

Bachelor's Thesis



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

SW and HW Integration of an IP PTZ Camera onto a Mobile Outdoor Robot

Jakub Chmel

Cybernetics and Robotics, Robotics
chmelja5@fel.cvut.cz

May 2015

<ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/Chmel-TR-2015-02.pdf>

Supervisor: Doc. Ing. Tomáš Svoboda, PhD

BACHELOR PROJECT ASSIGNMENT

Student: Jakub Chmel
Study programme: Cybernetics and Robotics
Specialisation: Robotics
Title of Bachelor Project: SW and HW Integration of an IP PTZ Camera onto a Mobile Outdoor Robot

Guidelines:

The outdoor robot for urban search and rescue (USAR) missions [2] lacks a high quality pan-tilt-zoom (PTZ) video stream which is criticized by some rescuers. This work should fill the gap. The goal is to integrate a PTZ camera into the system. This will require hardware and software contribution. From the hardware point of view, the main challenges are robustness against weather conditions and vibrations of the robot. The mounting should be flexible in place and base height. Also the electricity supply will need some design choices. Video grabbing and camera control must comply with the ROS publish/subscribe standards [4] and follow the ideas of a virtual camera [3]. The main publishing node should allow a fine grain control of video quality – sometimes there is trade-off between visual quality and the need of reducing the amount of data being published and transferred.

A part of the deployment procedure is also a verification of applicability of calibration procedure [1] for computing an initial external calibration.

Bibliography/Sources:

- [1] Jan Brabec: Automated camera calibration from laser scanning data in natural environments. Bachelor thesis, CTU FEE 2014.
- [2] Kruijff et al.: Designing, developing, and deploying system to support human-robot teams in disaster response. In *Advanced Robotics*, vol 28, issue 23.
- [3] T. Svoboda et al.: DR 2.3.3: Bi-directional cooperation of low-level vision modules and higher level control. Deliverable of project NIFTi.
- [4] Robot Operating System, <http://ros.org>

Bachelor Project Supervisor: doc. Ing. Tomáš Svoboda, Ph.D.

Valid until: the end of the summer semester of academic year 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 23, 2015

Acknowledgement / Declaration

I would like to thank my advisor Tomas Svoboda for his invaluable guidance and advices, patience and supporting during all year. I would also like to thank Vladimir Kubelka and Libor Wagner for patience in the beginning of using operation system ROS and for assistance in experiments.

Finally, I would like to thank my family for their endless support.

I declare that I have developed the presented work independently and that I have listed all information sources used in accordance with the Methodical guidelines on maintaining ethical principles during the preparation of higher education theses.

In Prague, 22 May 2015

.....

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 22. května 2015

.....

Abstrakt / Abstract

Terénní robot pro městské vyhledávací a záchranné mise (USAR) postrádá vysoce kvalitní pan-tilt-zoom (PTZ) streamované video. Funkce PTZ může být simulována pomocí existující virtuální kamery, která používá data z kamery LadyBug 3 (LB3). Nicméně kamera LB3 může používat pouze digitální zoom.

Tato práce je zaměřena na SW a HW integraci síťové (IP) PTZ kamery na terénního záchranářského robota. Tento typ kamer může přidat funkci optického zoomu do robotického systému. IP PTZ kamera (Axis 214 PTZ) je v současné situaci nejlepší volbou pro popsáný problém.

Cílem práce bylo vyvinout balíček pro Robotický operační systém (ROS), který umožní používat kameru na robotovi TRADR. Většina zdrojových souborů je napsána v Pythnu a některé v C++.

V práci bylo provedeno několik experimentů, které definují možnosti a limity vyvinutého řešení.

Klíčová slova: IP PTZ kamera, LadyBug kamera, optický zoom, kamera, ROS, TRADR

Překlad titulu: Integrace síťové (IP) pan-tilt-zoom (PTZ) kamery na terénního záchranářského robota.

The outdoor robot for urban search and rescue (USAR) missions lacks a high quality pan-tilt-zoom (PTZ) video stream. PTZ function can be simulated using an existing virtual camera, which uses data from LadyBug 3 (LB3). However, the LB3 camera may only use a digital zoom.

This work is focused on SW and HW integration of a network (IP) PTZ camera onto the mobile outdoor robot. These types of cameras can add the optical zoom function to the robotic system. IP PTZ camera (Axis 214 PTZ) is the best choice for described problem in current situation.

The goal of the thesis was to develop Robot Operating System (ROS) package that allows to use the camera on the TRADR robot. Most of source codes were written in Python and some in C++.

In the thesis were performed several experiments that define the possibilities and limits of the developed solution.


Keywords: IP PTZ camera, LadyBug camera, optical zoom, camera, ROS, TRADR

Contents /

1 Introduction	1	5.2 Start ROScore	22
1.1 Motivation	1	5.3 Launch the camera drivers	
1.2 Problem formulation	2	at the robot	22
1.3 Structure of the thesis	2	5.4 Launch the nodes at the base	
2 System specification	3	station	24
2.1 Software and hardware re-		6 Experiments	27
quirements	3	6.1 Optical zoom capability test ..	27
2.2 Selection of suitable camera	3	6.2 Amount of data transmitted ..	29
2.2.1 Summary of Axis 214		6.3 Comparison of two suggested	
PTZ parameters	5	camera positions	31
2.3 Placing the camera on the		6.4 Camera calibration	32
mobile robot	5	6.5 Controlling camera by mouse .	34
2.4 Required accessories	7	6.6 Video shaking induced by	
2.4.1 Camera power supply	7	the robot vibrations	34
2.4.2 Securing data connec-		7 Conclusion	36
tion with the robot	8	References	38
3 Implementation	9	A Enclosed CD	41
3.1 Package structure	10		
3.2 Custom messages	10		
3.3 Topics	10		
3.3.1 Images topics	11		
3.3.2 Command topics	11		
3.3.3 Info topics	11		
3.4 Nodes	12		
3.4.1 axis stream	12		
3.4.2 axis ptz	13		
3.4.3 axis visualizer	15		
3.4.4 axis keyboard teleop	15		
3.4.5 axis mouse teleop	15		
4 User guide	17		
4.1 Installation of ROS and im-			
port package	17		
4.2 Network Setup	17		
4.3 Installation of driver for con-			
verter RJ-45 to USB	17		
4.4 Basic camera setting	18		
4.5 HW connection	19		
4.6 Start and setting up RViz	19		
4.7 Troubleshooting	20		
4.7.1 Freezing in standby			
mode	20		
4.7.2 Problem with the			
mouse teleop node	20		
5 Operator guide	22		
5.1 Summary for the quickest			
start	22		

Tables / Figures

2.1. Tested cameras in previous work	4
2.2. Axis 214 PTZ – object resolution as function of the object size and camera distance.....	5
2.3. Axis 214 PTZ – overview of selected important parameters ..	6
2.4. Parameters of DC-DC convertor	7
5.1. Summary of the available camera controls	23
5.2. Axis drivers – range of parameters and default values ...	24
5.3. The effect of keystroke	25
6.1. Events affecting zoom.....	29
6.2. Amount of data transmitted – high resolution	30
6.3. Amount of data transmitted – medium resolution	30
6.4. Amount of data transmitted – low resolution	30
6.5. Amount of data transmitted – raw image	30
6.6. Events affecting the video quality	30
6.7. Results of the calibration scene with furniture	33
1.1. Rescue robot with mounted omnidirectional camera Ladybug 3	1
1.2. Possible scenarios	2
2.1. Camera Vivotek PZ7131	4
2.2. Camera Axis 215 PTZ	4
2.3. Camera Bosch MIC IP starlight 7000 HD	4
2.4. Camera Axis 214 PTZ	5
2.5. Camera Vivotek PZ8111/21	5
2.6. Possible places to mount the camera on the robot	6
2.7. Schematic connection diagram for the I/O terminal pins	7
2.8. DC-DC convertor	8
2.9. External network card	8
3.1. Diagram of the package structure	9
3.2. Callback function for topic <code>axis214PTZ/Relative</code>	14
3.3. Callback function for topic <code>axis214PTZ/Absolute</code>	14
3.4. Callback function for topic <code>axis214PTZ/PointInImage</code> ...	14
4.1. Network card configuration....	18
4.2. Setting anonymous mod for camera Axis 214 PTZ.....	19
4.3. RViz creating new visualization	20
4.4. Part of main screen RViz.....	21
5.1. Mouse teleop windows	26
6.1. Photo documentation of the experiment – the optical zoom capability	27
6.2. Spherical image from Ladybug – outdoor experiment	27
6.3. Range of the possible use of the optical zoom	28
6.4. Effect of compression	29
6.5. Photo documentation of the experiment – comparison of two suggested camera position .	31
6.6. Range of the possible use of the optical zoom	32



6.7. The calibration scene with furniture	33
6.8. Definition of the fixed place of the camera stand	34
6.9. Size of error in controlling the camera by mouse	35

Chapter 1

Introduction

1.1 Motivation

The TRADR outdoor robot for urban search and rescue (USAR) missions [1] lacks a high quality pan-tilt-zoom (PTZ) camera. This fact is criticized by some rescuers. The purpose of this project is to find a possible solution. PTZ function can be simulated by using an existing virtual camera [2], which uses data from Ladybug 3¹) (LB3). Ladybug is an omnidirectional camera, see Figure 1.1, which provides a spherical 12-megapixel image. It is very suitable for remote visual control and for various computer vision algorithms, however a human operator – rescuer – sometimes needs a closer and more detailed view. Rescuers need to inspect Registration marks of cars, safety signs or labels, names of barrels with chemicals or injuries on found persons. All of these operations can be done from a sufficient distance and it is not necessary to arrive closer. In many cases it is not even possible. Sometimes these objects can be in inaccessible place due to their height or depth. Examples of application are shown in Figure 1.2. The lack of optical zoom function is the biggest problem of current solution with a virtual camera because it uses only a digital zoom.

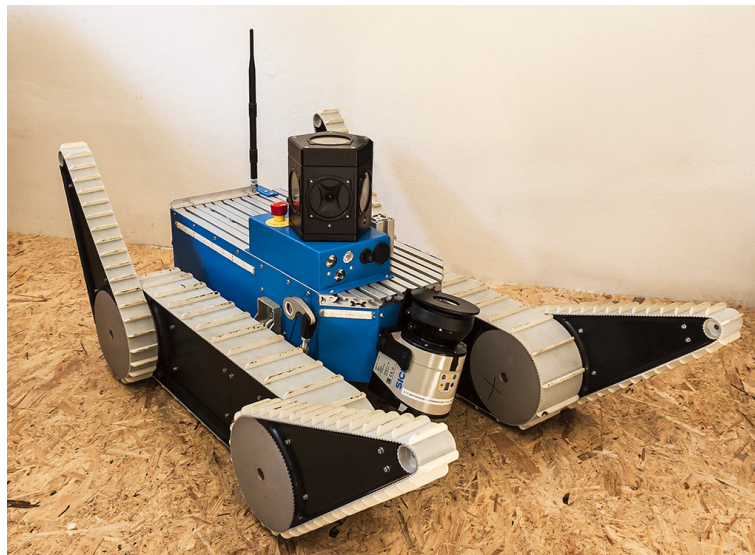


Figure 1.1. Rescue robot with mounted omnidirectional camera Ladybug 3 (the black thing on the robot).

¹) <http://www.ptgrey.com/ladybug3-360-degree-firewire-spherical-camera-systems>



a) Warehouse in fire



b) Collapsed warehouse



c) Crevasse



d) Haiti disaster area

Figure 1.2. Possible scenarios: Shows necessity to zoom at distant details and investigate holes. Original image is from [3].

1.2 Problem formulation

The goal of the work is to select the most suitable PTZ camera and integrate it to the system. This requires a solution of several hardware and software problems. From the hardware point of view: One of the main challenges is the robot vibration – the camera and its mounting should withstand the robot vibrations and also, despite the vibration, provide a usable video for the operator. The mounting should be flexible in place and base height. The electricity supply of the camera will need some changes. Also the physical transfer of data will require some solution because the robot does not have a free ethernet port. In the field of software: It is necessary to integrate the camera into the Robot Operating System (ROS) while respecting ROS publish/subscribe standards. It is therefore necessary to implement several ROS nodes that allow to control a position of camera and transmit video stream from the camera. Sometimes it is necessary some compromise between visual quality and amount of transmitted or published data. For this reason, it is imperative to be able to change the parameters of the transmitted video.

1.3 Structure of the thesis

This work is divided into the following chapters. Chapter 2 *System specification* establishes requirements for selection of the most suitable PTZ camera, discusses its mounting on the robot and describes necessary accessories. The following chapter 3 *Implementation* explains the structure of the project and describes the purpose of each ROS node and topic. Chapter 4 *User guide* gives the instructions for configuring the camera. Chapter 5 *Operator guide* focuses on practical and quick work with camera without the need to understand how each part works. Chapter 6 *Experimentis* documents executed tests and experiments.

Chapter 2

System specification

This chapter discusses camera selection and several design choices related to the camera integration. The last part of this chapter describes necessary accessories needed for connecting the camera to the current state of the mobile rescue robot.

2.1 Software and hardware requirements

On the camera market there are a lot of IP PTZ camera models. Therefore several requirements were defined to select the most suitable model.

The proposed solution should be waterproof (at least IP55 preferably IP65¹) and shock/vibration proof (MIL-Std 810E/F²) [3]. These standards are suitable for good function of the mobile robot. Other requirements are low weight and compactness of the whole solution because load capacity of the robot is limited. So the whole solution should not exceed 2 kilograms. The requirements (low weight and compactness) are quite strict and contradictory because the more resistant camera is the heavier and bigger it is.

The goal of this project is to expand the robot functions by adding an optical zoom into the concept with already mounted spherical Ladybug camera. The searched camera should be able to zoom in ideally to any scene point around the robot. The ideal is 360 degree rotation in axis pan including the possibility of a negative tilt for the camera to easily zoom in objects below the level of an optical axis. The size of the optical zoom should be at least 10x. Such a value of zoom is sufficient enough to display objects that the operator sees in the panoramic image. Larger values of the optical zoom (more than 15x) would be useful in robotic groups, where one robot does a role of observer for others.

The software requirements include the need of obtaining the video from the camera without noticeable delays. The resolution of images is not a priority. The camera does not have to replace the current solution with Ladybug, but should be its complement. It is therefore important to be able to easily adjust the video parameters (resolution, fps, compression etc.) because video is transmitted via wifi with limited transfer rate.

2.2 Selection of suitable camera

Two models of IP PTZ camera were already tested in previous work [4]. First tested camera was Vivotek PZ7131³), see Figure 2.1, which did not meet wanted requirements, more specifically the problem with video delays. The second tested camera was Axis 215 PTZ⁴), see Figure 2.2. The parameters of these cameras are shown in Table 2.1.

¹) <http://www.dsmt.com/pdf/resources/iprating.pdf>

²) http://everyspec.com/MIL-STD/MIL-STD-0800-0899/MIL-STD-810G_12306/

³) <http://www.vivotek.com/web/product/productdetail.aspx?model=pz7131>

⁴) <http://www.axis.com/global/en/products/axis-215>



Figure 2.1. Camera Vivotek PZ7131 [5].



Figure 2.2. Camera Axis 215 PTZ [6].

Parameter	Vivotek PZ7131	Axis 215 PTZ
Pan range	$-175^{\circ} \sim +175^{\circ}$	$-170^{\circ} \sim +170^{\circ}$ (360° via Auto-flip)
Tilt range	$-35^{\circ} \sim +90^{\circ}$	$0^{\circ} \sim 180^{\circ}$
Zoom	2.6x optical zoom	12x optical zoom
Minimum illumination	1.25 lux / F1.9 (color)	1 lux / F1.6 (color)
Frame rate	max 30 fps	max 30 fps
Resolution	max 640x480	max 704x480
Weight	352 g	730 g

Table 2.1. Parameters of tested cameras in previous work [4].

The results of tested cameras have helped to improve the requirements for the final solution and identify possible problems.

There are many IP PTZ camera manufacturers e.g. Axis Vivotek, Panasonic, Sony, Pelco, Bosh, Flir, LG and Samsung. The most suitable cameras with their parameters are listed in [4], page 25. The overview shows that the market do not offer small, waterproof and adequately light cameras. Manufacturers make waterproof camera through external accessory, called housing. For this reason, cameras are heavy and large. Very interesting model is a Bosch MIC IP starlight 7000 HD¹⁾ camera, see Figure 2.3, which meets required norms. However it is heavy (6.7 kg) and rather intended for mounting on a car and it is not suitable for the project.



Figure 2.3. Camera Bosch MIC IP starlight 7000 HD – meets required norms but it is heavy and rather intended for mounting on a car [7].

¹⁾ http://resource.boschsecurity.us/documents/MIC_IP_starlight_700_Data_sheet_enUS_16950843019.pdf

As a consequence the norms requirements were simplified. The decision was made between two models Axis 214 PTZ¹⁾, see Figure 2.4 and Vivotek PZ8111/21²⁾, see Figure 2.5. At the end the Axis 214 PTZ camera was chosen for several reasons: previous negative experience with the Vivotek³⁾ manufacturer; presence of this particular camera at the department; better parameters.



Figure 2.4. Camera Axis 214 PTZ [8].



Figure 2.5. Camera Vivotek PZ8111/21 [9].

2.2.1 Summary of Axis 214 PTZ parameters

Axis 214 PTZ camera provides a sufficient range of pan and tilt. The camera has 18x optical zoom. It enables to zoom in distant objects, see Table 2.2, which gives the resolution of an object of a defined size in a given distance from the camera. More values can be obtained on the manufacturer's website, see Lens calculator⁴⁾. The overview of important camera parameters are listed in Table 2.3.

Nevertheless the camera has also two disadvantages. It is not waterproof without housing and it has not been tested for impact resistance and vibrations.

optical zoom	Camera-object distance				
	1 m	5 m	10 m	25 m	100 m
1x	791	158	79	32	8
6x	5,308	1,062	531	212	53
12x	10,116	2,023	1,012	405	101
18x	14,937	2,987	1,494	597	151

Table 2.2. Axis 214 PTZ – object resolution as function of the object size and camera distance. It gives the resolution in pixels for object 1m long [10].

2.3 Placing the camera on the mobile robot

After selection of the camera, the next step is to determine the most suitable place for the mounting on the robot. Several factors are considered when choosing the place.

¹⁾ http://www.axis.com/files/datasheet/ds_214ptz_33006_en_0809_lo.pdf

²⁾ http://download.vivotek.com/downloadfile/downloads/datasheets/pz81x1datasheet_en.pdf

³⁾ The camera Vivotek PZ7131, Figure 2.1, has a problem with a delay of about 2 seconds.

⁴⁾ www.axis.com/global/en/tools/lens-calculator

Parameter	Axis 214 PTZ
Pan range	$-170^{\circ} \sim +170^{\circ}$
Tilt range	$-30^{\circ} \sim +90^{\circ}$
Zoom	18x optical zoom
Lens	4.1 – 73.8mm
Minimum illumination	0.3 lux / F1.4 (color) 0.005 lux / 1.4 (B/W)
Frame rate	max 30 fps
Resolution	max 704x576
Power	11 – 13 VCD max. 14 W
Weight	1110 g

Table 2.3. Axis 214 PTZ – overview of selected important parameters [11].

- The camera needs a good view without barriers from other parts of the robot.
- The camera body possibly should not occlude view-field of the panoramic camera.
- The camera must not interfere with other devices mounted on the robot.
- The camera should not be higher than other devices placed on the robot. Otherwise it can reduce the throughput of the robot.

We propose two possible places for mounting the camera on the robot, see Figure 2.6. Either in front of the robot next to the Ladybug camera or at the rear of the robot behind the camera Ladybug. The final decision was to place the camera at the rear because it better meets the requirements and moreover the robot is better balanced. The camera is attached to a 20 centimeters tall stand which is made from Item aluminium profiles. The highest point of the camera is 39.5 centimeters above the surface of the robot. The camera calibration from laser scanning data was used for computing an initial external calibration [12].

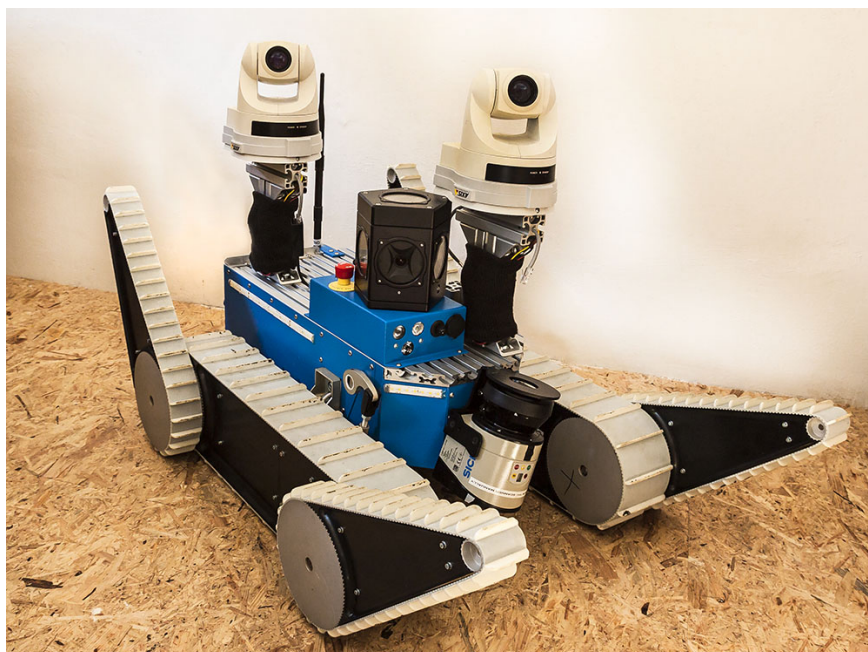


Figure 2.6. Two possible places to mount the camera on the robot. In both cases, the camera is placed on the stand which is made from Item aluminium profiles.

2.4 Required accessories

Two problems need to be solved for the camera to be able to operate on the mobile rescue robot. The first problem is to find a power source compatible with the robot, see Subsection 2.4.1. The second problem is that the robot has only one network card with one RJ-45 interface and it is occupied, see Subsection 2.4.2.

2.4.1 Camera power supply

The robot includes a 48V battery and a voltage converter, where one branch is brought out to the surface of the robot. This is done by Switchcraft L712AS¹⁾ jack, which voltage is 24 VDC. Axis 214 PTZ requires a voltage 12 VDC so it needs a DC-DC converter. The DC-DC converter Orion IP67 24/12-4²⁾ is convenient because it is small, light and waterproof (IP67), see Figure 2.8. More information about this converter find Table 2.4. Next, the Switchcraft 761KS15³⁾ jack is needed to connect the converter into the power source on the mobile rescue robot. Other side of the converter is connected up the I/O terminal pins on the camera according to the scheme in the Figure 2.7.

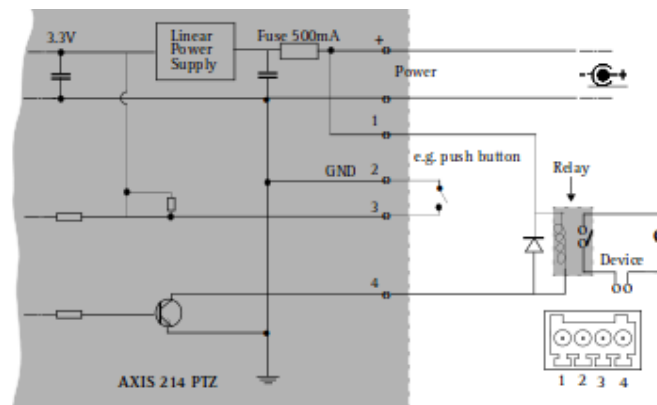


Figure 2.7. Schematic connection diagram for the I/O terminal pins on the camera [13]. Pins 1 and 2 are used for auxiliary DC power. The I/O terminal pins on the camera are numbered left to right.

Parameter	Orion IP67 24/12-4
Input voltage range	15-40 VDC
No load current at 24 V	1 mA
DC output voltage	12 V +/- 3%
Maximum continuous output current	4 A
Waterproof	IP67
Weight	50 g
Dimensions (h x w x d)	25 x 43 x 20 mm
Vibration	IEC68-2-6: 10-150 Hz / 1.0 G
Price	21 EUR

Table 2.4. Parameters of DC-DC convertor Orion IP67 24/12-4 [14].

¹⁾ http://cz.rs-online.com/web/p/napajeci-zasuvky-dc/7051525/?origin=PSF_435722|acc

²⁾ <http://www.victronenergy.com/upload/documents/Datasheet-Orion-IP67-24-12-DC-DC-converters-EN.pdf>

³⁾ http://cz.rs-online.com/web/p/napajeci-zastrcky-dc/7051519/?origin=PSF_435721|acc

■ 2.4.2 Securing data connection with the robot

The robot has no free interface RJ-45 but has enough free USB ports. The robot is equipped with an external network card – Axago ADE-SG¹⁾, see Figure 2.9. The installation procedure is described in Section 4.3.

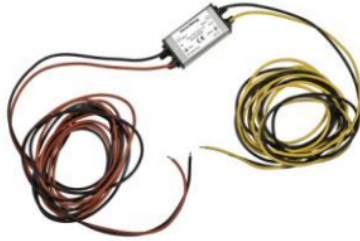


Figure 2.8. DC-DC convertor Orion IP67 24/12-4, which is used for camera power supply [14].



Figure 2.9. External network card Axago ADE-SG [15].

¹⁾ <https://www.alza.cz/kod/AG200a64>

Chapter 3

Implementation

This chapter describes briefly every topic the proposed system publishes/subscribes and every node. If you just want to use the module, you can skip this chapter and read the User guide, Chapter 4, Operator guide, Chapter 5.

The module is implemented as a ROS [16] package and is written for version of ROS – Indigo Igloo. Most of source codes are written in Python and some in C++. The following libraries are used: threading [17], urllib2 [18], OpenCV [19], cv_bridge [20], image_transport [21] and tf [22]. All these libraries are distributed with ROS Indigo Igloo.

The priority for the module is robustness and easily extensibility. The package is based on the source codes of the ROS camera driver for Axis network cameras [23] and uses http commands from the Axis API [24–25]. Figure 3.1 shows the structure of the nodes used in the module, topics for communicating between nodes and request/response communication with a http server in the camera. More detailed description of each part is given in the following subsections.

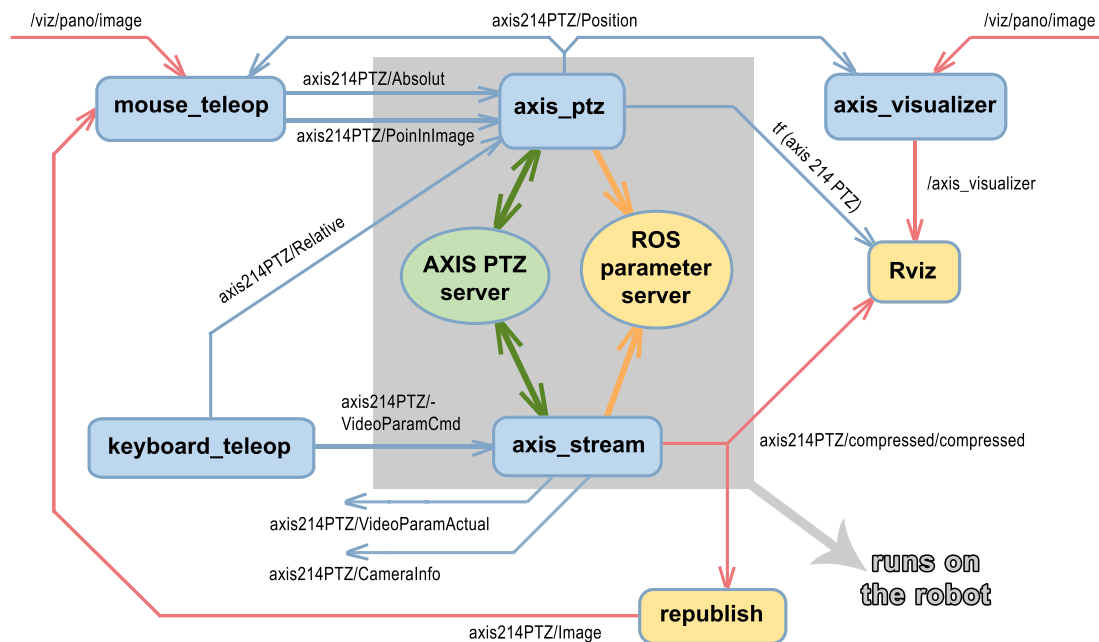


Figure 3.1. Diagram shows ROS nodes, topics and communication with http and ROS parameter server. The gray area contains the nodes that are run at the robot (others nodes run at the base station). The nodes are bounded by rectangles (blue – `axis_camera` nodes, orange – external nodes). Thin arrows are ROS topics (red arrows – contain image data, blue arrows – contain commands). Thick arrows represents communication with AXIS PTZ server (green) and ROS parameter server (orange).

3.1 Package structure

The name of developed package is `axis_camera`. Its package structure is describe in this subsection.

The catkin build system uses the following package structure: [26–27]:

- **[launch]**
All launch configurations are stored here.
- **[msg]**
Directory for the custom ROS messages. Both `msgAxisVideo` and `msgAxisPTZ` messages are stored here.
- **[nodes]**
All of the source files are located here.
- **CMakeLists.txt** [26]
This file is the input to the CMake build system for building software packages.
- **package.xml** [27]
Every ROS package must contain this file. This file defines properties about the package such as the package name, version numbers, authors, maintainers, and dependencies on other catkin packages.

3.2 Custom messages

The package uses three sensor messages (`sensor_msgs/CompressedImage`, `sensor_msgs/Image` and `sensor_msgs/CameraInfo`) [28–30], one geometry message (`geometry_msgs/Pose2D`) [31] and two custom messages. The structures of custom messages are defined in this subsection.

- `axis_camera/msgAxisVideo`

```
int32 fps
int32 width
int32 height
int32 compression
```

- `axis_camera/msgAxisPTZ`

```
int32 pan
int32 tilt
int32 zoom
```

3.3 Topics

This section describes briefly every topic the proposed system publishes/subscribes. The topics are divided in three categories according to function in the package. In category named image topics are the topics which work with the video streams from the Axis and the LadyBug cameras. In category named the command topics are all topics which contain commands controlling the camera. All topics which contains information about the camera are summarized in category named the info topics.

■ 3.3.1 Images topics

This subsection summarizes all of the image topics used in the package.

- The topic **axis214PTZ/compressed/compressed** is type of `sensor_msgs/CompressedImage`[28] message. This message contains a compressed JPEG image. The package uses this topic for original MJPEG streaming from Axis camera.

The following three topics are types of `sensor_msgs/Image` [29] message. This message contains an uncompressed image.

- The topic **axis214PTZ/Image** is created by ROS `image_transport` [21] package using its `republish` node. The `republish` listens on one base image topic (`axis214PTZ/compressed/compressed`) and republishes the images to `axis214PTZ/Image` topic. This topic is necessary for the calibration procedure using the cloud camera autocalibration package [12].
- The topic **/viz/pano/image** is the spherical image from LadyBug. The package needs this topic for its `mouse_teleop` node.
- The topic **/axis_visualizer** includes the spherical image extended of a red circle, which define the Axis camera look at point.

■ 3.3.2 Command topics

This subsection summarizes all of command topics, which contain information for controlling the camera position or the quality of video stream.

- The topic **axis214PTZ/VideoParamCmd** is type of `axis_camera/msgAxisVideo` message. The package uses this topic for changing the video parameters so that it can control the quality of video stream.
- The topic **axis214PTZ/Relative** is type of `axis_camera/msgAxisPTZ` message. The package uses this topic for changing the PTZ parameters so that it can comparatively control the camera position.
- The topic **axis214PTZ/Absolute** is type of `axis_camera/msgAxisPTZ` message. The package uses this topic for changing the PTZ parameters so that it can set up the absolute position of the camera.
- The topic **axis214PTZ/PointInImage** is type of `geometry_msgs/Pose2D` [31] message. The package uses this topic to determine the position of point in the image.

■ 3.3.3 Info topics

This subsection summarizes all of info topics, which contain information about the internal camera parameters and the current state of both video and PTZ camera parameters.

- The topic **axis214PTZ/CameraInfo** is type of `sensor_msgs/CameraInfo` message. This message defines meta information for the camera. This topic is necessary for the calibration procedure using the cloud camera autocalibration package [12].
- The topic **axis214PTZ/VideoParamActual** is type of `axis_camera/msgAxisVideo` message. The package shares all information about the published video after each video parameter change using this topic.
- The topic **axis214PTZ/Position** is type of `axis_camera/msgAxisPTZ` message. The package shares all information about the camera position after each PTZ parameter change using this topic.

3.4 Nodes

3.4.1 axis stream

- **subscribes:**

- axis214PTZ/VideoParamCmd

- **publishes:**

- axis214PTZ/compressed/compressed
- axis214PTZ/CameraInfo
- axis214PTZ/VideoParamActual

The **Axis stream node** subscribes the topic `axis214PTZ/VideoParamCmd`. If the node receives some message, it will do following steps:

1. It rewrites the video parameters from message to a http request and sends the request to the Axis http server. The response of Axis http server contains header and MJPEG data. These image data are published in the topic `axis214PTZ/compressed/compressed`.
2. It publishes the actual video parameters in the topic `axis214PTZ/VideoParamActual` and it also shared the actual video parameters with the ROS parameter server [16].
3. It publishes the camera information in the topic `axis214PTZ/CameraInfo`.

An example of the http request command:

```
http://192.168.0.90/axis-cgi/mjpg/video.cgi?req_fps=24&resolution=704x352&compression=0
```

A structure of http response includes these parts:

- header of html response – the first three lines
- label that defines the beginning of the image. It is the text `--myboundary\r\n`
- two parameters `Content-Type` and `Content-Length` that define type of image and length of one image
- data of one image

Last three items are periodically repeated until the camera produces images.

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=myboundary\r\n
\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
.
.
.
\r\n
```

3.4.2 axis ptz

■ subscribes:

- axis214PTZ/Relative
- axis214PTZ/Absolute
- axis214PTZ/PointInImage

■ publishes:

- axis214PTZ/Position
- tf (axis_214_PTZ)

The **Axis_ptz node** subscribes three topics `axis214PTZ/Relative`, `axis214PTZ/Absolute` and `axis214PTZ/PointInImage`. If the node receives some message, it will do following steps:

- Callback function for the topic `axis214PTZ/Relative`, see Figure 3.2:
 1. It sends the request for the actual values of PTZ parameters to the Axis http server.
 2. Then it counts next position using the received message and the response from the Axis http server.
 3. Then the node forms the http request for a change of position. It sends this request to the Axis http server.
 4. At the end it publishes the actual PTZ parameters in the topic `axis214PTZ/Position` and it also shared the actual PTZ parameters with the ROS parameter server [16].
- Callback function for the topic `axis214PTZ/Absolute`, see Figure 3.3:
 1. It rewrites the PTZ parameters from message to a http request and sends the request to the Axis http server.
 2. Then it publishes the actual PTZ parameters in the topic `axis214PTZ/Position` and it also shared the actual PTZ parameters with the ROS parameter server.
- Callback function for the topic `axis214PTZ/PointInImage`, see Figure 3.4:
 1. First, the node get the width and height of the published stream from the ROS parameter server.
 2. It rewrites the coordinate of the point from message to a http request and sends the request to the Axis http server.
 3. Then it publishes the actual PTZ parameters in the topic `axis214PTZ/Position` and it also shared the actual PTZ parameters with the ROS parameter server.

Examples of http request commands (the first command is the query for actual position; the second command is the request used by callback functions of topics `axis214PTZ/Relative` and `axis214PTZ/Absolute`; the third command is the request used by callback function of topic `axis214PTZ/PointInImage`):

```
http://192.168.0.90/axis-cgi/com/ptz.cgi?query=position
http://192.168.0.90/axis-cgi/com/ptz.cgi?pan=10&tilt=-15&zoom=3000
http://192.168.0.90/axis-cgi/com/ptz.cgi?imagewidth=704&imageheight=352&
camera=1&center=150,275
```

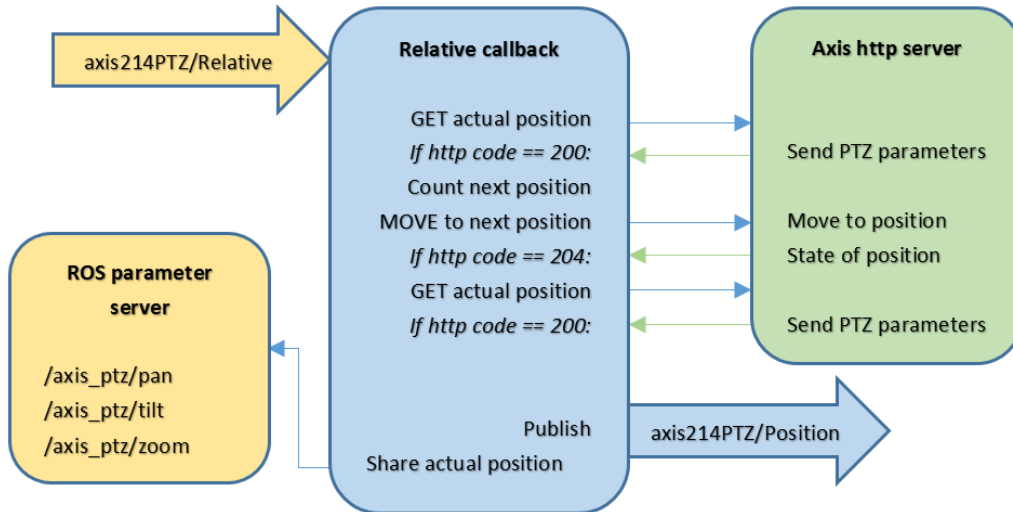


Figure 3.2. Diagram shows the communication during the processing the subscribed topic axis214PTZ/Relative by callback function.

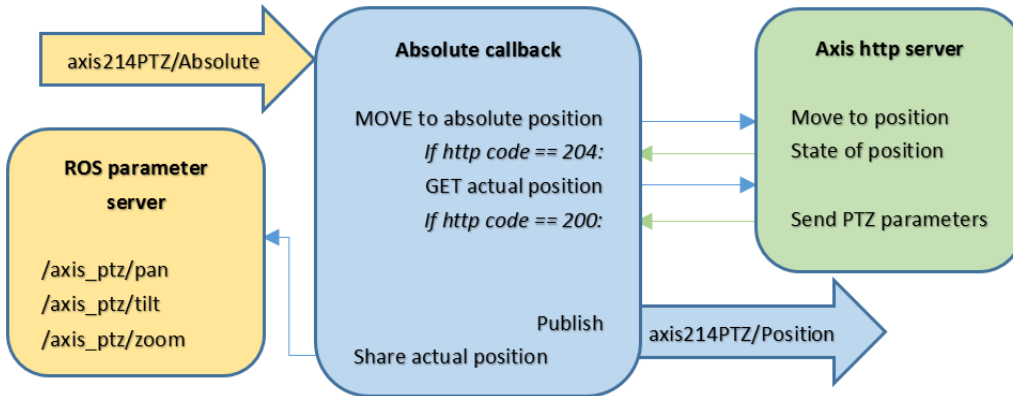


Figure 3.3. Diagram shows the communication during the processing the subscribed topic axis214PTZ/Absolute by callback function.

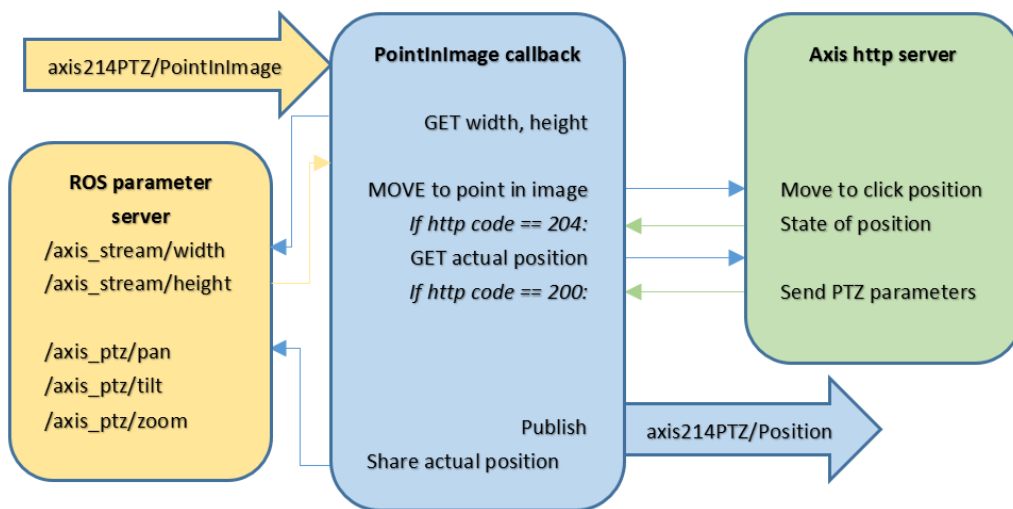


Figure 3.4. Diagram shows the communication during the processing the subscribed topic axis214PTZ/PointInImage by callback function.

■ 3.4.3 axis visualizer

■ subscribes:

- /viz/pano/image
- axis214PTZ/Position

■ publishes:

- /axis_visualizer

The node **axis_visualizer** counts the position of look at point in the image from the LadyBug camera. There is used the linear function that transforms the camera elevation and azimuth (tilt and pan) to the LadyBug image coordinates. The values of pan and tilt are subscribed from the topic `axis214PTZ/Position`. The image from LadyBug is subscribed from the topic `/viz/pano/image`. The position of look at point is represented by a red circle. The spherical image extended of the red circle is published in the topic `/axis_visualizer`.

■ 3.4.4 axis keyboard teleop

■ publishes:

- axis214PTZ/Relative
- axis214PTZ/VideoParamCmd

The node **keyboard_teleop** publishes the topics `axis214PTZ/VideoParamCmd` and `axis214PTZ/Relative`. This node converts keystrokes to required changes in the camera.

The values of the streaming video (fps, resolution and compression) is adjusted incrementally with keys (R, F, T, G, Y, H). Every change is sent in the topic `axis214PTZ/VideoParamCmd`.

Size of relative change of PTZ parameters depends on keystroke (A, D, W, S, Q, E). There is a possibility of basic motion or fast motion using keys combination `SHIFT +` one of the basic keys. Numbers (1 ~ 9) set the step size shift, higher value correspond to large step size. `SPACEBAR` sets the parameters for the home position. Every change is sent in the topic `axis214PTZ/VideoParamCmd`.

■ 3.4.5 axis mouse teleop

■ subscribes:

- axis214PTZ/Image
- /viz/pano/image
- axis214PTZ/Position

■ publishes:

- axis214PTZ/Absolute
- axis214PTZ/PointInImage

The node **mouse_teleop** allows to control the camera position by using mouse. It opens two openCV windows. The first one is video from Axis camera, the second one

is spherical image from LadyBug camera. These windows allow several events. By selecting events can imitate the style of operation publicly used applications. It is important to have three different types of action on the control position of the camera. That is why the click in combination with 2 other key buttons is used. Google maps uses right and left double click to zoom in and out and the map is movable by just moving with the mouse (this opportunity is not possible here).

In this work is used *click* for moving the camera as well as for zoom in and out when the combination click plus SHIFT and CTRL is applied. The right click stays free. It can easily happen, that the person, who uses the camera presses the button and camera turns the way he never intended to. This problem is solved by using double click (not just simple click) with the panorama pictures. Unlike with the Axis picture, where there was chosen simple click, because it does not cause any bigger problem when user presses the button several times by mistake. This solution seems to be simpler and clearer.

Chapter 4

User guide

The User guide assumes that user has basic experience with the ROS framework. The instructions were tested on Ubuntu 14.04 and ROS Indigo Igloo. The guide essentially describes installation procedure, setting the network parameters and setting up RViz. Moreover it includes section about troubleshooting, see Section 4.7.

4.1 Installation of ROS and import package

This package was written for ROS version Indigo and it was developed and tested for the operating system Ubuntu 14.04.

Install ROS Indigo¹⁾ as instructed on the ROS wiki [16]. Then insert the complete package into your working space²⁾ in ROS. For compiling all your ROS packages, open a terminal and type:

```
catkin_make
```

For compiling only this package, open a terminal and type:

```
catkin_make --pkg axis_camera
```

4.2 Network Setup

For the quickest start of the camera, do the following steps. Configure your network card to the network with address 192.168.0.120 and mask 255.255.255.0. The address can take the form 192.168.0.xxx, xxx can be any number but avoid 90 which is reserved for default address for all Axis cameras. An example of configuration you can see in Figure 4.1.

If the robot uses the network 192.168.0.0 and mask 255.255.255.0 for any other application at the same time, choose other network, but do not forget to change IP address in the camera. Always rewrite the hostname address in the camera nodes as well or do not start the nodes with default hostname.

4.3 Installation of driver for converter RJ-45 to USB

The Ubuntu operating system should install the standard drivers for the network card by itself. In case it does not start working automatically, follow the steps recommended by the manufacturer [32] and install the AXAGO ADE-SG converter manually.

- Download the driver from the manufacturer's website³⁾.
- Extract the compressed driver source file to your temporary directory.

¹⁾ <http://wiki.ros.org/indigo/Installation/Ubuntu>

²⁾ <http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>

³⁾ http://www.axago.cz/rs/_upload/File/ADE-SG-L.zip

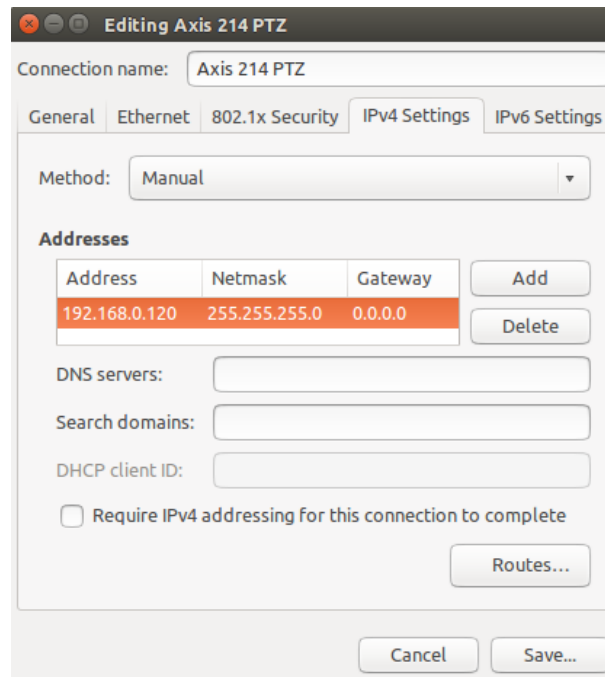


Figure 4.1. The example of network card configuration.

- Execute the following command to compile the driver (from temporary directory):

```
make
```

- Execute the following command to install the driver into your Linux (from temporary directory):

```
make install
```

- Configure your new USB network card as described in previous Section 4.2.

4.4 Basic camera setting

Connect the camera using an Ethernet cable. Some cameras may need a crossover cable for direct connection between PC and camera. Open your favorite web browser and type in the http row 192.168.0.90 (default IP address for Axis cameras). When first logging, you will probably need to create new user by entering your username and selecting password, or you will have to use the default user. Default user should be specified in the manual for individual cameras [13]. Old Axis camera models have following default user:

```
username: "root" password: "pass"
```

After logging into the camera, it is necessary to tick an option *Enable anonymous viewer login (no user name or password required)* and *Enable anonymous PTZ control login (no user name or password required)* in the camera setting. Each setting differs with the type of camera, but they all should enable access without the need for user authentication and password. For example, the setting for the Axis 214 PTZ camera is depicted at Figure 4.2.

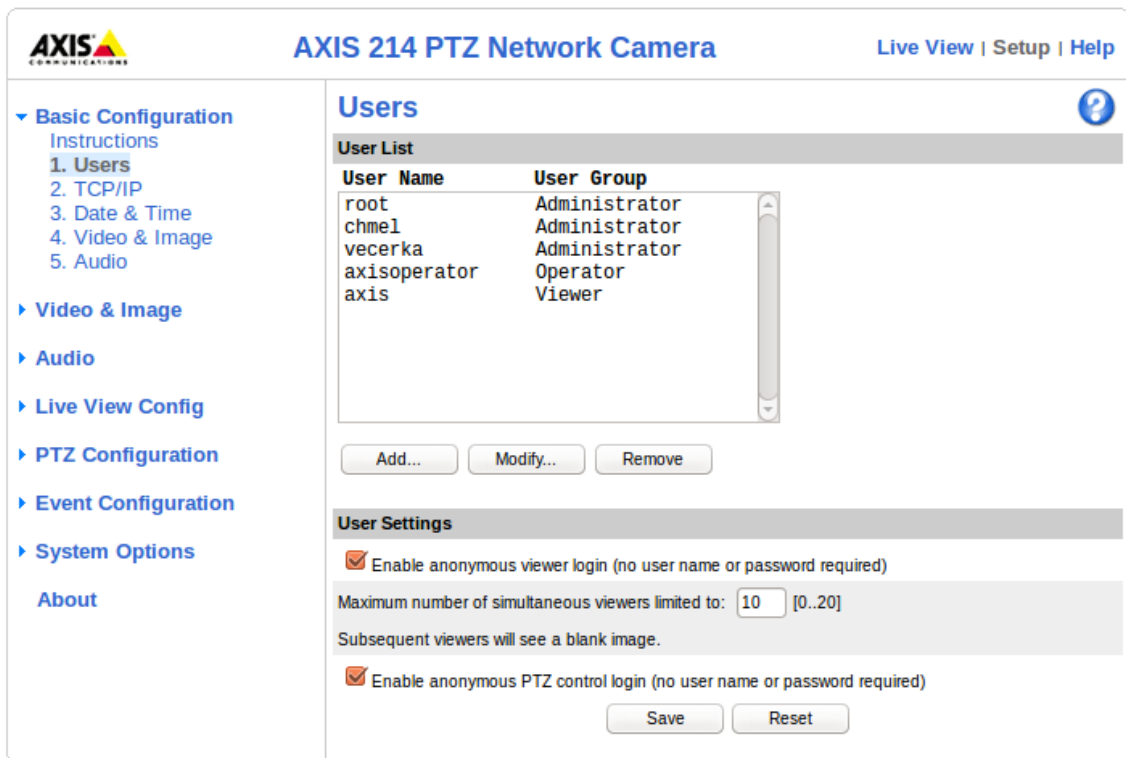


Figure 4.2. Setting anonymous mod for camera Axis 214 PTZ.

4.5 HW connection

For connecting the camera to the robot system, perform the following steps: First, connect the power supply and wait until the camera completes the initial calibration. Then connect the camera to the network card, which is connected to USB 2.0 (the network card does not work in USB 3.0). It is necessary to wait until the camera sets the IP address. Once the LEDs in the rear part of the camera lights on the light or blink green, the address is set.

4.6 Start and setting up RViz

The package produces several video streams and one *tf* topic that can be displayed by ROS program RViz [33]. You have to run the program RViz in new command line by this command.

```
roslaunch rviz rviz
```

After starting RViz click on the button *Add* in the part of program – *Displays* . Select item *Image* on the tab *By display type* for video topics or *Axes* for *tf* topic, see Figure 4.3. Then you select the type of topic and chose the name of topic that you want to display, see Figure 4.4.

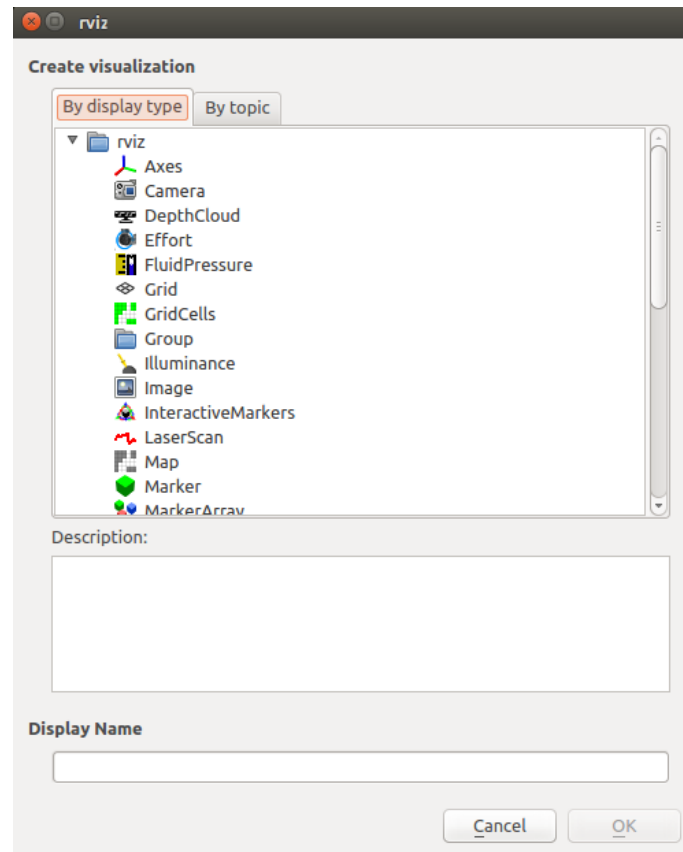


Figure 4.3. A window for creation of new visualization. The package produces topics which can be displayed by Image or Axes visualization.

4.7 Troubleshooting

The most frequent errors along with their solutions are listed in this subsection.

4.7.1 Freezing in standby mode

Freezing in standby mode happens when the camera server receives an invalid HTTP request. This status most often occurs after connecting the camera to the robot because it gets a wrong IP address. Do the following troubleshooting steps:

- Disconnect the power supply and the network cable. Wait for 5 seconds.
- Connect the power supply and wait 5 seconds.
- Connect the network cable and wait until the camera sets the IP address. Once the LEDs in the rear part of the camera put on the light or blink green, the address is set.
- Try to run the camera drivers.

4.7.2 Problem with the mouse teleop node

Two problems have been documented with the node `mouse_teleop`. ROScore sometimes does not want to register the subscribes. The solution is to start the node again or reboot the ROScore and try it again. The second problem is the freezing video stream which occurs when the wifi signal quality suddenly decreases. What often helps is to reduce quality of video stream which restarts the video stream. But in some cases it is needed to restart the node.

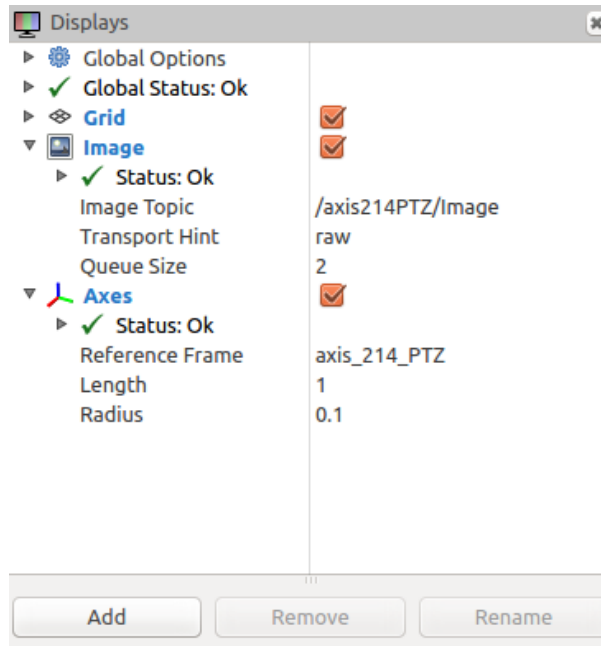


Figure 4.4. Displays – part of main screen RViz. Select Image Topic and Transport Hint to set up the topic that you want to see. There is chosen topic `/axis214PTZ/Image` with transport hint `raw`. Under the Axes part is selected Reference Frame `axis_214_PTZ`, which is Axis *tf* topic.

Chapter 5

Operator guide

The previous sections were focused on initial setup of the camera and the network card. The operator guide assumes that the whole system is made ready. This part assumes the least technical knowledge.

5.1 Summary for the quickest start

Write each command in a new command line.

```
roslaunch axis_camera axis_camera_driver.launch
roslaunch axis_camera axis_camera_driver_basestationWithMouse.launch
roslaunch rviz rviz
```

All events for controlling the camera are listed in Table 5.1.

5.2 Start ROScore

ROS core process [16] has to be started first, before any other ROS node. There are two ways how to run it. If you use the launch file, the roscore will start automatically. More info find in the following subsection 5.3. If you want to run some nodes separately and the ROScore is not running yet, run it with the command:

```
roscore
```

5.3 Launch the camera drivers at the robot

The package includes two main ROS nodes: `axis_stream` and `axis_ptz`. These nodes transfer orders and requests from an operator and other programs. They are after sent to the http server in the camera. You can run these nodes on the robot by launch file, write following command:

```
roslaunch axis_camera axis_camera_driver.launch
```

The **Axis_stream node** secures publishing a video stream as ROS sensor Compressed-Image message in the topic `axis214PTZ/compressed` in transport hint `compressed`. It responds to requests for changes to video parameters after launch. Further, it sends messages within the topic `axis214PTZ/VideoParamActual` with the current parameters of the video after the node starts and after every other change in the video parameters. At the same time, these parameters are shared with the ROS parameter server [16].

The **Axis_ptz node** secures all camera movements. It receives commands on absolute and relative changes in position. It also allows precise centering of the image to a point that is seen in the video. Further, it sends messages within the topic `axis214PTZ/Position` with the current position after each change. At the same time

EVENT	WHERE	EFFECT
together events for pan; tilt; zoom		
1 ~ 9	keyboard_teleop	sets step size (x)
SPACEBAR	keyboard_teleop	sets home position
double click	window Panorama image	to look at click position; zoom = 1
events for pan; tilt		
A	keyboard_teleop	pan - x°
SHIFT+A	keyboard_teleop	pan - 5*x°
D	keyboard_teleop	pan + x°
SHIFT+D	keyboard_teleop	pan + 5*x°
W	keyboard_teleop	tilt + x°
SHIFT+W	keyboard_teleop	tilt + 5*x°
S	keyboard_teleop	tilt - x°
SHIFT+S	keyboard_teleop	tilt - 5*x°
click	window Axis image	to look at click position
events for zoom		
Q	keyboard_teleop	zoom - 10*x units
E	keyboard_teleop	zoom + 10*x units
SHIFT+Q	keyboard_teleop	zoom - 100*x units
SHIFT+E	keyboard_teleop	zoom + 100*x units
CTRL+click	window Axis image	zoom - 5000 units
SHIFT+click	window Axis image	zoom + 2000 units
events for fps; resolution; compression		
R	keyboard_teleop	fps + 3 units
F	keyboard_teleop	fps - 3 units
T	keyboard_teleop	higher resolution
G	keyboard_teleop	lower resolution
Y	keyboard_teleop	compression + 10 units
H	keyboard_teleop	compression - 10 units

Table 5.1. Summary of the available camera controls. WHERE denotes the application context: keyboard_teleop means that the terminal window with running the keyboard_teleop must be active. Similarly for the graphical windows.

this position is transferred to the `tf` message and published. The position parameters are also shared with the ROS parameter server.

Sometimes you may need to modify the default parameters for the nodes, see Table 5.2, before the node starts. For example, when the WIFI signal is poor or when there are demanding applications running in the robot. There are two ways of doing it. The first is to adjust the parameters in the file `axis_camera_driver.launch`, which you will find here:

```
"ROS workspace"/src/axis_camera/launch/axis_camera_driver.launch
```

The second option is to run the nodes separately by following commands, where the parameters in brackets [...] are optional. A possible range of these parameters is stated in Table 5.2.

```
roslaunch axis_camera axis.py [_hostname:=<value> _frame_id:=<value>
  _width:=<value> _height:=<value> _compression:=<value> _fps:=<value>
  _timeoutSeconds:=<value> _quiet:=<value>]
```

```
roslaunch axis_camera axis_PTZ.py [_hostname:=<value>
_timeoutSeconds:=<value> _quiet:=<value>]
```

Parameter	Range	Default
hostname	any free address	192.168.0.90
frame_id	any free name	Axis_stream
width	704, 352 or 176	704 px
height	480/576, 240/288 or 120/144	576 px
compression	0 ~ 100	0
fps	1 ~ 30	24 fps
timeoutSeconds	Any positive values	5.0 s
quiet	True or False	False

Table 5.2. The Axis drivers parameters, their range and default value. The `hostname` is value of IP address that you set up in the camera. The `frame_id` parameter naming the video stream. The video parameters (`width`, `height`, `compression` and `fps`) affects the size of bandwidth. The `timeoutSeconds` parameter is the time when the program waits for a response from the server. The `quiet` parameter enables or disables typing messages on the terminal.

5.4 Launch the nodes at the base station

The package consists two parts. The previous subsection 5.3 described the first part. Nodes running at the base station belongs to the second part of the package. More information about that is in this section.

The node **keyboard_teleop** allows you to control the video parameters and camera position by using a keyboard. The values of the streaming video (fps, resolution and compression) are adjusted incrementally with keys (R, F, T, G, Y, H). Consequences of the keystrokes are summarized in Table 5.3.

The position of the camera is changed relatively to the actual position. The effect of the relative change can be set by the keys (A, D, W, S, Q, E). There is a possibility of basic motion or fast motion using the key combination SHIFT + one of the basic keys. Numbers (1 ~ 9) set the step size shift, higher value corresponds to a larger step size. The SPACEBAR sets the camera position to the home position. The effects of the keystrokes are summarized in Table 5.3.

The node **mouse_teleop** allows you to control the camera position by using mouse. This node opens two windows. One of them is the stream from Axis camera and the second is panoramic image from Ladybug, see Figure 5.1. The mode offers four options to control the camera position by mouse:

- To set an approximate position press the left mouse button twice in quick (double click) on the Panorama image window.
- To set up an exact position press the left mouse button once (click) on the Axis image window.
- To zoom the Axis image center press the left mouse button once with SHIFT key (SHIFT + click) on the Axis image window. It zooms about 1/5 of the zoom range.
- To zoom out the Axis image centre press the left mouse button once with CTRL key (CTRL + click) on the Axis image window. It zooms out about 1/2 of the zoom range.

keystroke	effect
R	increase the value of fps by 3 units
F	reduce the value of fps by 3 units
T	increase the resolution
G	reduce the resolution
Y	increase the value of compression by 10 units
H	reduce the value of compression by 10 units
1 ~ 9	set the value of step size (x) on the corresponding number
A	turn by $-x^\circ$ in pan axis
D	turn by x° in pan axis
W	turn by x° in tilt axis
S	turn by $-x^\circ$ in tilt axis
Q	zoom out by $10*x$ units
E	zoom in by $10*x$ units
SHIFT+A	turn by $-5*x^\circ$ in pan axis
SHIFT+D	turn by $5*x^\circ$ in pan axis
SHIFT+W	turn by $5*x^\circ$ in tilt axis
SHIFT+S	turn by $-5*x^\circ$ in tilt axis
SHIFT+Q	zoom out by $100*x$ units
SHIFT+E	zoom in by $100*x$ units
SPACEBAR	setting home position (pan=0, tilt=0, zoom=1)

Table 5.3. The effect of keystroke.

The node **axis_stream_republish** converts topic `axis214PTZ/Image` from a **compressed** quality to a **raw** quality, which is required for example for camera calibration.

The node **axis_visualizer** publishes the topic `/axis_visualizer`. It includes the spherical image extended of a red circle, which define what the Axis camera watches.

You can run the previous nodes at once typing one of the following commands:

```
roslaunch axis_camera axis_camera_driver_basestationWithMouse.launch
```

```
roslaunch axis_camera axis_camera_driver_basestation.launch
```

The first command launches all the four described nodes. The second command launches all nodes except the `mouse_teleop` node. It is useful when there is no need to control constantly the camera using mouse navigation. Sometimes, it might be advantageous to run nodes separately. It is possible by following commands:

```
roslaunch axis_camera keyboard_teleop_axis_cam
```

```
roslaunch axis_camera mouse_teleop.py
```

```
roslaunch image_transport republish compressed in:=axis214PTZ/Image raw
out:=axis214PTZ/Image
```

```
roslaunch axis_camera axis_visualizer
```



Figure 5.1. Mouse teleop windows.

Chapter 6

Experiments

This chapter contains the summary of experiments. We tested optical properties, amount of data transmitted, controlling camera by mouse and video shaking induced by the robot vibrations. We compared two suggested camera positions and it was verified procedure of camera calibration. Other image and video materials are stored in the project folder, which is described in Appendix A.

6.1 Optical zoom capability test

AXIS 214 PTZ has an 18x optical zoom. The experiment shows the optical zoom capabilities. The robot is 35 meters away from the observed object, see Figures 6.1 and 6.2. The images from Axis camera for several optical zoom values are summarized in Figure 6.3.



Figure 6.1. Photo documentation of the experiment – the optical zoom capability.



Figure 6.2. Spherical image from Ladybug – operator lacks an optical zoom function. Objects, which are close, may be partially enlarged using a digital zoom. However this is not enough.

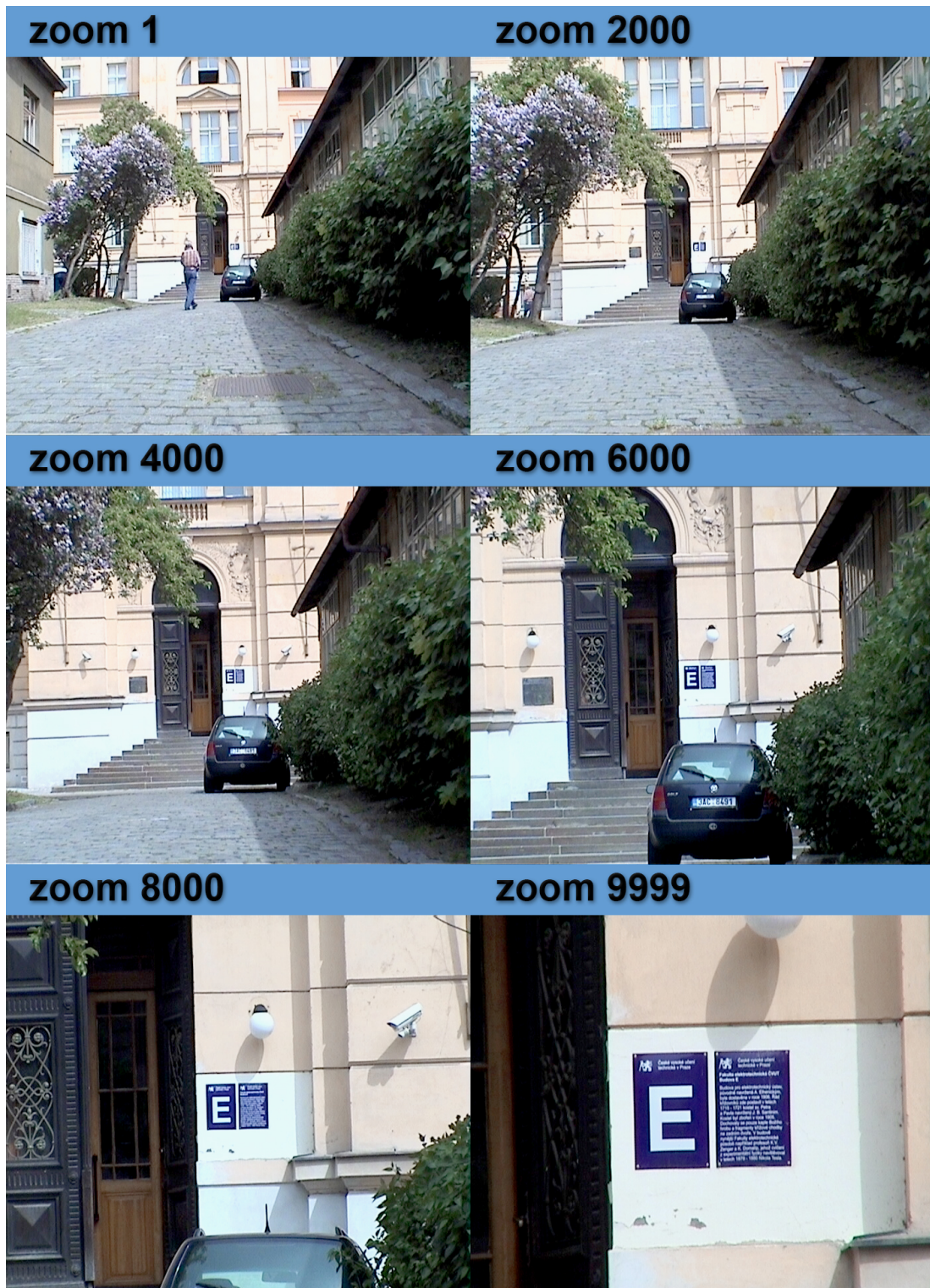


Figure 6.3. Axis 214 PTZ – range of possible use of the optical zoom. The size of the zoom is expressed in the camera units.

Optical zoom can be controlled by events which are listed in Table 6.1. The fastest uses of optical zoom function is in the mouse_teleop node by clicking on the Axis image window with pressed SHIFT (zoom in) or CTRL (zoom out).

EVENT	WHERE	EFFECT
1 ~ 9	keyboard_teleop	sets step size (x)
SPACEBAR	keyboard_teleop	sets home position
double click	window Pamorama image	to look at click position; zoom = 1
Q	keyboard_teleop	zoom - 10*x units
E	keyboard_teleop	zoom + 10*x units
SHIFT+Q	keyboard_teleop	zoom - 100*x units
SHIFT+E	keyboard_teleop	zoom + 100*x units
CTRL+click	window Axis image	zoom - 5000 units
SHIFT+click	window Axis image	zoom + 2000 units

Table 6.1. Summary of the available camera controls for setting the zoom parameter.

6.2 Amount of data transmitted

The camera drivers allow to change the amount of transferred data at the expense of video quality. The experiment shows the amount of the data that must be transferred depending on the video parameters (resolution, fps and compression). The effect of compression is demonstrated in the Figure 6.4. The topic `axis214PTZ/compressed/compressed` was used for measuring. This topic is the only one that is transmitted between the robot and the base station. Selected values of bit rate are summarized in the following Tables 6.2, 6.3 and 6.4.

The testing have shown that it is inappropriate to use the topic `axis214PTZ/Image` for applications running on the robot. This topic is created from the compressed version of stream by the republish node which is running on the base station. While video data are much bigger the quality is the same, see Table 6.5.



Figure 6.4. Four selected levels of compression show how the compression affects the visual quality of the image. The values up to level 30 are almost indistinguishable. The values between 30 and 60 are still applicable but the reduction in quality is visible. The values greater than 60 should not be used.

704x576	1 fps	15 fps	30 fps
0	239	2 190	3 440
30	118	1 350	2 340
100	43	540	950

Table 6.2. Amount of data transmitted in high resolution (704 x 576) are dependent on fps and compression. The values are given in kBit/s.

352x288	1 fps	15 fps	30 fps
0	87	1 020	2 040
30	43	530	820
100	8	104	180

Table 6.3. Amount of data transmitted in medium resolution (352 x 288) are dependent on fps and compression. The values are given in kBit/s.

176x144	1 fps	15 fps	30 fps
0	26	250	400
30	7	65	180
100	1.3	19	30

Table 6.4. Amount of data transmitted in low resolution (176 x 144) are dependent on fps and compression. The values are given in kBit/s.

704x576 raw	1 fps	15 fps	30 fps
0	1.24	18.0	30.2
30	1.22	18.2	30.6
100	1.36	18.3	30.5

Table 6.5. Raw image topic `axis214PTZ/Image` – Amount of data transmitted in high resolution (704 x 576). The Values are about 10 times larger than compressed versions of the video. The concrete measured values are described in Table 6.5. The values are given in MBit/s.

Events changing the video parameters (fps, resolution and compression) are listed in Table 6.6.

EVENT	WHERE	EFFECT
R	keyboard_teleop	fps + 3 units
F	keyboard_teleop	fps – 3 units
T	keyboard_teleop	higher resolution
G	keyboard_teleop	lower resolution
Y	keyboard_teleop	compression + 10 units
H	keyboard_teleop	compression – 10 units

Table 6.6. Summary of the available camera controls affecting the video quality.

6.3 Comparison of two suggested camera positions

The experiment is focused on comparing two proposed camera positions for mounting on the robot. One important function of the camera is to allow the operator to observe the nearby surface around the robot. This is important when the robot moves in demanding terrain. Furthermore, the camera can work with a robotic arm during collecting the sample.

Several things were placed around the robot, see Figure 6.5. Experiment shows differences of three pictures from each proposed camera position (see Figure 6.6): view downward in front of the robot; view downward in front of the continuous track; limits on the maximum optical zoom.

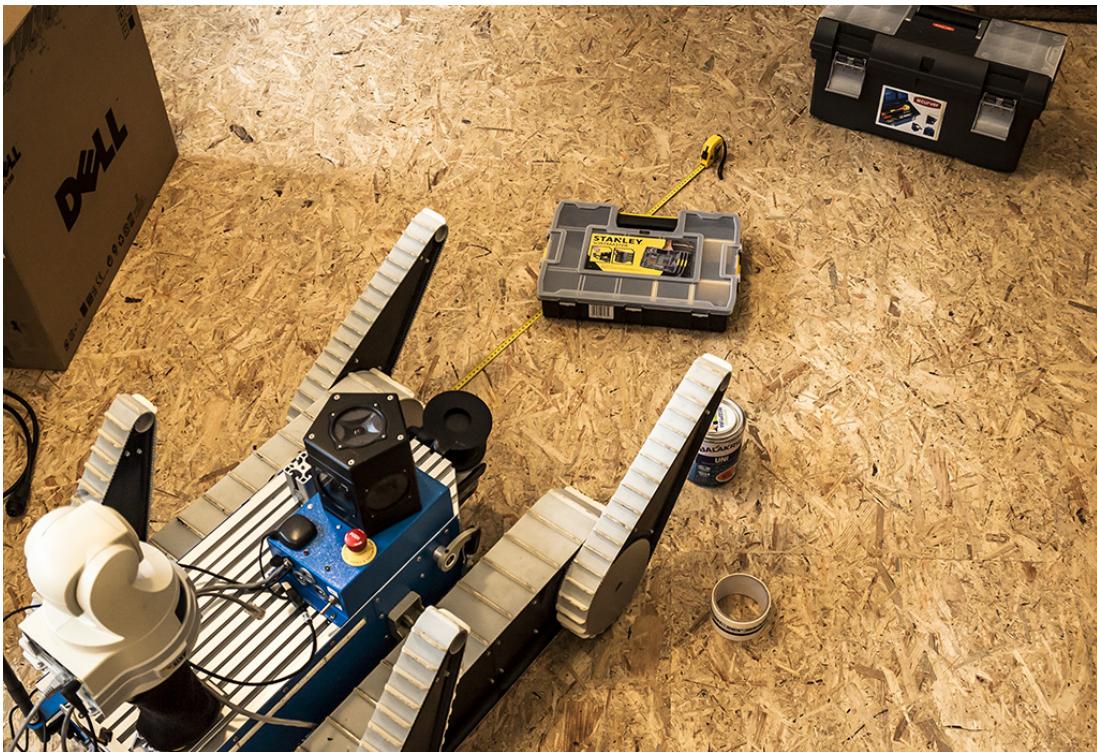


Figure 6.5. Photo documentation of the experiment – comparison of two suggested camera positions. Several things are placed around the robot: tape measure; smaller suitcase (35-75 cm in front of the robot); tin (25 cm in front of the right continuous track); white tape (15 cm next to the right continuous track); bigger suitcase (120 cm in front of the robot).

Each position has certain its advantages and disadvantages. The camera position at the rear of the robot cannot see well on the ground in front of the robot. The Ladybug camera takes up a lot of space in the image from the Axis. However, the camera has a good view on the area next to the continuous tracks. Therefore, this place is suitable for collecting the samples. Moreover this position allows to use the maximum zoom range nearby the continuous tracks. The camera position in front of the robot has better visibility of the immediate neighboring areas in front of the robot. Therefore, the frontal position is convenient for collecting samples by the robotic arm. The closest distance where the camera can see is 35 cm and it is when the camera is placed on the 20 cm tall stand. However, the nearest distance usable with the maximum zoom range is 87 cm far. The project package is written for mounting the camera at the rear of the robot. This position is generally more usable.



Figure 6.6. The left column shows images from the position at the rear of the robot. The right column shows images from the position in front of the robot.

6.4 Camera calibration

The goal of the experiment is a verification of applicability of calibration procedure for computing an initial external calibration. Internal camera parameters were received from ROS camera driver for Axis network cameras [23].

The calibration procedure needs to prepare a suitable scene for correct operation. Both the image from the calibrated camera and the data from the laser scanner have to see the same objects. In addition, objects must be chosen so that their key points (especially corners) can be simply identified. The Calibration procedure needs to mark

at least 4 (ideally more) key points in the camera image and the corresponding points in point cloud from the laser scanner. Key points should cover the entire space in front of the camera and they should not lie in one plane. This condition can be hardly fulfilled because the camera AXIS 214 PTZ is not much a widescreen camera compared to the Ladybug (the camera for which was developed the calibration procedure). If key points are in one plane, one of the calibrated parameters will be calibrated wrong. Therefore, there have been used several different calibration scenes to reduced this problem, an example of one of them is shown in Figure 6.7. The calibration results from the previous example can be found in Table 6.7.

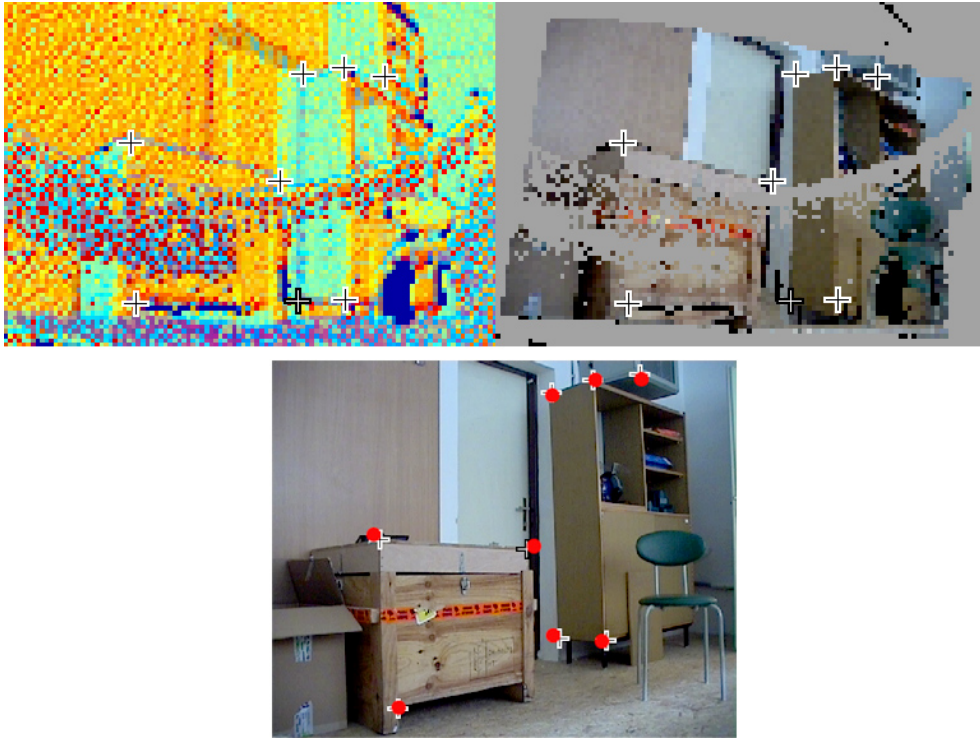


Figure 6.7. The calibration scene with furniture – a point cloud from a laser scanner is the top left image. Next image on right is the same point cloud but its points are colored by data obtained from the calibrated camera. Below is an image from the Axis 214 PTZ camera which was used for the camera calibration.

Parameter	Result	Manual measurement
X	-0.214 m	-0.290 m
Y	-0.044 m	-0.060 m
Z	0.455 m	0.435 m
Yaw	-1.595°	0°
Pitch	0.006°	0°
Roll	-1.483°	0°

Table 6.7. Results of the calibration scene with furniture (reprojection error: 10.1344). Values are relative to tf frame named `base_link`. There is visible a greater deviation in the parameter X compared to the manual measurement. The values of Euler angles (Yaw and Roll) are counted wrong in all calibration scenes. It failed to determine what caused it. That is the reason why the initial values of Euler angles are set manually.

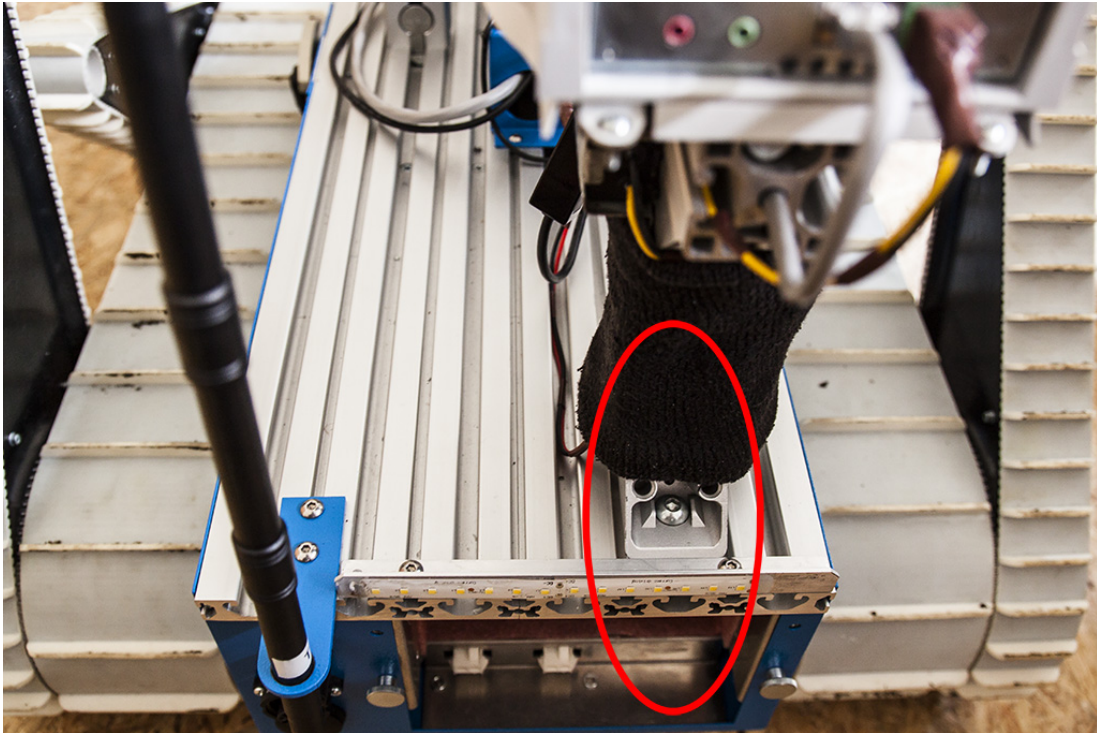


Figure 6.8. Definition of the fixed place of the camera stand – the red ellipse highlights the mounting place for the stand with camera. The stand is mounted to the second large slot from the right. Moreover, it touches the aluminum profile which holds the LEDs.

The camera must be placed in the same position as for the calibration. Therefore, the camera is placed on the stand which has fixed position on the robot, see Figure 6.8.

6.5 Controlling camera by mouse

One possible control of camera position is to use the node `mouse_teleop`. When the operator notices something suspect in the panoramic image from Ladybug camera, he might need to display it in higher resolution. Therefore he marks the look at point of interest by clicking on it. PTZ camera is then rotated so it shows the marked place in the center of its image. In fact, there are some errors because the PTZ camera and the Ladybug camera have some gap between them. It is about 43 centimetres. The goal of the experiment is to verify whether the error is significant.

Because the indication of the depth is not known, the coordinate systems Axis 214 PTZ and LadyBug may have the same position. This simplified model gets by with elevation and azimuth (tilt and pan). Practical application shows that it can be used the simplified model and there is no need to correct the error, see picture 6.9. A better idea about the experiment is possible to get by prepared screen recordings filmed during the experiments. The recordings are saved on the attached CD, see Appendix A.

6.6 Video shaking induced by the robot vibrations

The goal of the experiment is to verify the influence of vibrations during movement of the robot in a real environment. Effect of vibrations was tested with different optical zoom. Real environment was formed by cobblestones.



Figure 6.9. An example shows the scene where the point of interest is 3 m far. The enlargement part of the picture shows that the error between the wanted and the real position is approximately 7 pixels. The error is greater for objects near the robot. But the concept of navigation used by *mouse_telop* is to click on the panoramic image for receiving a widescreen image from PTZ camera. The widescreen image is not as susceptible to the error as an enlarged image. Then it is already possible precisely tuning position by image from Axis camera. Moreover the widescreen image is good for operator orientation.

It is not likely that the camera will be used with great value of zoom during the robot movement. Rather the robot arrives into the unknown environment and stops to explore the interesting objects. Nevertheless, it is good to know how much is the image shaking when the camera is used for navigation in difficult terrain.

Shaking video depends on a lot of factors e.g. kind of surface, the distance of observed things, robot speed and others. Generally the video is satisfactory up to 4,000 units of zoom in tested environment. Higher values are unuseable. A better idea about the experiment is possible to get by prepared screen recordings filmed during the experiments. The recordings are saved on the attached CD, see Appendix A.

Chapter 7

Conclusion

The goal of the thesis was the integration of IP PTZ camera onto the mobile rescue robot. For this purpose, the camera model Axis 214 PTZ was selected. It has not been tested for outdoor use (not waterproof), but it is lightweight (1110g) and cheaper (cost about 1000 EUR) compared to other outdoor cameras. The camera has an 18/times optical zoom. The minimum viewing angle is 2.7° in the horizontal level that corresponds to maximum resolution of $261px/1^\circ$.

The camera was placed on the stand at the rear of the robot so it does not hinder other devices and has a good visibility. There is only one particular space which the camera cannot see the ground around the robot well. It is the part which is directly in front of the robot, however both front flippers are clearly visible. Moreover, construction of the stand allows to quickly changed the position of the camera on the robot. The power supply is resolved by an external DC-DC converter and the data communication between the camera and robot is resolved by an external network card. Both accessories are attached to the stand. The whole solution including the camera weighs approximately 2.6 kg.

The project implements two main ROS nodes which communicate with the camera using the HTTP protocol. The first node publishes video stream with respect to the required quality as standard ROS topic (`sensor_msgs/CompressedImage`). The second node rotates the camera to a desired position. It is possible in two different ways: direct control angle; indirect control angle by shift the position of look at point in the `mouse_teleop` node. The size of the zoom can be set absolutely or relatively in units of the camera. The camera can be controled by several defined keys on a keyboard or mouse using the open windows with the video streams. The control is intuitive and it is designed by using the concept of zoom used in other applications (web maps, graphics programs). A current position of PTZ camera is painted on panoramic image as a red circle so that the operator exactly knows where the PTZ camera looks. Moreover the camera broadcasts its location and orientation in `tf` message.

The project includes two manuals. The first – *User guide* requires a basic experience with ROS framework. It is designed for initial setup and connection of PTZ camera into the robotic system. The second – *Operator guide* is intended for operators who will use only the functional camera. The *Operator guide* provides only the information necessary for controlling the camera.

The final solution was subjected to several tests that showed possibilities and limitations of the camera. The range of optical zoom is sufficient so for example it is possible to read a text on particular object which was hardly detected in image from Ladybug. Pan and tilt speed is satisfactory but the zoom speed could be faster. Image shaking while driving is noticeable a lot even at wide angle of view. But it is common for a camera that is not from one compact part. Unfortunately one practical application of PTZ camera (*look-at-point*) failed to complete. Controlling the camera by mouse was documented by several videos.

The thesis meets its the goals. The main control nodes are easily rewritable if manufacturer Axis comes up with a more preferred model of IP PTZ camera. Main result is that the robot gained the ability to display distant objects in high resolution. This feature can be now used along with other applications that can be implemented in the robotic system. For example, automatic storage of objects of interest, along with information about the position or it can help to collect samples by the robotic arm.

References

- [1] Kruijff et al.: *Designing, developing, and deploying system to support human-robot teams in disaster response*, In *Advanced Robotics*, vol 28, issue 23.
- [2] Tomáš Svoboda et al.: *DR 2.3.3: Bi-directional cooperation of low-level vision modules and higher level control*, Deliverable of project NIFTi.
- [3] Karel Zimmermann, Tomáš Svoboda, Ondřej Drbohlav, Tomáš Pajdla, Václav Hlaváč. *Proposal of camera sub-system for NIFTi project*, June 2010.
- [4] Jakub Chmel, Tomáš Svoboda. *ROS nodes for PTZ network cameras*, Individual project, CTU FEE 2015.
- [5] *Picture Vivotek PZ7131*. <http://www.networkwebcams.com/media/catalog...>, visited 2015-05-11.
- [6] *Picture Axis 215 PTZ*. http://www.techtradeoutlet.com/ebayPictures/Cameras/Axis_215.jpg, visited 2015-05-11.
- [7] *Picture Bosch MIC IP starlight 7000 HD*. http://geniusvision.net/test/images/MIC_IP_starlight_7000_HD.jpg, visited 2015-05-11.
- [8] *Picture Axis 214 PTZ*. <http://www.rimaelektronik.com/images/stories/...>, visited 2015-05-11.
- [9] *Picture Vivotek PZ8111/21*. <http://download.vivotek.com/downloadfile/downloads/productimages/pz81x1.png>, visited 2015-05-11.
- [10] *Lens calculator for AXIS 214 PTZ*. <http://www.axis.com/global/en/tools/lens-calculator>, visited 2015-05-06.
- [11] *Datasheet – AXIS 214 PTZ Network Camera*. http://www.axis.com/files/datasheet/ds_214ptz_33006_en_0809_lo.pdf, visited 2015-05-06.
- [12] Jan Brabec. *Automated camera calibration from laser scanning data in natural environments*, Bachelor thesis, CTU FEE 2014.
- [13] *User’s Manual – AXIS 214 PTZ Network Camera*. http://www.axis.com/files/manuals/um_214_37546_en_0911.pdf, visited 2015-05-07.
- [14] *Datasheet – Orion IP67 24/12 DC-DC converter*. <http://www.victronenergy.com/upload/documents/Datasheet-Orion-IP67-24-12-DC-DC-converters-EN.pdf>, visited 2015-02-05.
- [15] *Picture Axago ADE-SG*. <https://i.alza.cz/ImgW.ashx?fd=f4&cd=AG200a64>, visited 2015-02-05.
- [16] *Robot Operating System*. <http://wiki.ros.org/>, visited 2015-05-13.

-
- [17] *Python Documentation – 16.2.: Higher-level threading interface*
<https://docs.python.org/2/library/threading.html>, visited 2015-05-19.
- [18] *Python Documentation – 20.6.: extensible library for opening URLs*
<https://docs.python.org/2/library/urllib2.html>, visited 2015-05-19.
- [19] *OpenCV 2.4.11.0 documentation*
<http://docs.opencv.org/>, visited 2015-05-19.
- [20] *cv_bridge 0.1.0 documentation*
http://docs.ros.org/hydro/api/cv_bridge/html/python/, visited 2015-05-19.
- [21] *image_transport documentation*
http://wiki.ros.org/image_transport, visited 2015-05-19.
- [22] *tf documentation*
<http://wiki.ros.org/tf>, visited 2015-05-19.
- [23] *ROS camera driver for Axis network cameras.*
http://wiki.ros.org/axis_camera, visited 2015-05-16.
- [24] *VAPIX VERSION 3, Video Streaming API*
http://www.axis.com/files/manuals/vapix_video_streaming_52937_en_1307.pdf,
visited 2015-05-19.
- [25] *VAPIX VERSION 3, Pan Tilt Zoom API*
http://www.axis.com/files/manuals/vapix_ptz_52933_en_1307.pdf, visited 2015-05-19.
- [26] *CMakeLists.txt documentation*
<http://wiki.ros.org/catkin/CMakeLists.txt>, visited 2015-05-19.
- [27] *package.xml documentation*
<http://wiki.ros.org/catkin/package.xml>, visited 2015-05-19.
- [28] *sensor_msgs/CompressedImage message documentation*
http://docs.ros.org/indigo/api/sensor_msgs/html/msg/CompressedImage.html,
visited 2015-05-19.
- [29] *sensor_msgs/Image message documentation*
http://docs.ros.org/indigo/api/sensor_msgs/html/msg/Image.html, visited 2015-05-19.
- [30] *sensor_msgs/CameraInfo message documentation*
http://docs.ros.org/indigo/api/sensor_msgs/html/msg/CameraInfo.html, visited 2015-05-19.
- [31] *geometry_msgs/Pose2Dmessage documentation*
http://docs.ros.org/indigo/api/geometry_msgs/html/msg/Pose2D.html, visited 2015-05-19.
- [32] *Axago – Complete Computer Comfort, Drivers and firmware.*
<http://www.axago.cz/cz/servis-a-podpora/ovladace-a-firmware.html>, visited 2015-02-05.
- [33] *Rviz – user guide.*
<http://wiki.ros.org/rviz/UserGuide>, visited 2015-05-13.



Appendix A

Enclosed CD

- **[ROS_package]**

Contains all source codes. Functional Ros package is stored in directory [axis_camera]. Unfinished codes (especially look-at-point node) are stored in directory [unfinished].
- **[Videos]**

Contains several videos from experiments.
- **[Thesis]**

Contains the sources for this thesis.
- **thesis.pdf**

PDF version of this thesis.