



**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

# **OPTICKÁ LOKALIZACE VELMI VZDÁLENÝCH CÍLŮ VE VÍCEKAMEROVÉM SYSTÉMU**

OPTICAL LOCALIZATION OF VERY DISTANT TARGETS IN MULTICAMERA SYSTEMS

**SEMESTRÁLNÍ PROJEKT**

TERM PROJECT

**AUTOR PRÁCE**

AUTHOR

**Bc. JAN BEDNAŘÍK**

**VEDOUcí PRÁCE**

SUPERVISOR

**doc. Ing. ADAM HEROUT, Ph.D.**

BRNO 2015

## **Abstrakt**

Výtah (abstrakt) práce v českém jazyce.

## **Abstract**

Výtah (abstrakt) práce v anglickém jazyce.

## **Klíčová slova**

Klíčová slova v českém jazyce.

## **Keywords**

Klíčová slova v anglickém jazyce.

## **Citace**

Jan Bednařík: Optical Localization of Very Distant Targets in Multicamera Systems, semestrální projekt, Brno, FIT VUT v Brně, 2015

# Optical Localization of Very Distant Targets in Multicamera Systems

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Adama Herouta, Phd. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Bednařík  
January 3, 2016

© Jan Bednařík, 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.*

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Related work</b>	<b>3</b>
<b>3</b>	<b>System overview</b>	<b>4</b>
<b>4</b>	<b>Camera unit</b>	<b>5</b>
<b>5</b>	<b>Sensitivity analysis</b>	<b>6</b>
<b>6</b>	<b>Stationing and rectification</b>	<b>7</b>
6.1	Stationing . . . . .	7
6.2	Rectification . . . . .	8
<b>7</b>	<b>Detection and tracking</b>	<b>13</b>
<b>8</b>	<b>Cooperation among camera units</b>	<b>14</b>
<b>9</b>	<b>Target localization</b>	<b>15</b>
<b>10</b>	<b>Implementation</b>	<b>16</b>
10.1	Robot Operating System . . . . .	16
10.2	System architecture . . . . .	18
10.3	Application of Gazebo . . . . .	18
10.4	External libraries . . . . .	18
10.5	Design of GUI . . . . .	18
<b>11</b>	<b>Experiments and results</b>	<b>19</b>
<b>12</b>	<b>Conclusion</b>	<b>20</b>

# Chapter 1

## Introduction

- napad: rozdelit uvod do pokapitol (podle <https://uu.diva-portal.org/smash/get/diva2:648760/FULLTEXT01>)
- background - related work - problem formulation - system overview - thesis organization
  - Slouží k zasazení řešené problematiky do širšího kontextu - o optických dálkomerech -  
vyhody systému (pasivní radar) - využití
  - v podobě stručného obsahu jednotlivých kapitol definuje strukturu písemné práce.
  - další kapitoly –  
formulaci cíle práce, charakteristiku současného stavu řešené problematiky a teoretická  
a odborná východiska řešených problémů.

## Chapter 2

### Related work

- srovnani s ji existujicimi aktivnimi (pasivnimi) radary

## Chapter 3

# System overview

- koncept systemu (rozlozeni jednotek, pouzity hardware, propojeni, ...) - zminení základního konceptu běhu celého systému od detekce cíle až po jeho lokalizaci - zmínit základních problematiky celého systému: zastavení, rektifikace, detekce, tracking, výběr jednotek pro sledování, naježdění po optické ose, předání cíle, triangulace, ... - možnost řízení jednotky pomocí joysticku/klávesnice - zmínit problematiku zastavení a rektifikace a odkaz na příslušnou kapitolu - zmínit použití ROS konvence co se týká souřadných systému

## Chapter 4

# Camera unit

- z ceho sestava camera unit (zminit manipulator i kameru) - návrh kinematického řetězce modelu manipulátoru - ukazat 2D diagramy, 3D model z rvizu/gazeba - fotka - schema, kde bude ukazano, co je azimuthal a elevation axis
  - PTU Flir - HW/SW limity pohybu - krokove motory - proble proklouznuti kroku
  - nejake info o kamere



## Chapter 5

# Sensitivity analysis

The precision of the system can be defined in the means of the frame-by-frame Euclidean distance between the estimated location and the real (ground truth) location of the given target. The precision is impacted by multiple independent factors, thus it is essential to perform the sensitivity analysis in order to discover and prospectively alleviate the most prominent contributors of the overall error.

- vysvetleni typu chyb: - system error (systemova chyba) - nesoulad modelu CU s realnou konstrukci - nespravne mereni heading - detekce a tracking ? - uncertainty of the input of the system - GPS mereni - data inklinometru

- vysvetlit, ze se budeme snazit potlacit jen nektare chyby (neresime treba nepresnost mezi modelem a realnou CU z hledsiak translaci mezi klouby)

- rozdeleni na chybu rotace a translace - nepresnost v rotaci je daleko zavaznejsi, nez nepresnost v translaci - priklad a obrazek vlivu nepresnosti o x mrad na lokalizaci cile ve vzdalenosti y m/km

- zdroj chyb: - detekce - tracking - GPS pozice - natoceni vuci severu - rozliseni PTU - model PTU - translace mezi klouby - model PTU - rotace mezi klouby - uchyceni kamery - rotace podle opticke osy - rotace podle osy azimutu - rotace podle osy elevace

## Chapter 6

# Stationing and rectification

As explained in section 5 the precision of the whole system is dependent on on the uncertainty of the system input as well as on the imprecision of the camera unit construction. The process of stationing aims to alleviate the uncertainty of the system input while the main purpose of the rectification is to reduce the difference between the real camera unit and its model.

### 6.1 Stationing

Since the stationing is considered to be already working subsystem of the whole project (and thus is not dealt with within the scope of this work) only the main principle will be briefly described. The stationing is composed of two parts: finding the geographical north and finding the relative azimuthal and elevation angles between each pair of camera units.

#### 6.1.1 Geographical north

Though it is common practice to estimate the heading<sup>1</sup> using the magnetometer, this device is unsuitable for this project since the accuracy of the concurrent professional class magnetometers is insufficient (see Section 5). For instance the accuracy of the magnetometers meant for compassing applications produced by Honeywell, the multinational company focusing on aerospace systems, range from hundreds to thousands of milliradians [2].

In order to find the orientation of each camera unit placed in the outdoor environment, distinctive landmarks (created either by human or nature) with known geographical positions are used. For each such a landmark the manipulator is rotated so that the optical axis of the camera would intersect that landmark and both the azimuth and elevation value is registered. Using triangulation the geographical position of the camera unit is derived.

Different possible approach takes advantage of the celestial objects, such as the moon, sun or stars for which the current geographical position is known as well. Nevertheless this approach can only be used between the sunset and the dawn.

#### 6.1.2 Relative azimuth and elevation

To further reduce the impact of the uncertainty of the system input produced by the GPS (see Section 5) and the system error given by the imprecision of the heading estimation (see

---

<sup>1</sup>Heading is the term used to describe the angle between the torso of the human body and the geographical north [1]

Section 6.1.1) it is convenient to find the relative position of each camera unit with regards to the rest of the camera units.

The information about the geographical position of all camera units as obtained from the GPS sensors is distributed across the whole system. Each pair of camera units then automatically performs the following:

1. Set the azimuth and elevation of the manipulator so that the optical axis of the camera would intersect the expected location of the LED target of the other camera unit.
2. Using the visual clue adjust the azimuth and elevation so that the optical axis of the camera would intersect the center of the LED target of the other unit (see Figure 6.1).
3. Save the current azimuth and elevation values of both camera units and use those values to update the model of the system.

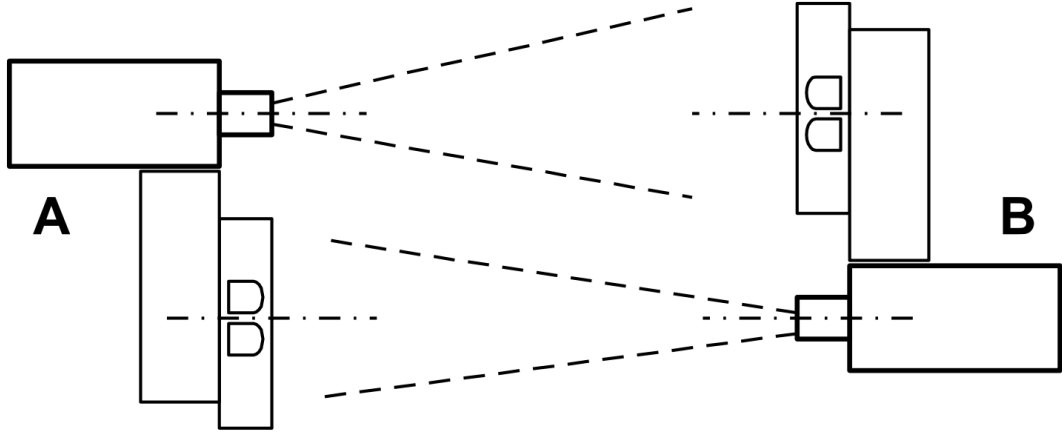


Figure 6.1: The schema of stationing process where two camera units attempt to align the optical axes of their cameras so that they would intersect the LED target of the other unit.

### 6.1.3 Horizontality

Since the camera unit is expected to be placed in an unknown outdoor terrain, it will never stand on an ideally horizontal surface. Thus it is necessary to either ensure that the unevenness of the surface is compensated by the suitable setting of the camera unit's stand or both the side tilt and front tilt angles of the stand must be estimated and integrated to the model of the given camera unit. For these purposes the inclinometer attached to the base plane of the camera unit (see Section 4) is used.

## 6.2 Rectification

The process of rectification serves the purpose of reducing the system error caused by the imprecise attachment of the camera to the manipulator. The model of the camera unit assumes that the camera is precisely attached to the manipulator so that the camera image sensor is positioned perpendicular to the azimuthal axis and the rows of the image sensor are parallel to the elevation axis (i.e. the camera is not rotated along the optical axis).

Regarding these requirements the rectification consists of three parts: rotation along the optical axis, rotation along the azimuthal axis, finding the default elevation angle.

### 6.2.1 Rotation along the optical axis

A custom made metal mount is attached to the bottom side of the camera. The mount is then attached to the manipulator using two opposing round tenons enabling for the rotation of the mount (together with camera) along the axis parallel to the optical axis of the camera (see Figure 6.2).

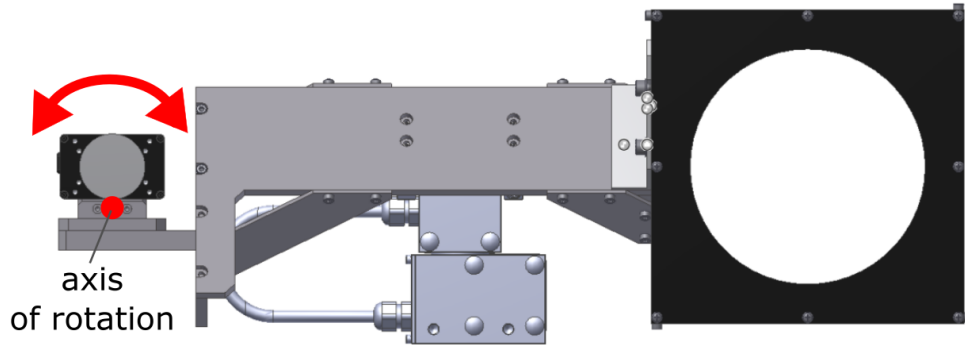


Figure 6.2: Front view of the top part of the camera unit. The red arrow shows the possible rotation of the camera along the axis parallel to the optical axis.

In this part the rectification target with three parallel horizontal black lines is used. Each line has different width so that the operator can select the most suitable one (given the distance of the target, ambient lighting conditions, etc.) As the first step a surveying automatic level is used to rotate the target so that the black lines become horizontal. Then the camera is pointed approximately to the center of the target. The camera image stream is blended with the same stream mirrored across the vertical axis. The operator then manually rotates the camera so that the black lines in this blended image stream appear visually aligned (see Figure 6.3). Once set, the mount with the camera is fixed to the manipulator using two set screws.

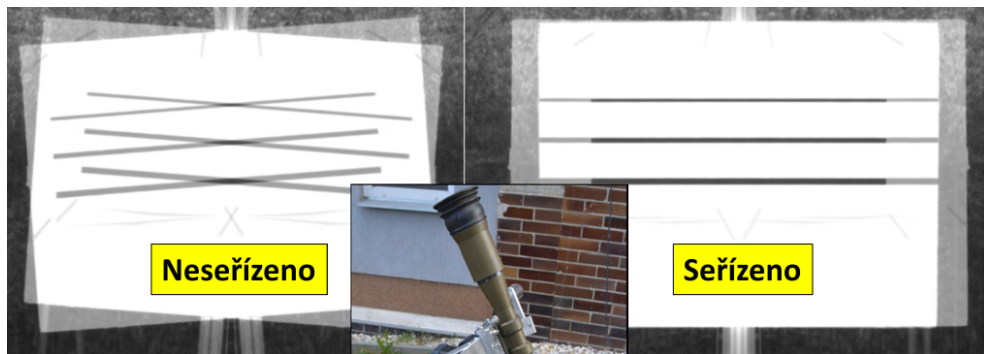


Figure 6.3: A blended image stream from the camera before (left) and after (right) rotating the camera along the optical axis to the correct position.

### 6.2.2 Rotation along the azimuthal axis

The mount can still rotate along the axis parallel to the azimuthal axis (see Figure 6.4). It is necessary to ensure that the optical axis of the camera is perpendicular to the elevation axis.

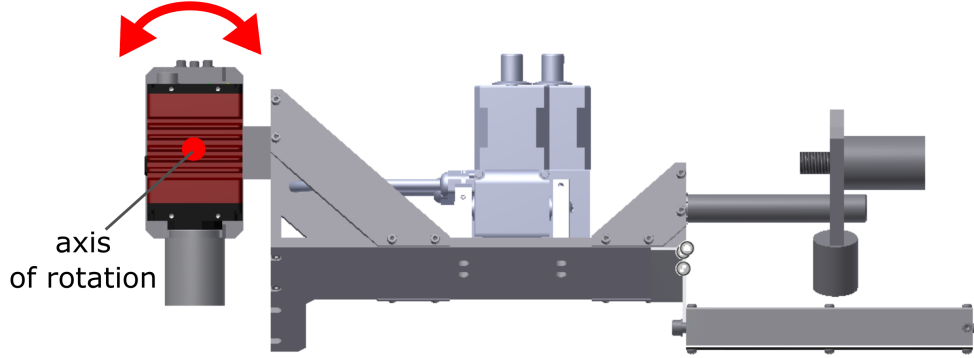


Figure 6.4: Top view of the top part of the camera unit. The red arrow shows the possible rotation of the camera along the axis parallel to the azimuthal axis.

The same target from the first part of the rectification is used, but two black crosses are added to the selected horizontal black line. The distance  $d_{ao}$  m between two crosses equals to the distance between the azimuthal and optical axis (which is known from the engineering design, see Figure 6.6).

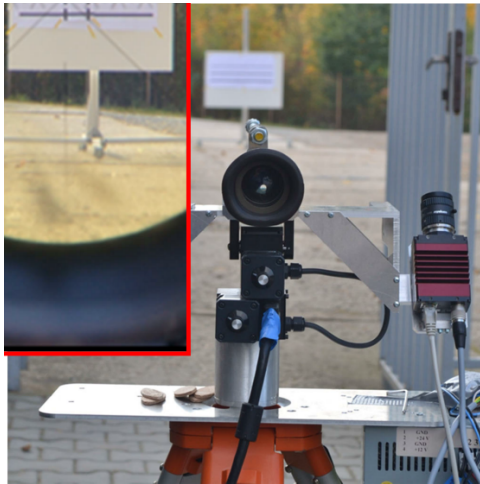


Figure 6.5: A telescope mounted on top of the manipulator. A person looking through a telescope sees a crosshair - a tip of a triangle.

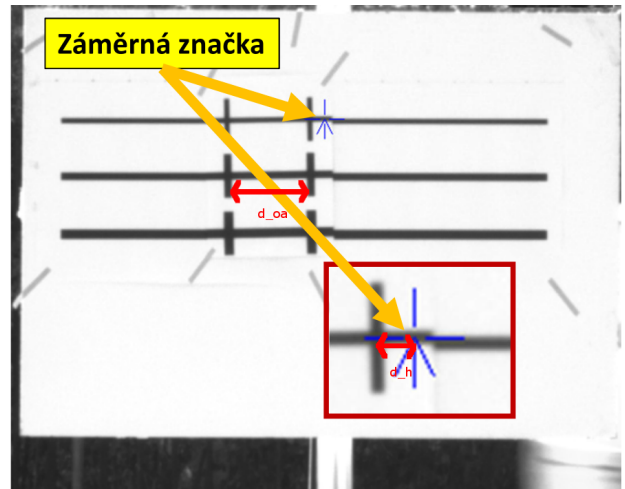


Figure 6.6: Rectification target with the pairs of black crosses. The two crosses in a pair are  $d_{ao}$  m apart. A digital crosshair is displayed in order to find the horizontal offset  $d_h$ .

A military optical monocular telescope (see Figure 6.5) is mounted on top of the manipulator. The optical axis of the telescope intersects the azimuthal axis, it is perpendicular to it and it intersects the left cross of a given pair on the rectification target. The camera is rotated so that its optical axis (represented by the digital crosshair) intersects the right cross on the target and then is fixed using set screws. As the screws are tightened

the camera is unintentionally rotated a bit again which causes the visual offset between the crosshair and the cross on the target. The offset expressed in pixels is recorded and transformed to the default angle  $\beta$  expressed in milliradians (see Figure) of rotation along Z-axis of the joint **camera** in the camera unit model (see Section 4):

$$\beta = \arccos \frac{focal\_length}{offset},$$

$$offset = pixel\_offset * pixel\_size$$

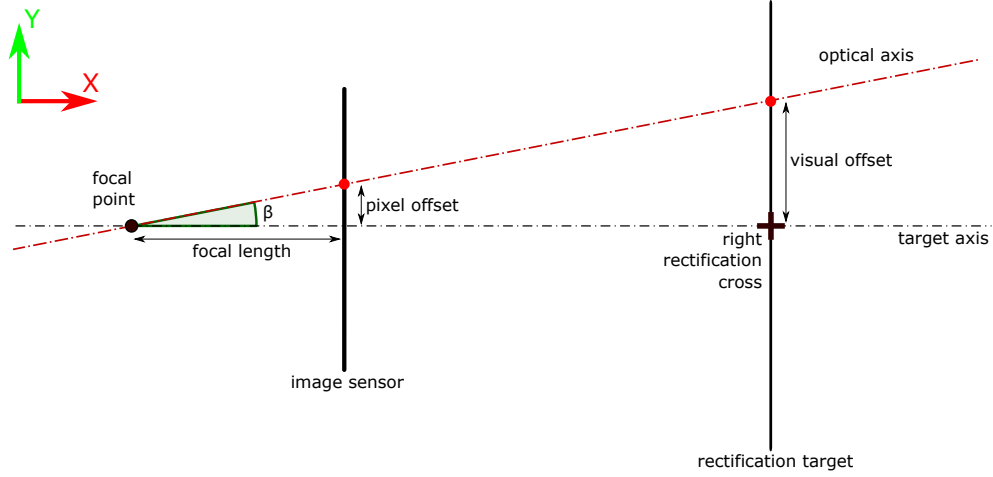


Figure 6.7: The top view schema of a rectification target being projected to the image sensor of the camera. The value of the angle  $\beta$  is one of the output of the rectification process.

### 6.2.3 Finding the default elevation angle

Given the application of the system the camera is expected to mainly observe the sky. Considering the limited elevation range of the manipulator **Flir PTU D46-70** (see Section 4), the camera must be mounted to the manipulator with the default elevation angle approximately  $-60^\circ$ . However, after fixing the camera it is necessary to find default elevation angle precisely.

For this purpose a pair of rectification targets, which consist of vertical black and white lines representing the marks of a ruler. The targets are positioned in a row with the distance of a few meters so that the front target would overlap approximately half of the rear target when observed from the camera. Both targets must be rotated so that the lines become horizontal, then the operator manually adjusts the elevation of the manipulator so that the digital crosshair would intersect the same mark on both targets where the two marks form a straight line (see figure 6.8). Once found the current elevation angle is recorded and integrated to the model of the camera unit as an angle of rotation along the Y-axis of the joint **camera** (see Section 4).



Figure 6.8: Front target of a pair of the rectification targets used to find a default elevation angle (left). A screenshot from the image stream of the camera with the crosshair focused on a row where the marks of the rulers align (right).

## Chapter 7

# Detection and tracking

- === DETEKCE VZDALENÝCH OBJEKTU V OBRAZE === - === SLEDOVÁNÍ OBJEKTU VE VÍCEKAMEROVÝCH SYSTÉMECH === - === ALGORITMY VHODNÉ PRO ŘEŠENÍ ÚLOHY ===
  - TLD - možnost sdílení info o appearance modelu mezi CUs



## Chapter 8

# Cooperation among camera units

- vyber druhe jednotky pro trackovani cile - najezd po opticke ose - predani cile

## Chapter 9

# Target localization

- triangulace - === LOKALIZACE OBJEKTU VE VICEKAMEROVYCH SYSTEMECH  
=== - === ALGORITMY VHODNE PRO RESENI ULOHY ===  
== (obrazek z rvizu (protnuti primek) ==

## Chapter 10

# Implementation

The whole system is built on the robotic framework Robot Operating System (for details see Section 10.1). Since the ROS defines multiple conventions, restrictions and best practices the whole system design including the selection of a programming language, a programming and a communication paradigm, a target platform and a tool for physical simulations is impacted by the possibilities of this framework.

As of writing this text a current state of the implementation mainly builds on the virtual environment provided by the physical simulator Gazebo (see Section 10.3). When confronted with the overview of all subsystems making up the whole system presented in Section 3 so far the following parts are already designed, implemented and/or integrated:

- manual control of the manipulator using peripheral devices
- manual selection of a target and distribution of its appearance to all CUs
- integration and utilization of a OpenTLD tracker
- triangulation of a 3D positions of a target within global frame
- integration of all subsystems

Furthermore a few additional tools were utilized and/or implemented as the necessary building blocks allowing for further development and testing:

- a functional model of a whole system in Gazebo environment
- a standalone application for rectification

### 10.1 Robot Operating System

Despite its name the ROS<sup>1</sup> is not an operating system but rather a collection of open source libraries, tools and conventions which serve the purpose of a middleware running alongside a real operating system, however it provides the programmer with the hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management [3].

Since the original motivation for developing ROS was to support the collaboration among the experts in the field of robotics in the means of a common software platform

---

<sup>1</sup>The official website of ROS: <http://www.ros.org>

[4], a huge developer community has formed around ROS which resulted in wide-scale penetration of this framework as well as the support for a huge range of hardware devices.

### 10.1.1 Application of ROS

The OLS is designed to become a relatively complex system, thus it exhibits non-trivial implementation requirements such as a need to distribute the computation among multiple computers, the real time performance, integration with physical simulator, etc. Since ROS is a mature framework satisfying the most of these requirements (see Table 10.1), it was chosen as a main implementation platform.

Table 10.1: The table lists the most important requirements of the OLS and describes how the ROS framework addresses them.

OLS requirements	ROS features
native support for hardware such as Prosilica camera, manipulator Flir PTU D46-70, joystick, keyboard	nodes implementing image capture from Prosilica cameras, capturing events from keyboards and joysticks
modularity and reusability of source code	each subsystem is represented by a separate process (node), straightforward reusability
distribution of computation among multiple computers	provides abstraction layer for distributing nodes across devices
simple data exchange among subsystems	the publisher subscriber paradigm [3], support for custom message formats
real time performance	C++ implementation
modelling and simulating the robot	custom URDF language <sup>2</sup> for robot modelling, integration with Gazebo
specifying a kinematic chain, heavy 3D transformation computation	native support for computing transformation between frames using package <code>tf</code>
complex visualization, debugging	a tool <code>rviz</code> <sup>3</sup> for visualizing frames, transformations, robot models, image streams etc.
physical simulation	integration with Gazebo

### 10.1.2 Standard ROS nodes

- jake uzly jsou pouzity

---

<sup>3</sup>aohj

<sup>3</sup>hoho

## 10.2 System architecture

- architektura systému - propojení uzlů, komunikace přes zprávy

== rqt\_graph celého systému == == dvojice snímku z gazebo == - pohled na celou scénu s benzínkou - pohled přes jednu jednotku - OpenCV, se zaměřováním

## 10.3 Application of Gazebo

== obrázky z rvízu, kde je vidět jak scéna tak stream z virtuálních kamer ==

## 10.4 External libraries

Besides the framework ROS a few other publicly available libraries are used within the implementation.

**OpenCV** Open Source Computer Vision Library<sup>4</sup> is a free open source library providing algorithms for image processing, computer vision and machine learning. Version 2.4.11 is used as it is a component of ROS Indigo.

**Eigen** This open source C++ template library<sup>5</sup> implements the data structures and methods for fast and convenient solving of linear algebra problems.

**OpenTLD** The OpenTLD library<sup>6</sup> represents an open source C++ implementation the TLD tracking algorithm (see Section 7).

## 10.5 Design of GUI

- předbežný návrh GUI

- použití ROSu, ukázat diagram z rqt\_graph, návrh komunikace přes zprávy - použití existujících uzlů v ROSu - gazebo - hardware in a loop - rozvržení komunikace uzlů - použití rvízu pro vizualizaci a ověření funkčnosti

---

<sup>4</sup>The official website of OpenCV: <http://opencv.org>

<sup>5</sup>The official website of Eigen: <http://eigen.tuxfamily.org>

<sup>6</sup>The official website of OpenTLD: <http://www.gnebehay.com/tld>

## Chapter 11

# Experiments and results

- vlastní datová sada - experimenty na získané a porizované datové sady - === EXPERIMENTY NA REALNÝCH DATECH ===

## Chapter 12

## Conclusion

- zhodnocení dosažených výsledků se zvlášť vyznačeným vlastním přínosem studenta - zhodnocení z pohledu dalšího vývoje projektu vzhledem k diplomové práci - možnosti dalšího vývoje: modelování okolí ve 3D nebo využití této info z mapy, paralelizace náročných částí na GPU - věci, co se zatím nebraly v úvahu - systém je umístěn na rozhraní dvou a více UTM zón

# Bibliography

- [1] Manne Henriksson. Estimation of heading using magnetometer and gps., 2013.
- [2] Honeywell. Magnetic sensors product catalog - sensing earth's magnetic field [online].  
[https://aerospace.honeywell.com/~media/Images/Plymouth%20Website%20PDFs/Magnetic%20Sensors/Technical%20Articles/Sensors\\_Product\\_\\_Catalog.ashx](https://aerospace.honeywell.com/~media/Images/Plymouth%20Website%20PDFs/Magnetic%20Sensors/Technical%20Articles/Sensors_Product__Catalog.ashx), 2015.
- [3] Jason M. O’Kane. *A Gentle Introduction to ROS*. CreateSpace Independent Publishing Platform, 10 2013.
- [4] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.