INSTITUT FÜR INFORMATIK

DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Diplomarbeit

Development and Implementation of a Pointing, Acquisition and Tracking System for Optical Free-Space Communication Systems on High Altitude Platforms

Bernhard Epple

Aufgabensteller:	Prof. Dr. Hans Jürgen Ohlbach
Betreuer:	Markus Knapek
Abgabetermin:	20. Mai 2005

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Diplomarbeit

Development and Implementation of a Pointing, Acquisition and Tracking System for Optical Free-Space Communication Systems on High Altitude Platforms

Bernhard Epple

Aufgabensteller: Prof. Dr. Hans Jürgen Ohlbach
Betreuer: Markus Knapek

Abgabetermin: 20. Mai 2005

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 20. Mai 2005

. . . .
(Unterschrift des Kandidaten)

Abstract

Laser free-space communications technology has a major potential to complement radio frequency (RF) and microwave technology for wireless data transport and backhaul traffic. In order to design reliable inter-platform, platform-to-satellite, and optical downlink terminals, stratospheric tests are necessary. The Capanina Stratospheric Optical Payload Experiment (STROPEX) is one step in this direction in terms of gaining system performance experience and gathering atmospheric index-of-refraction turbulence data. It is not within the scope and budget of the project to design a commercial optical terminal for future high altitude platform (HAP) links. The experiment is focused on experimental verification of the chosen acquisition, pointing, and tracking systems, measurement of atmospheric impacts (turbulence) and successful verification of a broadband downlink from a stratospheric testbed (HAP/balloon/aircraft).

The purpose of this thesis is to develop and implement a pointing, acquisition and tracking (PAT) system for use with an optical free-space communication terminal on a high altitude platform. The developed system will be part of the hardware used within the Capanina project. In particular it is the designated system for the STROPEX test sessions which is part of this project.

For developing the system we will identify the challenges given by the layout of the STROPEX trial and offer a combination of hardware- and softwarebased solutions.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AoI	Area of Interest, camera feature that allows for a custom image resolution
Capanina	Not an acronym or abbreviation, the project is named after the restaurant in Italy where initial project discussions were held
CVS	Compact Vision System, a small PC in a compact case manufactured by National Instruments
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center)
DNO	Dark Noise Offset, the mean pixel value of images taken with the camera at no light, exact value depends on exposure time
FELT	Free-space Experimental Laser Terminal, system that is mounted on the payload of the stratospheric balloon
FoV	Field of View (of a camera), normally given in degree or radian to give a distance independent representation
FPS	Frames Per Second
GPS	Global Positioning System, a space based navigation system using 24 satellites orbiting the earth every 12 hours at an altitude of approximately 20200 kilometers above the earth surface
HAP	High Altitude Platform, unmanned object like a balloon or a Zeppelin flying at an altitude of approximately 20 kilometers
ITU	International Telecommunication Union, headquartered in Geneva (CH), is an international organization within the United Nations System for coordinating global telecom networks and services of governments and the private sector
PAT	Pointing, Acquisition and Tracking
PID	Proportional, Integral and Derivative. Terms used to compute the output of a PID controller
RF	Radio Frequency
STROPEX	Stratospheric Optical Payload Experiment, part of the EU-funded Capanina project
UAV	Unmanned Aerial Vehicle
WGS84	World Geodetic System 1984, reference system used in the GPS system to specify positions

Chapter 1

Introduction

1.1 Optical Free Space Communication

With the increasing need for broadband connections in our daily life, the limitation of today's technologies become more obvious. For example, high speed cable/fiber connections are widely used in urban areas but they are too expensive to cover rural areas and unusable for mobile applications. Today, microwaves are used as a complementary technology to cable/fiber connections. But this solution faces problems like frequency scarcity and energy loss due to wave propagation characteristics.
Since lasers can be focused to beams with a low divergence, they can transfer signalling power to the receiver with less energy loss than microwaves can. Therefore optical free-space communication systems can work with less power consumption than microwave based systems, while offering higher data rates at the same time.

Compared to microwave communication, optical free-space communication promises the following advantages:

- higher data rates with less tansmitting power
- little interference with other transmission systems due to the low divergence angle
- unaffected by the frequency scarcity experienced with radio frequency communications
- no limitations given by the International Telecommunication Union (ITU)
- better protection against eavesdropping

Optical free-space communication is not a perfect technology, because the impact of atmospheric attenuation and atmospheric turbulence is greater on optical systems than on microwave systems. Therefore the goal of current research activities at the German Aerospace Center (DLR) and the European Union (EU) is not to replace radio frequency and microwave systems but to develop a complementary system and to find ways to avoid or reduce the effect of atmospheric impacts. Use of such systems is mainly targeted at the backhaul traffic of inter-platform, platform-to-platform and platform-to-ground terminals, where "platform" is defined as an object like a satelite or an unmanned aerial vehicles (UAV) [Ger05]. This research focus is supported by publications like the recently published "Lighter-than-Air-Technology - Potentials for Innovation and Application" working report from the Office of Technology Assessment at the German Parliament [GO05], which states that high altitude platforms bear a high potential for future developments in the fields of telecommunication and military observations. As these fields deal with transfers of high data volumes, they are suitable candidates for the application of optical free-space communication. More publications on this topic are [GDT+01],[DFO97],[CMA00],[GTK+01] and [TG01] which are listed in the bibliography.

For optical free-space communication two classes of modulation schemes are applicable. The first class is the incoherent schemes, in which Intensity Modulation with Direct Detection belongs. This scheme is also used for fiber-optic transmissions and uses on/off-keying of the carrier laser

for data transmission.
The second class is the coherent transmission schemes, which use all attributes of the carrier laser (amplitude, frequency, phase-position, and polarization) for keying the data on the carrier. Systems in this class have better transmission characteristics than incoherent systems but also have a higher system complexity.

The system used for data transmission in this thesis is Intensity Modulation with Direct Detection as this scheme is well know from fiber-optic transmission and the hardware setup for this scheme has not as much complexity as for the other schemes.

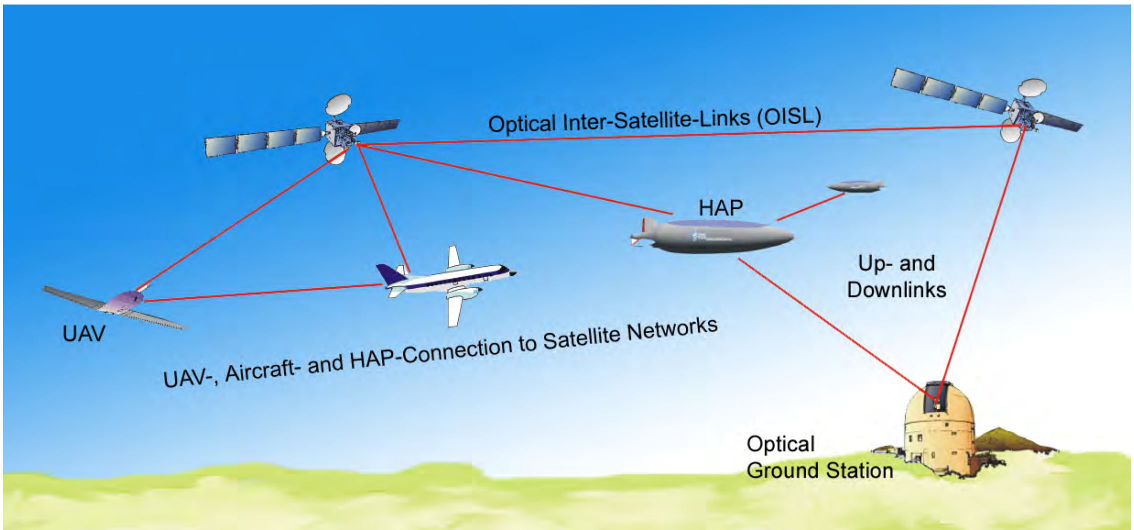


Figure 1.1: Aeronautical application scenarios for optical free-space communication

1.2 The Capanina Project

The system developed in this thesis is part of a research project called Capanina. The name of this project is not an acronym. It is the name of the restaurant in Italy where initial project discussions were held.

1.2.1 Project Description

Capanina is an EU funded program with the goal of evaluating and testing optical free-space technologies for the delivery of broadband backhaul links on aerial platforms. In order to achieve this goal, the project will develop an optical broadband communication system that can be used on high altitude platforms (HAPs) like stratospheric ballons or Zeppelins. These high altitude platforms can deliver connectivity to a wide area (100 to 400km diameter) and can later be used as network backbones.

All systems that are developed within this project will be verified during three main testing campaigns. The system developed in this diploma thesis is subject to the second testing campaign.

1.2.2 Trial 1: Tethered Balloon in England

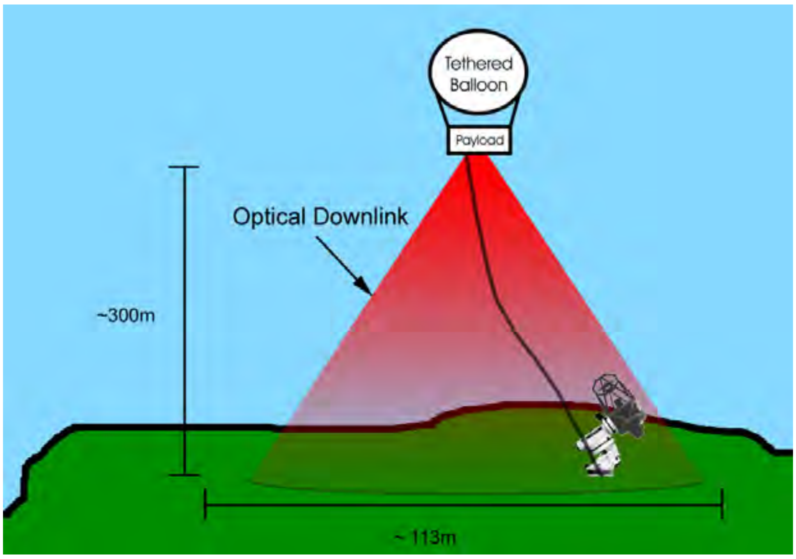


Figure 1.2: Test scenario for Trial 1

The first trial has already been held in Pershore (UK). The test sessions were conducted over several weeks and used a tethered balloon at an altitude of 300 m above ground. During the session the following tasks were completed:

- Demonstration of optical video transmission from Free-space Experimental Laser Terminal (FELT) to ground station with a data rate of 270 Mbps
- Demonstration of end-to-end network connectivity
- Demonstration of services such as: high speed internet, video-on-demand
- Assessment of the suitability of the tethered aerostat technology to deliver "Broadband for All"

For setting up the downlink from the balloon, a laser with a high divergence angle was used. With this high beam divergence angle and the relatively low altitude of the balloon, it was sufficient to simply point the laser straight down from the balloon for transmitting data to the ground station, which was located below the balloon. So this session did not require a Pointing, Acquisition and Tracking (PAT) System on the Free-space Experimental Laser Terminal (FELT) for establishing the downlink.

Figure 1.3: The mobile ground station in front of the tethered balloon during Trial 1, Pershore, UK

1.2.3 Trial 2: STROPEX in Sweden

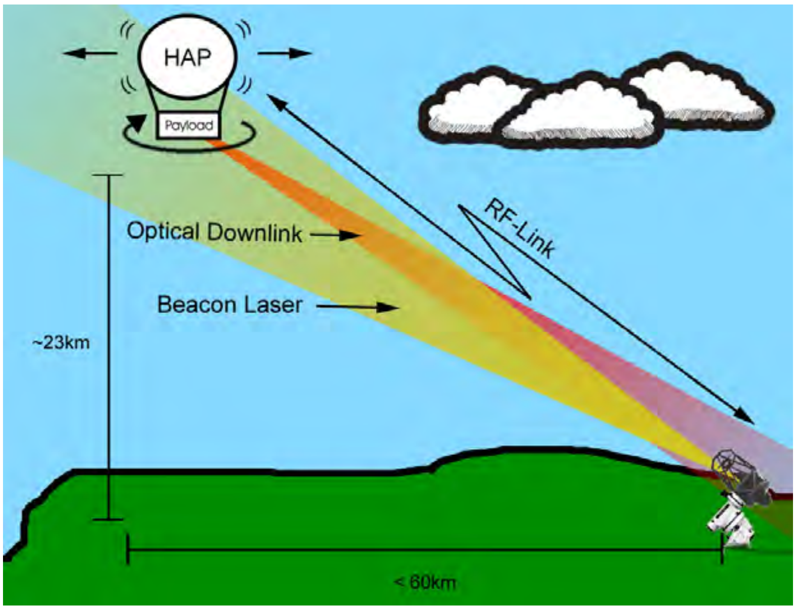


Figure 1.4: Test scenario for Trial 2

The second trial is named Stratospheric Optical Payload Experiment (STROPEX) and will be held in August 2005 in Kiruna, Sweden.
The focus of these experiments is laid on the verification of the chosen PAT systems, the measure-

ment of the atmospheric impacts on the data link and the successful verification of a broadband downlink from a stratospheric testbed.

During the second trial, the developed system will be mounted on a stratospheric balloon which will ascend to an altitude of 22 kilometers. From this altitude, the communication system will acquire the designated ground station and establish an optical downlink with it. The link distance will be up to 63 kilometers and a downlink data rate of 2.5 Gbps is targeted.

1.2.4 Trial 3: Pathfinder Aircraft in Hawaii

The exact details about the third trial are still to be determined, but it is most likely that the tasks will be similar to the ones from the second trial. The HAP for this trial will be NASA's Pathfinder Plus aircraft which will fly at an altitude of approximately 18 to 20 kilometers. The main difference to the second trial is that this aircraft can fly with a velocity of approximately 125 km/h, so the impact of the atmosphere on the optical system is expected to change very much. The system for the second trial was designed to meet the expected requirements of the third trial as well.

Figure 1.5: Pathfinder Plus aircraft over Hawaii. At the center of the aircraft you can see the containers for the payload where the FELT has to fit in

More information about the project, the test sessions and its current status can be found on the project website [CAP05].

1.3 Purpose of this Diploma Thesis

The purpose of this diploma thesis is to develop the Pointing, Acquisition and Tracking (PAT) system that will be used on the FELT during the second Capanina trial. Also the PAT system is tailored for use with optical free-space communication on high altitude platforms, it should use common hardware and a modular design, so it can be adapted for use with other platforms. In the following sections, the system requirements are given and the system specific problems are named.

1.3.1 Pointing, Acquisition and Tracking (PAT) Systems

PAT-systems are an essential part for successfully establishing an optical free-space link in mobile environments. As their name suggests, they operate in three phases for setting up the link.

The pointing phase is normally done by blind pointing of the transmission laser towards the receiver based on a-priori knowledge like the transmitter and receiver positions.
During the acquisition phase the exact position of the receiver has to be located and the transmission laser has to be readjusted towards this new location. For marking the ground station, two techniques are commonly used. One is to place a beacon laser at the receiver and point it towards the HAP and the PAT system has to find this beacon. The other approach is to place a retro-reflector at the receiver. For detecting the receiver the PAT system scans with the transmission laser or an additional beacon laser over the uncertainty area. As soon as the laser hits the retro-reflector it gets reflected back to the PAT system. The system can detect this reflection and with it, the ground station.
If the receiver has successfully been detected, the tracking phase begins. The goal of the tracking phase is to keep the transmission laser targeted onto the receiver.

As long-range optical free-space communication systems are still under development and differ significantly in the used hardware, they all use custom made PAT systems which are optimized for the particular systems. Most of the systems have only been tested in the laboratory, so it is uncertain if they will work in real world conditions. Therefore a new PAT system has to be developed for use within the Capanina project.

1.3.2 Dedicated System

During the STROPEX tests, a beacon laser will point from the ground station towards the FELT on the balloon. The developed system has to be able to reliably acquire this beacon laser and to stay focused on it. For detecting the beacon laser, the FELT will be equipped with a camera for visual acquisition of the beacon and a periscope in front of the camera for moving its field of view. The hardware will be set up in a way, that if the camera points on the center of the beacon laser, the transmission laser will target on the receiver of the ground station. For a successful discovery of the beacon laser and for a sufficient tracking accuracy, fast image analysis algorithms have to be found and implemented.
The main reason why we have to use visual detection of the beacon laser is that the environment in our testing scenario is very complex, so using hardware based techniques, in particular photodetectors and other electrical sensors, as are used for example by laser guided bombs, would be problematic and error-prone.

For exchanging commands and status information between the optical payload and the ground station, a common radio frequency link (RF link) has to be implemented.

The hardware of the FELT may be changed during the development of the project, so the PAT-system will mainly consist of of-the-shelf hardware components as these can easily be replaced by similar components. This also makes the system reusable for future projects.

1.3.3 Challenges

The system will have to cope with the following problems and needs to offer a hardware or software implemented solution.

1.3.3.1 Balloon Movement

Most of the existing and proposed systems use positioning information for the pointing phase and therefore need accurate positioning information for reducing the uncertainty area in which the receiver has to be located during acquisition. If the information is accurate enough, no special steps have to be done during the acquisition phase. This approach has been chosen by NASA for its "Altair UAV-to-Ground Lasercomm 2.5 Gbps Demonstration" [OLM+03].
The following will show why we have to find another approach for locating the ground station.

Previous measurements during the MIPAS-B2 experiment [FVMK+04] have shown that the balloon can travel horizontally over a distance of up to 60 kilometeres within two hours. The horizontal velocity can reach up to 100 km/h. It has also been measured that the winds might cause the balloon to rotate with an angular speed of up to 36 deg/s. Finally, there is also a pendulous movement with two degrees amplitude and a typical period of 1.3 to 1.6 seconds.

To reduce the impact of these movements, the experiment is conducted during a period when the weather is normally fine with low winds, so the results of the MIPAS-B2 measurements can be understood as a worst case scenario.

Due to the unpredictable movement of the balloon, the FELT does not know its own position, its heading or even the position of the ground station.
The balloon and the ground station are equipped with GPS receivers for exchanging position information over the RF link, so we can acquire some positioning data that can be used for acquisition. However the received GPS data does not contain any information about the balloon's heading, so closed-loop acquisition of the beacon laser is not possible.

The movement of the balloon not only causes problems for the acquisition of the beacon laser, it also tightens the requirement of fast image analysis algorithms to achieve a high processing frame rate. A high processing frame rate is needed for good tracking accuracy and to finish the acquisition in a reasonable time.

As a basis of all assumptions and calculations in this thesis, the worst case scenario is defined as following: The balloon is travelling with a speed of 100 km/h at an altitude of 22 kilometers and a distance of 60 kilometers away from the ground station.

1.3.3.2 Image Quality

During the test, the lighting conditions will change due to the movement of the sun and the changing weather conditions. The camera has to adapt to these changes to prevent over- and underexposure of the recorded images and to guarantee the visibility of the beacon laser.

1.3.3.3 Reflections and Background Light

To detect the beacon laser in the camera image, the beacon laser has to be more powerful than the general background illumination caused by the sunlight being reflected from the surface of the earth. If the beacon laser is not bright enough, it will blend in with the background light and therefore be invisible to the camera. This fact has been taken into account for the power calculations of the beacon laser, but unfortunately the beacon laser will not be the only visible bright spot on the recorded images. Due to hardware restrictions, the power for the beacon laser can not be raised to a level that could guarantee that the beacon is the brightest visible spot on the images. So the detection algorithm has to be able to discover the laser beacon even if there is more than one bright spot in the image.

1.3.3.4 Tracking Accuracy

The needed tracking accuracy is defined by the divergence of the transmission laser and the movement of the balloon. The transmission laser has a divergence of 1.2 milliradian (mrad). Therefore the required tracking accuracy for the system is 0.6 mrad in every direction. Combined with the assumed motion of the balloon we get the following requirements for the tracking system:

In the worst case scenario the horizontal velocity can be neglected because of the high rotational speed of 36 deg/s. If we divide the rotational speed by the needed tracking accuracy of 0.6 mrad, we end up with an adjustment frequency of approximately 1047 Hz.
As this high frequency is hard to reach with a common image processing system, the hardware will be designed in a way to reduce or even control the rotational velocity of the camera.

1.4 Thesis Overview

The thesis is structured in the following way.

Chapter 2 gives an overview of the developed hardware as this defines several constraints for the PAT system.
In chapter 3 we determine what the recorded images during the second trial will look like, as the PAT system has to be able to work on them. When we have gathered enough information about the images, we can use this knowledge in chapter 4 to develop the needed algorithms for solving the problems mentioned in chapter 1.3.3.
In chapter 5, an overview of the implemented system is given and details about the different software modules are provided. To complete the system development, the conducted tests are documented in chapter 6 and a conclusion of this thesis is given in chapter 7.

Chapter 2

Free-space Experimental Laser Terminal Hardware

This chapter gives a short overview of the hardware used for the free-space experimental laser terminal (FELT), which defines some constraints for the software. The description will not be complete and only includes the parts that are important for the development of the software. For more details about the hardware, see Appendix B.

2.1 The Compact Vision System

The software will run on a Compact Vision System (CVS) 1454 manufactured by National Instruments. This system has mainly been chosen because of its small dimensions of 10.2 cm x 12.7 cm x 6.4 cm and its weight of only 977 g. It contains an Intel CPU with an instruction rate of 833 MIPS, 128 MB DRAM and 32 MB nonvolatile memory. The installed operating system is the Pharlab ETS real time operating system. Because the CVS is sold by National Instruments, it ships already set up to execute programs written in LabView, which is a graphical programming language developed and sold by National Instruments. LabView can easily interface to external DLLs, for example written in C/C++, so other languages can also be used.

Figure 2.1: CVS 1454

2.2 Optical Hardware

2.2.1 The Camera

The camera is a Basler 602f IEEE 1394 monochrome CMOS camera. It supports various modes and will be used in the Mono 8 mode, which produces 8 bit monochrome images with a maximum frame rate of 100 fps at full resolution. The full resolution is 656 pixels width and 491 pixels height and the size of one pixel is 9.9 $\mu m \times$ 9.9 μm . The camera also supports the specification of custom resolutions via the area of interest (AoI) feature. With the use of custom resolutions higher frame rates are possible. The quantum efficiency (figure B.1) and other data are given in appendix B.1.

2.2.2 The Beacon Lasers at ground station

Two beacon lasers with a wave length of 810 nm and a divergence angle of 4 mrad will be at the ground station. The power of each of these lasers will be 5 W. The two lasers are set up to be incoherent to each other, so they will not interfere with each other and be seen by the camera as one single laser with a power of 10W. As it can not be distinguished between the two beacon lasers we will refer to them as one beacon laser throughout this thesis.

The used lasers are so called multimode lasers which have the following characteristics.
The intensity profile at the receiver can be assumed as being uniformly distributed unlike the intensity profile of single mode lasers which have a gaussian intensity profile.

2.2.3 Lasers on FELT

On the FELT there are two types of lasers. One laser is the transmission laser which has a wavelength of 1550 nm. The other lasers are beacon lasers used to track the FELT from ground station. The beacon laser has a wavelength of 986 nm. Maybe additional lasers will be added to the system. The power of the lasers has still to be decided.

2.2.4 The Lens and the Field of View

The lens in front of the camera has a focal length of 69 mm. The resulting field of view (FoV) can be calculated by following formula

$$FieldOfView\ (in\ rad) = \frac{SensorEdgeLength}{FocalLength} \tag{2.1}$$

So the field of view of the camera will be 70 mrad (4 °) in height and 94 mrad (5.4 °) in width. To simplify our calculations we will always refer to the field of view as being 70 mrad or 4 °.
The diameter of the lens is 25 mm.

2.2.5 The Filter

To reduce the amount of incoming background light a bandpass filter is used. The bandwidth of the filter is ranging from 800 nm to 850 nm. The transmission curve for the filter is given in figure B.2 in the appendix.

2.3 Periscope

The persiscope is custom manufactured in cooperation with RoboTechnology GmbH. The construction allows the field of view of the camera to rotate around the vertical and horizontal axes. For these rotations, two motors with controllers from Maxon Motors are used. The controllers ship with a driver for Windows which can be interfaced from within LabView. The two controllers have a resolution of 0.005 deg and allow motions of ± 70 ° in the two axes. The maximum angular velocity of the motors is specified to be >100 deg/s and the angular acceleration is designed to be >27.6 deg/s^2. For aligning the field of view of the camera with the transmission and beacon laser, these parts are placed in the optical path of the periscope. Figure 2.3 gives an overview of the optical layout of the periscope. The cross-section of the optical path is shown in the upper right corner of Figure 2.3. In this sketch you can see that the lasers are mounted below the field of view of the camera, so they do not block the field of view of the camera, but use the same optical path.

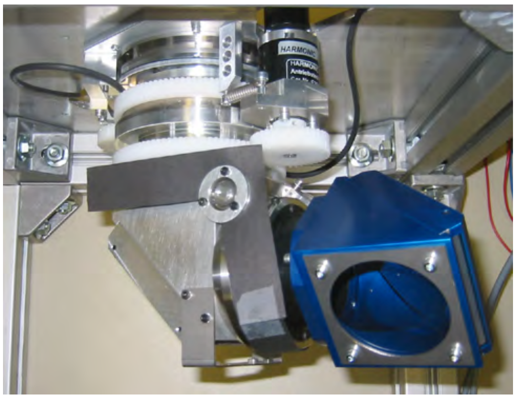


Figure 2.2: The periscope mounted in a test stand

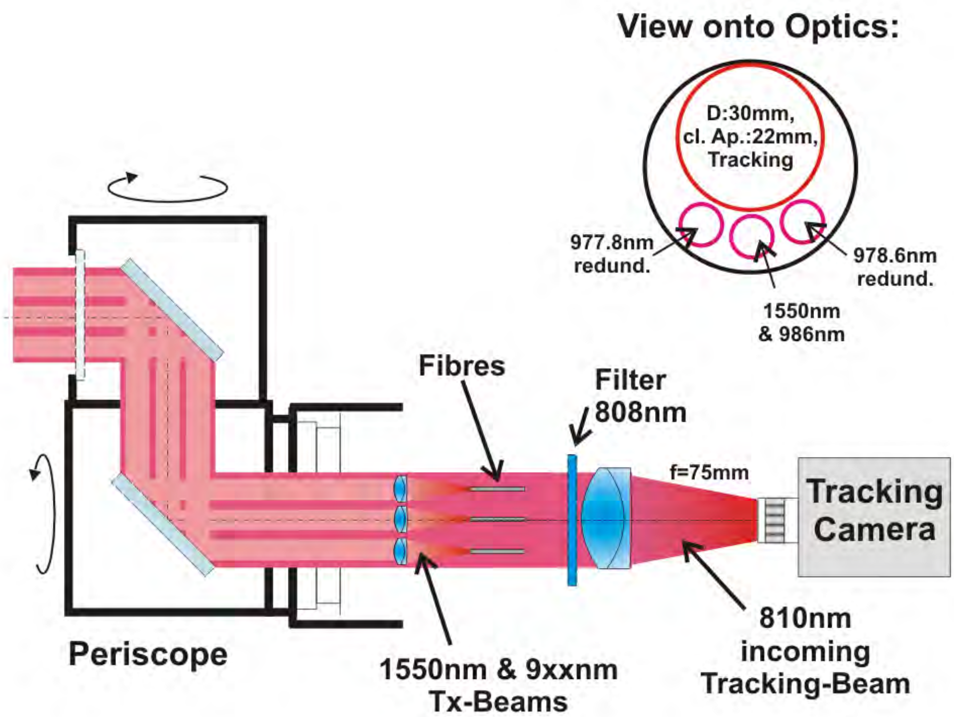


Figure 2.3: Schematic overview of the periscope and the FELT optical system. The sketch is rotated clockwise by 90 degrees. In the upper right corner of the drawing, the positions of the camera and the lasers in the optical path are shown

2.4 Additional Hardware

As additional hardware, GPS receivers are used for getting positioning information and a gyroscope is used for getting information about the balloon's rotation. According to the balloon operators the accuracy of the received GPS information is ±50 m horizontally and ±100 m vertically. The gyroscope gives the rotational speed with an accuracy of 1 deg/s.

Chapter 3

The Captured Image

The key consideration about camera calibration and image analysis is to know what the recorded images would look like. There have not been enough ressources in the projects budget to do a flight with the camera just for taking some images, so a simple test scenario had to be created. The images recorded within the test scenario should be similar to those that will be recorded during the trials. Before running the test, some calculations on the constraints of the test scenario were done, to predict the look of the recorded images. For testing the formulae quality, the predicted images were compared to the images taken during the test. The developed formulae could then be used to determine the look of the images for the second trial.

3.1 Camera Calibration

The main influencing factor on the look of the recorded images is the camera itself. So for calculating the look of the images, it is necessary to gain some knowledge about the characteristics of the camera.

3.1.1 Image Format and Camera Attributes

The Basler 602f camera is a monochrome camera, so the beacon laser will appear as a bright (light grey to white) spot in the recorded images. The camera can record images in two different formats, one is the 8 bit format (pixel values are between 0 and 255) and the other is the 10 bit format (0-1023). The images are always captured in the 10 bit format and later converted to the 8 bit format if needed. This conversion is done on the camera itself.

For the transfer from the camera to the CVS, the 10 bit images are encoded as 16 bit images with only 10 bits effective so the data size of a transferred 10 bit image is exactly twice as large as an 8 bit image. The data size of the image is one limiting factor for the frame rate of the camera (others are shutter time, configuration of the firewire bus, and the camera hardware itself). The frame rate for 10 bit images is limited to 50 frames per second (fps) and with 8 bit images the limit is 100 fps. As a higher frame rate is better for PAT purposes, the camera will be used in the 8 bit mode.

The conversion from the 10 bit to the 8 bit format is influenced by two camera attributes, gain and brightness.

The values for gain can range from 0 to 255 and represent an amplification factor for the pixel values in the image ranging from 1 to 4 times. A table of example values is given on page 52. This is useful for us since we are interested in good visibility of the bright spots in the image for easier spot detection.

The values for brightness can range from 0 to 1023 with a default value of 717. If the brightness is set to a value below 717 the whole image is darkened by a specific amount and if it is set above 717 the whole image is brightened by a specific amount. The strength of the brightening or darkening depends on the gain setting. If the gain is set to 0, changing the brightness value by 4 will result in a change of 1 in the image. If the gain is set to 255, changing the brightness by 1 will result in a change of the pixel values by 1. With these two attributes the image can be influenced after it has been captured.

A more important attribute is the shutter value, as it directly affects the recording of the image. The shutter value determines the exposure time and with it the amount of light that can pass trough to the sensors for generating the image. This again will determine the brightness of the recorded objects in the image.

The values for shutter range from 1 to 4095. For calculating the exposure time, the shutter value is multiplied by 20 μs. A shutter value of 500 forces the camera to a frame rate of 100 frames per second. As this is also the maximum frame rate of the camera at full resolution, shutter value in the system will be limited to 500 so that the camera will not be slowed down by the exposure time.

For developing a PAT system, it is interesting to determine the responsiveness of the camera to the beacon laser, the background light, and to changes of the shutter value. Unfortunately the light responsiveness of each camera varies slightly and depends on the wavelength of the received light, so even the manufacturer can not give reliable information about it. Therefore some measurements have been done in the laboratory.

3.1.2 Camera Responsiveness Measurements

For test setup, we targeted the camera with a filter in front on a halogen bulb as the illumination source. The distance between the bulb and the camera was big enough to get a homogeneous intensity distribution over the sensor. The filter was similar to the one chosen for the final system. The light intensity hitting the camera was measured by a light detector with the same filter on it. The images were captured into the 10 bit format to remove the effects of the brightness and gain values as these are well known and would only complicate the measurements. During the measurements, the bulb was turned to different intensities and a series of images was taken with different shutter values. Unfortunately there were problems with the system as it produced some strange values which did not fit with our expectations. When we re-ran the measurements, the strange values were gone. So we had to do several runs to get reliable results.

By comparing the mean values of these images with respect to the corresponding intensities and shutter values, we got the following results.

First, if images are taken in complete darkness, the images will still have a mean value between 49 and 60, depending on the shutter time. For images taken in the 8 bit format, assuming direct conversion between the 10 and 8 bit format, the values for the Dark Noise Offset (DNO) should range from 12 to 15. The measurements show values from 0 to 14. As the error introduced by this behavior seemed tolerable, no further investigations have been done on it.
The second observation confirmed what we were expecting. With slight irregularities, the pixel values can be assumed as a linear response to the shutter values depending on the light intensity. With these results, we could develop a model for the behavior of the pixel values.
For diagrams and further details about the measurement results, see Appendix A.2.

3.1.3 Pixel Value Model

For determining the value of the Dark Noise Offset for a given shutter value s, the following formula could be derived from the measurement results:

$$DNO(s) = 49 + \frac{s}{400} \tag{3.1}$$

As the shutter value will be limited to a value of 500, the formula gives values for the DNO ranging from 49 and 50.25. If the DNO is assumed as a constant value of 50, we will get a maximum error in the calculation of the pixel values of 1, which will be sufficient for our purpose.

The derived formula for the effect of changing the shutter value on a given pixel value can now be described by the following.

$$PixelValue_{new} = (PixelValue_{old} - DNO) \times \frac{ShutterValue_{new}}{ShutterValue_{old}} + DNO \tag{3.2}$$

For every $PixelValue_{old} \geq DNO$.

After taking one image with the camera, this formula can be used to calculate the best suited shutter value for taking the next image. As it is also interesting to know what images can be expected during the trials, another formula has been developed. This formula takes the light intensity into account.
The developed formula is

$$PixelValue = m \times I + DNO \tag{3.3}$$

where m is the responsivity of the camera and I is the light intensity in $\frac{W}{m^2}$. From the measurements and the rough indications in the camera manual we derived

$$m = 3.2 \times ShutterValue$$

The calculated pixel values always have to be converted into integers and they are limited to a range from 0 to 1023.

If a laser with another wavelength is used, the factor for calculating m has to be adjusted since the responsiveness of the camera is dependent on the wavelength of the received light.

The given formulae and values are all for the captured 10 bit images. As we will use 8 bit images for the implementation of the system, the given values have to be converted from 10 bit to 8 bit with respect to the settings for brightness and gain.

3.2 The Test Scenario

The test scenario had been designed as follows:

The club house of the Akademischer Seglerverein München e.V. is located in Herrsching on Ammersee and provides a good view towards the top of the Peissenberg which is located approximately 25 kilometers away from the club house. A 70 mW laser with a wavelength of 986nm was placed at the club house and pointed towards the top of the Peissenberg. The divergence angle of the laser was measured to be 5.9 mrad. The camera was placed on top of the Peissenberg and had the club house in its field of view. For getting a focused image on the camera, a lens with a focal length of 150 mm and an effective diameter of 25 mm was used in front of the camera. For measuring the effects of a filter, we did two runs, one without a filter in front of the camera and one with a 986 nm filter with a bandwidth of $\pm 5nm$ and a guaranteed transmission of 70 percent. Figures 3.1 and 3.2 illustrate the experimental setup:

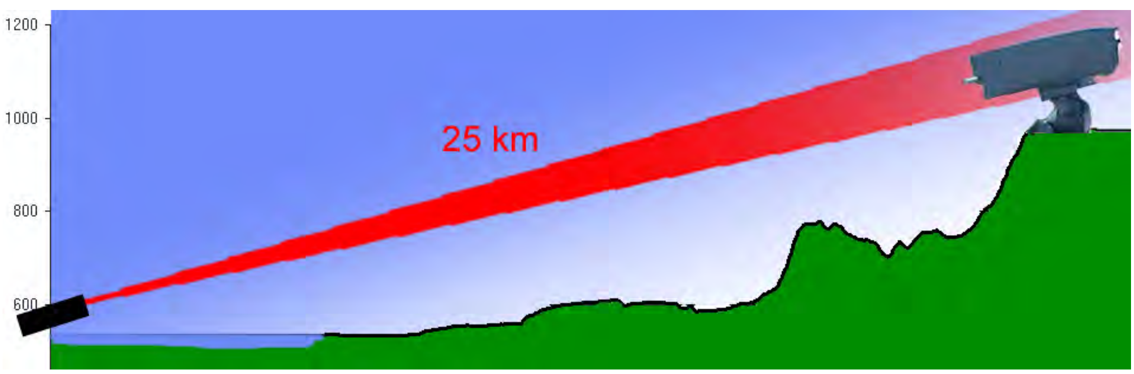


Figure 3.1: Test setup for the Peissenberg Experiment

Figure 3.1 gives an overview on the test setup with the laser being located near Ammersee and the camera on the Peissenberg. For additional information, the height profile of the testing distance has been included in the graphic. The scale on the left side of the figure gives the height above sea level. The laser beam and the camera are not to scale.

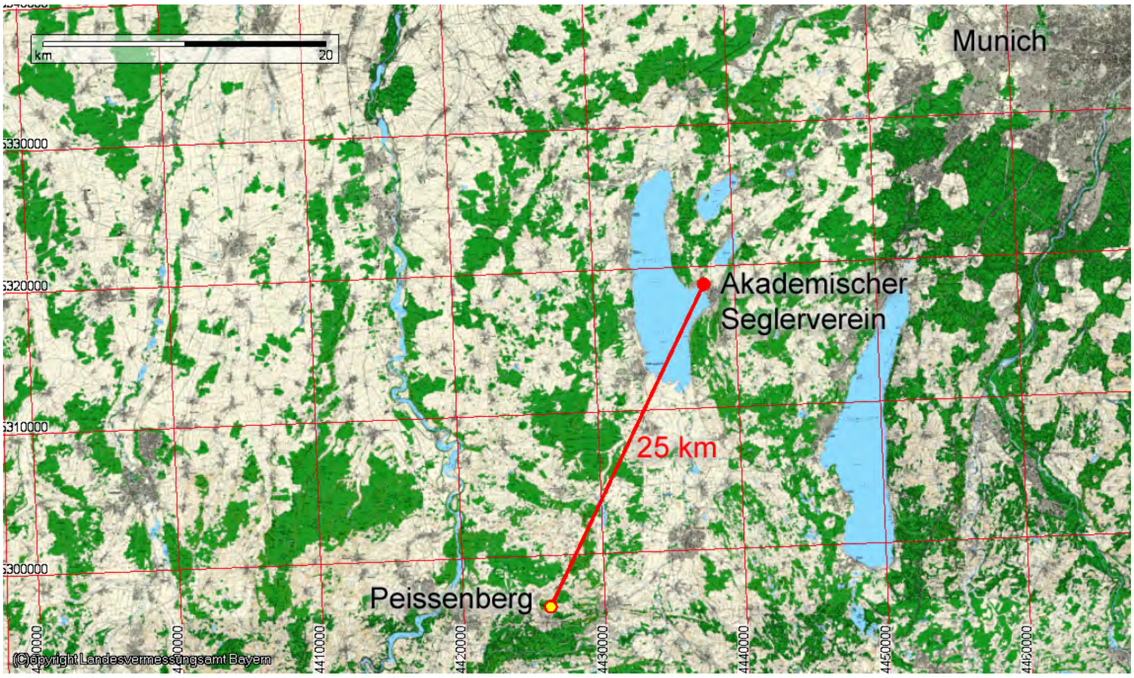


Figure 3.2: Map of the testing region for the Peissenberg Experiment

Figure 3.2 shows a map of the region where the experiment was conducted. Munich is shown in the upper right corner for orientation.

3.3 Calculating the Image

If the laser light hits the lens and gets diffracted by it, the amplitude and intensity distribution change. This effect, known as the Fraunhofer diffraction pattern [ST91], [Goo96], causes the camera sensor placed in the focus of the lens to record the diffraction image of the laser light. Figures 3.3 and 3.4 illustrate this effect.

Figure3.4 shows, that nearly all of the intensity is contained within the central lobe of the curve and the intensity contained in the other lobes can be neglected. This simplyfies the calculations.

For calculating the diameter of the central lobe on the camera sensor in our test setup, Giggenbach [Gig04, p. 119, f. 5.12] states the following formula:

$$\begin{aligned} D_{Sensor} &= 2,44 \times \lambda \times \frac{f}{D_{Rx}} \\ f &: \text{focal length of the lens} \\ D_{Rx} &: \text{diameter of the lens} \end{aligned} \tag{3.4}$$

If the formula 3.4 is used with the values of our test setup, the spot of the laser can be expected to cover a circle with a diameter of 1.22 pixels on the sensor of the camera. This means the laser will be visible in a square of at least two by two pixels in the recorded images.

For predicting the values of the covered pixels in the images with formula 3.3, the intensity that is received by the pixels on the sensor has to be calculated. This could be done accurate

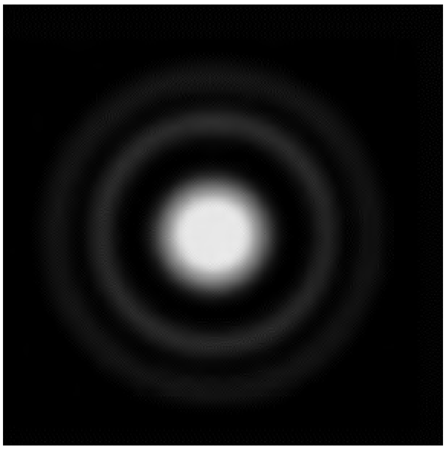


Figure 3.3: Airy disks in the lens focus

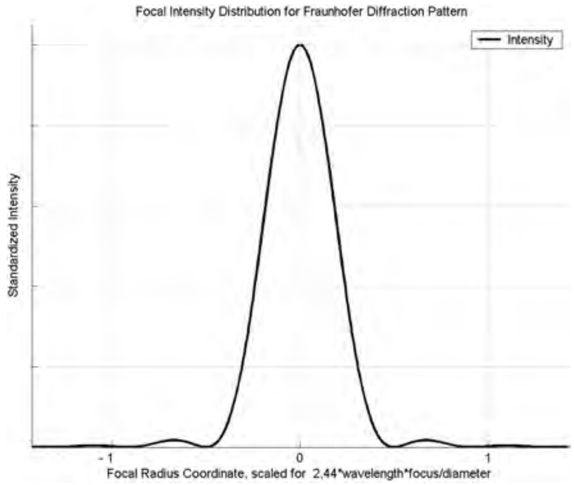


Figure 3.4: Normalized intensity distribution in lens focus

by using other formulae from Giggenbach [Gig04, p. 119, f. 5.13] and some math, but for our purpose it will be sufficient if we assume the intensity within the central lobe to be 85 percent of the overall intensity I_{Rx} received by the lens. For calculating I_{Rx} following formulae can be used:

$$\begin{aligned} r_l &= tan\left(\frac{\alpha_l}{2}\right) \times \Delta d \\ &\text{and} \\ I_{Rx} &= \frac{P_l}{r_l^2 \times \pi} \end{aligned} \tag{3.5}$$

where r_l is the radius of the laser beam at the receiver lens, α_l and P_l are the divergence and the power of the laser and Δd is the distance between the laser source and the lens of the receiver.

For our test setup, this results in an intensity I_{Rx} of $4.0098 \times 10^{-06} \frac{W}{m^2}$. The intensity on the sensor pixels is calculated by:

$$I_s = 0.85 \times I_{Rx} \times \frac{A_l}{A_p} \tag{3.6}$$

where A_l is the area of the lens and A_p the area of the pixels.

This results in a value for I_s of $14 \frac{W}{m^2}$. Using formula 3.3 and assuming that the focused beacon laser always centers on one pixel on the sensor, we get the values given in table 3.2 for this pixel and its 8 neighbors in accordance to selected shutter values.

If we assume the focused laser as being equally distributed on four pixels, we get the values shown in table 3.2.

The conversion between 10-bit values and 8-bit values in the given examples was done using a brightness value of 717 and a gain value of 0, so these two attributes did not affect the conversion.

As tables 3.1 and 3.2 show, the appearance of the recorded spot is depending on the position of the beacon laser on the camera sensor. Since it is not possible to control the exact position of the beacon laser, the appearance of the beacon laser will differ between the recorded images. The beacon laser can only be expect to cover an area between four and nine pixels in the images, with the values given in the tables.

Unfortunately we can not fully rely on these results.

Shutter Value	Pixel Value Center (8-bit)	Pixel Value Neighbors (8-bit)
1	24	13
10	124	13
20	236	14
22	255	14
50	255	16
100	255	16
500	255	51
...	...	...
3145	255	255

Table 3.1: Assorted pixel values assuming spot centered on one pixel

Shutter Value	Pixel Values (8-bit)
1	16
5	30
10	47
15	64
20	81
25	98
...	...
71	255

Table 3.2: Assorted pixel values assuming spot equally distributed on four pixels

3.4 Errors in the Calculated Image

There are some errors contained in the calculated results.

First, the formulae are all for perfect optical systems. As usual, actual systems are never perfect as these systems are very complex and difficult to create and they will automatically produce slightly different results.

Second, if the sensor of the camera is not exactly in the focus of the lens it will receive a changed Fraunhofer diffraction pattern. There are many different abberations possible, but the system will mainly notice spherical abberations. This is illustrated in figure 3.5. As you can see,

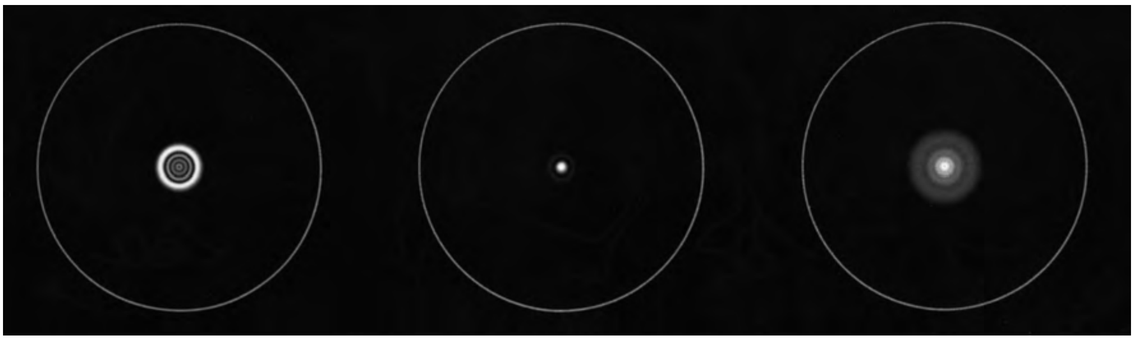


Figure 3.5: Effect of spherical aberrations. Sensor placement from left to right: before focus, at focus and after focus[dWV03]

the size of the focused laser and the intensity distribution will change depending on the focusing of the camera sensor.

Third, we did not take the received background illumination into account, so the recorded images will be brighter than the calculated images, as the intensity of the background light gets added to the intensity of the beacon laser.

Fourth, as we will use a filter for taking some images, we will see different aberrations caused by the characteristics of the filter. How the filter affects the light of the beacon laser is hard to estimate.

Fifth, the atmosphere causes the received light to fade over time so we can not be sure to receive the full intensity of the light. These fades can reduce the received intensity by up to 10dB.

3.5 Inspection of the used Formulae

For inspection of the used formulae, we calculated the values for a spot size of 4 x 4 pixels for assorted shutter values. We then compared these values with the 4 x 4 centers of the spots in the recorded images. A diagram of the results is given in figure 3.6. For recording useful images and data, we first did a test run when the sun was high on the horizon and one after the twilight had started. With the two test runs we could visualize the impact of the background light and the effect of the filter. Unfortunately the filter caused stronger abberations than we had expected, so the results of these images were not usable for validating the formulae. A comparison of the different types of images we recorded is given in figure 3.7 at the end of this section.

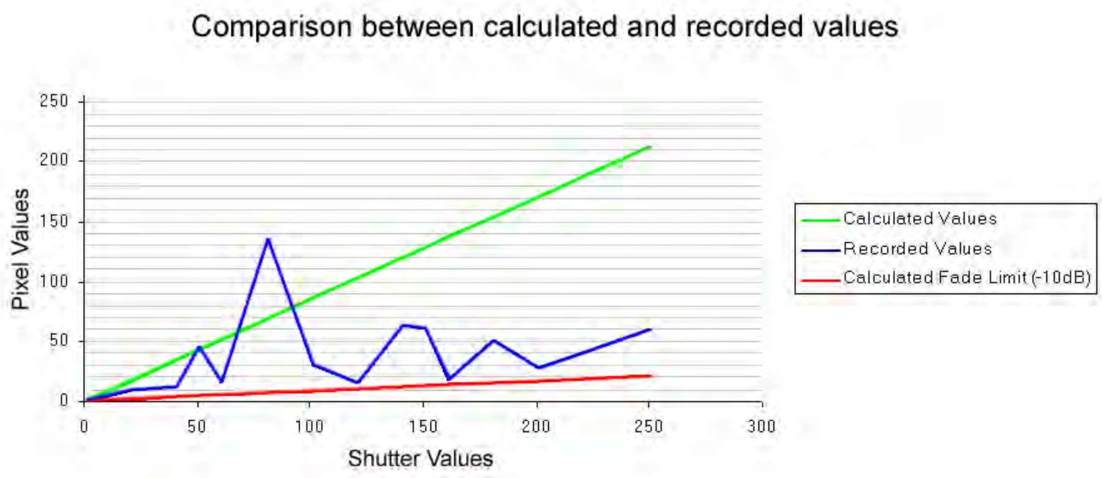
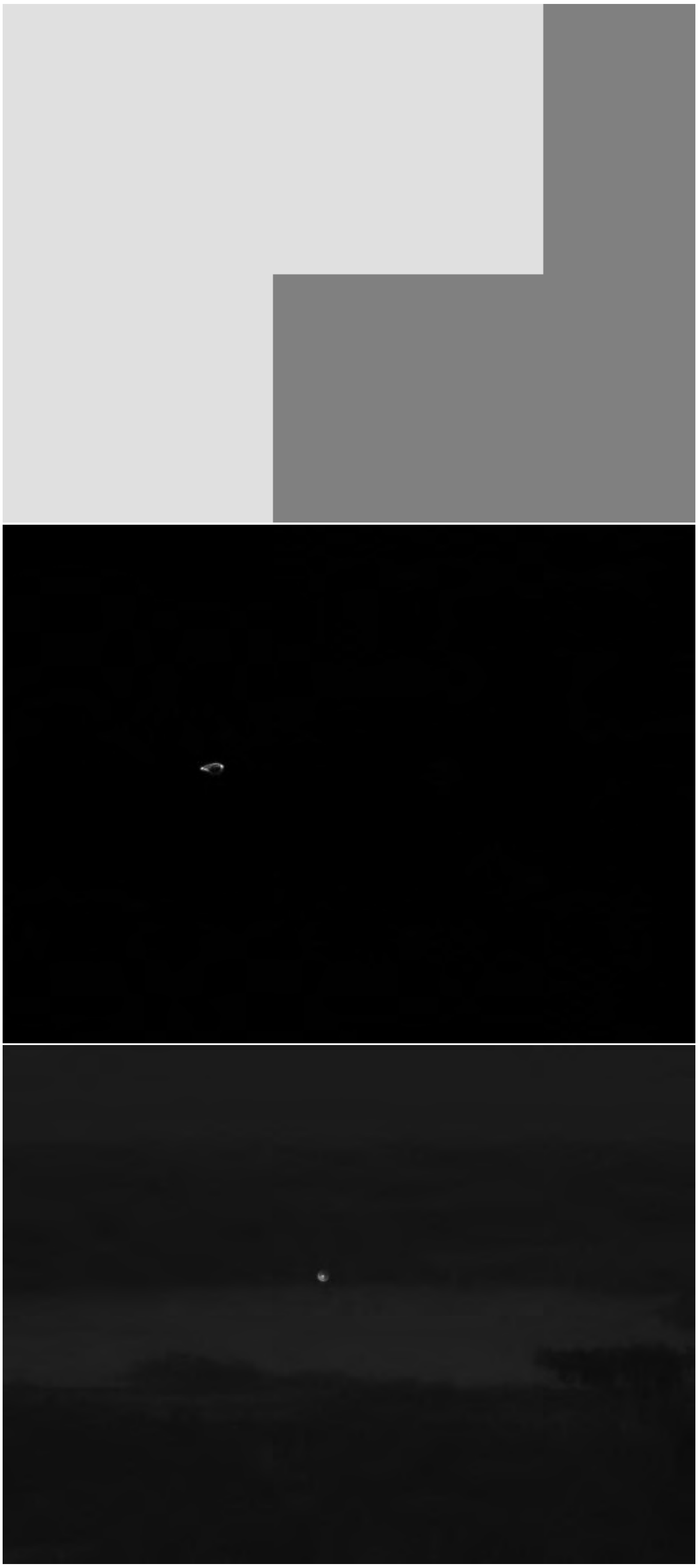


Figure 3.6: Comparison between calculated and recorded values

What figure 3.6 shows, is that the recorded values stay mainly between the calculated values (the maximum curve) and the calculated values damped by 10 dB (the minimum curve). There is one peak that is above the maximum curve. It is most likely that this peak was created by a malfunction in the testing program. If we assume that the gain was falsely set to 255 instead of 0 the value would fit very well between the two curves. The experiences made during the responsiveness measurements tells us that this type of error is possible with our test system.



This images was taken during daylight with a shutter value of 101 and no filter in front of the camera. As you can see it is completely over exposed and it is impossible to detect the spot.

This images was also taken during daylight with a shutter value of 101, but with a filter in front of the camera. The spot is now visible, but the filter caused the images of the beacon laser to get stretched and the light intensity seems to be unevenly distributed in the spot.
(The offset of the spot in this image compared to its position in the next image is caused by moving the camera during removal of the filter)

This images was taken during twilight with a shutter value of 101 and no filter. Compared to the first image, the background light is heavily reduced and the beacon laser is visible as a bright spot. As the filter was not used for this image, the aberration from the previous image is gone, but the intensity is still not evenly distributed over the spot. Nevertheless, the spot has a bright center and circular shape as expected.
(In front of the spot you can see the Ammersee)

Figure 3.7: Comparison of the different images taken during the Peissenberg Experiment

3.6 Conclusion for the Images during Trial Two

With the previous chapters following details about the images during the second trial can be given:

- In a perfect optical system the focussed beacon laser in Trial Two would have a diameter of 0.55 pixel on the camera sensor
- As the optical system will be more accurate than the system in our test, we can expect the beacon laser not to be recorded with a diameter larger than 15 pixels.
- The fades and the background light have a strong and unpredictable influence on the recorded images. But with a good filter, the effect of unwanted background light can be reduced, so the background light should not be a problem. For the fades, it has been shown that assuming an attenuation of 10 dB gives good results.
- If the FELT is further away than 32 km from the beacon laser, the camera will receive less intensity from the beacon laser than it received during our test. This may enable fades to extinguish the beacon laser in a few images, but as the laser will mainly traverse through thinner air than during our experiment, the effect of the fades will not be as strong as those noticed during the experiment. But we will have to make the system robust against fades
- As there remain some uncertainties within the system, we will have to run a similar test with the final systems to confirm that our assumptions are still correct
- Unfortunately we have not been able to record any reflections (i.e. sunlight reflecting off a window), so we can not say anything about their characteristics. It may be usefull to conduct a test to record some reflections.

Chapter 4

Algorithms

Some of the problems given in chapter 1.3.3 had to be solved by software on the FELT. These problems are the calibration of the camera for a constant image quality, a fast and robust image analysis algorithm for the detection of the beacon laser, and the whole logic for the pointing, acquisition and tracking of the beacon laser.

4.1 The Calibration Algorithm

The Basler 602f firewire camera has no automatic controls that control the image quality, so the user has to take care of this. Moreover, the definition of a good image depends on the purpose of the image. In our case, a good image allows for good visibility of the beacon laser. For good visibility of the beacon laser the camera should record images in which the beacon laser is as bright as possible while everything else is as dark as possible. Another thing we have to keep in mind is that there will be other bright spots from reflections in the images which can be brighter than the beacon laser. So the camera has to be calibrated in a way that keeps the beacon laser separated from the background illumination even if there are brighter objects in the image. The goal of the camera calibration algorithm is to produce images with histograms that allow for a good separation of the background (sunlight reflected from earth surface) and foreground (light received from beacon laser or reflections). In a good image the separation should be possible by thresholding the image.

Figure 4.1 and 4.2 show two images taken with the camera and their histograms. The camera and a laser were pointing at the ceiling of the laboratory for taking these images. Between taking the images, only the shutter value of the camera was changed.

The first image is a bad image for separation as the pixel values of the beacon laser are very close to the pixel values of the background light. In this image it is no easy to determine a threshold value for separation of the foreground values from the background values. One might say that the peak caused by the pixel values of beacon laser is clearly visible at the upper end of the histogram, but another thing one can see is that the pixel values of the background light feather into this peak. So it is not guaranteed that this peak is singly created by the values of the beacon laser and if this image is thresholded some parts of the background will always remain in the thresholded image.

Another problem with this image is that the pixel values of the background spread over a long range in the histogram. So it might have happen that the pixel values of the beacon laser already have dissolve with the pixel values of the background light and the peak at the upper end of the histogram is caused by a reflection that is brighter than the beacon laser.

The second image is very well suited for detecting the beacon laser. In the histogram you can see the values of the background light are located at the lower end of the scale and do not spread over a long range of values. As there are only a few values of the beacon laser in comparison to the amount of values of the background light, the values of the beacon laser are not visible in the

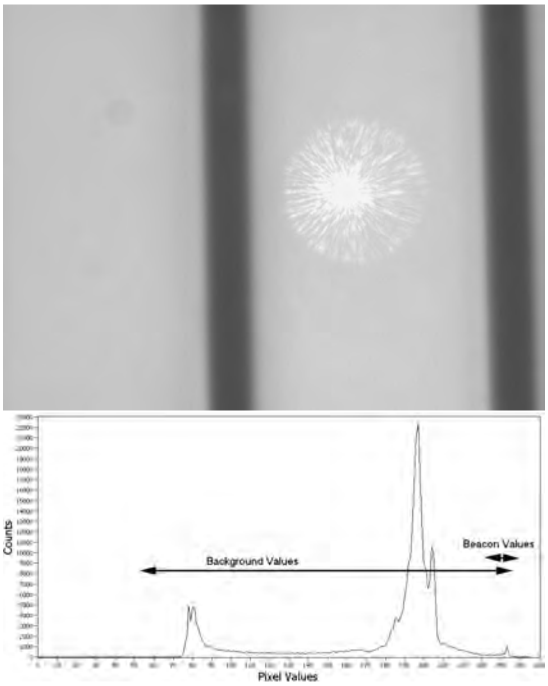


Figure 4.1: Image with a bad histogram for blob extraction

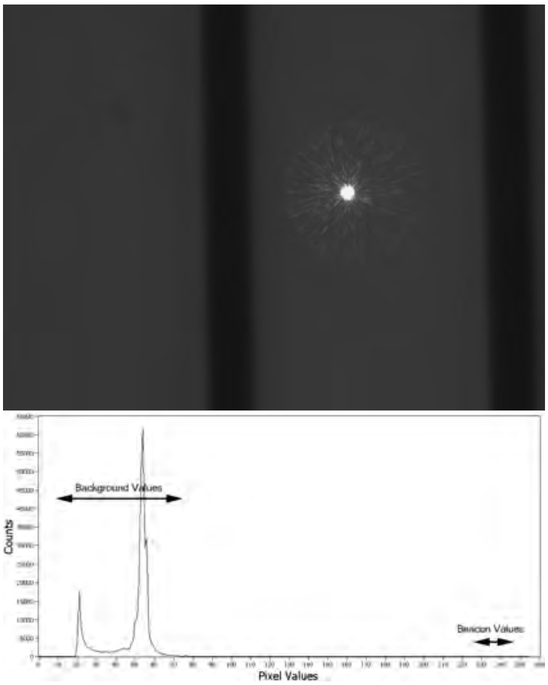


Figure 4.2: Image with a good histogram for blob extraction

given representation of the histogram, but from the image you can clearly see that the beacon laser is visible. Since there is a big gap in the histogram between the pixel values of the background light and the pixel values of the beacon laser it is easy to find a threshold value that separates the beacon laser from the background.

This shows that we can make the transition from a bad to a good image by only adjusting the shutter value. For calibrating the camera to continuously take good images, i.e. calculating a suitable shutter value, we have to use the knowledge about the characteristics of the images and the beacon laser.

The recorded images can be seen as a representation of the intensity of the light which is received by the camera, and therefore the pixel values of the background light can be seen as the representation of the current intensities of the background light. This also applies to the pixel values of the beacon laser. The power of the beacon laser has been calculated to be as high as the power of the background light received by the camera in the full field of view. As the intensity of the background light adds to the intensity of the beacon laser received at the camera, we can expect the intensity of the beacon laser to be twice as high as the intensity of the background light. In Trial Two, the filter will change this ratio in favor of the beacon laser.
As the pixel values in the images represent the intensities of the received light, we can say that the ratio of the intensities of the background light to the beacon laser is the same as the ratio of their pixel values. So if the beacon laser has been chosen to have at least twice the intensity of the background light at the camera of the FELT, the pixel value of the beacon laser will also be twice the pixel value of the background light. This knowledge can be used to calibrate the camera for recording images in which the pixel values of the background are located in the lower half of the histogram and the pixel values of the beacon laser are located in the upper half of it.

The camera calibration algorithm works as follows:

1. The first image is always taken with the camera settings set to default values

2. The histogram of the image is generated

3. The mode of the histogram is searched.
   Since most of the pixels in an image should represent parts of the background, the mode should always give a good assumption for the value of the background.

4. Now we can use the formulae from chapter 3.1.3 to "move" the mode in the next image to a designated position. For this, a value of 320 has been found to give good results (for 10-bit images). So the formula for calculating a suitable shutter value is as follows:

$$ShutterValue_{New} = \frac{270 \times ShutterValue_{Old}}{CalculatedPosition - DNO} \tag{4.1}$$

5. The calculated $ShutterValue_{New}$ is used as a setting for getting the next image and the algorithm is continued at step 2, with the next taken image as input.

With this algorithm, the mode of the histogram should always be a value of about 320 and therefore the beacon laser will always have pixel values above 640. So it is possible to use a threshold value of 600 for separate the beacon laser from the background.

The image shown in figure 4.2 has already been taken using this algorithm, so it already has been shown that this algorithm can produce good images. Now the algorithm will be tested in an environment with bad conditions for good beacon laser visibility. This is shown in figure 4.3 and 4.4. Because it was not possible to create a test scenario in the laboratory where exactly the same constraints apply as in real world conditions, these images are only suitable to illustrate the effect of the algorithm.

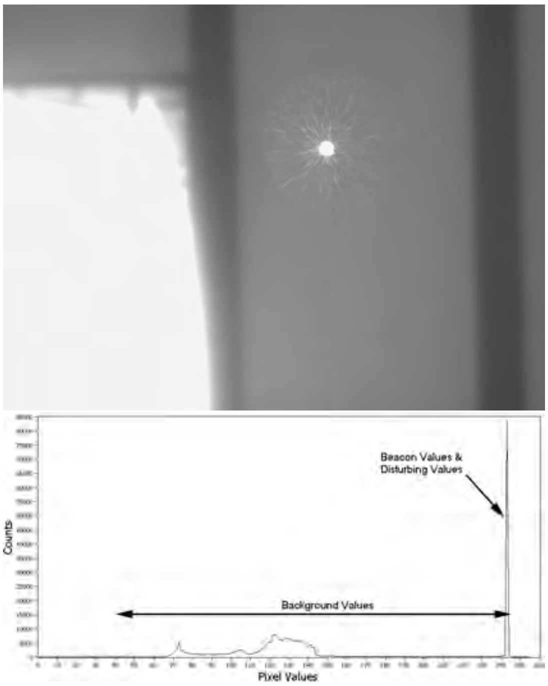


Figure 4.3: Image taken without camera calibration

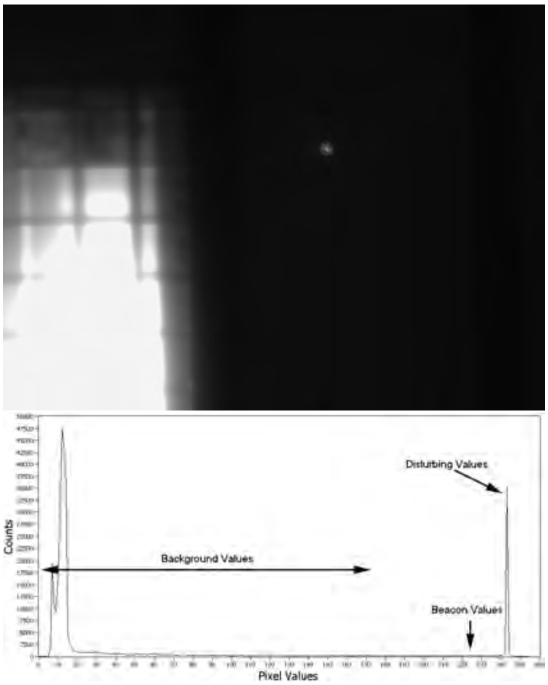


Figure 4.4: Image taken using the camera calibration algorithm

For taking these images the fluorescent lamp on the ceiling of the laboratory has been used as a disturbing object which is more powerful than the beacon laser. Even in this problematic environment, the algorithm produces good results.

The algorithm will give faulty results if the area of the disturbing object covers more than the half of the field of view, because in this case the light of the disturbing object gets misinterpreted as being the background light, since it will be the mode of the histogram. During Trial

2 we can exclude this situation from our considerations as it is planned to hold the Trial in the afternoon/night and such big reflections will be very unlikely. But it will be an issue for Trial 3 and therefore a better criteria for detecting the value of the background light will have to be developed by then.

4.2 Image Analysis

The image analysis is the key part of the acquisition of the beacon laser. It has to be fast to finish the acquisition in a reasonable time and it has to give reliable results for preventing the tracking of a false spot. As the rotational speed of the balloon can be controlled by rotating the periscope, we no longer need 1046 fps as mentioned at the beginning, but we targeted a frame rate of about 30 to 50 for the final system.

4.2.1 Considerations

For detecting the beacon laser we had two procedures in mind. The first one was to modulate the beacon by switching it on and off and to detect it in the images by generating difference images. The second one was to use a static beam and run a blob extraction algorithm on the images for extracting the beacon laser from it. In some systems a third approach is possible. This is the use of a polarized beacon and a polarization filter at the receiver, which can reduce the influence of background light by more than 50% and therefore it is most likely that the beacon will be the only visible spot. For the distances we have to overcome with the beacon laser, one needs powerful lasers. The only lasers available for application with high power requirements are multimode lasers, which can not be polarized.

The use of a static and a modulated beam will be discussed in the following part, but first we have to define the conditions we have for a successful visual detection of the beacon laser.

For detecting the beacon laser successfully we have to detect the beacon in at least two consecutive images. The second detection is needed to be able to begin the tracking of the spot. Otherwise it could be possible that the spot is detected but when the tracking begins, it has already disappeared from the image and we have to restart the acquisition. Another good thing about detecting the beacon laser in two consecutive frames is that we can calculate the current rotational speed with this information.

The formula for the velocity in one direction is given by

$$v = \frac{\Delta d * \#Pixels}{FieldOfView \times \Delta t} \tag{4.2}$$

where v is the speed, Δd is the distance between the spot positions in the two images, Δt is the time that has past between the two images and $\#Pixels$ is the overall number of pixels on the sensor in the direction of motion.

In our setup this results in the two formulae

$$v_h = \frac{\Delta d_h * 491}{4^\circ \times \Delta t} \quad \text{for the velocity in the hight axis}$$
$$v_w = \frac{\Delta d_w * 656}{5.4^\circ \times \Delta t} \quad \text{for the velocity in the width axis}$$

and finally

$$v_{approx} = \sqrt{v_h^2 + v_w^2} \tag{4.3}$$

for the approximated rotational velocity.

This information can be used for adjusting the rotational speed of the periscope to virtually lock the camera on the beacon laser.

With a modulated beacon, the detection of the beacon laser has some additional complexity compared to the detection of a static beam, but it has some advantages that made us think about it. The big advantage of a modulated beacon laser is that it is unique and the reflections on the ground will not be visible in the difference images. So this approach already contains the verification of the detected spot. The generation of a difference image should also be less complex than the extraction of blobs from an image, so it seems as if a higher frame rate could be possible using difference images. Unfortunately the rotation of the balloon is visible in the images, so for calculating the difference image this rotation has to be compensated by some extra calculations and this will reduce the processing frame rate. A further problem is that the frame rate of the camera has to fit for the frequency of the modulated beacon in order to enable the camera to record the images with the laser turned on and off, which is not a trivial task.

We finally decided to use a static beam and verify the detected spot via the RF link since the implementation of a modulated beacon seemed to be rather difficult or even impossible with the chosen beacon hardware.

4.2.2 Blob extraction Algorithm

For blob extraction from a single image, some criteria (features) for defining the blobs to extract have to be given. Common criteria used for this task are object color, shape or some pattern. Using a pattern- or shape-based approach seemed out of reach for us as these approaches normally need much processing power for calculating correlations, normalizations etc. So we decided to extract the blobs based on their grey values in the image.

The only thing we can say about the color of our beacon laser is that it will have a grey value higher than the background illumination. So we can use the value of the background light as threshold to create a binary image with the background having a value of 0 and all other objects having a value of 1. The threshold for the binarization of the images is derived from the camera calibration algorithm. The next step is to extract the marked objects to further reduce the number of candidates by comparing their sizes to the expected spot size calculated in chapter 3.6. A further suitable criteria for this would be the compactness of an object.

4.2.2.1 A Naive Algorithm for Blob Extraction

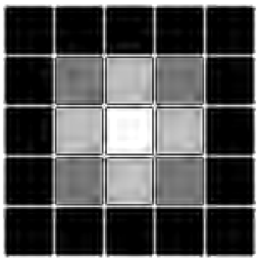


Figure 4.5: 4- and 8-Neighborhood

Blob extraction in a binary image can be achieved by two runs through the image. All operations are only done on pixels with a value of 1, the other pixels are ignored.

The first run scans the image from the upper left corner to the lower right corner. During the scan every pixel is compared with its neighbors in the row above and to its left, where the definition of a neighbor has to be given for every case.

Common neighborhoods are 4- and 8-Neighborhood. These neighborhoods are shown in figure 4.5. 4-Neighborhood (light grey in the figure) defines the neighbors as being those pixels which are the direct neighbors on the X- and Y-Axis to the current pixel. 8-Neighborhood (dark grey) includes the 4-Neighborhood and additional the four direct neighbors on the diagonal axes.

If one of the defined neighbors already has a label assigned to it, the pixel gets the same label assigned. If some of the neighbors have different labels, the pixel gets the lowest of these labels assigned. If the neighbors have no label at all, a new label is created and assigned to the pixel.

If the whole image has been processed, the first run is finished and the blobs should be marked by different labels. In some cases this produces blobs which are marked by two or more different labels as shown in figure 4.6. For correcting this error, a second run is needed. The second run runs from the lower right corner to the upper left corner of the image

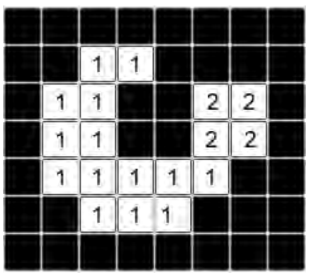


Figure 4.6: Labelling error after 1st run

and compares the pixels with its neighbors on the lower row and to its right using the same rules as for the first run.

After this run, all pixels belonging to the same object are marked with the same label. For calculating the number of blobs the number of labels has to be counted and for determining the blob positions, the center of gravity of a certain label can be calculated.

It is obvious that this algorithm can be optimized in terms of run time and gathered information, as this algorithm only runs directly on the image, produces no information about the blobs and does not make use of any additional data structure other than a counter for the labels.

4.2.2.2 Optimized Blob Extraction Algorithm

The first optimization is that the second run does not have to operate on all pixels of the image if the first run is also used for transferring the image data into a more efficient data structure.

A common principle for reducing image data is runlength encoding [Hab00, p. 261 ff.]. For doing this the image is linescanned and adjacent neighbors are no longer stored as separate pixels, but as intervals. These intervals contain the starting position and the length (number of adjacent pixels) of this interval. Pixels with non interesting values are discarded. In our case these are all pixels with a value of 0. So instead of storing the line containing

$$0000111000111110000000000000000$$

we only store

$$(4,3),(10,5)$$

The given interval represents three 1s starting at index 4 and five 1s starting at index 10. The 0s are not stored in this representation.

These intervals can now be linked to the blobs they belong to. With this data structure it is possible to work efficiently on the intervals for gathering information about the image content and for finishing the labelling algorithm. As we are not interested in further processing the image data, we can further compress the stored data by directly adding the pixels to the blobs they belong to. A blob consits of four counter variables. The first one is the size of the blob. Every time a pixel is added to the blob, this counter has to be raised by 1. Two other counters are used to take the sum of the x- and y-coordinates of all pixels added to the blob. By dividing each of these counters with the size of the blob we can calculate the center of gravity for this blob. As we decided that the size of a blob may not be sufficient to distinguish the blob of the beacon laser from other blobs, a counter for the pixels sitting on the edge of the blob has been added. This counter gives the circumference of the blob. A pixel is detected to be on the edge of a blob if not all of its direct neighbors on the X- and Y-Axis are part of this blob. Based on the size and the circumference of a blob we can calculate its compactness which is defined as:

$$Compactness = \frac{Circumference^2}{Size} \tag{4.4}$$

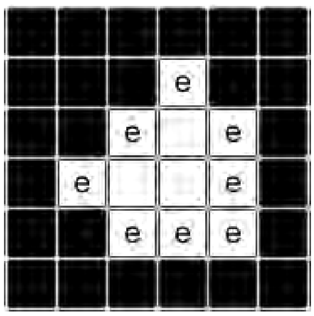


Figure 4.7: The pixels marked with an "e" are defined as being on the edge of this blob

Unfortunately the compactness does not always give usefull results for small objects as shown in figure 4.8, but as the beacon laser is expected to produce a spot bigger than shown in the example we hope that the compactness could be still usefull for us.

With the new data structure the second run has only to be done on a heavily reduced amount of data and it generates additional information about the extracted blobs. This saves us followup runs through the image compared to the naive algorithm.

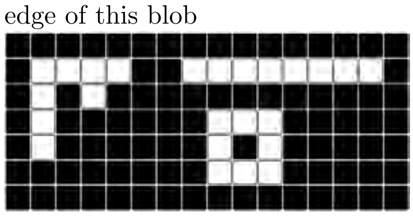


Figure 4.8: Blobs with same size and same compactness

Tests with an implementation in C of the naive and the optimized algorithm on the CVS have given frame rates of 15 fps for the naive and 36 fps for the optimized algorithm with images taken at full resolution. The implementation done with LabView reached 22 fps. The code for all three implementations was not optimized, so better results are possible.

4.3 Periscope Control Theory

In figure 2.3 you can see that the periscope is able to rotate around two axes. These axes are used to set the azimuth and nadir angle of the view of the camera on the FELT. The construction of the periscope causes the elevation axis to rotate when the azimuth axis is rotated. The ratio of the rotation is 1:-1 which means, when the azimuth axis is doing one revolution in a give direction, the nadir axis does one revolution in the opposite direction. If the nadir angle shall not change while the azimuth axis is rotating, it is required to rotate the nadir axis the same way as the azimuth axis is rotated. This behaviour is a trade-off for enabling the periscope to rotate endlessly around the azimuth axis.
If the nadir angle is supposed be changed, the nadir axis has to be rotated faster or slower than the azimuth axis. The nadir angle can be calculated according to table B.6 by:

$$NadirAngle = \frac{P_{Na} - P_{Az} - D_{start}}{0.0005} \tag{4.5}$$

where P_{Na} and P_{Az} are the positions of the nadir and azimuth axis given by the corresponding encoders and

$$D_{start} = P_{Na_0} - P_{Az_0}$$

where P_{Na_0} and P_{Az_0} are the initial positions of the two axes. In the initialized state, the nadir angle is always 0°, so this position is taken as reference position for calculating the nadir angle.

Because of the drift of the balloon it is necessary to permanently adjust the nadir angle during the acquisition scan. Since this also involves permanently measuring the positions and velocities of the two axes, this is best done by using a controller.

When the PAT system is tracking the beacon laser, it is necessary to permanently adjust the rotation of the two axes for compensating the motion of the balloon and keeping the beacon laser in the center of the image. This is best done with a controller, too. Keeping the two axes in constant motion based on information gathered from the images enables the system to "follow" the beacon laser even if it is not visible in a few consecutive images. This is a big advantage over systems which update the positions of the axes directly from the image information, since it makes the system more robust against fades of the beacon laser.

4.3.1 Control Theory Basics

If a system is well known, it is possible to control its output value y by using an open-loop controller. This is shown in figure 4.9.

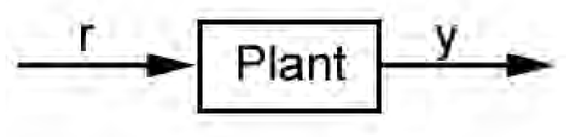


Figure 4.9: Block diagram for an open-loop controller

Since the system given for the open-loop controller is well known, the plant P can be described by a function $P(r)$ which takes the reference value r as input and produces the output value y. This approach has been used for calibrating the camera (see chapter 4.1), where a model of the pixel values is used for calculating the best shutter value.

In many cases, a system is influenced by various internal or external disturbances like parameter variations, delays, other systems, mechanical influences and so on. Measuring or predicting all these disturbances is normally very difficult or even impossible. The same applies to creating an accurate model of the system, so this approach is no longer feasible.

In such a case, the system is extended to receive feedback from the system output. With the fed back value of the output y of the system, the system is able to calculate its current system error $e = r - y$. Instead of using e as input for the plant, it is used as input of a controller C, which calculates the input value u for the plant. A block diagram of a closed-loop controlled system is given in figure 4.10.

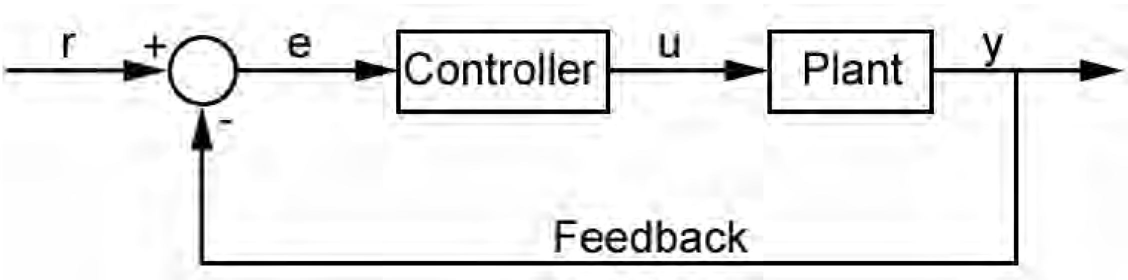


Figure 4.10: Block diagram of a closed-loop controlled system

The use of feedback from the system output enables the controller to estimate the behavior of the system and to calculate suitable input values for the requested reference values. With using this controller layout, it is no longer necessary to develop a model for the whole system, but to develop a model for the controller. This is also not a trivial task, but it is easier than modelling the system.

The development of a model for the controller is always a trade-off between cost and benefit of the model. The more accurate the model is supposed to be, the more expensive it will be to develop and in most cases such a high accuracy is not needed and developing an optimal model would not be worth the effort.

Commonly used approaches for controller models are P-, PI-, and PID- controllers, since these controllers provide a good trade-off between cost and benefit [RK02, p. 67-68]. The following section explains the concept of PID controllers.

4.3.2 PID Controller

The term PID is an abbreviation for proportional, integral and derivative and describes the parts of the mathematic model of the controller. The controllers are given by the following formulae which calculate the input value u for the plant from the error value e:

$$\begin{aligned}
\text{Proportional Controller:} \quad & u_t = P \times e_t \\
\text{Proportional-Integral Controller:} \quad & u_t = P \times e_t + I \times \int e_t dt \\
\text{Proportional-Integral-Derivative Controller:} \quad & u_t = P \times e_t + I \times \int e_t dt + I \times e'_t
\end{aligned} \tag{4.6}$$

where u_t and e_t are the input and error value at time t [RK02, p. 35].
With this, the transfer function $C(s)$ for a PID controller is given as [Föl94, p. 33-37]:

$$C(s) = P + \frac{I}{s} + Ds = \frac{Ds^2 + Ps + I}{s} \tag{4.7}$$

The effects of raising P, I, and D on the system are given in table 4.1. This can be used as rule of thumb for tuning the controller. In this case, tuning means finding suitable values for P, I, and D.

Parameter	Rise Time	Overshoot	Settling Time	Sum Squared Error
P	Decrease	Increase	Small Change	Decrease
I	Decrease	Increase	Increase	Eliminate
D	Small Change	Decrease	Decrease	Small Change

Table 4.1: Effects on the system of raising the values for P, I, and D [Wik05]

4.3.3 Tuning the PID Controller

Once the system has been set up for using the PID controller, suitable values for P, I, and D have to be found. This can be done in several ways and depends on the system the controller is targeted for and it is not always necessary to set all three attributes.

As an example for tuning a PID controller, the PID controller for the persiscope in the acquisition phase is used.

The periscope can oscillate around its axes without taking damage, so it has been decided to use the Ziegler-Nichols method described in [Wik05] for tuning the controller. Other methods have been suggested by Ziegler-Nichols, Oppelt, Rosenberg and Chien-Hornes-Reswick and can be found in [Mor05]. All methods have in common that they give certain rules for calculating the values of P, I, and D after some measurements have been done.

The Ziegler-Nichols method is based on step changing the reference value for the system and measuring the system output response. First the response of the system using $P = 1$, $I = 0$ and $D = 0$ is given in figure 4.11. With this setting, the controller is forwarding the error to the plant and not doing any changes to it.

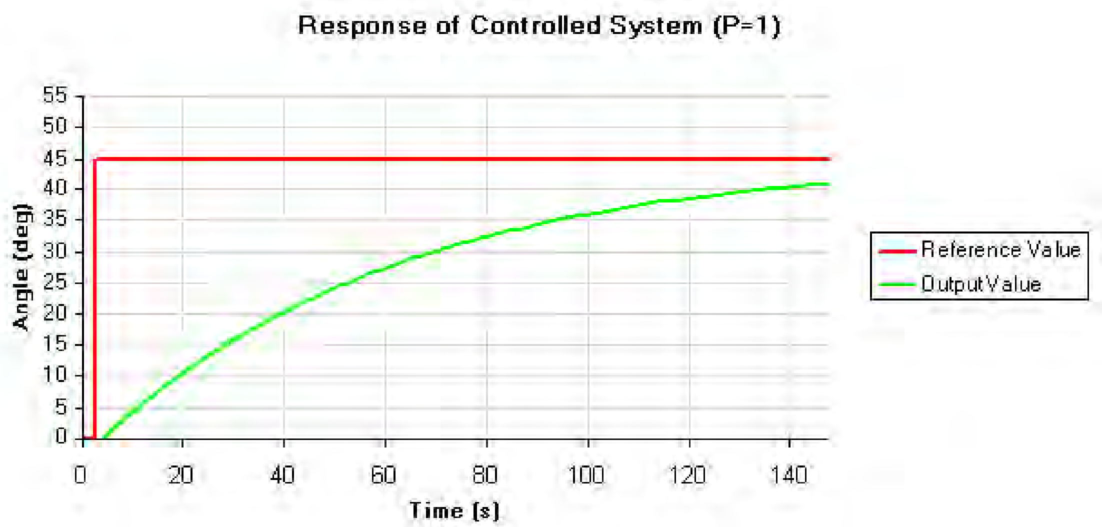


Figure 4.11: Step change response of the controlled system with $P = 1$, $I = 0$ and $D = 0$

This figure shows why the controller needs to be tuned. First, the output value does not reach the reference value and second, the rise time of the system is longer than 140 seconds. Both characteristics are not acceptable for the PAT system as it requires high pointing accuracy.

Since optimization of a controller can be a very complex and time consuming task, it is common practice to predefine certain goals for the quality of the system and stop tuning the system when these goals are reached.

For the quality of a system two values are significant. The first value is the rise time of the output

value in response to the step change. Since the nadir angle is not expected to change with huge steps during the acquisition phase, the requirements for the rise time are not very strong in the PAT system. It will be sufficient if the system reacts with a rise time of 10 second to a step change of 45°.
The second significant value is the settling time after the reference value has been reached. This value is more important because a long settling time means also an inaccuracy in pointing the camera. As a dynamical system might never settle exactly at the reference value, it is quite common to define a range as target for the settling of the system. Since the field of view of the camera is 4° and the GPS error is neglectable, it would be sufficient for the PAT system if the nadir angle stays within a range of ±1° around the reference value. Nevertheless a accuracy of ±0.5° is targeted.

According to the Ziegler-Nichols method, the first step for tuning the controller is to set the I and D values to zero and to raise the P value until the system starts to oscillate continuously. This value is called P_{crit} and the measurements give it a value of 630. The period duration of the oscillation is called T_{crit}. The measurement is shown in figure 4.12.

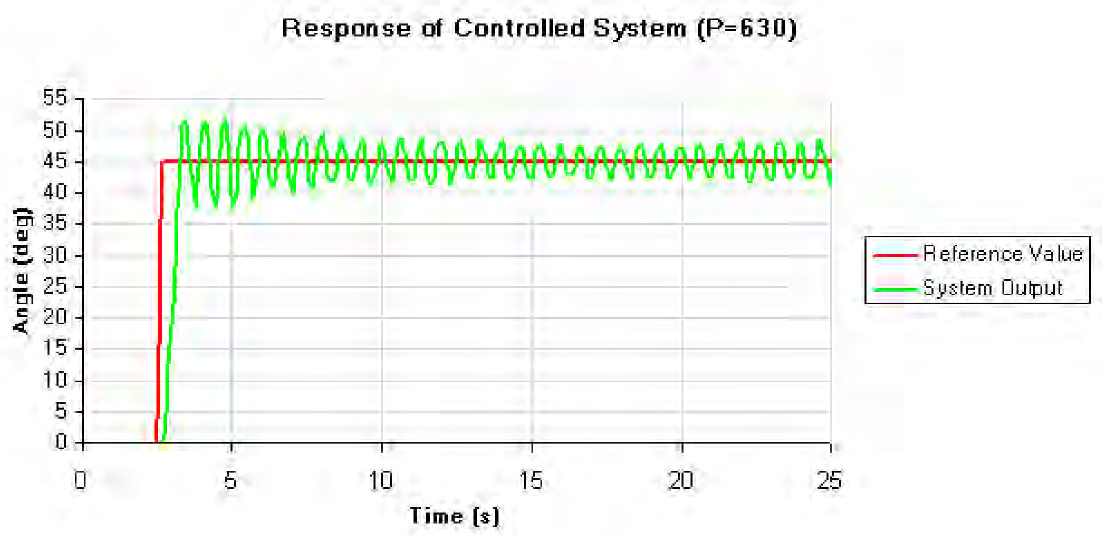


Figure 4.12: Controlled system with an oscillating step change response. $P = 630$, $I = 0$ and $D = 0$

With this value for P, the rise time is now approximately six seconds and the output value reaches the reference value. The oscillation around the reference value has an amplitude of approximately 3 degrees and the oscillation period is approximately 0.48 seconds.

Now the values for P, I, and D can be calculated using some simple rules.
According to Ziegler-Nichols, the rule for calculating P for a P controller is $0.5 \times P_{crit}$. This value results in the response of the PAT system given in figure 4.13.

With the calculated value of 315 for P, the rise time is now below one second and the settling time is also below one second. The output value settles with an accuracy of 0.005°. Ziegler-Nichols give further instructions (see table 4.2) on how to calculate the values of P, I, and D, but these values did not give better results for the system. Since the received results are well within the targeted values, no further optimization efforts have been done.
The whole tuning process has been redone width different step sizes and gave always the same result.

The controller for the tracking of the beacon laser will be implemented in a similar way.

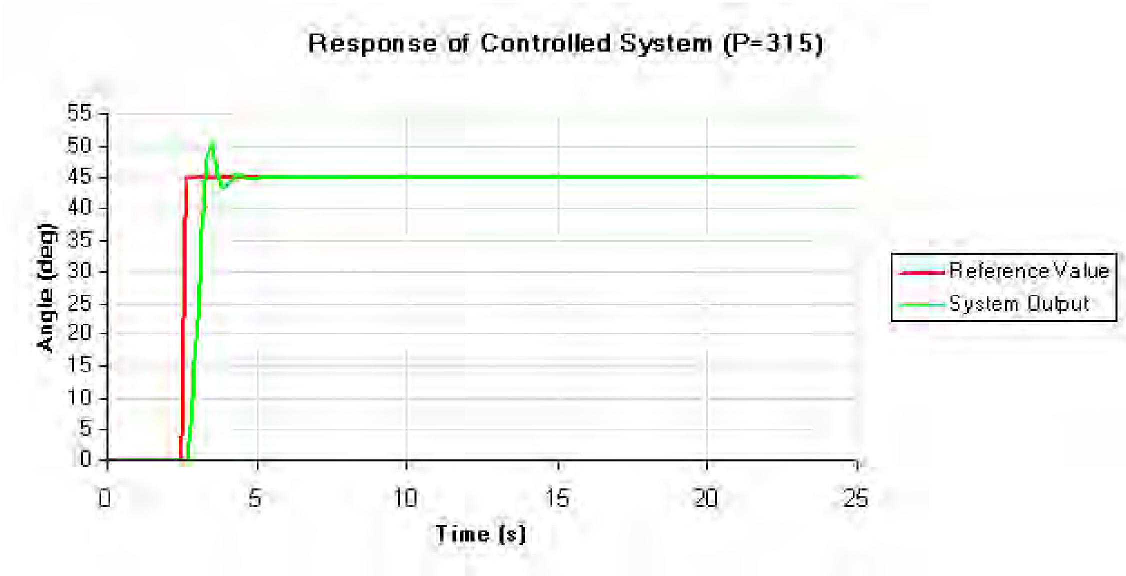


Figure 4.13: Step change response of the controlled system using the value for P suggested by Ziegler-Nichols. $P = 315$, $I = 0$ and $D = 0$

Controller	P	T_n	T_v
P Controller	$P = P_{crit} \times 0.5$		
PI Controller	$P = P_{crit} \times 0.45$	$T_n = 0.85 \times T_{crit}$	
PID Controller	$P = P_{crit} \times 0.6$	$T_n = 0.5 \times T_{crit}$	$T_v = 0.12 \times T_crit$

Table 4.2: Tuning Rules given by Ziegler-Nichols: $I = \frac{P}{T_n}$, D $= P \times T_v$

4.4 Pointing, Acquisition and Tracking

As mentioned in chapter 1.3.3.1 the system can not use positioning data to perform the pointing phase. A compass was considered to enable accurate pointing. However, due to electro magnetic interference caused by other hardware parts, it was determined that a compass may give unreliable results.
Another idea for getting additional information about the heading of the optical payload was to add a sun sensor to the system, but there had not been enough time given in the project planning for assembling and installing one.

Scanning an uncertainty area that is bigger than the field of view for the receiver is normally done in either raster scanning or spiral scanning or a combination of these two methods. Raster scan means that the uncertainty area is scanned line by line, starting at one of its corners while spiral scan means that the uncertainty area is scanned in a circular way, starting at the center. Because our system is constantly moving, in particular rotating, these scan patterns do not work for us. To overcome this problem and to reduce the size of the uncertainty area, the system will calculate a circle on the Earth's surface on which the ground station will be located. This circle is calculated from the GPS data of the balloon and the ground station.

Actually, only the nadir angle between the balloon and the ground station has to be calculated. The nadir angle is defined as 0° if the ground station is located exactly below the balloon and as 90° if the ground station is located at the same altitude as the balloon.
If the nadir angle is known, the system has to perform a scan on the corresponding circle for the beacon laser. During the scan, the nadir angle has to be updated continuously due to the drift of the balloon.

For controlling the speed of the scan over ground, the gyroscope data is used. Without this information, it would be hard to perform the scan in a reasonable time. As we can control the speed of the scan, it is also possible to adapt the scanning speed to the frequency of the image analysis, which is also a great help. The acquisition with the support of a gyroscope is significantly faster and less complex than without a gyroscope, since the rotation rate of the balloon is known. Nevertheless, a scan is necessary to locate the beacon spot because the absolute heading of the balloon cannot be read from the gyroscope, only the rotation rate. Gyroscopes with a high temporal stability are exceedingly expensive and therefore cannot be used in the trial. Since the rotation rate of the balloon is known, the scan velocity can be adapted to get an optimal relative scan velocity between balloon rotation and periscope rotation, which minimizes the time required for the scan and to ensure a detection of the spot. If the relative scan velocity is too high in relation to the capabilities of the image analysis, the system might miss the beacon spot.

For detecting the beacon laser in the recorded images, the image analysis algorithm from section 4.2 is used. As there can be false spots caused by reflections returned by the image analysis, it is necessary that the acquired beacon is verified to be the correct one. If the the system is sure that it has discovered the beacon laser, it can start to track it.

4.4.1 Processing the GPS Data

The balloon and the ground station are each equipped with GPS receivers for determining their positions. With this information we are able to calculate the angle between the two systems. Unfortunately these positions are not completely accurate since the GPS system always has some inaccuracies. These inaccuracies are dependent on the number of visible GPS satelites and the position of the receiver. When we know how the angle between the two systems is calculated, we can calculate the impact of these errors.

4.4.1.1 Calculating Angle between two Positions

GPS uses the geodetic reference system WGS84 (World Geodetic System 1984) for giving positioning data. In this reference system an ellipsoid is used to represent the earth surface as reference for the height of a position. The ellipsoid is defined by two values: a, which denotes the equatorial radius, and f, which is the flattening of the earth spheroid. These parameters are defined in the system as $a = 6378.137km$ and $1/f = 298.257223563$ (reciprocal of the flattening). The three axis of the system are defined as latitude, longitude and ellipsoid or as the following Cartesian coordinates:

- The origin of the axes is the center of the earth (ellipsoid)
- The x-axis goes through the equator in the direction of the Greenwich meridian (0° longitude)
- The z-axis points through the north pole
- The y-axis is chosen, so that the x-, y-, and z-axis form a right hand system

For giving angles between two objects, commonly two angles are used. Azimuth which describes the angle in the horizontal axis and elevation which describes the angle in the vertical axis. For flying objects, the nadir angle is used instead of the elevation angle. Before we can calculate these angles from the GPS data we have to transfer the WGS84 coordinates into the Cartesian coordinate system.

The conversion between the two systems can be done by the following formula [alm99, p. K12]:

$$\overrightarrow{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (aC+h)\,cos(lat)cos(long) \\ (aC+h)\,cos(lat)sin(long) \\ (aS+h)\,sin(lat) \end{pmatrix}$$

where

$$C = \{cos^2(lat) + (1-f)^2 sin^2(lat)\}^{-\frac{1}{2}}$$
$$S = (1-f)^2 C$$

The positions of the two objects are given by the vectors $\vec{v_1}$ and $\vec{v_2}$, for which the GPS positions in the WGS84 system are known. Both locations are described by geodetic latitude lat, geodetic longitude $long$, and height h. The station at position $\vec{v_1}$ points at location $\vec{v_2}$.

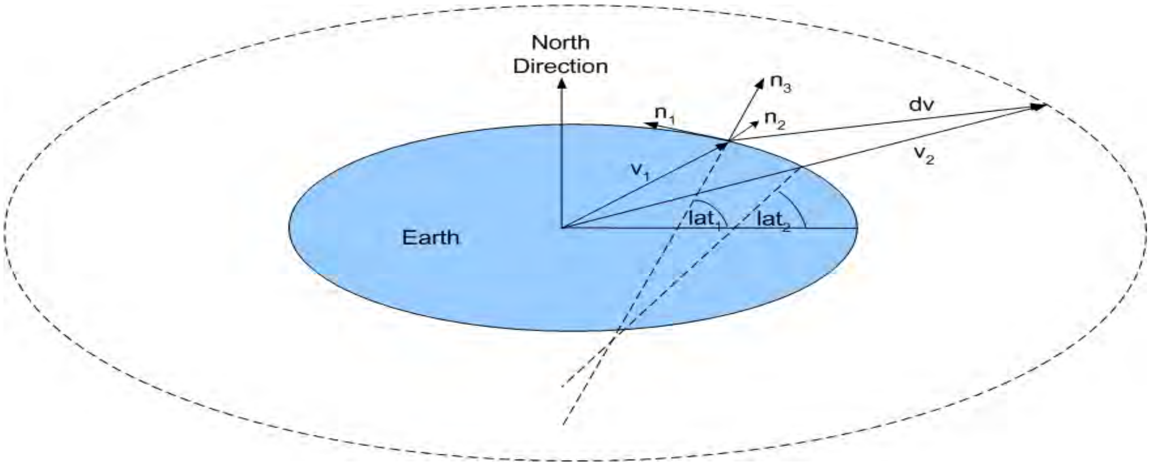


Figure 4.14: Vector system for calculating angle between two positions

The difference vector between $\vec{v_1}$ and $\vec{v_2}$ is given by:

$$\vec{dv} = \vec{v_1} - \vec{v_2}$$

The derivative of $\vec{v}$ in respect to the latitude is given by

$$\vec{n_1} = \frac{\Delta}{\Delta lat}\vec{v} = \begin{pmatrix} \frac{\Delta x}{\Delta lat} \\ \frac{\Delta y}{\Delta lat} \\ \frac{\Delta z}{\Delta lat} \end{pmatrix} =$$

$$= \begin{pmatrix} \{-a \times C \times sin(lat) \times \left[1 + C^2 \times cos^2(lat) \times \left(f^2 - 2f\right)\right] - h \times sin(lat)\} \times cos(lat) \\ \{-a \times C \times sin(lat) \times \left[1 + C^2 \times cos^2(lat) \times \left(f^2 - 2f\right)\right] - h \times sin(lat)\} \times sin(lat) \\ a \times S \times cos(lat) \times \left[1 - C^2 \times sin^2(lat) \times \left(f^2 - 2f\right)\right] + h \times cos(lat) \end{pmatrix}$$

$\vec{n_1}$ is the first determining vector of the surface tangent to the Earth spheroid at the target position (lat,long,h). To ensure that this vector points to the north pole, the z-component should be greater zero. Otherwise the vector has to be multiplied by -1. $\vec{n_1}$ is normalized to a length of 1.

The second determining vector n_2 is calculated by a cross product of the northern direction $\vec{e_z} = \left(\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right)$ and $\vec{n_1}$:

$$\vec{n_2} = \vec{e_z} \otimes \vec{n_1}$$

$\vec{n_2}$ is also normalized to 1. The normal to the surface $\vec{n_3}$ is given by the cross product:

$$\vec{n_3} = -\vec{n_1} \otimes \vec{n_2}$$

A negative sign is added to point the vector away from the Earth center. The image vector $\vec{v_0}$ of the difference vector $\vec{d_v}$ in the surface plane $\vec{n_1} - \vec{n_2}$ is given by

$$\vec{v_0} = \vec{n_1} \times \left(\vec{n_1} \bullet \vec{dv}\right) + \vec{n_2} \times \left(\vec{n_2} \bullet \vec{dv}\right)$$

where the scalar product is denoted by $\bullet$. With the given formulae the elevation β' and azimuth α' angle are given by

$$\alpha' = acos\left(\frac{\vec{v_0} \bullet \vec{n_1}}{\|\vec{v_0}\|}\right)$$

$$\beta' = acos\left(\frac{\vec{v_0} \bullet \vec{dv}}{\|\vec{dv}\|}\right)$$

To get the correct values of the azimuth and elevation, α' and β' have to be modified under the following conditions:

$$\alpha = \begin{cases} 2\pi - \alpha' & if \quad \overrightarrow{v_0} \bullet \overrightarrow{n_2} < 0 \\ \alpha' & else \end{cases}$$

$$\beta = \begin{cases} -\beta' & if \quad \overrightarrow{dv} \bullet \overrightarrow{n_3} < 0 \\ \beta' & else \end{cases}$$

4.4.1.2 Error contained in the GPS Information

The balloon operator stated the maximum error of the GPS system on the FELT as $\leq \pm 50m$ horizontally and $\leq \pm 100m$ vertically.
The ground station GPS data will have a maximum error of $\leq \pm 30m$ horizontally and $\leq \pm 50m$ vertically.

We used the developed formulae to calculate the maximum error we could expect for the calculated elevation and azimuth angles. As the error is very small compared to the distances we will have between the ground station and the balloon, the angular error is also very small. Using the previously given formulae, a maximum error of ± 0.15 degree for both angles has been calculated.

As the field of view of the camera is four degrees, this error will not affect the quality of the acquisition.

4.4.2 Circular Scan for Ground Station

The speed of the scan is essential for the total duration of the acquisition phase. If the scan speed over ground is too slow, the acquisition of the beacon laser will take too long. If the scan speed is too high, it might not even be possible for the image analysis to detect the beacon laser.

4.4.2.1 Determining the Ideal Scan Speed

The term scan speed used in this thesis is meant as the speed of the scan over ground i.e. it is the relative speed between the rotational speed of the periscope and the rotational speed of the balloon. The direction of this scan does not matter as the acquisition can be done in both directions.

The given parameters for the calculations are a maximum rotational speed of the balloon of 36 deg/s and the field of view of the camera is 4°. The processing frame rate of the camera will have to be measured after the implementation, but it is assumed that a frame rate between 30 and 50fps for processing full sized frames can be reached.

The maximum possible scan speed is mainly determined by the capabilities of the image analysis. With the given parameters it is possible to calculate the maximum speed as a function of the processed frame rate (two detections of the spot are required):

$$MaximumScanSpeed = ProcessedFrameRate \times \frac{FieldOfView \quad (= 4^\circ)}{2 \quad (detections)} \tag{4.8}$$

The exact scan speed will have to be discussed after the image analysis has been implemented.

As the direction of the scan is not relevant for a successful detection, the periscope will be rotated with the balloons rotation to get a higher scan speed than the rotational speed of the balloon and against it to get lower speeds. For example, if the balloon is rotating with 36 deg/s and a scan speed of 60 deg/s is targeted, the periscope would have to rotate with a speed of 96 deg/s against the rotation of the balloon. If the periscope is rotated with the rotation of the balloon, only a periscope speed of 24 deg/s is needed.

This effect also has to be considered if we try to determine the optimal scan speed without the information of the gyroscope. This could be the case if the gyroscope is broken. A broken

Processed Frame Rate	Max. Scan Speed	Max. Time for Acquisition
10fps	20deg/s	18s
20fps	40deg/s	9s
30fps	60deg/s	6s
40fps	80deg/s	4.5s
50fps	100deg/s	3.6s

Table 4.3: Maximum scan speeds in dependency of the maximum processed frame rate

gyroscope can be detected by measuring the duration of the current scan. If the beacon laser has not been detected after a certain amount of time (see table 4.3) it is most likely that the information of the gyroscope is wrong and the system is running the scan with unknown speed.

In case of acquisition without gyroscope support, the scan speed is not known and can only be guessed. When guessing, it is always possible that the chosen periscope speed is resulting in a very slow scan speed and the acquisition will not finish in a reasonable time, e.g. if the balloon is rotating with 36 deg/s in one direction and the periscope is rotating with 40 deg/s against it, the resulting scan speed is 4 deg/s. It is also possible that the chosen periscope speed results in a scan speed too high for the image analysis. So if the gyroscope is broken, the right choice of speed will have to be determined by trial and error.

Periscope Speed	Min. Scan Speed	Max. Scan Speed	Req. Processed Frame Rate	Max. Time for Acquisition
10deg/s	0deg/s	46deg/s	23fps	∞
20deg/s	0deg/s	56deg/s	28fps	∞
30deg/s	0deg/s	66deg/s	33fps	∞
40deg/s	4deg/s	76deg/s	38fps	90s
50deg/s	14deg/s	86deg/s	43fps	25.71s

Table 4.4: Periscope speeds and their constraints for acquisition without gyroscope support

Table 4.4 shows that if the system could reach a processed frame rate of 40 fps it could finish the acquisition without gyroscope support within 90 seconds. Another observation from this table is that if the system has a processed frame rate of 33 fps and does a scan with 30 deg/s periscope speed without success, then the only reason can be that the resulting scan speed is close to 0 deg/s, i.e. the balloon is rotating with approximately 30 deg/s in the opposite direction. If the system now does a scan with a periscope speed of 30 deg/s in the opposite direction, the resulting scan speed will be around 60 deg/s and the acquisition should finish within approximately six seconds (Table 4.3). A good strategy for a scan without gyroscope support is to run a scan with 30 deg/s periscope speed in any direction. If that scan is not finished within 12 s it is obvious that the balloon is rotating in the opposite direction. If a scan is now done in the opposite direction the acquisition should finish within the next 12 s. So the overall scan time is 24 seconds which is the same as for the acquisition with gyroscope support and a scan speed of 15 deg/s.

4.4.2.2 Scanning Algorithm

The scan algorithm works as follows:

1. The scan velocity is set according to the previous considerations and the performance of the final system.
2. The scan is started.
3. If the beacon laser is detected in the images, the control theory from section 4.3 is used to center the beacon laser in the images.
4. The area of interest of the camera is changed to a smaller size and the tracking loop is started.

If the beacon laser can not be detected in step 3, the scan has to be restarted with a different speed. This scan follows the previous considerations for a scan without support by a gyroscope.

If problems within the GPS system occur, it is possible to do the acquisition scan without GPS information. This is done by starting the acquisition scan with a nadir angle of 1 degree. After every revolution around the azimuth axis, the nadir angle is raised by 1 degree. Sooner or later, the beacon laser should be detected by the PAT system. Raising the nadir angle by only 1 degree instead of 4, which is the size of the field of view, can be seen as a safety margin because of the pendulous movement of the balloon. If the gyroscope is also broken, then the duration for one revolution has to be done based on the done in the previous section.

4.4.3 Tracking

As the position of the spot in the image is already known during the tracking, the system no longer has to analyze the whole image. It will be sufficient if it analyzes only a small area around the discovered spot for detecting its motion (actually this is the motion of the camera/balloon). As no longer the whole image is needed for the analysis, it is now possible to use the area of interest feature of the camera for enabling higher frame rates than 100 fps. This is usefull as the basic rule for the tracking is: the faster the image analysis is, the more accurate the pointing is and therefore, a better data transmission is possible.

Even if the link budget for the beacon laser has been calculated to include fades of up to 10 dB, it might be possible that some fades cause the beacon laser to disappear from the images. Typically, the duration of these fades stay below 10 ms. For compensating these effects, the periscope has to rotate continuously against the movement of the balloon. If the rotation of the periscope is well adjusted to the movement of the balloon and the beacon laser disappears from the images due to a strong fade, the camera's field of view should stay over the position of the beacon laser for longer than 10 ms.

Therefore the goal of the tracking algorithm is to keep the spot in the center of the image at all times using the control theory from section 4.3. As soon as the spot moves away from the center, the movement of the periscope has to be adjusted to recenter the camera's field of view over the beacon laser. For determining the position of the beacon laser in the images, a simple center of gravity (COG) algorithm is used.

$$COG_x = \frac{\sum_x \sum_y x \times P(x,y)}{\Sigma_x \Sigma_y P(x,y)} \qquad COG_y = \frac{\sum_x \sum_y y \times P(x,y)}{\Sigma_x \Sigma_y P(x,y)} \tag{4.9}$$

where $P(x, y)$ denotes the value of the pixel at position x,y in the 8-bit image.

If the center of gravity is known, its horizontal and vertical angular distance from the center of the image can be calculated from the field of view of the camera.

$$\begin{aligned} \Delta d_h &= (x_C - COG_x) \times \frac{4^\circ}{491} \\ \Delta d_w &= (y_C - COG_y) \times \frac{5.4^\circ}{565} \end{aligned} \tag{4.10}$$

where x_C and y_C are the coordinates of the center of the image.
This data can be used as input for the periscope controller.

Chapter 5

Implementation Details

5.1 Programming Languages Used

The programming languages used have mainly been determined by the chosen hardware. As mentioned before, the main programming language was LabView from National Instruments, as this is the default language for use with the CVS. When LabView was not suited for a task, C has been used.

LabView is a graphical programming language in which the data flow of the program is modeled. This approach may be confusing for people who are used to functional or object oriented programming languages.

Another important point for choosing LabView was that the development environment contains a lot of prebuilt modules that could be used for the software. For example the Basler 602f camera could be used right out of the box. The graphical user interfaces could be done with a few drags and drops and setting up the TCP/IP connection and the communication between the software modules was also done within a few clicks.
But LabView also had some disadvantages for the implementation.

First, LabView does not have a concept of tasks that can be started, grouped, stopped, etc. for writing multithreaded applications. For doing this, you have to use the so called Timed Loops which are nothing more than while-loops with some timing constraints attached to it. As a while-loop can only be started once (except when it is nested in another loop), you have to write your "tasks" as endless running loops which are not so easy to synchronize.

Second, when working with arrays, the performance of LabView is quite poor as there is internally a lot of copying around of the array data. Unluckily, the image analysis is done on arrays that contain the image data, so LabView was not suitable for this part. For implementing the image analysis, C was chosen as programming language.

Third, the diagrams in which the software is modeled get quickly crowded and complex. So it is not always easy to keep an overview of the different parts of the software. Writing comments in the code is also problematic as they have to be inserted in text boxes in the diagrams, which make the diagrams grow bigger and sometimes more confusing. It is also not possible to comment out some parts of the program for testing and debugging. These problems can also be seen as a kind of advantage for LabView as this forces the developers to plan the software well ahead and to keep additional documentation for maintaining an overview of the software.

Fourth, data between different parts of the software can not be shared by using variables or similar methods. Instead, one has to use a construct called a Notifier, which is not as easy to use as just setting or getting the value of a variable.

Because the diagrams of the developed software are very big and unhandy for printing, they would fill many pages but not provide much information. Therefore it has been decided not to include them in this thesis. Only the developed C code is given in appendix C.

5.2 Software Design

The software on the FELT has been designed in a modular way, as parts of the software had to be changed for the different development and test setups. For example, at the beginning of the software development, there was no idea of what the periscope will look like and how it will be controlled, so a simple construction of two servos and a mirror were used until the periscope was available.

The splitting of the software was done according to the different tasks the software had to accomplish and according to the parts of the hardware they use. For preventing any unwanted interconnections between the modules, there should be, for example, only one module interacting with the camera and only one module controlling the periscope.

An overview of the software structure is given in figure 5.1

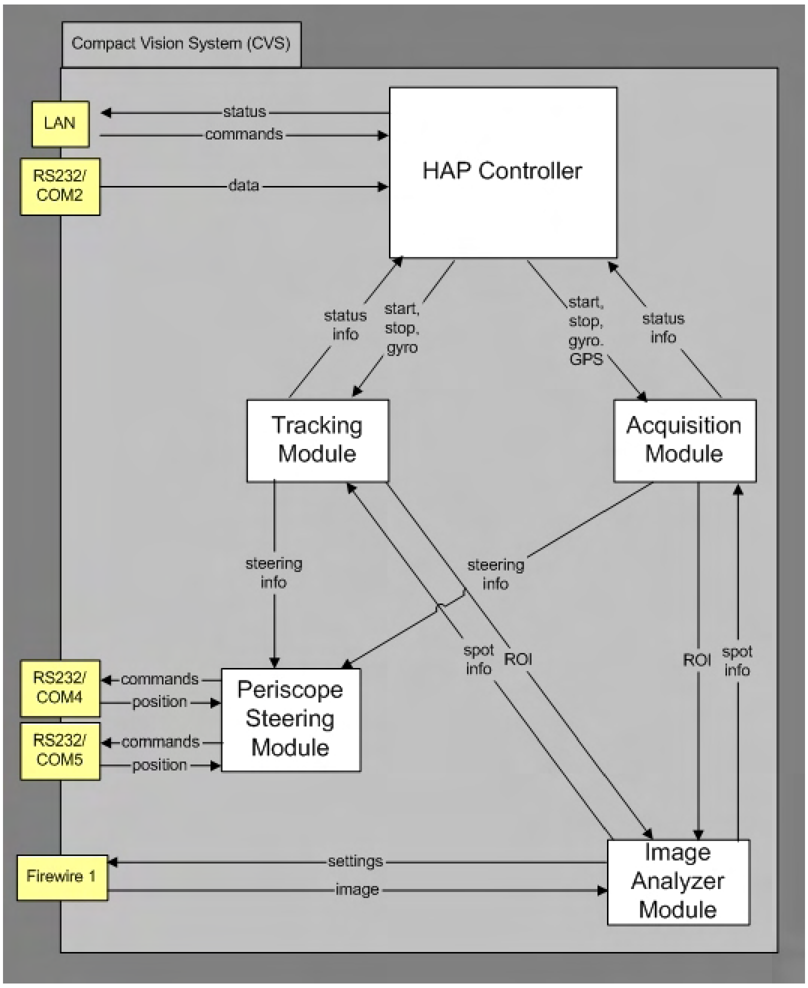


Figure 5.1: FELT software structure

5.3 Module Description

The software has been split into six modules which will be described in this section. Also, some smaller software parts were developed (like a driver for reading the gyroscope data), but these parts are rather straight forward and therefore not described in this thesis.

5.3.1 Ground Station

Actually, the ground station is not part of the FELT software, but since it will control the FELT software via the RF link, it has been developed together with the FELT software and can be seen as part of it. As the ground station is rather complex, only the part interacting with the FELT is described within this section.

The ground station software has a simple GUI for manually sending commands to, and for displaying data received from, the FELT software. Internally, it mainly consists of two Timed Loops. The first one handles the GUI inputs and sends the commands to the FELT software. The second loop manages the TCP/IP connection with the FELT and receives and processes the data coming from the FELT. The protocol used for communication between ground station and FELT is given in appendix B.4. As the bandwidth of the RF link is heavily limited ($<$ 9.6kb/s) and not only used for the communication between our two software parts, the communication between these two parts has been designed for a small footprint. Things like transmitting images from the tracking camera are therefore out of scope.

The GPS data is not part of our system. Instead, it is transmitted from the balloon to the balloon operation center and then forwarded to the ground station. Instead of transmitting both GPS positions to the FELT, the ground station does the calculation of the current nadir angle for the FELT and only transmits the calculated angle. This will be done every second because this is the update rate of the GPS information. As the GPS information will be transmitted every second it will also be used as a ping signal for detecting a broken connection.

5.3.2 FELT Controller

The FELT Controller Module is responsible for the management of the TCP/IP server and the communication with the ground station. It permanently listens for incoming commands from the ground station and controls the work flow of the other software modules on the CVS. The FELT Controller Module also has to make sure that no race conditions appear on the Periscope Steering and the Image Analyzer Module as these two modules are access critical.

When the CVS is powered on, this module immediately starts the TCP/IP server and begins to listen on the designated port for incoming messages. It will only stop when the CVS is powered off. According to the content of the received messages, the module starts or stops the other modules, sets values of notifiers or transmits requested data to the ground station. Beside this communication, the FELT Controller Module permanently sends some status information to the ground station. For details about this information, see appendix B.4.
The TCP/IP server and the processing unit for the messages are both implemented in Timed Loops.

In addition to the ground station commands, the FELT Controller Module also reads the gyroscope data from one of its COM port. It decodes the received data strings and distributes the contained information to the other modules.

5.3.3 Image Analyzing Module

The Image Analyzer Module interacts with the camera connected to the first firewire port of the CVS. This communication follows the "IIDC 1394-based Digital Camera Specification Ver.1.31" [TA 04] of the "1394 Trade Association". The low level part of the module reads the images from the camera and controls the settings for shutter, brightness, and gain in accordance with the gathered image information. For this the algorithms from chapter 4.2 are used.

The information about the current captured image is stored in a notifier for access by other modules.

For better performance, the module can be set to operate in acquisition or in tracking mode. The mode defines which, and how, information is gathered and what size the area of interest has.

5.3.4 Periscope Steering Module

The Periscope Steering Module interacts with the motor controllers of the periscope via ports COM4 and COM5 of the CVS. Mainly, it receives motion commands (see appendix B.11) via a notifier from other modules and converts these into commands for the motor controllers of the periscope. For a precise control of the periscope, the module implements the control theory developed in section 4.3.

When implementing the controller for the periscope steering, it has been noticed that the driver for communicating with the motor controller was quite slow and only a sampling rate of 8 Hz could be reached. Another problem that occurred with this driver was that it could not be used together with the operating system of the CVS. So the communication protocol for the motor controllers had to be reimplemented. Due to this delay and the late delivery of the periscope, the implementation of the tracking controller could not be finished prior the delivery date of this thesis and therefore no tests with the final system could be done.

5.3.5 Acquisition Module

The task of the Acquisition Module is to reliably discover the beacon laser of the ground station. It is started by the FELT Controller Module during the acquisition phase. For the acquisition, the module runs the algorithm described in chapter 4.4.2.

During implementation it has been noticed that changing the area of interest of the camera can take up to 0.6s. This is because for changing the AOI, the camera has to be reinitialized with the new AOI as parameter. As the Acquisition Module starts the Tracking Module when it has discovered the beacon laser, and therefore implicitly changes the AOI of the camera, the locking of the camera on the beacon laser becomes a very important part.

5.3.6 Tracking Module

The task of the Tracking Module is to keep the discovered laser spot in the center of the image. This is done by reading the information about the current image from the Image Analyzer Module and adjusting the camera motion accordingly via the Periscope Steering Module. The Tracking Module is started by the Acquisition Module after the ground station beacon has been discovered.

If the spot should get lost despite the high frame rate, the Tracking Module has to detect this error and recover the spot. As there has not been enough time to develop a special recovery strategy and because it is very unlikely to lose the spot once it has been discovered, the module simply starts the Acquisition Module and stops.

Chapter 6

Experimental Verification

For evaluating the developed system several tests will be run. This chapter give details about the various test setups.

6.1 Laboratory Test Stand

For tests in the laboratory, the parts of the FELT were placed on an antenna drive system. As all cables (power supplies, ethernet, ...) of the system can be looped through the drive, the system can be freely rotated by the drive. The setup is shown in figure 6.1.

As the periscope was not available for tests during this stage, a substitute for it had to be assembled. Two common servo motors normally used for model helicopters and a small mirror were used to assemble the periscope replacement. The mirror was mounted in front of the camera and could be moved around its two axes using the servo motors. By moving the mirror, the field of view of the camera could be moved and the system could try to discover a laser that was projected on the ceiling above the test stand.

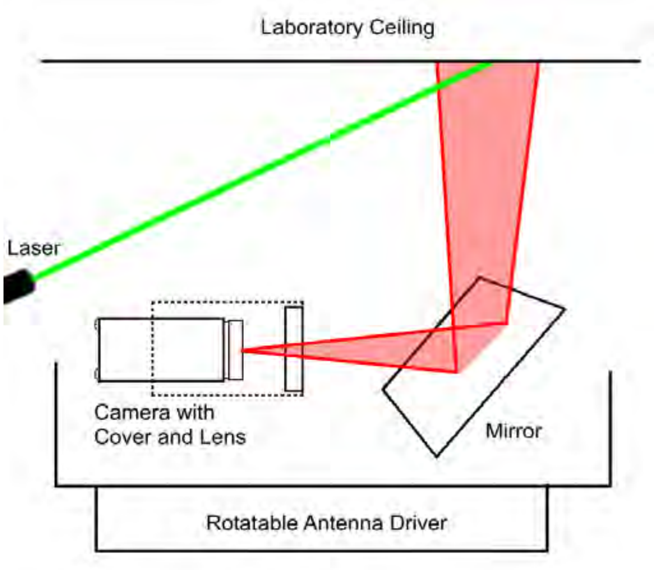


Figure 6.1: Laboratory test stand setup

As the coordinate system for the two servos was not the same as the horizontal coordinate system used internally in the software, a conversion between the two coordinate systems had to be calculated. The coordinate systems are shown in figure 6.2.

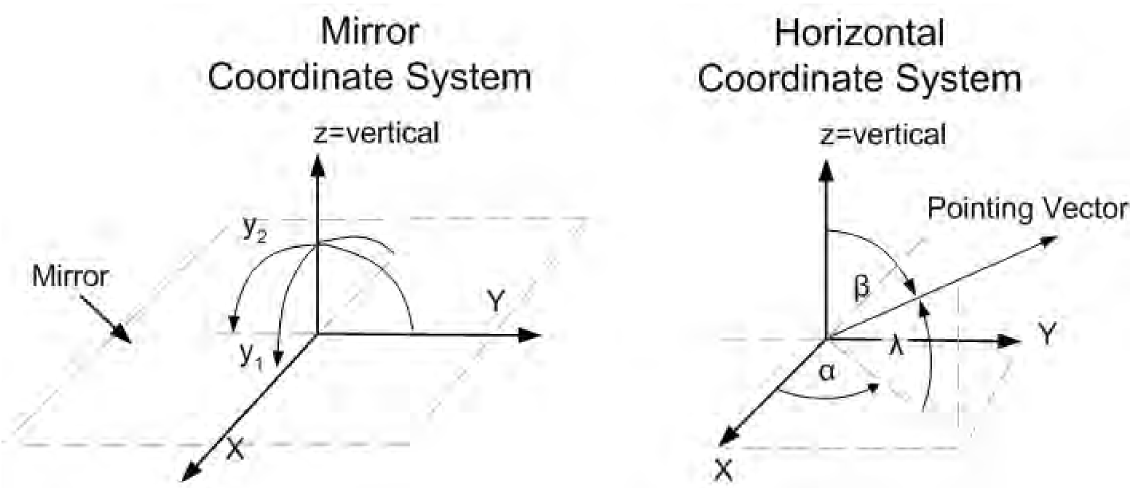


Figure 6.2: The mirror and the horizontal coordinate systems for the first test stand

6.1.1 Coordinate Transformation

The rotation matrix for rotations in the horizontal coordinate system are given as

$$D_{hor} = D_3(\alpha)D_2(\lambda)$$

where α is the Azimuth angle and λ the Elevation angle. D_2 denotes the rotation around the Y-axis, which is done first, and D_3 denotes the rotation around the Z-axis, which is done after D_2. This can be written as:

$$D_{Hor} = \begin{pmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos(\lambda) & 0 & sin(\lambda) \\ 0 & 1 & 0 \\ -sin(\lambda) & 0 & cos(\lambda) \end{pmatrix}$$

$$D_{Hor} = \begin{pmatrix} cos(\alpha)cos(\lambda) & -sin(\alpha) & cos(\alpha)sin(\lambda) \\ sin(\alpha)cos(\lambda) & cos(\alpha) & sin(\alpha)sin(\lambda) \\ -sin(\lambda) & 0 & cos(\lambda) \end{pmatrix}$$

The Cartesian vector is given by:

$$\overrightarrow{x_{Hor}} = D_{Hor}\overrightarrow{e_x} = D_{Hor} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} cos(\alpha)cos(\lambda) \\ sin(\alpha)cos(\lambda) \\ -sin(\lambda) \end{pmatrix}$$

With $\lambda = -(90° - \beta)$, the given equation can be rewritten for the nadir angle β.

$$\overrightarrow{x_{Hor}} = \begin{pmatrix} cos(\alpha)sin(\beta) \\ sin(\alpha)sin(\beta) \\ cos(\beta) \end{pmatrix}$$

Similar equations are valid for the mirror coordinate system

$$D_{Mirror} = D_2(y_1)D_1(y_2)$$

where y_1 and y_2 are the angles of the mirror. The coordinates of the mirror are calculated from the coordinates of the motors (m_1, m_2) by

$$y_1 = 2m_1$$
$$y_2 = 2m_2$$

The rotation matrix of the mirrors is calculated by

$$D_{Mirror} = \begin{pmatrix} cos(\gamma_1) & 0 & sin(\gamma_1) \\ 0 & 1 & 0 \\ -sin(\gamma_1) & 0 & cos(\gamma_1) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\gamma_2) & -sin(\gamma_2) \\ 0 & sin(\gamma_2) & cos(\gamma_2) \end{pmatrix}$$
$$D_{Mirror} = \begin{pmatrix} cos(\gamma_1) & sin(\gamma_1)sin(\gamma_2) & sin(\gamma_1)cos(\gamma_2) \\ 0 & cos(\gamma_2) & -sin(\gamma_2) \\ -sin(\gamma_1) & cos(\gamma_1)sin(\gamma_2) & cos(\gamma_1)cos(\gamma_2) \end{pmatrix}$$

The Cartesian vector is given by

$$\overrightarrow{x_{Mirror}} = D_{Mirror}\overrightarrow{e_z} = D_{Hor} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} sin(\gamma_1)cos(\gamma_2) \\ -sin(\gamma_2) \\ cos(\gamma_1)cos(\gamma_2) \end{pmatrix}$$

The Cartesian vectors are set equal to calculate the mirror angles from the horizontal coordinates:

$$\overrightarrow{x_{Mirror}} = \overrightarrow{x_{Hor}}$$

and the mirror coordinates result to

$$\gamma_2 = asin\left(-sin(\alpha)sin(\beta)\right)$$
$$\gamma_1 = asin\left(\frac{cos(\alpha)cos(\beta)}{cos(\gamma_2)}\right)$$

Therefore the motor axes are calculated by

$$\gamma_2 = \frac{1}{2}asin\left(-sin(\alpha)sin(\beta)\right)$$
$$\gamma_1 = \frac{1}{2}asin\left(\frac{cos(\alpha)cos(\beta)}{cos(\gamma_2)}\right)$$

With this formula it is possible to scan a given circle with the camera.

6.1.2 Test Results

Since the test system did not use the periscope, the control theory for the system could not be tested. It was also not possible to change the size of the AoI between acquisition and tracking phase, since the system was not able to lock the field of view on the detected spot. This had to be done on the second test stand. The last system part that could not be tested was the GPS systems, so a static nadir angle had to be provided for the test system.

Unfortunately the servos tended to overshoot and were not accurate enough for the demands of optical free-space communication. So the accuracy of the system could not be measured, but its functionality could be verified.

Even if the servos were not accurate, the system was able to acquire and track the spot with the antenna drive rotating at speeds of up to 30 deg/s. The system was also controllable via RF link and transmitted its status information to the ground station.

6.2 Long Optical System Range Test

For verifying the calibration algorithm of the camera and the assumptions about the beacon laser a second test has been done at the Peissenberg. The camera was located again at the top of the Peissenberg. The beacon laser was located near Gilching which is approximately 40 km north of the Peissenberg. For this test, the same hardware (Lasers, Lenses, Filters, ...) has been used that also will be used during Trial 2.

The result of this test was that the calibration algorithm works as expected and that the visibility of the beacon laser is better than expected. The results of the analysis of the recorded images is given in table 6.1 and Figure 6.3 shows a series of the recorded images. When these images were take, clouds were moving along the sky and creating changing lighting effects. Unfortunately, again there were no reflections visible in the field of view of the camera, so that there is still no knowledge from measurements about the characteristics of reflections.

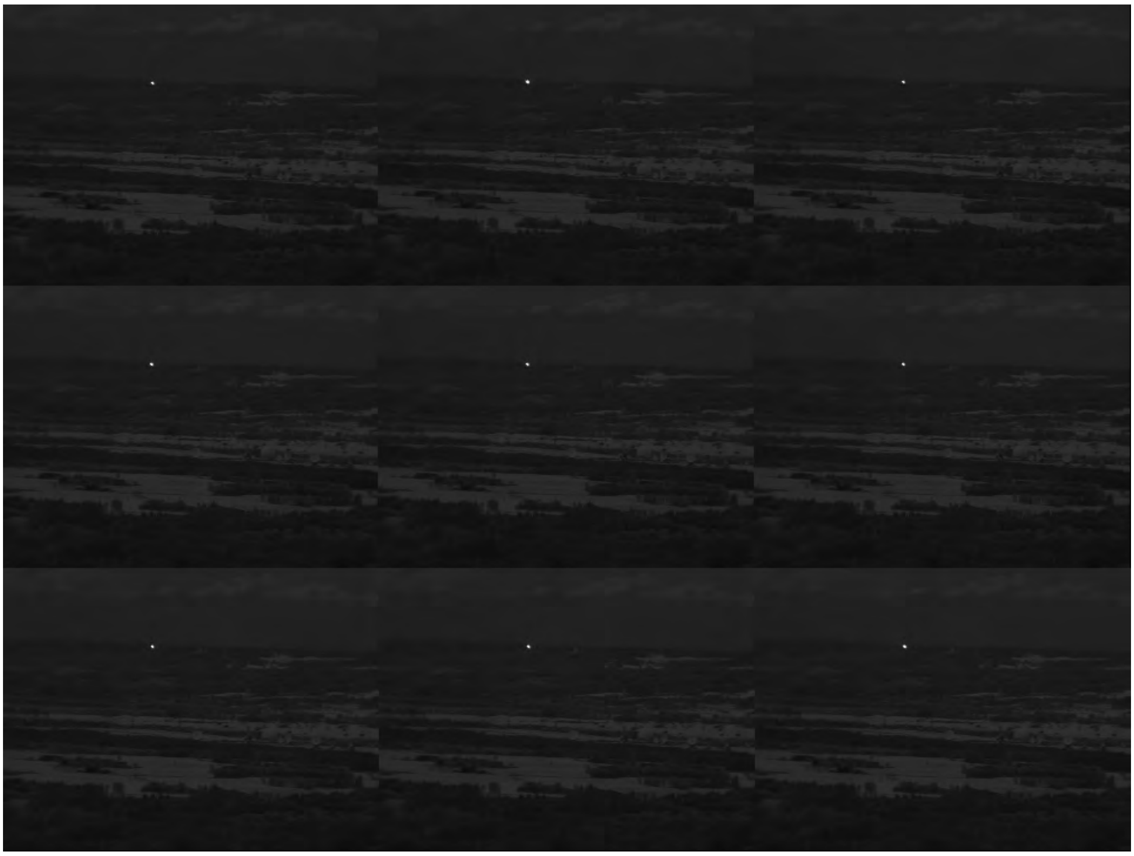


Figure 6.3: Series of images recorded with the calibration algorithm

Maximum Shutter Value	70
Min. Shutter Value for Spot Saturation	45
Min. Shutter Value for Spot Detection	15
Min. Spot Diameter	2
Max. Spot Diameter	6
Min. Spot Size (incl. safety margin)	2
Max. Spot Size (incl. safety margin)	50

Table 6.1: Results from the second Peissenberg experiment

6.3 Planned Tests

Before the system will be tested within Trial 2 of the Capanina projects several other tests are planned

6.3.1 Short Range Field Test

For this test, the system will be placed on the roof of the building of the Institute for Communication and Navigation at the DLR. The mobile optical ground station will be placed in a van. The van will drive on the nearby roads of the building and the system will have to acquire and track the ground station. This test system is also used as a development platform, so a WLAN connection has been added to the FELT for accessing the CVS from the development PC and debugging the system.

6.3.2 Long Range Field Test

This test is mainly for verifying the assumptions about the optical system. The FELT will be placed on the top of the Hoernle, a small mountain approximately 100 km south of Munich. The ground station will be placed at the DLR outpost in Weilheim at the Starnberger See. The distance between the two systems will be about 40 km.

6.3.3 Airborne Field Test

This will be the last test before Trial 2. It is planned to mount the FELT on hot-air balloon or a Zeppelin that flies over the DLR compound, where the ground station will be located. All parts of the system will be tested during this trial.

Chapter 7

Conclusion

In this thesis, several problems for the development of a pointing, acquisition and tracking system for the use with optical free-space communication have been named and solutions have been provided. Also some of the problems were specific to the requirements of the Trial 2 of the Capanina project (e.g. periscope control theory), most of the problems were general problems that will occur within most other PAT systems (e.g. spot detection, tracking, development of a controller). The description of the developed solutions for these problems are given in a general way and paired with the fundamental knowledge that led to these solutions. This helps others to adapt the given solutions to similar problems. For example, the calculations of the recorded images for Trial 2 are done with the values given by the test setup, but since all formulae and the theoretic background are given, it should not be a problem to adapt these calculations for other setups. Another example is the developed algorithm for extracting the beacon laser from the recorded images. This algorithm is fast and easy to implement, so it can be used for any application that involves visual tracking of a bright object.

In this thesis it has also been shown, that it is possible to use of the shelf hardware for implementing a PAT system. Only for the steering of the field of view of the camera, some customized hardware had to be used, because the available hardware could not produce results accurate enough for the requirements of optical free-space communication.

There is one challenge that has not been solved by this thesis. Since there is little knowledge about the actual shape and intensity of the beacon laser at the tracking camera, only a few criteria for decision making about the spots in the images are given and it is not always possible for the system to decide if a bright spot in the image is the beacon laser or just a reflection. In these cases, interaction with the user at the ground station is needed to make the decision. A solution for this problem would be to use a modulated beacon.

Future work can be done on developing completely autonomous systems that are able to setup an optical free-space communication link without the help of humans.

Chapter 8

Acknowledgements

Fist, I have to thank everyone of the optical communications group at the DLR. Thank you for your patience in explaining to me the basics of optics and electro-technics. I have learned so much that the time with you was nearly like a second study.
Second, I have to thank Silke Mattle and Brandon Wilkerson for proofreading this thesis and Moritz Hammer for always answering my questions about just everything, specially LaTeX.
Third, I want to thank Prof. Dr.-Ing. Uwe Stilla for the information on image analysis he provided and his willingness to give further assistance.
Last but not least I want to thank Markus Knapek for all the interesting talks and brainstorms we had about pointing, acquisition and tracking and for all his help with this thesis.

Appendix A

Measurements

A.1 MIPAS-B2 Experiment

The following diagrams illustrate the data gathered during the measurements of the MIPAS-B2 Experiment.

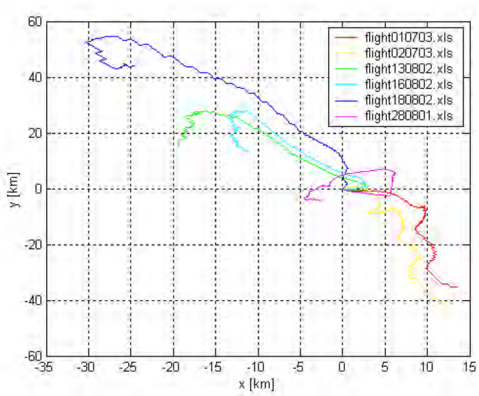


Figure A.1: Movement of the balloon during various measurements

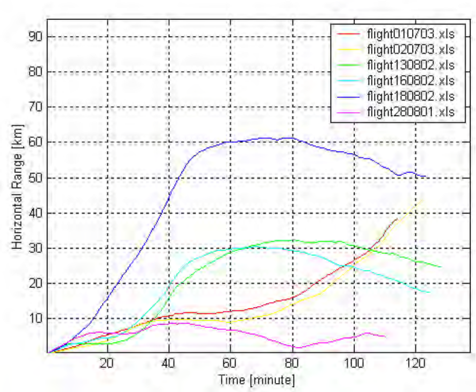


Figure A.2: Distance between balloon and launch site during various measurements

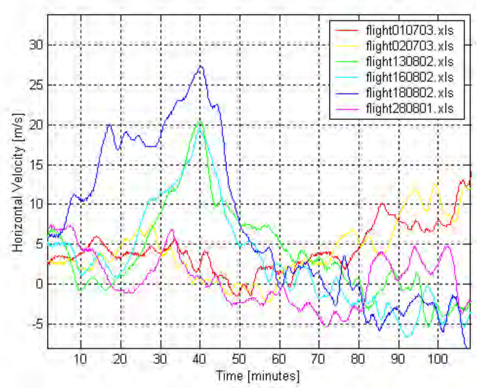


Figure A.3: Horizontal velocity of the balloon during various measurements

A.2 Camera Responsiveness Measurements

The following diagrams illustrate the data gathered during the responsiveness measurements with the Basler 602f camera.

The first measurements were performed for several shutter times of the camera at complete darkness. The camera has been used at 10 and 8 bit resolution. Table A.1 gives the measured dark noise offset (DNO) for the various shutter times. Actually the 8 bit values should be one fourth of the 10 bit values, but this does not hold for these measurements. A change in the ratio of the 10 to 8 bit values can only be explained by setting the values of brightness and gain, but these values have been double checked to be correct. Since this behavior only introduces a small error, no further research has been done on this topic.

Shutter Time	10 bit	8 bit
1	49	1
401	50	3
801	52	5
1201	54	6
1601	54	7
2001	55	8
2401	56	9
2801	57	10
3201	58	11
3601	59	12
4001	60	14

Table A.1: Dark noise measurement

The second measurements were performed for several shutter times of the camera. The illumination source was a halogen lamp. A filter at 980nm with 10nm bandwidth was used in front of the camera's sensor. The camera has 10 bit resolution. Values in the diagram stay below 1024, since the DNO was subtracted.

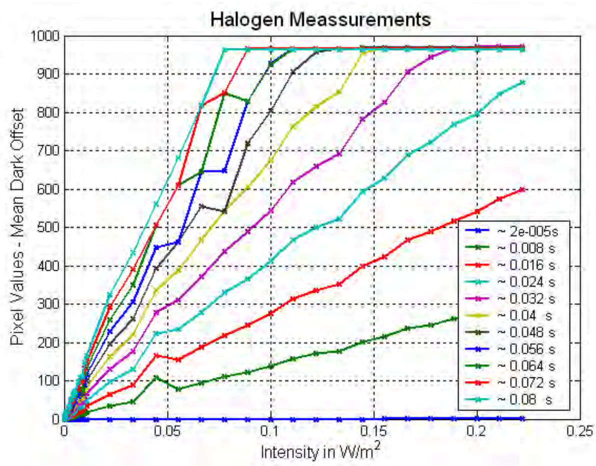


Figure A.4: Measured responsivity curves of the camera

Appendix B

Specifications

B.1 Basler 602f Camera

The following table shows the specifications of the Basler 602f camera used for the image acquisition on the FELT.

Sensor Size (H x V Pixels)	656 x 491 (mono)
Sensor Type	Micron MT9V403 - 1/2 inch, Progressiv Scan CMOS, Global Shutter
Pixel Size	9,9 μm (H)x 9,9 μm (V)
Max. Frame Rate at Full Resolution	100 frames/s in 8 bit output modes, 50 frames/s in 16 bit output modes (more with AOI)
Video Output Format	Mono 8 (8 bits/pixel), Mono 16 (16 bits/pixel - 10 bits are effective), Pseudeo YUV 4:2:2 (16 bits/pixel avg.)
Gain and Brightness	Programmable via IEEE 1394 bus
Exposure Time Control	Programmable via IEEE 1394 bus
Synchronization	External via External Trigger signal
Power Requirements	+8.0 to +36.0 VDC (+12 VDC nominal), < 1% ripple, 1.7 W max @ 12 VDC (typical), supplied via IEEE 1394 cable
I/O Electrical Characteristics	Inputs: opto-isolated, 5 VDC nominal, 10 mA nominal, Outputs: opto-isolated, 2 to 35 VDC maximum forward voltage, 100 mA max collector current
Max. Cable Length	1394: 4.5 m, I/O: 10 m
Lens Adapter	C-mount
Housing Size (L x W x H)	Without lens adapter: 59 mm x 44 mm x 29 mm, With C-mount adapter: 67.3 mm x 44 x 29
Weight	< 100g (typical)
Conformity	CE, FCC

Table B.1: Specifications of the Basler 602f camera

The camera manual gives the following table for the amplification effect of the gain setting.

Decimal	Hex	dB	Amplification Factor
0	0x00	0.0	x 1.0
28	0x1C	2.5	x 1.3
43	0x2A	3.5	x 1.5
85	0x55	6.0	x 2.0
128	0x7F	8.0	x 2.5
170	0xAA	9.5	x 3.0
213	0xD4	10.9	x 3.5
255	0xFF	12.0	x 4.0

Table B.2: Gain settings and their effect

Figure B.1 gives the quantum efficency of the camera, which determines the responsivity of the camera for specific wavelength.

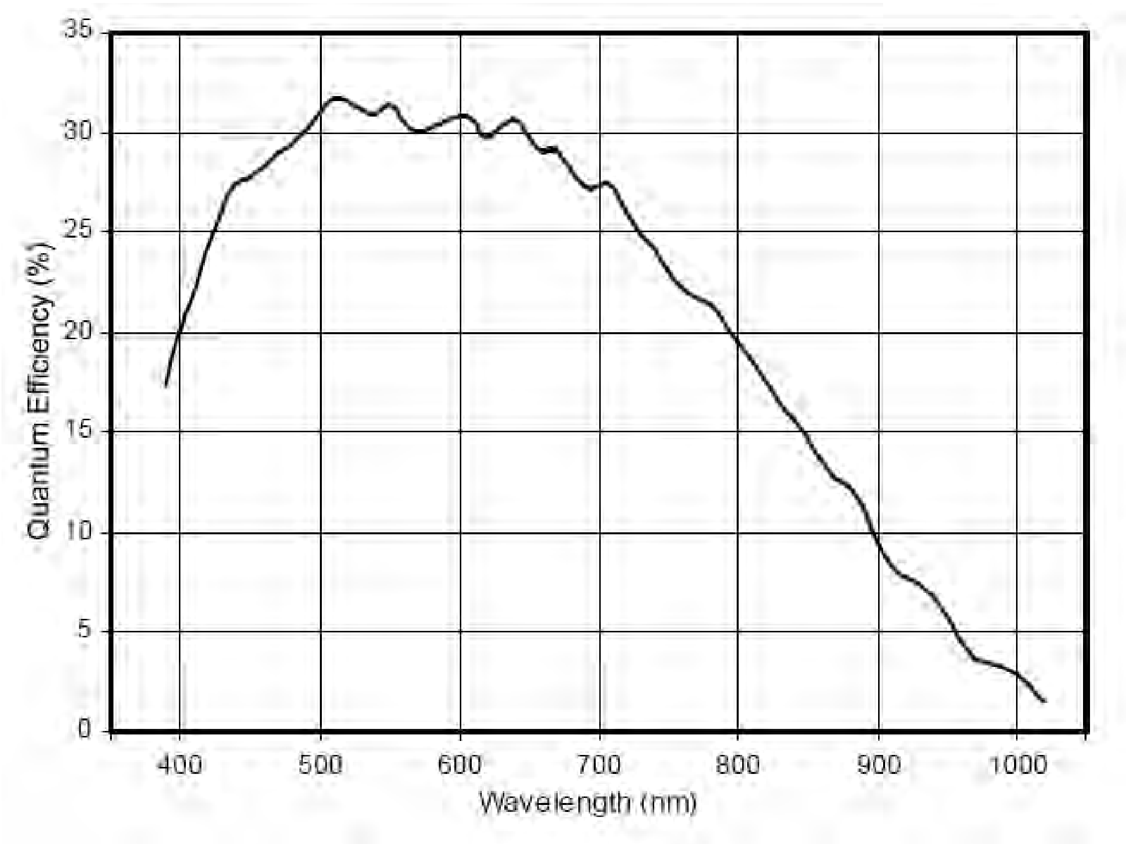


Figure B.1: Quantum Efficiency of the Basler 602f camera

B.2 Periscope

The following tables provide information about the resolution of the encoders used for measuring the positions of the two axes and about the conversion between encoder counts and angles.

Weight	$\approx$ 3.2 kg
Optical Path Length	$\approx$ 290 mm
Optical Diameter	$\approx$ 53 mm
Mirror Evenness @ 1550 nm	$< \lambda/2$
Mirror Reflectivity @ 800 - 1600 nm	$>$ 98 %
Wavefront Deformation at Emission Window @ 1550 nm	$< 1\ \lambda$
Angular Range Azimuth	∞
Angular Range Nadir	$\pm$ 170 deg
Maximum Azimuth Velocity	$\approx$ 120 deg/s
Maximum Nadir Velocity	$\approx$ 240 deg/s

Table B.3: Periscop Specification

Lines per Revolution	18,000
Interpolation Factor	10
Quadrature	4
Counts per Revolution	720,000
Degrees per Count	0.000500
Milliradian per Count	0.008727
Microradian per Count	8.726646

Table B.4: Encoder resolution

Degree	Milliradian	Counts
0.00050	0.009	1.00
0.00057	0.010	1.15
0.00100	0.017	2.00
0.00573	0.100	11.46
0.01000	0.175	20.00
0.05730	1.000	114.59
0.10000	1.745	200.00
0.57296	10.000	1145.92
1.00000	17.453	2000.00

Table B.5: Conversion between axes angles and encoder counts

Counts	Degrees	Az. Motor Rev.	El. Motor Rev.	rad	mrad	μrad	$\frac{m}{22km}$
1.0	0.0005	0.00035	0.00017	0.00001	0.00873	8.7	0.192
5.0	0.0025	0.00174	0.00087	0.00004	0.04363	43.6	0.960
5.2	0.0026	0.00181	0.00090	0.00005	0.04538	45.4	0.998
10.0	0.0050	0.00347	0.00174	0.00009	0.08727	87.3	1.920
20.0	0.0100	0.00694	0.00347	0.00018	0.17453	174.5	3.840
100.0	0.050	0.03472	0.017361	0.00087	0.87267	873	19.2
114.6	0.058	0.03979	0.019896	0.00100	1.00007	1000	22.0
1000.0	0.500	0.34722	0.173611	0.00873	8.72665	8727	192.0
2000.0	1.000	0.69444	0.347222	0.01745	17.45329	17453	384
10000.0	5.000	3.47222	1.736111	0.08727	87.26646	87266	1924
11459.0	5.730	3.97882	1.989410	0.10000	99.99864	99999	2207
100000.0	50.000	34.72222	17.361111	0.87267	872.66463	872665	26218
114591.6	57.296	39.78875	19.894375	1.00000	1000.00036	1000000	34263
180000	90	62.5	31.25	1.57080	1570.796	1570796	N/A
360000	180	125.0	62.50	3.14159	3141.593	3141593	N/A
720000	360	250.0	125.00	6.28319	6283.185	6283185	N/A

Table B.6: Conversion between counts, axes angles and motor revolutions

B.3 Filter

The filter has a bandwidth from 800 to 850 nm. The transmission for the filter is given in figure B.2.

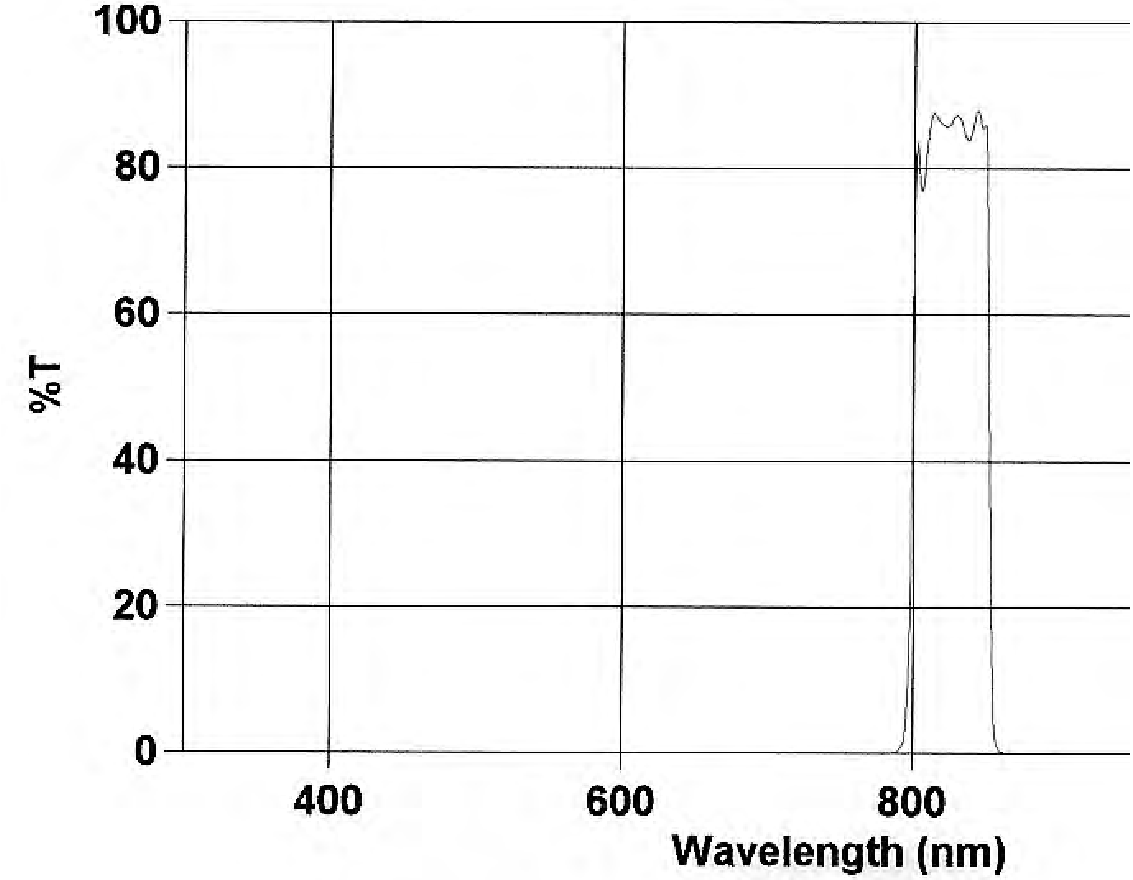


Figure B.2: Transmission curve of the 800-850 nm filter

B.4 TMTC Commands

The TMTC commands are used for the communication between the FELT terminal and the ground station. The protocol is based on binary packages to reduce the volume of transmitted data. There is no further compression used in order to keep the procession time as low as possible. The data block size is a word (8 Bit), so the available data fields can have sizes of 8, 16, 24, etc. bits.

B.4.1 Message Structure

The STX (0x02) byte marks the beginning of a message and the ETX (0x03) marks the end. The empty message 0x02 0x03 is used as ping-message which is periodically sent to signal the receiver that the connection is still working. If one of these special bytes occur within a message, they are escaped with the DLE byte (0x10). After the STX word follows the TypeID. TypeID is a word indicating how the content of the message has to be read. The TypeID and the content of the messages is described in the following section

B.4.2 Message Content

The available messages are Commands (0x11) for sending commands from ground station to HAP, Image Info (0x12) for transmitting information about the current image and camera settings, Rotation Info (0x13) for transmitting information about the rotation of the balloon and Status Info (0x14) for transmitting information about the current Status of the HAP.

Command Name	ID	Parameter	Values	Description
Idle	0x00			Stops all running loops
Start Acquisition	0x11			Starts the Acquisition Loop
Start Tracking	0x12			Starts the Tracking Loop
Set UseGyro	0x13	usegyro	0 = false/no 1 = true/yes	Use Gyro for Acquisition?
Set Nadir	0x14	nadirangle	0 - 90 deg	Use this nadir angle for acquisition
Set Brightness	0x15	brightness	0 - 1023	Adjust the brightness setting of the camera
Set Gain	0x16	gain	0 - 255	Adjust the gain setting of the camera

Table B.7: Commands (0x11)

Field Number	Field Name	Allowed Values	Note
0	Number Of Spots	0 - 255	255 means >= 255
1	Histogram Min	0 - 255	
2	Histogram Max	0 - 255	
3	Histogram Mean	0 - 255	
4	Histogram Background	0 - 255	Approx. position of the background light values
5	Camera Brightness MSB	0 - 1024	
6	Camera Brightness LSB		
7	Camera Shutter MSB	1 - 4095	
8	Camera Shutter LSB		
9	Camera Gain	0 - 255	
10	Frames per Second	0 - 255	255 means >= 255
11	AOI Width MSB	1 - 656	
12	AOI Width LSB		
13	AOI Height MSB	1 - 491	
14	AOI Height LSB		
15	Best Spot Candidate X Offset MSB	-328 - 328	Distance of the best suitable spot from the image centre
16	Best Spot Candidate X Offset LSB		
17	Best Spot Candidate Y Offset MSB	-246 - 246	
18	Best Spot Candidate Y Offset LSB		

Table B.8: Image Info (0x12)

Field Number	Field Name	Allowed Values	Note
0	Nadir Angle	0 90 deg	
1	Azimuth Angle MSB	0 360 deg	
2	Azimuth Angle LSB		
3	Azimuth Speed MSB	-127 127 deg/s	255 means >= 255, -255 <= -255, 0 means 0 (during acquisition) or unknown (in any other state)
4	Azimuth Speed LSB		
5	Gyro Speed MSB	-127 127 deg/s	127 means >= 127, -127 <= -127
6	Gyro Speed LSB		

Table B.9: Rotation Info (0x13)

Status ID	Status Name	Description
0x00	Idling/Ready	The system is doing nothing and waiting for commands
0x11	Acquisition running	
0x12	Tracking running	
0xE1	Place for error messages	
0xE...	...	

Table B.10: Status Info (0x14)

B.5 Internal commands of FELT software

For the notifiers used within the FELT software one general rule applies. A value of -1 indicates that this loop has been stopped, 0 causes the loop to stop and 1 causes the loop to start.

For most of the internal communication within the FELT software the data of the TMTC link is used. One exception are the motion commands for the periscope as these are not contained within the TMTC commands.

Command Code	Command Name	Argument 1	Argument 2
-1	Stop		
0	Rotate	Nadir angle in deg	Azimuth speed in deg/s
1	MoveBy	X value	Y value
2	MoveTo	X value	Y value
0xE...	...		

Table B.11: Motion Commands

B.6 Risk Assesment

No.	Project Risk	Level (L,M,H)	Impact (1-10)	Contingency Plans
1	TMTC Link Failure	L	10	The system can not be powered up and the software can not be started. If the system is permanently powered on, then it would be possible to tell the software to start automatically with the acquisition. But communication between FELT and ground station would still not be possible.
2	System error after power on	L	10	In rare cases, the CVS starts in a faulty state where the camera can not be operated properly. This state can be discovered by the FELT software and can be solved by manually resetting the CVS. To solve this problem a working TMTC link is mandatory.
3	Network Failure	L	10	If a problem with the LAN on FELT occurs, the system will not be able to operate. For example if the HUB breaks, the CVS can no longer communicate with the periscope or the TMTC link.
4	Gyroscope Failure	L	1	The gyroscope is only used during acquisition. Failure can be detected by the software and an alternative acquisition algorithm is available. The acquisition will take longer than with support of the gyroscope.
5	Periscope Failure	L	10	If the periscope does not operate properly it might not be possible to conduct the acquisition or tracking of the ground station. The impact depends on the severity of the error.
6	Camera Failure	L	10	If the camera does not operate properly it might not be possible to conduct the acquisition or tracking of the ground station. The impact depends on the severity of the error.
7	Wrong Parameters for Algorithms	L	5	Some of the algorithms use parameters which have been predetermined from results of several tests. If these parameters should be wrong it is possible to update them via TMTC link. This might involve some guessing, but should be possible from information transmitted by the system to the ground station.
8	GPS system failure	L	7	The GPS data is only used during acquisition. If the GPS system fails it is possible to tell the system to conduct a circular scan for the ground station without the GPS data. In this case the acquisition will take longer. It might be possible that the acquisition will take too long.

9	Reflections and Background Light	L	10	It might be possible that reflections might get misinterpreted as being the beacon laser. If that is the case it is possible to tell the system it is tracking the wrong spot via TMTC link. If the link budget has been calculated wrongly it might be possible that the beacon laser is not visible to the camera. In this case acquisition and tracking of the ground station is not possible. Measurements have shown that the calculation of the link budget should be correct.
10	Camera Calibration Error	L	8	If the automatic camera calibration does not produce usable images it is possible to set the camera attributes via TMTC link. The error should be detectable at the ground station from the transmitted information.
11	Periscope Controller Failure	L	10	If the controller for the periscope fails, e.g. the periscope oscillates around the elevation axis, acquisition and tracking might not be possible. The impact depends on the type of failure.
12	Malfunctioning of the communication module	L	7	If the communication module does not operate properly, the TMTC link might not work. The impact depends on the type of failure and the state of the system in which the malfunction occurs.
13	Malfunctioning of the periscope steering module	L	8	If an error in the periscope steering module occurs it might break the periscope, although this is very unlikely. Most errors will introduce some inaccuracies in the pointing. This can be compensated by sending offset values via the TMTC link.
14	Malfunctioning of the image analyzer module	L	7	Malfunctions of the image analyzer module will mainly be caused by failures of other parts of the system. If this module fails it will not be possible to acquire or track the ground station. Resetting the CVS might solve an existing problem.
15	Malfunctioning of the acquisition and tracking modules	L	10	These modules have a low complexity but rely heavily on other parts of the system. A failure of these modules will mainly be caused by failure of other parts. Acquisition and tracking of the ground station will not be possible.
16	CVS Failure	L	10	If the CVS breaks, the whole system will not be able to operate.

Table B.12: Risk Assesment

Appendix C

Source Code

C.1 Blob Detection Header File

```
//File: PATAnalysis.h
#ifdef PAT_EXPORTS
#  define EXPORT __declspec(dllexport)
#  pragma message("automatic export to PATAnalysis.lib")
#else
#  define EXPORT __declspec(dllimport)
#  pragma message("automatic link to PATAnalysis.lib")
#  pragma comment(lib, "PATAnalysis.lib")
#endif

void EXPORT acquisitionAnalysis(unsigned long* LVImagePtr,
     unsigned long* Blobs, unsigned char* maxBlobs, unsigned long*
     equivTable, unsigned char* numOfBlobs, unsigned long* bestSpotID,
     unsigned char* threshold, unsigned char* minBlobSize, unsigned
     char* maxBlobSize);

unsigned char getLabel(unsigned char* image, unsigned long height,
     unsigned long width, long pos, long yOffset, long xOffset);

void equalize(unsigned long* labels, unsigned long x, unsigned long y);

void reduceLabels(unsigned long* equivTable, unsigned long x,
                  unsigned long y);

void getLabelAndBorder(unsigned char* image, unsigned long pos,
     unsigned char* label, unsigned char* currentLabel, unsigned
     long* equivTable, unsigned char threshold, char* isOnBorder);
```

C.2 Blob Detection Code

```
//File: PATAnalysis.ppt
#define PAT_EXPORTS
#include "PATAnalysis.h"
#include <stack>
#include <stdio.h>
#include <limits.h>
```

```
#include <iostream>
using namespace std;

//these values are predefines because the image size
//is fixed during acquisition
#define   LVLINEWIDTH (664)
#define   HEIGHT (491)
#define   WIDTH (565)
/*Blob content:
Size, Circumference, Compactness, Position Y, Position X,
Sum of Y, Sum of X */
#define   BLOBSIZE (7)


/*
 This function takes an 8-Bit image and detects
 the bright spots in it. (Spots above threshold)

 The image passed to the function needs a LVBorderSize of 3
 or more for preventing errors

 Note: This code is not optimized, it is just meant as proof
 of concept to see if the algorithm can compete with
 the implementation written in LabView.
*/

void EXPORT acquisitionAnalysis(unsigned long* LVImagePtr,
    unsigned long* Blobs, unsigned char* maxBlobs, unsigned long*
    equivTable, unsigned char* numOfBlobs, unsigned long* bestSpotID,
    unsigned char* threshold, unsigned char* minBlobSize, unsigned
    char* maxBlobSize)
{
  //setup of label environment
  unsigned long i = 0;
  unsigned long j = 0;
  unsigned long x = 0;
  unsigned long y = 0;
  unsigned char label = 0;
  unsigned char currentLabel = 1;
  unsigned char maxBlobsReached = 0;
  unsigned char* image = (unsigned char*)*LVImagePtr;


  char isOnBorder = 0;

  //reset equivTable
  for (i=*maxBlobs;i--;)
  {
    equivTable[i] = i;
    for(j=BLOBSIZE;j--;)
    {
      Blobs[i*j] = 0;
    }
  }
```

```
/*1st run through complete image*/
for (y=0;y<HEIGHT;y++)
{
  for (x=0;x< WIDTH;x++)
  {
    //j is now used as position marker
    j = y * LVLINEWIDTH + x;

    if(maxBlobsReached == 0)
    {
      if(image[y * LVLINEWIDTH + x] >= *threshold)
      {
        //this pixel is interesting, label it and add it to a blob
        //get the label of pixel
        isOnBorder = 0;
        getLabelAndBorder(image, j, &label, &currentLabel,
                          equivTable, *threshold, &isOnBorder);
        //check if we have maxBlobs detected
        if(label <= *maxBlobs)
        {
          //add this pixel to the blob marked by label
          //Size
          Blobs[label*BLOBSIZE]++;
          //Circumference
          if(isOnBorder)
          {
            Blobs[label*BLOBSIZE+1]++;
          }
          //Compactness, Position Y, Position X
          //will be calculated later
          //Sum of Y
          Blobs[label*BLOBSIZE+5] += y;
          //Sum of X
          Blobs[label*BLOBSIZE+6] += x;
        }
        else
        {
          maxBlobsReached = 1;
        }
      }
      else
      {//mark pixel as not interesting
        image[y * LVLINEWIDTH + x] = 0;
      }
    }
  }
}

/****************/
/*End of 1st run*/
/****************/

if(maxBlobsReached)
```

```
{
  //stop if we have too many Blobs
  *bestSpotID = 0;
  *numOfBlobs = 255;
  return;
}

for(i=currentLabel-1; i>0; i--)
{
  reduceLabels(equivTable,equivTable[i],i);
}

//join blobs according to equivTable
for(i=currentLabel-1; i>0; i--)
{
  if(equivTable[i] != i)
  {//if this blob has a neighbor blob, join them
    //Size
    Blobs[equivTable[i]*BLOBSIZE] += Blobs[i*BLOBSIZE];
    //Circumference
    Blobs[equivTable[i]*BLOBSIZE+1] += Blobs[i*BLOBSIZE+1];
    //Compactness, Position Y, Position X will be calculated later
    //Sum of Y
    Blobs[equivTable[i]*BLOBSIZE+5] += Blobs[i*BLOBSIZE+5];
    //Sum of X
    Blobs[equivTable[i]*BLOBSIZE+6] += Blobs[i*BLOBSIZE+6];
    //delete Entry from equivTable
    equivTable[i] = 0;
  }
  else if(Blobs[equivTable[i]*BLOBSIZE] < *minBlobSize ||
          Blobs[equivTable[i]*BLOBSIZE] > *maxBlobSize)
  {//if it is not connected to any other blob
   // checkt its size and "remove" blobs with wrong size
    equivTable[i] = 0;
  }
}

/****************/
/*end of 2nd run*/
/****************/

//all blobs with a wrong size are already removed from the memory
//find most compact blob
//and count blobs
*bestSpotID = 1;
*numOfBlobs = 0;
for(i=1; i<currentLabel; i++)
{
  if(equivTable[i])
  {
    *numOfBlobs = *numOfBlobs+1;
    //Compactness
    Blobs[i*BLOBSIZE+2] = Blobs[i*BLOBSIZE+1]*Blobs[i*BLOBSIZE+1] /
                          Blobs[i*BLOBSIZE];
```

```
      if(Blobs[i*BLOBSIZE+2]<Blobs[*bestSpotID*BLOBSIZE+2])
      {
        *bestSpotID = i;
      }
      //PosX
      Blobs[i*BLOBSIZE+3] = Blobs[i*BLOBSIZE+5]/Blobs[i*BLOBSIZE];
      //PosY
      Blobs[i*BLOBSIZE+4] = Blobs[i*BLOBSIZE+6]/Blobs[i*BLOBSIZE];
    }
  }
  for(i=currentLabel; i<*maxBlobs; i++)
  {//remove unused blobs
    equivTable[i] = 0;
  }
}

void getLabelAndBorder(unsigned char* image, unsigned long pos,
                       unsigned char* label, unsigned char* currentLabel,
                       unsigned long* equivTable, unsigned char threshold,
                       char* isOnBorder)
{
  //array for collecting neighbors
  unsigned long nbs[12] = {0};
  //counter variable
  unsigned long i = 0;
  //index of smallest neighboring label
  unsigned long j = 0;

  nbs[0] = image[pos - 1326];
  nbs[1] = image[pos - 1327];
  nbs[2] = image[pos - 1328];
  nbs[3] = image[pos - 1329];
  nbs[4] = image[pos - 1330];
  nbs[5] = image[pos - 662];
  nbs[6] = image[pos - 663];
  nbs[7] = image[pos - 664];
  nbs[8] = image[pos - 665];
  nbs[9] = image[pos - 666];
  nbs[10] = image[pos - 2];
  nbs[11] = image[pos - 1];

  for(i=0;i<12;i++)
  {//find smallest neighboring label
    if(nbs[i]>0 && nbs[i] < 256)
    {
      if(nbs[j] == 0)
      {
        j=i;
      }
      else if(nbs[j]>0 && nbs[i]<nbs[j])
      {
        j=i;
      }
    }
```

```
    //check the direct neighbors (not diagonal)
    if(nbs[7] == 0 || nbs[11] == 0)
    {
      *isOnBorder = 1;
    }
  }
  //nbs[j] is now the smalles neighboring label
  if(nbs[j]>0)
  {
    //mark the pixel with the label
    *label = nbs[j];
    //image[pos] = *label;

    //all neighbors are equivalent to this pixel
    //and therefore should have the label stored in nbs[j]
    //add this fact to equivTable
    for(i=0;i<12;i++)
    {//it can't be smaller, otherwise nbs[i] would be the smallest label
      if(nbs[i]>nbs[j])
      {
        equalize(equivTable,nbs[j],nbs[i]);
        //nbs[i] = nbs[j];
      }
    }
  }
  else
  {//pixel has no labelled neighbor, assign new label
    //prevent overflow
    if(*currentLabel < 255)
    {
      *label = *currentLabel;
      *currentLabel = *label + 1;
    }
  }
  //mark pixel with current label
  image[pos] = *label;

  if(!*isOnBorder)
  {//if the pixel has a direct neighbor to the left and above it,
    //check the neighbors to the right and below it
    //if they will become a neighbor when the algorithm proceeds
    nbs[0] = image[pos + 1];
    nbs[1] = image[pos + LVLINEWIDTH];
    if(nbs[0] < threshold || nbs[1] < threshold)
    {
      *isOnBorder = 1;
    }
  }
}

unsigned char getLabel(unsigned char* image, long pos,
long yOffset, long xOffset)
{
  long x;
```

```
  long y;

  x = (pos%LVLINEWIDTH) + xOffset;
  y = (pos/LVLINEWIDTH) + yOffset;

  return image[y * LVLINEWIDTH + x];
}

/*makes entries to equivTable and propagates the added equivalences*/
void equalize(unsigned long* equivTable, unsigned long x, unsigned long y)
{
  if(x > y)
  {//maintain order, therefore change x and y
    equalize(equivTable,y,x);
    return;
  }
  if((x==y) || (equivTable[y] == x))
  {//if both are already equivalent, stop here
    return;
  }
  if(equivTable[y] == y)
  {//if there is no equivalence for y, (i.e. y=y), then y=x
    equivTable[y] = x;
  }
  else
  {//propagate the equivalence
    equalize(equivTable,equivTable[y],x);
    equivTable[y] = x;
  }
}

/*removes unneccessary labels*/
void reduceLabels(unsigned long* equivTable, unsigned long x,
                  unsigned long y)
{
  if(equivTable[x] != x)
  {
    reduceLabels(equivTable, equivTable[x],y);
  }
  else
  {
    equivTable[y] = x;
  }
}
```

Bibliography

[alm99] *The astronomical Almanch for the year 1999.* U.S. Government Printing Office, 1999.

[CAP05] CAPANINA Consortium. Capanina stratospheric broadband. http://www.capanina.org, Feb 2005.

[CMA00] N. J. Colell, J. N. Martin, and I. F. Akyildiz. The halo network. *IEEE Communications Magazine*, 38(6):142–148, 2000.

[DFO97] G. M. Djuknic, J. Freidenfelds, and Y. Okunev. Establishing wireless communications services via high-altitude aeronautical platforms: A concept whose time has come? *IEEE Communications Magazine*, 35(9):128–135, 1997.

[dWV03] Spektrum der Wissenschaft Verlag. Basics 2 - mein teleskop. *Sterne und Weltraum*, 2003.

[Föl94] O. Föllinger. *Regelungstechnik: Einführungin die Methoden und ihre Anwendung.* Hüthig, eighth edition, 1994.

[FVMK+04] F. Friedl-Vallon, G. Maucher, A. Kleinert, A. Lengel, C. Keim, H. Oelhaf, H. Fischer, M. Seefeldner, and O. Trieschmann. Design and characterisation of the balloon-borne Michelson Interferometer for Passive Atmospheric Sounding (MIPAS-B2). *Applied Optics*, 2004.

[GDT+01] D. Grace, N.E. Daly, T.C. Tozer, A.G. Burr, and D.A.J. Pearce. Providing multimedia communications from high altitude platforms. *International Journal of Satellite Communications*, (19):559–580, 2001.

[Ger05] German Aerospace Center (DLR). Optical communications group. http://www2.dlr.de/KN/KN-DN/groups/optic/ocg/ocg, 2005. Feb.

[Gig04] D. Giggenbach. *Optimierung der optischen Freiraumkommunikation durch die turbulente Atmosphäre - Focal Array Receiver.* PhD thesis, Universität der Bundeswehr München - Fakultät für Elektrotechnik und Informationstechnik Institut für Informationstechnik, 2004.

[GO05] R. Grünwald and D. Oertel. Leichter-als-luft-technologie - innovations- und anwendungspotenziale. Technical report, Büro für Technikfolgen-Abschätzung beim Deutschen Bundestag (TAB), 2005.

[Goo96] J. W. Goodman. *Introduction to Fourier Optics.* McGraw-Hill, second edition, 1996.

[GTK+01] D. Grace, J. Thornton, T. Konefal, C. Spillard, and T.C. Tozer. Broadband communications from high altitude platforms - the helinet solution. Invited Paper for Wireless Personal Mobile Conference, Aalborg, Denmark, 2001.

[Hab00] P. Haberäcker. *Praxis der Digitalen Bildverarbeitung und Mustererkennung.* Hanser, 2 edition, 2000.

[Mor05] R. Moros. Regelung: Wahl regler u. einstellregeln f. strecke mit ausgleich h. ordnung. http://techni.tachemie.uni-leipzig.de/reg/regeintn.html, Feb 2005.

[OLM+03] G. G. Ortiz, S. Lee, S. Monacos, M. Wright, and A. Biswas. Design and development of a robust atp subsystem for the altair uav-to-ground lasercomm 2.5 gbps demonstration. In *Proceedings of the SPIE*, volume 4975, pages 103–114, 2003.

[RK02] J. Rossmann and H. Kiendl. Steuerungs- und regelungstechnik. Skript zur Vorlesung, 2002.

[ST91] B. Saleh and M. Teich. *Fundamentals of Photonics*. Wiley-Interscience Publications, 1991.

[TA 04] 1394 Trade Association. *IIDC 1394-based Digital Camera Specification Ver.1.31*, 2004.

[TG01] T.C. Tozer and D. Grace. High-altitude platforms for wireless communications. *IEE Electronics and Communications Engineering Journal*, 13(3):127–137, 2001.

[Wik05] Wikipedia, the free encyclopedia. Pid controller. http://en.wikipedia.org/wiki/PID_controller, May 2005.