

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## VÝPOČET HLOUBKOVÉ MAPY ZE STEREO KAMERY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

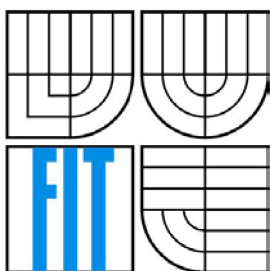
AUTHOR

DANIEL BEREK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# VÝPOČET HLOUBKOVÉ MAPY ZE STEREO KAMERY

DEPTH MAP ESTIMATION FROM STEREO IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DANIEL BEREK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MICHAL ŠPANĚL, Ph.D.

BRNO 2015

## **Abstrakt**

Tato práce je věnována problému stereo vidění a 3D rekonstrukce ze stereo obrazu. Prostorové vnímání je realizováno pomocí hloubkové mapy, k jejímu vytvoření se využívá disparitní mapa. Cílem této práce je vytvořit aplikaci pro odhad vzdálenosti objektů od vozidla s využitím metod pro výpočet hloubkové mapy ze stereo kamery. Součástí práce je také popis jednotlivých kroků pro dosažení cíle, jako např. získání obrazu, kalibrace kamery, rektifikace obrazu a odhad disparity. Výsledkem práce je experimentální řešení pro monitorovací systém do auta, které se zaměří na detekování chodce před vozidlem, a následně pomocí hloubkové mapy vypočítá jejich vzdálenosti. Práce také zahrnuje vytvoření testovacích dat, které slouží pro vyhodnocení přesnosti tohoto systému.

## **Abstract**

This thesis is dedicated to the problem of stereo vision and 3D reconstruction from stereo image. Spatial perception is realized by using depth map, which is created by using disparity map. Main goal of this work is to create an application for estimating object distance from car using methods for depth map estimation. The work also includes description of main steps to achieve goal, like image capturing, stereo camera calibration, image rectification and disparity estimation. The result of this work is an experimental solution for monitoring system used in cars, which focuses on pedestrian detection in front of the car, and calculates its distance by using depth map. Part of this work is also creating test set, which is used for evaluating the accuracy of this system.

## **Klíčová slova**

Hloubková mapa, Mapa disparity, Kalibrace stereo kamery, Detekce chodce, C#, EmguCV.

## **Keywords**

Depth map, Disparity map, Stereo camera calibration, Pedestrian detection, C#, EmguCV.

## **Citace**

Daniel Berek: Výpočet hloubkové mapy ze stereo kamery, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Výpočet hloubkové mapy ze stereo kamery

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Španěla, Ph.D.

.....  
Daniel Berek  
20. května 2015

## Poděkování

Rád bych poděkoval vedoucímu práce panu Ing. Michalu Španělovi za jeho odborné vedení, trpělivost a cenné rady při konzultacích.

© Daniel Berek, 2015

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

1	Úvod.....	2
2	Zpracování obrazu pro odhad vzdálenosti .....	3
2.1	Stereovidění .....	3
2.2	Model dírkové kamery.....	3
2.3	Kalibrace kamery.....	4
2.4	Metody výpočtu disparity mapy .....	5
2.5	Triangulace .....	6
2.6	Detekce .....	7
3	Asistenční systémy.....	8
3.1	Druhy snímačů pro asistenční systémy.....	8
3.2	Kamerové systémy.....	9
3.3	Asistenční systémy vybaveny kamerou.....	10
4	Návrh metody pro měření vzdálenosti chodců od vozidla.....	12
4.1	Získání snímků .....	13
4.2	Kalibrace kamery.....	14
4.3	Rektifikace obrazů .....	14
4.4	Odhad disparity.....	15
4.5	Výpočet vzdálenosti.....	15
4.6	Detekce chodců.....	16
4.7	Segmentace .....	16
5	Implementace.....	18
5.1	Programové prostředí.....	18
5.2	Použité metody .....	18
5.3	Testovací aplikace .....	20
6	Testování a hodnocení .....	22
6.1	Návrh testování.....	22
6.2	Popis prováděných experimentů.....	23
6.3	Diskuze .....	29
	Závěr.....	30
A	DVD.....	34
B	Plakát .....	35

# 1 Úvod

V současnosti jsou asistenční systémy již samozřejmostí v každém automobilu, jelikož výrazně zvyšují bezpečnost a pohodlí řidiče i ostatních účastníků silničního provozu. Můžeme je rozdělit na aktivní a pasivní. Aktivní systémy jsou schopné předcházet nehodě a patří mezi ně např. stabilizační a blokovací systémy. Většinou podporují vozidlo a působí v kritických situacích přímo, aniž by řidič mohl zabránit jejich působení. Naopak pasivní systémy spíše zmírňují následky nehody (airbagy, pásy, deformační zóny) [1]. Systémy podporující řidiče jsou používány zejména v luxusních vozech a upozorňují řidiče na možná nebezpečí. Hlavní součástí jsou radarové nebo kamerové snímače, které získávají informace o odstupu vpředu jedoucích vozidel nebo o překážkách před vozidlem.

Bakalářská práce pojednává o metodách použitých v této oblasti, zaměřené na detekování chodce, a kde pro detekování a odhad vzdálenosti je používána stereo kamera. Odhad vzdálenosti je realizován pomocí hloubkové mapy. Cílem této práce je navrhnout a implementovat aplikaci pro odhad vzdálenosti objektů od vozidla s využitím hloubkové mapy získané ze stereo kamery. Aplikace je implementována v programovacím jazyce C# s využitím knihovny EmguCV.

Abychom se orientovali v oblasti stereovize v rámci počítačového vidění, je nutné se seznámit s teoretickými základy. Následující kapitola 2 obsahuje teoretické základy, na kterých stojí navrhovaný systém, jako např. základy stereovidení, kalibraci stereo kamery a její použití, důvod použití epipolární geometrie a model dírkové komory. Kapitola 3 obsahuje popis některých dostupných asistenčních systémů, ze kterých jsem se inspiroval při tvorbě této práci. Kapitola 4 se již zabývá návrhem pro výpočet vzdálenosti a detekování chodce, a kapitola 5 obsahuje implementace vlastní práce spolu s použitým nástrojem. Kapitola 6 popisuje používaná testovací data, experimenty, které byly vykonané s programem a následně jejich vyhodnocení. V závěrečné kapitole 7 jsou popsány dosažené výsledky, možná vylepšení či možnosti dalšího vývoje a zhodnocení celé práce.

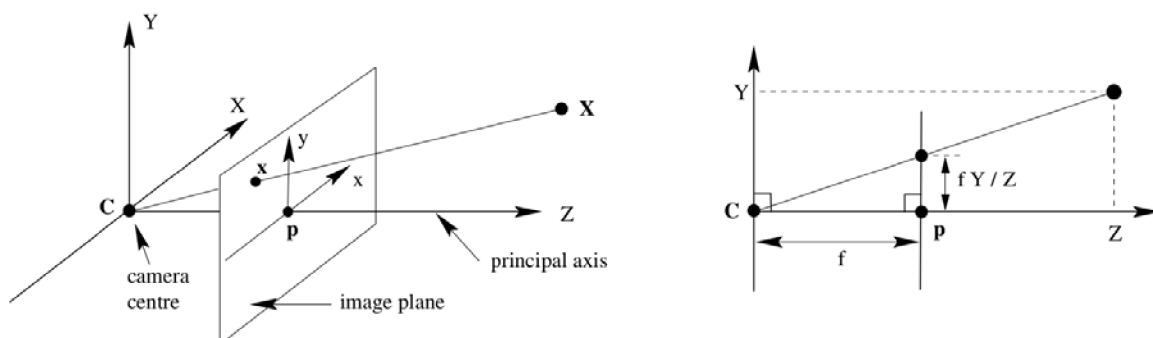
## 2 Zpracování obrazu pro odhad vzdálenosti

### 2.1 Stereovidění

Stereovidění je obor, který se zabývá rekonstrukcí třetího rozměru, tedy vzdáleností v prostoru. Prostorového vnímání, neboli stereovidění, je možné dosáhnout i s jedním okem, při takovém vnímání se v prostoru využívají tzv. monokulární podněty. Prostorového vnímání je však většinou dosáhnuto pomocí dvojice kamer, na které je zaměřena i tato práce. Vnímání objektu z různých pohledů umožňuje pasivním způsobem dopočítat vzdálenost objektu od stereoskopických senzorů. Pasivní způsob znamená, že senzory během pozorování při získávání obrazových informací žádným způsobem nezasahují do scény, využijí jen pocházející světlo ze scény [19].

Abychom mohli určit mapování 3D bodu pozorované scény do 2D obrazového bodu a popsat jej matematickými vztahy, musíme nejprve zvolit geometrický model kamery. Nejprimitivnějším modelem kamery je model dírkové kamery ang. pinhole camera model.

### 2.2 Model dírkové kamery



Obrázek 2.1: Model dírkové kamery [14].

Jak lze vidět na obrázku 2.1, bod středového promítání  $C$  je zároveň středem souřadného systému kamery. Od bodu  $C$  ve vzdálenosti  $f$  se nachází rovina promítání, která je kolmá na přímce  $Z$ . Přímka  $Z$  je reprezentovaná jako optická osa kamery. Průsečík této osy a roviny promítání je definován jako bod  $p$ . Principiální bod  $p$  je zároveň středem 2D souřadného systému kamery. Model vykonává středovou projekci. Perspektivní promítání mapuje bod  $X(X, Y, Z)$  z 3D prostoru do 2D prostoru (rovina promítání) skrz bod  $x(x, y)$  pomocí přímky projekce [15] [16]. Projekční přímky procházejí v jednom bodu  $C$ . Mapování, tedy projekci můžeme vyjádřit zjednodušeným maticovým zápisem:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.1)$$

V takovém tvaru je obrazový bod v souřadném systému kamery. V praxi je ale vhodnější, abychom body měli v obraze na základě pixelových souřadnic. Přejít na obrazové souřadnice může být proveden s využitím transformační matice (kalibrační matice). Celkový proces lze vyjádřit matematickým modelem (převzatým z [17]):

$$s \cdot m' = K \cdot [R|t] \cdot M', \quad (2.2)$$

neboli maticovým zápisem:

$$s \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \left[ \begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{array} \right] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (2.3)$$

kde:

(u, v) jsou pixelové souřadnice projekce bodu,

(X, Y, Z) jsou globální souřadnice bodu v 3D prostoru,

K je kalibrační matice kamery obsahující vnitřní parametry, kde:

$c_x, c_y$  jsou středy obrazu,

$f_x, f_y$  jsou ohniskové vzdálenosti v pixelech,

R, t jsou matice rotační a translační – vnější parametry kamery.

## 2.3 Kalibrace kamery

Pro zjištění vnitřních a vnějších parametrů kamery (uvedené v rovnici 2.2) můžeme použít metodu kalibrace. Tudiž přesnost celého systému je velice závislá na kalibraci. Kamery zaměříme na známou strukturu, která má mnoho identifikovatelných bodů a díky tomu po rozpoznání významných bodů můžeme vypočítat vnitřní a vnější parametry kamery. Kalibruje se vytvořením určitého množství obrazů pod různými úhly a vzdálenostmi. Kalibrace se nejčastěji provádí pomocí kalibračního vzoru např. šachovnice. Pokud používáme stereo kamery, pak vzájemné pozice mezi kamerami zjistíme také kalibrací, která je odvozena od základu epipolární geometrie.

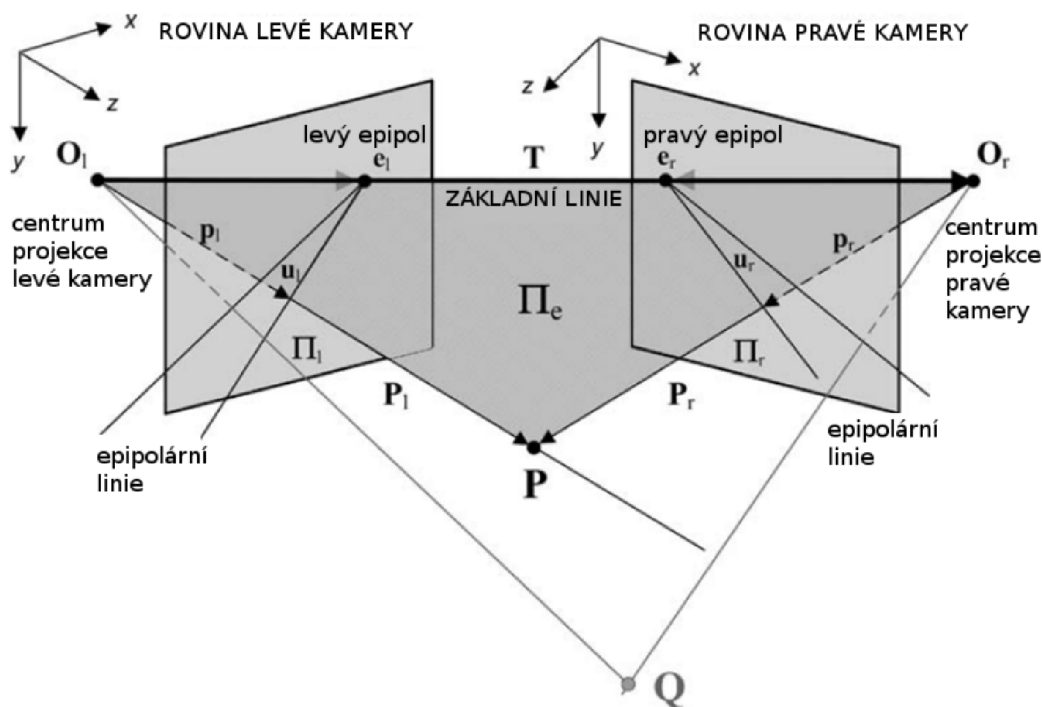
### Epipolární geometrie

Tato část teorie je převzatá z disertační práce [18].

Epipolární geometrie je vnitřní projektivní geometrie mezi dvěma pohledy kamer, je nezávislá na struktuře scény a je závislá pouze na vnitřních a vnějších parametrech kamery. Tyto parametry jsou reprezentovány fundamentální maticí  $F$ . Máme-li dva pohledy kamer a 3D bod  $P$  ze scény, pak bod  $P$  je zobrazen v prvním pohledu jako bod  $p_r$  a v pohledu druhém jako  $p_l$ . Platí vztah:

$$p_r^T F p_l = 0 \quad (2.3)$$





Obrázek 2.2: Epipolární geometrie pohledu dvou kamer [18]

Průnikem epipolární roviny  $\Pi_e$  a projekční roviny kamery  $\Pi_r$  vznikne epipolární linie (přímka)  $u_r$ . Body  $p_l$  a  $p_r$  se nachází v jedné epipolární rovině. Nejdůležitějším důsledkem epipolární geometrie je mapování bodu  $p_l$  na přímku  $u_r$ . Klíčový vztah pro výpočet je:

$$u_r = Fp_l \quad (2.4)$$

Tento vztah se často využije při hledání korespondencí. Pokud v jednom obraze máme bod  $p_l$ , pak pro nalezení korespondujícího bodu  $p_r$  nemusíme prohledat celý pravý obraz, ale stačí jen hledat na přímce  $u_r$ .

## 2.4 Metody výpočtu disparity mapy

Disparitám v základním pojmu rozumíme jako rozdíl mezi hodnotami. V rámci stereovidění je to rozdíl mezi pozicemi stejného 3D bodu z prostoru v levém a pravém obraze. Mapa disparity tedy obsahuje pro každý pixel v levém obraze vzdálenost (v pixelech) mezi dvěma odpovídajícími si body. Stereo korespondenční metody se zabírají touto problematikou, tedy nalezením korespondujících si bodů.

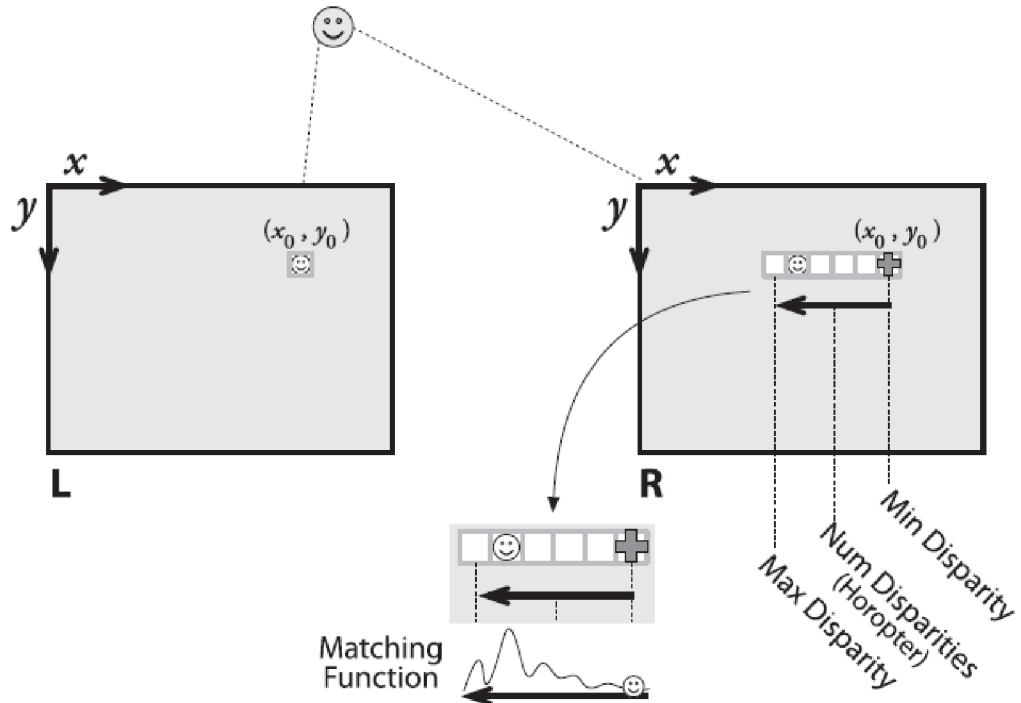
Hojně používané metody, které jsou dostupné i v knihovny OpenCV, jsou metody:

- Block Matching – BM (algoritmus popsáný v odborném článku [20])
- Semi-Global Block Matching – SGBM (algoritmus popsáný v odborném článku [21])

Oba algoritmy porovnávají sousední pixely po blocích. Zatímco BM porovnává sousední pixely jen na korespondujícím řádku za použití *SAD* (*Sum of Absolute Difference*) oken, SGBM porovnává bloky ve čtyřech nebo osmi směrech [6]. Je z toho zřejmé, že algoritmus SGBM najde mnohem víc korespondující si bodů, ale zároveň je pomalejší při výpočtech.

Block-Matching algoritmy mají 3 fáze pro nalezení korespondencí [6]:

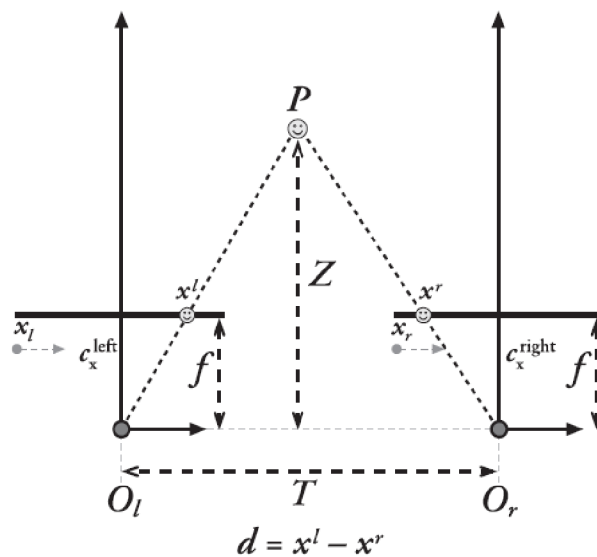
1. Normalizace obrazového jasu a textury
2. Hledání korespondencí na odpovídající epipolární přímce za použití SAD oken
3. Filtrace za účelem eliminace falešných korespondencí



Obrázek 2.3: Korespondující si body jsou umístěny na stejné epipolární řádce (vypočítané z rovnici 2.4) a s metodou BM pak určíme rozsah hledání na řádce [6].

## 2.5 Triangulace

Když používáme dvě kamery dírkového modelu, a navzájem jsou optické osy kamery rovnoběžné, pak na základě triangulace lze vypočítat hloubku scény.



Obrázek 2.4: Triangulace dvou kamer [6].

Z obrázku 2.4 lze vidět, že hloubku  $Z$  lze rekonstruovat pomocí známého parametru kamery, kde:

$O_l, O_r$  jsou středy kamer,

$T$  je vzdálenost mezi kamerami,

$f$  je ohnisková vzdálenost,

a  $d$  je disparita vypočítaná s využitím stereo korespondenčních algoritmů, kde:

$x^l - x^r$  jsou korespondenční body umístěné na jedné řádce.

Pak můžeme popsat rovnici [6]:

$$\frac{T - (x^l - x^r)}{Z - f} = \frac{T}{Z} \quad (2.5)$$

Po zjednodušení dostaneme vzdálenost:

$$Z = \frac{fT}{d} \quad (2.6)$$

## 2.6 Detekce

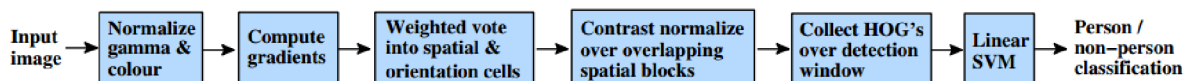
V rámci bakalářské práce budu také využívat rozpoznávání objektů z obrazových dat. V této části popisují stručný přehled základů detekce.

Existuje řada detektorů, které jsou určeny pro různé typy úloh, oblasti využití a složitosti hledaných tvarů. Složité tvary se ve většině případů nedají popsat analyticky. Takový detektor se snaží zjednodušit obrazová data a nechá jen podstatné vlastnosti obrazů, které jsou typické pro daný tvar.

Na zjednodušení slouží různé deskriptory, které popisují společné vlastnosti tohoto tvaru. Např. při detekování obličejů můžeme určit relaci mezi očima a ústy, ale také může být deskriptorem barva nebo geometrický tvar. Pak rozpoznávání funguje na základě strojového učení, kde se pomocí testovacích sad naučí, jak vypadají jednotlivé tvary. Aby detektory mohly úspěšně fungovat, musí zpracovat tisíce dat. Často se proces detekce skládá z následujících operací [22]:

- Extrakce pozadí/popředí
- Extrakce objektu
- Klasifikace objektu
- Sledování objektu
- Vhodná reakce na detekovaný objekt

Pro detekci osob se nejčastěji používají detektor HOG (Histogram of Oriented Gradient). Základním principem detektorů je rozdělení obrázku na menší spojité části, kde v každé buňce je vypočítán lokálně histogram směru gradientu nebo orientace hran. Každá buňka je diskretizovaná podle směru jeho gradientu. Základem normalizace histogramu je seskupení buněk do větších bloků, čímž budou odolné vůči změnám světelných podmínek a kontrastu.



Obrázek 2.5: Kompletní postup detekce při kombinovaném HOG deskriptoru a lineárních SVM klasifikátorů [7].

## 3 Asistenční systémy

Na bezpečnost silničního provozu má zásadní vliv chování všech účastníků silničního provozu, tedy chodců, cyklistů a především řidičů dopravních prostředků. Příčinou mnoha nebezpečných situací nebo nehod je stále lidský faktor. V důsledku toho vznikly asistenční systémy, které upozorňují řidiče na nebezpečné situace, nebo přímo zasahují do řízení vozidla, aniž by řidič mohl zabránit jejich působení. Podle toho můžeme rozdělit asistenční systémy do dvou skupin [23]:

- Asistenční systémy podporující vozidlo
- Asistenční systémy podporující řidiče

### Asistenční systémy podporující vozidlo

Tyto systémy působí v kritických situacích, a většinou řidič ani nepozná účinky systémů. Mezi tyto systémy patří například [23]:

- Protiblokovací systém ABS
- Elektronická stabilizace jízdy ESP
- Regulace prokluzu pohonu ASR

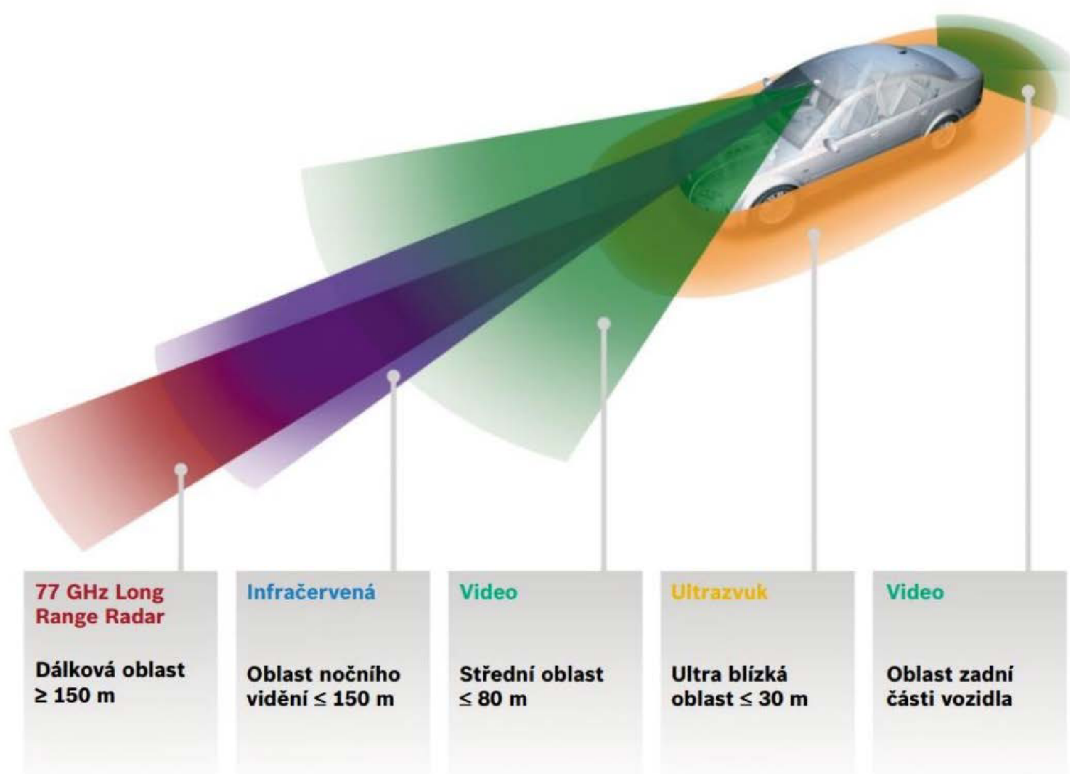
### Asistenční systémy podporující řidiče

Tyto systémy do řízení vozidla většinou nezasahují, poskytují jen informace o možných nebezpečích, a díky tomu řidič získává větší přehled o dopravní situaci. Mezi tyto systémy patří například [23]:

- Adaptivní kontrola vzdálenosti ACC
- Vizuální zobrazovač HUD
- Parkovací asistenční systém APS
- Asistenční systém udržování jízdního pruhu LDW
- Systém rozpoznávání dopravních značek

## 3.1 Druhy snímačů pro asistenční systémy

Většinou, aby tyto systémy mohly splnit svoji funkci, získávají informace z okolního prostředí právě díky snímačům. Různé snímače splňují různé úkoly a mají určitý dosah podle jejich charakteristiky. Jak na obrázku 3.1 lze vidět, můžeme snímače rozdělit podle jejich dosahu.



Obrázek 3.1: Dosah jednotlivých snímačů [24].

Radarové snímače se používají pro určování rychlosti jedoucích vozidel. Rychlost zjišťují pomocí Dopplerova jevu při vysílání a přijímání elektromagnetického vlnění [25].

Infračervené snímače se podobají video snímačům, ale jsou schopny přijímat záření, které leží mimo rozsah viditelného světla, a mají vyšší dosah. Ultrazvukové snímače slouží pro měření objektů bližších než 5 metrů, a nejčastěji se používají pro parkovací systémy[23].

Můj systém je založený na využití obrazových snímačů. Využívají se v systémech např. pro udržování vozidla v jízdním pruhu nebo rozpoznávání dopravních značek či vozidla. V další kapitole jsou popsány základní parametry kamerových systémů.

## 3.2 Kamerové systémy

Většinou se kamerové systémy skládají z obrazového snímače a objektivu. U levnějších kamer se nejčastěji používají plastové objektivy, které jsou integrovány na snímače. V případě kvalitnějšího typu kamery jsou objektivy vyrobeny z kvalitních čoček a jsou vyjímatelné ze snímače dle potřeby. Mezi základní parametry kamery patří rozlišení a citlivost snímacího čipu, snímková frekvence a ohnisková vzdálenost [3] [13].

- Rozlišení kamery určuje počet bodů (pixelů), které je schopen snímací čip zobrazit. Čím víc má obraz pixelů, tím více bude obsahovat detailů.
- Citlivost snímacího čipu udává, v jakém míře je sensor citlivý na světlo. Citlivé senzory poskytují čistý a jasný obraz i při horších světelných podmínkách. Existují dva typy senzorů:

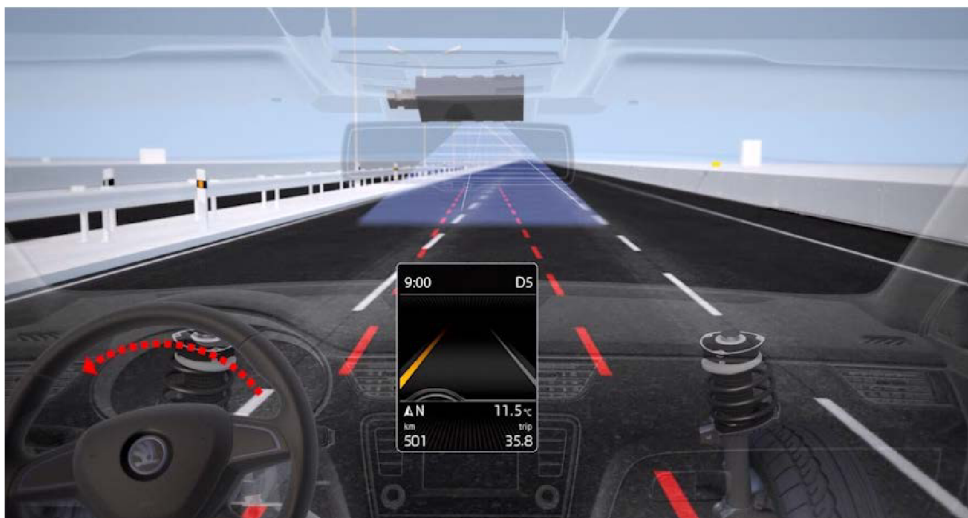
CCD (Charge Coupled Device) a CMOS (Complementary Metal Oxide Semiconductor) senzor. Starší, ale kvalitnější (citlivější), jsou CCD senzory, které byly vyvinuty pro kamerový průmysl. Citlivost je ovlivněna i velikostí snímacích pixelů, protože na větší pixely dopadá více světla.

- Snímková frekvence určuje, kolik snímků za sekundu je kamera schopna poskytnout.
- Ohnisková vzdálenost čočky určuje horizontální šířku zachyceného obrazu v určité vzdálenosti. Čím vyšší bude hodnota, tím užší bude zachycený obraz. Pokud chceme zachytit objekty v širokém okolí, je nutné vybrat čočky s menší ohniskovou vzdáleností.

### 3.3 Asistenční systémy vybaveny kamerou

#### Škoda Lane Assistant a Traffic Sign Recognition [26]

Systémy využívají kameru, která je umístěná na vnitřním zpětném zrcátku. Systém Lane Assistant při opuštění jízdního pruhu zobrazí vizuální informaci a zahájí i korekční řízení. Systém Traffic Sign Recognition rozpozná a zobrazí dopravních značky.



Obrázek 3.2: Asistenční systém Lane Assistant [26].

## Subaru EyeSight®<sup>[2]</sup>

Společnost Subaru vyvinula systém, který dokáže udržet vzdálenost od vpředu jedoucího auta, rozpozná chodce a cyklisty s pomocí stereo kamery umí i vypočítat jejich vzdálenost, směr a rychlost pohybu. Kamery spolupracují i se spalovacím motorem, převodovkou a s brzdovými systémy, a v případě potřeby zahájí i korekci. Systém získal nejvyšší skóre na testech IIHS – prevence čelní srážky [27].



*Obrázek 3.3: Systém EyeSight® používá stereo kamery namontované na čelní sklo [28].*

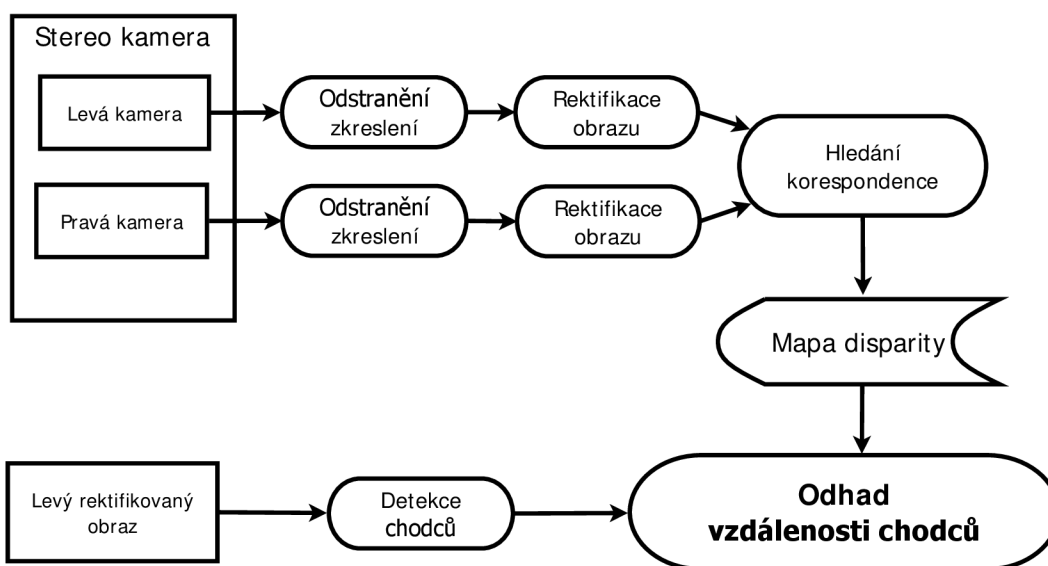
## 4 Návrh metody pro měření vzdálenosti chodců od vozidla

Cílem této práce je vytvořit systém, který podporuje řidiče a upozorní jej v případě blížící se překážky. Lze to realizovat pomocí stereo kamery, díky čemuž po získání stereo obrazu lze pasivním způsobem dopočítat vzdálenost snímaných objektů, jak je popsáno v kapitole 2. Nejefektivnější využití je ve městech a osídlených oblastech, protože silniční provoz je tam hustější, tudíž vozidla mají nižší rychlost, při které tyto systémy fungují nejlépe. Při návrhu jsem se inspiroval již existujícími systémy, jako např. systém EyeSight® (popsáno v kapitole 3.3). Oblast, na kterou jsem se zaměřil, je rozpoznávání chodců na cestě, na přechodu nebo kdekoli v určité blízkosti před autem a výpočet jejich vzdálenosti.

Výsledná aplikace by měla upozornit řidiče při detekci chodce a také by měl odhadnout jejich vzdálenosti pomocí hloubkové mapy. Návrh řešení přebírá a kombinuje metody z již existujících řešení. Navrhovaný systém se skládá z několika částí:

- Získání snímků
- Kalibrace kamery
- Rektifikace obrazů
- Odhad disparity
- Výpočet vzdálenosti
- Detekce chodců
- Segmentace

Na obrázku 4.1 lze vidět závislosti jednotlivých částí, a také ukazuje kroky, jak dostanu ze vstupních obrazů k výstupem aplikaci.



Obrázek 4.1: Návrh výsledného systému.



## 4.1 Získání snímků

Jak již bylo zmíněno, oblast využití tohoto systému bude v silničním provozu, a to pro zvýšení bezpečnosti. Kamery budou používat pro monitorování prostoru před vozidlem a mají významnou roli v rámci této systému, kvůli tomu je nutné vybrat vhodný typ kamery. V navrhovaném systému jsou potřebné dvě stejné kamery. Kamery obsahují citlivé senzory, které umožňují snímání i ve špatných světelných podmínkách. Z hlediska detekování objektů jsem vybral vhodné rozlišení obrazu a zvýšení snímkovací frekvence, která umožňuje detekování i v případě rychlého pohybu. Také jsem vybral čočky s menší ohniskovou vzdáleností, abych jsem věděl zachytit objekty v širokém okolí. Na obrázku 4.2 lze vidět použitá USB 3.0 kamera od společnosti E-con systems.



Obrázek 4.2: Použitá kamera v rámci této práce [4].

Následující tabulka 4.1 obsahuje významné parametry kamer a také nastavení, které budu používat pro pořízení snímků.

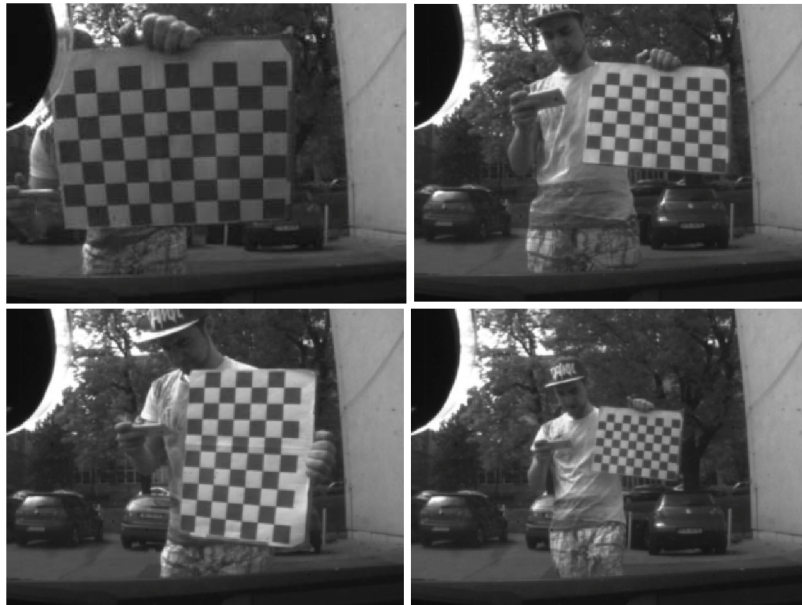
Rozlišení obrazu	640 x 480 pixelů
Snímkovací frekvence	45 snímků za sekundu
Úhel pohledu	53,8 stupňů
Světelnost	F/1.4
Velikost snímacího čipu	4,8 x 3,6 mm
Typ snímače	CMOS
Barevný model	RAW Bayer

Tabulka 4.1: Použité nastavení kamer pro pořízení snímků.

I kvalitní čočky způsobují zkreslení obrazu, což je zvětšení okrajových bodů předmětu oproti bodům ve středu. Pro úspěšnou rekonstrukci obrazových dat je nutné toto zkreslení odstranit. V následující sekci je popsáno, jak budu eliminovat tuto chybu.

## 4.2 Kalibrace kamery

V rámci mé práce jsem se rozhodl, že pro kalibraci použiji obrázek šachovnicového vzoru (podle článku [5]), na kterém lze snadno rozpoznat kontrastní body ze scény v místech křížení bílých a černých čtverců. Jediná věc, kterou potřebují znát, je velikost čtverců na vzoru. Šachovnici budu vytisknout nejméně na formát A3, abych věděl kalibrovat ze vzdálenosti až jednoho metru. Během kalibrace budu vytvořit snímky pod různými úhly a vzdálenosti tak, aby všechny vnitřní rohy vzoru byly viditelné.



Obrázek 4.3: Ukázka jak bude probíhat kalibrace pomocí šachovnicového vzoru.

## 4.3 Rektifikace obrazů

Pro odhad disparity ze dvojice snímků je nejlepší, když kamery jsou identické a korespondující body v obou snímcích byly ve stejné výšce. Této polohy můžu dosáhnout v případě, že kamery jsou posunuty jen na horizontální ose, a jejich optické osy jsou rovnoběžné (protínají v nekonečnu) a neobsahovaly zkreslení obrazu. Také je vhodné, aby kamery měly stejnou ohniskovou vzdálenost a další vnitřní parametry. Teorie epipolární geometrie a korespondence jsou popsány v kapitole 2.6. Tato konfigurace je v praxi téměř nedosažitelná, i když jsou kamery od jednoho výrobce, protože neexistují dokonalé čočky a polohu nemůžeme dokonale nastavit. Pro dosažení této konfigurace používám rektifikace stereo snímků.

Rektifikace budu provádět s využitím kalibrace, především ze zjištěných vnitřních a vnějších parametrů kamery. Na Obr. 4.2 lze vidět, že korespondující si body jsou umístěny na jedné řádce. Výhodou je, že po nalezení korespondujícího bodu v levém obraze již nemusím prohledat celý pravý obraz.



Obrázek 4.4: Po rektifikaci korespondující si body na jedné řádce

## 4.4 Odhad disparity

Jak již bylo zmíněno v kapitole 2.4, disparita vyjadřuje rozdíl mezi body v levém a pravém obraze, které jsou projekcemi stejného 3D bodu na scéně. Rozhodl jsem se, že pro odhad disparity pro detekování z auta použiji metodu Block Matching, protože oproti metodě Semi-Global Block Matching porovná bloky jen na korespondujících řádcích, z čehož vyplývá, že sice kvalita výsledků bude horší, ale proces bude výrazně rychlejší. Cílem je, abych po detekování chodců stanovil jejich vzdálenosti, to znamená, že není třeba kvalitní výpočty a přesné kontury. Rychlost je důležitým aspektem, protože po detekování chodců musí být jejich monitorování stálé a s co nejmenší časovou odezvou.

Výstupem Block Matching algoritmu je disparitní mapa, která je potřebná pro tvorbu hloubkových map a též pro výpočet vzdálenosti.

## 4.5 Výpočet vzdálenosti

Hloubková mapa je obraz, kde každý bod nese informace o jeho hloubce v prostoru neboli v mém případě vzdálenost od objektu. Vzdálenosti jsem odhadnul na základě triangulace, popsané v Kapitole 2.5. Při použití triangulaci je nutné, aby optické osy kamery byly rovnoběžné a měli stejnou ohniskovou vzdálenosti a stereo obrazy byly rektifikované.

Pro výpočet vzdálenosti jsem použil rovnice 2.6 z kapitoly 2.5. Pro její uplatnění musím určit ohniskovou vzdálenost a vzdálenost mezi kamerami. V rámci této práce hodnoty budu získat z reprojekční matice  $Q$  [6], která získám po rektifikaci obrazu.

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T_x} & (c_x - c'_x)/T_x \end{bmatrix}, \quad 4.1$$

kde:

$f$  je ohnisková vzdálenost.

$c_x, c_y$  jsou x, y-ové souřadnice principiálního bodu levé kamery.

$c'_x$  je x-ové souřadnice principiálního bodu pravé kamery.

$T_x$  je vzdálenost mezi kamerami.

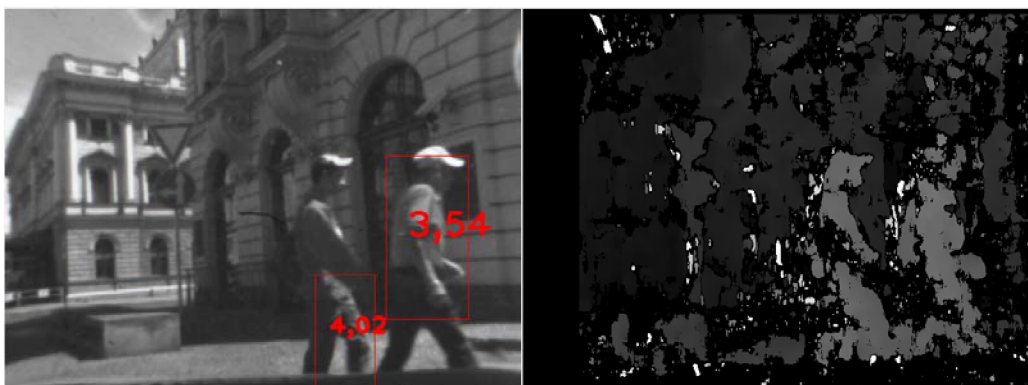
Při výběrů tohoto řešení musím zabezpečit správnou kalibraci kamery.

## 4.6 Detekce chodců

V rámci mé práce jsem se rozhodl, že budu používat již existující detektor, protože vytvoření vlastního detektoru by bylo časově náročné vzhledem k času pro psaní bakalářské práce. V této práci jsem se zaměřil na detekování chodce, takže jsem vybral detektor, který je specializovaný právě na tento účel. Podle výzkumu Navneeta Dalala a Billa Triggse [7], kde bylo srovnáno více detektorů pro detekování lidí, vynikajícího výsledku dosáhl detektor HOG (Histograms of Oriented Gradients). Testy byly provedeny na více než 1800 obrázcích. Výhodné je, že detektor je implementovaný i v knihovně OpenCV. Základný princip fungování detektoru je popsán v kapitole 2.

## 4.7 Segmentace

Po úspěšné detekci dostanu pozice detekčního okna. Detekční okno je výstupem detektorů, ve kterém nalezneme chodce. Vzdálenost chodce budu vypočítat z hloubkových map. Výpočet zkomplikuje skutečnost, že v detekčním okně nejsou jen hloubkové informace o chodci, ale také informace o okolních bodech, které musíme eliminovat při výpočtech.



Obrázek 4.5: Detekční okno po úspěšné detekci (vlevo) a vypočítaná hloubková mapa (vpravo).

Proto je nutné určit, které pixely patří chodcům. Po prostudování více segmentačních algoritmů [8] jsem našel metodu Connected-component labeling. Tato metoda je používána nejme na rozdělení regionů v binárním obraze. Hloubkové, neboli disparitní mapy jsou velmi podobné binárnímu obrazu, protože pixely mají omezenou hodnotu.

Algoritmus prohledá všechny pixely obrazu a zařadí je do komponentů na základě shodnosti sousedních pixelů. Komponenty sdílejí podobné hodnoty intenzity pixelů. Sousedství může být ve čtyřech nebo osmi směrech. Detailní popis jak lze použít ve video sekvencích lze najít v literatuře [9].

# 5 Implementace

Tato kapitola obsahuje implementace jednotlivých metod během tvorby programu, jak byly navrženy v kapitole 4. Vstupem aplikace je snímaný stereo obraz a výsledkem aplikace je informace pro uživatele o detekovaných chodcích a jejich vzdálenosti. V následující kapitole 5.1 jsou popsány programovací prostředky a použité knihovny. Další podkapitola 5.2 obsahuje významné metody a funkce, které byly použity při implementaci v mé práci. Kapitola 5.3 obsahuje zajímavé části uživatelských rozhraní, které jsem použil i při testování.

## 5.1 Programové prostředí

Celá aplikace byla napsána ve vývojovém prostředí Microsoft Visual Studio. Mezi jeho výhody patří, že programování i vytvoření grafických rozhraní můžeme provést na jednom místě. Visual Studio také obsahuje integrovaný debugger a funkci IntelliSense [10], která zobrazí seznam členů nebo poskytne informace o parametrech metod. K vytvoření grafického uživatelského rozhraní jsem použil WinForms Designer, které je spojeno s kódem event-driven programovacím modelem, takže kód se provádí na základě uživatelských akcí. Jako programovací jazyk jsem si zvolil objektově orientovaný jazyk C#. Metody popsané v návrhu jsou implementovány s využitím knihovny Emgu CV [11].

Knihovna EmguCV je vytvořena na základě známé knihovny OpenCV, poskytuje stejné funkce pro práci s obrazem, jen je určena pro vývojové prostředí Visual Studio a podporuje .NET kompatibilní jazyky, jako jsou např. C#, VB, VC++. Knihovna je distribuovaná i pod licencí GNU GPL verze 3. Aplikace byla implementována v rámci 64 bitového operačního systému Microsoft Windows 8.

## 5.2 Použité metody

### Kalibrace kamery

Jak již bylo zmíněno, pro kalibraci kamery budou použity šachovnicové vzory. K tomuto účelu jsem použil následující funkci z knihovny EmguCV:

```
FindChessboardCorners(),
```

která v případě nalezení šachovnicového vzoru vrátí souřadnice všech vnitřních rohů šachovnice. Za vstupem je třeba určit vstupní obraz, který musí být ve stupních šedi, a také počet rohů čtverců po sloupcích a po řádcích v šachovnici. Jakmile jsou kalibrační obrazy zpracovány, můžeme předat souřadnici ke kalibrační funkci. Následující funkce slouží k tomu, abychom zjistili vnitřní a vnější parametry kamery. V EmguCV knihovně máme k dispozici funkci:

```
StereoCalibrate(),
```

kde za vstupem musíme určit 3D body z reálného objektu, v mém případě to budou rozměry jednotlivých čtverců na vzoru a 2D obrazové body získané z předcházejících funkcí. Také musíme určit veli-

kost snímku a počet rohů na šachovnici. Výstupem kalibrace je distorzní vektor, kterým můžeme odstranit radiální a tangenciální zkreslení kamery. Pomocí kalibrace také získáme translační vektor a rotační matici kamery. Translační vektor určuje posunutí pravé kamery vůči levé. Rotační matice určuje relativní pozici kamer.

## Rektifikace

Pomocí rektifikace můžeme dosáhnout toho, aby stereo snímky byly koplanární a sjednotily se zobrazovací roviny kamer, což výrazně urychlí vyhledání korespondujících bodů.

Pro rektifikaci obrazu používám funkce:

```
stereoRectify(),
```

kdy vstupem funkce jsou distorzní, rotační a translační vektory, které jsou získané z kalibrací. Pak funkce vypočítá pro obě kamery zvlášť rotační matice, které zaručují, že korespondující body budou ve stejné výšce, a projekční matici, která určuje nové souřadnice kamery.

Funkce také vypočítá reprojekční matice  $Q$ , které slouží pro zobrazení 2D bodů ve 3D prostoru pomocí disparitu. V mém případě budou hodnoty z těchto matic použity pro výpočet vzdálenosti objektu. Abych nemusel počítat rektifikace pro všechny obrazy ve videosekvencích, mohou být tyto hodnoty reprezentovány ve formě mapy. Tato mapa určuje pro všechny pixely ze vstupních obrazů nové pozice pro rektifikovaný obraz. Pro zjištění mapy používám funkci:

```
initUndistortRectifyMap(),
```

vstupem funkce jsou rotační a projekční matice a výsledkem budou mapy pro x-ové a y-ové osy.

Jakmile jsem zjistil tyto mapy pro všechny obrazy ze vstupu, již můžu provést rektifikace pomocí mapovací funkce. Na tento účel použiji funkce z knihovny:

```
remap(),
```

## Tvorba disparity

Jak již bylo řečeno, pro výpočet disparity musím najít korespondující si body. V EmguCV je pro hledání stereo korespondencí mnoho metod. V rámci testování jsem se rozhodl použít dvě funkce, které jsou implementovány v knihovně. První je funkce StereoSGBM(), která používá metodu semi-global block matching, tedy porovnává po blocích, a nejen jednotlivě, pixely zvlášť. Vstupem funkce musí být rektifikovaný obraz. Ve funkci také můžeme určit nejmenší možnou hodnotu disparity, počet disparit a velikost SAD oken. Podrobnější popis viz OpenCV dokumentace [12].

Druhá funkce, kterou jsem zvolil, je StereoBM(), která používá metodu block-matching. Metoda SGBM sice dává lepší výsledek, ale je výrazně pomalejší. Při zpracování obrazu ve videosekvencích je důležitějším faktorem rychlost výpočtu. Porovnání dvou metod vzhledem k přesnosti a časové náročnosti je popsáno při testování v kapitole 6.

## Výpočet vzdálenosti

V literatuře existuje řada metod, kterými můžeme vypočítat hloubkovou mapu, tedy vzdálenost objektů. V některých případech musíme znát ohniskovou vzdálenost nebo vzdálenost mezi kamerami. Při implementaci jsem se řídil knihou určenou pro práci s knihovnou OpenCV [6], podle které vzdálenost můžeme vypočítat na základě triangulace dle rovnici 4.1. Každou hodnotu jsem vypočítal pomocí

funkcí. Bázi (vzdálenost mezi kamerami) a ohniskovou vzdálenost jsem určil pomocí matice Q získané z rektifikace a disparity pomocí stereo korespondenčních metod.

## Detekce chodce

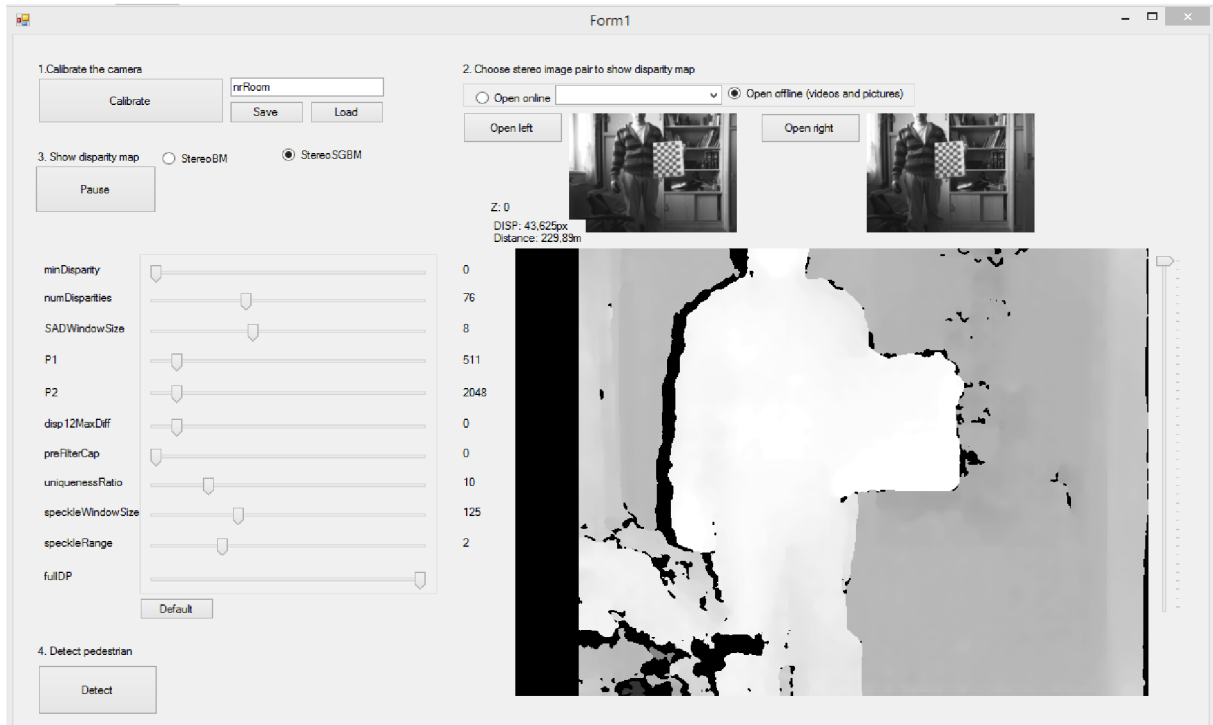
Pro detekování chodců jsem využil již existující metodu, která klasifikuje chodce s využitím HOG deskriptoru. Při implementaci jsem využil strukturu HOGDescriptor z knihovny EmguCV. Struktura také podporuje výpočty vykonané pomocí grafické karty. Knihovna obsahuje implementace pro CUDA architektury, kde v případě nalezení grafické karty používám strukturu GpuHOGDescriptor.

## Segmentace chodce

Abych dosáhl lepšího výsledku při výpočtu vzdálenosti chodce, musím sjednotit disparitní hodnoty v místě detekování. V knihovně existuje implementace pro metodu blob extrakce. Metoda vytvoří regiony v obraze, které se liší v binárních vlastnostech, jako např. jas, barva. Z centrálních regionů pak používám disparitní hodnotu, která se objeví nejvíc krát.

## 5.3 Testovací aplikace

Testovací aplikaci lze rozdělit na 3 části, každá část odpovídá jednomu WinForms oknu, které je implementováno pomocí Visual Studia. Po spuštění programu, aplikace otevře jeho hlavní část.

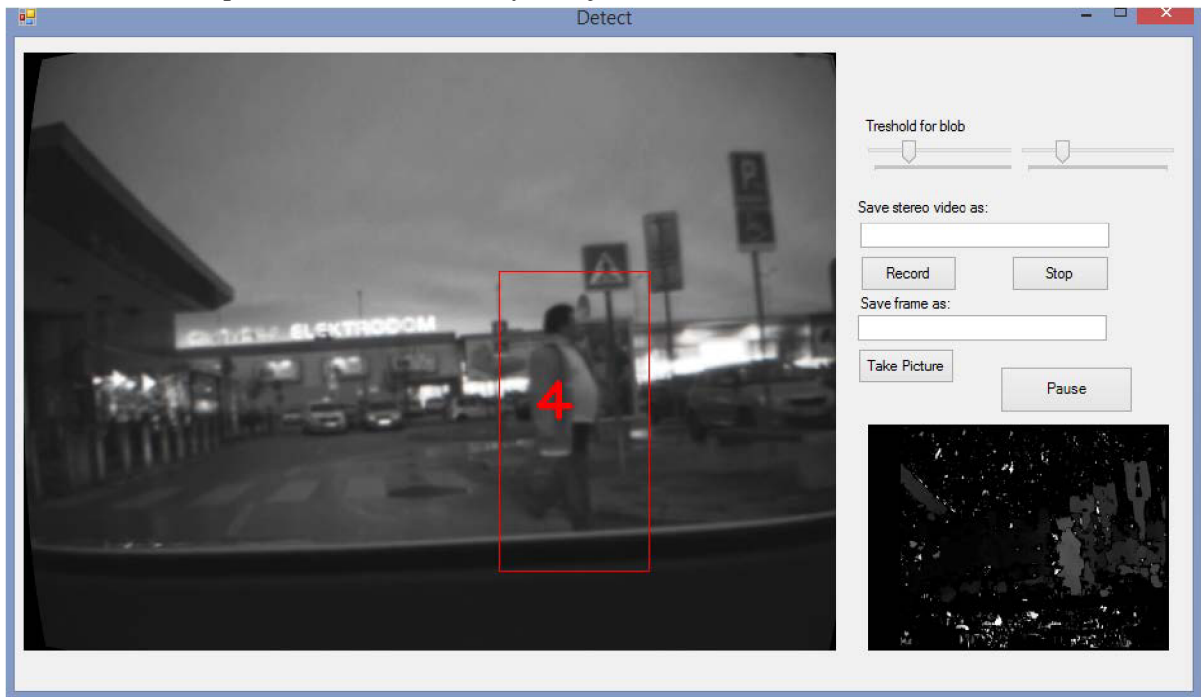


Obrázek 5.1: Ukázka grafického rozhraní hlavního části programu. V pravé horné části vidíme stereo obrazy, pod nimi vidíme disparitní mapu.

Jak lze na obrázku 5.1 vidět, uživatel v první fázi musí kalibrovat stereo kamery stisknutím tlačítka „Calibrate“, po kterém se otevře kalibrační okno nebo je možno vybrat z uložených kalibračních vzorů.



V druhé fázi pro pořízení stereo snímků se otevřou obrazy z dostupných video zařízení nebo se otevře video soubor (nebo obrázek). V třetí fázi se pak volí z dostupných stereo korespondenčních metod. V případě metody SGBM lze nastavit parametry funkce. Po stisknutí tlačítka „Detect“ se otevře nové okno, které slouží pro zobrazení detekovaných objektů.



Obrázek 5.2: Po detekování chodce se zobrazí jeho vzdálenost, která je vypočítaná z disparitních map (menší obrázek vlevo). V rozhraní lze také uložit aktuální snímek nebo nahrát video.

## 6 Testování a hodnocení

Tato kapitola se zabývá testováním systému a vyhodnocováním prováděných experimentů. V rámci testování jsem použil vlastní datové sady. V kapitole 6.1 je popsána návrh testování a použité datové sady, v kapitole 6.2 jsou popsány prováděné experimenty a shrnuty dosažené výsledky. Kapitola 6.3 pak obsahuje diskuzi dosažených výsledků a vlastní poznatky k testováním.

### 6.1 Návrh testování

Funkčnost a přesnost navrhované a implementované metody byly ověřeny pomocí vlastních datových sad. Jak bylo řečeno, systém budu používat v autě pro detekci chodců a vypočítání jejich vzdálenosti. V rámci testování jsem se snažil zjistit, do jaké míry lze používat systém jako celek v reálném prostředí. Podle toho jsem se stanovil tři druhy testu, ve kterých jsem se zaměřil na testování přesnosti odhadnuté vzdálenosti a také přesnost detektorů.

První test sloužil pro odhadnutí nejlepšího možného výsledku při měření vzdálenosti. Abych mohl získat přesné výpočty, musel jsem určit, jaké bude nevhodnější nastavení k vzhledem výpočtů, a také prostor pro měření. Vybral jsem prostor v interiéru, kterým je chodba na kolejích. Zde jsem mohl eliminovat ostré stíny, a díky umělému osvětlení také přeexponované a podexponované snímky.

V druhém testu jsem již zvýšil náročnost na výpočet vzdálenosti. Zvolil jsem prostor v exteriéru, kde intenzita přímého slunečního záření během testování stoupala a klesala kvůli stínu. Související homogenní plochy také zvýšily náročnost, protože dílčím krokem hloubkových map je nalezení korespondenčních bodů, přičemž korespondenční metody jsou úspěšné s texturovaným obrazem. Testovací prostředí se již podobá reálným oblastem použití, tedy zkoumal jsem i úspěšnost HOG deskriptorů.

V posledním testu jsem testoval systém přímo v autě v silničním provozu. Jezdil jsem v centru Brna, kde zaměřil jsem se na detekování chodce. Po testování v reálném prostředí jsem dostal jasný pohled, jak lze použít mého systému v skutečnosti.

### Popis testovacích dat

Jak již bylo řečeno, pro pořízení stereo snímku jsem používal dvě stejné USB 3.0 kamery vypůjčené z ústavy 3D grafiky. Při tvorbě snímku jsem používal rozlišení 640x480 pixelů a 45 snímku za sekundu s fixní ohniskovou vzdáleností 5mm. Ohniskovou vzdálenost jsem měl vždy na nejnižší hodnotě, protože kamera poskytovala jen manuální nastavení a v jiné vzdálenosti jsem nemohl zajistit totožné hodnoty. Zvýšená snímací frekvence sloužila k tomu, aby objekty v pohybu nebyly rozmazané v obraze.



Obrázek 6.1: Použitý kamerový systém namontovaný na stativ.

Jak lze na obrázku 6.1 vidět, při vytvoření testovacích dat jsou kamery umístěny vedle sebe a posunuty jen v horizontální ose. Bázi, tedy vzdálenost mezi kamerami, jsem určil tak, abych dostal určitý rozsah, ve kterém lze použít disparitní mapu pro odhad vzdálenosti. Jelikož budu používat systém v autě, chtěl jsem dosáhnout minimální vzdálenosti od 1,5 do 2 metrů. Této hodnoty jsem dosáhl při umístění kamer ve vzdálenosti 10 až 15 centimetrů od sebe. Při takové bázi teoreticky, podle rovnice, jsem mohl dosáhnout maximální vzdálenosti, která činí 70 až 135 metrů (maximální vzdálenost jsem určil ze situace, kdy disparita pro daný bod bude právě jeden pixel). V skutečnosti ale s takovou malou hodnotou nelze pracovat, protože není možno dosáhnout ideálního nastavení. Když disparita namísto jedné bude dvě, pak ve výpočtech vzdálenosti se budu mýlit až o 60 metrů.

Vzhledem k výše uvedeným důvodům budu tvořit testovací sady v prvním a druhém testu tak, že kamery budou umístěny na stativ do stabilní polohy a objekt bude umístěn před kamerou od 1,5 do 15 až 20 metrů. Po metrech budu ukládat pořízené stereo snímky a následně se vyhodnotí získaná data.

## 6.2 Popis prováděných experimentů

### Test 1 – Interiér

Tento test sloužil k tomu, abych zjistil po správné kalibraci kamery orientační přesnost vypočtené disparity, vzdálenost a také otestování rozpoznávání lidí.

Jako první je nutné kalibrovat stereo kamery, pro kalibraci jsem vytvořil 50 párů snímků s šachovnicovým vzorem.



Obrázek 6.2: Funkce úspěšně rozpozná vnitřní rohy šachovnice. Rozměry jednotlivých čtverců je 38x38mm.

Po kalibraci získáme přesný model kamery, včetně jejích vnitřních a vnějších parametrů. Po rektifikaci obrazů ukládám mapu, která slouží k mapování nekalibrovaného obrazu do rektifikovaného tvaru. Po rektifikaci také získám reprojekční matici  $Q$ , z které určím vzdálenost mezi kamerami, a ohniskovou vzdálenost v pixelech:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 325,19 \\ 0 & 1 & 0 & 283,56 \\ 0 & 0 & 0 & 654,55 \\ 0 & 0 & 0,0079 & -0,002813 \end{bmatrix} \quad 6.1$$

Pro ověření hodnoty jsem přepočítal ohniskovou vzdálenost získanou z rovnic na milimetrech podle rovnici:

$$f = \frac{f_p * width_{CMOS\ sensor}}{width_{image}}, \quad 6.2$$

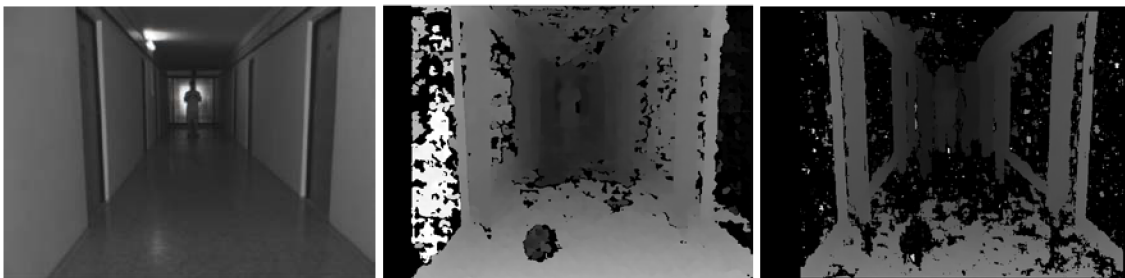
kde šířka snímacího čipu je 4.8mm a šířka obrazu je 640px. Po dosažení hodnoty do rovnic jsem zjistil, že podle kalibrace je ohnisková vzdálenost 4,909mm. Kalibraci lze považovat za úspěšnou, protože ve skutečnosti jsem měl ohniskovou vzdálenost 5mm (chybovost je menší než 2%).

Pro vytvoření disparitních map jsem použil algoritmy Semi-Global Block Matching a Block Matching



Obrázek 6.3: Z rektifikovaných obrazů (levý) je vypočítána disparitní mapa podle algoritmu Semi-Global Block Matching (střední) a Block Matching (pravý), vzdálenost objektů je 2 m.

Na kolekci obrázků 6.3 lze vidět, že algoritmus SGMB najde mnohem víc korespondujících si bodů, protože porovná bloky v osmi směrech. Oba algoritmy našly hranici objektů při nižší vzdálenosti.



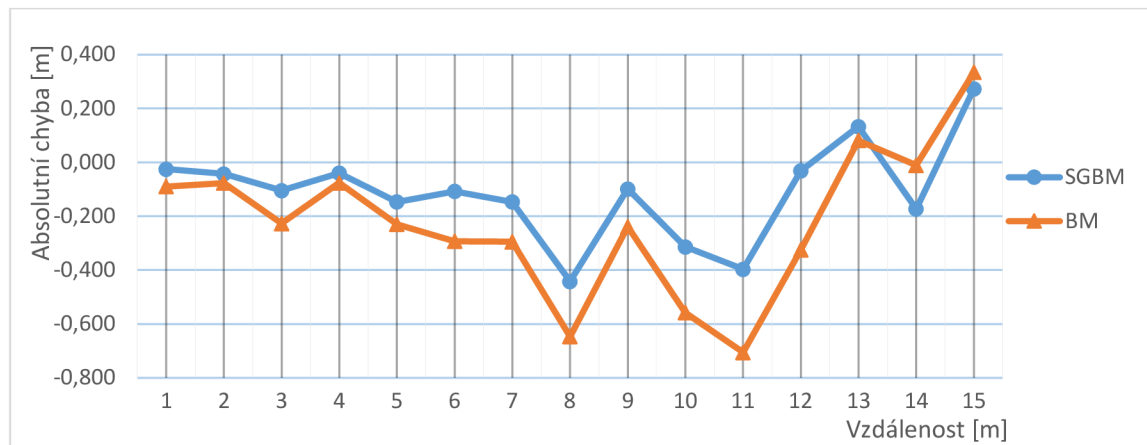
Obrázek 6.4: Stejně algoritmy používané jako na obr. 6.3, vzdálenost objektů je 12 m.

Při vyšších vzdálenostech oba algoritmy těžko rozpoznali hranici objektů, avšak algoritmus SGBM opět dal lepší výsledek.

Měření probíhalo tak, že jsem stál před kamerou ve vzdálenosti od 2 do 15 metrů a ukládal jsem obrazy. Pro výpočet disparitních map jsem vzal kolekci bodů, o kterých jsem věděl, že jsou umístěny ve stejné vzdálenosti od kamery a vypočítal jsem jejich vzdálenost podle návrhu v kapitole 4.5. Výsledky měření byly porovnány se skutečnými vzdálenostmi, čímž jsem mohl orientačně stanovit přesnost mé metody. V tabulce 6.1 jsou popsány dosažené výsledky:

Skutečný vzdálenost [m]	Vzdálenost s metdou SGBM [m]	Vzdálenost s metodou BM [m]	Absolutní chyba ( $\Delta_x$ ) u metody SGBM [m]	Absolutní chyba ( $\Delta_x$ ) u metody BM [m]	Relativní chyba ( $\delta_x$ ) u metody SGBM [%]	Relativní chyba ( $\delta_x$ ) u metody BM [%]
1,5	1,475	1,435	-0,025	-0,065	-1,695	-4,530
2	1,957	1,966	-0,043	-0,034	-2,197	-1,729
3	2,895	2,878	-0,105	-0,122	-3,627	-4,239
4	3,96	3,964	-0,040	-0,036	-1,010	-0,908
5	4,853	4,917	-0,147	-0,083	-3,029	-1,688
6	5,892	5,815	-0,108	-0,185	-1,833	-3,181
7	6,853	6,852	-0,147	-0,148	-2,145	-2,160
8	7,558	7,795	-0,442	-0,205	-5,848	-2,630
9	8,901	8,86	-0,099	-0,140	-1,112	-1,580
10	9,686	9,756	-0,314	-0,244	-3,242	-2,501
11	10,603	10,691	-0,397	-0,309	-3,744	-2,890
12	11,968	11,707	-0,032	-0,293	-0,267	-2,503
13	13,133	12,948	0,133	-0,052	1,013	-0,402
14	13,827	14,162	-0,173	0,162	-1,251	1,144
15	15,273	15,061	0,273	0,061	1,787	0,405

Tabulka 6.1: Porovnání vzdálenosti se skutečnými vzdálenostmi při bázi  $B=126\text{mm}$  a ohniskové vzdálenosti  $f=654,55\text{px}$ .



Obrázek 6.5: Absolutní chyba u jednotlivých metod.

### Shrnutí experimentů:

Z tabulky 6.1 lze vidět, že obě metody dají stejný výsledek. Relativní chyba se vzdáleností nezvýšila, stala se konstantní. Můžeme říci, že do 15 metrů obě metody fungují úspěšně s průměrnou chybou 1,9 %. Obrázek 6.5 ukazuje, že absolutní chyba u obou metod se vzdáleností zvýšila s větším rozptylem.

V chodbě jsem také otestoval, jak funguje detekce na video záběrech. Vytvořil jsem záznam toho, jak jsem prošel na chodbě s metrovým krokem a pomocí HOG deskriptorů jsem detekoval „chodce“ a z disparitních map jsem vypočítal vzdálenost v detekčním okně. Jak lze na obrázku 6.5 vidět, pro vzdálené objekty se výpočet vzdálenosti již zhoršuje.



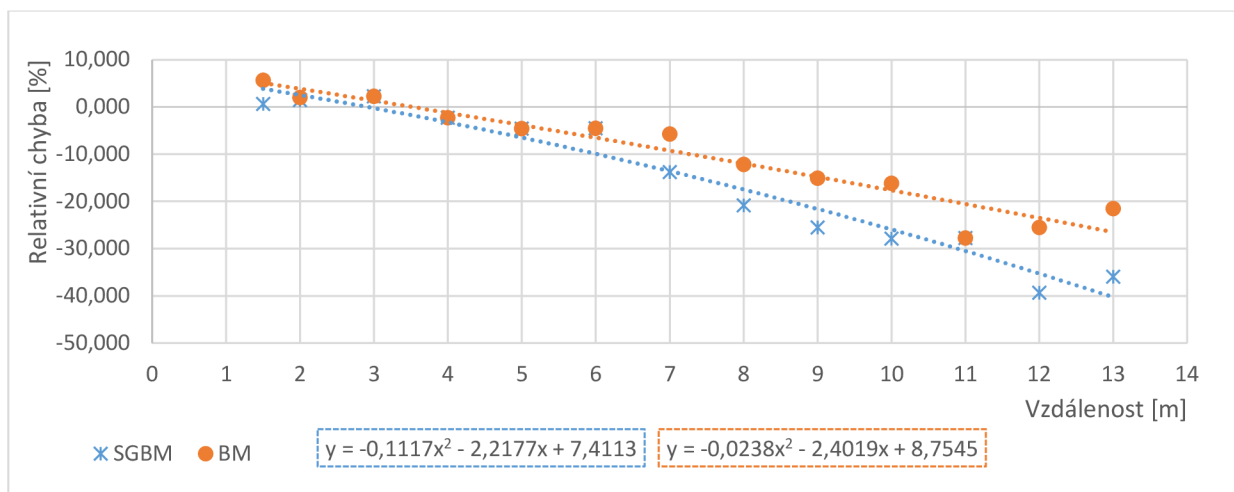
Obrázek 6.6: Po detekci je zobrazeno detekční okno a uvnitř okna pro spojené komponenty z disparitních map je vypočítaná vzdálenost: 3 m (vlevo), 5m (na střed) a 10 m (vpravo).

## Test 2 – Exteriér

V druhém z testů jsem dodržel stejné postupy jako v předcházejícím testu. Opět jsem kalibroval kameru a vytvořil jsem video záznam toho, jak procházím před kamerou s metrovým krokem od 5 do 20 metrů. V tomto testu jsem již vypočítal vzdálenost osoby po její detekci. V tabulce 6.2 jsou dosažené výsledky:

Skutečný vzdálenost [m]	Vzdálenost s metodou SGBM [m]	Vzdálenost s metodou BM [m]	Absolutní chyba ( $\Delta_x$ ) u metody SGBM [m]	Absolutní chyba ( $\Delta_x$ ) u metody BM [m]	Relativní chyba ( $\delta_x$ ) u metody SGBM [%]	Relativní chyba ( $\delta_x$ ) u metody BM [%]
1,5	1,51	1,59	0,010	0,090	0,662	5,660
2	2,03	2,04	0,030	0,040	1,478	1,961
3	3,07	3,07	0,070	0,070	2,280	2,280
4	3,91	3,91	-0,090	-0,090	-2,302	-2,302
5	4,78	4,78	-0,220	-0,220	-4,603	-4,603
6	5,74	5,74	-0,260	-0,260	-4,530	-4,530
7	6,15	6,62	-0,850	-0,380	-13,821	-5,740
8	6,62	7,13	-1,380	-0,870	-20,846	-12,202
9	7,17	7,82	-1,830	-1,180	-25,523	-15,090
10	7,82	8,61	-2,180	-1,390	-27,877	-16,144
11	8,61	8,61	-2,390	-2,390	-27,758	-27,758
12	8,61	9,56	-3,390	-2,440	-39,373	-25,523
13	9,56	10,7	-3,440	-2,300	-35,983	-21,495

Tabulka 6.2: Test v exteriéru. Hodnoty získané z detekčního okna při bázi  $B=124\text{mm}$  a ohniskovou vzdálenosti  $f=693\text{px}$ .



Obrázek 6.7: Relativní chyba u jednotlivých metod.

Jak lze v tabulce 6.2 vidět, absolutní chyba se vzdáleností již výšila, kde metoda BM dosáhl lepší výsledku. Obrázek 6.7 ukazuje, že chybovost se vzdáleností zvýší a je téměř lineární.

Metody ve vyšších vzdálenostech (nad 14 metrů) mají již špatnou rozlišovací schopnost a těžko rozpoznají hrany menších objektů, jak to lze vidět na obrázku 6.8:



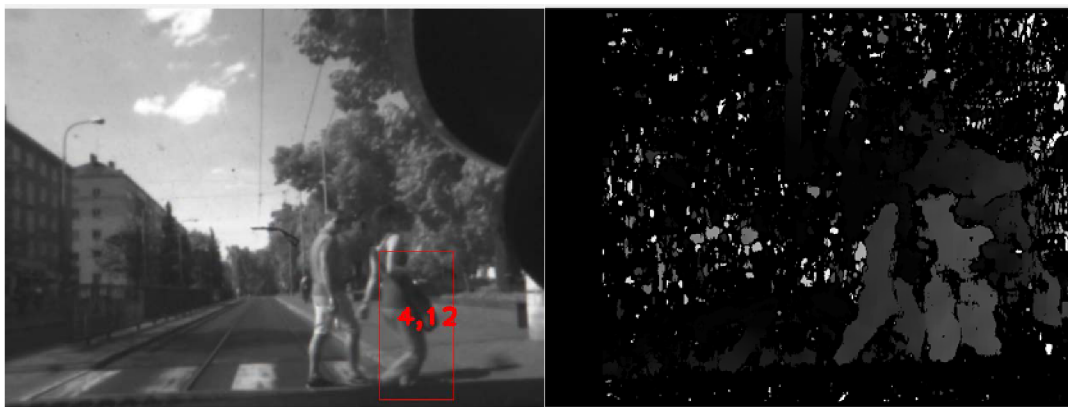
Obrázek 6.8: Detekční osoba je 14 metrů vzdálená od kamery. Na levém obrazu je po detekci zobrazena její vzdálenost vypočítaná z disparitních map s metodou SGBM (vpravo).

### Shrnutí experimentů:

Venku, kde počasí již ovlivňuje měření, jsem těžko dosahoval ideálního nastavení kamery. Při porovnání dvou metod jsem zjistil, že v reálném prostředí, kde se často opakují vzory na obrazech a struktura scény je slabší, je vhodnější použít metodu Block Matching, protože lépe vypočítá disparity. Oproti tomu metoda Semi-Global Block Matching zachytí více detailů ve slabě texturované oblasti. Osoby, které jsou vzdálené více než 14 metrů a je při jejich detekci použito rozlišení 640x480, již nelze detekovat s HOG deskriptorem.

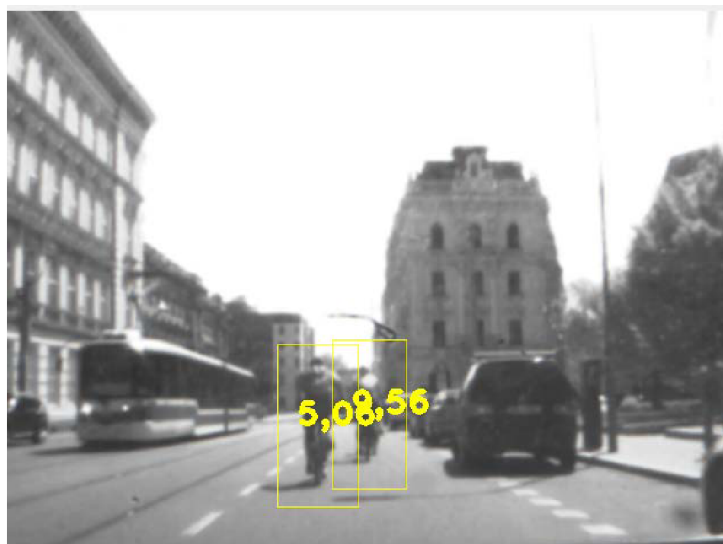
## Test 2 – Reálné prostředí

V tomto testu jsem již neměřil žádné hodnoty, ale otestoval jsem, jak lze použít mého systému v reálném čase a prostředí. Testovací data jsem vytvořil z auta a sledoval jsem, v jakém míře systém detekuje chodce a vypočítá jejich vzdálenosti.



Obrázek 6.9: Testování systémů v reálném prostředí s metodou BM.

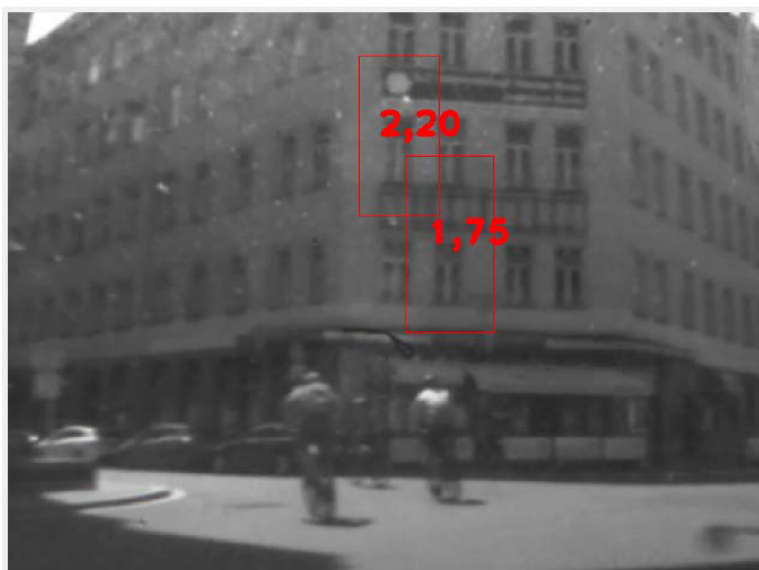
Jak lze vidět na obrázcích 6.9, detekce byla úspěšná, a z disparitních map šlo i poměrně dobře vypočítat vzdálenost chodců. HOG deskriptor umožňuje i vícenásobné detekování v jednom obraze. Z experimentů jsem také zjistil, že detektor je schopný detekovat i cyklisty.



Obrázek 6.10: Detekce cyklistů s vícenásobným detekováním.

Během testování jsem se také setkal s případem, kde se objevily oba typy chyb. Chodci ani cyklisti nebyli detekováni navzdory tomu, že se na obrázcích objevili. Tento jev se nazývá „false negative“. V druhém případě byla detekována falešná místa v obraze, která se podobají textuře nebo jinou formou, jsou to tzv. „false positive“ detekce.





Obrázek 6.11: V obrazu lze vidět „false negative“ a „false positive“ detekce.

## 6.3 Diskuze

V testovací části se ukázalo, že metody pro výpočet hloubkových map ze stereo obrazu dokáží velice dobře rekonstruovat hloubku, to však v případě, kdy scéna byla řádně strukturovaná. Nejlepšího výsledku v rámci tvorby disparitní mapy jsem dosáhl v prvním testu, průměrná chyba měření do 15 metrů byla méně než 2% u obou metod. Z druhého testu je již vidno, že chyba se lineárně zvětšuje se vzdáleností. Je to dáno tím, že i po několika pokusech jsem těžko dosáhl ideálního nastavení a kalibrace kamery. Během testování jsem zjistil několik faktorů, které zhoršují měření. Stereo kamera neposkytuje automatické nastavení expozice. Musel jsem často během pořizování video záznamu znovu exponovat obrazy. Výsledkem byla slabší disparitní mapa, neboli v horším případě nereálné hodnoty naměřených vzdáleností.

Detekce pracuje až do 15 metrů při rozlišení 640x480 pixelů, avšak použití manuálního nastavení zaostření bylo problémem. Kamery mají nižší hloubku ostrosti, a těžko jsem uměl zaostřit na scénu, což ovlivnilo i efektivitu detektoru. I za takových okolností lze konstatovat, že detekce fungovala výborně.

V rámci testování jsem použil notebook Lenovo ThinkPad Edge E531 s procesorem Intel Core i5-3230M bez dedikované grafické karty. Doba trvání výpočtu disparitní mapy s metodou BM se pohybovala okolo 40ms, s metodou SGMB 116ms výpočet detekce trval průměrně 560ms. Celkový čas pro odhad vzdálenosti chodce je 600ms, který k vzhledem využití tohoto systému je podprůměrný.

# Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat aplikaci pro odhad vzdálenosti objektů od vozidla s využitím hloubkové mapy získané ze stereo kamery. Výsledkem práce je experimentální řešení asistenčních systémů pro vozidla, které po detekování chodce vypočítá jeho vzdálenost a upozorní řidiče.

Teoretická část práce obsahuje stručný přehled o stereovidění a 3D rekonstrukci, a také obsahuje informace o asistenčních systémech používaných v dnešních automobilech. V návrhu jsem inspiroval z již existujících metod, ze kterých jsem se pokusil vybrat ty nejvhodnější dle využití takového systému. Především jsem považoval za důležité např. pořádnou kalibraci kamery, výběr vhodných stereo korespondenčních metod a detektorů. V rámci této práce jsem použil metodu SGBM (Semi-Global Block Matching) a BM (Block Matching) pro vytvoření disparitní mapy. Hloubkovou mapu, neboli vzdálenost v prostoru, jsem odhadnul na základě triangulace s využitím disparitní mapy. Pro detekování chodce jsem použil HOG deskriptor. Vzdálenost chodce jsem vypočítal z hloubkové mapy, kde v místě detekování jsem použil jen potřebné hodnoty určené metodou Connected Component Labeling. Systém jsem úspěšně implementoval podle návrhu s využitím knihovny EmguCV ve vývojovém prostředí Microsoft Visual Studio.

Práce také obsahuje vytvoření testovacího datasetu, který sloužil pro odhadnutí přesnosti systému. Při experimentování s řešením byla stanovena jeho orientační přesnost. V prvním testu se ukázalo, že po nalezení dostatečného množství stereo korespondencí a po úspěšné kalibraci lze docela přesně vypočítat hloubkovou mapu. Přesnost výpočtu vzdálenosti u obou metod se pohybovala okolo 2%. V druhém testu jsem dosáhl horšího výsledku s přesností 20%, kde scéna neměla vhodnou strukturu. Výsledky jsou silně ovlivněny použitím chybně nastavené kamery. V rámci testování v autě jsem zjistil, že pokud zaostření nebo expozice nejsou stejné u obou kamer, pak stereo korespondenční metody vrací nereálné hodnoty. Také bylo prokázáno, že HOG detektor rozpoznal chodce téměř z každé pozice do 15 metrů.

Aplikace ovšem není dokonalá, v budoucnu bych chtěl znovu vytvořit záznamy tak, aby kamery nedisponovaly žádnými negativními faktory vzhledem ke kvalitě výpočtu. Dále bych se zaměřil na optimalizaci, doba výpočtu disparitní mapy s metodou BM se pohybovala okolo 40ms a u detektoru kolem 560ms. V knihovně EmguCV HOG detektor podporuje CUDA architektury, s využitím grafické karty typu nVidia GT 555M v rámci testování jsem dosáhl hodnoty až 200ms, takže je silně doporučeno její použití. Implementoval bych i ostatní metody v OpenCL jazyce, abych dostal akceptovatelnou výpočetní dobu, protože při hodnotě 600ms to zatím není plnohodnotný systém.

# Literatura

- [1] HAVLÍK, Karel. 2005. *Psychologie pro řidiče: zásady chování za volantem a prevence dopravní nehodovosti*. Vyd. 1. Praha: Portál, 223 s. ISBN 80-717-8542-3.
- [2] BEZPEČNOST SUBARU: Bezpečnost před kolizí. 2011. *Subaru Česká Republika* [online]. [cit. 2015-05-13]. Dostupné z: <http://www.subaru.cz/bezpecnost-pred-kolizi.html>
- [3] Webová kamera. 2010. *Wikipedie: Otevřená encyklopedie* [online]. [cit. 2015-05-13]. Dostupné z: [http://cs.wikipedia.org/wiki/Webov%C3%A1\\_kamera](http://cs.wikipedia.org/wiki/Webov%C3%A1_kamera)
- [4] See3CAM\_10CUG. *E-con Systems: Your Product Development Partner* [online]. [cit. 2015-05-13]. Dostupné z: <http://www.e-consystems.com/IMP-USB3-Globalshutter-Camera.asp>
- [5] PERIS, Martin. 2011. OpenCV: Stereo Camera Calibration. *Blog.martinperis.com* [online]. [cit. 2015-05-13]. Dostupné z: <http://blog.martinperis.com/2011/01/opencv-stereo-camera-calibration.html>
- [6] 2008. BRADSKI, Gary R. *Learning OpenCV*. Sebastopol: O'Reilly. ISBN 978-0-596-51613-0.
- [7] DALAL, N. a B. TRIGGS. 2005. Histograms of Oriented Gradients for Human Detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*[online]. IEEE, s. 886-893 [cit. 2015-05-14]. DOI: 10.1109/CVPR.2005.177. ISBN 0-7695-2372-2. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1467360>
- [8] BC. NOVOTNÝ, Stanislav. 2012. *Detekce mobilního robotu zpracováním obrazu*. Brno. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Jiří Krejsa, Ph.D.
- [9] 2010. WALCZYK, Robert a Armitage ALISTAIR. *Comparative study on connected component labeling algorithms for embedded video processing systems* [online]. Las Vegas: CSREA Press [cit. 2015-05-15]. ISBN 1-60132-153-8. Dostupné z: <http://researchrepository.napier.ac.uk/3901/>
- [10] IntelliSense. 2013. *Microsoft: Developer Network* [online]. [cit. 2015-05-15]. Dostupné z: <https://msdn.microsoft.com/cs-cz/library/hcw1s69b.aspx>
- [11] *EmguCV* [online]. [cit. 2015-05-15]. Dostupné z: [http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page)
- [12] Camera Calibration and 3D Reconstruction. *OpenCV documentation* [online]. [cit. 2015-05-15]. Dostupné z: [http://docs.opencv.org/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html)
- [13] *Základní parametry kamer* [online]. [cit. 2015-05-16]. Dostupné z: <http://www.cctv-kamerove-systemy.cz/zakladni-parametry-kamer/>
- [14] *A C++ code to compute OpenGL 4x4 GL\_MODELVIEW\_MATRIX from 2D-3D points homography: Homography and pinhole camera model* [online]. [cit. 2015-05-16]. Dostupné z: [https://brain-trekking.wordpress.com/2013/06/02/a-c-code-to-compute-opengl-4x4-gl\\_modelview\\_matrix-from-2d-3d-points-homography/](https://brain-trekking.wordpress.com/2013/06/02/a-c-code-to-compute-opengl-4x4-gl_modelview_matrix-from-2d-3d-points-homography/)

- [15] BREDNA, Martin. Homografie a epipolární geometrie. *Tribolit* [online]. Univerzita Tomáše Bati ve Zlíně Fakulta aplikované informatiky, 2010 [cit. 2015-05-17]. Dostupné z: <http://trilobit.fai.utb.cz/homografie-a-epipolarni-geometrie>
- [16] BUREŠ, Lukáš. Plzeň, 2011. Dostupné také z: [http://home.zcu.cz/~lbures/Bachelor\\_Thesis/Bachelor\\_Thesis\\_Lukas\\_Bures\\_CZ.pdf](http://home.zcu.cz/~lbures/Bachelor_Thesis/Bachelor_Thesis_Lukas_Bures_CZ.pdf). Bakalářská práce.
- [17] VAVROŠ, Ondřej. *Meření vzdálenosti stereoskopickým senzorem*. Brno, 2014. Diplomová práce.
- [18] KRUPÍČKA, Radim. *Měření a analýza pohybu ruky u parkinsoniků*. Praha, 2013. Disertační práce.
- [19] BABINEC, Adam. *Tvorba hloubkové mapy pomocí stereovidění pro navigaci mobilního robota*. Brno, 2012. Bakalářská práce.
- [20] YONG-SHENG CHEN, YI-PING HUNG a CHIOU-SHANN FUH. Fast block matching algorithm based on the winner-update strategy. *IEEE Transactions on Image Processing* [online]. **10**(8): 1212-1222 [cit. 2015-05-17]. DOI: 10.1109/83.935037. ISSN 10577149. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=935037>
- [21] HIRSCHMULLER, H. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 2008, **30**(2): 328-341 [cit. 2015-05-17]. DOI: 10.1109/TPAMI.2007.1166. ISSN 0162-8828. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4359315>
- [22] ČERNÍK, Ján. *VIZUÁLNÍ DETEKCE OSOB V KOMERČNÍCH APLIKACÍCH*. Brno, 2012. Diplomová práce.
- [23] VLK, František. *Automobilová elektronika 1: asistenční a informační systémy*. 1. vyd. Brno: Prof.Ing.František Vlk, DrSc, 2006, 269 s. ISBN 80-239-6462-3.
- [24] BOSCH: *Pressforum* [online]. [cit. 2015-05-17]. Dostupné z: <http://press.bosch.cz/>
- [25] HARRIS, Tom. How Radar Detectors Work. *How Stuff Works* [online]. 2006 [cit. 2015-05-17]. Dostupné z: <http://auto.howstuffworks.com/radar-detector1.htm>
- [26] Kameru využívající asistenti. *Škoda* [online]. [cit. 2015-05-18]. Dostupné z: <http://www.skoda-auto.cz/models/nova-octavia/asistencni-systemy/>
- [27] Subaru models with EyeSight® get the highest possible score in IIHS front crash prevention tests. *Subaru* [online]. 2015 [cit. 2015-05-18]. Dostupné z: <http://www.subaru.com/engineering/eyesight.html>
- [28] SCHREIBER, Ronnie. Subaru Uses Stereo 3D Tech in New EyeSight™ ADAS. *The truth about cars* [online]. 2012 [cit. 2015-05-18]. Dostupné z: <http://www.thetruthaboutcars.com/2012/03/subaru-uses-stereo-3d-tech-in-new-eyesight-adidas/>

# Seznam použitých zkratek

<b>ABS</b>	Anti-lock Brake System
<b>ACC</b>	Adaptive Cruise Control
<b>APS</b>	Automated Parking System
<b>ASR</b>	Anti-Slip Regulation
<b>BM</b>	Block Matching
<b>C#</b>	je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft
<b>CCD</b>	Charge Coupled Device
<b>CMOS</b>	Complementary Metal Oxide Semiconductor
<b>CUDA</b>	Compute Unified Device Architecture
<b>EmguCV</b>	open source knihovna na zpracování obrazu, určena pro vývojové prostředí Visual Studio a podporuje .NET jazyky
<b>ESP</b>	Electronic Stability Program
<b>GNU GPL v3</b>	GNU General Public License je licence pro svobodný software
<b>HOG</b>	Histogram of Oriented Gradients
<b>HUD</b>	Head-Up Display
<b>LDW</b>	Lane Departure Warning
<b>OpenCL</b>	Open Compute Language
<b>SAD</b>	Sum of Absolute Difference
<b>SGBM</b>	Semi-Global Block Matching
<b>USB</b>	Universal Serial Bus
<b>Visual Studio</b>	vývojové prostředí od Microsoftu

# A DVD

Příložený kompaktní disk obsahuje:

- složka **BP** – elektronická verze této práce ve formátu PDF a DOC.
  
- složka **application** – obsahuje soubor .sln pro otevření programu v programovacím prostředí Microsoft Visual Studio.
  - složka **BC\_StereoCalib** – zdrojové kódy aplikace napsané v jazyce C#.
  - složka **BC\_StereoCalib/bin/Debug** – spustitelná aplikace doplněná o potřebné dynamické knihovny.
  
- složka **demo** – demonstrační videa a obrázky.
  
- složka **library** – potřebné knihovny k aplikaci.
  
- složka **dataSet** – testovací sada a kalibrační matice kamery.
  
- **README.txt** – zde jsou popsány jak přidat knihovny a přeložit aplikace a návod k použití aplikace.
  
- **plakat.pdf**

# B Plakát

## VÝPOČET HLOUBKOVÉ MAPY ZE STEREO KAMERY



*Autor:* Daniel Berek (berekdaniel@gmail.com)

*Vedoucí:* Ing. Michal Španěl, Ph.D.

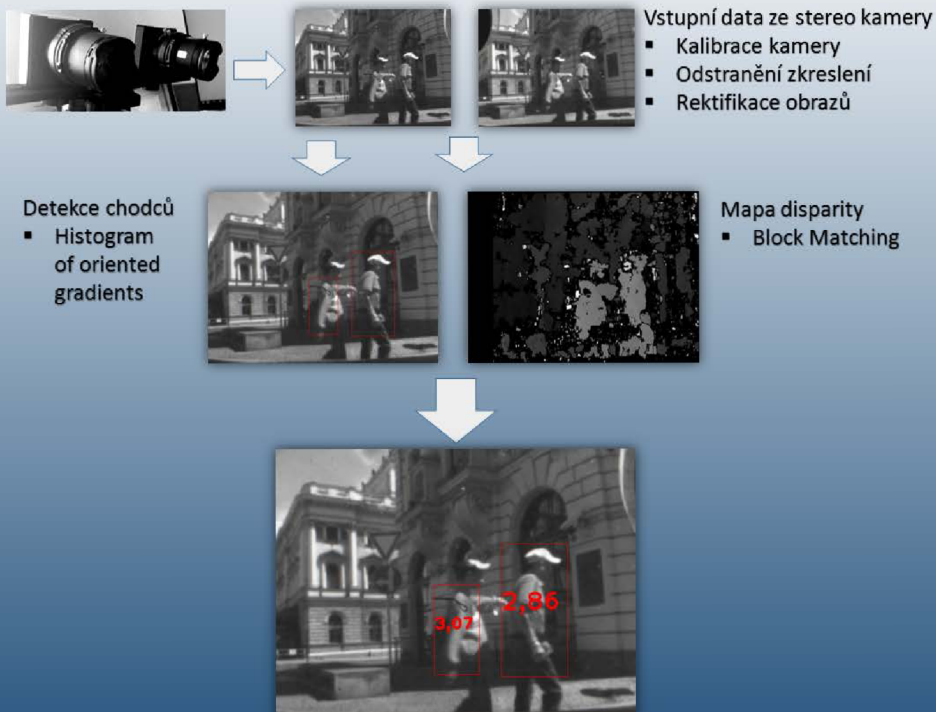
### Cíl práce

Monitorovací systém do auta

- Detekce chodců
- Odhad jejich vzdálenosti



### Výsledek



Fakulta informačních technologií, VUT Brno, 2015