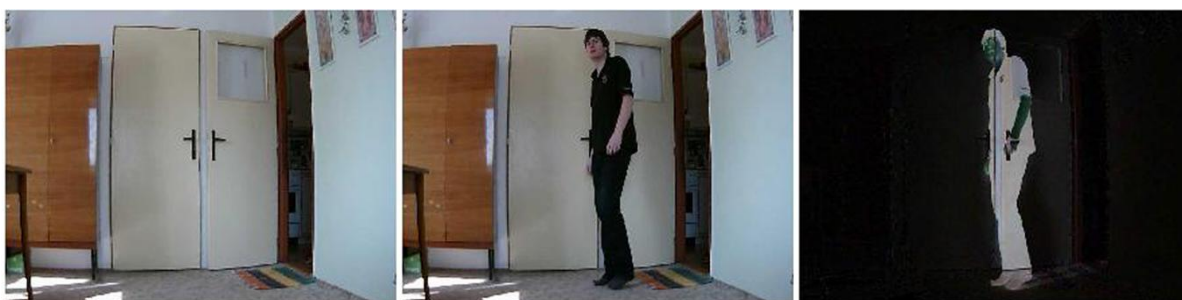
<sup>1</sup> Při tvorbě obrázku byly použity dva po sobě jdoucí snímky, ale s vyšším časovým posunem, takže při běžných FPS, by bylo nutné použít snímky s vyšším rozestupem od sebe.

rozdíl dvou po sobě jdoucích snímků je velmi malý. V tomto případě bychom museli odečítat snímky s větším časovým rozestupem (například snímek  $i$  a snímek  $i-5$ ), místo snímků následujících hned po sobě. Rozestup mezi snímky se nedá přesně určit, ale je nutné vyzkoušet několik hodnot.

## 4.2 Odečtení pozadí

Další metodou je metoda s pevně definovaným statickým pozadím. Tato metoda spočívá v tom, že se od aktuálního snímku odečítá snímek definovaný jako pozadí. Výsledkem je opět rozdílový snímek (obrázek 10), ve kterém jsou vidět změny, které na scéně nastaly. I zde vidíme, že před dalším postupem je nutná další úprava obrázku. Matematický zápis vidíme v rovnici 4.2. Malá písmena  $x$  a  $y$  jsou opět souřadnice pixelu, velké  $X$  a  $Y$  jsou velikost obrázku.  $\text{Frame}(x,y,i)$  je aktuální snímek,  $\text{bg}(x,y)$  je definované pozadí<sup>11</sup>. Výsledek metody je na obrázku 10.

$$\text{diff } x,y,i = \text{frame } x,y,i - \text{bg } x,y \quad 0 < x \leq X, 0 < y \leq Y \quad (4.2)$$



Obr. 10 - Odečtení pozadí

Zleva: pozadí, frame  $i$ , rozdílový snímek

Tato metoda je vhodná, když máme pozadí opravdu statické a přepokládáme vysoký kontrast mezi pohybujícím se objektem a pozadím. Metoda je naopak nevhodná, máme-li pozadí, ve kterém se může něco pohybovat. Jakákoliv, i malá změna na pozadí se totiž velmi negativně podepíše na výsledném rozdílovém

<sup>11</sup> Jako pozadí se ve většině případů definuje první nebo nultý snímek. Potom:  $\text{bg}(x,y) = \text{frame}(x,y,0)$

snímku – stíny, odlesky, pohybující se záclony atd. Všechny tyto malé změny nám velmi komplikují použití tohoto postupu.

### 4.3 Odečítání předzpracovaného pozadí

Tato metoda je velmi podobná metodě přechozí, ale od aktuálního snímku se neodečítá pevně dané pozadí, ale toto pozadí se upravuje v závislosti na aktuálním snímku. Úprava spočívá v transformaci jasu pozadí. Když je hodnota jasu daného pixelu aktuálního snímku větší než hodnota jasu pozadí na stejné pozici, zvýší se jas pozadí o 1. Když je menší, sníží se o 1. Když je stejný, pozadí neupravujeme. Pro větší názornost jsme zde přidali část kódu programu, který pozadí upravuje.

```

if frame(x,y,i) > bg(x,y)
    bg'(x,y) = bg(x,y) + 1
else
    bg'(x,y) = bg(x,y) - 1
end

```

$$diff\ x,y,i = frame\ x,y,i - bg'\ x,y \quad 0 < x \leq X, 0 < y \leq Y \quad (4.3)$$

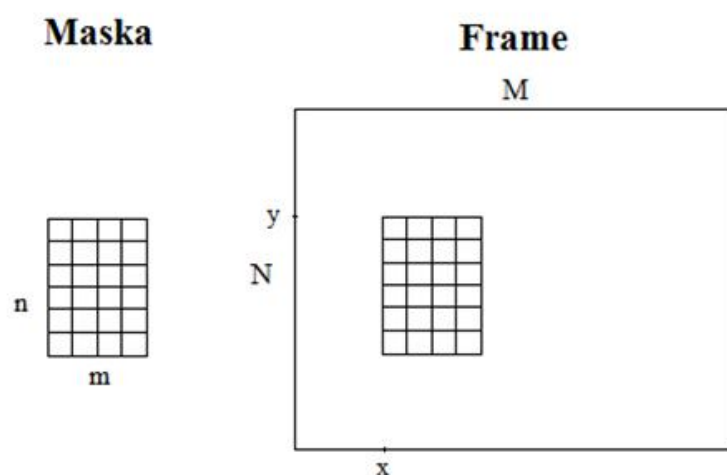
Další postup je již shodný jako v případě předchozí metody. Tato malá korekce jasu pozadí nám ovšem zlepšuje výsledky – odstraňuje šum v okolí objektu, který je způsoben odlesky a stíny. Výsledný rozdílový snímek vidíme na obrázku 11.



Obr. 11 - Odečítání předzpracovaného pozadí  
Zleva: pozadí, snímek  $i$ , rozdílový snímek

## 4.4 Detekce pomocí masky

Jako poslední se zaměříme na detekci objektu pomocí masky. Tato metoda se od ostatních liší tím, že od aktuálního snímku neodečítáme pozadí. Můžeme tedy tento postup použít i v případě změn na pozadí. Metoda spočívá v tom, že si předem definujeme masku, a postupným posouváním hledáme v obraze místo, které je této masce nejvíce podobné. Jako kritérium podobnosti se používá střední absolutní odchylka MAD (mean absolute deviation). Matematický zápis vidíme v rovnici 4.4. Popis proměnných je zřetelný z obrázku 12. Pozice s minimální MAD se nazývá BMP (best matching position) [7]. Výsledek vidíme na obrázku 13. Je potřeba si uvědomit, že při tvorbě ilustračního obrázku byla maska definována z velmi podobného snímku (asi o 10 snímků vpřed, stejná vzdálenost objektu od kamery, stejné světelné podmínky atd.), a proto je výsledek velmi dobrý. V praxi by maska nemohla být takto podobná a výsledkem by nemuselo být takto přesné určení pozice objektu.



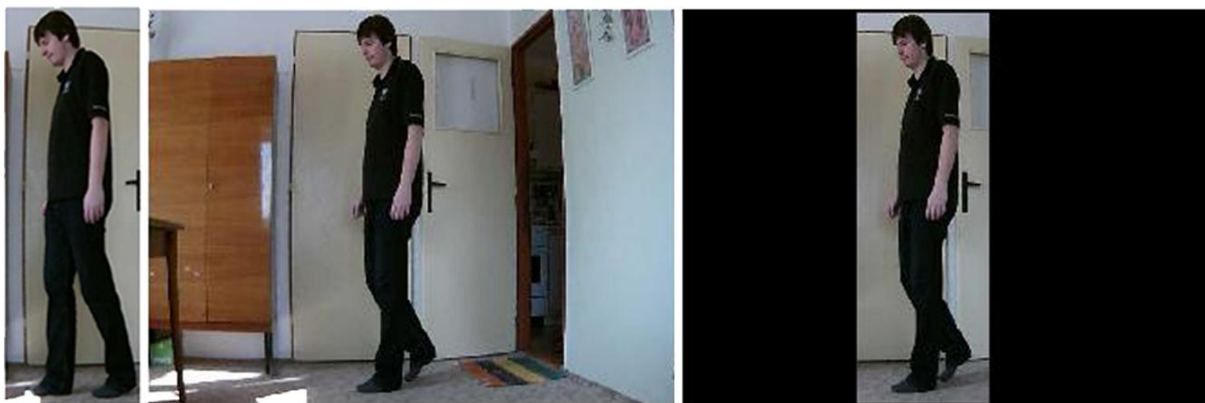
Obr. 12 - Princip detekce pomocí masky [7]

$$MAD_{x,y} = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |frame(x+i, y+j) - mask(i,j)| \quad (4.4)$$

$$BMP_{x,y} = \min MAD_{x,y}$$

Pro obě rovnice platí stejné podmínky:

$$1 \leq x \leq M - m, 1 \leq y \leq N - n$$



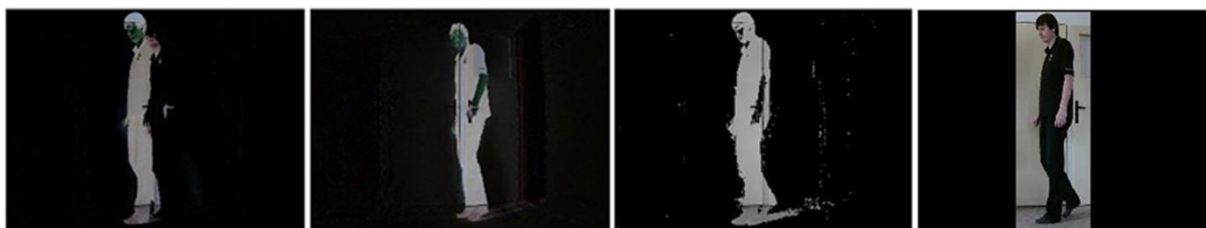
Obr. 13 - Detekce pomocí masky

Zleva: maska, frame  $i$ , vyhledané místo s největší podobností

Výhodou této metody je to, že nemusíme mít úplně statické pozadí, ale je potřeba dobrý kontrast mezi objektem a pozadím, aby bylo možné výsledek vhodně prahovat. Nevýhodou je složitější výpočet, náročnější na výpočetní výkon. Hlavní nevýhodou ovšem je, že objekty různě vzdálené od kamery, mají různou velikost, a proto je nutné použít různé velikosti masek. Různé masky musíme také použít, když budeme chtít detekovat různé typy objektů. Výhodou je, že získaný objekt máme podle masky rovnou i klasifikovaný. Testování snímku na podobnost se všemi maskami je však výpočetně náročné a trvá delší dobu.

## 4.5 Porovnání metod

V této části kapitoly porovnáme výše pospané metody. Výsledky jednotlivých metod vidíme na obrázku 14. Je vidět, že zdánlivě nejlepšího výsledku dosáhneme odečítáním po sobě jdoucích snímků. Tato metoda je také velmi rychlá a nevádí jí



Obr. 14 - Porovnání metod na detekci objektu

Zleva: rozdíl framu  $i$  a  $i+1$ , odečtení pozadí, odečtení upraveného pozadí, detekce maskou



malé změny na pozadí. Problém ovšem nastává při pohybu kamery. V rozdílovém snímku budou výrazné úplně všechny změny a výsledek bude nepoužitelný.

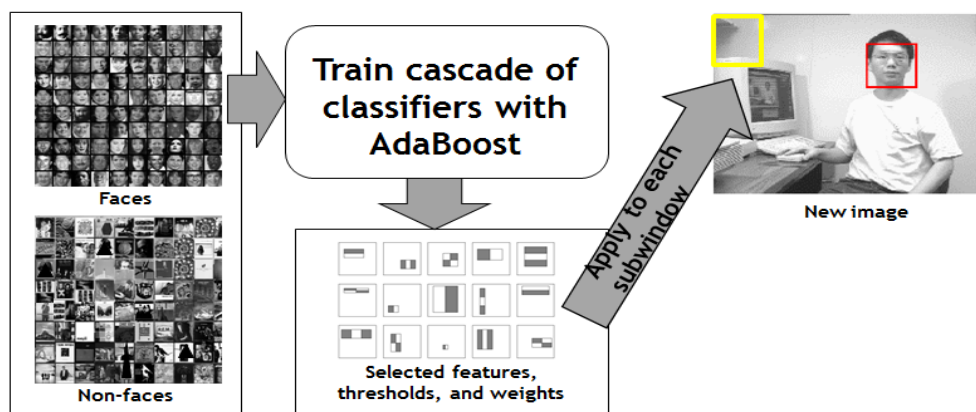
Další obrázek je výsledek po odečtení pozadí. Tato metoda je také velmi rychlá, ale zde se negativně projeví malé změny na pozadí. Vidím, že objekt na obrázku není čistě ohraničen, ale v okolí objektu zůstala velmi velká světlá místa. Takto velké světlé plochy se špatně odfiltrovávají a špatně se s objektem pracuje.

Na dalším obrázku vidíme metodu s odečtením předem upraveného pozadí. Je zde vidět velmi výrazně ohraničený hlavní objekt a v okolí více malých světlých míst. Tyto malé plochy se dají velmi účinně odstranit vhodným prahováním a uzavřením objektu. I když je tato metoda výpočetně pomalejší než předchozí dvě zmíněné metody (nelze s obrázkem pracovat jako s maticí, ale musí se porovnat každý pixel jednotlivě), při tvorbě aplikace jsme použili právě tuto metodu.

Poslední porovnávanou metodou je detekce pomocí masky. Tento postup velmi dobře určí pozici objektu, ale jak už bylo uvedeno výše, přesnost velmi závisí na volbě vhodné masky. Dalším problémem může být to, že při nedostatečném kontrastu objektu a pozadí, se výsledek nepodaří vhodně prahovat. Pro naše použití tato metoda není vhodná z důvodu různé vzdálenosti objektu od kamery. Bylo by nutné použít při prohledávání snímku více masek a celý výpočet by byl pomalejší než metoda předchozí.

### 4.6 Viola – Jones

Kromě výše zmíněných základních metod na detekci objektu v obraze existují samozřejmě ještě metody pokročilejší. Z těchto metod jsme se rozhodli představit metodu Viola-Jones, založenou na Haarových klasifikátorech. Tato metoda z roku 2001 je jedna z prvních, která fungoval na detekci objektu v reálném čase. [13.7] Hlavní využití této metody je detekce obličeje, ale po natrénování se samozřejmě dá použít i na detekci jiných objektů. Obecný princip metody vidíme na obrázku 15.



Obr. 15 - Princip Viola - Jones detekce [5]

Základem jsou trénovací databáze. V těchto databázích máme uloženy obrázky, které jsou/nejsou obličejem (nebo jiné objekty). Obrázek se rozloží na okna 24x24. Každému oknu se na základě rozložení jasu uvnitř přiřadí Haarovský klasifikátor. Kombinace těchto klasifikátorů se trénuje pomocí AdaBoost algoritmu. Poté se prohledává obrázek, ve kterém chceme objekt detekovat a hledá se v něm místo s podobnou kombinací Haarovských klasifikátorů. Nalezená místa se pomocí několika stupňů klasifikace kaskádově testují na podobnost s natrénovanou databází. Když místo splní podmínky všech stupňů, je vyhodnoceno jako obličej. Metodu jsme vyzkoušeli na videu pomocí volně dostupného zdrojového kódu [13.7]. Výsledky vidíme na obrázku 16.

Mezi výhody této metody patří schopnost natrénovat se na různé druhy objektů. Nevýhodou je nutnost velké trénovací databáze, z tohoto důvodu jsme metodu pouze vyzkoušeli, ale nepoužívali jsme ji v praktické části práce.



Obr. 16 - Výsledky detekce obličejem Viola - Jones metodou

## 4.7 Klasifikace objektu

Základním principem klasifikace objektů je přiřadit objekt do určité třídy a určit tím, o jaký objekt se jedná (člověk, automobil, zvíře atd.). Metody, jak zařadit objekt do dané třídy, jsou založeny na porovnávání určitých parametrů obrazu (tvar, barva atd.) s databází vzorů. Z předchozí části už víme, jak detekovat objekt v obraze. Před vlastní klasifikací objektu je nutné obrázek po detekci upravit.

První operací aplikovanou na získaný rozdílový snímek většinou bývá prahování.<sup>III</sup> Původní šedotónový obrázek s 255 jasovými úrovněmi se upraví na obrázek se dvěma hodnotami jasu (0 a 255). Všechny pixely s hodnotou jasu větší než práh se nastaví na hodnotu 255, zbytek se vynuluje.

Po prahování zpravidla následují další morfologické operace. Základní morfologické operace jsou dilatace a eroze, otevření a uzavření objektu (více se můžeme dočíst např. v [3]). My jsme v naší práci používali pouze uzavření objektu. Výsledný snímek po prahování a uzavření vidíme na obrázku 17. Je vidět, že po uzavření se odstraní malé plochy bílých, šumových pixelů. S takto upraveným obrázkem můžeme dále pracovat.



Obr. 17 - Úprava obrázku  
Zleva: původní obrázek, prahovaný obrázek, uzavřený obrázek

<sup>III</sup> Někdy je vhodné obrázek ještě před prahováním invertovat. Na výsledek to ovšem nemá vliv.

### 4.7.1 Pomocí masky

Princip této metody je velmi podobný jako u detekce pomocí masky. Liší se v tom, že zde již pracujeme s detekovaným objektem (nezáleží na tom, jakou metodu detekce jsme použili) a upraveným pro další zpracování.

Parametrem pro zařazení do třídy je tvar objektu. Do databáze natrénujeme různé masky pro různé objekty a porovnáváme, které masce je objekt v obraze nejvíce podobný. Opět zde platí stejná nevýhoda jako při detekci pomocí masky a to, že pro různé vzdálenosti objektů od kamery (různá velikosti objektu v obraze) musíme použít různé masky nebo nějakým způsobem transformovat obraz na stejnou velikost jako má maska. Také musíme použít různé masky pro různé pohledy na objekt (auto zepředu má jinou masku než automobil ze strany).



Obr. 18 - Masky pro různé objekty [11]

Na obrázku 18 vidíme příklady, jak mohou vypadat masky pro různé objekty. Natrénováním více masek do databáze jsme schopni dosáhnout vyšší přesnosti klasifikace. [11]

## 5. Návrh kamerového systému

Praktická část této práce se zabývá návrhem kamerového systému. My jsme kamerový systém realizovali pomocí kamery Vivotek PT8133, proto se v této kapitole budeme zabývat jejími parametry. Dále zde budou popsány způsoby připojení jedné nebo více kamer k PC a jejich instalace. V poslední části budou rozebrány možnosti ovládání kamery.



Obr. 19 - Kamera Vivotek PT8133 [14]

### 5.1 Technické parametry kamery

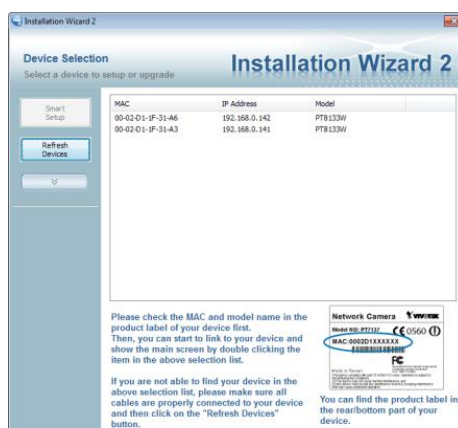
Typ kamery	PT
Spotřeba	cca 5.1W
Rozsah otáčení	-175°- 175°
Rozsah naklápění	-35 – 90°
Síťové rozhraní	10/100 Mbit/s Ethernet, RJ45
Snímací čipu	¼" CMOS
Rozlišení	Max. 1280 x 800 px
Snímkovací frekvence	Max. 30 snímků/s při max. rozlišení
Komprese videa	MJPEG, MPEG-4, H.264,
Integrovaný objektiv	Ohnisková vzdálenost 3.6 mm,
Horizontální úhel záběru	58.39°
Vertikální úhel záběru	35.58°

V tabulce vidíme základní technické parametry kamery. Pro naše používání je nejdůležitější rozsah otáčení a naklápění. Dalším důležitým parametrem je rozlišení videa. My jsme z důvodu větší rychlosti aplikace nepoužívali plně rozlišení (1280x800px), ale 800x600px. Dalšími důležitými parametry jsou velikost snímacího čipu a ohnisková vzdálenost objektivu. Těmito parametry jsou dány úhly záběru ve vertikálním a horizontálním směru. [14]

## 5.2 Připojení a instalace a kamery

V této části se budeme zabývat připojením kamery k PC a následnou instalací. Máme několik možností, jak kameru k PC připojit. Prvním, nejjednodušším způsobem je připojit kameru „přímo“. Pomocí síťového kabelu připojíme kameru k PC a nastavíme na PC pevnou IP adresu. Po spuštění Installation Wizard 2 (je nutné nainstalovat z CD) se prohledá síť a kamery se přiřadí IP adresa. Tento způsob připojení je vhodný pro práci s jednou kamerou.

Chceme-li pracovat s více kamerami, musíme je připojit pomocí síťových prvků (např. router) do stejné sítě jako PC. Po spuštění Installation Wizard 2 se prohledá síť a každé kameře opět přiřadí IP adresa. K orientaci mezi připojenými kamerami slouží MAC adresa, která je jedinečná pro každou kameru. Poslední možností je použití bezdrátové komunikace. Tuto možnost zde nebudeme podrobněji rozebírat, protože jsme ji v naší práci nepoužívali. Na obrázku 20 vidíme okno Installation Wizardu po přiřazení IP adres dvěma kamerám.



Obr. 20 - Okno Installation wizard

## 5.3 Ovládání kamery

Zde budou popsány základní možnosti ovládání kamery. První možností je základní ovládání, které nabízí výrobce. Další způsobem, jak ovládat kameru, je ruční posílání příkazů do adresového řádku internetového prohlížeče. Poslední popsanou možností je ovládání kamery pomocí námi vytvořené Matlab aplikace.

### 5.3.1 Základní ovládání

Základní ovládací panel se nám zobrazí dvojklikem na IP adresu kamery v Installation Wizardu. Tento panel vidíme na obrázku 21. Levá část panelu slouží k ovládání, pravá část k zobrazení. Ovládání je intuitivní, pomocí šipek ovládáme kameru. Můžeme zde nastavit rychlost náklonu a rychlost otáčení. Protože jsme kameru neměli připevněnou ke stropu, ale položenou na stole, je záznam z kamery otočený. Z toho důvodu je otočeno také ovládání – šipka vlevo otočí kameru vpravo.



Obr. 21 - Základní ovládání

### 5.3.2 Ovládání pomocí webového prohlížeče

Kromě základního ovládání pomocí panelu nabízí kamera možnost ovládání přes webový prohlížeč. Stačí jednoduše do adresového řádku webového prohlížeče zadat URL s příkazem. Kromě příkazů k pohybu kamery se přes prohlížeč dá i upravovat nastavení kamery. Toto ovládání se dá využít, pokud uživatel má vlastní webové stránky. Uživatel si může jednoduše, např. v HTML vytvořit vlastní ovládací panel pro pohodlný přístup odkudkoliv s připojením k internetu. V další části kapitoly bude popsána syntaxe příkazů a bude zde ukázáno pár základních příkazů.

#### Základní syntaxe příkazu

<http://<servername>/cgi-bin/<subdir>/<subdir>.../<cgi>.<ext>>

[\[?<parameter>=<value>\[&<parameter>=<value>...\]\]](http://<servername>/cgi-bin/<subdir>/<subdir>.../<cgi>.<ext>?<parameter>=<value>[&<parameter>=<value>...])

- a) `servername`: zde se zadává IP adresa kamery, kterou chceme ovládat
- b) `subdir`: skupina příkazů (ovládací, nastavení atd.)
- c) `parameter`: název parametru (move, pan atd.)
- d) `value`: vlastní příkaz (left, right atd.)<sup>IV</sup>

#### Ukázka základních příkazů

Existuje celá řada příkazů, rozdělených do několika skupin. Zde budou popsány pouze dvě základní skupiny příkazů – skupina ovládacích příkazů a skupina zobrazování. Příkazy z ostatních skupin se používají méně často, my jsme v naší práci tyto příkazy téměř vůbec nepotřebovali. Slouží především k pokročilému nastavení kamery (dají se použít např. ke správě uživatelských účtů) a k ostatnímu administrátorskému spravování. [14]

---

<sup>IV</sup> Po části `parameter` vždy následuje `value`. Chceme-li v jednom příkazu posílat více parametrů, oddělujeme je znakem `&`



### a) Skupina ovládání kamery

[http://<servername>/cgi-bin/viewer/camctrl.cgi?\[channel=<value>\]\[&camid=<value>\]](http://<servername>/cgi-bin/viewer/camctrl.cgi?[channel=<value>][&camid=<value>])

[\[&move=<value>\]](#) – Move = (home | up | down | left | right)

[\[&auto=<value>\]](#) – Auto = (pan | patrol)

[\[&speedpan=<value>\]](#) – speedpan = (-5 ÷ 5)

Výsledný příkaz:

<http://192.168.0.1/cgi-bin/viewer/camctrl.cgi?channel=0&camid=1&move=left&speedpan=-5>

### b) Skupina zobrazování

[http://<servername>/cgi-bin/viewer/video.jpg?\[channel=<value>\]](http://<servername>/cgi-bin/viewer/video.jpg?[channel=<value>])

[\[&resolution=<value>\]](#) – resolution = (1280x800 | 800x600)

[\[&quality=<value>\]](#) – quality = (1÷5)

[\[&streamid=<value>\]](#) – streamid = (1÷n)

Výsledný příkaz:

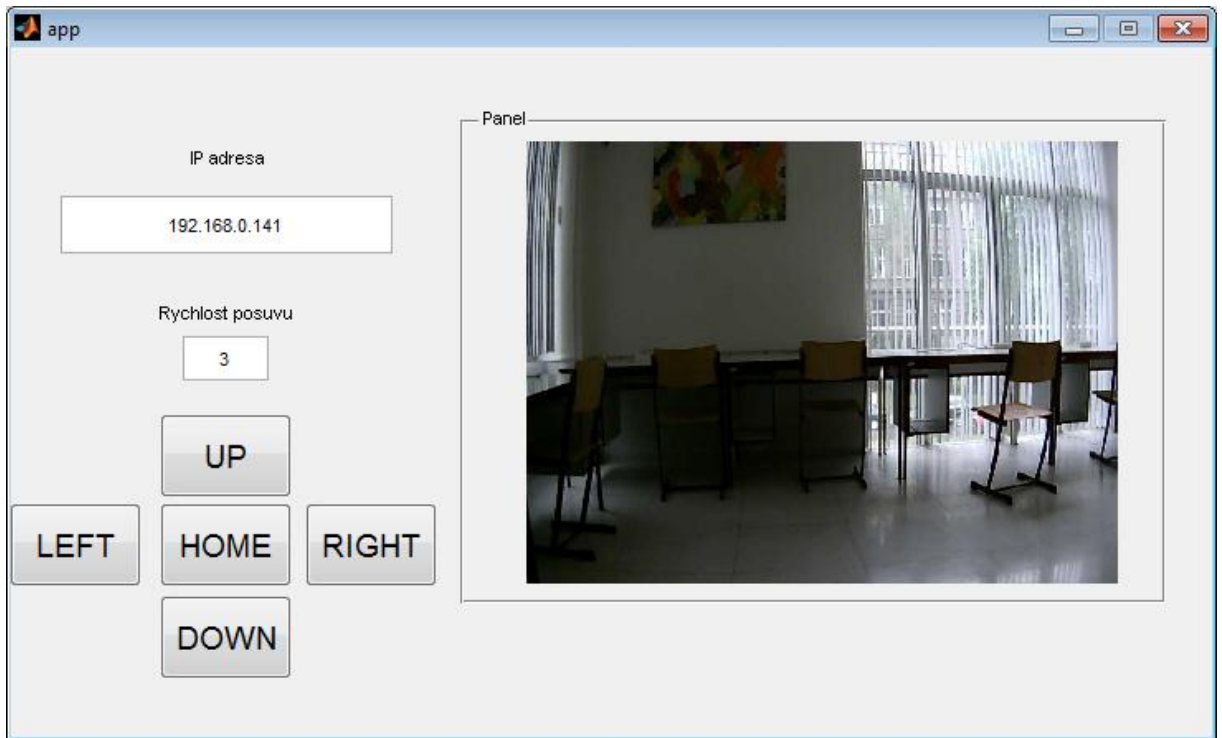
<http://192.168.0.1/cgi-bin/viewer/video.jpg?resolution=800x600&quality=5>

### 5.3.3 Ovládání pomocí Matlab aplikace

Poslední vybranou možností ovládání, kterou zde budeme popisovat, je ovládání pomocí námi vytvořené Matlab aplikace. Tuto aplikaci jsme si vytvořili na vyzkoušení URL příkazů z předchozí části a také kvůli vyzkoušení práce s GUI v Matlabu.

Kamera se ovládá pomocí GUI (na obrázku 22). Toto GUI je podobné výchozímu ovládání, je však značně zjednodušené a přizpůsobené našim požadavkům (již jsme si otočili obraz a ovládací příkazy). Levá část GUI je část ovládací a pravá část slouží k zobrazení scény. Po zadání IP adresy kamery

do požadovaného okna se nám zobrazí pohled kamery. Poté můžeme pomocí tlačítek v levé části kameru jednoduše ovládat. Tlačítka nedělají nic jiného, než že posílají daný URL příkaz do interního Matlab webového prohlížeče.

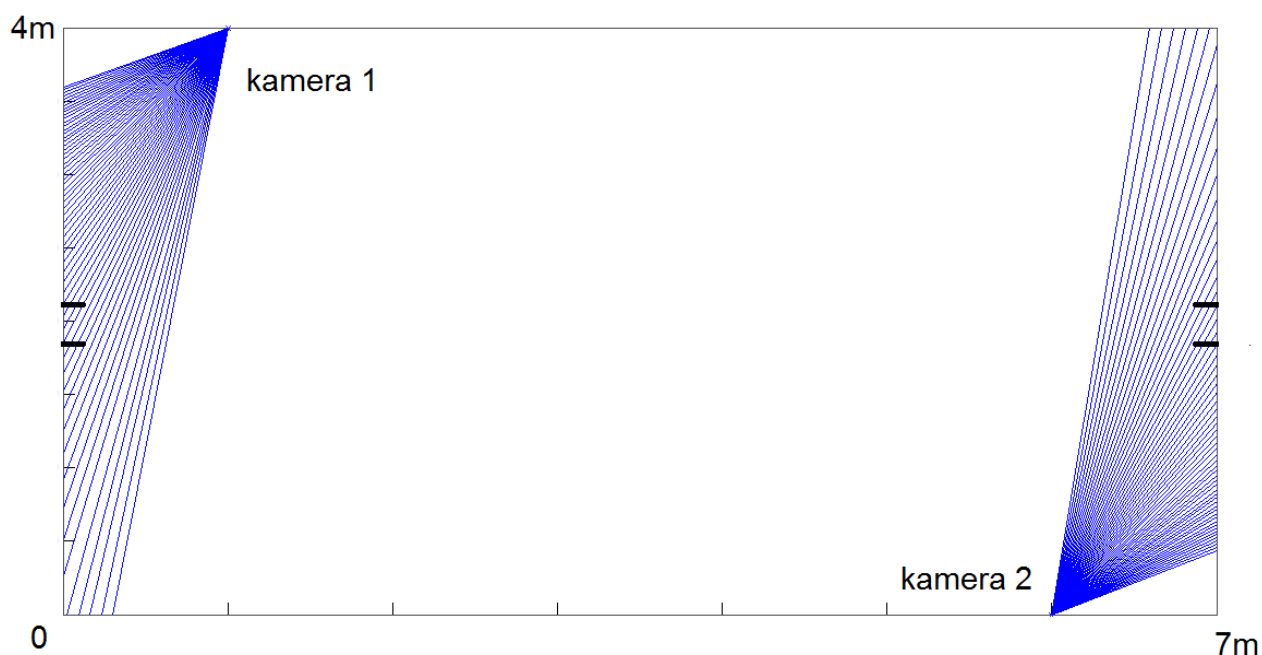


Obr. 22 - Ovládací GUI

Aplikace má výhodu v tom, že chceme-li ovládat další kameru, nemusíme zavírat okno, ale stačí zadat IP adresu jiné kamery.

## 6. Aplikace na monitorování prostoru

Tato kapitola bude zaměřena na praktickou část práce. Při tvorbě aplikace bylo nutné nejprve vybrat vhodný prostor (ne zbytečně jednoduchý ani zbytečně složitý). Dalším krokem po zvolení bylo prostoru určit počet a rozmístění kamer. Jako pomůcku vizualizaci pokrytí prostoru jsme si v Matlabu vytvořili aplikaci. Vstupem do aplikace je definice prostoru, pozice kamer, zorné úhly a směry natočení kamer. Tato aplikace po zadání vstupních parametrů vizualizuje pokrytí prostoru kamerami. Na obrázku 23 vidíme výstup z aplikace – vybraný prostor, základní rozmístění kamer a prostor, který kamery pokrývají. Velikost prostoru jsme zvolili 7x4metry. Rozmístění kamer je symetrické, obě kamery jsou 1m vzdálené od stěn místnosti. Natáčením kamer jsme schopni pokrýt celou místnost. Modrou barvou je znázorněno pokrytí prostoru kamerami 1 a 2 ve výchozí pozici – středy zorných polí obou kamer směřují přibližně na vchody. Černé čáry vlevo a vpravo symbolizují vchody do místnosti. Zorné úhly jsou  $58^\circ$  a odpovídají hodnotě udávané v technických parametrech kamery.



Obr. 23 - Monitorovaný prostor se dvěma kamerami

Po definování prostoru a počtu kamer bylo nutné stanovit, co by měla aplikace umět a jak by měla fungovat. Základní požadavky na funkci aplikace byly takovéto:

a) Monitorování prostoru

*Sledování vstupů do místnosti.*

b) Detekce pohybu

*Po detekování pohybu vybrat aktivní kameru.*

c) Sledování pohybu natáčením kamery

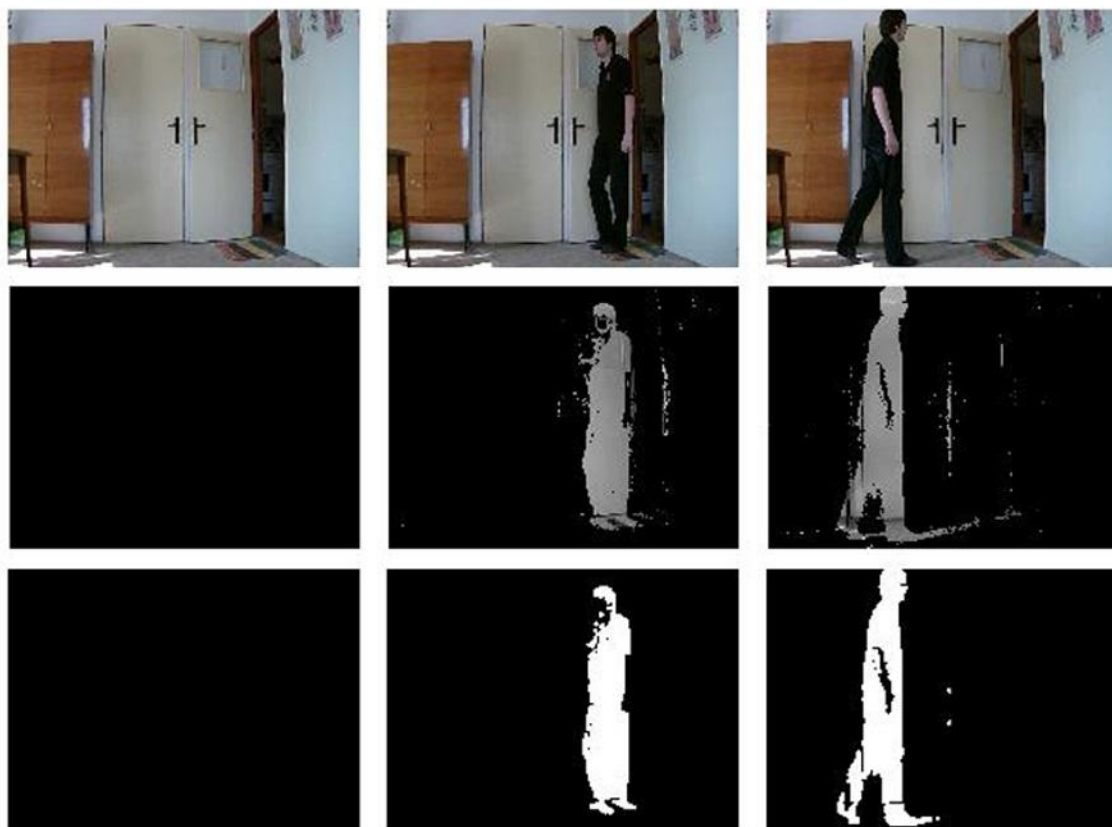
*Kamery se otáčejí podle pohybu objektu v záběru.*

d) Přepnout kamery, když objekt vystoupí z úhlu záběru

*Když objekt opustí záběr jedné kamery, přepnout na druhou.*

### 6.1 Statická kamera, pohybující se objekt

Základním problémem při řešení byl výběr vhodné metody na detekci objektu. Měli jsme pohybující se objekt a pohybující se kameru. Z toho důvodu jsme



Obr. 24 - Výsledky při statické kameře

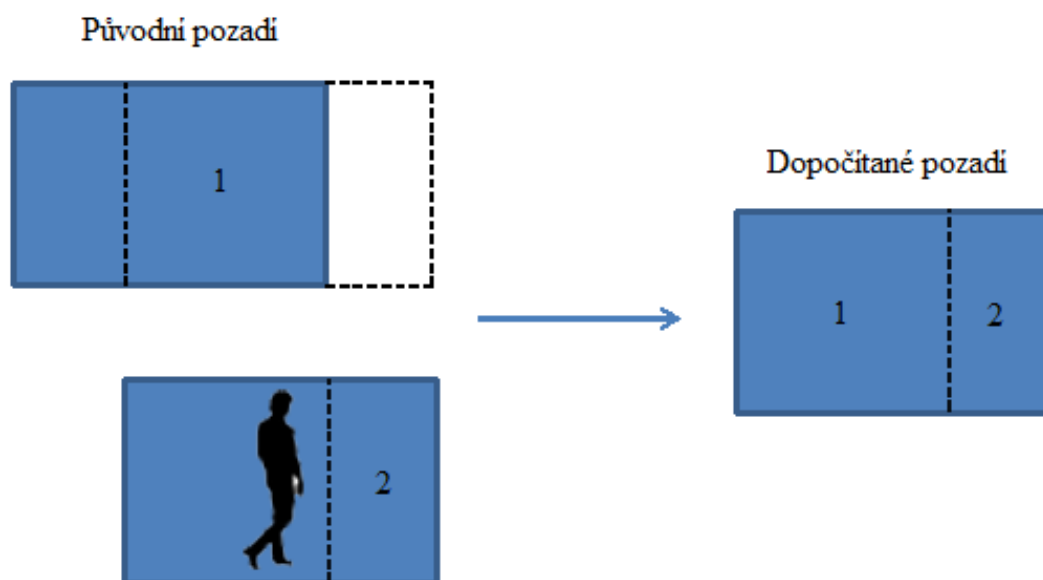
vybrali metodu, která od aktuálního snímku odečítá předzpracované pozadí. Metodu jsme vyzkoušeli při sledování scény, nejprve bez pohybující se kamery. Výsledky jsou vidět na obrázku 24. V první řadě jsou původní snímky z kamery, v prostřední řadě jsou výsledky odečítání a v dolní řadě výsledky po prahování a uzavření. Vidíme, že při statické kameře metoda funguje bez problémů a dává výborné výsledky.

### 6.2 Pohybující se kamera

Po vyzkoušení metody se statickou kamerou jsme mohli pokračovat s pohybující se kamerou. Problémem při snímání pohybující se kamerou je změna pozadí při každém pohybu kamery. Z principu metody je zřejmé, že pro detekci objektu musíme znát přesné pozadí, které budeme odečítat, pro každou pozici kamery. Získat toto pozadí není úplně jednoduché. Vyzkoušeli jsme dvě metody, jak vhodné pozadí získat. První metodou bylo dopočítání z předchozího pozadí. Druhá metoda spočívá v nasnímání pozadí před vlastním spuštěním aplikace na monitorování.

#### 6.2.1 Dopočítávání pozadí

První metodou na zjištění pozadí pro danou pozici kamery, je metoda dopočítání pozadí z pozadí předchozí pozice. Vycházeli jsme z toho, že objekt nevyplňuje celou scénu, ale ideálně jen oblast ve středu záběru. Po pohybu kamery tak můžeme chybějící část pozadí dopočítat. Princip dopočítávání v horizontálním směru je znázorněn na obrázku 25. Stejným způsobem se pozadí dopočítává i ve vertikálním směru. Vzdálenost v pixelech, o jakou se kamera pohne, jsme odečetli pomocí grafického programu. Velikost posunu se lišila pro vertikální a horizontální pohyb kamery. Pro různou rychlost pohybu kamery se posuv v pixelech samozřejmě také lišil. Museli jsme proto odečíst posuv pro všechny rychlosti.



Obr. 25 - Princip dopočítávání pozadí

Hlavním problémem této metody bylo zkreslení objektivu. Při horizontálním posunu dochází u okrajů obrázku k soudkovitému zkreslení obrazu (dochází k efektu „rybího oka“). Toto zkreslení se ještě zvýrazní při několikanásobném posunutí kamery. Je vidět, že pozadí dopočítané (obrázek 26 vlevo) a skutečné (vpravo) se opravdu velice výrazně liší.<sup>v</sup> Následkem tohoto zkreslení nejsme schopni detekovat v daném snímku objekt. Z tohoto důvodu by bylo nutné zkreslení odstranit. Jelikož by odstraňování zkreslení vedlo ke značnému zpomalení celé aplikace, rozhodli jsme se pro jinou metodu získání pozadí.



Obr. 26 - Zkreslení obrázku při dopočítávání

<sup>v</sup> Levý, svislý pruh na obrázku vlevo je stejný, jako na obrázku vpravo, ale vidíme, že jak se po pruzích pohybujeme směrem doprava, začínají se obrázky posupně více a více lišit.

### 6.2.2 Načtení pozadí na začátku

Druhou metodou, jak získat vhodné pozadí pro každou pozici kamery, je načtení celého pozadí ještě před vlastním spuštěním skriptu na monitorování. Pro správnou činnost aplikace je nutné, aby pozadí bylo nasnímáno se stejným krokem posunu, jaký se bude používat při vlastním sledování pohybu. Velikost tohoto kroku, stejně jako počet kroků<sup>VI</sup>, si uživatel nastaví v hlavním GUI.

Načtená pozadí bylo nutné správně indexovat, aby pozadí bylo vždy zpracováváno s odpovídajícím snímkem. Načtená pozadí vidíme na obrázku 27. V našem případě jsme pro urychlení načítání snímali pozadí pouze pro horizontální natáčení kamery.

Při použití této metody se musíme smířit s tím, že načtení pozadí trvá určitou dobu. Pro správný chod aplikace je nutné, aby podmínky při načítání pozadí do databáze byly co nejvíce podobné podmínkám při sledování. Může se stát, že se změní objekty na pozadí nebo světelné podmínky na scéně. V tom případě je nutné nasnímat pozadí do databáze znovu.



Obr. 27 - Načtená pozadí

---

<sup>VI</sup> Ve většině případů není nutné načítat pozadí pro všechny pozice kamery. Pro náš typ prostoru si vystačíme si s načtením pozadí do překryvu zorných úhlů kamer.

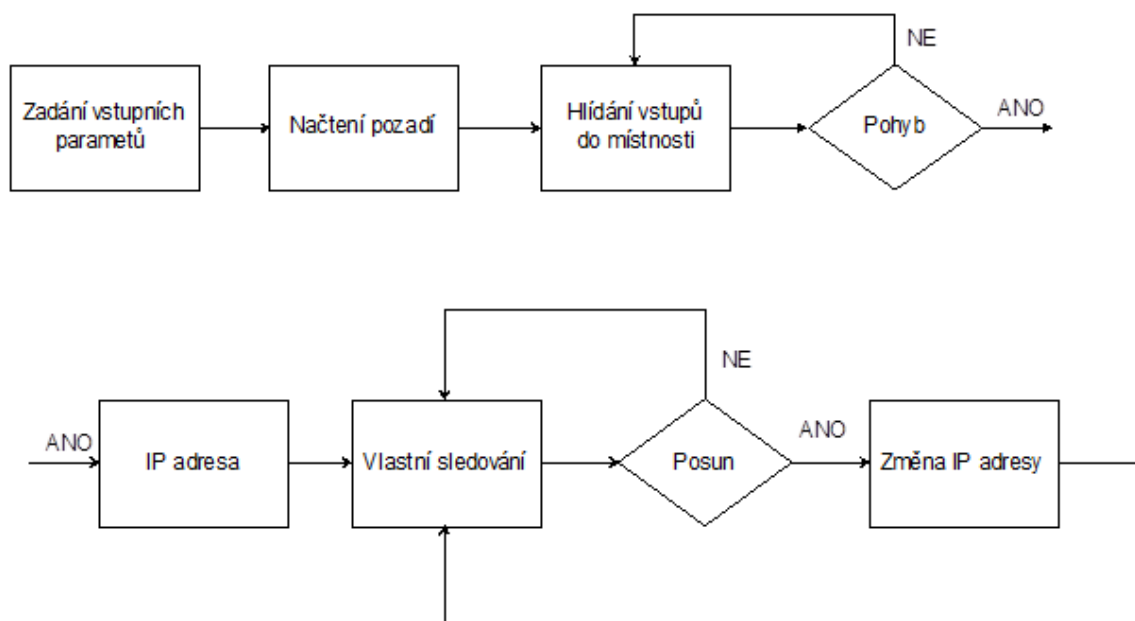
### 6.3 Popis funkce

V této části si popíšeme princip, na kterém aplikace funguje. Budou zde rozedrány jednotlivé bloky vývojového diagramu na obrázku 28.

První dva bloky slouží k zadání vstupních parametrů a k načtení pozadí. Po spuštění aplikace musí uživatel zadat tyto vstupní parametry:

- IP adresu první a druhé kamery
- Rychlost otáčení kamer  $-5\div 5$
- Počet načtených snímků pozadí
- Hranici překryvu záběrů

Po zadání požadovaných hodnot stačí kliknout na tlačítko načíst pozadí a aplikace začne ukládat pozadí do složky. Ukládají se pozadí ze záběru obou kamer.

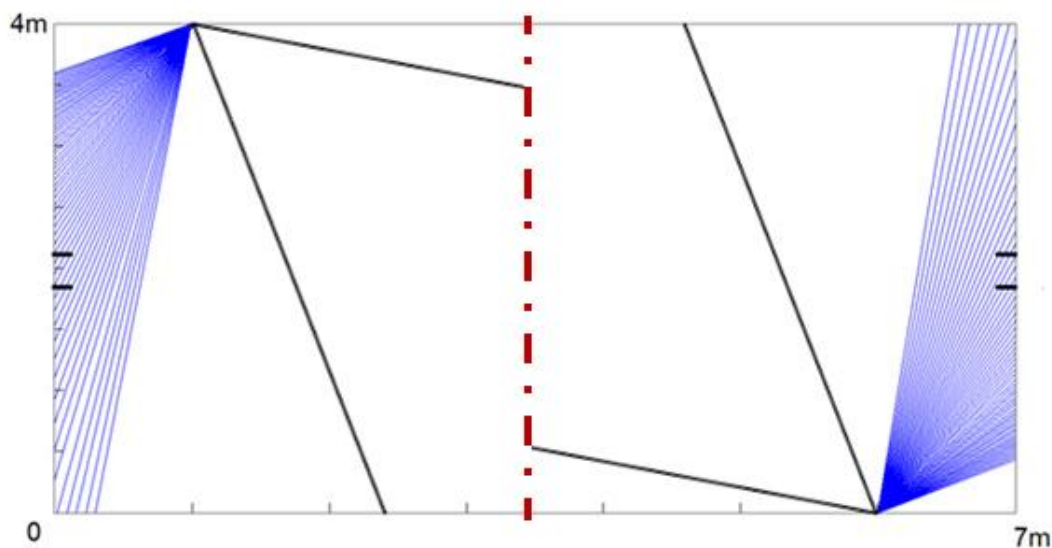


Obr. 28 - Princip funkce aplikace

Po načtení pozadí se spustí funkce, která má za úkol sledovat vstupy do místnosti. Obě kamery jsou v pozici „home“, jsou tedy nasměrované ke vchodům do místnosti. Nastane-li u jednoho vstupu pohyb, funkce vrátí IP adresu té kamery, která pohyb zaznamenala. Tato IP adresa je vstupem do hlavní funkce naší aplikace.



Funkce na sledování je nejdůležitější funkcí celé aplikace. V této funkci se odehrává zpracování aktuálního záběru a pozadí. Z těchto dvou snímků se detekuje objekt a jeho pozice. Ve výsledném obrazu jsme si definovali oblast zájmu - ROI (region of interest). Z rozdílového snímku jsme si vypočítali pozici těžiště objektu. Když se tato pozice nacházela v ROI, neposílal se žádný příkaz. Když se však objekt



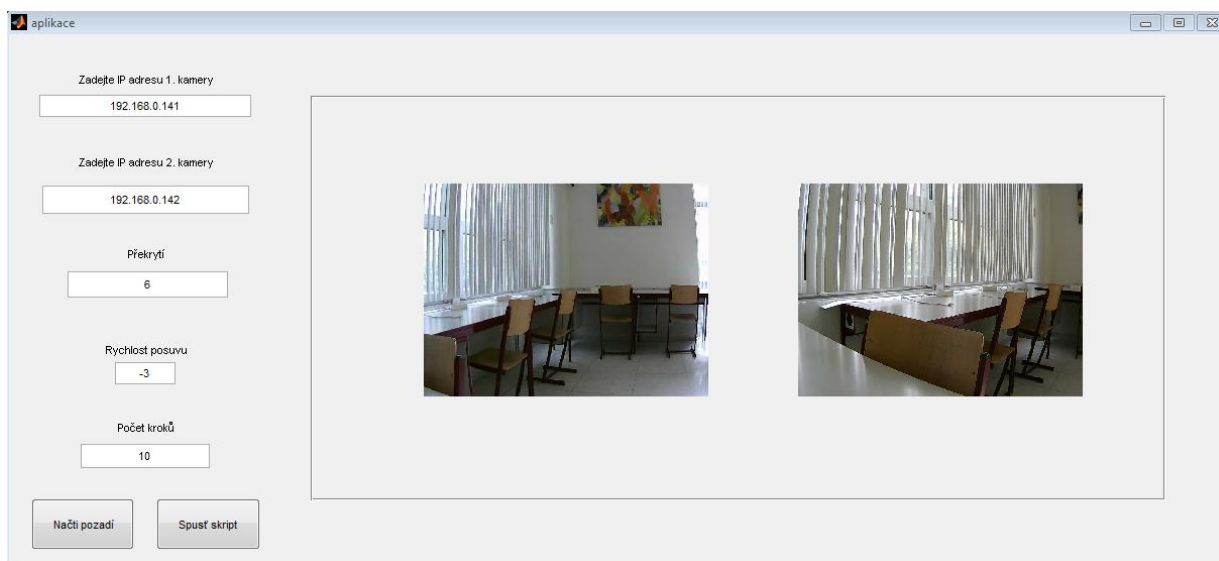
Obr. 29 - Zorná pole kamer při předání

pohnul tak, že poloha těžiště opustila ROI, poslal se příkaz kameře, která se následně natočila tak, aby se těžiště v ROI opět nacházelo. Poté se pro danou pozici definovalo nové pozadí z načtené databáze. Jedním z výstupů této funkce je tedy příkaz k pohybu kamery.

Dalším výstupem je hodnota, udávající pozici kamery. Pohybuje-li se kamera doprava, hodnota se zvýší o 1, při pohybu doleva se o 1 sníží. Přesáhne-li hodnota určitou mez, znamená to, že se překryl záběr obou kamer a aplikace zavolá funkci na přepnutí IP adres. Ilustrační obrázek 29 zobrazuje zorná pole kamer při předání. Modrou barvou jsou zde znázorněna zorná pole ve výchozí pozici, černou barvou při předání objektu. Červenou, čerchovanou čarou je znázorněna hranice pro předání objektu. Přepnutím IP adres se změní ovládání kamer – začne se zpracovávat záběr z druhé kamery a sledování objektu tak pokračuje, ale už druhou kamerou.

## 6.4 Popis GUI

V této části budeme popisovat jednoduché grafické rozhraní, přes které se aplikace ovládá (obrázek 30).



Obr. 30 - GUI

V levé části aplikace si uživatel nastaví IP adresy připojených kamer. Další nastavované parametry jsou rychlost posuvu (-5 až 5), počet kroků a index přechodového pozadí. Po nastavení těchto hodnot je potřeba načíst pozadí. K tomu slouží tlačítko vlevo dole. Po načtení pozadí můžeme spustit monitorování prostoru vedlejším tlačítkem. V pravé části GUI jsou záběry z obou kamer.

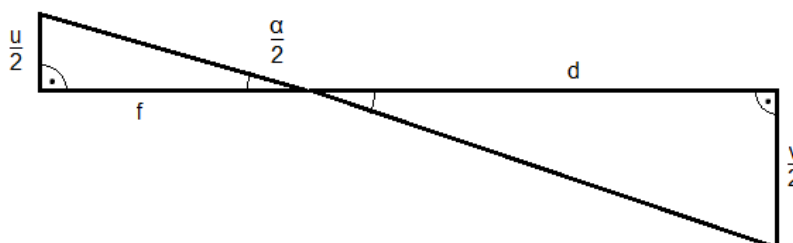
Aplikace není odolná vůči zadávání neplatných hodnot do zadávacích polí. Je potřeba zadat IP adresu v přesně požadovaném formátu, s oddělovacími tečkami. Stejně tak je nutné zadat rychlost otáčení z požadovaného rozsahu. Při zadání špatných hodnot aplikace nevrátí chybové hlášení s výzvou k zadání správných hodnot, ale pouze vypíše chybu v Matlabu.

## 6.5 Další možnosti aplikace

V této části si popíšeme, co dalšího kromě sledování objektu nám aplikace nabízí. Kromě zaměření objektu a jeho sledování, je možné, aby aplikace o objektu prozradila i další důležité informace. My jsme do aplikace implementovali funkci na výpočet vzdálenosti objektu od kamery, funkci na výpočet rychlosti pohybu objektu a funkci na klasifikaci člověka, která nám dá informaci o tom, jestli již daný konkrétní člověk byl v záběru nebo ne.

### 6.5.1 Určení rychlosti a vzdálenosti objektu od kamery

Rychlost pohybu objektu jsme určili velmi jednoduše. Měřili jsme dobu trvání (Matlab funkcí *tic/toc*) mezi prvním pohybem u jednoho vchodu a přepnutím mezi kamerami. Vzdálenost vchodu od místa přepnutí známe, můžeme tedy dopočítat rychlost pohybujícího se objektu..



Obr. 31 - Určení vzdálenosti objektu

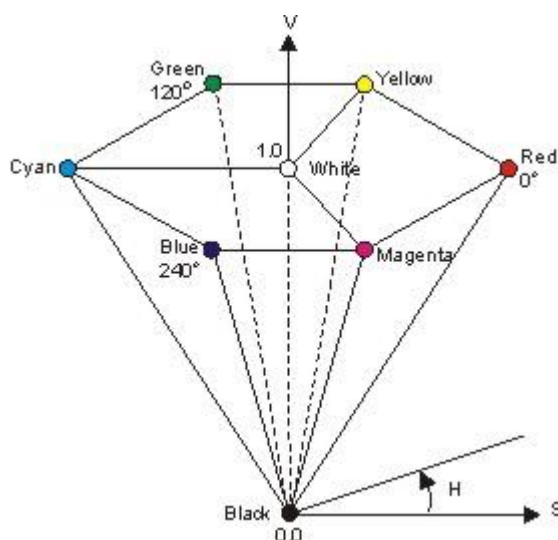
Určení vzdálenosti objektu od kamery je důležitá informace o objektu. Výpočet je založený na poměru skutečné výšky objektu a výšky detekovaného objektu v obraze. Z těchto znalostí spočítáme, jaká výška odpovídá výšce celého obrazu a pomocí sinovy věty můžeme dopočítat vzdálenost.

$$d = \frac{v \cdot \sin 90 - \frac{\alpha}{2}}{2 \cdot \sin \frac{\alpha}{2}} \quad (6.1)$$

V praxi obvykle neznáme přesnou velikost objektu, proto ani takto určená vzdálenost není zcela přesná. V našem případě, kdy detekovaný objekt byl vždy člověk, jsme zvolili velikost 1,75m.

### 6.5.2 Klasifikace objektu podle barvy

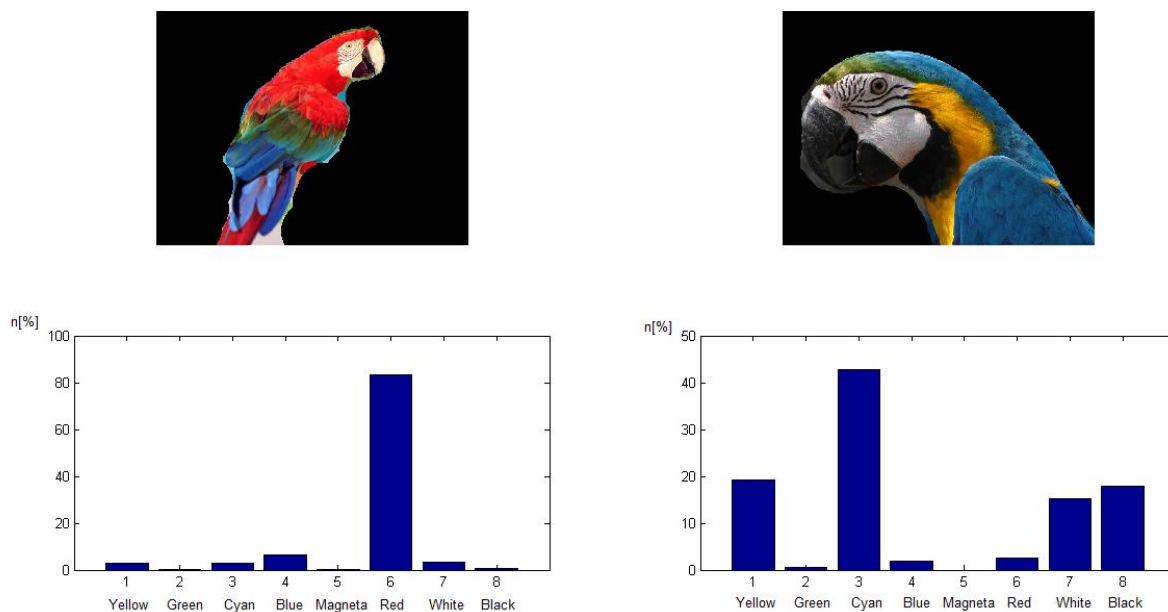
K jednoduché klasifikaci člověka na scéně je možné použít jeho barevné charakteristiky. Předpokládáme, že dva rozdílní lidé budou různě oblečeni a tak bude výsledný barevný histogram těchto dvou lidí odlišný. Pro jednoduchý popis barvy se nám nejvíce hodí barevný prostor HSV (Hue, Saturation, Value). V tomto barevném prostoru je informace o tónu barvy obsažena pouze v jedné proměnné H.



Obr. 32 - Barevný prostor HSV [13.13]

Celý barevný prostor jsme si rozdělili na šest barevných intervalů podle obrázku 32 a další dva intervaly, jeden pro bílou a druhý pro černou. Pro konkrétní objekt jsme poté počítali, kolik procent pixelů přísluší danému barevnému tónu. Vznikl nám tedy histogram, specifický pro každý objekt. Tyto histogramy jsme porovnávali pro různě barevné objekty. Protože černé a bílé pixely se nacházejí poblíž středu barevného prostoru na ose  $V$ , je informace o úhlu  $H$  těchto pixelů zavádějící. Porovnávali jsme tedy pouze pixely s vyšší sytostí barvy. Do výsledného histogramu jsme započítávali pouze pixely se sytostí  $S > 0.5$  a s jasem  $V > 0.2$ . Černé a bílé pixely jsme musely zpracovávat samostatně. Do bílého intervalu jsme započítávali pixely s jasem  $V > 0.5$  a se sytostí  $S < 0.3$ . Pixely jsme brali jako černé, když měly hodnotu jasu  $V < 0.1$ .

Metodu jsme vyzkoušeli na dvojici různobarevných obrázků. Výsledky vidíme na obrázku 33. Jsou zde znázorněny vstupní obrázky pronásobené maskou. V případě člověka nám tuto masku vrátí funkce na detekci objektu. Zbyde nám tedy barevný objekt ohraničený černou plochou.



Obr. 33 - Barevné charakteristiky obrázků

Pod oběma obrázky vidíme sloupcový graf, který znázorňuje, kolik procent pixelů přísluší dané barvě. Je zde vidět, že v prvním obrázku převládá červená barva, proto nejvíce procent pixelů přísluší k šestému sloupci. Na obrázku vpravo máme hned čtyři výraznější barvy – cyan, žlutou, černou a bílou.

Nyní by stačilo vytvořit databázi odpovídajících objektů. Různé objekty by se nasnímaly v různých vzdálenostech a po porovnání výsledků by se stanovil práh v procentech. Rozdíl pod prahem by znamenal, že se jedná o stejný objekt, různé objekty by měly hodnotu rozdílu vyšší než práh.

## 7. Závěr

Hlavním cílem této práce bylo seznámit se způsoby návrhu a optimalizace bezpečnostních kamerových systémů. Protože problematika kamerových systémů je značně rozsáhlá, rozhodli jsme se po domluvě s vedoucím práce, že se v teoretické části práce zaměříme na problematiku kamerových objektivů. Další kapitola teoretické části se zabývá rozбором metod na detekci objektu z kamerového záznamu a metodami na substrakci pozadí. Základní metody jsou podrobněji popsány a porovnány ve čtvrté kapitole. Také jsme zde popsali jednu pokročilejší metodu, konkrétně metodu Viola-Jones.

Praktická část práce se již zabývala reálným návrhem kamerového konkrétního kamerového systému. Nejprve bylo nutné seznámit se s instalací a ovládáním kamery. Technické parametry, instalace a ovládání použitých kamer jsou popsány v páté kapitole. Poté jsme si museli stanovit vlastnosti kontrolovaného prostoru. K lepší orientaci jsme si jako pomůcku naprogramovali aplikaci v Matlabu. Tato aplikace nám vizualizuje rozmístění a zorné úhly kamer v zadaném prostoru. Pomocí této aplikace můžeme tedy přímo vidět, zda je námi zvolený počet kamer pro pokrytí prostoru vyhovující nebo je nutné kamery přidat.

Určili jsme, že na pokrytí námi zvoleného prostoru postačí dvě kamery. Navrhli jsme tedy aplikaci, která bude tento konkrétní kamerový systém obsluhovat. Aplikace monitoruje oba vchody do místnosti a čeká na pohyb. Když nastane u jednoho z vchodů pohyb, vybere se kamera, která pohyb bude sledovat. Po přesunu objektu do oblasti, kde se zorné pole kamer překrývá, se přepne na druhou kameru, která ve sledování objektu pokračuje. Nevýhodou je, že naprogramovaná aplikace je funkční pouze při použití v jednom konkrétním

prostoru. Pro správnou funkci programu je nutné nejdříve nasnímat statické pozadí a až poté aplikaci spustit. Musí být navíc zajištěn dobrý kontrast mezi objektem a pozadím, aby došlo k jasné detekci pohybujícího se objektu. Další nevýhodou je delší doba zpracování záznamu – musíme pokaždé počkat na pootočení kamery, než uložíme další snímek. Tento problém je ovšem velmi zásadní a může v některých případech způsobit nefunkčnost celé aplikace.

Posledním bodem v zadání práce bylo určení vzdálenosti objektu od kamery, směru a rychlosti pohybu sledovaného objektu. Kromě funkce na zjištění rychlosti pohybu jsme tyto funkce z důvodu vyšší rychlosti neimplementovali přímo do aplikace, ale vyzkoušeli jsme jejich funkčnost zvlášť. Vzdálenost objektu se nedá stanovit zcela přesně, ale alespoň jsme se ji pokusili odhadnout pomocí výšky objektu v záběru. Směr pohybu je jasně daný otáčením kamery, která objekt sleduje. Rychlost pohybujícího objektu jsme zjišťovali pomocí času, za který objekt dorazí od vchodu místnosti do oblasti překryvu. Další informací, kterou můžeme o objektu zjistit je jeho barevná charakteristika. Toho jsme využili a navrhli jsme metodu, která nám řekne, jestli naším prostorem již daný objekt procházel. Pro plnou funkčnost by bylo nutné zpracovat výsledky z databáze různých objektů a na základě těchto výsledků vytvořit rozhodovací algoritmus.

## 8. Seznam použité literatury a zdrojů

- [1] AGHAJAN H., CAVALLARO A.,  
Multi-Camera Networks: Principles and Applications  
ISBN 978-0123746337, Academic Press,
- [2] SONG Dezhen, Sharing a Vision,  
ISBN 978-3-540-88064-6, Springer, 2009
- [3] HLAVÁČ Václav; SEDLÁČEK Miloš, Zpracování signálu a obrazu  
ISBN 978-80-01-03110-0, rok vydání 1999
- [4] DOBEŠ Michal, Zpracování obrazu a algoritmy v C#  
ISBN 978-80-7300-233-6, rok vydání 2008 (první vydání)
- [5] RAMSRI Goutham Golla, Real-time face detection and tracking  
Prezentace dostupná:  
[www.dropbox.com/s/17udeu1ojmq8bck/Ramsri\\_Face\\_detection\\_and\\_tracking.pptx](http://www.dropbox.com/s/17udeu1ojmq8bck/Ramsri_Face_detection_and_tracking.pptx)
- [6] YE Yiming, TSOTSOS John K., BENNET Karen, HARLEY Eric  
Proceedings of the 1998 IEEE Workshop on Visual Surveillance (str. 10-17)  
ISBN 0-8186-8320-1, rok vydání 1998
- [7] LIU Yi, XIAO Ming-ming  
Computer Application and System Modeling (str. 361-365)  
ISBN 978-1-4244-7235-2, rok vydání 2010
- [8] SADYKHOV Rauf Kh., KUCHUK Sergey A  
Intelligent Data Acquisition and Advanced Computing Systems (str. 425 - 428)  
ISBN 978-1-4799-1426-5, rok vydání 2013
- [9] DONG Liu, LIN Xi, New Trends in Information Science and Service Science  
(str.692 - 695)  
ISBN 978-1-4244-6982-6, rok vydání 2010



- [10] LAI Kong Chien, CHANG Yeow Peng,  
Computer Engineering and Technology (str. 17-21)  
ISBN 978-1-4244-6347-3, rok vydání 2010
- [11] MARSZALEK Marcin, SCHMID Cordelia  
Computer Vision and Pattern Recognition (str. 1-8)  
ISBN 1-4244-1179-3, rok vydání 2007
- [12] MEINGAST M., Computer Vision (str. 1-8)  
ISBN 978-1-4244-1631-8, rok vydání 2007
- [13] Webové stránky
- (1) <http://www.alpyn.cz>, 2013
  - (2) <http://fyzweb.cz/>, rok 2013
  - (3) <http://photo.mysteria.cz>, rok 2013
  - (4) <http://www.digimanie.cz>, rok 2013
  - (5) <http://www.paladix.cz>, rok 2013
  - (7) <http://docs.opencv.org>, rok 2014
  - (8) <http://elektrika.cz/data/clanky/ochrana-objektu-a-digitalni-kamerove-systemy>
  - (9) <http://www.delta-plzen.cz/zabezpeceni/kamerove-systemy/>, rok 2013
  - (10) <http://www.netrex.cz/>, rok 2014
  - (11) <http://www.a1securitycameras.com/>, rok 2013
  - (12) <http://www.kamerovesystemy.org/>, rok 2013
  - (13) <http://www.blackice.com/colorspaceHSV.htm>, rok 2014
  - (14) <http://digiarena.e15.cz/>, rok 2014
- [14] Manuál ke kameře Vivotek PT8133, rok 2014

## 9. Přílohy

A) Hlavní aplikace na sledování

`/aplikace/hlavni/aplikace.m`

Po spuštění se otevře GUI ovládající aplikaci. Pro správné fungování je třeba, aby byly k PC připojeny obě kamery.

B) Aplikace na ovládání

`/aplikace/ovladani/ovladani.m`

Po spuštění se otevře GUI na ovládání kamery. Pro správné fungování musí být k PC připojena alespoň jedna kamera.

C) Aplikace na vizualizaci pokrytí prostoru

`/aplikace/prostor/prostor.m`

Spouští skript na vizualizaci pokrytí prostoru. Neobsahuje GUI.

D) Viola-Jones algoritmus

`/aplikace/Viola/VJ.m`

Spouští Viola-Jones algoritmus na detekci obličeje v přiloženém videu. Neobsahuje GUI.

E) Barevná charakteristika objektu

`/aplikace/barvy/barvy.m`

Po spuštění zobrazí barevné charakteristiky načtených obrázků.

F) Výpočet vzdálenosti objektu od kamery

`/aplikace/vzdalenost/vzdalenost.m`

Po spuštění spočítá vzdálenost objektu od kamery.