HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Electronics, Communications and Automation
Department of Automation and Systems Technology

LULEÅ
UNIVERSITY
OF TECHNOLOGY

Xi Chen

Effective Localization and Tracking Utilizing Stereo Vision for Planetary Micro-Rover ROSA

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology

Espoo, August 10.2009

Supervisors:

Professor Aarne Halme
Helsinki University of Technology

Professor Kalevi Hyyppä
Luleå University of Technology

Instructor:

Pekka Forsman
Helsinki University of Technology

Preface

The research for this Master Thesis was done at the Helsinki University of Technology's Automation Technology Laboratory during the years 2008-2009. First of all, I would like to sincerely thank my supervisors Professor Dr. Aarne Halme and Professor Dr. Kalevi Hyyppä for giving me the opportunity to undertake this thesis, providing me valuable guidance and constructive comments. To my dedicated instructor Pekka Forsman, thank you very much for spending so much time for improving my thesis. I wish to thank Mikko Elomaa who provides an excellent and timely support for the hardware of the robot. Many thanks to Tomi Ylikorpi who proposes the thesis topic and gives suggestions. I would like to thank Janne Paanajärvi, who gives me a lot of guidance from the theory to the implementation of the stereo vision. Many thanks to Vicky Wong who explain the previous work to me warmly. I would like to express my gratitude to the staff of the AS department – Antti Maula and Sami Terho who kindly to help me . I want to thank my classmates who bring me a lot of wonderful time during the last two years.

Finally I desire to express my deepest gratitude to my parents who encourage me to join this master program and give me support and motivation for this entire endeavor.

Espoo, August 10, 2009

Xi Chen

Helsinki University of Technology Abstract of the Master's Thesis

Author:	Xi Chen		
Title of the thesis:	Effective Localization and Tracking Utilizing Stereo Vision for Planetary Micro-Rover ROSA		
Date:	August 10, 2009	**Number of pages:**	110
Faculty:	Faculty of Electronics, Communications and Automation		
Department:	Automation and Systems Technology		
Program:	Master's Degree Programme in Space Science and Technology		
Professorship:	Automation Technology (Aut-84)		
Supervisors:	Professor Aarne Halme (TKK) Professor Kalevi Hyyppä (LTU)		
Instructor:	Pekaa Forsman (TKK)		

Planetary exploration mainly depends on orbiters, rovers and landers. Especially the rovers provide a large covering area and close-up investigations about the planets. ROSA is a prototype tracked rover for the Mars exploration developed in the project sponsored by European Space Agency (ESA). The localization of ROSA is relied on stereo vision from the stereo cameras standing on top of the lander. However, due to the fixed position of the lander and the stereo cameras as well as the small field of view (FOV) of the cameras, the area where ROSA can be localized is very limited. Moreover, the stereo cameras are located too far to observe the geographical features in detail and to find out the interesting samples. In order to enlarge the localization area, a new system is designed and implemented; meanwhile it provides close-up observation of geographical features and takes images of the interesting samples. A pan-tilt unit (PTU) is installed to hold the bar of the stereo cameras. It can rotate around the pan and tilt axes to enable the cameras to cover larger areas. To ensure that the rover stays in the images of both cameras when the localization of rover is required and also that the cameras can autonomously follow the rover, a tracking method is implemented. It is related with the active marker, utilizing the background subtraction and blob connection to obtain the accurate LED position in the images, which helps to infer the rover position in the images. This tracking procedure is thoroughly consistent with the localization method. On the basis of the rover position in the images, the PTU is controlled to move the cameras to a proper orientation to ensure that the rover always appears in both images. A model for localization is built, which provides accurate localization information with less computation. Another camera is installed in the front panel of ROSA to take images and send them back to the ground. To control the mobility of ROSA in a correct and easy way, a new program with graphic user interface is developed for the ground station. Furthermore, a tracking program with video feedback has been developed.

Keywords: tracking, rover, ROSA, stereo vision, PTU, GUI, localization, marker, planetary exploration

Contents

List of Tables

List of Figures

Symbols and Abbreviations

A	a matrix in least square method
b	a vector in least square method
d	dimension of the image sensor
f	focal length
fx	focal length in horizontal pixels
fy	focal length in vertical pixels
h	radius of the window
I	identity matrix
g_i	minus of the derivative of the kernel
kc, k_1, k_2, k_3, $k_4$and k_5	distortion coefficients
m	horizontal FOV
M_{00}	zeroth moment
M_{10}, M_{01}	first moments
n	vertical FOV
o	principle point
(ox, oy)	principle point coordinates
O	projection center
Obxyz	basic frame
$O_l$xyz	left camera reference frames
Orxyz	right camera reference frames
Opxyz	PTU initial frame
OXYZ	camera frame
OZ	optical axis
P	a point in space
P_b	coordinates in the base frame
P_c	reconstructed coordinates of a point in the left camera frame; coordinates of a point in the camera reference frame
P_{c1}, P_{c2}	the coordinates of the same point in different camera frames.

P_p	coordinates of a point in the PTU frame
P_w	coordinates of a point in the world reference frame
P_{init}	a point in initial PTU frame; the coordinates of the same point in the initial PTU frame
P_{move}	a point in the final frame of PTU
P_{temp}	a point in new frame of PTU
p	projection of point P on the image plane
R	3×3 rotation matrix
R_r^l	rotation matrix between the right and left camera frames
R_1	rotation matrix between the world reference frame and the camera frame
T	3D translation vector
T_r^l	translation vector between the right and left camera frames
${}^{c}_{w}T$	transformation of world reference frame relative to the camera frame
${}^{pm}_{c}T$	transformation from the camera frame to the moved PTU frame
${}^{pi}_{pm}T$	transformation from the moved PTU frame to the initial PTU frame
${}^{pi}_{pm}T_1$, ${}^{pi}_{pm}T_2$	transformations between different PTU orientations and the initial PTU frame
${}^{w}_{p}T$	the transformation of the PTU frame relative to the world reference frame
T_1	translation vector between the world reference frame and the camera frame
T_2	translation vector between the origins of the two frames
T_3	translation vectore between the base frame and the PTU frame
(Xc,Yc,Zc)	coordinates in camera frame
(x_d, y_d)	coordinates of the distorted points
(X_l, Y_l, Z_l)	coordinates in the left camera frame
(X_r, Y_r, Z_r)	coordinates in the right camera frame
(X_w, Y_w, Z_w)	coordinates in the world reference frame
(x,y,f)	coordinates of p in camera frame
(x_{im},y_{im})	coordinates of p in the image plane
$x_1, x_2 \ldots x_n$	the n data points in the d-dimensional space R^d.
α	angle of view

α_c	skew coefficient, the angle between the x and y pixel axes
ψ	angle in pan axis
π	image plane
φ	angle in tilt axis
CAN	Controller Area Network
CCD	charge coupled device
CG	Center of Gravity
DSDP	docking and sample delivery port
DSS	drilling and sampling subsystem
ESA	European Space Agency
FOV	field of view
GCS	Ground Control Station
GIM	Generic Intelligent Machines
GIMI	GIM Interface
GUI	graphical user interface
IMP	Imager for Mars Pathfinder
LED	Light Emitting Diode
MDP	mobile drilling platform
MER	Mars Exploration Rover
MPF	Mars Pathfinder
MRoSA2	Micro Robots for Scientific Application 2
NASA	National Aeronautics and Space Administration
ROSA	Robot for Scientific Applications
OGCS	On-Ground Control System
PLC	Payload Cab
RFMU	Rover Functional Mock-Up
PTU	pan and tilt unit
PSPE	Payload Support for Planetary Exploration
RSS	Robotic Sampling System
SSF	Space Systems Finland Ltd
TKK	Teknillinen korkeakoulu
USB	Universal Serial Bus
VTT	Technical Research Centre of Finland

Chapter 1

Introduction

The human exploration of other planets could be traced back to several centuries ago. At that time, some low magnitude telescopes were utilized to observe the far away planets (William96). This human exploration had a significant development till the second half of the 20th Century. Since that time, human and robots have conducted this exploration. Many orbiters, landers and rovers have been launched towards other planets. The Moon and Mars appeal to human beings the most among all of these planets, and they also draw more effort on exploration. Many spacecraft, unmanned as well as manned, have been landed on the Moon, some missions also including rovers. The exploration of Mars fully relies on the unmanned spacecraft, especially in the last twenty years some orbiters, landers and rovers have provided significant information about Mars. Because of unmanned missions, these high-tech landers and rovers conduct the most of the experiments and exploration on Mars; therefore they are the critical parts of the exploration missions.

In this chapter, we will introduce Micro Robots for Scientific Application 2 (MRoSA2), which is a prototype of the micro rover aiming at exploration on Mars. Later, the objective and the thesis outline are presented.

1.1 MRoSA2- Micro Robots for Scientific Application 2

1.1.1 History

The Human being is characterized by endless curiosity and the continuous effort to fulfill that. The ability to convert curiosity to dreams and the ability to achieve those dreams make humans what they are today. Setting a foot on Mars is such a dream that drives the interest of many scientists around the world. However, to make this dream true many prerequisites need to be fulfilled, many experiments need to be performed to assess the situation, climate, risks, and several other important factors on Mars. The only way to perform these, without jeopardizing human life, is to use unmanned vehicle to scrutinize Mars surface and gather valuable information. This information can later be used by the scientists to retrieve and analyze important properties of Mars, which will allow evaluating the possibilities of fulfilling the long-lasting dream of mankind, to set foot on Mars. The retrieved information from the unmanned vehicle can also be used for researching another very popular and widely debated topic–extra-terrestrial life.

As far as our understanding of life, water is essential to life. Currently, based on the observation and research about Mars, it is commonly believed that there was liquid water on Martian surface millions of years ago, which was attested by the gaping canyons and river-like channels. Life might have evolved on Mars during that time, if that is true, the life might still survive under the ground[Nicholas08]. One of the main goals of Mars mission is to find out the evidence of life. At present, most of the samples are taken directly from the Martian surface, such as rocks and soils. These samples have been eroded due to the long time exposition to outer environment. In order to find out the life signs on Mars, some higher requirements are proposed by the ESA Exobiology Science Team Studies, such as accessing soil samples unpolluted by surface activities, reaching soil layers sheltered from surface process and so on. Specifically, the soil samples should be taken one meter deep under the surface, and the rock samples should be taken inside the rock after being drilled inside a few centimeters[Anttila02].

In order to fulfill these requirements, the project of Micro Robots for Scientific Application 2 was proposed and funded by ESA. The purpose of this project is to develop a prototype of Mars micro-rover named “MRoSA2”, which could drill on the Martian surface and take samples[ESA98].

In this project, Space Systems Finland Ltd (SSF, Finland), Technical Research Centre of Finland (VTT, Finland) and Helsinki University of Technology (TKK, Finland) designed a Robotic Sampling System (RSS). This system is composed by three parts: the rover, the lander mock-up and a control computer, shown in Figure 1.

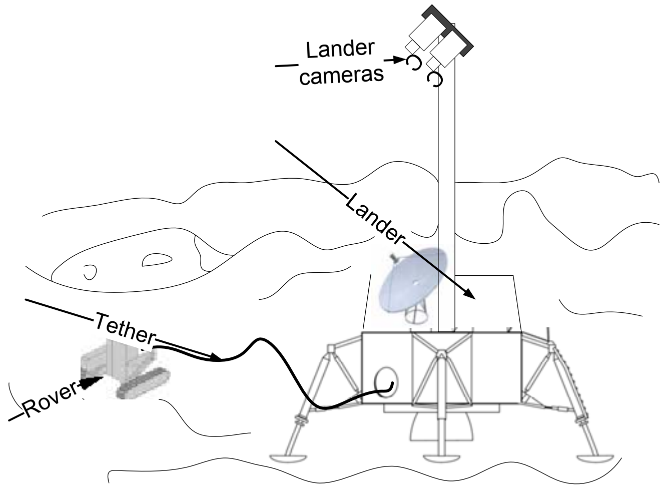


Figure 1 System overview

In the mission design, the rover should take samples from interested targets and return them to the lander. Parts of the samples should be taken by a drill which can drill into the regolith and rocks to take the samples under the surface. Each sample about 1 cm^3 is stored in the drill bit and for each drilling mission about ten samples are stored and returned to the lander. Several drilling missions are required for the rover. The power supply for the rover is provided by the lander through a tether, meanwhile, the tether also establishes the communication between the lander and the rover. The tether is wound around a scroll at its starting point. As the rover moves towards the target, the tether is reeled out in which case the tether is broken if the distance between the lander and the rover becomes longer than the one of the reeled out tether. When the rover needs to return the samples to the lander, it is required to exactly follow the trajactory towards the target for safely winding the tether around the scroll into the tether mechanism to protect the tether. Therefore the localization of the rover is important to control the motion of the rover precisely. Due to the uneven terrain, the localization in three dimensions is necessary. The rover can travel in a circular area with a radius of maximum 15 meter around the lander.

The lander is stationary with two stereo cameras standing on top of a mast. The stereo cameras take images of the rover. The lander onboard computer processes all the images to get the rover's localization information through the stereo vision. The lander communicates with the ground station on the earth through satellite, and communicates with the rover through the tether. It receives commands from the ground station and mostly these commands are related to control the motion of the rover. These commands are sent to the rover to command the rover to move to the interested spot and/or take samples. The system status of the rover are also sent back to the lander through the tether; these status and localization information together with the stereo images are sent back to the ground station during the communication period between the lander and the ground station. This information is analyzed and displayed on the ground station computer. On the basis of this information and the mission, the operators on the ground station send commands to the lander to execute before the communication sets up next time.

1.1.2 Hardware components

The rover consists of three parts: a mobile drilling platform (MDP), a drilling and sampling subsystem (DSS) and a docking and sample delivery port (DSDP). Figure 2 shows the rover with only MDP and DSS modules.

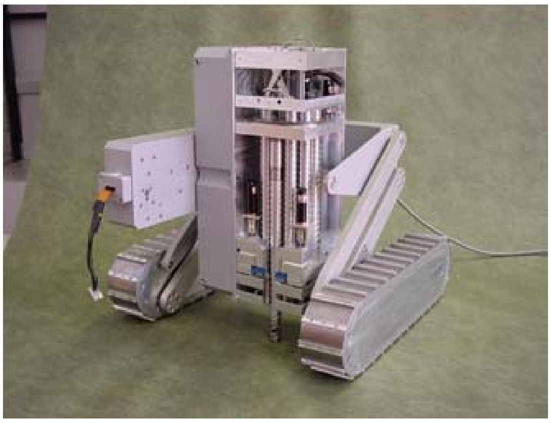


Figure 2 Rover with the MDP and DSS modules[Suomela02]

From a very high level functionality, the MDP serves as the platform to carry the DSS to the precise position to ensure the DSS drills on the target. Furthermore, it can drive back to the lander to deliver the samples. The DSS is capable of drilling into the rock and the Martian surface to get samples. DSDP is installed on the lander, which can transport the samples from the rover to the lander.

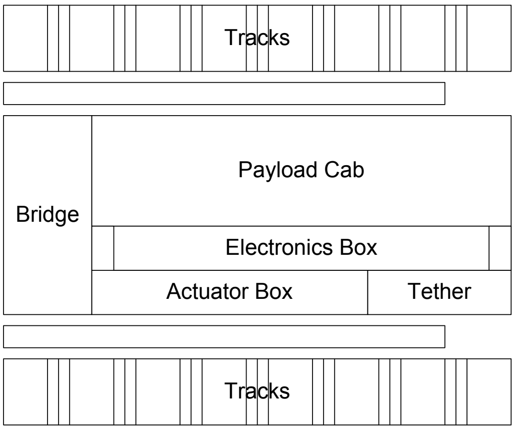


Figure 3 System structure of the mobile drilling platform

As can be seen from Figure 3, mechanically the rover can be divided into five different sections: (1) central payload cab, (2) actuator box, (3) tether, (4) bridge, and (5) track compartments. These sections are briefly discussed in following paragraphs.

The central payload cab is used to accommodate the control electronics of the rover and the DSS. The payload cab is also used to store the payload that is drilled out of the surface. For better flexibility of drilling activity, this section of the rover has several degrees of freedom for movement. It can be lifted up and down using four levers allowing a ground clearance from -20 cm to 20 cm. Additionally, the payload cab can also be tilted in a rotary position providing the independence of drilling in any angle without moving the rover itself[Suomela02].

The actuator box houses the lifting and tilting motors for the DSS. The motors are cleverly placed in the box to save space while still providing the rover full flexibility of moving around. This has been done by the use of a cylindrical gear and a conical worm gear. The choice of these gears also eliminates the need for a backlash and breaks, saving additional space[Lev00].

In front of the actuator box, the tether mechanics is located. The tether mechanics consists of feed and reel motors for winding and unwinding of the tether, power supply for the rover, and serial links for remote controlling the rover from the spacecraft[Lev00]. The total length of the tether is 40 meters.

The bridge, originally designed to contain the tether system, now only acts as a channel between the tracks. The tether system was moved from the bridge due to some control issues related to winding and unwinding.

The track compartments, one of the most important parts of the rover, are composed of two sealed symmetric modules. Each of the modules contains track actuators to drive the track and the corresponding control mechanics. The control mechanics for the tracks consists of a DC motor and a micro-controller ATMEL AVR128 with controller area network (CAN) to control the motor. The CAN is used to communicate with the micro-controllers from the central processor located at the payload cab. The track belts are rotated using an innovative design consisting of a worm gear. As the worm gear rotates, the cleats on the belt traverse down the thread of the worm, resulting in a rotational motion of the track belt[Lev00]. The design also has the advantage of saving space and making the breaking system redundant.

After completing this project, the department of Automation and Systems Technology of TKK retained a similar rover with the MDP module for future research and education. We name this rover with MDP module as “ROSA” in this thesis.

There are many circuit boards for power, control .etc, inside ROSA, and the main board is ReadyBoard™ 710 with an Intel 650 MHz Low-Voltage Celeron processor computer[ReadyBoard] which runs Linux operation system. The mobility program is running in the system to send and receive commands with the controller of motors and CAN bus.

1.2 Thesis objective

In the previous thesis work, a stereo system was developed to localize ROSA[Busch07]. The software is programmed in matlab running in Windows operating system. The stereo cameras are fixed on the lander at a height of 1.5 meter above the ground with a FOV of about 50 * 50 degrees. The lander is stationary once set up. Moreover, the localization needs the images from both cameras, and the distance between the cameras are about 40cm, thus the overlapping image is much smaller than the single image. Therefore, the localization area is strictly limited in a narrow strip in front of the cameras about 4m long and 3m wide. In this condition, the rover can only traverse in a specific small area around the rover, which degrades the original design that the rover should be able to explore a circular area with a radius about 15 meters. As a result, one objective of this thesis is to enlarge the localization area. To solve this problem, a pan and tilt unit (PTU) is installed to hold the stereo cameras on top of it. As it rotates, the stereo cameras move with it, which enlarge the observing area. A program is developed to automatically detect ROSA and control PTU to rotate in pan and tilt axes to make sure both cameras can “see” ROSA.

Considering the transparency, stability, customability and fast operation, in the new design of the ground station Linux is chosen as the platform in place of Windows. In the previous stereo system, the stereo cameras can't run in Linux; the localization program runs in matlab. The program uses the Epipolar geometry and image rectification to process the images, which requires a large amount of computation. In order to make the localization working in Linux, also independent of any other software, a new localization method with less computation is used in the program in C++.

Previously the software for the mobility control of ROSA was developped in Windows and can't be utilized in Linux[Wong08]. And the operation of the software is based on the command line to control ROSA, which is not convenient compared to a graphic user interface. So the new ground station software is made to control ROSA, which runs in Linux with a graphic user interface.

Compared to the lander, the rovers can traverse around the landing area or even explore more far from the lander. This advantage allows ROSA observing the Martian surface and the surrounding environment much more thoroughly than the lander. In order to utilize this advantage, a visual system is implemented on ROSA. A webcam is installed in front of ROSA. The images “seen” by ROSA will be displayed on the ground station.

1.3 Thesis overview

The thesis is organized as follows: in Chapter 2, two similar robotic systems to ROSA for Mars exploration are introduced. One is the Mars Pathfinder which had successfully fulfilled its mission on Mars, the other one is a research project "Payload Support for Planetary Exploration" sponsored by ESA. Later in this chapter, some basic tracking knowledge is presented, and the mean shift and camshift tracking algorithms are described in detail due to their wide usage in many tracking applications.

In Chapter 3, a novel design of the system is presented. The chapter starts with an introduction to the lander system, especially the PTU control software. This is followed by a description of the ground station software for controlling ROSA. The specification of the software developed for controlling the marker board and the communication model based on GIMnet are described afterwards. The chapter ends with an elaboration of the software development environment and the visual subsystem.

In Chapter 4, the theory of stereo vision is introduced, and the localization algorithm is developped. The mathematical model of PTU is figured out, which forms the transformation betweeen different PTU orientations. Based on this model, the transformation between the camera frames is also built.

In Chapter 5, a description of the tracking procedure used to follow the robot is presented. It is followed by a discussion of how the PTU uses the marker boards to autonomously control the cameras to follow the robot.

Chapter 6 introduces all the experiments: the stereo cameras and individual camera calibration; the transformation between several frames; the localization and tracking experiments.

At last, the conclusions and the future work are presented in Chapter 7.

Chapter 2

Literature Review

For the last decades, several landers and rovers have been launched on Mars and been successful with their missions. For example, the Mars Exploration rovers, Spirit and Opportunity, are still exploring on Mars. They explore on Mars, conducting experiments, recording geographical features and so on without any lander to cooperate in the mission. The Mars Pathfinder Mission contained a lander and a rover named Sojourner which cooperated together to complete the mission [Pathfinder]. After the success of the Mars Pathfinder (MPF) Mission, ESA started a “Payload Support for Planetary Exploration (PSPE)” project to study the feasibility of robot planetary exploration, which consists of a landing system, an imaging head, and Nanokhod rover. Since in the ROSA project the system is composed of a lander and a rover also, we will have a close look at the functionality of the rover and lander of MPF and PSPE. Nowadays, many algorithms concerning video tracking have been proposed and applied in research. Some basic concepts of video tracking and popular algorithms are introduced in this chapter.

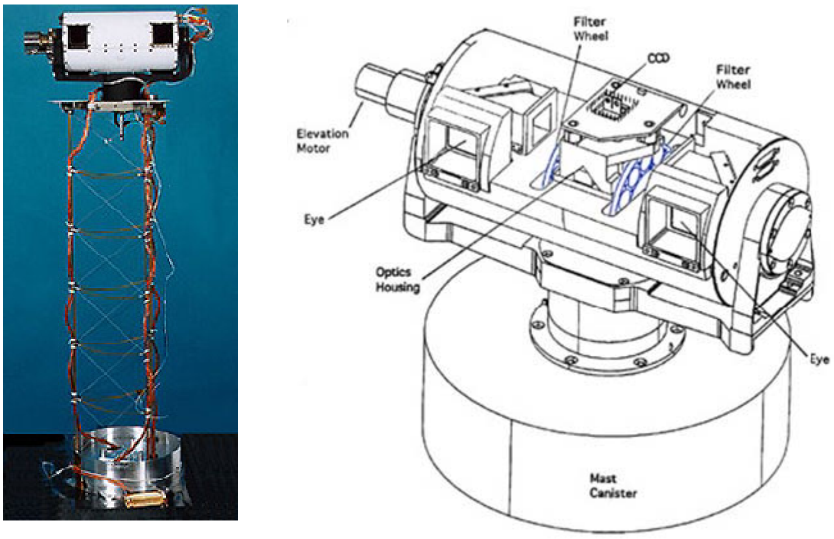
2.1 Mars Pathfinder

The MPF mission was funded by NASA, starting on 4 July 1997 and ending on 26 September 1997 for a total of 83 sols (1 sol = 1 Martian day = 24.6 hours). During the mission, the microrover, Sojourner, traversed about 100 meters and almost circumnavigated the lander with a furthest radius 13 meters from the lander.

2.1.1 Visual System - Imager for Mars Pathfinder

The lander carried many instruments for scientific and engineering purposes. The Imager for Mars Pathfinder (IMP) is one of the key instruments, shown in Figure 4. The imager generated high-resolution color, stereo, and multispectral panoramas

[Smith97]. Based on these images, a 3D terrain model around the landing site was built which was used for the localization of the rover. It can also provide images of the rover for tracking or recording.



(a) (b)

(a) Flight Model IMP with mast deployed (b) the mechanical sketch of IMP

Figure 4 The structure of IMP [IMP]

Inside IMP, there are a CCD detector and a filter wheel. One color image is taken with one filter; several color images are taken separately and layered onto one another to form a full color image. The CCD detector is 256×512, divided into two halves; each camera has an active element of 256×248 pixels. The Field of view for each camera is 14.4° × 14.0°.

Figure 5 shows a simplified optical design of the IMP. The filter wheel and many parts of the camera have been left out. The light travels through the windows, reflects by the mirror and prism, passes through the filter and lens, finally projected on CCD detector [IMP]. Because of the high-grade design of the cameras, the images show a clear and accurate reflection of the observed scenes, which are guaranteed to produce precise stereo (3D) images.

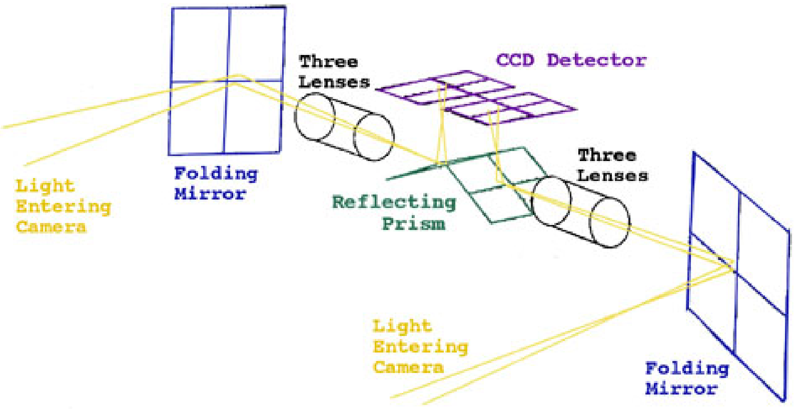


Figure 5 Simplified optical design of IMP [IMP]

The IMP sits on a deployable mast. When the mast is fully deployed, the IMP is 1.5 meters above the ground. The IMP can rotate horizontally and vertically. When it takes a panorama image, it rotates step by step, taking photos at each step, and these photos are combined to form a panorama.

2.1.2 The rover control workstation (RCW)

Sojourner is controlled by the human operator using the rover control workstation (RCW). In this software, based on the images taken by the IMP, a 3D map of the Martian surface is made. Wearing the special goggles, the 3D scene is presented in front of the eyes. A model of the rover can be placed and moved in the scene; the XYZ position, heading and ranges of the rover at that position in the scene are also given on the screen [RCW], shown in Figure 6.

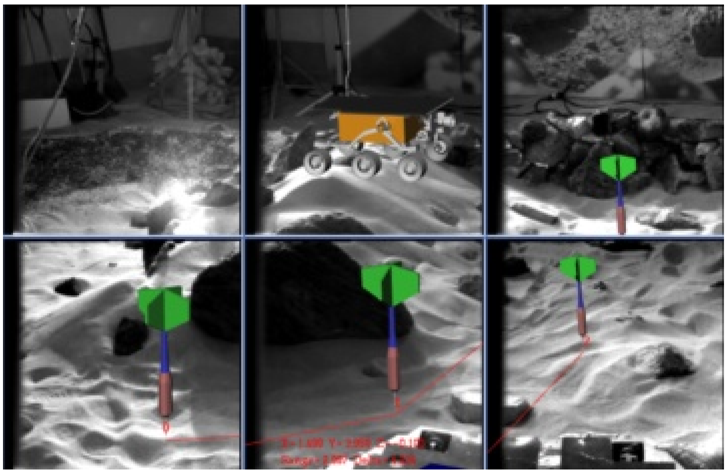


Figure 6 The procedure to localize the rover on a 3D terrain model [RCW]

For calculating the orientation of the destination, put the rover module on that point, all the position information is given about the destination. By choosing several points on the planned path, all the coordinate information will be stored in some commands, which will be sent to Sojourner during the next communication period between the lander and the ground control station. The rover position during the sol cannot be accessed till the end-of-sol that the downlink navigation telemetry transfers back the end-of-sol stereo images. The stereo images of the rover can be used to determine the position and orientation of the rover at end-of-sol. The true rover traversal path can be reconstructed by the stereo images of the rover tracks[Wilcox98].

Figure 7 shows the RCW software. The rover command can be selected and stacked in the command sequence [Cooper98], which will be uploaded to the rover during the next uplink.

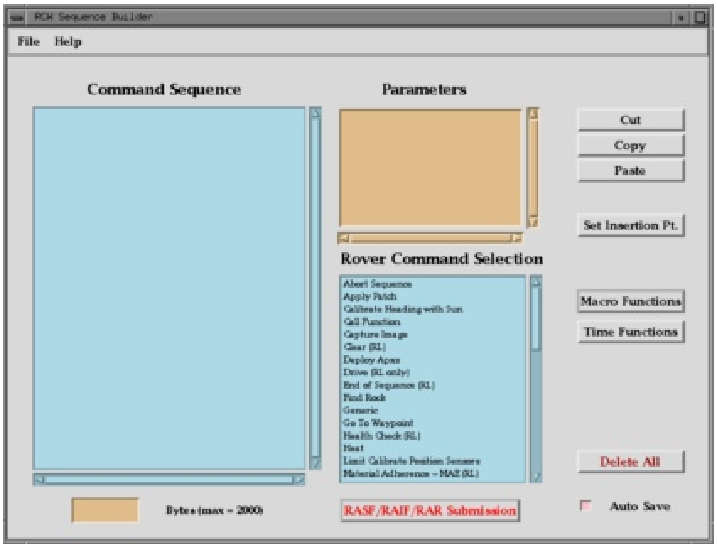


Figure 7 Rover control workstation [RCW]

2.2 Payload Support for Planetary Exploration (PSPE)

2.2.1 System composition

The PSPE project aims to study the feasibility of robot planetary exploration. It contains a lander, an imaging head, and a rover named Nanokhod.

The lander system carrys all the scientific instruments and control system for the rover, and equipments for the communication with the Ground Control Station (GCS).

The imaging head has two stereo cameras which are space-qualified and a pan and tilt unit which has two degrees of freedom. They perform two tasks during the mission: taking high-resolution stereo panoramic images to build the 3D terrain model around the landing site and tracking the Nanokhod to localize it [Vergauwen01].

The Nanokhod is a simple, efficient and tracked-driven rover. It carried scientific instruments loaded in a “Payload Cab” (PLC) for the in-situ measurements. A tether system was installed in the rover for getting the power supply and communication from the lander, which constrains the rover to move in the vicinity of the lander with a radius of about 20 meters. The components of the whole system are shown in Figure 8.

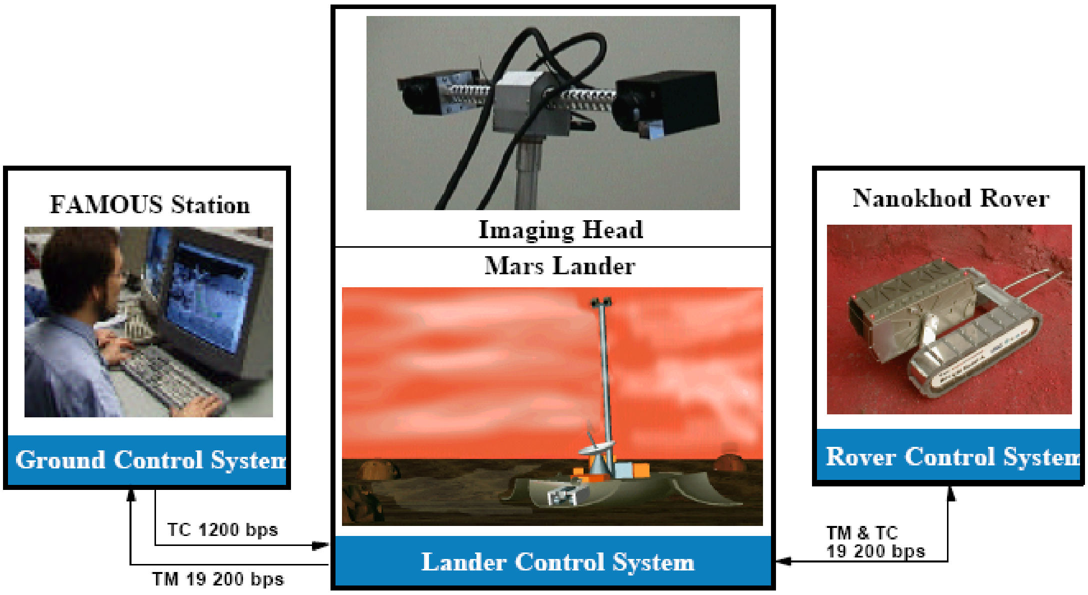


Figure 8 System components of the PSPE [Landz04]

The On-Ground Control System (OGCS) has two responsibilities: to generate the 3D terrain model and define the path and communicate with the lander.

2.2.2 Calibration of the system

Since the cameras were used for the 3D modeling and localization, the precise calibration had to be carried out before the usage of the stereo vision system. Different from the other stereo systems where the cameras were still during the mission, the calibration of the Imaging Head was more complicated. Except calibrating the relative transformation between the stereo cameras, the calibration of the pan and tilt unit was also needed to be carried out.

The calibration of the stereo cameras adopts the epipolar theory. When the lander touches down, the stereo cameras take photos about the landing areas, and these images are sent back to the GCS. Through computing the epipolar geometry of these images, the relative transformation between the two cameras of the imaging head could be obtained.

However, the transformation between the two cameras is not sufficient to reconstruct the terrain model and get the localization of the rover. The transforms between the cameras and the imaging head and between the Imaging head and the lander are also needed. The Imaging head frame is set in the middle of the two cameras, so that the transformation from one camera to the Imaging head and from the Imaging head to another camera is the same.

The transform from the imaging head to the lander is complicated. It required the calibration of the tilt axis and the pan axis.

The setup of the calibration of the tilt axis is shown in Figure 9. At the beginning, the stereo cameras take images covering the horizontal-line area in the figure (a). Then the Imaging head moves in the tilt axis to the second Image head frame. The cameras take images covering the vertical-line area, making sure the two areas overlap in some certain area. Build the 3D model from each pair of the images. Due to the overlapping, the reconstructed models contains the same features in some part. By aligning the corresponding features in the two models, the relative transformation between the first and second Imaging Head frames can be found[Landz04].

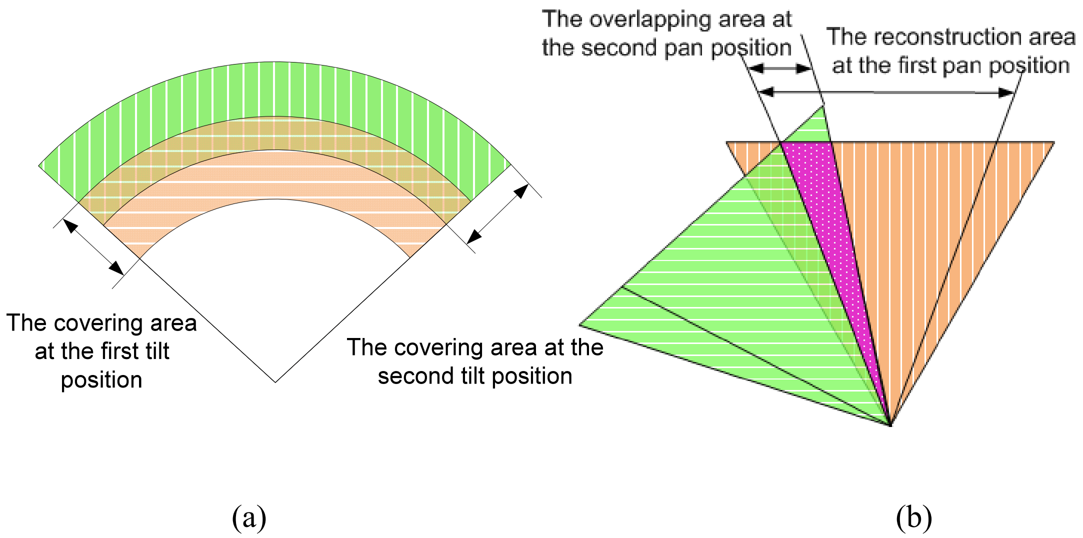

The covering area at the first tilt position
The covering area at the second tilt position
The overlapping area at the second pan position
The reconstruction area at the first pan position
(a)
(b)

Figure 9 Symbolic representation of the setup for the computation of relative transformation for tilt axis (a) and pan axis (b)

The calibration of the pan axis is slightly different from the tilt. The setup is shown in Figure 9 (b). Before and after the movement of the pan axis, the stereo cameras take two pairs of images: the first pair of images covering the vertical-line area, the second pair of images covering the horizontal-line area. The overlapping areas in each pair are used to build the 3D model. However, it is important to make sure that one image overlapped with the overlapping area of another pair of image, shown in the dotted area in the figure. In the first pair of the images, build the 3D model. Find the corresponding features through epipolar between the right image from the second pair overlapped with one image from the first pair. By the corresponding features, the position of the camera taking the right image in the second pair can be estimated in the frame of the first pair. Given the transformation between the camera and the Imaging head, the transformation between the two Imagin head frames can be calculated.

While calibrating the tilt axis, due to the unavoidable vibration of the pan axis, the calibration had to be calculated in an iterative procedure with an estimated pan value till this solution converged [Landz04].

Navigation requires precise localization. To simplify the algorithm of the localization, active markers were fixed on the rover. Four LEDs were installed in the four corners on the top surface of the rover. By controlling the on and off state of the LEDs one by one, through background subtraction and blob detection, the location of the marker can be yielded in the frame of the images. By the stereo vision, the rover position with respect to the lander can be calculated.

2.3 Basics of video tracking

Object tracking has obtained a great deal of attention these years. Due to the powerful computing abilities of the computers and the high quality video cameras as well as the increasing application field, the tracking of objects has got more attention in research, and consequently numerous algorithms have been proposed.

The video tracking can be divided into three steps: detecting of the interesting objects, tracking the object from one frame to another, and recognizing the behavior of the object [Yilmaz06].

The features widely used for the representation are points, contours and geometric shapes. When the object is small in an image, it is suitable for the point representation. The object can be represented by one point, usually the centroid of the object or a set of points [SERBY04]. The contour defines the boundary of the objects, which is often used for the representation of a non-rigid object. Rectangles and ellipses are often used as geometric shapes for tracking simple rigid objects or nonrigid objects.

Tracking usually utilizes the different features for different application. The commonly used features are edge, color and optical flow.

Edge belongs to the contour-based representation. The boundary of the object usually generates strong changes on the image which can be represented as edges. Another advantage of edges is that they are less sensitive to the illumination. Many edge detection algorithms have been proposed. Due to the simplicity and accuracy, Canny edge detector [Canny86] is very popular in the applications.

Color is another popular feature that is easy to perceive and process. The appearance of color is influenced by the illumination and the reflection of the objects, also the color spaces such as RGB (red, green, blue), HSV (Hue, Saturation, Value) are sensitive to noise. However, due to the large number of colored objects and the simplicity of the detection, the color is also widely used as one of the tracking features.

Optical flow is a good way to get motion information between a sequence of fames without any other prior knowledge about the content of the frames. It works under the image brightness constraint which assumes the brightness remains constant for the corresponding pixels in the consecutive frames. This prerequisite somehow limits the application of optical flow in some situations. It has two branches: dense optical flow and sparse optical flow. The dense optical flow associates velocity with some pixels or the displacement that represents the distance a pixel has moved between the previous frame and the current frame. Horn and Schunck [Horn81], Lucas and Kanade [Lucas81] introduced some of the most popular algorithms for computing the dense optical flow. The algorithm of the sparse optical flow relies on specific features which have certain desirable properties. Tracking only features decreases the computational intensity and is relatively reliable and robust.

2.4 Mean shift algorithm

Mean Shift is a simple nonparametric technique for the estimation of the density gradient. It is related to the discipline of kernel density estimation, whereas the kernel refers to some function that has local focus, for instance, a Gaussian distribution. It is a very useful tool for finding modes in a set of data samples and manifesting an underlying probability density function (PDF) in N-dimension space (R^N) [Comani02]. The PDF can be used in many feature spaces, such as color space, scale space and so on.

The meanings of the mean shift algorithm can be expressed intuitively in Figure 10 [Ukrai]. The balls present the data set or the features in some space. Initially, choose an interested region among the data. In the window, the center of mass is calculated, and the distance between it and the center of the window is the mean shift vector:

$$m(x) = \left[\frac{\sum_{i=1}^{n} x_i g_i \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g_i \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right] \tag{2.1}$$

x_1, $x_2 \dots x_n$ are the n data points in the d-dimensional space R^d. g_i is the minus of the derivative of the kernel. h is the radius of the window.

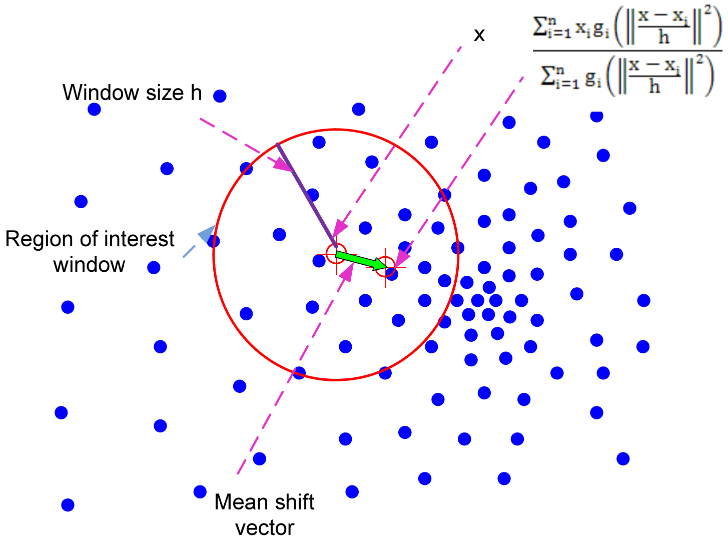

x
$\frac{\sum_{i=1}^{n} x_i g_i\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g_i\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}$
Window size h
Region of interest window
Mean shift vector

Figure 10 Intuitive description of the mean shift algorithm

Image can be considered a collection of data. The different features or properties can be seen as the data set, so the mean shift algorithm can also be applied on images. First, set a search window with the initial location, shape and size. Then calculate the center of mass inside the search window; move the center of the search window to the center of mass previously calculated. Repeat this step till convergence [Cheng95] or until the window moves less than a preset threshold [Comani02].

The above application can be simplified by a rectangular kernel. A rectangular kernel is just as its name expresses that there is no falloff with distance from the center, until a sharp edge transition to zero. The simplified mean-shift vector equation is to calculating the center of mass of the image pixel distribution [Bradski08].

$$x_c = \frac{M_{10}}{M_{00}} \tag{2.2}$$

$$y_c = \frac{M_{01}}{M_{00}} \tag{2.3}$$

The zeroth moment is calculated as

$$M_{00} = \sum_x \sum_y I(x, y) \tag{2.4}$$

The first moments are

$$M_{10} = \sum_{x}\sum_{y} xI(x, y) \tag{2.5}$$

$$M_{01} = \sum_{x}\sum_{y} yI(x, y) \tag{2.6}$$

The mean shift algorithm has many advantages. It can handle arbitrary feature spaces; it does not require the prior information on the data clusters; initially only one parameter needs to be determined: the window size. But on the other hand, the selection of the window size is very important for the algorithm. An inappropriate window size can cause the modes to be merged or generate unexpected mode.

2.5 Camshift algorithm

Camshift [Bradski98] stands for continuously adaptive mean shift. Compared to mean shift, it can automatically change the size of the search window which makes it more popular in the video tracking since the tracked objects can change the dimension in the images. Similar to mean shift, camshift can be applied in any feature space. It is very popular to be applied in the color space, such as color-based video tracking.

First, process the image from the RGB color system to the HSV color system. The HSV system defines a color in terms of three constituent components:

- Hue, the color type (such as red, black, or green)
- Saturation, the "vibrancy" or "purity" of the color
- Value, the brightness of the color

Figure 11 shows the HSV color system. As value and saturation are low, the hue becomes quite noisy, and it can easily change in a wild range. In this case, the color presented by the hue is not accurate. To avoid this problem, a lower and an upper threshold are set for the image. For the pixels with brightness lower than the low threshold, the hue value is ignored. Similarly when the brightness value is greater than the upper threshold, the hue value of the pixels are also ignored. As for the saturation, a lower threshold is also set and it works at the same level as the lower threshold for the brightness.

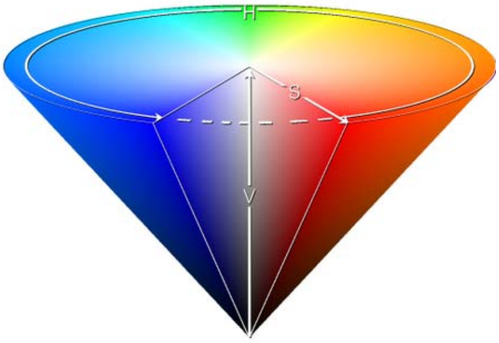


Figure 11 HSV color system (Image sourced from The Wikipedia)

After covering the images from the RGB to HSV space, separate out the Hue from the Saturation and Value. The color model is created by taking 1D histogram from the Hue channel. The histogram is only created once at the beginning of tracking and saved for the future use. The stored histogram is used as a model or look-up table, covert the incoming video frames into a probability image, which each pix presents the probability of this pix to be the tracked object. The probability ranges from 0 (no probability) to 1 (the maximum probability) [FT07].

Figure 12 shows an example of how to calculate the probability. The left graph is the Hue histogram, and each bin is stacked together in the image on the right. The ratio of each bin in the whole stack shows the probability.

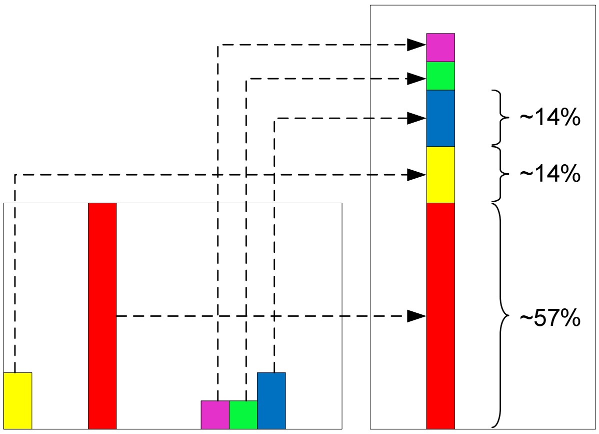


Figure 12 Hue histogram and the stacked histogram for calculating the probability

The window size can be automatically adjusted during the searching procedure in camshift. The size is a function of the zero moment, which presents the distribution area under the search window.

When camshift is utilized in the video tracking, it does not need to calculate the color distribution of the whole image; it only calculates the area a bit larger than the search window. This considerably saves the computing power and makes the algorithm faster than some other algorithms.

The procedure of applying Camshift in images is similar as the one for meanshift. The slight differences are the searching area is a bit larger than the searching window and the window size changes with the zero moment.

Since in video tracking by camshift the color information is separated from the brightness, the object tracking has more tolerance for the large variation of the illumination. However, due to the wild swing of hue in low brightness and saturation, the camshift does not work well in dim environments.

The Mars atmosphere is filled with a huge amount of dust particles. These particles scatter the incoming sunlight, which leads to the sky with a significant brightness also in redness toward the horizon at midday[Thomas99]. The atmoshpere and weather conditions on Mars are still not so clear for the human beings. Due to the complication and uncertainty on Mars, no online tracking technique has been applied on Mars yet. It is a challenge for applying the current tracking techniques on Mars, meanwhile it will also be thrilling to successfully realize tracking on Mars.

Chapter 3

System Structure

The whole system consists of three parts: the lander mock-up, the rover-ROSA, and the ground station.

3.1 The lander

The lander mock-up works for the navigation of ROSA in the system. Previously it had two stereo cameras fixed on top of it which form stereo vision to obtain the localization information of ROSA. Due to the limited field of view (FOV) and fixed position of each camera, the scenery of the system is limited in the narrow strip shape in front of the cameras. In order to enlarge the localization area, a pan and tilt unit is installed on the lander.

3.1.1 The stereo cameras

For stereo vision, high qualified cameras are the prerequisite for obtaining accurate localization information. Since the lander is only a mock-up at the moment, the selection of cameras is constrained to the range of affordable webcams. In our system, two Philips Webcam SPC900NC are chosen as the stereo cameras, shown in Figure 13. It utilizes the VGA CCD sensor. The maximum video resolution is 640 × 480 pixels. The focus is manually adjustable, and the frame rate can be set from 5fps to 60fps.

Figure 13 Philips Webcam SPC900NC

The camera bar with two stereo cameras and a pan-tilt unit are shown in Figure 14. The aluminum camera bar sits on top of the pan and tilt unit. The baseline (the distance between the optical axes of the two cameras) is 40 cm. The height of the lander can be adjusted. In our system, it is adjusted at the height of 1.5 meter.

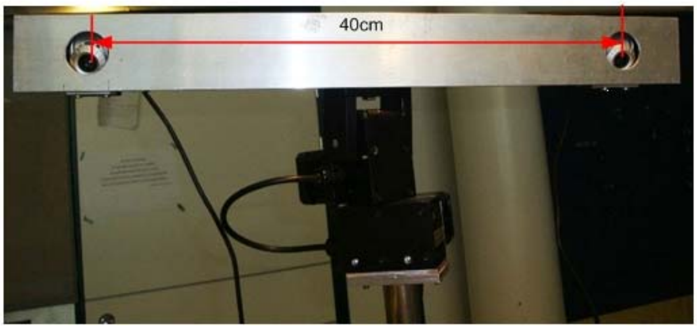
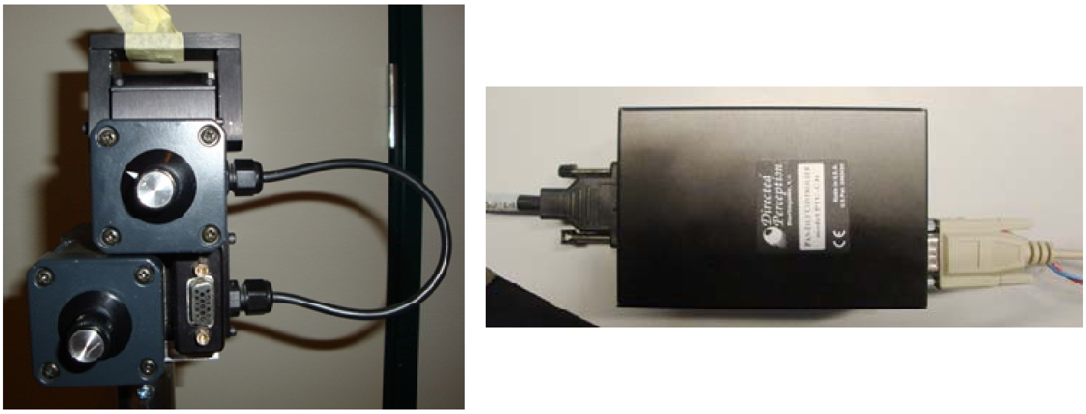


Figure 14 The Imaging Head of the lander mock-up

3.1.2 The pan and tilt unit

The Computer-Controlled Pan-Tilt Unit can provide a fast and accurate positioning of cameras and other payloads. The model used in our system is PTU-46-70 from Directed Perception Inc, shown in Figure 15 (a). It can carry a load of over 6 pounds, and the maximum speed is over 60°/second with a resolution of 46.284 seconds of arc per position (0.0129°). Due to its fine resolution, it is well suited for the precise control of position, speed and acceleration [PTU06].



(a) Pan-tilt unit (b) Pan-tilt controller

Figure 15 Pan-tilt unit and controller

The pan-tilt unit is connected to the pan-tilt controller, shown in Figure 15 (b); the power can be supplied from an AC/DC power supply or from a battery. The host computer sends a command through the RS-232 to the controller which drives the PTU to move to a new position with a certain speed and acceleration or get the PTU status according to the command. It can perform self calibration upon reset, change position and speed on-the-fly, and control the power consumption from the host. The power supply can range from 11 VDC to 30 VDC. In order to achieve the highest performance of PTU, the highest voltage within the allowable range should be used. To obtain the quietest performance, a lower voltage should be used. In our system, the two cameras and a camera bar sit on top of the PTU and they move according to the PTU movement. As the aluminum camera bar is about 50 cm long, when it rotates in pan axis, a torque is generated. In order to provide enough torque and save power at the same time, 16VDC is used for the PTU. The system structure of PTU is shown in Figure 16.

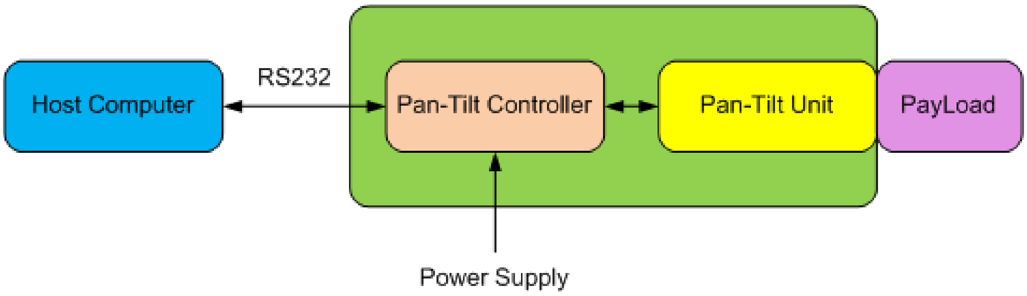
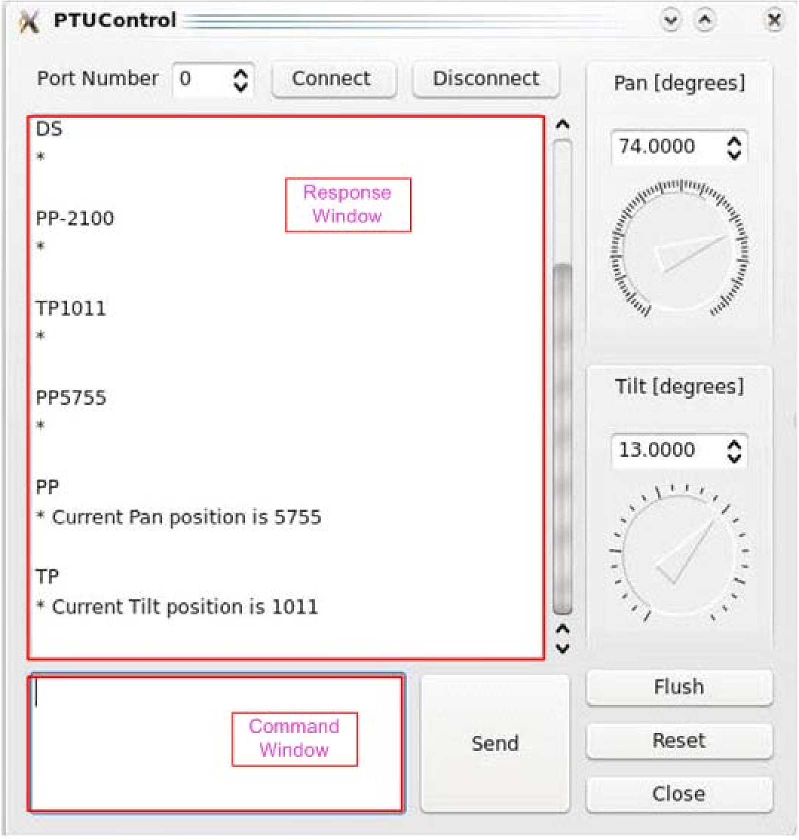


Figure 16 The system structure of PTU

Prior to any position operation, the pan and tilt axis should be reset in order to calibrate themselves. The hardware limitation of the turning angle of pan axis is -160 degrees to +160 degrees and -45 degrees to 30 degrees for the tilt axis. The direction of PTU “looking down” represents minus. In the PTU operation, the position limit can be enforced by software. When limit-enforce is enabled, commands out of the limitation return an error message and are not executed. There are two position execution modes, immediate and slaved. In the immediate position execution mode, the commands are executed as soon as received by the controller. In the slaved execution mode, all the commands are stacked up until some specific command is sent to force all the commands to execute one by one.

The communication between the host and the controller is RS232 serial communication. The baudrate is 9600 bps with 1 start bit, 8 data bit, 1 stop bit and no parity. Hardware handshaking and XOF/XOFF are not used.

The command can be sent by terminal interface software in Linux, such as “minicom”. However, it requires the operator to set the communication parameters in “minicom” before setting up the communication and remember all the commands of PTU and input them in the text interface, which is inconvenient and prone to errors. Moreover, there is no interface for the other software to control the PTU throught the “minicom”. In order to overcome these drawbacks, the “PTU control” software is made. The graphic user interface of the software is shown in Figure 17.



Port Number 0

Pan [degrees] 74.0000

Tilt [degrees] 13.0000

```
DS
*

PP-2100
*

TP1011
*

PP5755
*

PP
* Current Pan position is 5755

TP
* Current Tilt position is 1011
```

Figure 17 The GUI of “PTU control” software

The usage of “PTU control” software

The GUI can be divided into three parts: setting up communication, sending commands and getting responses. In Linux each serial device has a unique ID. In order to set up the communication between PTU and the software, the ID of PTU is checked and the corresponding port number selected. All the commands can be sent from the command window. The system properties of PTU usually only need to be set once, in the later application they do not need to be reset. These kinds of commands can only be sent from the command window. Most of the operation of PTU is to change its position in the pan and tilt axes. For convenience, the right panel provides two control groups for adjusting the pan and tilt positions. The position can be entered or adjusted from the spin box for a precise positioning. Moreover, the dial provides an easy and intuitional way to adjust the position, but it can only adjust position in steps of whole degrees. The unit of all the inputs is degree. The functions of the other buttons are as follows: “Flush” to flush the serial port; “Reset” to reset the PTU; “Close” to quit the program.

The structure of the ”PTUcontrol” class is shown in Figure 18. The functions can be grouped into several modules. “Serial” module provides the hardware level serial communication; “Basic” module provides the basic control of PTU. These functions set up the foundation for the GUI related PTU commands in the “PTU GUI” module; furthermore, they are the interface for controlling PTU without the GUI, which are especially useful when integrating the PTU with another program. “PTU GUI” module implements all the GUI related control of the PTU.

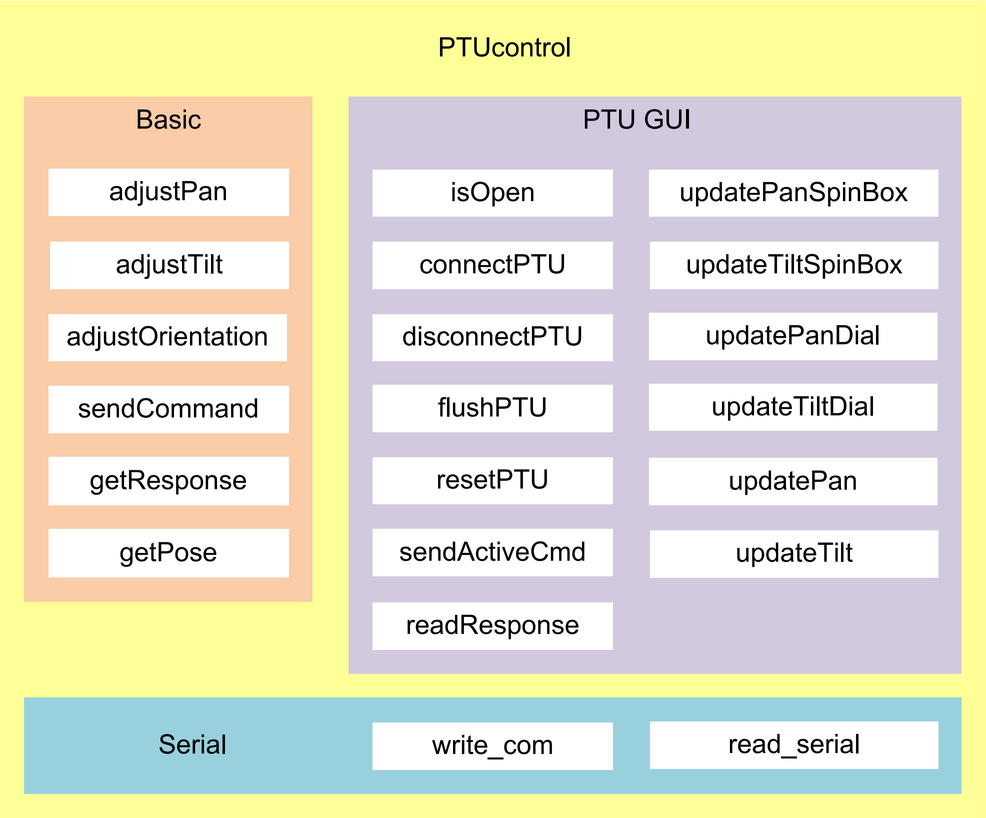
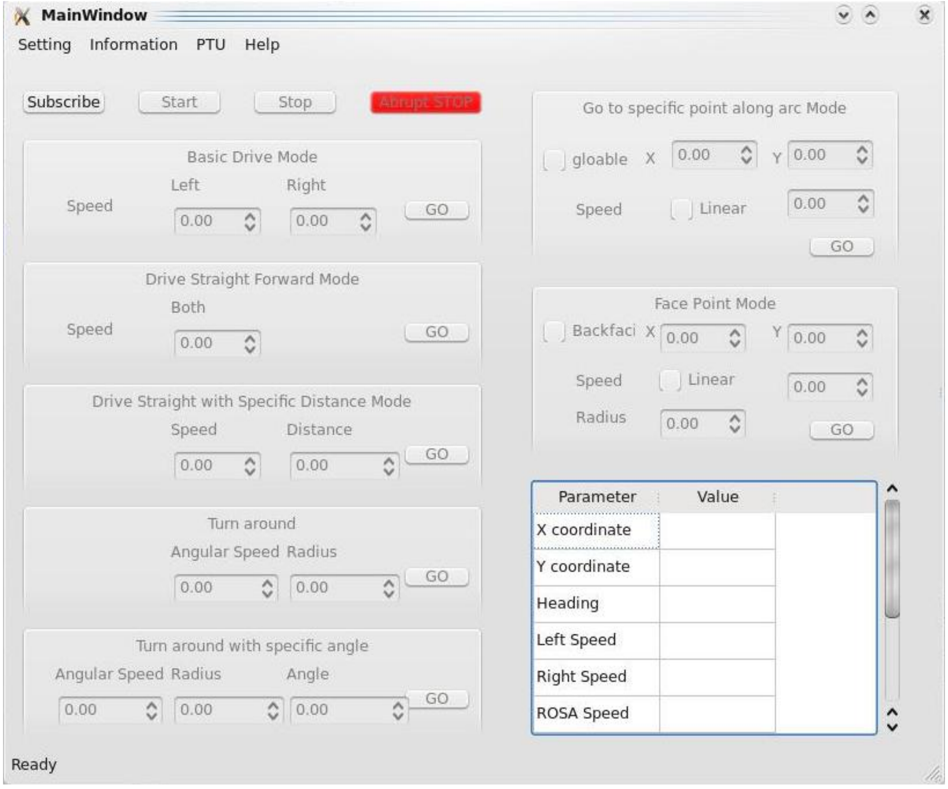


Figure 18 “PTUcontrol” class

3.2 The rover-ROSA

The mobility system of ROSA can be divided into two parts. One part is running in the computer of ROSA to control the motors through the CAN Bus, interfacing with the low level hardware; the other part is running in the ground station to send control command through some communication. The design and theoretical calculation of this library have been described and implemented in [Wong08]. However the part running in the ground station is programmed in Windows with a text based user interface. In order to implement the ground station in Linux as well as to provide an easy and friendly user interface, the software "Ground Station" is made. The main interface is shown in Figure 19.



Parameter	Value
X coordinate	
Y coordinate	
Heading	
Left Speed	
Right Speed	
ROSA Speed	

Figure 19 The main interface of the "Ground Station" software

ROSA has seven driving modes in the mobility library.

1 Basic Drive Mode: set the left and right track speeds separately. If the speeds of both tracks are the same, ROSA will drive straight, otherwise it will turn and the trajectory depends on the speeds.

2 Drive Straight Mode: as the name states, set the speed for the rover which goes straight forward or backward.

3 Drive Straight with Specific Distances Mode: set the speed and distance that ROSA should travel.

4 Turn around: The rover travels along a circle with the user-defined radius and angular speed. In Figure 20, 2-dimension coordinates is set on ROSA. The origin is its Center of Gravity(CG). The positive axis of Y points to the forward direction. XY axes form the orthogonal systems. The radius can be positive or negative. When it is

positive, the rover will turn right, otherwise it will turn left. When the angular speed is positive, the rover will turn counterclockwise, otherwise it will turn clockwise. The direction is illustrated in Figure 20.

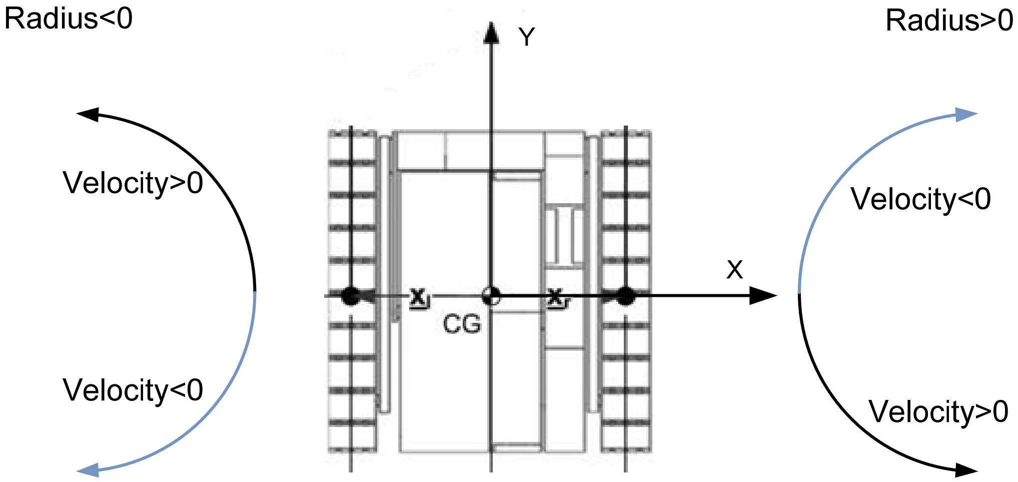


Figure 20 The direction of turning w.r.t radius and angular velocity

5 Turn around with specific angle: it is similar to "Turn around" but the rover will stop after turn a specific angle. The absolute value of the angle will be taken into the calculation.

In the above five modes, the position is calculated based on the encoder measurements. The following two modes are related to the localization of the rover. The position is obtained from the stereo cameras of the lander.

6 Go to specific point along arc Mode: Given the target coordinates, the software can automatically calculate the turning radius and specific angles. The rover will travel along the calculated arc to the target point.

7 Face Point Mode: The rover can travel to the appointed global position with the appointed radius or the minimum radius. It can also travel backface to the target position which means that the tail of the rover face the target position. This is for the purpose that the rover traverses back to the lander as well as for rewinding the tether.

The last two modes are not implemented but given an interface in the software. The flow of the "Ground station" is described as Figure 21. First the program connects to the GIMnet (see Figure 22), which will be introduced in the section 3.4.2, then set up the communication with ROSA. After successfully setting up the communication, the rover will be activated by start. During the process of motion, it can be stopped at any time. The system parameters (see Figure 24) and acceleration (see Figure 24) of ROSA can be set while ROSA is idle. The status of ROSA (see Figure 25) can be updated in real time when ROSA stops moving.

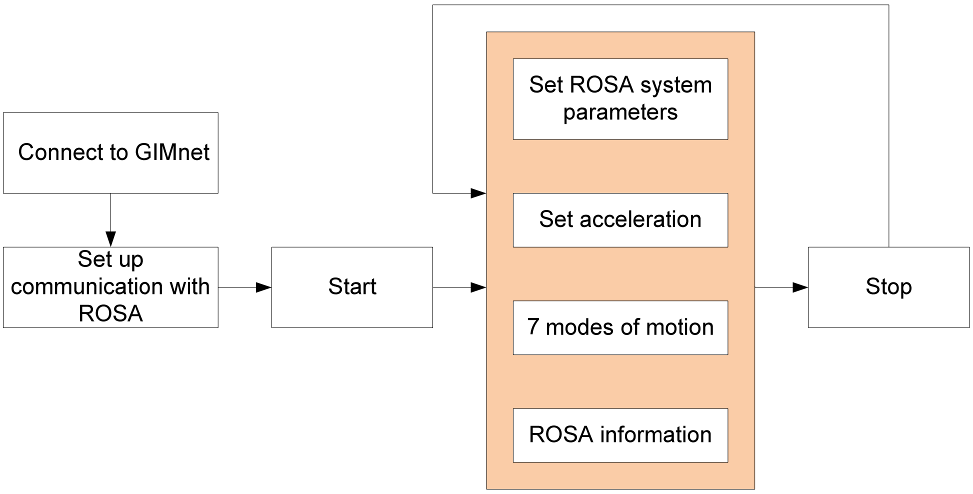
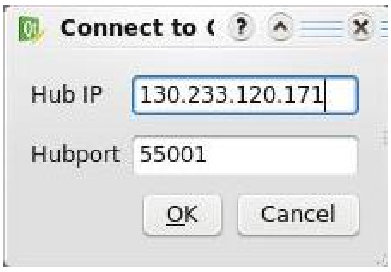


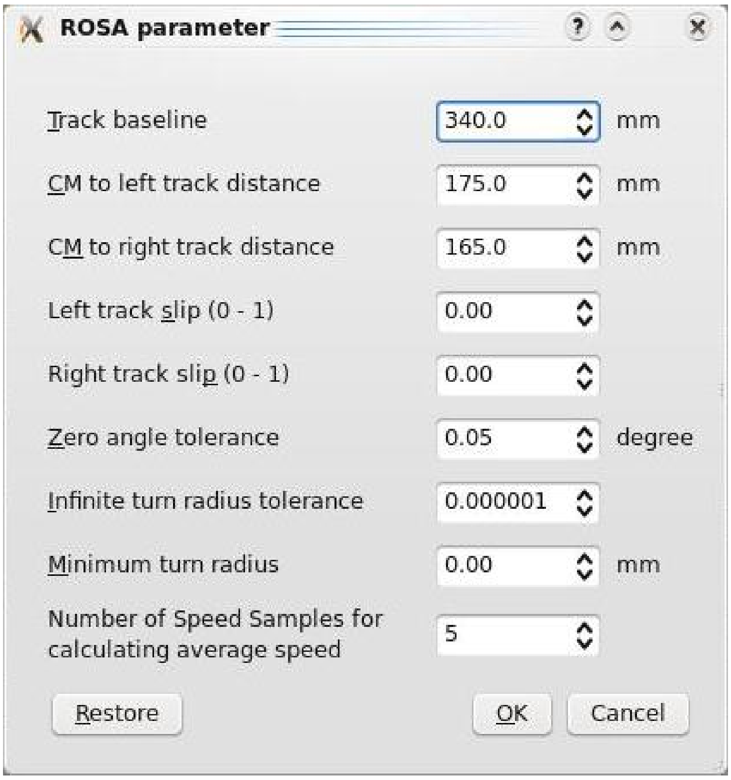
Figure 21 The procedure of operating "Ground Station"



Hub IP 130.233.120.171

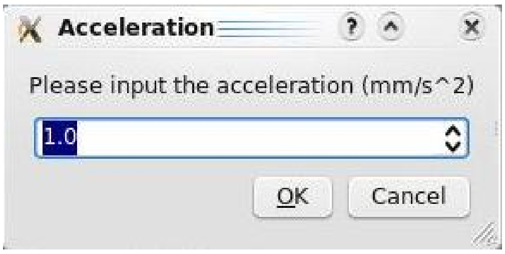
Hubport 55001

Figure 22 GUI of connection of GIMnet



Track baseline	340.0	mm
CM to left track distance	175.0	mm
CM to right track distance	165.0	mm
Left track slip (0 - 1)	0.00	
Right track slip (0 - 1)	0.00	
Zero angle tolerance	0.05	degree
Infinite turn radius tolerance	0.000001	
Minimum turn radius	0.00	mm
Number of Speed Samples for calculating average speed	5	

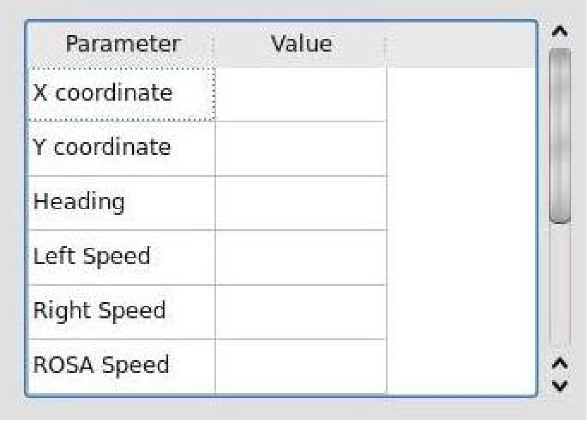
Figure 23 GUI of setting the system parameter of ROSA



Please input the acceleration (mm/s^2)

1.0

Figure 24 GUI of setting the acceleration of ROSA



Parameter	Value
X coordinate	
Y coordinate	
Heading	
Left Speed	
Right Speed	
ROSA Speed	

Figure 25 The status of ROSA shown on the GUI

XY coordinates coincides with the one fixed on ROSA before ROSA starts moving, showing in Figure 20.

3.3 The marker board

The localization of ROSA is based on the stereo vision. Stereo vision refers to the ability to percept or capture the three dimensional information. Common algorithms of stereo vision involve receiving images from two or more cameras from different viewpoints and find the corresponding points from different images and reconstruct 3D information [Trucco98]. The active marker will decrease the difficulties of finding the corresponding points in the images. The active marker used on ROSA is a marker board.

The marker board is composed of three LEDs which are white GaN/InGaN of type RL8-W110-360. They are coated with phosphorescence which makes the LEDs emit light in all directions. The diameter of the LEDs is 8 mm. The marker board is shown in Figure 26.

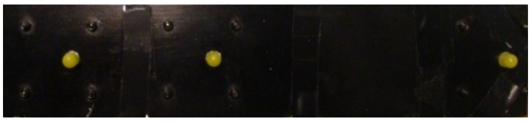


Figure 26 The marker board

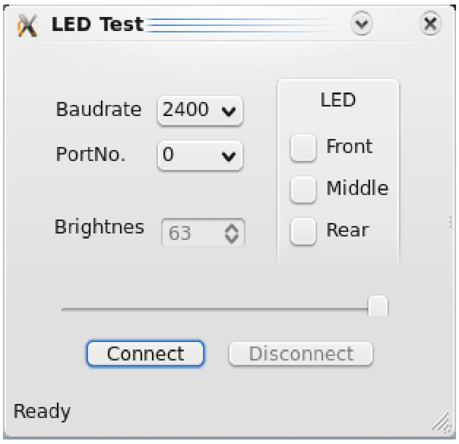
The distances between the neighboring LEDs are 80mm and 170mm. It is fixed on the top of ROSA on one side. The LED far away from the middle one is in the forward direction of ROSA. The purpose of the unequal distances is to distinguish the front and rear parts of ROSA.

The power for the marker board is between 12V and 18 V. A voltage regulator inside the marker board circuit converts the input power to 5V which is connected to the microcontroller and the LED drivers.

The communication interface of the marker board is through RS232. The specification is 2400 baud, 8 data bits, 1 stop bit and no parity.

The LEDs on the marker board can be controlled independently for the ON and OFF state. The brightness of the LEDs can also be adjusted, from pure off to pure on. Zero means completed OFF. As the number increases, the LEDs are getting brighter. If some LED is off, the adjustment of brightness will not turn the LED on, but the level is memorized. When the LED is turned on, it automatically work on the last brightness setting.

In order to control the marker board easily, the Marker board control program is made, and the GUI is shown in Figure 27. Before starting the serial communication, the baud rate and the port number should be set correctly. The port number is the ID number of the marker board in a Linux system. The LEDs can be turned on and off separately and their brightness can be adjusted by the spinbox or the horizontal bar.



Baudrate 2400

PortNo. 0

Brightnes 63

Ready

Figure 27 GUI of marker board control program

The previous design of the marker board in the system is shown in Figure 28. In the previous system implementation, the power supply for the marker board is an external adjustable DC power supply, and the marker board is directly connected to the ground station computer by the RS232 interface.

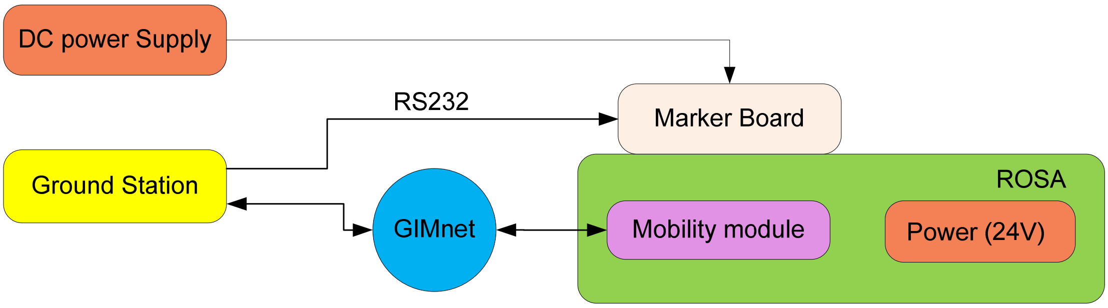


Figure 28 Previous marker board design in the system

As ROSA moves around, the power and communication cables have to be long enough to cover all the activity areas. Also when ROSA heads in a different direction, these cables tend to tangle with each other or even ROSA, which will damage the cables causing the marker board to cease working, or even will influence ROSA working properly.

In order to solve this problem, the system structure relative to the marker board is redesigned in another way, shown in Figure 29.

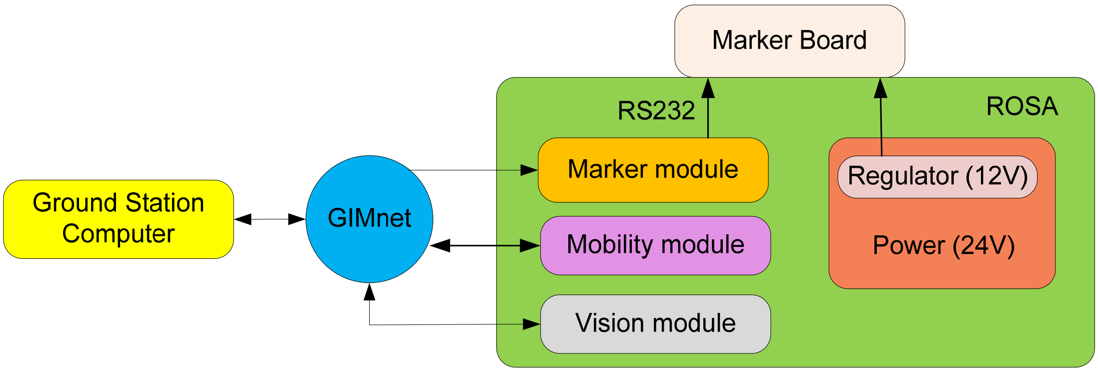


Figure 29 The new marker board design in the system

In this design, the power supply of the marker board comes from ROSA. The input power in ROSA is 24 volts, which is transformed to different voltages providing to a variety of circuits inside it. In order to convert 24 volts to 12 volts for the marker board, a regulator circuit is built in ROSA.

In ROSA, the main power supply is 24V. Since the marker board accepts power from 12V to 18V, a converting circuit is made to transform 24V to 12V. The schematic diagram is shown in Figure 30.

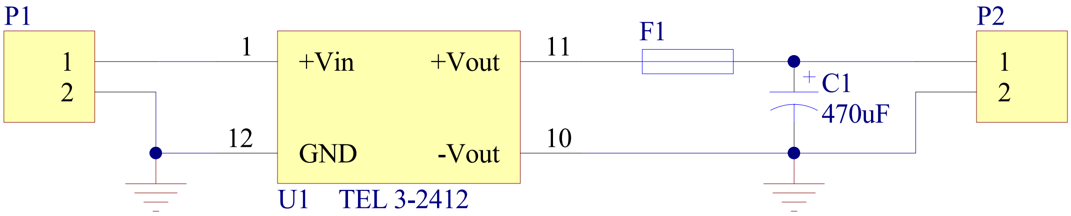
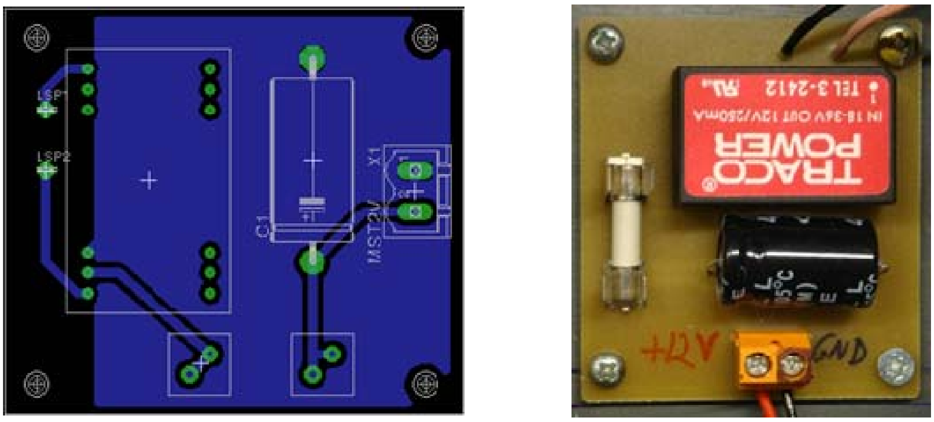


Figure 30 The schematic diagram of the converter circuit for the marker board

In this circuit, the model of the convert chip is TRACO POWER TEL 3-2412 DC/DC converter. Its input voltage ranges from18 to 36 VDC, the output voltage is 12 VDC, and the maximum output current is 250mA [TRACO]. The PCB and the circuit are shown in Figure 31.



(a) the PCB board

(b) the converter circuit

Figure 31 The converter PCB and circuit

3.4 Communication

3.4.1 Communication system design

The theoretical communication design of the whole system is shown in Figure 32. ROSA communicates with the lander through the tether which can transport the data and power. The lander communicates with the ground station though satellite. However, in our system, the communication between the lander and the ground station is impossible through satellite, also the tether is not functional. Thus, the practical communication links of the whole system is shown in Figure 33.

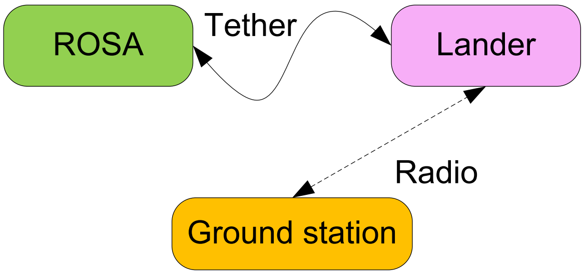


Figure 32 The theoretical communication design

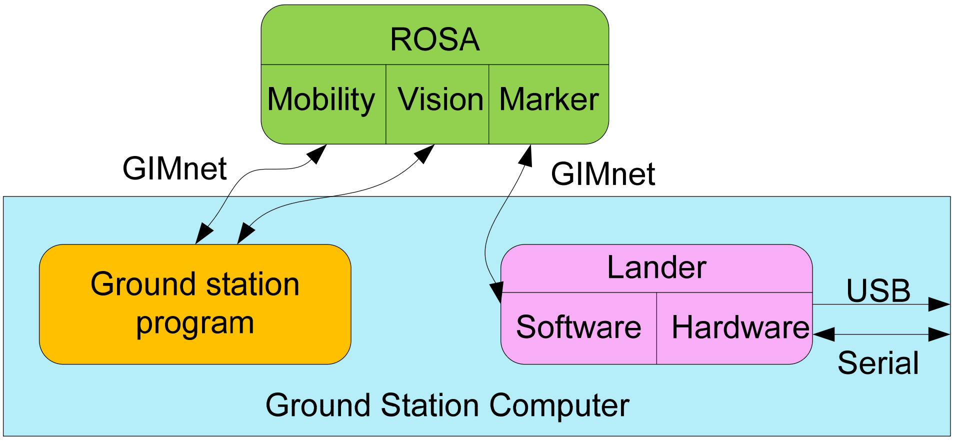


Figure 33 The practical communication design

Instead of communicating with the Lander through the tether, ROSA communicates with the Ground Station computer through GIMnet. The on-board lander programs will run in the ground station computer with the ground station programs. There are two hardware connection links between the lander and the ground station computer: the stereo cameras on the lander are connected via USB ports and the PTU controller via the serial port.

3.4.2 GIMnet

GIMnet is part of the research of the Center of Excellence in Generic Intelligent Machines (GIM) at Helsinki University of Technology and Tampere University of Technology [GIMnet]. GIMnet is a remote process communication infrastructure, which works as the base architecture for the software which utilizes GIMnet for communication with other software in the same network [Saar07]. The structure of GIMnet is introduced by the example of the connection between ROSA and the ground station computer, shown in Figure 34.

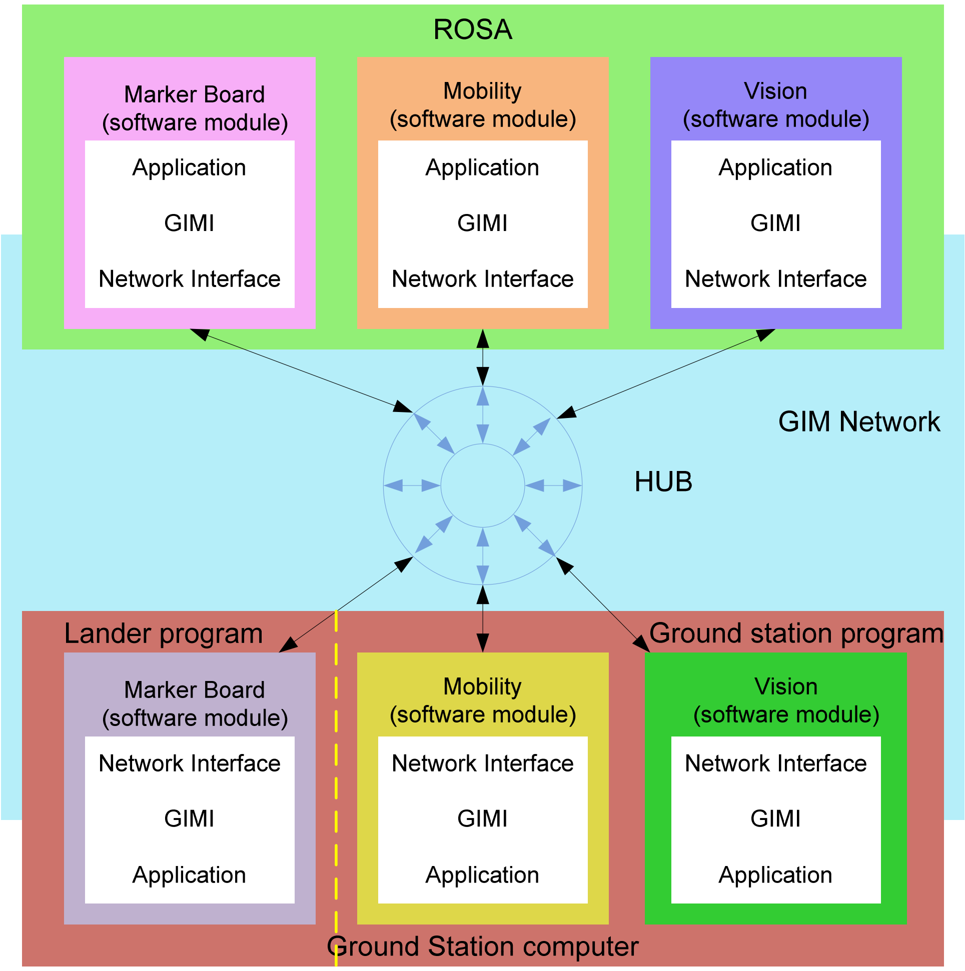


Figure 34 GIMnet communication between ROSA and the ground station

The HUB is the backbone of the GIMnet; as the name tells, it performs the tasks of a hub. It is fully platform dependent, and runs only on Linux. The Generic Intelligent Machine Interface (GIMI) provides some API for developing the modules for the GIMnet application [GIMI]. The software modules separately run in the network. In our system, there are six modules, three in ROSA and two in the ground station programs and one in the lander program. The six modules form three pairs of functions, one for the mobility control of ROSA, one for the control of the marker board, one for the visual subsystem of ROSA. Taking the mobility software module as an example, the schematic of the theory of the communication by GIMnet is shown in Figure 35.

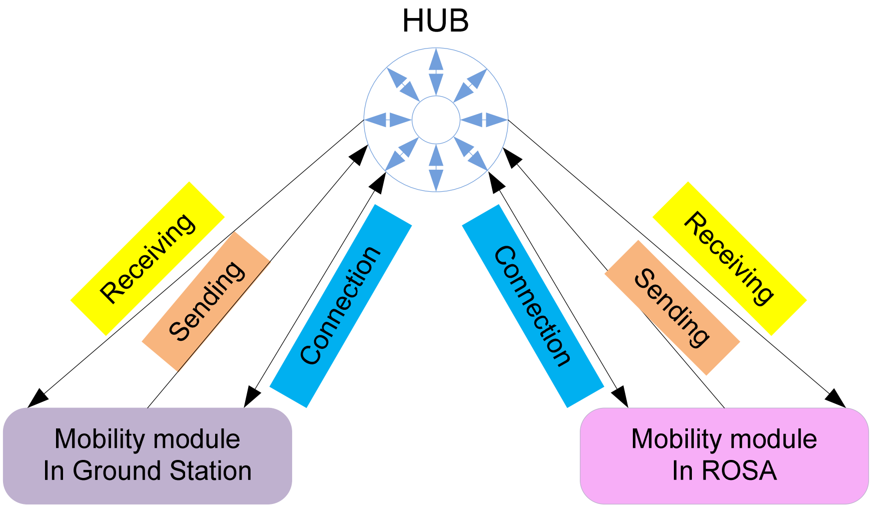
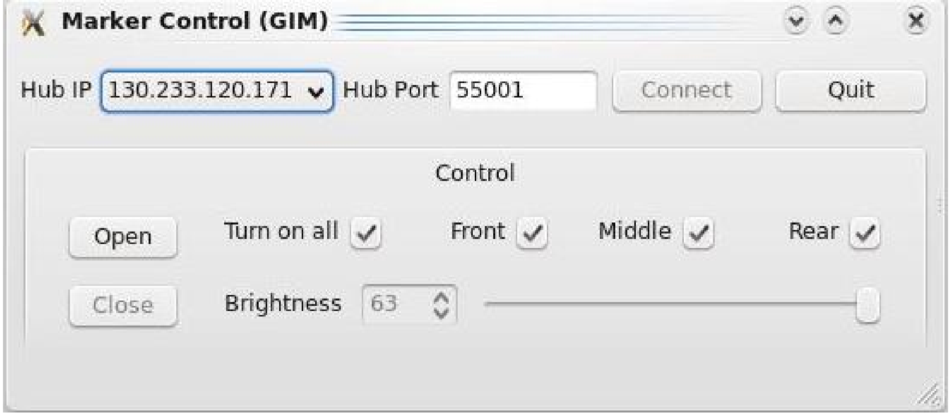


Figure 35 Schematic of the communication theory of GIMnet

Before communicating with each other, both of them connect to the same hub, at the mean time, each module registers their name and receives an ID, both of which can be used to address the module. Each module also provides some service to the other modules as well as subscribes to some service from the other module. The communication between each module is based on messages. Each message has a type ID (major type) to distinguish it from other messages as well a minor type ID to separate the different senders. Each message carries some protocols that are designed in the corresponding modules. After receiving the message, the modules perform as these protocols. A filter can be used to permit the messages with the desired types and senders to be in the message queue. The messages are deleted from the message queue after arriving at the destination.

3.4.3 Marker board GIMnet module

In Section 3.3, the new design of the marker board in the system has been introduced. The control commands will be sent through GIMnet. Based on the marker board control software, a control program of marker board through GIMnet is made. The graphical user interface of this software is shown in Figure 36.



Hub IP 130.233.120.171 Hub Port 55001

Turn on all ☑ Front ☑ Middle ☑ Rear ☑

Brightness 63

Figure 36 GUI of the marker board control through GIMnet

Before connecting to the hub, the hub IP address and port number need to be specified. All the commands previously sent directly through the serial port are all encapsulated into messages with a different type ID. The module for the marker board control in the ground station provide their own service also subscribe the service from the module in ROSA, and vice versa.

The program structure of the module for the marker board control in the Ground station is shown in Figure 37. The functions in the class can be grouped into two categories. In the “gimfunction”, all the functions are combined with the GIMI APIs to control the marker board. In the “Marker GUI”, the functions bundle the GUI command with the functions in the “gimfunction”.

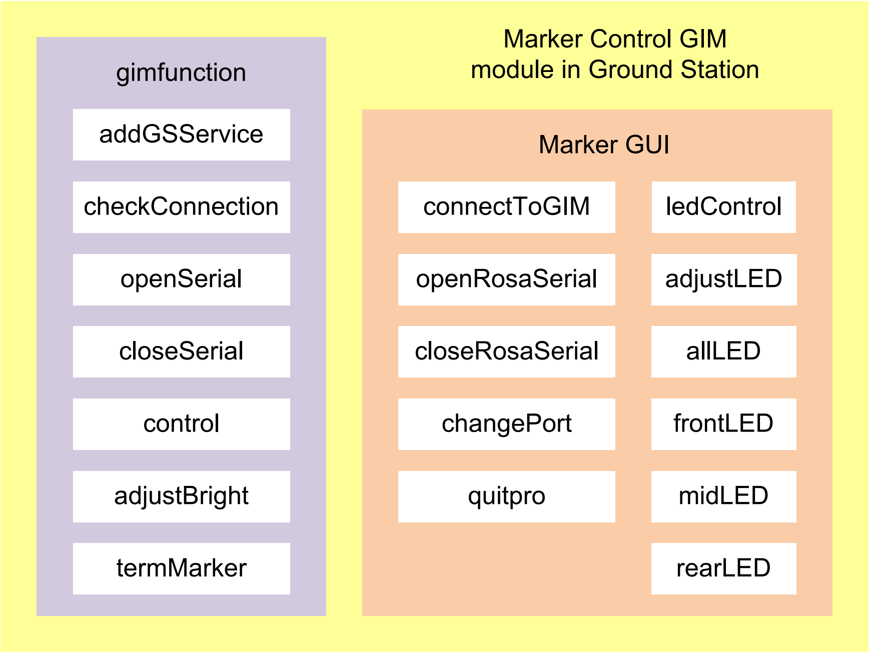


Figure 37 The Marker Control GIM module in Ground Station

The module in the ROSA is different from the one in the ground station. It controls the marker board in the hardware level based on the corresponding message received from the ground station through GIMnet. It does not have GUI.

3.5 Software system design

In the previous implementation of the project, the program was composed of two parts: mobility control in Dev-C++ software with a text user interface and the localization and marker board control by Matlab in the Windows operating system, shown in Figure 38. The data communication between the mobility control and the localization are through the Matlab Engine.

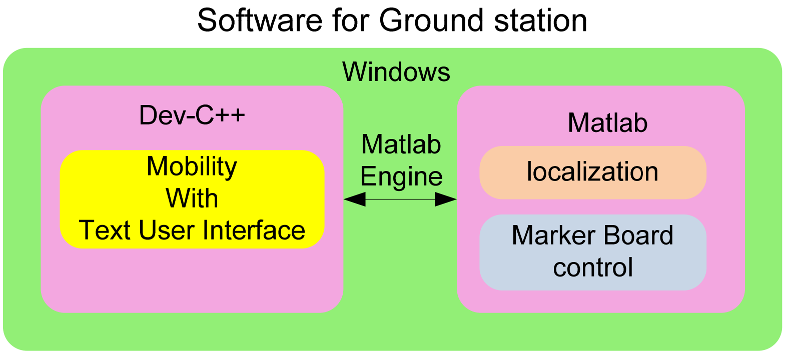


Figure 38 The old software system design in the ground station

Compared to the previous software system design, the new one has three different key points:

- Linux is utilized as the operating system in the ground station computer instead of Windows.
- Qt is utilized to develop the graphic user interface for all the application programs.
- OpenCV is utilized to process the video and images instead of Matlab.

The development environment is shown in Figure 39.

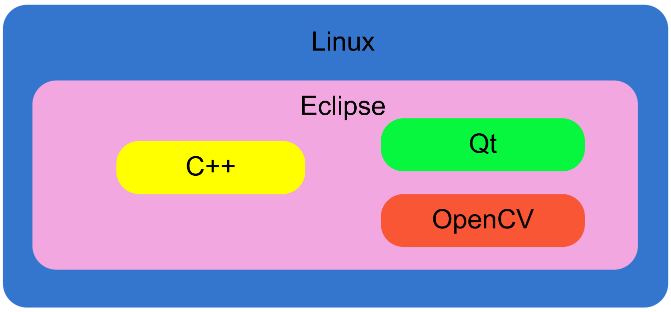


Figure 39 The program development environment

In the new software system design, the operating system is Linux. The main advantage of Linux in our system is that now we can understand everything that is going on in the ground station computer. Windows is like a black box, much system information cannot be handled outside. If there is a conflict between the developed software and the system or some system errors emerge, there is no way to fix the system to adjust it to the developed software. Furthermore, part of the software running on Windows is unnecessary for our developed program. These software consume many system resources, slowing down the whole system, as well as decrease the system stability. In Linux, the kernel can be configured and the software installed according to the program requirements. Thus the system is compact, transparent and stable.

Qt by Trolltech is a standard framework for developing high-performance cross-platform applications [Jasmin]. It has good portability between Windows, Linux and Mac OS X. It can also transport between desktop and embedded operating systems. It can integrate development tools with a cross-platform IDE. In our system, it is integrated with Eclipse. It is object-oriented, providing a C++ class library which can easily extend the program [Daniel07]. It provides all the functionality which is needed by a high-performance GUI. In the previous sections, all the GUIs are made based on Qt.

OpenCV (Open Source Computer Vision) is an open source computer vision library. It is designed for computational efficiency mainly aimed at real time computer vision [OpenCV]. It is written in C and C++ running under Windows, Linux and Mac OS X. Matlab is powerful generic software. It provides a powerful toolbox for image processing, as well as many other toolboxes, numerous easily-used functions. These functions provide a powerful and comparative easy application for image processing. However, Matlab is built upon Java, and Java is built upon C. So when a Matlab program is running, the code will first be interpreted to Java then to C, and finally be executed. Since OpenCV is written in C or C++, all these interpretations are omitted. The programs execute much faster than Matlab. Moreover, Matlab consumes a lot of system resources which slows down the whole system. When data communicates between Matlab and C/C++ programs, some special programs, such as Matlab Engine have to be utilized, which further slows down the whole process.

Previously, all the image processing and localization were realized in Matlab, which is slow for a realtime application. In order to improve the realtime performance and, also to eliminate the dependence to some special software, like Matlab, OpenCV is used for the new implementation of the image processing.

The new software system design in the ground station computer is shown in Figure 40. The software system can be divided in two parts: one for the ground station, one for the lander. Due to the communication design, both of the software tools run on the ground station computer. For the ground station, the software mainly control ROSA, setting and checking status, sending commands for traverse and getting the position information by encoders. Another feature of the ground station is the display of the images from the webcam carried by ROSA. This module is introduced in the next section.

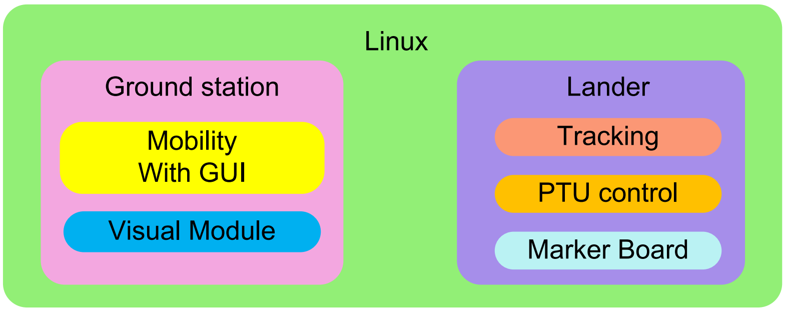


Figure 40 The new software system design for the ground station computer

3.6 Visual subsystem in ROSA

As ROSA moves in the vicinity of the lander to take samples, it is an opportunity to observe the Martian surface in a close distance. Even though there are two stereo cameras on the lander to observe ROSA and its surrounding, these two cameras are too far to get the clear images compared to the camera on ROSA. These images can be used to direct the operators on the ground station to find the interested samples and command ROSA to move to the desired place, also for the scientists to study the geologic situation on Mars later. To fulfill these requirement, a camera is installed in front of ROSA, shown in Figure 41. It is connected to the computer on ROSA through USB port.

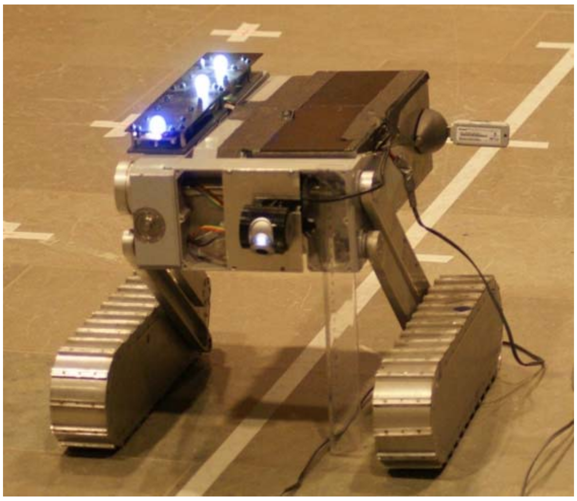


Figure 41 A camera installed in front of ROSA

This visual subsystem contains two programs running separately in ROSA and the ground station. The program in ROSA utilizes the camera module in the machine control interface (MaCI) to turn on/off the camera, take images and send them to the server in the GIMnet. MaCI is a library for robot programming based on the modular and reusable design [MaCI]. In the ground station, another program receives images from the server through GIMnet, displays and saves the images. The structure of the modules in ROSA and ground station is shown in Figure 42.

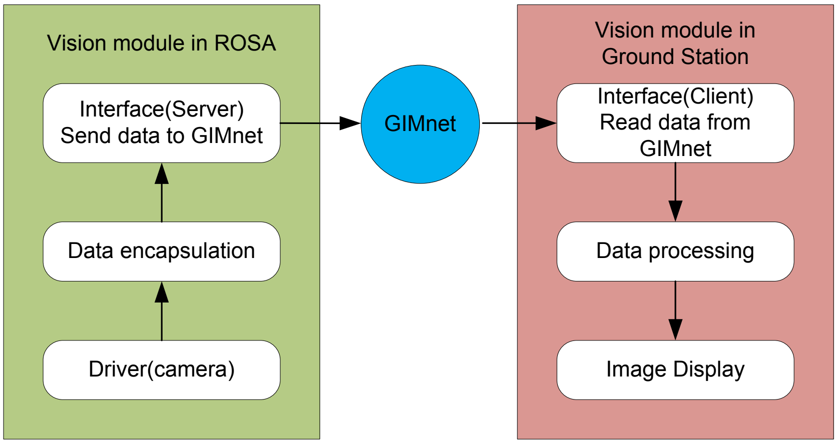
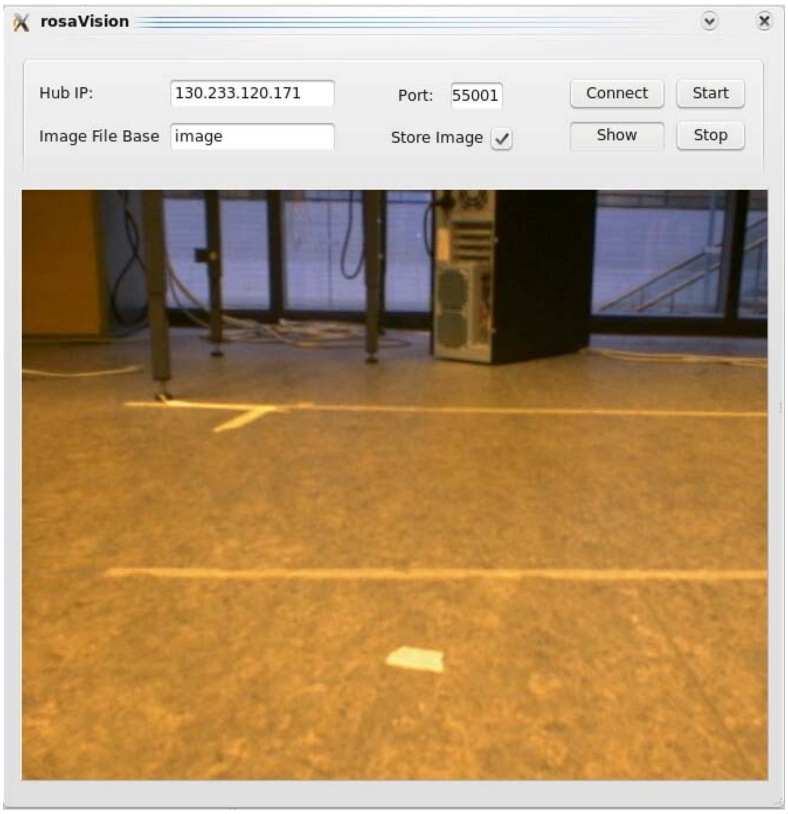

Vision module in ROSA
Interface(Server) Send data to GIMnet
Data encapsulation
Driver(camera)
GIMnet
Vision module in Ground Station
Interface(Client) Read data from GIMnet
Data processing
Image Display

Figure 42 Modules for the visual subsystem

The GUI of ”ROSA Vision” is shown in Figure 43. After setting the Hub IP and port number, ”Connect ” button sets up the communication between the program to GIMnet. ”Start” and ”Stop” buttons work as their names state to start and stop receiving images from GIMnet. The recieved image is displayed in the GUI if ”Show” button is pressed down, as seen in Figure 43. The users can choose whether to save the recieved images by checking ”Store Image” checkbox. The naming convention is the letters in the ”Image name Base” , followed by the image number. The image number starts from 0 when the program starts running, and increases 1 after one image is saved.



Hub IP: 130.233.120.171

Port: 55001

Image File Base image

Store Image ☑

Figure 43 GUI of software “ROSA Vision”

Chapter 4

Localization

As ROSA traverses in the vicinity of the lander, it is very important to get the localization information with respect to some known coordinates system. The localization information includes the position and orientation of the rover. The position usually can be obtained from the odometry sensors, such as wheel encoders, and the orientation can be calculated through some sensors, like the accelerometers and gyroscopes [Everett95]. However, the accumulated error of the encoder sensors in ROSA makes the measurement accuracy decrease as the rover travels a longer distance, especially when the rover traverse on the rough surface. At the moment, no orientation sensor is installed in ROSA. Due to these reasons, a stereo camera system is adopted to get the localization information of ROSA.

In this chapter, firstly, the stereo vision system which is used for localization is introduced, secondly, a mathematical model is built for the pan and tilt unit to model the rotation in the pan and tilt axes. At last, the transformation of several coordinates system is depicted.

4.1 Stereo vision

Stereo vision refers to inferring the 3 dimensional information of an object or scenery or the distance information with multiple images taken from different viewpoints. It is widely used in the robotics and many other application areas.
Before introducing the theory of the stereo vision, first, let’s have a look of the camera model and its relative parameters.

4.1.1 Camera model and parameters

Most of the models of the cameras can be considered as a pinhole or perspective model, shown in Figure 44. The coordinates system OXYZ is the camera frame. O is the projection center also the lens center. OZ is the optical axis which is perpendicular to the image plane π. Their intersection o is the principle point. The distance between projection center O and the image plane is the focal length f. P is a point in camera frame with coordinates (Xc,Yc,Zc) and p is its projection on the image plane. The coordinates of p is (x,y,f) in camera frame and (x_{im},y_{im}) in the image plane [Trucco98].

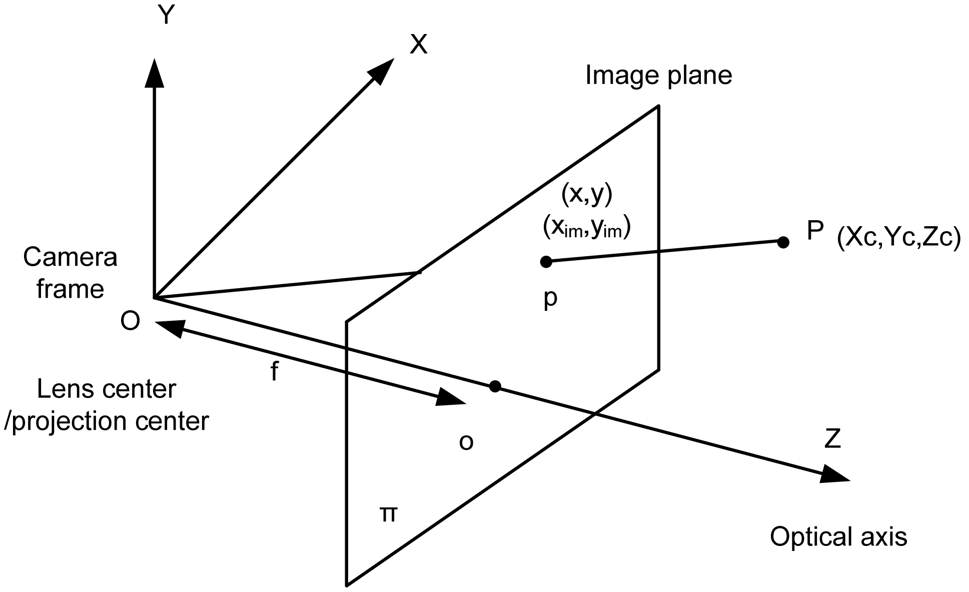


Figure 44 The perspective camera model

The perspective model of the camera is given by

$$x = f\frac{Xc}{Zc} \qquad (4.1)$$

$$y = f\frac{Yc}{Zc} \qquad (4.2)$$

The camera model gives the relation between a point in space and its projection in the image plane in the camera reference frame. However, the camera reference frame is invisible and intangible and mostly unknown. Usually it is only used as a intermediate frame to form a transformation between a world reference frame and the image frame.

The intrinsic parameters links the pixel coordinates of an image point to the coordinates of the same point in the camera reference frame.

The extrinsic parameters define the location and orientation of the camera reference frame to a known world reference frame.

Given the intrinsic and extrinsic parameters, a transformation between the point in space and its projection on the image plane is determined.

The intrinsic parameters include:

fx focal length in horizontal pixels

fy focal length in vertical pixels

(ox, oy) principle point coordinates

α_c skew coefficient, the angle between the x and y pixel axes

kc distortion coefficients

The transformation between the camera reference frame and the image frame can be expressed as

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \alpha_c f_x & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ 1 \end{bmatrix} \tag{4.3}$$

In most cases, the skew coefficient is 0 which is the case for the cameras in our system.

The extrinsic parameters include:

R 3×3 rotation matrix

T 3D translation vector

The transformation between the camera reference frame and the image frame can be expressed as

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \tag{4.4}$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \text{ and } T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}.$$

Plugging (4.4) into (4.3) gives

$$x_{im} - o_x = f_x \frac{r_{11}X_w + r_{12}Y_w + r_{13}Z_w + T_x}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + T_z} \tag{4.5}$$

$$y_{im} - o_y = f_y \frac{r_{21}X_w + r_{22}Y_w + r_{23}Z_w + T_y}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + T_z} \tag{4.6}$$

The above equations form a transformation between a point in world reference frame and its projection in image frame. However, given the pixel coordinates of the image projection and the intrinsic and extrinsic parameters, there are still three unknowns (X_w, Y_w, Z_w) in two equations, so no real solution for the coordinates exists in the world reference frame. Thus two stereo cameras are necessary to get the 3 D coordinates.

The parameters of a stereo system

The parameters of a stereo system also include the intrinsic parameters and extrinsic parameters. The intrinsic parameters are the same as the definiation above. The extrinsic parameters refer to the relative position and orientation between the two cameras, which is expressed as

$$\begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} = R_r^l \begin{bmatrix} X_l \\ Y_l \\ Z_l \end{bmatrix} + T_r^l \tag{4.7}$$

$$R_r^l = \begin{bmatrix} r_{11s} & r_{12s} & r_{13s} \\ r_{21s} & r_{22s} & r_{23s} \\ r_{31s} & r_{32s} & r_{33s} \end{bmatrix} \text{ and } T_r^l = \begin{bmatrix} T_{xs} \\ T_{ys} \\ T_{zs} \end{bmatrix}.$$

4.1.2 3D reconstruction

Assume the world reference frame coincides with the left camera reference frame, that is $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, thus the relation between the point in world reference frame and its projection becomes

$$x_{iml} - o_{xl} = f_{xl} \frac{X_w}{Z_w} \tag{4.8}$$

$$y_{iml} - o_{yl} = f_{yl} \frac{Y_w}{Z_w} \tag{4.9}$$

The rotation matrix between the world reference frame and the right camera reference frame equals to R_r^l and the translation vector is T_r^l.

So the relation between the point in world reference frame and its projection in right camera reference frame is

$$x_{imr} - o_{xr} = f_{xr} \frac{r_{11s}X_w + r_{12s}Y_w + r_{13s}Z_w + T_{xs}}{r_{31s}X_w + r_{32s}Y_w + r_{33s}Z_w + T_{zs}} \tag{4.10}$$

$$y_{imr} - o_{yr} = f_{yr} \frac{r_{21s}X_w + r_{22s}Y_w + r_{23s}Z_w + T_{ys}}{r_{31s}X_w + r_{32s}Y_w + r_{33s}Z_w + T_{zs}} \tag{4.11}$$

Organizing the above four equations into an equation group is

$$\begin{bmatrix} f_{xl} & 0 & o_{xl} - x_{iml} \\ 0 & f_{yl} & o_{yl} - y_{iml} \\ f_{xr}r_{11s} + (o_{xr} - x_{imr})r_{31s} & f_{xr}r_{12s} + (o_{xr} - x_{imr})r_{32s} & f_{xr}r_{13s} + (o_{xr} - x_{imr})r_{33s} \\ f_{yr}r_{21s} + \left(o_{yr} - y_{imr}\right)r_{31s} & f_{yr}r_{22s} + \left(o_{yr} - y_{imr}\right)r_{32s} & f_{yr}r_{23s} + \left(o_{yr} - y_{imr}\right)r_{33s} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_{xr}T_{xs} + (o_{xr} - x_{imr})T_{zs} \\ f_{yr}T_{ys} + \left(o_{yr} - y_{imr}\right)T_{zs} \end{bmatrix} \tag{4.12}$$

f_{xl}, f_{yl}, o_{xl}, o_{yl} and f_{xr}, f_{yr}, o_{xr}, o_{yr} are the intrinsic parameters of the left and right camera respectively. R_r^l and T_r^l are the stereo extrinsic parameters. All these parameters can be obtained by camera calibration. Thus there are only three unknowns of the coordinates in the four equations, which is an overdetermined system.

In this overdetermined system, the least square method is used to find the approximate solution.

The equation (4.12) can be written in the form as

$$\mathrm{Ax} = \mathrm{b} \tag{4.13}$$

The least square method aims to minimize the difference between the left and right part of the equations, that is

$$\min_{\mathrm{x}} ||\mathrm{Ax} - \mathrm{b}|| \tag{4.14}$$

The solution [Anton05] of x is

$$\mathrm{x} = (\mathrm{A}'\mathrm{A})^{-1}\mathrm{A}'\mathrm{b} \tag{4.15}$$

As long as the point in space can be detected the pixel coordinates in both image planes, the coordinates of the point in space can be calculated.

This method is used in our stereo system to reconstruct the 3D points. By detecting the image of the LEDs in each camera frame, the 3D coordinates of the LEDs can be calculated.

4.2 The model of pan and tilt unit

The 3D coordinates of the point in space can be calculated in the left camera frame. However, the stereo camera bar stands on top of the pan and tilt unit, so the cameras rotate as the PTU rotates around its pan and tilt axes. When the cameras move, the left camera reference frame also changes. To find out the relation between the coordinates in different frames, the transformation of the frames due to the rotation of PTU has to be figured out.

The pan and tilt axes are lined out in Figure 45. The pan axis and tilt axis intersects at O, which is the center of the rotation. When the pan angle and tilt angle are zero, that is no rotation in any axes, this orientation is called initial orientation. A fixed coordinates system is aligned with the initial orientation, named PTU initial frame. The origin is the rotation center O; the pan axis coincides with the Z axis; the tilt axis coincides with the Y axis; X axis corresponds to the orthogonal coordinates system.

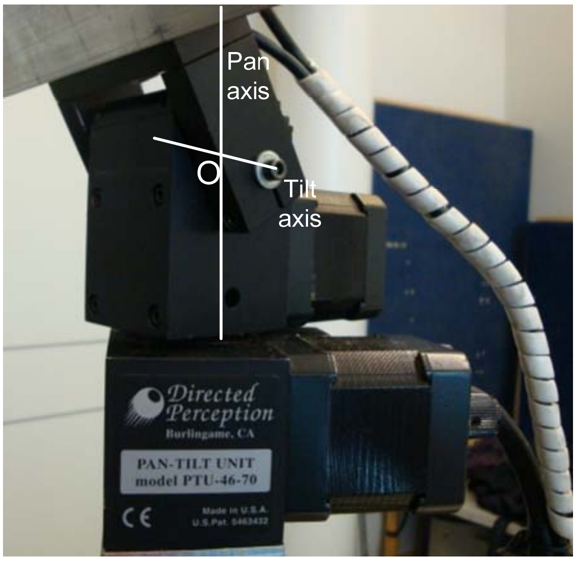


Figure 45 Pan and tilt axes of PTU

The rotation can be decomposed into two kinds of procedure:

- Rotate in pan axis then rotate in tilt axis.
- Rotate in tilt axis then rotation in pan axis.

4.2.1 Rotate in pan axis then tilt axis

This rotation sequence is depicted in Figure 46.

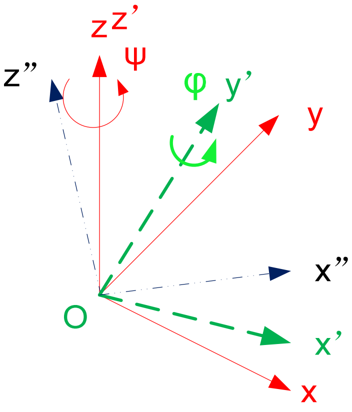


Figure 46 Euler rotation

Oxyz is the coordinates aligned with the initial orientation. After rotating ψ degree in pan axis, the new frame Ox'y'z' is generated. z' axis coincides with z axis, x axis moves to x' and y axis moves to y'. Meanwhile the tilt axis also moves from y axis to y' axis.

The rotation matrix around z axis is

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (4.16)$$

Given a stationary point P, its coordinates is P_{init} in initial frame and P_{temp} in the new frame, named intermediate frame. The transform between the two coordinates is

$$P_{init} = R_z(\psi)P_{temp} \qquad (4.17)$$

Next the PTU rotates φ degree in tilt axis, that is the y' axis, a new frame Ox"y"z" is generated named final frame. y" axis coincides with the y' axis; x' and z' axes moves to the x" and z" axes respectively. The rotation matrix is

$$R_y(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \qquad (4.18)$$

In this rotation procedure, the initial frame rotates about its axis first, then rotates about another axis in the new frame. This sequence of rotation is Euler angle rotation. For Euler rotation, the latest rotation matrix is multiplied by the right side of the former rotation matrix [John04]. Assume the coordinates of point P in the final frame is P_{move}, thus its relation with the coordinates in the initial frame is

$$P_{init} = R_z(\psi)R_y(\varphi)P_{move} \qquad (4.19)$$

4.2.2 Rotate in tilt axis then pan axis

This rotation sequence is depicted in Figure 47.

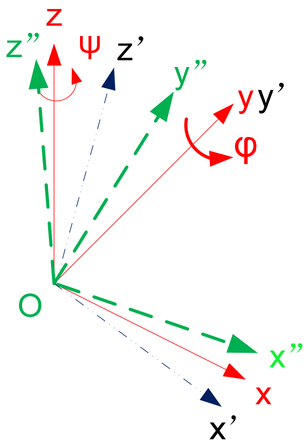


Figure 47 Fixed axis rotation

First the initial orientation frame Oxyz rotates around the tilt axis, y axis about φ degree to frame Ox’y’z’. y’ axis coincides with the y axis, x and z axes moves to the x’ and z’ axes. After rotation, the pan axis doesn’t move which still coincides with the z axis. Then the PTU rotates in pan axis, which is about the z axis in the initial frame.

In this rotation procedure, all frames only rotate around the axes in the initial frame. This rotation sequence is fixed axis rotation. In this rotation, the latest rotation matrix is multiplied at the left side of the former rotation matrix [John04]. Thus the coordinates transform of P in initial frame and the latest frame is

$$P_{init} = R_z(\psi)R_y(\varphi)P_{move} = {}^{pi}_{pm}TP_{move} \tag{4.20}$$

The equation (4.19) is identical to the equation (4.20). Therefore, the PTU model is independent of the rotation sequence; it is only related to the rotation angle in each axis.

4.2.3 Transformation between camera frames

The camera bar is installed on top of the PTU. When the PTU is at the initial orientation, we name the left camera reference frame as initial left camera frame. The cameras only move when the PTU moves. The camera reference frame has a fixed transformation ${}^{pm}_{c}T$ relative to the moved PTU frame. The transformation of the moved PTU and the inital PTU orientation is ${}^{\mathrm{pi}}_{\mathrm{pm}}\mathrm{T}$. Thus the left camera frame with respect to the initial PTU frame can also be modeled as:

$$\mathrm{P_{init}} = {}^{\mathrm{pi}}_{\mathrm{pm}}\mathrm{T}\,{}^{\mathrm{pm}}_{\mathrm{c}}\mathrm{TP_c} \tag{4.21}$$

${}^{\mathrm{pm}}_{\mathrm{c}}\mathrm{T}$ is constant and can be calculated by measurement. ${}^{\mathrm{pi}}_{\mathrm{pm}}\mathrm{T}$ is decided by the pan and tilt angles. $\mathrm{P_c}$ is reconstructed coordinates of a point in the left camera frame. $\mathrm{P_{init}}$ is the coordinates of the same point in the initial PTU frame. This equation also relates the different left camera frames relative to the PTU orientation, that is

$${}^{\mathrm{pi}}_{\mathrm{pm}}\mathrm{T_1}\,{}^{\mathrm{pm}}_{\mathrm{c}}\mathrm{TP_{c1}} = {}^{\mathrm{pi}}_{\mathrm{pm}}\mathrm{T_2}\,{}^{\mathrm{pm}}_{\mathrm{c}}\mathrm{TP_{c2}} \tag{4.22}$$

${}^{\mathrm{pi}}_{\mathrm{pm}}\mathrm{T_1}$ and ${}^{\mathrm{pi}}_{\mathrm{pm}}\mathrm{T_2}$ are the transformation for different PTU orientations; $\mathrm{P_{c1}}$ and $\mathrm{P_{c2}}$ are the coordinates of the same point in different camera frames.

As long as we know the pan and tilt angles, the coordinates of a point in the different frames are relatived by the equation (4.22). This allows all the reconstructed 3D coordinates are transformed to a uniform left camera reference frame, which provides more information about the whole localization area

4.3 Coordinate transformations

The above section builds the model between the left camera reference frames relative to the pan and tilt angles, which is able to transform all the camera frames to a uniform camera frame. However, the camera frames is invisible and untouchable, it is very difficult to get the visual concept about the reconstructed coordinates relative to the commonly used world reference frame.

In order to form a visual intuitive reconstruction result, a model between the initial left camera reference frame and a visible frame set on the lander is built.
The procedure can be depicted in Figure 48:

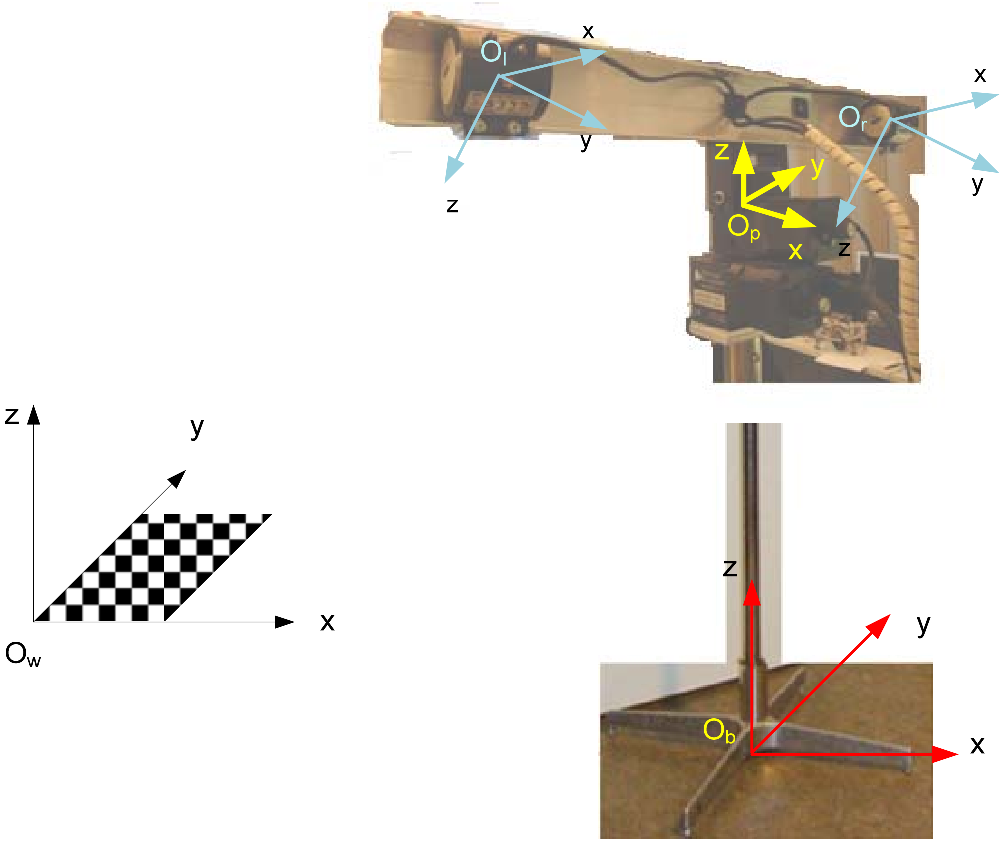


Figure 48 Coordinate frames of the whole system

In Figure 48, Owxyz is the world reference frame. $O_l$xyz and Orxyz are the left and right camera reference frames. Opxyz is the PTU initial frame. Obxyz is the base frame.

The world reference frame is attached on a chessboard, which is used for obtaining the extrinsic parameters of the left camera. Through the camera calibration, the transformation between the world reference frame and the initial left camera frame can be expressed as

$$\begin{bmatrix} P_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_w \\ 1 \end{bmatrix} = {}_w^cT \begin{bmatrix} P_w \\ 1 \end{bmatrix} \tag{4.23}$$

In this equation, R_1 and T_1 are the rotation matrix and the translation vector obtained by calibration. P_c and P_w are the coordinates of a point in the camera and the world reference frame. ${}_w^cT$ represents the transformation of world reference frame relative to the camera frame.

During the calibration, align the world reference frame with the same orientation of the initial PTU frame, thus there is only translation between these two frames. It can be written as

$$\begin{bmatrix} P_w \\ 1 \end{bmatrix} = \begin{bmatrix} I & T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_p \\ 1 \end{bmatrix} = {}^{w}_{p}T \begin{bmatrix} P_p \\ 1 \end{bmatrix} \tag{4.24}$$

where I is identity matrix. T_2 is the translation vector between the origins of the two frames. P_w and P_p are the coordinates of a point in the world reference frame and PTU frame. ${}^{w}_{p}T$ represents the transformation of the PTU frame relative to the world reference frame.

Plugging equation (4.24) into (4.23) gives

$$\begin{bmatrix} P_c \\ 1 \end{bmatrix} = {}^{c}_{w}T\,{}^{w}_{p}T \begin{bmatrix} P_p \\ 1 \end{bmatrix} \tag{4.25}$$

It builds the model of the transformation between the initial left camera frame and the initial PTU frame.

The initial PTU frame is easily perceived compared to the camera frame, whereas it is still not convenient compared to the basic frame. The basic frame has the same orientation as the initial PTU frame, but shifts the origin from the PTU rotation center to the intersection of the pole of the lander with the ground. The transform can be expressed as

$$\begin{bmatrix} P_p \\ 1 \end{bmatrix} = \begin{bmatrix} I & T_3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_b \\ 1 \end{bmatrix} = {}^{p}_{b}T \begin{bmatrix} P_b \\ 1 \end{bmatrix} \tag{4.26}$$

P_b is the coordinates in the base frame. T_3 is the translation vector between the base frame and the PTU frame. Plugging (4.26) into (4.25), the transformation between the initial left camera frame and the base frame is

$$\begin{bmatrix} P_c \\ 1 \end{bmatrix} = {}^{c}_{w}T\,{}^{w}_{p}T\,{}^{p}_{b}T \begin{bmatrix} P_b \\ 1 \end{bmatrix} \tag{4.27}$$

which can also be given as

$$\begin{bmatrix} \mathrm{P_b} \\ 1 \end{bmatrix} = ({}^{\mathrm{c}}_{\mathrm{w}}\mathrm{T}\,{}^{\mathrm{w}}_{\mathrm{p}}\mathrm{T}\,{}^{\mathrm{p}}_{\mathrm{b}}\mathrm{T})^{\mathrm{T}} \begin{bmatrix} \mathrm{P_c} \\ 1 \end{bmatrix} \tag{4.28}$$

After coordinates transformation from the initial left camera reference frame to the base frame, the reconstructed positions are intuitive for humans to observe.

The parameters in the transformation are measured in the experiments(section 6.2).

Chapter 5

Tracking

The purpose of ”tracking” in our project is for the localization. If the stereo cameras are stationary, when ROSA moves around, there can be some time that ROSA is out of the images of both cameras. The objective of tracking is to ensure that ROSA always stays in the images of both cameras at the position when localization is needed. This can be accomplished by rotating the PTU which moves the stereo cameras to perceive ROSA. The tracking procedure has two steps: detecting ROSA in the images and controlling PTU based on ROSA’s position in the images. Since the purpose of the tracking is to enlarge the localization area, and in the localization, the LEDs’ positions infer to the localization of ROSA, so the detection of ROSA in images finally becomes to the detection of LEDs in the images.

5.1 LEDs’ detection

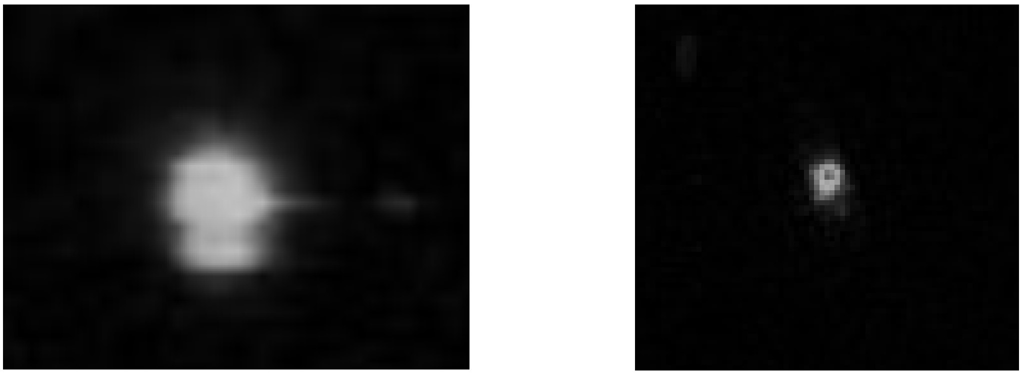
The localization procedure of ROSA is similar to the one of Nanokhod. The localization doesn’t process continuously but is only obtained when ROSA stops. In order to get dense localization information, ROSA has to traverse only a few centimeters then stop to do the localization. Based on this prerequisites, as well as to correspond with the localization method, the tracking can be realized during the procedure of localization. When ROSA stops at the desired place where the localization is required, the program will track ROSA first to make it sensed in both cameras, then localize it.

In Chapter 4, the localization algorithm in this stereo system has been explained; one indispensable information is accurate coordinates of the LEDs in both images. In the “Marker Control” program, the control of LEDs is flexible. They can be turned on and off together or separately. When they are on at the same time and far from the cameras, the LEDs partially overlap or connect in the images, bringing difficulties to detect the center of each LED and also decreasing the accuracy of the detected positions of the LEDs, and consequently the localization information.

To solve this problem, the LEDs will be turned on and off separately, so the detection of the center of the LEDs will be processed respectively. In this way the overlapping problem can be completely avoided.

The detection of all the LEDs are the same. When the rover is stationary, stereo images will be taken with all the LEDs off. These two images work as the background images which will be subtracted from the other images taken later. Then one LED is turned on and two images are taken. All the raw images are color images, to reduce the computation, they are converted to the gray level images. By substracting the background images from the images with LED on, the differential image pair is obtained. For each differential image, the same image processing is utilized to calculate the center of the LED.

During the time for taking images, except the status change of LEDs, no other change in the images, thus in the differential image the only image related to the image of LED. The differential images of the LED area look like Figure 49. The image in Figure 49 (a) is taken in the dark environment, so the image directly reflect the shape of LED at ON status. Figure 49 (b) is taken in a bright environment. Due to the non-uniform illumination of the LED in all the directions and the view point of the camera relative to the LED, the LED image does not reflect its true shape. The shapes of the outer and inner boundaries are not circular, and they are not concentric. In order to detect the center of the LED, generally three methods can be applied: finding the centroid of the outer contour, finding centroid of the inner contour and finding the connected pixels related to LED, then calculating the centroid of the connected areas. For both kind of LED images, the outer contours and the connected areas can be detected. Compared to the contours, the connected areas are more reliable to detect. Thus the center of LED is obtained by detecting the centroid of the connected area.



(a) LED in dim illumination

(b) LED in bright illumination

Figure 49 Differential images of LED in different illumination

To find the connected area of the LED, the grey image has to be converted to binary image by applying a threshold. Theoretically the images except the LED area should be pure black, however, due to the CCD noise and variation of the illumination, the non-LED area is not pure back. Thresholding can also eliminate the image noise [Rafael02]. With the binary image, the blob detection algorithm is utilized [Bloblib]. The blob detection labels all the white connected pixels as one blob. By calculating the average of all the horizontal and vertical coordinates of the connected pixels respectively, the centroid of the blob is obtained, which is considered as the LED center. The whole procedure of a single LED detection is depicted in Figure 50.

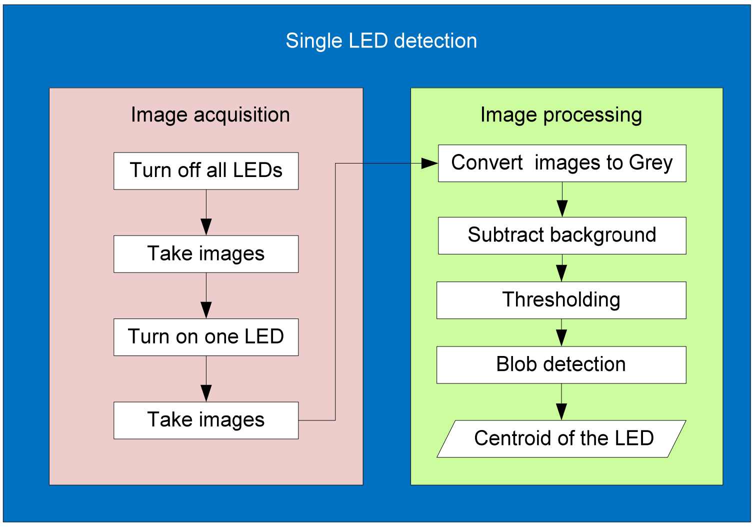
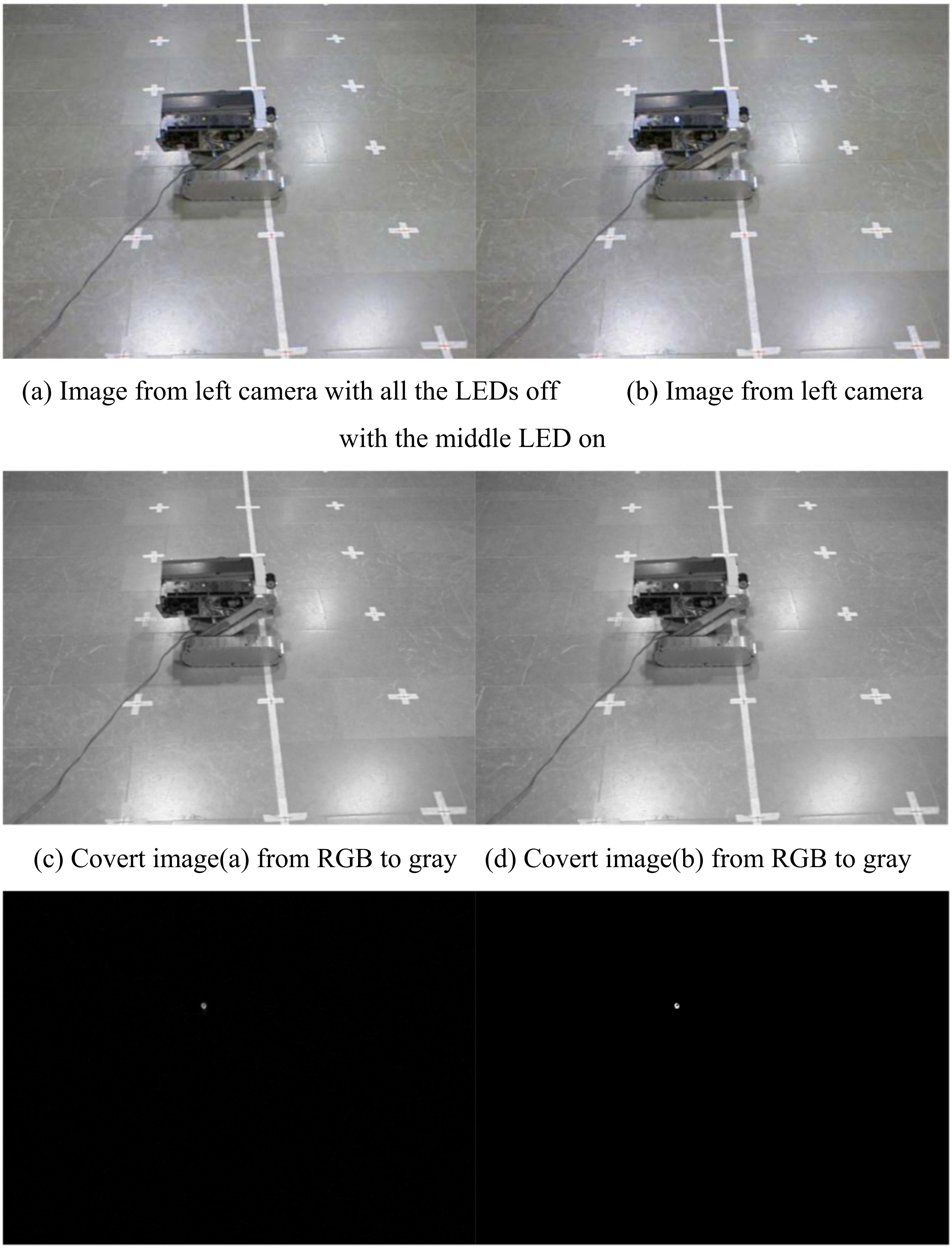


Figure 50 The procedure of the LED detection

Figure 51gives the specific example of the LED detection during the application.



(a) Image from left camera with all the LEDs off (b) Image from left camera with the middle LED on

(c) Covert image(a) from RGB to gray (d) Covert image(b) from RGB to gray

(e) The differential image between (d) and (c) (f) Image after thresholding (e)

Figure 51 Example of LED detection

5.2 PTU following

The purpose of installing PTU is to ensure ROSA can be ”seen” by both cameras. Except the field of view, the covering area depends on the orientation of PTU. By adjusting the pan and tilt angles, the stereo cameras are placed at some orientation so that ROSA is in both images. To fulfill this objective, the tracking program is made to autonomously detect ROSA and to adjust the PTU to place the stereo cameras at some orientation to cover the rover in their images.

The detecting of ROSA can be done by detecting the LED on it. The method of detecting LED has been introduced in the last section. The tracking is based on the left camera image and uses the middle LED to represent the presence of ROSA. After detecting the middle LED from the left camera, the angles to be rotated in the pan and tilt axes are calculated.

The calculation is depicted in Figure 52. The image size is 640 × 480 pixels. The inner rectangular is 100 × 80 pixels. The center of the inner rectangular coincides with the center of the image. Inside the inner rectangular is called ”safe zone”. When the LED is detected in the safe zone, no movement is requred for the PTU. Only when the LED exceeds the safe zone PTU needs to be rotated. The moving quantity in pan and tilt is determined by the distance between the center of LED and the center of the image, shown in Figure 52. The detected coordinates of the LED in the image is (x,y). Given the horizontal FOV is m degrees, the movement in pan axis is

$$\mathrm{pan} = \frac{\mathrm{x} - 320}{640} \times \mathrm{m} \tag{5.1}$$

Given the vertical FOV is n degrees, the movement in tilt axis is

$$\mathrm{tilt} = \frac{\mathrm{y} - 240}{480} \times \mathrm{n} \tag{5.2}$$

This calculation aims to move the image of the middle LED to the center of the left camera image. Since the minimum distance between the cameras and ROSA is about 1.5 meters, at this distance or furthur, if the object in the middle area of the left camera, it must be in the image of the right camera. This guarantees that after the

LED successfully moves to the middle area of the left camera, it can be observed by both cameras.

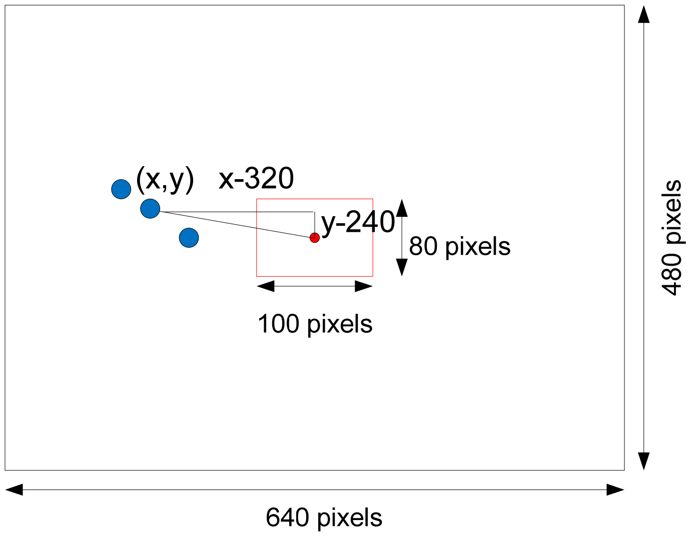


Figure 52 Illustration of the calculation of the movement

The procedure of the PTU follow-up is shown in Figure 53.

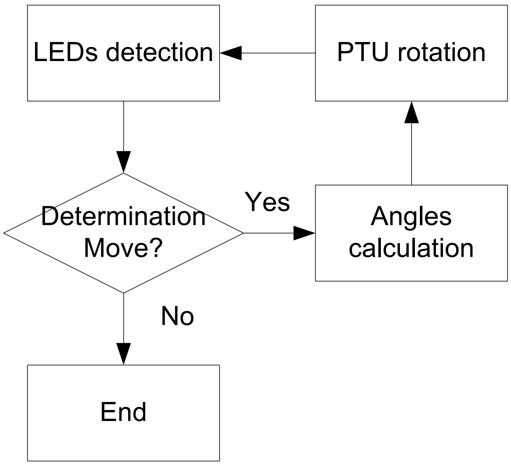


Figure 53 The procedures of PTU follow-up

The purpose of moving the LED into the central area of the image is to achieve the high accuracy of the localization. The lenses in the cameras introduce distortions in images. Usually there are two main kinds of distortion: radial distortion and tangential distortion [Bradski08]. The radial distortions are modeled as follows:

$$x = x_d(1 + k_1r^2 + k_2r^4 + k_5r^6) \tag{5.3}$$

$$y = y_d(1 + k_1r^2 + k_2r^4 + k_5r^6) \tag{5.4}$$

The tangential distortions are modeled as below:

$$x = x_d + [2k_3y_d + k_4(r^2 + 2x_d^2)] \tag{5.5}$$

$$y = y_d + [k_3(r^2 + 2y_d^2) + 2k_4x_d] \tag{5.6}$$

(x_d, y_d) is the coordinates of the distorted points, k_1, k_2, k_3, $k_4$and $k_5$are distortion parameters which have been calculated during the calibration in section 6.1.2, and $r^2 = x_d^2 + y_d^2$. As the distortion equations show, the distortion is null at the image center. As the points go far away from the center points, the distortion increases. However, the distortion model is nonlinear, which is difficult to utilize in the localization model, thus the distortion is igonored in the model. Placing the LEDs images in the central area of the image can decrease the influence of the distortion, therefore increase the accuracy of the localization.

The FOV of the stereo cameras is not provided by its datasheet, but with the calibration result, it can be calculated according to the equation [Ernest93]:

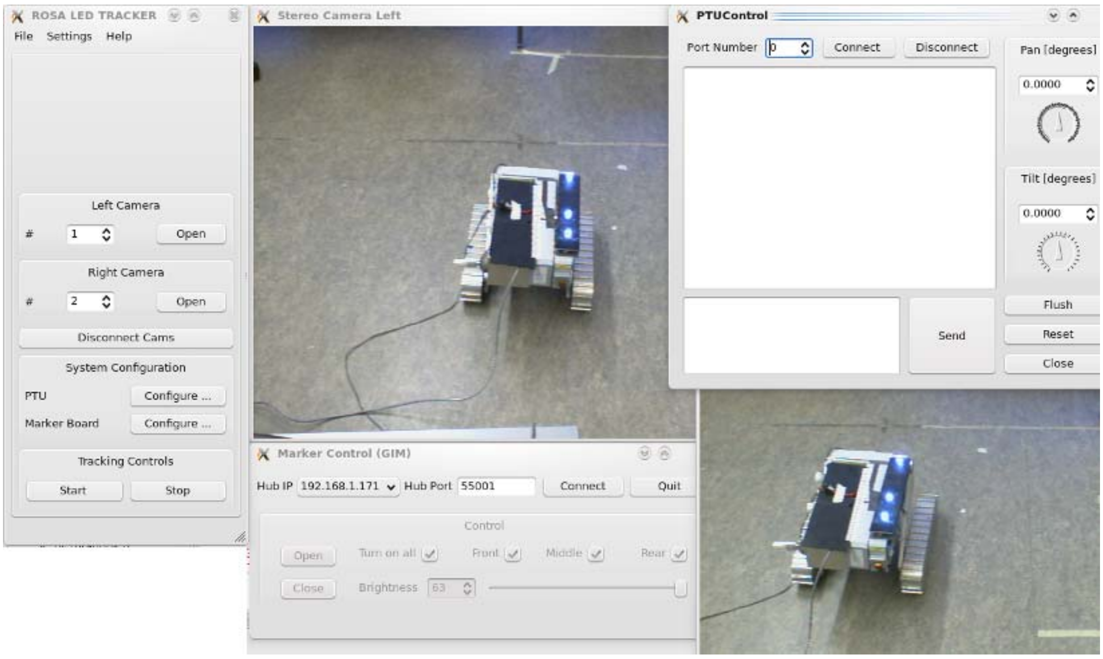
$$\alpha = 2\tan^{-1}\frac{d}{2f} \tag{5.7}$$

α is the angle of view, which describes the angular extend of the area of a scene that a lens can cover. Mostly the angle of view is interchangeable with the field of view. It can be expressed in diagonal, horizontal, or vertical. d is the dimension of the image sensor, and f is the effective focal length. In this calculation, both of them use the pixel as unit. d is 640 pixels in horizontal and 480 pixels in vertical. f is also calibrated in horizontal and vertical, the result is given in section 6.1.2. By calculation, the field of view in the equation (5.1) and (5.2) is $m = 44.8°$ and $n = 34.4°$.

5.3 Tracking software

The implementation of this tracking software ”ROSA LED Tracker” follows the procedure introduced above. Figure 54 shows the GUI when the software is running.

The left panel contains most of the operation for the software. It can start showing the video from the two stereo cameras. Each camera has a unique ID number in the Linux system. Selecting the number in the GUI will open the corresponding camera. The cameras can be manually disconnected from the software anytime in order to close or choose other cameras. The videos for the left and right cameras are displayed in separate windows.



Left Camera # 1

Right Camera # 2

PTU Configure ...

Marker Board Configure ...

Port Number 0

Pan [degrees] 0.0000

Tilt [degrees] 0.0000

Hub IP 192.168.1.171 Hub Port 55001

Turn on all ☑ Front ☑ Middle ☑ Rear ☑

Brightness 63

Figure 54 GUI of “ROSA LED Tracker” software

Before starting tracking, the PTU and the marker board should be connected to the system. This is done by the system configuration control group in the GUI. During the configuration, the GUI of “PTU control” is shown for the communication configuration. Furthermore, it can interact with the tracking procedure that allows the user to control the PTU manually in case the tracking algorithm fails. The configuration of the marker board is done in the same way as the PTU. After successfully configuring the cameras, the PTU and the marker board, the tracking procedure can be triggered to start by pressing the key ”p” on the keyboard.

Chapter 6

Experiments

6.1 Camera calibration

The camera calibration aims to get the intrinsic and extrinsic parameters of the mathematical model of the camera system. The success of the calibration has a great influence on the localization and coordinates transformation procedures. Here the Matlab calibration Toolbox is used to calibrate the cameras [Bouguet].

6.1.1 Stereo cameras calibration

The experiment setup is shown in Figure 55.

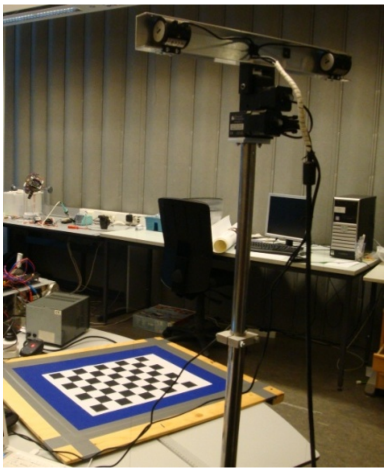


Figure 55 Stereo camera calibration

During the calibration, the chessboard is placed in different orientations. In each orientation the left camera and right camera take an image containing the chessboard separately. All the calibration images taken by the left camera are shown in Figure 56.

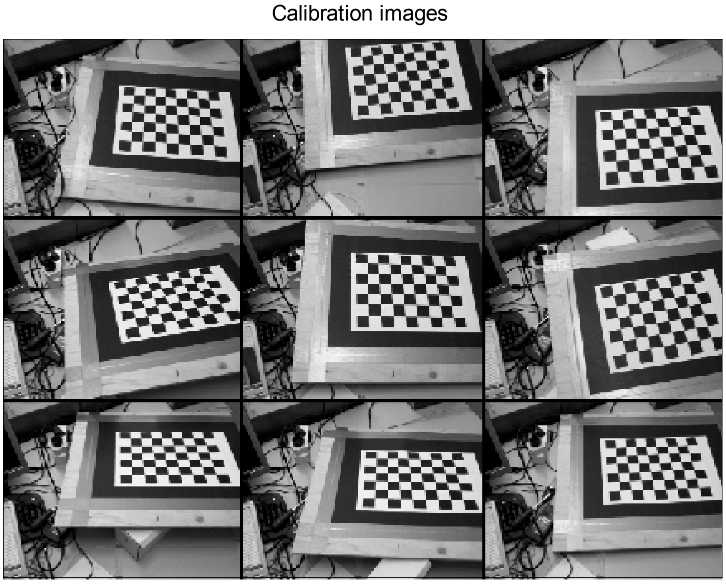


Figure 56 Calibration images for the left camera in stereo calibration

By the Matlab calibration toolbox, the intrinsic parameters of both of the cameras and their rotation matrix and translation vector are calculated, as shown in Table 1 and Table 2.

Table 1 Intrinsic parameters by stereo calibration

Intrinsic parameters	Left camera	Right camera
fx (pixels)	762.48± 38.64	776.45± 36.81
fy(pixels)	769.61± 26.04	773.56± 22.43
o_x(pixels)	338.26± 8.06	317.81± 8.44
o_y(pixels)	294.20± 47.58	241.89± 48.95
α_c	0.00	0.00
k_c	-0.04 0.10 0.005 0.007 0.000	0.03 0.09 0.003 0.005 0.000

Table 2 Extrinsic parameters of stereo cameras system

Extrinsic parameters	Value
R	$\begin{bmatrix} 0.9995 & -0.0026 & 0.0300 \\ 0.0011 & 0.9987 & 0.0505 \\ -0.0301 & -0.0505 & 0.9983 \end{bmatrix}$
T	$[-398.55997\ -0.27456\ 3.47171]^T$

The arrangement of the chessboard relative to the stereo cameras is shown in Figure 57.

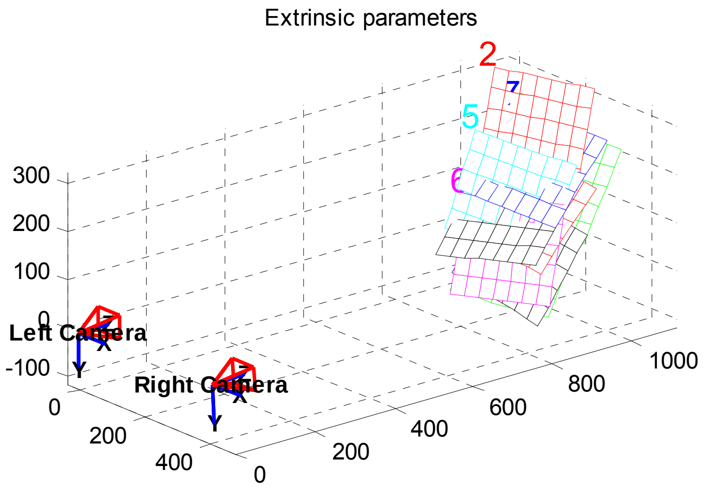


Figure 57 The arrangement of the chessboard relative to the stereo cameras

In Table 1, we can see that the uncertainties of focal length and the principle points are too large. If these values are utilized in the reconstruction, the reconstruction error will automatically increase due to these parameters.

To get a good calibration result, the calibration patterns should occupy as big areas as possible in the images. However, due to the large baseline of the stereo cameras, the overlapping areas of the images are small near the cameras. The images in Figure 56 show the optimum arrangement of the chessboard which tries to occupy the whole image as much as possible. Nevertheless, the chessboard can only occupy a specific part of the images, the same for the right camera calibration images.

In order to get accurate intrinsic parameters, independent calibrations are applied to the left camera and right camera respectively.

6.1.2 Individual camera calibration

In the individual camera calibration, the calibration proceeds independently for the left and right cameras. Each camera takes images of the chessboards to get the best coverage of the image plane. The calibration images taken by the left camera are shown in Figure 58.

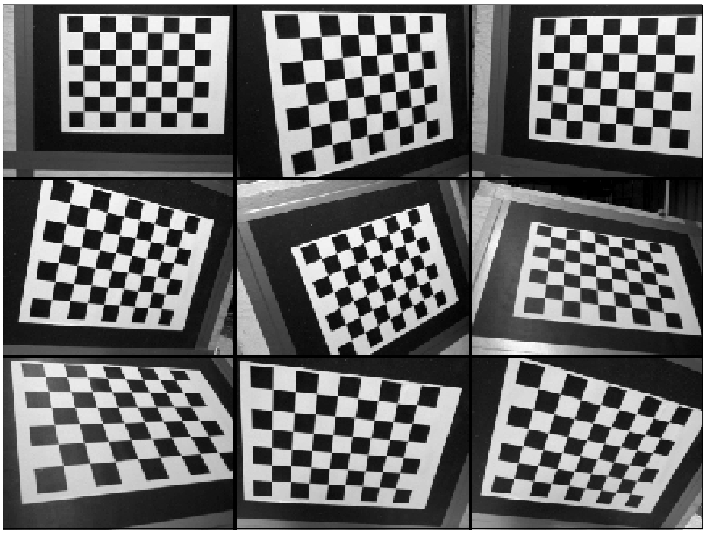


Figure 58 Calibration images of the left camera

Using these images, the intrinsic parameters of the left camera and the right camera are shown in Table 3.

Table 3 Intrinsic parameters by individual calibration

Intrinsic parameters	Left camera	Right camera
fx (pixels)	776.32 ± 2.59	768.94 ± 2.61
fy(pixels)	775.02 ± 2.61	766.85 ± 22.43
o_x(pixels)	332.12 ± 4.12	326.90 ± 8.44
o_y(pixels)	266.70 ± 2.86	233.08 ± 48.95
α_c	0.00	0.00
k_c	-0.020 -0.007 0.003 0.002 0.000	-0.015 -0.042 0.001 0.001 0.000

These parameters with small uncertainties are used for all the other calculations.

6.2 Coordinate systems transformations

This experiment follows the procedure introduced in section 4.3.

Place the chessboard with the same orientation of the PTU initial frame. The left camera takes the calibration image, shown in Figure 59. The world reference frame is marked in the image. The rotation matrix and translation vector are shown in Table 4.

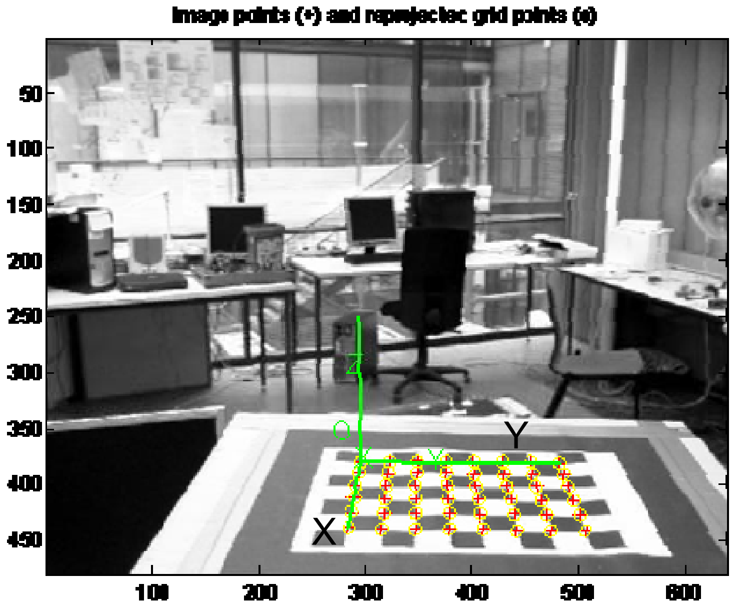


Figure 59 The world reference frame during the calculation of the initial left camera frame and the inital PTU frame

Table 4 Extrinsic parameters of the left camera relative to the world reference frame

Extrinsic parameters	Value
R_1	$\begin{bmatrix} -0.0283 & 0.9996 & 0.0040 \\ 0.2383 & 0.0029 & -0.9712 \\ -0.9708 & -0.0284 & -0.2383 \end{bmatrix}$
T_1	$[-55.82 \;\; 168.73 \;\; 1169.67]^T$

the relation between the world reference frame and the initial left camera reference frame can be expressed as:

$$P_{\mathrm{init}} = R_1 P_w + T_1 \tag{6.1}$$

There is only translation between the world reference frame and the PTU initial frame. By measurement, the translation vector in (4.24) is

$$T_2 = \begin{bmatrix} 1080 \\ 240 \\ 372 \end{bmatrix}$$

Similarly, through measurement the translation vector between the PTU initial frame and the base frame is

$$T_3 = \begin{bmatrix} 0 \\ 12.65 \\ -1511 \end{bmatrix}$$

Thus the transformation between the initial left camera reference frame relative to the base frame on the lander is

$$P_{\mathrm{init}} = \begin{bmatrix} -0.0283 & 0.9996 & 0.0040 \\ 0.2383 & 0.0029 & -0.9712 \\ -0.9708 & -0.0284 & -0.2383 \end{bmatrix} P_b + \begin{bmatrix} 173.7 \\ -1401.9 \\ -334.7 \end{bmatrix} \tag{6.2}$$

6.3 Localization

Since the localization of ROSA is obtained by detecting the LEDs which are fixed on ROSA, so the 3D reconstructed coordinates of the middle LED on the marker board are used to represent the position of ROSA.

6.3.1 Experiment setup

The experiment setup is shown in Figure 60. Tens of points on the ground are marked in front of the lander. The horizontal direction of the marks is parallel with the edge of the camera bar when the PTU is at the initial orientation. Wireless communication is utilized between ROSA and the ground station computer. The coordinates of the marked points on the ground are shown in Figure 61.

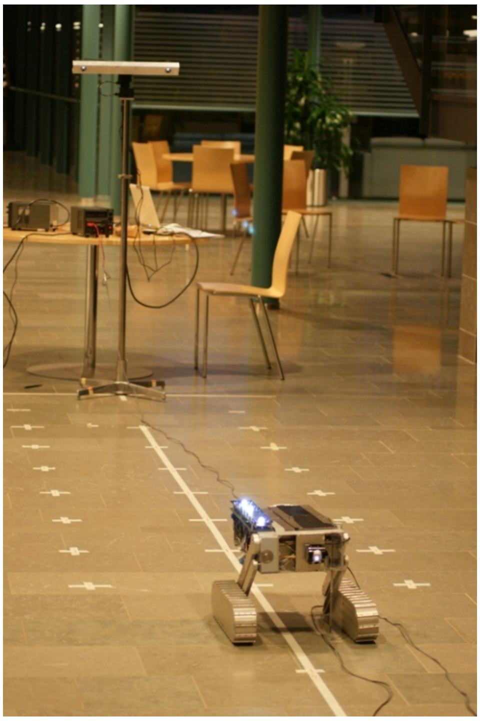


Figure 60 Localization experiment setup

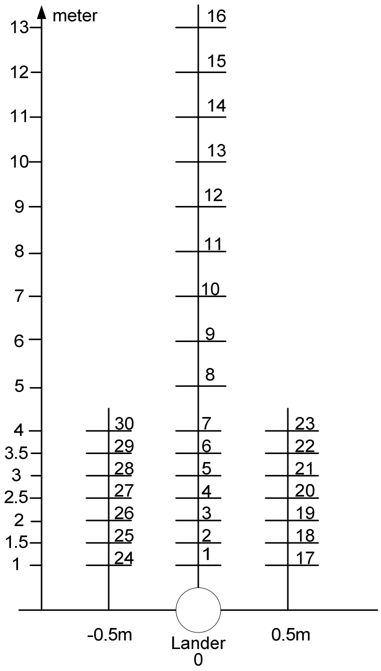


Figure 61 The arrangement of the marking points on the ground

In the figure, each crossing is labeled by a number to distinguish from the others. Totally 30 points are marked. The markers start at 1 meter in front of the lander. Besides the line right ahead the lander, there are two lines parallel to it. The line at the left side is assumed in the negative direction; the other one is in the positive direction. Both of them are 0.5m away from the middle line. In the range of 1m to 4m in the vertical direction, points are marked at each 0.5m along all three lines. In the further area, the points are marked every 1 meter along the middle line.

6.3.2 Covering area and accuracy test

As the stereo cameras can rotate along with the PTU, the observation areas are enlarged considerably. The experiment aims to find out the limitation of the localization area within which the error of the localization is tolerant.

ROSA travels along the line which is right ahead the lander, stopping at the point-number from 1 to 16. The pan angle remains zero degree during the whole experiment; the tilt angle starts at -45 degree. As ROSA moves far from the lander, the tilt angle is adjusted to -30°, -15° and 0° to make sure the rover stays in the view of both of the cameras. At each point, the centers of the three LEDs are detected on the images and are reconstructed to 3D coordinates with respect to the left camera reference frame. With different PTU orientation, the left camera reference frames are also different. In each camera frame, calculate the distance between the reconstructed 3D coordinates of the middle LED when ROSA is at two adjacent points. Compare the calculated distance with the true distance, the errors of the localization is obtained relative to the approximate distance between the lander and the ROSA.

For example, with the tilt angle -45°, two points-No1 and No2 are visible in this left reference frame. The distance between these two points is 0.5m. When ROSA stops at these two points, reconstruct the coordinates of the middle LED in the left camera frame. Through calculation, the measured distance by stereo vision is 499.5mm, thus the error is 0.5mm at point No2, which is 1.5 m away from the lander.

Table 5 shows the point number, the tilt angle when the point is observed by both cameras, the true distance on the ground between every two adjacent points, the calculated distance between the adjacent points in the same frame, the errors between the true and the calculated distance, the error rate which is the ratio of the error to the true distance, and the distance between each point and the lander.

Table 5 The experiment data for the covering area and accuracy test

Point No	tilt(degree)	true distance (m)	calculated distance (mm)	Error(mm)	Error rate (%)	Distance between the point and the lander(m)
1	-45					1
2	-45	0.5	499.5	-0.5	-0.1	1.5
3	-30	0.5				2
4	-30	0.5	484.7	-15.3	-3.06	2.5
5	-15	0.5				3
6	-15	0.5	485.1	-14.9	-2.98	3.5
7	-15	0.5	469.5	-30.5	-6.1	4
8	-15	1	962.7	-37.3	-3.73	5
9	-15	1	911.0	-89.0	-8.9	6
10	-15	1	842.2	-157.8	-15.78	7
11	0	1				8
12	0	1	1040.0	40.0	4	9
13	0	1	776.7	-223.3	-22.3	10
14	0	1	1060.1	60.1	-6	11
15	0	1	436.1	-563.9	-56.39	12
16	0	1	1012.7	12.7	-1.27	13

In the table, we can see that at 15th point, the error rate is 56.39% which is abnormal compared to its adjacent points, thus this point is considered as an exceptional point and not included in the below analysis.

Figure 62 shows the error rate of distances calculated by stereo vision relative to the distance between the point and the lander. Near the lander, the error rate is about 0.1%, which almost keeps the increasing tendency till about 15% at the point 7 meters away. After that, no general laws exist to expect the change of the errors. However, except two points, the rest are all less than 10%.

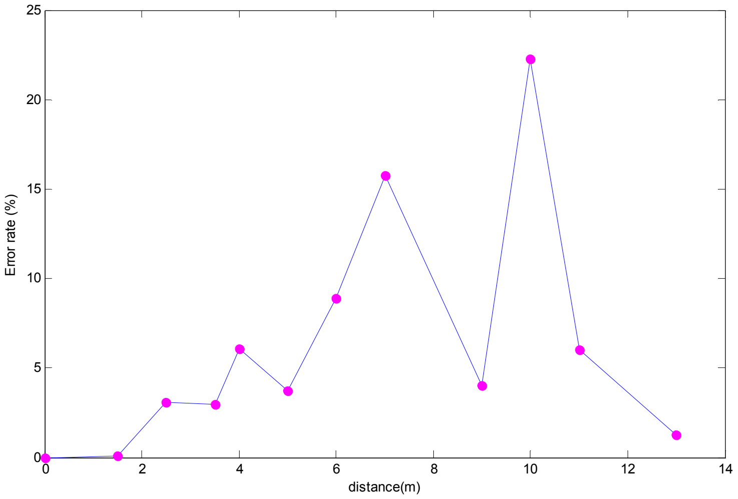


Figure 62 Error rate of distances of the movement of ROSA calculated by stereo vision relative to the distance between the point and the lander

The above experiment shows the localization accuracy with respect to the large movement, such as 0.5 meter and 1 meter. However, at each point, the heading of ROSA can only be calculated by detecting at least two LEDs. The accuracy of the reconstruction of the three LEDs determines the accuracy of the heading, which is reflected by the accuracy of the calculated distances between the LEDs by stereo vision.

The true distance between the middle and rear LEDs is 80mm and 170mm between the middle and front LEDs .

Table 6 shows the point number, the calculated distances between the middle LED and the front and rear ones, also the error rates correspondingly. The error rate is the ratio of the calculated error to the true distance.

Table 6 The distance between the LEDs calculated by stereo vision

Point No	Calculated distance between the middle and rear LEDs(mm)	Error (mm)	Error Rate (%)	Calculated distance between the front and middle LEDs(mm)	Error (mm)	Error Rate (%)
1	78.0	-2	-2.49	168.1	-1.9	-1.06
2	77.5	-2.5	-3.07	167.3	-2.7	-1.58
3	76.0	-4	-4.89	167.0	-3	-1.75
4	76.99	-3.01	-3.75	166.3	-3.7	-2.12
5	77.1	-2.9	-3.55	167.0	-3	-1.75
6	77.1	-2.9	-3.59	165.8	-4.2	-2.45
7	77.6	-2.4	-2.90	164.5	-5.5	-3.19
8	83.1	3.1	3.95	165.1	-4.9	-2.85
9	75.9	-4.1	-5.01	173.3	3.3	1.97
10	83.3	3.3	4.17	162.0	-8	-4.70
11	79.6	-0.4	-0.38	162.2	-7.8	-4.53
12	149.8	69.8	87.34	197.3	27.3	16.09
13	138.2	58.2	72.82	314.6	144.6	85.07
14	144.0	64	80.02	156.1	-13.9	-8.17
15	180.8	100.8	126.10	163.3	-6.7	-3.92
16	282.5	202.5	253.14	182.1	12.1	7.11

As seen in the table, from the 12th point, the errors increase sharply, so the calculated results have lost the meaning to represent the true distances between the LEDs. From the first point to the 11th, the error rate relative to the distance between ROSA and the lander is shown in Figure 63.

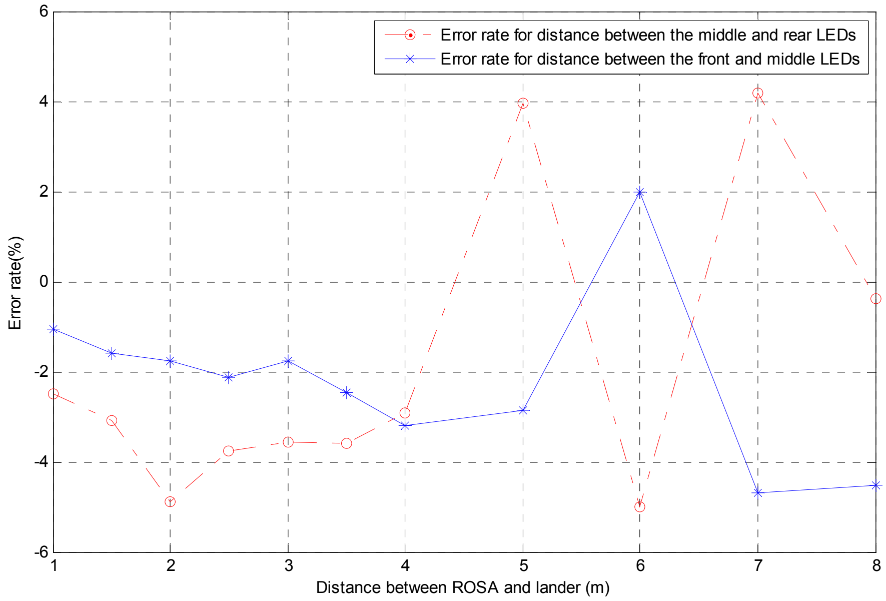


Figure 63 Error rate of distances between LEDs calculated by stereo vision relative to the distance between the point and the lander

The above figure shows that the error rate is in the range of 5% till 8 meters away from the lander. It increases as ROSA moves far from the lander in general. The error rate reflects the ability of the stereo vision of the lander to distinguish the closely lying objects in an image.

Calculation of the inclination and heading of ROSA

According to the above experiments, the localization can be obtained by the reconstruction of the 3D position of the middle LED. Besides the localization, the heading and inclination of ROSA can also be calculated by detecting at least two LEDs. This procedure is depicted in Figure 64. After reconstruction of two LEDs in the left camera frame Olxyz, the coordinates of the two LEDs can be transformed to a reference frame. Usually for calculating the heading and inclination, a frame related to the base of the lander is chosen as the reference frame, for example, the base frame Obxyz. The XY plane of the base frame is the ground surface.

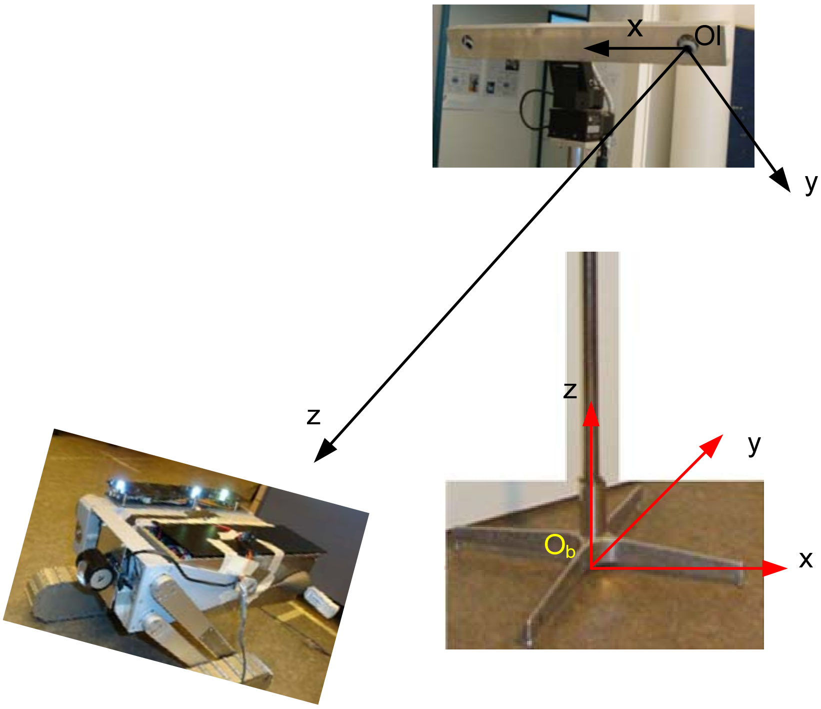


Figure 64 Inclination of ROSA

The reconstructed coordinates of the two LEDs form a vector in 3D coordinate frame. This vector indicates the heading and inclination of ROSA in the reference frame, shown in Figure 65. First the vector $\overrightarrow{OD}$ is transformed to the origin of the reference frame. The projections of $\overrightarrow{OD}$ on the X, Y and Z axes are $\overrightarrow{OA}$, $\overrightarrow{OB}$ and $\overrightarrow{OC}$ respectively. α is the included angle between the vector $\overrightarrow{OD}$ and its projection $\overrightarrow{OE}$ on the XY plane, which also represents the inclination of ROSA. β is the included angle between $\overrightarrow{OE}$ and the X axis, which indicates the heading of ROSA. Thus the inclination and heading of ROSA can be obtained by reconstruction of two LEDs in the reference frame. However, since the LEDs are lined parallel to the Y axis on the coordinate frame of ROSA, the rotation of ROSA around its own Y axis or axes parallel to the Y axis can not be detected.

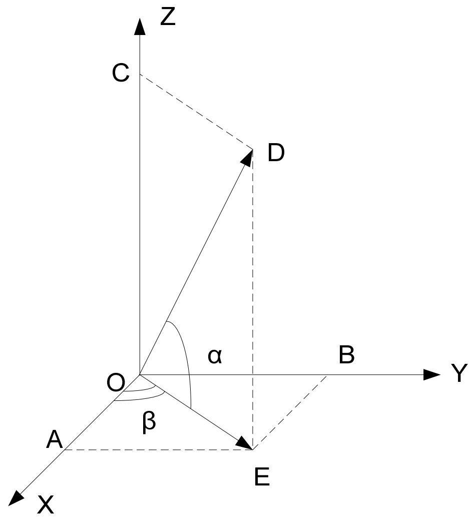


Figure 65 Inclination and heading schematic

6.3.3 Integration of the localization result

The reconstruction of LEDs is carried out in several different left camera frames. To utilize all the localization data in different frames, the transformation among the frames has to be built. Usually by converting all the coordinates in different frames to a common frame the localization information is widely enriched. To test the accuracy of localization after transforming into one common frame, the same points are used to find the errors after the transformation.

As introduced in Chapter 4, the extrinsic parameters of the camera determine the transformation between the camera frame and the world reference frame which is determined by the chessboard used for the calibration. Keeping the chessboard in the same place and taking calibration images with different PTU orientations, the transformation between the world reference frame and two different camera frames can be found respectively.

In this experiment, six points near the lander are reconstructed in two different camera frames respectively. In both frames the tilt angle is -30°; the pan angles are 0° and 10°. The calibration images marked the world reference frame are shown in Figure 66 and Figure 67.

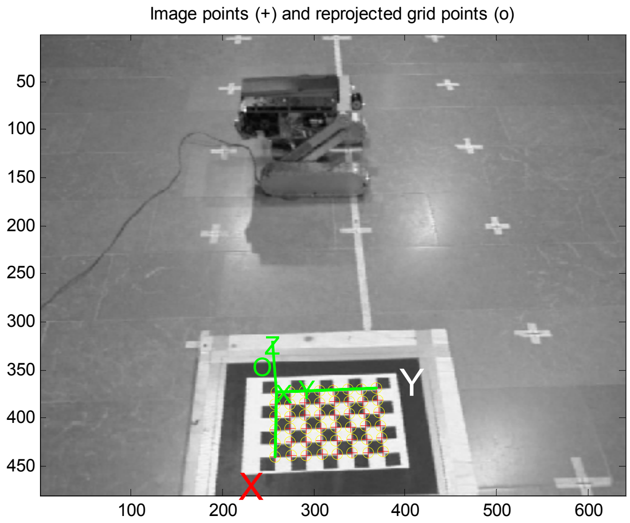


Figure 66 Calibration image with 0 degree in pan and -30 degree in tilt

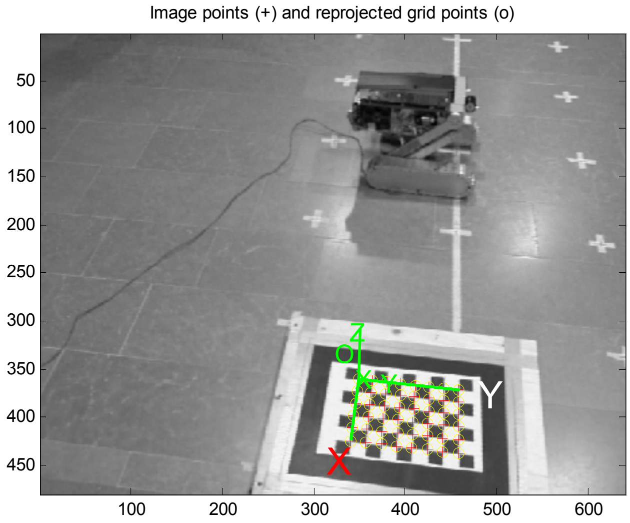
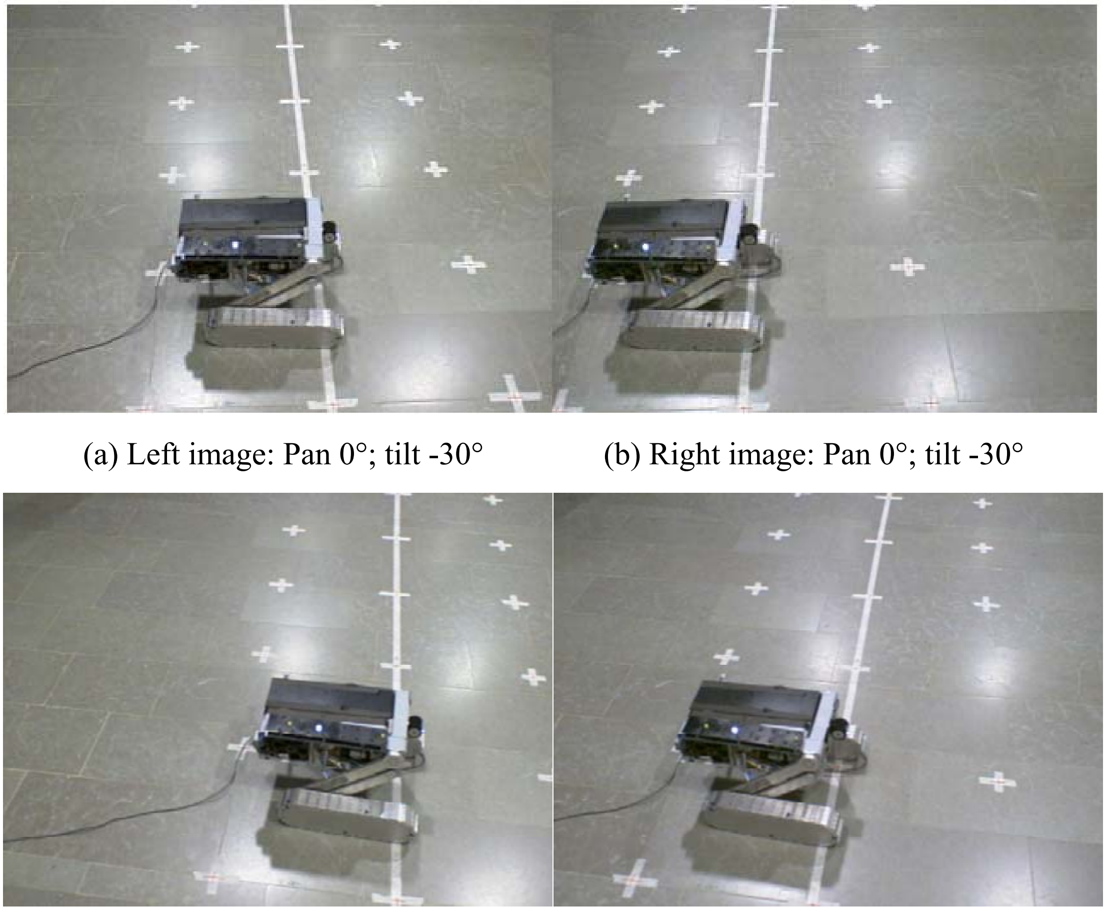


Figure 67 Calibration image with 10 degree in pan and -30 degree in tilt

There are six points tested in this experiment. Figure 68 shows the images taken at one point with the middle LED on. (a) and (b) are images taken by the left and right cameras when the pan angle is 0° and tilt angle is -30°; (c) and (d) are images taken by the left and right cameras when the pan angle is 10° and tilt angle is -30°.



(a) Left image: Pan 0°; tilt -30°

(b) Right image: Pan 0°; tilt -30°

(c) Left image: Pan 10°; tilt -30°

(d) Right image: Pan 10°; tilt -30°

Figure 68 Images at one point for localization integration

The points used for the calibration are shown in Figure 69. After reconstructing the six points in two camera frames, covert the coordinates into the world reference frame according to the equation (4.23). The result is shown in Appendix A. The distance between the two coordinates in the world reference frame of the same point reflects the accuracy of the localization integration, named transformation error. They are written besides the points, enclosed in the boxes. The distance between the adjacent points is labeled along the line. The underlined data is calculated in the frame with pan angle 0 degree, the data in the opposite side along the line is calculated in the frame with pan angle 10 degree. The average error of these distances is about 7mm, however, the error of the transformation at each point is in the range of 70mm to 100mm, more than 10 times of the distance error in each frame. Thus the transformation errors are not only caused by the error of reconstruction, but also the errors of extrinsic parameter calculation. As a result, a precise transformation among the frames is one of the key factors to get an accurate integration of localization information.

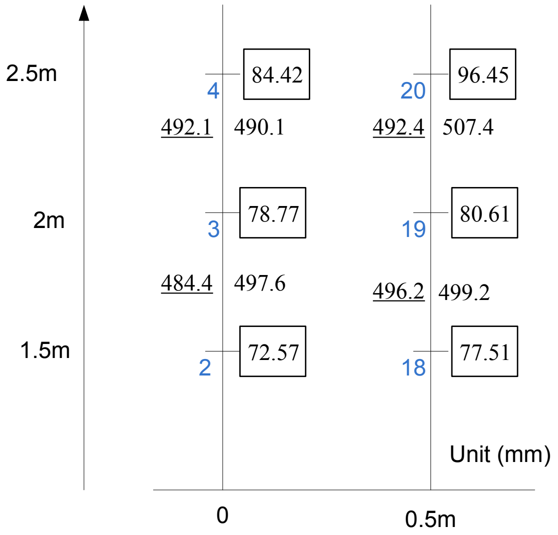


Figure 69 The coordinate transformation error test

6.3.4 Repeatability

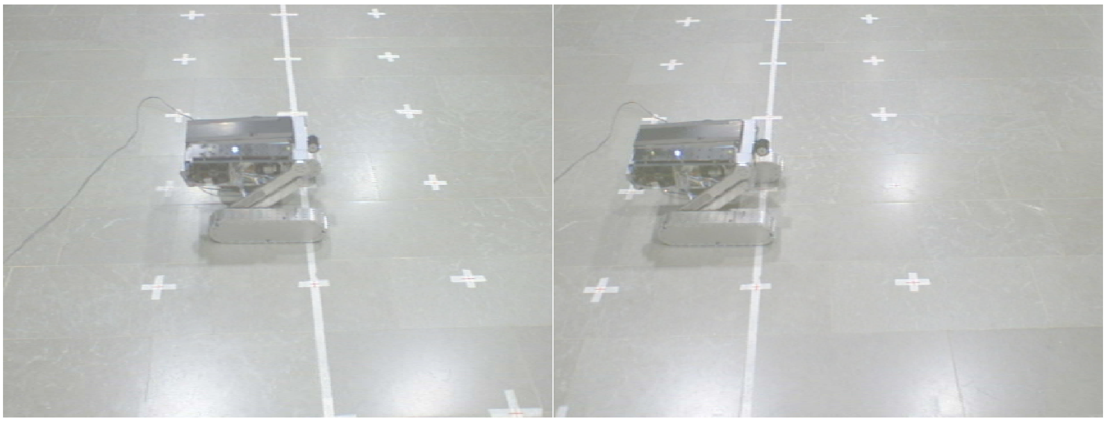
The repeatability is also one important property for a stereo system. To test the repeatability, at each point the localization is calculated for 5 times. The mean value of the 5 coordinates is considered as the true value of the reconstructed coordinates. The distances between the mean coordinates and the other 5 measured coordinates can be calculated. The repeatability can be evaluated by the standard deviation of the distances. Standard deviation measures the variability of a data set. A low standard deviation means the data is close to the mean value.

The standard deviation can be calculated by the equation (6.3).

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{6.3}$$

where μ is the mean value of the variable.

In this experiment, three points are chosen for the localization repeatability. All the points are shown in Figure 70. Figure 70(a) shows the images from the left camera at the three points, and (b) is from the right camera. The point number and the standard deviation of the distances between the measured and the mean coordinates are shown in Table 7.



(a) Three positions covered from left camera (b) Three positions covered from right camera

Figure 70 Positions for the repeatability test

Table 7 Standard deviation of the test data for repeatability

Point Number	2	3	4
Standard deviation (mm)	0.28	0.21	0.29

The standard deviation is with 0.3mm, thus the repeatability of the stereo system is very good.

6.4 Tracking experiment

The tracking experiment aims to find out the limitation of the tracking area, the robust of the tracking program and the tracking performance.

The experiment is carried out in a lobby. ROSA goes straight away from the lander and stops at the marker points, shown in Figure 71.

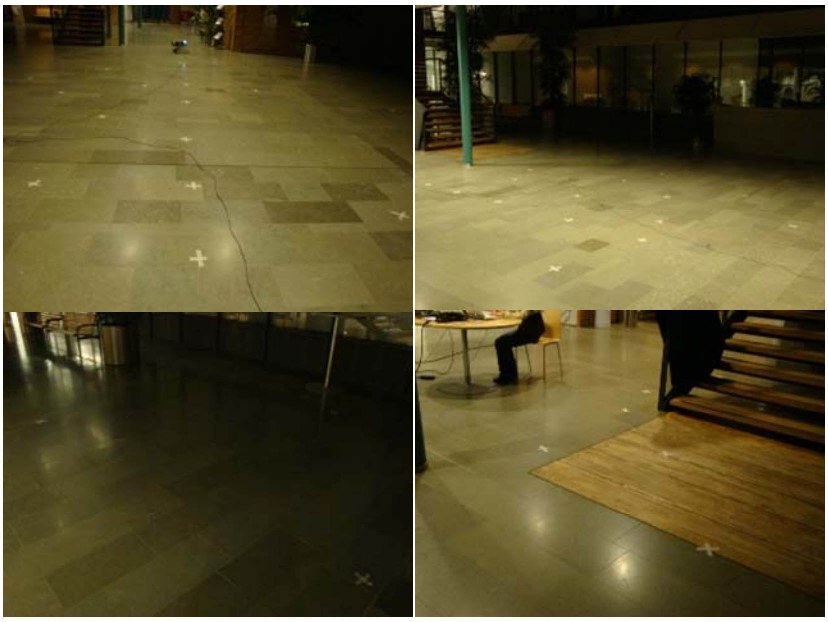


Figure 71 The marked places where ROSA stops

As the rover moves far away from the lander, the LED image gets smaller and smaller. When the LED can't be detected from the images, the limitation of the tracking area is reached, which is the maximum radius of the tracking area. Then ROSA turns and travels along the wall of the lobby to test the tracking performance till it comes to the rear of the lander. The tracking stops when the rotation in pan axis reaches the limitation of the turning angle, which is 160°. The above route is called tracking route1, shown in Figure 72.

Then ROSA moves inside the space of route1 and travels towards the other side of the lobby which is the tracking route2.

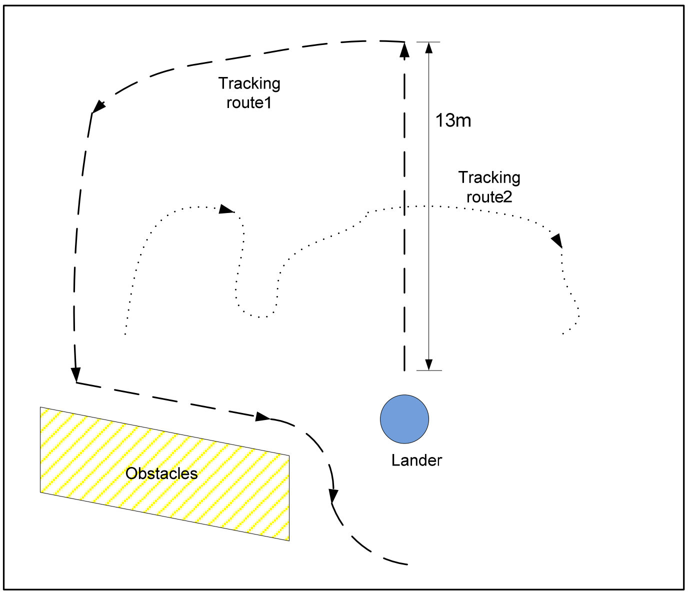


Figure 72 Tracking routes

In route 1, ROSA stops at 24 places and 13 places in route 2. At each point, the program detects the middle LED in the left camera, reads the current pan and tilt angles and calculates the rotation angles in pan and tilt axes, and sends a command to the PTU controller. The recorded data is shown in Appendix B. The largest tracking radius is 13 meters in the experiment.

The centers of middle LED images detected in the left camera for both routes are drawn in Figure 73. Before tracking, the centers of LEDs are distributed around the images.

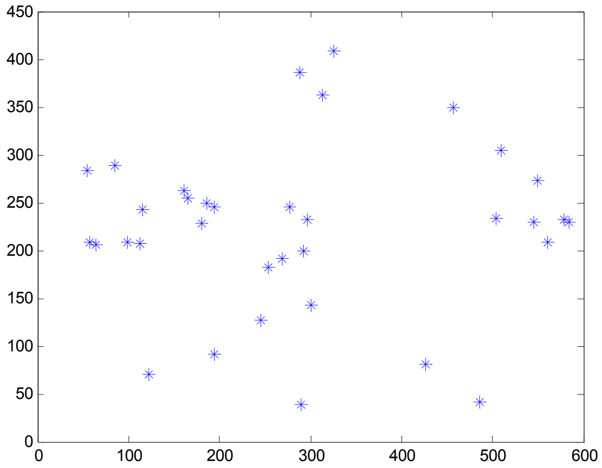


Figure 73 Centers of middle LED detected in the left camera before tracking

After tracking, the middle LED is detected in both cameras. The distribution of the LED centers in left and right images are shown in Figure 74 and Figure 75.

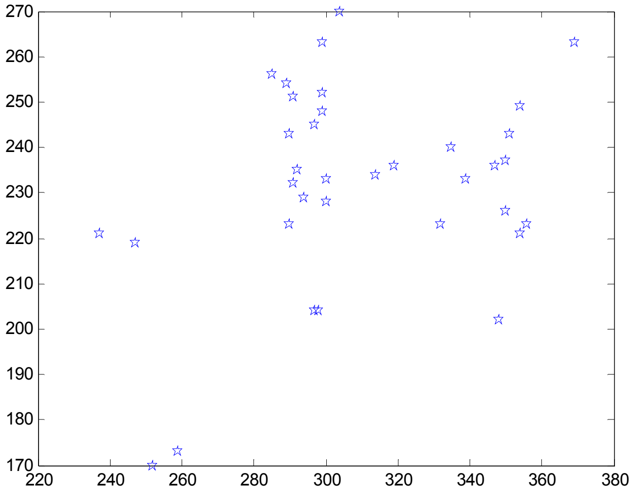


Figure 74 Centers of middle LED detected in the left camera after tracking

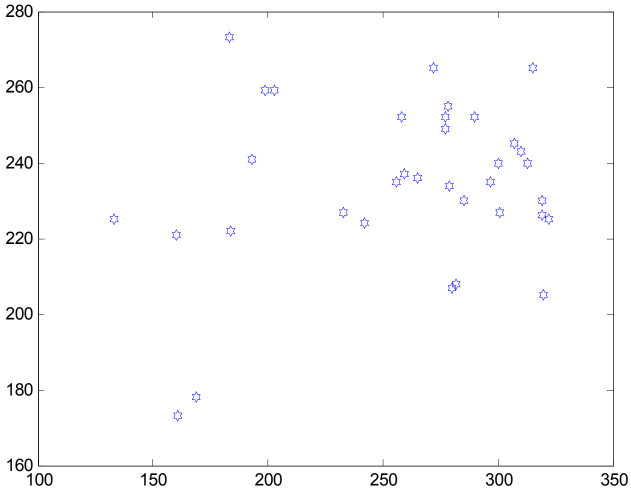


Figure 75 Centers of middle LED detected in the right camera after tracking

As seen in the figures, the centers of LED can be detected in both cameras, which fulfills the requirement of tracking. The distribution of the LEDs concentrates in the middle area of the images which is in accord with the algorithm.

During the tracking, the pan and tilt angels are shown in Figure 76 and Figure 77 for route 1 and route 2.

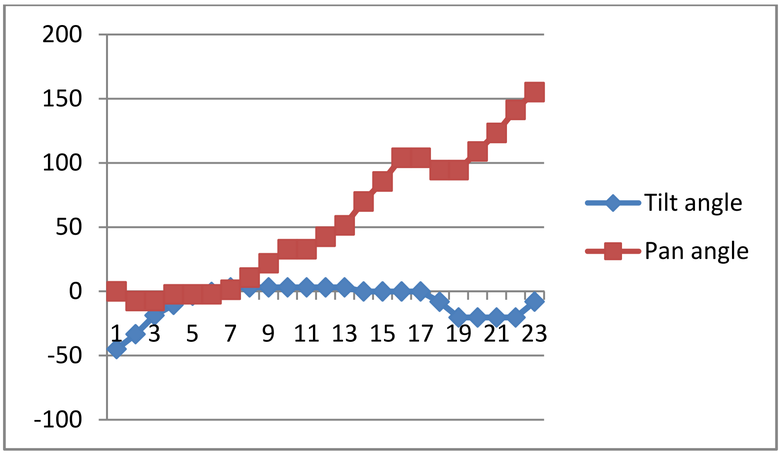


Figure 76 Pan and tilt angles in tracking route 1

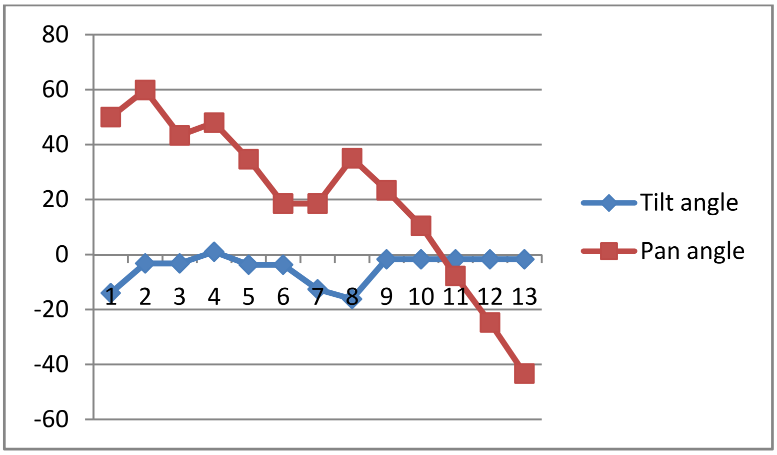


Figure 77 Pan and tilt angles in tracking route 2

During the tracking, there is no failure at all. The performance of the tracking program is very stable.

After the installation of the PTU, the covering area of the stereo cameras is a sector with the radius 13 meters and the minimum angle of sector 320°. Compared to the covering area with the fixed orientation of the stereo cameras (tested in the experiment 6.3.2), the covering area is significantly enlarged.

Chapter 7

Conclusion and future work

In the planetary exploration, rovers are widely used to gather information about the planets. In this thesis, all the efforts are put for improving a prototype of the Mars explorision rover-ROSA.

The software for the mobility control of ROSA is made. The software is running on Linux in the ground station computer, used by the operators to control the motion of ROSA. There are several different operation modes to drive, meanwhile getting the status of the rover. The operation is largely simplified by a graphical user interface. The communication between the software and ROSA is through GIMnet, which is a communication infrastructure designed for robot applications.

Previously, the stereo cameras are fixed on the lander, which limits the covering areas for the localization. To enlarge the localization area, a pan and tilt unit is installed on the lander, which has two degrees of freedom - the pan and tilt axes. When PTU rotates, the stereo cameras which amounted on top of PTU also move. Thus the sensing areas of the stereo cameras are significantly enlarged. To get the localization information of ROSA, the stereo cameras are calibrated and a model for reconstructing the 3D position of the LED is built. In this model, the least square methods are adopted to solve the overdetermined system with four equation. Compared to finding epipolar line and rectifying images, this model significantly reduces the computation. The localization experiments show the accurancy of the position with respective to the distance between the lander and the rover.

The installation of PTU enlarges the localization area. By adjusting the pan and tilt angles at suitable poisition, ROSA can be covered by the stereo cameras, which is the fundamental requirement for localization. To make this procedure autonomous, a program is made which can detect the ROSA and calculate the angular movement in pan and tilt axes, and to send the commands to the PTU controller to adjust the positions. After this procedure, ROSA can be "seen" by both cameras. The goal is to try to keep ROSA in the middle part of the images. For adjusting the PTU orientation, a program with GUI is made, which simplifies the control of PTU. To detect ROSA, a marker board with three LEDs is fixed on top of it and connected with its on-board computer. A program with GUI runs in the ground station computer to control the status and brightness of the LEDs through GIMnet. The tracking software integrates the PTU and LED programs to detect ROSA and adjust PTU. The software also provides the video feedback of the two stereo cameras for better observation.

Closer observations of interesting areas and objects can be made and recorded by the camera installed in front of ROSA. When ROSA traverses, the images are taken and transferred back to the ground station for the research of the Martian surface.

Future work

Currently, the detection of ROSA depends on the detection of the LEDs. During the procedure, ROSA has to remain stationary. If the tracking procedure can be independent of the LEDs and in real time, the time for communicating with the marker board during each tracking would be saved.

One main achievement of this thesis is the expansion of the localization area. As shown by the experiments, the localization information at different PTU orientations still have significant errors. One key factor causing the error is the inaccurate transformation between the different camera frames, which can be solved by the PTU calibration. Since the PTU has a large number of orientations with arbitary pan and tilt angles, the calibration can be chosen for tens of orientations which can cover the whole tracking area. The accurate calibration of PTU will lead an accurate transformation between different camera frames in which the localization is implimented, in turns a meaningful localization information can be obtained in the whole observable area.

One camera has been installed in front of ROSA to observe the Martian surface. With another camera installed besides it, the two cameras can form another stereo camera systems. By the stereo vision, it could detect obstacles and change the path autonomously, which would increase the reliabity and flexbility of the motion control of ROSA.

References

[Anttila02] Anttila, M., Suomela, J., & Saarinen, J. The MicroRoSA2 Activity - Conclusion and Future Plans. 2002 .(pp. S. 19–21). ESTEC, Noordwijk,The Netherlands: 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation.

[Anton05] Howard Anton and Chris Rorres. Elementary Linear Algebra (9th ed.), John Wiley and Sons, Inc., ISBN 978-0-471-66959-3. 2005

[Bloblib] Blob extraction library.[Online;accessed 23-May-2009]

URL:http://opencv.willowgarage.com/wiki/cvBlobsLib

[Bouguet] Bouguet, J.Y. Camera Calibration Toolbox for Matlab. [Online accessed 20-Jun-2009].

URL: http://www.vision.caltech.edu/bouguetj/calib_doc/

[Bradski98] Bradski, G. R.. Computer Vision Face Tracking For Use in a Perceptual User Interface. *Intel Technology Journal Q2* . 1998

[Bradski08] Gary Bradski, Adrian Kaehler.Learning OpenCV: Computer Vision with the OpenCV Library.1st edition. O'Reilly Media. 2008

[Busch07] Busch, S. (2007). *Design and Implementation of a Stereo Vision Based Navigation System for the Micro-RoSA-2 Mars Rover*. Espoo, Finland.

[Canny86] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

[Cheng95] Y.Cheng. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Machine, (pp. 790-799). 1995

[Comani02] Dorin Comaniciu, Peter Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, pp. 603-619. 2002

[Cooper98] B. Cooper, *Driving on the Surface of Mars Using the Rover Control Workstation*. Proceedings of. SpaceOps '98, Tokyo, Japan, 2a008. 1998

[Daniel07] Daniel Molkentin. The Art of Building Qt Applications.2007 ISBN: 978-1593271473

[ESA98] ESA. Micro Robots for Scientific Applications 2 - Statement of work. 1998. *MMA/H/SOW4.98* , Issue 1.0.

[Ernest93] Ernest McCollough. Photographic Topography Industry: A Monthly Magazine Devoted to Science, Engineering and Mechanic Arts (Industrial Publishing Company, San Francisco): 399–406. 1893

[Everett95] Everett, H.R.. Sensors for mobile robots: theory and application. 1995.

[FT07] How OpenCV's Face Tracker Works. March 2007[Online; accessed 15-March-2009]

URL: http://www.cognotics.com/opencv/servo_2007_series/part_3/sidebar.html

[GIMnet] GIMnet [Online;accessed 16-March-2009]

URL:http://gim.tkk.fi/GIMnet

[GIMI] GIMI[Online;accessed 16-March-2009]

URL:http://gim.tkk.fi/GIMnet/GIMI

[Horn81] B. K. P. Horn, B. G. Schunck. Determining optical flow. Artificial Intelligence 17: 185–203. 1981

[IMP] How does the IMP work [Online; accessed 20-February-2009]

URL: http://mars.jpl.nasa.gov/MPF/imp/howdoes/how_does.html

[Jasmin] Jasmin Blanchette, Mark Summerfield. C++ GUI Programming with Qt 4 (2nd Edition).ISBN: 0132354160.

[John04] John J. Craig, Introduction to Robotics: Mechanics and Control, 3rd edn, Pearson Education, Harlow, 2004

[Landz04] Landzettel K.; Steinmetz B-M.; Brunner B.; Arbter K.; Pollefeys M.; Vergauwen M.; Moreas R.; Xu F.; Steinicke L.; Fontaine B. A micro-rover navigation and control system for autonomous planetary exploration. Advanced Robotics, Volume 18, Number 3, 2004 , pp. 285-314(30)

[Lev00] Levomäki, T.P. Control System of Micro Robot for Planetary Exploration. 2000

[Lucas81] B. D. Lucas, T. Kanade. An iterative image registration technique with an application to stereo vision. Proceedings of the 1981 DARPA Imaging Understanding Workshop (pp. 121–130), 1981.

[MaCI] MaCI[Online;accessed 04-June-2009]

URL:http://gim.tkk.fi/MaCI

[Nicholas08] Nicholas J. Tosca, Andrew H. Knoll, and Scott M. McLennan. Water Activity and the Challenge for Life on Early Mars. (30 May 2008).*Science* **320** (5880), 1204. [DOI: 10.1126/science.1155432]

[OpenCV] OpenCVWiki.[Online;accessed 05-March-2009]

URL:http://opencv.willowgarage.com/wiki/

[Pathfinder] NASA Facts:Mars Pathfinder.National Aeronautics and Space Administration, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

[PTU06] Pan-Tilt Unit (Model PTU-D46) User's Manual, Version 2.14.0, October 24, 2006.Directed Perception, Inc.

[Rafael02] Rafael C.Gonzales; Richard E.Woods. Digital Image Processing (2nd Edition). Prentice Hall. 2002.

[RCW] MFEX: Microrover Flight Experiment: Rover Control Workstation. Jet Propulsion Laboratory, California Institute of Technology and the National Aeronautics and Space Administration. [Online; accessed 20-February-2009]

URL: http://marsprogram.jpl.nasa.gov/MPF/roverctrlnav/rcw.html

[ReadyBoard] ReadyBoard™ 710 Single Board Computer Reference Manual.P/N 5001736A Revision A. Ampro Computers, Incorporated.

[Saar07] Saarinen, J., Maula, A., Nissinen, R., Kukkonen, H., Suomela, J., and Halme, A.. GIMNet - Infrastructure for distributed control of Generic Intelligent Machines. In 13th IASTED International Conference on Robotics and Applications Telematics, page 6 p. Springer, Würzburg. 2007.

[SERBY04] SERBY, D., KOLLER-MEIER, S., AND GOOL, L. V. Probabilistic object tracking using multiple features. In *IEEE International Conference of Pattern Recognition (ICPR).* 184–187. 2004.

[Smith97] Smith, P. H.. The imager for Mars Pathfinder experiment. *JOURNAL OF GEOPHYSICAL RESEARCH* , 4003-4025. 1997

[Suomela02] Suomela, J., Saarinen, J., Halme, A., Anttila, M., Laitinen, S., Kaarmila, P., et al. Micro robots for scientific applications 2 - development of a robotic sampling system. 2002.*2nd IFAC Conference on Mechatronic Systems.* Berkeley California, USA.

[Thomas99] Thomas, N., W. J. Markiewicz, R. M. Sablotny, M. W. Wuttke, H. U. Keller, J. R. Johnson, R. J. Reid, and P. H. Smith. The color of the Martian sky and its influence on the illumination of the Martian surface, *J. Geophys. Res.*, 104(E4), 8795–8808. 1999

[TRACO] TRACO POWER DC/DC Converter TEL 3Series 3W. Rev. 07/01. TRACO ELECTRONIC AG

[Trucco98] Trucco, E. and Verri, A.. Introductory Techniques for 3-D Computer Vision. Prentice Hall PTR, Upper Saddle River, NJ, USA. ISBN 0132611082. 1998

[Ukrai] Yaron Ukrainitz, Bernard Sarel. Mean ShiftTheory and Applications [Online;accessed 14-April-2009]

URL:http://www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/mean_shift/mean_shift.ppt

[Vergau01] Vergauwen, M., Pollefeys, M., Van Gool, L.J. A stereo vision system for support of planetary surface exploration. Proceedings of the Second International Workshop on Computer Vision Systems, Vol. Vol. 1 pp.298-312. 2001

[Wilcox98] Wilcox, B. and Nguyen, T.. Sojourner on Mars and lessons learned for future planetary rovers. In Proc. 28th International Conf. on Environmental Systems (ICES'98), Danvers, MA, Society of Automotive Engineers. 1998.

[William96] William, S.. The Planet Mars: A History of Observation and Discovery September.1996

[Wong08] Wong, V. *Autonomous Navigation System for MRoSA2 with Focus on Mobility and Localization.* Espoo, Finland. 2008

[Yilmaz06] Yilmaz, A., Javed, O., and Shah, M. 2006. Object tracking: A survey. ACM Comput. Surv. 38, 4, Article 13. (Dec. 2006), 45 pages. DOI = 10.1145/1177352.1177355
http://doi.acm.org/10.1145/1177352.1177355

Appendix

Appendix A

Table 8 3D coordinates for the integration of the localization result

	0 degree in Pan axis		10 degree in Pan axis		coordinates in ground frame
Point No	coordinates in left camera frame(mm)	coordinates in ground frame(mm)	coordinates in left camera frame(mm)	coordinates in ground frame(mm)	difference(mm)
2	1.0e+003 * [0.1425 -0.0358 -1.7310]	1.0e+003 * [2.5027 0.1409 2.7392]	1.0e+003 * [-0.0798 -0.0116 -1.7663]	1.0e+003 * [2.5748 0.1407 2.7473]	72.57
3	1.0e+003 * [0.1696 0.3015 -2.0777]	1.0e+003 * [2.9870 0.1396 2.7294]	1.0e+003 * [-0.1391 0.3245 -2.1181]	1.0e+003 * [3.0649 0.1385 2.7412]	78.77
4	1.0e+003 * [0.1814 0.6402 -2.4344]	1.0e+003 * [3.4788 0.1223 2.7251]	1.0e+003 * [-0.2138 0.6617 -2.4763]	1.0e+003 * [3.5621 0.1215 2.7385]	84.42
18	1.0e+003 * [-0.3690 -0.0270 -1.7664]	1.0e+003 * [2.5042 -0.3718 2.7492]	1.0e+003 * [-0.5924 -0.0647 -1.7418]	1.0e+003 * [2.5806 -0.3750 2.7620]	77.51
19	1.0e+003 * [-0.3371 0.3173 -2.1223]	1.0e+003 * [3.0004 -0.3691 2.7405]	1.0e+003 * [-0.6471 0.2792 -2.0995]	1.0e+003 * [3.0798 -0.3715 2.7543]	80.61

20	1.0e+003 * [-0.3065 0.6594 -2.4752]	1.0e+003 * [3.4928 -0.3674 2.7315]	1.0e+003 * [-0.7020 0.6255 -2.4663]	1.0e+003 * [3.5872 -0.3673 2.7511]	96.45

Table 9 Distances for the integration of localization test

Point No	0 degree in Pan axis		10 degree in Pan axis	
	Distance(mm)	Error(mm)	Distance(mm)	Error(mm)
2-3	484.4	15.6	490.1	9.9
3-4	492.1	7.9	497.6	2.4
18-19	496.2	3.8	499.2	0.8
19-20	492.4	7.6	507.4	-7.4

Appendix B

Table 10 Experiment data for the tracking route 1

Point No	Before tracking					After tracking	
	middle LED center in left camera image(pixel)	current tilt (°)	tilt movement (°)	current pan (°)	pan movement (°)	middle LED center in left camera image (pixel)	middle LED center in right camera image(pixel)
1	(427, 81)	-45	11.59	0	-7.52	(350,237)	(193,241)
2	(289,39)	-33.41	14.66	-7.52	0	(290,243)	(184,222)
3	(246,127)	-18.75	8.24	-7.52	5.20	(314,234)	(242,224)
4	(300,143)	-10.51	7.07	-2.32	0	(300,233)	(256,235)
5	(293,199)	-3.44	2.99	-2.32	0	(292,235)	(259,237)
6	(269,191)	-0.45	3.57	-2.32	3.59	(319,236)	(300,240)
7	(186,249)	3.12	0	1.27	9.42	(299,248)	(277,252)
8	(161,263)	3.12	0	10.69	11.18	(299,263)	(272,265)
9	(165,255)	3.12	0	21.87	10.90	(299,252)	(278,255)
10	(297,232)	3.12	0	32.77	0	(294,229)	(279,234)
11	(181,228)	3.12	0	32.77	9.77	(300,228)	(285,230)
12	(194,246)	3.12	0	42.54	8.86	(297,245)	(277,249)
13	(55,284)	3.12	-3.21	51.40	18.63	(291,232)	(265,236)

14	(99,209)	-0.09	0	70.03	15.54	(297,204)	(280,207)
15	(57,209)	-0.09	0	85.57	18.49	(298,204)	(282,208)
16	(277,246)	-0.09	0	104.06	0	(291,251)	(258,252)
17	(457,350)	-0.09	-8.02	104.06	-9.63	(354,249)	(290,252)
18	(326,408)	-8.11	-12.25	94.43	0	(304,270)	(183,273)
19	(113,207)	-20.36	0	94.43	14.55	(259,173)	(169,178)
20	(115,243)	-20.36	0	108.98	14.41	(237,221)	(133,225)
21	(64,206)	-20.36	0	123.39	18.00	(252,170)	(161,173)
22	(123,71)	-20.36	12.32	141.39	13.85	(290,223)	(233,227)
23	(61,203)	-8.04	0	155.24	18.21		

Table 11 Experiment data for the tracking route 2

Point No	Before tracking					After tracking	
	middle LED center in left camera image(pixel)	current tilt (°)	tilt movement (°)	current pan (°)	pan movement (°)	middle LED center in left camera image (pixel)	middle LED center in right camera image(pixel)
24	(194,92)	-13.99	10.79	49.99	8.86	(339,233)	(297,235)
25	(545,229)	-3.2	0	59.85	-15.82	(356,223)	(319,226)
26	(254,182)	-3.2	4.23	43.33	4.64	(335,240)	(310,243)
27	(509,305)	1.03	-4.73	47.97	-13.29	(351,243)	(307,245)
28	(550,273)	-3.70	0	34.68	-16.17	(369,263)	(315,265)
29	(313,362)	-3.70	-8.90	18.51	0	(285,256)	(199,259)
30	(85,289)	-12.60	-3.57	18.51	16.52	(247,219)	(160,221)
31	(486,42)	-16.17	14.44	35.03	-11.67	(347,236)	(313,240)
32	(504,233)	-1.73	0	23.36	-12.94	(350,226)	(319,230)
33	(579,232)	-1.73	0	10.42	-18.21	(354,221)	(322,225)
34	(561,208)	-1.73	0	-7.79	-16.95	(348,202)	(320,205)
35	(584,229)	-1.73	0	-24.74	-18.56	(332,223)	(301,227)
36	(288,386)	-1.73	-10.65	-43.3	0	(289,254)	(203,259)