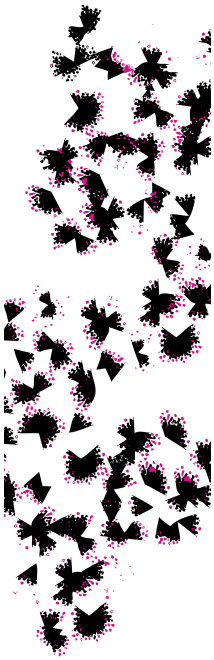


Implementation of a Monocular Structured Light Vision System for Pipe Inspection Robot PIRATE

M. (Mark) Reiling

MSc Report

Committee:
Prof.dr.ir. S. Stramigioli
Dr.ir. J.F. Broenink
Dr.ir. E.C. Dertien
Dr.ir. C.J.A. Pulles

August 2014

Report nr. 016RAM2014
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Summary

The aim of the Pipe Inspection Robot for Autonomous Exploration (PIRATE) project is to develop an autonomous robot platform for in-pipe inspection of small diameter, low pressure (urban) gas distribution mains.

In order to assess the quality of the pipe detailed information on condition of the network and accurate location of deformation of the pipes, bends, dents and other sources for stress on the pipe wall is needed. Quantitative information on current network diameter and consistency is necessary for (autonomous) navigation of the robot through the pipes.

This thesis describes the development of a prototype compact structured light sensor which is capable of measuring inner pipe geometry. The presented arrangement of projector and image sensor for optical triangulation optimizes sensitivity within available space. Moreover, this arrangement gives the camera an unobstructed view.

The designed system utilizes a laser and diffractive optical element to project a fixed pattern on the surface of the pipe. A circular light pattern is chosen to simplify the detection and extraction of the required features to reconstruct the pipe's inner surface. Processing algorithms to extract the features are optimized for low computational load.

Contents

1 Introduction

The aim of the Pipe Inspection Robot for Autonomous Exploration (PIRATE) project is to develop an autonomous robot platform for in-pipe inspection of small diameter, low pressure (urban) gas distribution mains.

In order to assess the quality of the pipe detailed information on condition of the network and accurate location of deformation of the pipes, bends, dents and other sources for stress on the pipe wall is needed. Quantitative information on current network diameter and consistency is necessary for (autonomous) navigation of the robot through the pipes.

In previous studies(Drost, 2009; Mennink, 2010; Brilman, 2011) a prototype of a vision system has been developed that uses a camera and a laser cone projection. This setup can be used as sensor for inspection as well as navigation. Furthermore, image processing software has been written to detect the projected laser circle from which the diameter of the pipe is extracted.

Research has been done (Brilman, 2011) on a suitable embedded platform. The embedded platform used by PIRATE will be the Gumstix Overo Water COM. This is a computer-on-module containing an ARM processor. Embedded Linux will be used as operating system.

1.1 Problem Statement and Goals

The problem is that the prototype vision system is to large for use in small diameter pipes. Therefore, the current prototype must be miniaturized.

The goals of this research assignment are:

- Design a front module for Pirate, housing the vision system, which is able to move through pipes as small as 57 mm, including obstacles like mitered bends and T-joints.
- The software interface between the 'head' of Pirate and the control loop will be sets of data points representing the projected circle of the laser inside the pipe.

1.2 Outline

First, a paper is presented which summarizes the background, design and results of this research. In Chapter 3, the software architecture is described. Last, Chapter 4 ends with conclusions and recommendations.

2 Paper: A Monocular Structured Light Vision Sensor for Inner Pipe Inspection and Surface Reconstruction for Embedded Applications

Abstract - This paper describes the design of a prototype compact structured light sensor. The presented arrangement of projector and image sensor for optical triangulation optimizes sensitivity within available space. Moreover, this arrangement gives the camera an unobstructed view. This allows measuring the inner pipe geometry in front of the sensor so it can be used for obstacle detection. A circular light pattern is used to simplify the detection and extraction of the required features to reconstruct the pipe's inner surface. Image processing algorithms to extract the features are optimized for small computational load and are executed on an embedded platform.

A Monocular Structured Light Vision Sensor for Inner Pipe Inspection and Surface Reconstruction for Embedded Applications

***Abstract*— This paper describes the design of a prototype compact structured light sensor. The presented arrangement of projector and image sensor for optical triangulation optimizes sensitivity within available space. Moreover, this arrangement gives the camera an unobstructed view. This allows measuring the inner pipe geometry in front of the sensor so it can be used for obstacle detection. A circular light pattern is used to simplify the detection and extraction of the required features to reconstruct the pipe's inner surface. Image processing algorithms to extract the features are optimized for small computational load and are executed on an embedded platform.**

I. INTRODUCTION

The aim of the Pipe Inspection Robot for Autonomous Exploration (PIRATE) project [5] is to develop an autonomous robot platform for in-pipe inspection of small diameter, low pressure (urban) gas distribution mains. This low pressure network mostly consists of smooth PE, PVC and grey cast iron pipes.

In order to assess the quality of the pipe detailed information on condition of the network and accurate location of deformation of the pipes, bends, dents and other sources for stress on the pipe wall is needed. Quantitative information on current network diameter and consistency is necessary for (autonomous) navigation of the robot through the pipes.

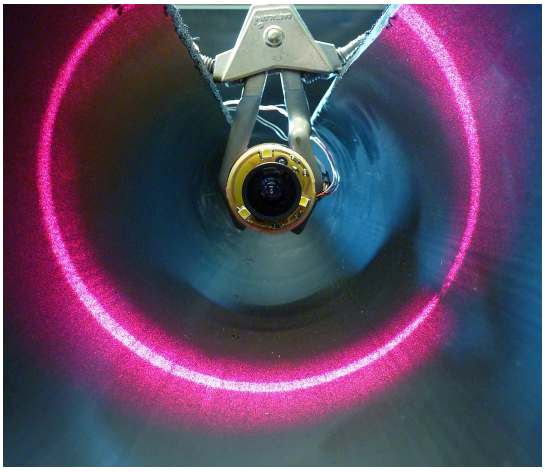


Fig. 1. Sensor prototype and projected pattern in a 80 mm pipe

A. Requirements

Design requirements [6] have been determined based on statistical data available on the (Dutch) gas distribution net [7]. The main usage area for this sensor poses restrictions on the size. The robot (and sensor) must be able to move through a sharp (mitered) bend between pipes with an inner diameter of 57 mm. Measuring inner pipe geometry is required for pipe diameters in the range of 57 mm up to 119 mm.

As previously mentioned, the PIRATE project is aiming towards the development of an autonomous robot. This goal poses restrictions on power usage, available on-board computational power and data storage capacity. No strong demands are posed for these restrictions but they are important guidelines for the design.

B. Related research

Structured light sensors are used for inner pipe inspection and obstacle detection in (semi) autonomous robots, however strong limitations exist in size and minimum detectable radius.

Various methods have been presented using cameras for landmark detection and obstacle avoidance for (semi) autonomous inspection robots [8][1]. However, the need for camera separation for a stereo setup intended for small diameter pipes is a severe drawback. Acquiring 3D information can also be achieved from one camera using time-of-flight imagery, in [10] this concept is used for landmark detection in pipelines.

A problem of aforementioned solutions is the limited amount of feature points to be visible for a stereo vision or time-of-flight imagery system. This makes it difficult to perform quantitative measuerments, especially for a robot assessing mainly in smooth PE and PVC pipes.

Various research robots aimed for pipe inspection use laser or ultrasonic profiling sensors to accurately map the environment [11][12]. Although both laser and ultrasonic profiling can be used, the latter cannot always be used to map the surface of the pipe in front of the robot. The reason being that specular reflection will only occur if the average depth of the surface irregularities is substantially less than the wavelength of the source. In smooth pipes, an ultrasonic sensor therefore needs to be directed perpendicular to the pipe wall. Additionally, an array of transducers or a rotating transducer would be necessary to map the full circumference of the pipe. Furthermore, the small diameter of urban gas pipes means that the frontal area of the robot is small and a multisensory approach, as used in most (semi) autonomous robots is not possible.

In order to measure the geometry in front of the robot for obstacle avoidance, landmark detection and surface reconstruction, a laser profiling sensor seems to be the best solution.

All the mentioned (semi) autonomous robots are targeted at pipes with diameters between 200 mm and 600 mm and

most use a commercially available sensor. Commercially available sensors have been found for diameters as small as 127 mm [15]. Multiple sensors have been developed for research purposes [9][13], however, none of these target either the required wide range in pipe diameters from 57 mm to 119 mm, nor have the capability of storing the raw (color) images and measure inner pipe geometry in front of the robot.

C. Proposed Concept

The confined workspace in which the sensor will operate dictates the overall shape of the sensor. Furthermore, the limited resources enforce optimal use of the geometric shape of the pipes to limit computational load while keeping good accuracy and precision.

The proposed concept, shown in Figure 2, is a laser profiling sensor that uses a camera and laser projector to project a circular shaped pattern on the pipe wall. The choice for a circular shaped pattern is based on previous studies [2][3], where efficient algorithms are presented to detect landmarks.

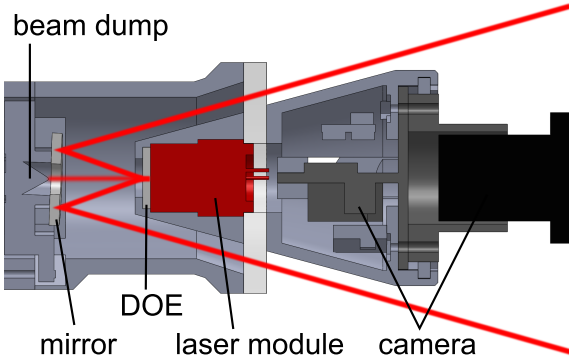


Fig. 2. A cross-cut of the sensor's concept

1) Sensor configuration: Since the sensor has to detect obstacles that are in front of the robot, the optical axes of camera and projector should be oriented in the same direction. Also, when the projector is not positioned in the field of view of the camera, imagery data can be stored for off-site analysis. Two possible configurations are illustrated in Figure 3.

It is desirable to place the projector onto the optical axis of the camera for equal sensitivity of points along equal radii. This configuration will also give a one-to-one correspondence between any point on the laser cone detectable by the camera and the corresponding set of points on the image plane, as shown in Figure 3(b). The reason being that the focal point of the camera O_c is located within the (virtual) projection cone. In case the point O_c is located outside the cone, ambiguities exist when reconstructing the image.

Based on aforementioned arguments, the projector is positioned behind the camera and the corresponding optical axis are aligned. A problem arises when considering the physical sizes of the components of the sensor. The projector has to be placed far enough from the camera for it not to obstruct the projected beam. This distance plus the length of the projector and camera lens is problematic for a sensor that is required to fit onto a robot able to move through a mitered bend in small (57 mm) diameter pipes.

The implemented solution is to use a mirror, making optimal use of the available space between camera and projector. Projector and camera are placed back to back and a mirror is placed at the backside of the sensor as illustrated in Figure 2.

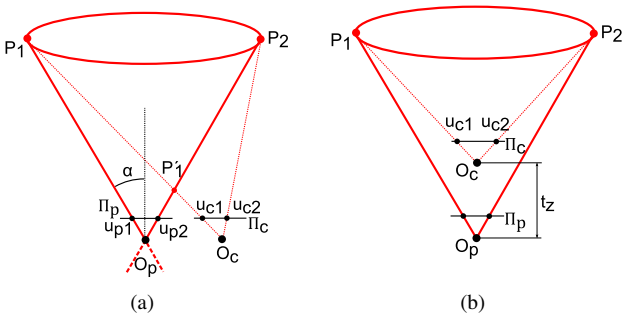


Fig. 3. Optical arangement for triangulation

2) Projector: The pattern is created by diffraction of the laser light with a diffractive optical element (DOE). Unfortunately, the projection pattern created by the DOE contains a central laser spot due to undiffracted light in the centre of the diffraction pattern. Stray light, originating from this undiffracted spot, will limit the sensitivity of the sensor. A beam dump is integrated to limit the intensity of this stray light.

II. Analysis

This section summarizes the theory used to reconstruct the illuminated environment from an image frame. Furthermore, the impact of important design parameters of the configuration in Figure 3(b) are discussed.

It is assumed that the camera is calibrated and we can model it as a pinhole camera. Unless noted otherwise, coordinates are with respect to the camera coordinate frame. Homogeneous coordinates are denoted with a tilde sign on top to differentiate them from cartesian coordinates. The analysis presented here will largely follow the work described in [2].

A. Optical triangulation

Points in space illuminated by the projector are reconstructed using optical triangulation. In Figure 3(b), the points P, O_c and O_p, define a triangle that is fully determined by the point correspondences and the intrinsic and extrinsic parameters of the sensor. As the triangle is known, the 3D coordinates of the point can be calculated. It is noted that the line O_cO_p is called the triangulation base and its length the base distance t_z. The base distance is an important design parameter as it has a large effect on the measurement resolution and the measurement range of the system. Note

that the configuration illustrated in Figure 3(b) still holds when the virtual image of the mirror is considered.

B. Camera Model

The pinhole model relates the camera frame coordinates $\tilde{X_c} = [X_c, Y_c, Z_c, 1]^T$ with the image plane coordinates $\tilde{u_c} = [u'_c, v'_c, w]^T$ as shown in Equation 1, where the 3×3 matrix K_c is called the camera intrinsic matrix.

$$\tilde{\boldsymbol{u}}_c = \boldsymbol{K}_c[\boldsymbol{I}|\boldsymbol{0}]\tilde{\boldsymbol{X}}_c\,, \quad \boldsymbol{K}_c = \begin{bmatrix} f_{c,u} & s_c & u_{c,0} \\ 0 & f_{c,v} & v_{c,0} \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The intrinsic camera matrix models the following intrinsic parameters:

- $f_{c,u}$ and $f_{c,v}$ are the camera normalised focal lengths expressed in number of pixels. Pixels are assumed square: $f_{c,u} = f_{c,v} = f_c$
- $u_{c,0}$ and $v_{c,0}$ are the coordinates of the principal point $\boldsymbol{p}_c$ of the image sensor, expressed in pixels.
- s_c, the skewness factor of the camera. The camera plane is assumed to be orthogonal: $s_c = 0$

The actual pixel coordinates $\boldsymbol{u}_c = [u_c, v_c]^T$ can be recovered by dividing $\tilde{\boldsymbol{u}_c}$ through w.

C. Projector Model

The projector is considered an inverse camera. As the projector plane is a virtual plane the focal distance and optical centre can be chosen arbitralily and are assumed to be equal to those of the camera.

$$\tilde{\boldsymbol{u}}_p = \boldsymbol{K}_p[\boldsymbol{I}|\boldsymbol{0}]\tilde{\boldsymbol{X}}_p\,, \quad \boldsymbol{K}_p = \boldsymbol{K}_c \tag{2}$$

The projector frame coordinates $\boldsymbol{X}_p$ can be related with the camera frame coordinates $\boldsymbol{X}_c$ using a homogeneous coordinate transformation: $\tilde{\boldsymbol{X}}_p = \boldsymbol{P}^c_p\tilde{\boldsymbol{X}}_c$. Equation 2 can then be written as in Equation 3 where the rotation matrix $\boldsymbol{R}^p_c$ and the translation vector $\boldsymbol{t}^p_c$ are the extrinsic parameters of the ASV system.

$$\tilde{\boldsymbol{u}}_p = \boldsymbol{K}_p[\boldsymbol{R}^p_c|\boldsymbol{t}^p_c]\tilde{\boldsymbol{X}}_c \tag{3}$$

The circular pattern that is used for the pipe profiling system is generated by projecting a cone of light that is described in the projector reference frame by the parametric equation given in Equation 4, parameterized by u and θ, where α is the half fan angle of the cone.

$$\boldsymbol{X}_p = \begin{bmatrix} u\tan\alpha\cos\theta \\ u\tan\alpha\sin\theta \\ u \end{bmatrix} \tag{4}$$

An implicit equation of the cone in the projector reference frame is obtained by adding the squares of X_p and Y_p, referring to Equation 5.

$$X_p^2 + Y_p^2 - Z_p^2\tan^2\alpha = 0 \tag{5}$$

In Equation 4 and Equation 5 it is implicitly assumed that the apex and central axis of the cone coincide with the focal point and optical axis of the projector respectively. In this way the projector image plane coordinates describe a perfect circle with radius r_p and positioned at p_p as can be derived from Equation 2 and Equation 4, see Equation 6. Furthermore, the cone projector is symmetric around its optical axis and the rotation matrix therefore only depends on the angles of rotation around the x- and y-axes of Ψ_c and can be written as $R^p_c(\varphi, \omega) = R^p_{c,y}(\omega)R^p_{c,y}(\varphi)$.

$$\boldsymbol{u}_p = \begin{bmatrix} r_p\cos\theta + u_{p,0} \\ r_p\sin\theta + v_{p,0} \end{bmatrix}, \quad r_p = f_p\tan\alpha \tag{6}$$

D. 3D reconstruction

The optical axes of camera and projector are aligned, therefore the extensive parameters can be described as follows:

$$\boldsymbol{R}^p_c = \boldsymbol{I}; \quad \boldsymbol{t}^p_c = [0, 0, t_z]^T \tag{7}$$

As a result, both $\boldsymbol{u}_p$ and $\boldsymbol{u}_c$ desribe perfect circles at equidistant points from a plane perpendicular to the optical axis. By defining

$$r_c = ||\boldsymbol{u}_c - \boldsymbol{p}_c||; \quad r_p = ||\boldsymbol{u}_p - \boldsymbol{p}_p||; \quad R_c = \sqrt{X_c^2 + Y_c^2} \tag{8}$$

Equation 1 and Equation 3 can be written using Equation 7 as in Equation 9 which defines two equations for the two unknowns.

$$r_c = \frac{f_c}{Z_c}R_c; \quad r_p = \frac{f_c}{Z_c + t_z}R_c \tag{9}$$

By substraction, Equation 9 can be resolved for R_c and Z_c, see Equation 10, where the distance $d_r = r_c - r_p$ is the radial disparity.

$$R_c = \frac{Z_c}{f_c}r_c; \quad Z_c = \frac{r_p t_z}{d_r} \tag{10}$$

E. Alignment

So far, it was assumed that camera and projector were perfectly aligned. In the event of a misaligment it is possible to rectify the image, i.e. mathematicaly align image and projector plane. As long as the focal point O_c is located within the projection cone, a one-to-one relation is maintained.

Due to the quantisizing nature of the image sensor the rectification process can generate some distortion of the image. Furthermore, the result relies on the accuracy at which $\boldsymbol{R}^p_c$ and $\boldsymbol{t}^p_c$ can be determined.

Finding point to point correspondences is made easier using epipolar geometry. In Equation 11, the fundamental matrix is used to describe these correspondences. Equation 12 describes the epipolar lines in the camera's image plane Π_c on which the laser projection can be found.

$$\tilde{\boldsymbol{u}}_p^T\boldsymbol{F}\tilde{\boldsymbol{u}}_c = 0, \qquad \boldsymbol{F} = \boldsymbol{K}_p^{-T}(\boldsymbol{t}^p_c \times \boldsymbol{R}^p_c)\boldsymbol{K}_c^{-1} \tag{11}$$

$$\tilde{\boldsymbol{l}}_c = \boldsymbol{F}^T\tilde{\boldsymbol{u}}_p \tag{12}$$

Using Equation 6 as input for Equation 12 and adding shifts in $\boldsymbol{t}_c^p$ it is found that a translation in t_x or t_y results in a shift of the epipolar line. In case of a misalignment in $\boldsymbol{R}_c^p$, $\boldsymbol{l}_c$ describes a distorted circle. This introduces a slight deviation in disparity causing a change in sensitivity along equal radii.

F. Resolution

The equations for coordinate reconstruction as defined in Equation 10 can be used to analyse the errors on the coordinate reconstruction. In order to analyse the impact of the spatial image quantization as a function of the extensive parameters, $f_c \tan\alpha$ is substituted for r_p. The absolute error due to quantization in the spatial (radial) direction is then computed using the partial derivative of Equation 10 with respect to r_c.

The absolute errors Δ_{R_c,r_c} and Δ_{Z_c,r_c} are expressed in Equation 13 and Equation 14 where Δ_{r_c} is the sampling error. From Equation 13, it follows that the resolution of the system is enhanced when $t_z > 0$.

$$\Delta_{R_c,r_c} = \left| \frac{f_c t_z \tan^2\alpha}{(r_c - f_c \tan\alpha)^2} \right| \Delta_{r_c} \tag{13}$$

$$\Delta_{Z_c,r_c} = \left| \frac{f_c t_z \tan\alpha}{(r_c - f_c \tan\alpha)^2} \right| \Delta_{r_c} \tag{14}$$

Equation 10 can be used to rewrite these equations as a function of R_c.

G. Measurement Range

The measurement range of the sensor is determined by the base distance t_z between camera and projector, the half fan angle α of the laser cone and the field of view v of the camera. The field of view is determined by the dimensions of the image sensor, here denoted by $N_u N_v$ pixels, and the focal length f_c of the camera. Assuming $N_v = N_u$, the field of view v is expressed as:

$$v = 2 \arctan \frac{N_v}{2 f_c} \tag{15}$$

Note that in case the camera is placed in front of the projector ($t_z > 0$), a constraint is imposed on the minimum radial distance to be measured (assuming that $\alpha > v$) and that if $t_z > 0$, a constraint is imposed on the maximum distance to be measured (assuming that $\alpha < v$).

As noted in the previous section, the resolution is enhanced when $t_z > 0$. The minimum radial distance $R_{c,min}$ for this case follows from the geometry of the system and is expressed in Equation 16; the minimum measurement distance in z-direction is given in Equation 17.

$$R_{c,min} = \frac{\tan\alpha \tan\frac{v}{2}}{\tan\frac{v}{2} - \tan\alpha} t_z \tag{16}$$

$$Z_{c,min} = \frac{R_{c,min}}{\tan\alpha} - t_z \tag{17}$$

The maximum distances to be measured are in theory infinite, however in practice limited by the amount of light reflected back to the camera and the sensitivity of the camera image sensor.

III. Implementation

Complete surface reconstruction is possible only if the position and orientation of the sensor inside the pipe is known. Asuming a perfect pipe, the precise radial position and orientation of the sensor can be found [2]. However, distortions of the pipe wall and the inherently limited accuracy of the sensor makes obtaining accurate orientation and radial position of the sensor difficult, if not impossible.

To aid in finding the accurate orientation along the pipes centre a set of sensors comprising an inertial measurement unit (IMU) will be used. In addition, the state of the robot, to which the sensor will be attached, can be used to find approximate orientation and radial position of the sensor. In Figure 4 the realized prototype and sensor board are shown.

In order to link multisensory data in case the sensor is moving, time differences between measurements have to be known. Therefore, it is neccessary to apply timestamps to the measurents.

Timestamps of the images taken by the USB webcam are added when the first byte of a frame is received. The latency between image exposure and the timestamp beeing applied is caused by processing time in the cameras image signal processor (ISP), bus latency and the delay in detection of the first bytes arrival by the driver. The precision in which this delay can be determined (Jitter) impacts the accuracy in measuring the pipes geometry.

The software structure and algorithms for laser detection and keeping track of timestamps will be discribed in subsection III-B. First, the mechanical design is presented and design details are explained.

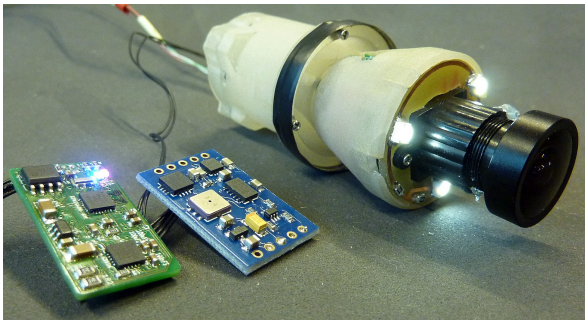


Fig. 4. Sensor in 3D printed housing with sensor- and control board

A. Mechanical design

In order to quickly evaluate the design, all body parts of the sensor are fabricated using a 3D-printer. The use of this fabrication technique allows complex shapes for relatively low cost and minimal development time.

An exploded view is shown in Figure 5 and the realized prototype in Figure 4. The prototype has a total length of 80 mm and has at its widest point a diameter of 31 mm. The size and shape of the presented prototype can pass a mitered bend inside a 57 mm pipe as required.

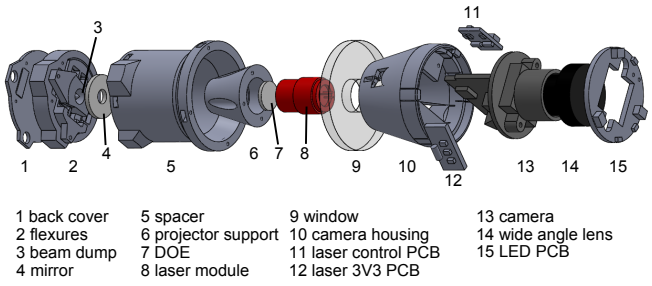


Fig. 5. Exploded view of the sensor

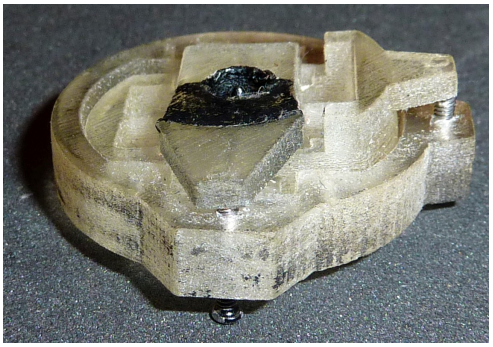


Fig. 6. 3D printed flexure support with beam dump

The small size means that the free path for the laserbeam, as can be seen in Figure 2, is narrow and tolerances are small. These small tolerances pose a problem due to the limited accuracy of 3D-printing.

Of both radial and orientational misalignment, the latter has the most impact on the path of the laser beam. This is caused by the fact that orientational misalignments of projector, mirror and camera accumulate. Furthermore, this misalignment is proportional to the length of the path while a radial deviation is constant. In addition, orientational misaligment causes variations in sensitivity along equal radii as mentioned in section II.

The presented solution is a flexure mechanism which allows for orientational adjustments that are free of play. Note that orientation is only necessary along two axis, as the setup is rotational symmetric. Although orienting either camera, projector or mirror has the same effect, solely the latter two can be used to aim the laser beam for a clear path inside the housing. Furthermore, calibration is easier if the full projection is visible and the mirror support can be easily accessed without blocking the path of the laserbeam.

1) Mirror: The required mirror must have a top reflective surface; a back coating on a transparent surface will introduce multiple reflections due to internal reflections. In addition, the mirror should be as thin as possible to save space and must be ring shaped to allow placement of the beam dump behind the mirror. No commercial available mirror was found that met these requirements. The solution was to fabricate the mirror from a harddisk; it has a top reflective coating, is extremely flat and is easy to machine. The mirror and flexure support are shown in Figure 6.

2) Mirror Flexure Support: 3D-printing allowed the flexure support to be fabricated as a single piece. Ideally, the rotational axis would be along the mirrored surface but in order to keep a small diameter it is positioned slightly backwards. Adjusting the orientation of the mirror will therefore cause a slight deviation in t_z but has no impact on symmetry.

3) Beam Dump: The projector consist of a 5 mW laserdiode of 655 nm inside a 10 mm x 14 mm housing with collimating lens and a DOE (DE-R 219 of HOLOEYE Photonics [14]) to create the projection pattern. As noted in subsection I-C, the projection pattern created by the DOE contains a central laserspot due to undiffracted light in the centre of the diffraction pattern. It is found that this severely limited the contrast in the pictures.

In order to limit the amount of back reflection and scattering of the undiffracted light in the sensor, a beam dump is designed. The basic principle of a beam dump is to convert light into heat. The energy that has to be absorbed here is extremely low as only a 5 mW laser is used and the 0^{th} order intensity (undiffracted spot) is only about 10% [14]. The design is based upon a common conical design available at many optical suppliers. The beam dump is integrated into the design of the mirror flexure support as can be seen in Figure 6 and is painted matte black.

B. Lasercurve extraction algorithms

This subsection describes the approach for the laser curve extraction, making use of the fact that all image data is radially oriented around the principal point.

The GStreamer framework [17] is used to implement the algorithms. A main advantage is that it is optimized for applications that put high demands on latency and uses timestamping to keep track of this. As mentioned in section III timestamps are required to be able to combine different measurements.

The processing steps used in successive order are:

- color space conversion (YUY2 to Y42B),
- undistortion and polar mapping,
- thresholding,
- dilation,
- weighted averaging.

Both the undistortion mapping as the polar mapping are not one-to-one and require interpolation to find values inbetween pixels. As the mapping can be calculated a-priori, an optimal interpolation scheme can be chosen. The resulting composite mapping is subsequently converted to a much faster fixed-point representation using nearest neighbour interpolation.

In the current implementation, however, undistortion and polar mappings that were available as fixed point created using nearest neighbour interpolation are combined. It is noted that this double quantization will limit the resolution and have to be improved.

1) Color space conversion: The camera used is a webcam with a maximum resolution of 1280 x 960 which outputs frames in a YUY2 color space format (YUY2 uses the

YUV type color space, properly called YCbCr for digital video). This color format is ideal for the proposed algorithm because the Y (luma) channel represents the intensity and can be directly used as input in the proposed algorithm. In mostly dark environments, only a little more information can be extracted from the color information than is acquired using the intensity channel. To lower computational load, a resolution of 640 x 480 is used.

2) Undistortion: Several nonlinear distortions are introduced in the lens. Especially wide-angle lenses are known to have large distortions. Two methods have been investigated to acquire correct measurements; Camera calibration (Undistortion mapping) and the use of a lookup table. Considering both methods, camera calibration is chosen for optimal performance. The undistortion and polar mapping are performed in successive order and can be written as one composite mapping. Therefore, the addition of the undistortion mapping has no further impact on performance.

3) Polar mapping: The polar mapping is shown in Equation 18 and Equation 19.

$$r_c = \sqrt{(u_c - u_{c,0})^2 (v_c - v_{c,0})^2}, \quad r_c < r_{c,max} \tag{18}$$

$$\theta = \arctan \frac{v_c - v_{c,0}}{u_c - u_{c,0}} \tag{19}$$

The advantage of having this representation is that the laser can be found in the vertical direction on every column, reason being the approximately radial orientation of the original image. The preceeding makes finding the laser curve a one-dimensional search problem on every column.

Another advantage is that not all pixels have to be mapped because the area in which the laser can be found is known. The minimum distance can be calculated from Equation 16 and rewriting for r_c, the maximum distance depends on the dynamic range of the sensor and can be adjusted downwards.

4) Thresholding and dilation: Thresholding is used in order to eliminate pixels from the image that do not contain a value of interest. These values are generally considered noise, but can also be reflection of the laser itself. The circle projected by the laserbeam and captured by the camera often contains many small unwanted gaps. A 3×3 dilation kernel is used to eliminate these gaps and create a better profile picture.

5) Weighted averaging: Ideally, the shape of the beam is used, however it has been found [4] that finding the weighted averaged value in each column is faster and gives good results. This algorithm will give sub-pixel accuracy to this measurement.

C. Calibration

The calibration procedure has two goals; finding the intensive and extensive parameters of the system and calibration of the mirror to align the optical axis of the lens and projector. The calibration can be described as a three step procedure:

- calibrate camera,
- calibrate mirror,
- find extensive parameters $\boldsymbol{R}_c^p$, $\boldsymbol{t}_c^p$ and α.

All calibration is performed using the cameras maximum resolution of 1280 x 960. For the use in the algorithms on the embedded target a resolution of 640 x 480 is used. Therefore, the parameters in the camera matrix K_c should be divided by 2 while the distortion parameters are identical.

1) Camera calibration: Calibration of the camera is performed using the available functions in OpenCV [16]. The camera model used is defined in Equation 1, only the skew factor $s_c = 0$. The lens distortion is modeled using 2 tangential and 3 radial distortion parameters. The tangential distortion occurs in case the lens is not aligned parallel to the image sensor, radial distorion models the distortion of the lens itself.

2) Mirror calibration: Translational misalignment will preserve the shape of the captured laserbeam, referring to section II where a projection of the laserbeam onto a surface perpendicular to the z-axis of Ψ_c should be a circle.

A calibration pattern with four circular markers is used to measure the orientation of the camera to the projection plane using a perspective-n-Point (PnP) algorithm. The shape is measured by an edge detection algorithm and a least square fitting is used to fit an elipse to the detected edge points. The calibration is performed by equalizing the major and minor radii.

3) Characterization: Because 3D printed parts are used, the actual position of the focal point of the projector can deviate from the intended design. Due to inaccesability to the inside of the sensor the precise location has to be reconstructed from the measured laser projection.

Using the same setup as used for the mirror calibration where radii of the laser projection are measured along the z-axis of Ψ_c using a micrometer. At each interval, several measurements are taken at each distance and an ellipse fitting is applied using ordinary least squares (OLS), thereby reconstructing a part of the laser cone.

By extrapolation of this cone the focal point of the projector can be found and the orientation is the deviation of the principal axis form the z-axis of Ψ_c. Results of the calibration are shown in Table I.

Parameter	Value
Camera focal distance f_c	375.5 px
Projector fan angle α	34.8 deg
Triangulation base distance t_z	76.3 mm
Orientational misalignment φ, ω	1.0 deg, 0.1 deg
Translational misalignment t_x, t_y	1.4,mm, 0.8 mm

TABLE I

PARAMETERS OF THE REALIZED PROTOTYPE

From Equation 16 and using Table I the minimum radius of a pipe that enables measurement of the full circumference in one image frame is 46.8 mm. Allowing rotation around the optical axis a smaller pipe can be measured. Using the full width of the image sensor, 194 degrees of the cirumference of a pipe with radius of 37.8 mm can still be measured.

D. Experiments

All tests are executed in a PVC pipe with an inner diameter of 118 mm. During the tests the sensor is held in place by stand and is approximately aligned along the centre of the pipe. Since no reference calibration procedures were available and the aligment of the sensor inside the pipe is not measured, no precise mapping to metric data could be done. Therefore all measured distances are expressed in pixels.

1) Regression analysis: Main distortions are expected to be periodic and primarily due to a misaligment of the sensor inside the pipe as discussed in section II. Other distortions may come from deformations of the pipe and are assumed to be smooth and elliptical.

A trigonometric discrete Fourier transform is used to construct a linear regression model. The data is fit using the first four terms of the series and the result is shown in Figure 7. The cause of the disturbance at the series pixels between 168 and 174 is that the wiring of the camera is blocking the laserbeam and are marked as outliers. The standard error of the estimate σ_{est} is 0.4 pixels and the estimated mean $\overline{r}$ is 172.0 pixels.

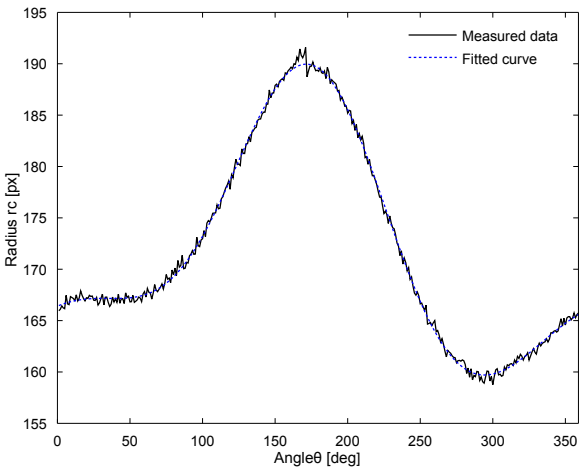


Fig. 7. A 4-term Fourier series curve fiting

2) Noise analysis: Averaging measurements can provide insight into the shape of the noise. A sample size of 100 is used to calculate the mean of measurements from a fixed position, all zero mean noise ideally will lower by a factor of $\sqrt{(}N)$. Measurements of the first 30 pixels are shown in Figure 8 in which a systematic error is seen. The main contribution to this error is expected to be the nearest pixel interpolation (quantization) of the combined polar and undistortion mapping.

This quantization error is likely lowered when the polar and undistortion mapping have floating point precision and only the composite mapping is converted to fixed point notation using nearest neighbour interpolation. Obviously, this error can also be lowered by increasing the image resolution at the cost of higher computational load.

3) Estimated resolution: From the data in Table I and σ_{est}, an estimate can be made for the resolution of the sensor. The quantization error Δ_r in Equation 13 is set to σ_{est} and $\overline{r}$ is used for the value of r_c giving an estimated resolution of 0.35 mm.

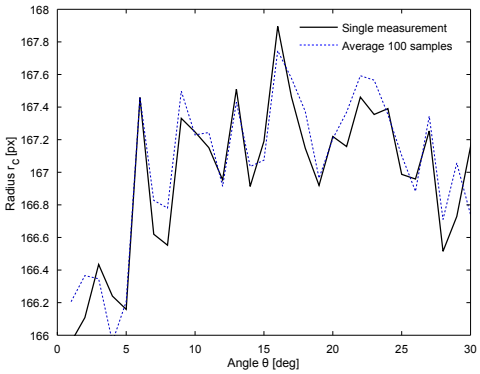


Fig. 8. Effects of averaging 100 measurements

4) Algorithm performance: All experiments are done on a Gumstix Overo Water COM which has an ARM Cortex A8 core running at 720 MHz. The embedded target is running a 3.6 Linux kernel and an EMDebian Wheezy filesystem is installed. All software uses the hardfloat ABI and is compiled using an EMDebian toolchain and 03 optimization. All software is written in C and C++ and all hardware specific optimizations are left to the compiler.

Images are taken at a resolution of 640 x 480 and captured at a rate of 15 frames per second. The average duration over 100 runs of all individual implemented algorithms is shown in Table II.

Algorithm	time [ms]
threshold	4
dilation	30
mapping	12
weighted averaging	3

TABLE II
AVERAGE DURATION OVER 100 RUNS

IV. CONCLUSIONS

This paper presents a prototype structured light sensor that can be used for both quality inspection as well as obstacle and landmark detection for autonomous navigation in small diameter pipes. The arangement of camera and projector maximizes the base distance for optical triangulation within the available space for best sensitivity.

Experiments were done to analyse the resolution of the sensor and it is shown that the dominant noise is due to the quantization in the polar mapping. The standard error of the total noise (quantization and zero-mean noise) is found to be less then 0.4 pixels resulting in a precision of 0.35 mm. The precision of the current design is likely to be improved when the polar and undistortion mapping is calculated with better resolution. However, most improvement can be achieved by increasing the resolution of the camera.

All algorithms are tested on an embedded platform and it is demonstrated that the required processing time of the algorithms for laser detection is less then 50 ms on a Gumstix Overo Water COM.

The size of the presented prototype is small and is able to pass a mitered bend inside a 57 mm pipe. At this diameter, a 360 degrees scan of the pipe can be done as was required when the wide angle lens is replaced by an ultra wide angle or fisheye lens with a minimum vertical field of view of 125 degrees. When the camera and laser module are replaced with smaller ones which are commercially available, the design can be scaled down.

REFERENCES

[1] A. Ahrary, Li Tian, S. Kamata, and M. Ishikawa. "An autonomous sewer robots navigation based on stereo camera information," in Tools with Ar- tificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on, pages 6 pp.633, 2005.

[2] E. H. Drost, *Measurement system for pipe profiling*, Msc Thesis, March 2009.

[3] T. Mennink, *Self Localization of PIRATE Inside a Partially Structured Environment*, Msc Thesis, December 2010.

[4] M. F. Brilman, *Embedded Vision System for PIRATE* Bsc Thesis, June 2011.

[5] E. C. Dertien. *Design of an inpection robot for small diameter gas distribution mains*, Phd Thesis, 2014.

[6] C. Pulles, E. C. Dertien, H. J. van de Pol, and R. Nispeling. "Pirate, the development of an autonomous gas distribution system inspection robot," in International Gas union Research Conference, IGRC 2008, Paris, France, USA, October 2008. Curan Associates.

[7] Pieter van Vollenhoven. *Gasexplosie na breuk van gasdistributieleiding.* Technical report, Raad voor de transportveiligheid, 2002.

[8] Erich Rome, Joachim Hertzberg, Frank Kirchner, Ulrich Licht, and Thomas Christaller. "Towards autonomous sewer robots: the {MAKRO} project". Urban Water, 1(1):57 70, 1999.

[9] Moritz Ritter and Christian W. Frey. "Rotating optical geometry sensor for inner pipe-surface reconstruction."

[10] Jens T. Thieleman, Gøril M. Breivik and Asbjørn Berge. "Pipeline Landmark Detection for Autonomous Robot Navigation using Time-of-Flight Imagery."

[11] H.-B. Kuntze and H. Haffner, "Experiences with the development of a robot for smart multisensoric pipe inspection," in Proc. 1998 IEEE Int. Conf. on Robotics & Automation, Leuven, Belgium, May 1998.

[12] Amir A. F. Nassiraei, Yoshinori Kawamura, Alireza Ahrary, Yoshikazu Mikuriya and Kazuo Ishii, "Concept and Design of A Fully Autonomous Sewer Pipe Inspection Mobile Robot "KANTARO"," in 2007 IEEE Int. Conf. on Robotics & Automation, Roma, Italy, 10-14 April 2007.

[13] Bao Hua Zhuang and Wenwei Thang, "A non-contact laser sensor for pipe inner wall inspection using circular optical cutting method,"

[14] Internet: `http://holoeye.com/diffractive-optics/standard-doe-plastics/` [aug. 11 2014]

[15] Internet: `http://www.maverickinspection.com/services/laser-pipe-profiling/laser-pipe-profiling-intro/` [aug. 11 2014]

[16] Internet: `http://docs.opencv.org/` [aug. 11 2014]

[17] Internet: `http://gstreamer.freedesktop.org/` [aug. 11 2014]

3 Software Implementation

3.1 Use Case

The aim of the Pipe Inspection Robot for Autonomous Exploration (PIRATE) project Dertien (2014) is to develop an autonomous robot platform for in-pipe inspection of small diameter, low pressure (urban) gas distribution mains. The structured light vision sensor, presented in the previous chapter, will be fitted to the front of the inspection robot Pirate.

A schematic overview of the current implementation is given in Figure 3.1. The robot is fully controlled by an operator. Local control loops are implemented on the slave nodes and connected via an RS485 bus. A control panel is connected to an embedded platform where the input is translated to setpoints and sent over the RS485 bus. Feedback of the state of the robot is also sent over this bus.

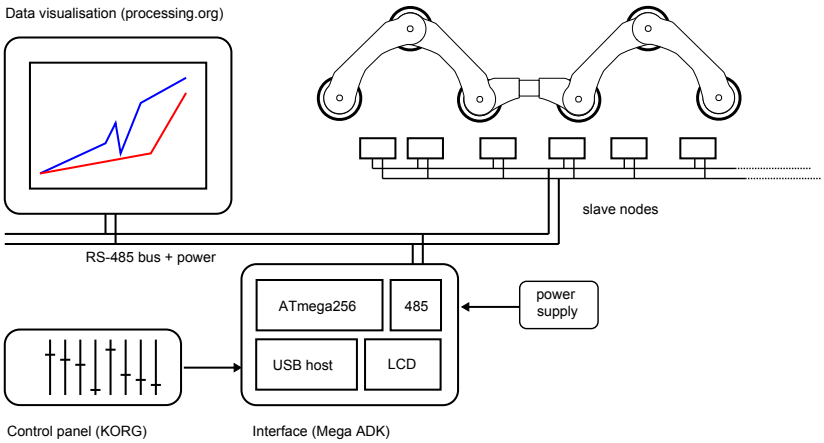


Figure 3.1: Current implementation with control panel (Dertien, 2014)

To aim towards autonomous control of the robot through the pipe network, all processing for navigation, obstacle detection and control must be implemented on the robot itself. A Gumstix Overo Water computer-on-module (COM) (Gumstix, 2013) is chosen (Brilman, 2011) as the main computational platform on the robot. The Overo Water has an ARM Cortex-A8 core running at 720 MHz, a DSP and has 256 MB NAND memory.

Research has been done towards more autonomous control for PIRATE (Doggen, 2010; Mennink, 2010). The algorithms for simultaneous localization and mapping (SLAM) are computationally heavy and not yet optimized. Therefore, they will be executed on a more powerful PC.

The focus is primarily on tethered operation Dertien (2014). An expansion board is designed for the Overo Water that is connected to the PC with a CAT5 Ethernet cable. Power over Ethernet (PoE) is used to supply up to 25.5 W. Details of the expansion board are given in Appendix A.

A schematic system overview of the implementation that is described above is given in Figure 3.2.

3.2 Functional Requirements

It became apparent during the time of this research that a tethered robot is required for further development. This resulted in the use case described in the previous section. Requirements are identified by the examination of this use case.

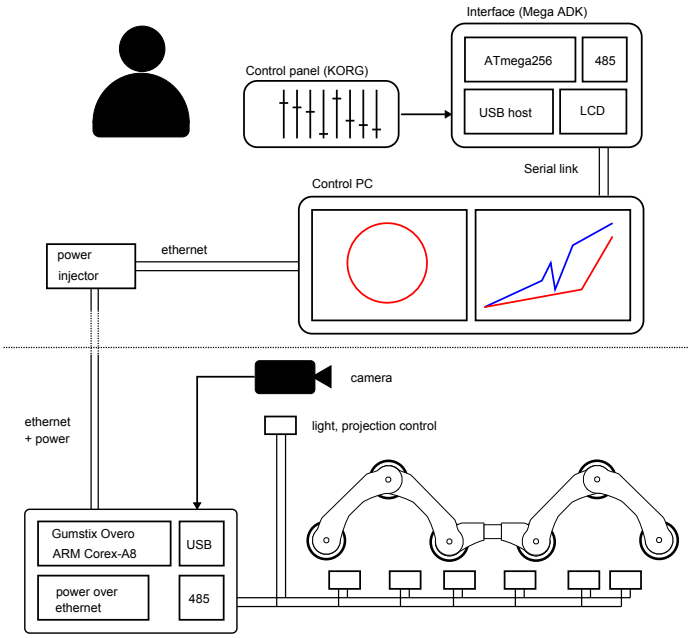


Figure 3.2: Setup with the camera, Overo Water COM and host PC (Dertien, 2014)

At the start of this research, two goals are drafted that are written in Chapter 1. The second goal stated that the interface between the sensor's output and control loop are sets of data points representing the projected circle. In the newly identified use case, the control loop will be implemented at the host PC.

Requirement 1: *Image processing algorithms must be implemented on the Overo Water COM*

To aim towards autonomous navigation it is demanded that the image processing must be performed on the Overo Water COM.

Requirement 2: *Incoming image frames must contain timestamps that are linked to the processed data.*

In order to accurately map the environment, multisensory data is necessary. To combine these data, the relative time between the measurements must be known; therefore, timestamps must be applied to the sensor's images.

Requirement 3: *The software architecture should be modular.*

Modularity of the software provide a means to reuse functional components. Particularly in the current prototyping stage, more functionality will be added and components may need to be rearranged.

3.3 The GStreamer Framework

Gstreamer is an open-source framework for creating streaming media applications and can be used to process any kind of data flow (Wim Taymans, Steve Baker, Andy Wingo, Ronald S. Bultje, Stefan Kost, 2014). The software development kit focuses on documentation and stability for developers building cross-architecture applications(GStreamer, 2014). The design of the framework is component based and can be arranged into arbitrary pipelines. It is designed to have little overhead above the impact of what the core functionality of the components use.

An overview of the GStreamer Framework is given in Figure 3.3

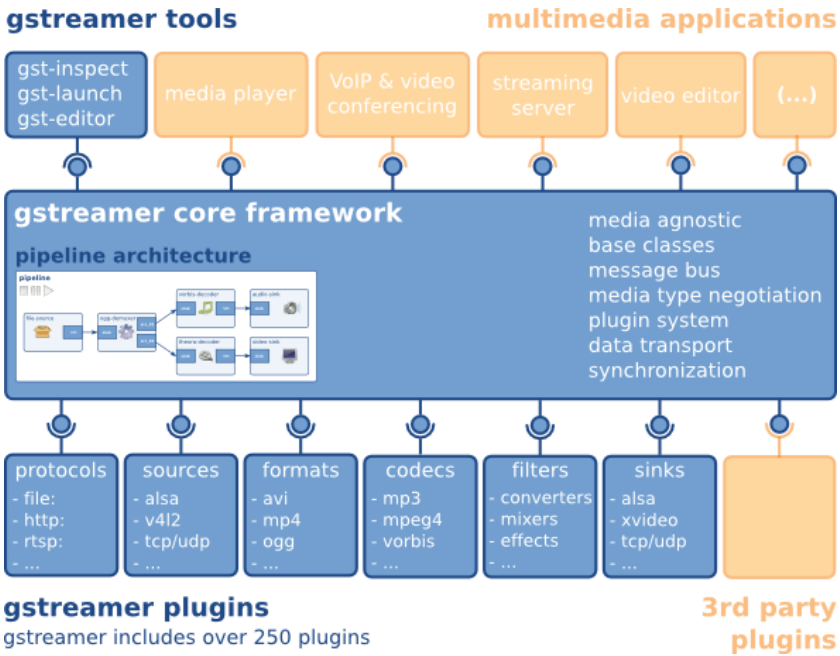


Figure 3.3: Overview of the GStreamer Framework (Wim Taymans, Steve Baker, Andy Wingo, Ronald S. Bultje, Stefan Kost, 2014)

Requirement 3 states that the software architecture should be modular. There are several options to fulfil this requirement. Using the GStreamer framework dictates the architecture of the software to be modular.

One of its core functionality provides a mechanism for synchronization between components. This is implemented using timestamps and fulfils Requirement 2.

An advantage is that there are many available plugins that provide functionality necessary for the application at hand; color space conversions and video and image codecs for example.

A disadvantage is the steep learning curve if not familiar with the C language and the GObject and GLib libraries. Furthermore, the pipeline architecture might not be the best solution for future implementations. Added to that, the number of provided components that will eventually be used is limited. This is due to the fact that many algorithms will be written for a specific purpose.

A feature to consider is that there are plugins available for hardware accelerated codecs. A large bandwidth is required to send live video over the network and storing raw images takes up a lot of space. Codecs to compress video or images are however computationally heavy.

The following plugins are available for use on the Overo Water COM (Texas Instruments, 2014):

- gst-dmai (supports most of the TI davinci and OMAP chips),
- gst-openmax (only OMAP3 is supported),
- gst-dsp (only OMAP3 is supported).

3.3.1 GStreamer Terminology

In this section, a short explanation is given on the terminology used to describe the software architecture.

- Interfaces are called *pads*.

- An *element* is the basic building block and has often one specific function. Elements have sink pads and source pads and are respectively the element's input and output. An element can have more than one sink or source pad, a multiplexer, for example.
- A collection of elements is called a *bin* and is a subclass of an element.
- The top-level bin is called the *pipeline*. The pipeline provides a bus for the application and manages the synchronization of all elements and bins.
- Data is stored in *buffers* and flows between elements from a source pad to a sink pad. Most data transfers, however, are performed by dereferencing pointers for performance.

3.4 Software Structure

Based on Figure 3.2 and the choice for the GStreamer framework, a software structure is determined. A top-level overview of the implemented software structure is illustrated in Figure 3.4. The pipeline on the Overo Water COM collects data from the USB and RS485 bus and serializes and transmits this data to the host PC. Processing of this data can be done in the pipeline, however, this is not required for operation.

The pipeline on the host PC can be an interface between the network socket API and the application for the control and user interface of the robot. The pipelines on the Overo Water COM and host PC can be thought of as one single pipeline. The benefit of this structure is that functionality, added as elements in the pipeline on the host PC, can in principle be migrated to the Overo Water COM with no adjustments other than the need to recompile for the ARM processor architecture.

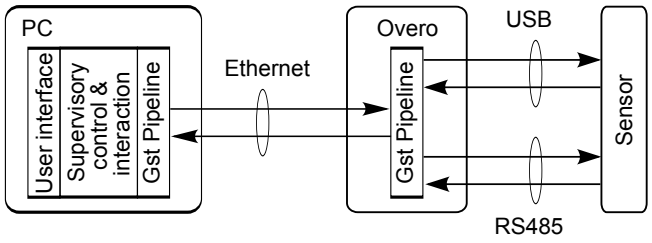


Figure 3.4: Schematic overview of the software structure

3.4.1 Pipeline Architecture on the Overo Water COM

Based on Requirement 1, the image processing must be performed on the Overo Water COM. In Figure 3.5 the pipeline architecture is drawn. The RS485 interface is not yet implemented.

Functionality is implemented in the various bins. The image processing bin reads the received frames, adds timestamps and performs polar and undistortion mapping as described in Chapter 2. The image processing bin outputs the processed data, raw colored video frames and some test data on different source pads.

The buffers are then sent through the polar data, video and test send bins to the RTP bin. The RTP bin implements the real-time protocol, it adds the real-time protocol (RTP) headers to the buffers and sends them across the specified UDP sockets.

Some research have been done to find a protocol that addresses the issue of the non-deterministic behaviour of the Ethernet protocol. Especially the Open Real-Time Ethernet (ORTE) (Petr Smolik, Pavel Pisa, 2008) seemed to be an attractive protocol. ORTE is an open source implementation of the Real-Time Publish Subscribe protocol (RTPS).

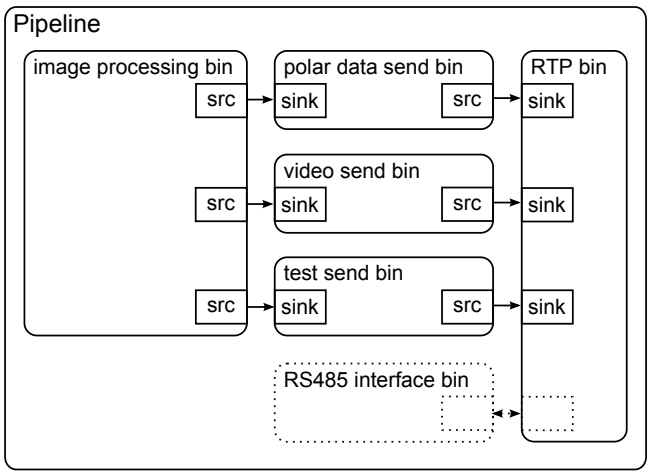


Figure 3.5: Pipeline architecture of the software for the Overo Water COM

Considering that there is no standard element provided that implements the RTPS protocol and that most features of the RTPS protocol are not required in the current implementation, the RTP is chosen.

The RTP can detect out of sequence arrival of data packets and transmits the timestamps. The RTP is stacked on top of the User Datagram Protocol (UDP) for minimal overhead of the header on top of the data. RTP is used in conjunction with the RTP Control Protocol (RTCP) and is used to monitor transmission statistics for quality of service (QoS).

Image Processing bin

The v4l2src reads frames from the webcam and adds a timestamp as soon as the first byte of a frame is detected. The webcam that is used outputs frames in the YUY2 color format. The Y-channel represents the luminance and is the only channel used for processing. The channels are not mapped sequentially in memory and a color space conversion to Y42B is performed to rearrange the data. The elements of the image processing bin are shown in Figure 3.6.

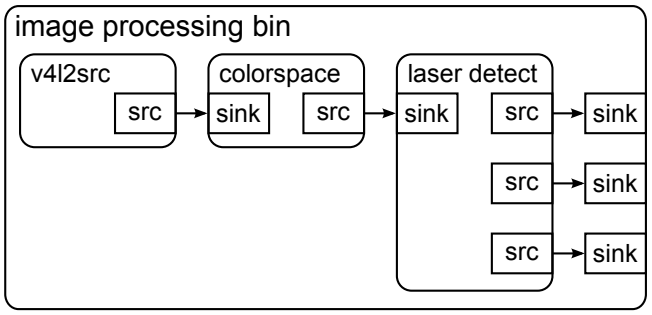


Figure 3.6: Elements that make up the image processing bin

It is noted that performance gain can be achieved by rewriting the polar and undistortion mapping. This mapping is a pixel to pixel mapping as explained in Chapter 2 and can be performed directly on the YUY2 image frame.

Polar Data Send Bin

This bin provides an element that can send pre-recorded data over the network in order to test the pipeline without the sensor. When actual processed data from the sensor is send, it merely functions as an interface between the image processing bin and RTP bin.

Video Send Bin

This bin contains elements to scale down the resolution of the raw images and limit their maximum framerate in order to lower computational load. This method is much more efficient than compressing the images using a codec. The performance can be increased by implementing hardware accelerated elements as mentioned in Section 3.3.

Test Send Bin

This bin is used to send data in order to test the intermediate results between algorithms. For example, this bin is currently used to send the image data after polar and undistorion mapping.

3.4.2 Pipeline Architecture on the host PC

A new pipeline is created at the host PC. Data is acquired from the network sockets and send to the addressed bin. The ximagesink elements output the data using the X Window System. Only the polar data receive bin is implemented in code, the video and test receive bins are implemented using the gst-launch tool provided by the GStreamer framework. For more detail, see Appendix C.

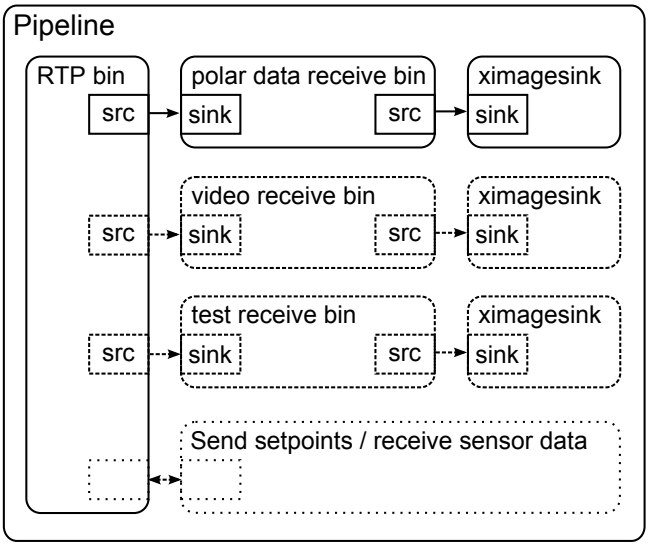


Figure 3.7: Pipeline architecture of the software for the PC

The RS485 bin is not implemented.

Polar Data Receive Bin

The measured data are send to this bin. The makegraph element creates a 2D image from the measurement data in order to display the results as a video stream on the host PC. A color space transformation is used to get a supported format for the ximagesink.

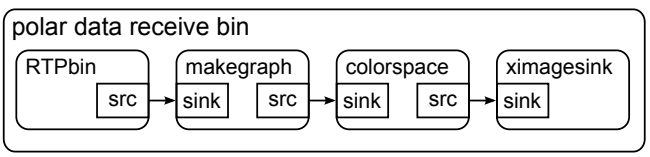


Figure 3.8: Elements that make up the polar data receive bin

4 Conclusions and Recommendations

4.1 Conclusions

The problem statement was that the prototype vision system is too large for use in small diameter pipes. The first goal is to

> *design a front module for Pirate, housing the vision system, which is able to move through pipes as small as 57 mm, including obstacles like mitered bends and T-joints.*

In this thesis, a prototype sensor is presented that is able to pass a mitered bend inside a 57 mm pipe. The design is described in Chapter 2. The arrangement of camera and projector maximizes the base distance for optical triangulation within the available space for best sensitivity. This is achieved by using a mirror to lengthen the path of the laser beam within the short available distance. The design can be scaled down and is mostly limited by the size of the camera.

At the smallest diameter of 57 mm, a 360 degrees scan of the pipe can be done if the wide angle lens is replaced by an ultra-wide angle or fisheye lens with a minimum vertical field of view of 125 degrees.

Several problems are encountered which had to be resolved. The first problem was is issue of stray light; unwanted scattering of the light limited the sensitivity. The applied solution is to use a beam dump which converts light into heat.

During the design of the structural parts of the sensor the question arose how to precisely align the projector, camera and mirror. The solution is to place the mirror on a flexure mechanism. This mechanism is designed to have a low stiffness in the two rotational axis and a high stiffness in all other directions. This mechanism is easy to fabricate using a 3D printer.

The second goal of this research relates to the image processing algorithms. Restating the second goal from Chapter 1:

> *The software interface between the 'head' of Pirate and the control loop will be sets of data points representing the projected circle of the laser inside the pipe.*

During this research, the computational load of existing processing algorithms to extract the location of the laser beam is lowered further. The main improvement has been to use the luminance channel instead of grabbing RGB images and converting these to greyscale. Further improvements have been made by changing the interpolation for the polar mapping to a simple nearest neighbour interpolation. The output of the processing algorithm is a set of data representing the projected circle of the laser inside the pipe.

A new situation arose during this research and by means of use case analysis, some requirements were identified for the software. All requirements have been met:

- The software is implemented using a pipeline architecture. The pipeline is build using components that are reusable. (Requirement 3)
- The software for image processing is implemented on the Gumstix Overo Water Com. (Requirement 1)
- The GStreamer framework provides is a mechanism for synchronization between components. (Requirement 2)

4.2 Recommendations and Future Work

4.2.1 Image Processing

Thresholding

Thresholding is used to reduce the noise in the image. The threshold must be set high enough to eliminate most reflections. However, the threshold that is needed depends on the conditions in the pipe. Due to the weighted average algorithm, a too low threshold can result in a difference in the calculated centre and the actual centre. A higher threshold means that a smaller width of the beam is used to find the sub-pixel centre which lowers precision. A weighted average around the peak value could be investigated.

Dilation

Dilation was added (Brilman, 2011) to fill the gaps in the data. Dilation decreases the resolution and results in a diminishing ability to detect small features. It could be better to fill missing data values (gaps) in the resulting polar plot by interpolation. Moreover, this interpolation processes the data of the resulting polar plot which is much less than the two dimensional picture on which the dilation algorithm depends, this could lower the computational load.

Color Space Conversion

Detection of the laser projection uses only the luminance (Y) channel. The color space conversion to Y42B rearranges the pixel data such that the different color channels are stored sequentially in memory. To lower computational load, the undistortion map and polar map should process the YUY2 data directly.

The disadvantage is that this requires a camera that supports the YUY2 format. It is noted that this conflicts with the modularity of the software, however, many camera's use this color format.

Undistortion and Polar Mapping

The undistortion map and polar map should be calculated with higher precision before they are combined into one composite mapping. The current implementations uses two maps that are quantized to the nearest pixel. Combining both mappings when they have sub-pixel accuracy increases accuracy and should lower the quantization noise.

4.2.2 Sensor Design

Implementation on PIRATE and Miniaturization

The implementation of the sensor on PIRATE requires the design of a mechanism to direct the sensor through joints in the pipe network. Furthermore, the sensor should be radially aligned with the pipe for best performance. To accommodate space for the implementation of this mechanism, the sensor should be miniaturized. Smaller laser modules and cameras exist and allows further miniaturization of the sensor.

Determine Radial and Orientational Position

Acquiring the accurate relative pose of the system in the pipe is expected to be difficult. Using multiple concentric cones (Drost, 2009) will increase size. It could be investigated if small (ultrasonic or infrared) sensors can be used to measure the pose fo the system inside the pipe.

It could even be investigated if the undiffracted light in combination with a (one or two dimensional) positional sensitive device (PSD) can be used to to pose a constraint on the estimate of the sensors relative pose in the pipe. If either the radial or orientational position is known, the other should easily be calculated from the sensors data.

A Overo Expansion Board

For the Gumstix Overo Water COM is an expansion board developed for PIRATE ('PirateStix') at the robotics and mechatronics (RaM.) lab which has the following features:

- power over Ethernet (PoE) standard (802.3at Type2) for a maximum power usage of 25.5 W,
- rs232 to rs485 circuit with automatic transmit enable using a classic NE555 timer,
- USB2.0 port to support the camera in the structured light sensor,
- RS485 bus with a 6 V power line for the motors.

A.1 Shape

In order to keep the shape within a standard module of the robot, a stacked design was needed. The shape of the PCBs are designed such that they fit within a module as is illustrated in Figure A.1. The Ethernet connector is placed as far as could be from the back end of the module to accommodate the solid Ethernet plug within the module. Furthermore, the Ethernet socket is close to the radial centre of the module such that forces on the cable have less impact on the alignment of the robot inside the pipe. A realized PirateStix with Gumstix Overo Water COM can be seen in Figure A.2

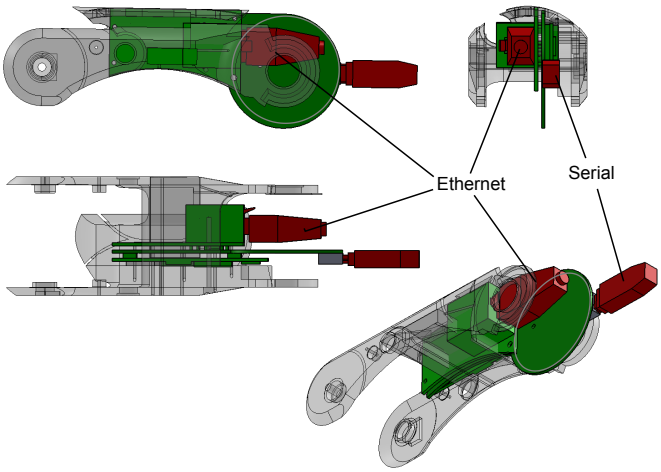


Figure A.1: Design of the Shape for the PirateStix

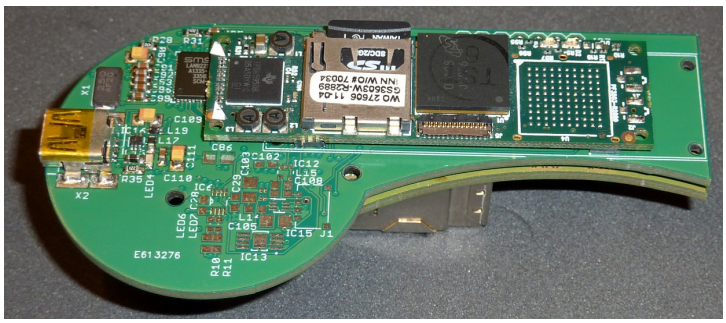


Figure A.2: Realized PirateStix

A.2 RS485

The Universal asynchronous receiver/transmitter (UART) protocol sends a start bit at the beginning of every byte. A monostable circuit (using the ancient NE555) is used to enable the RS485transmit driver.

B Hardware Design: Flexure and Beam Dump

B.1 Flexure Support with Integrated Beam Dump

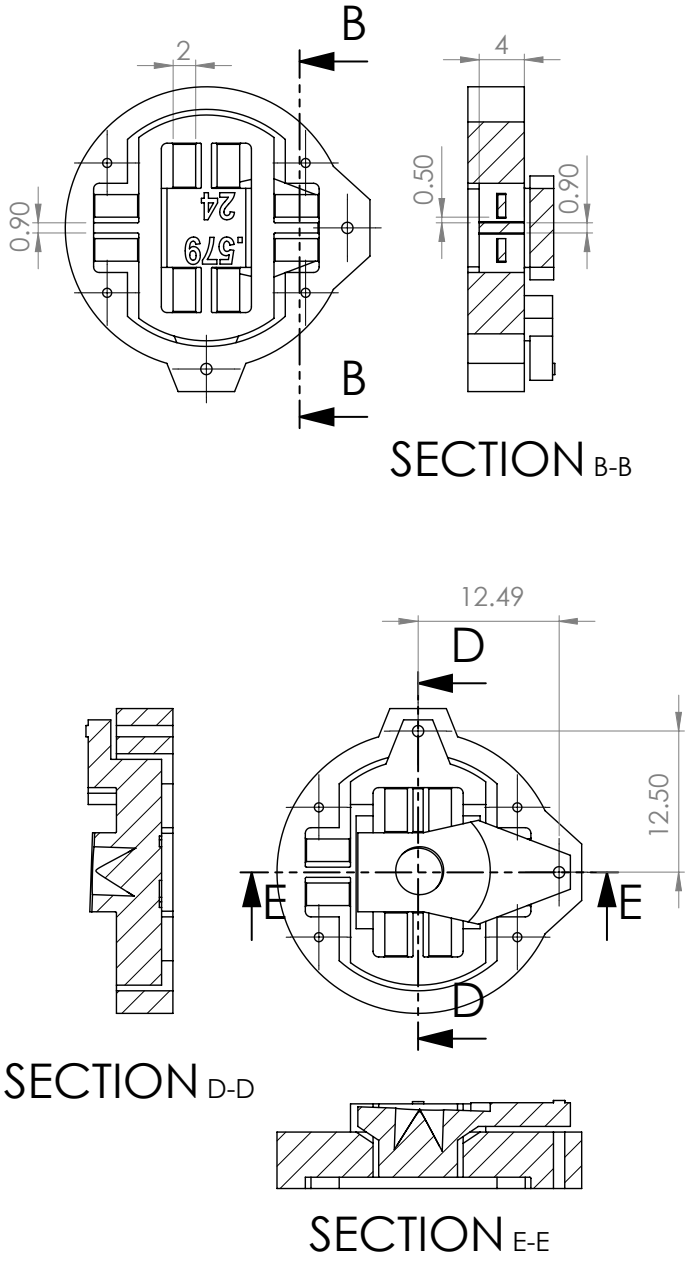


Figure B.1: Flexure support with integrated beam dump

B.2 Beam Dump

The principle of a beam dump is to convert light into heat. Reflecting the light multiple times increases efficiency. See Figure B.2.

Several prototypes were tested. The ribbed design, shown in Figure B.3, did not work as expected. The reason is that the resulting edges of the ribs of the 3D printed parts are not sharp. This causes backscattering of the laserlight. The final design that was used is shown in Figure B.4.

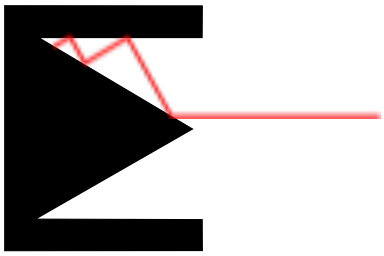


Figure B.2: Beam dump principle

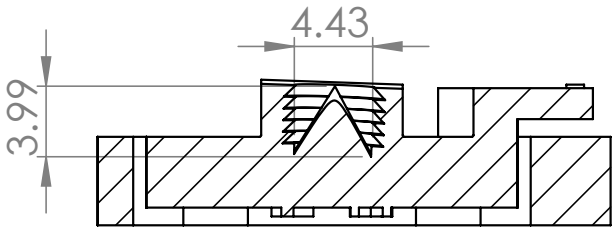


Figure B.3: Ribbed design

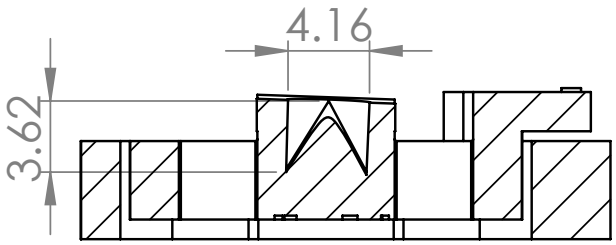


Figure B.4: Smooth design

C GStreamer RTP Pipelines

Copy and paste the given rules as one line in a terminal or create a script. Naturally, GStreamer and all required plugins should be installed.

C.1 Video pipeline

```
gst-launch -v gstrtpbin name=rtpbin latency=100 udpsrc port=5000
caps="application/x-rtp, media=(string)video, clock-rate=(int)90000,
encoding-name=(string)RAW, sampling=(string)YCbCr-4:2:0,
depth=(string)8, width=(string)640, height=(string)480,
colorimetry=(string)BT601-5" ! rtpbin.recv_rtp_sink_0 rtpbin. !
rtpvrawdepay ! ffmpegcolorspace ! ximagesink sync = false udpsrc
port=5001 ! rtpbin.recv_rtcp_sink_0 rtpbin.send_rtcp_src_0 !
udpsink port=5003 sync=false async=false
```

C.2 Test pipeline

```
gst-launch -v gstrtpbin name=rtpbin latency=100 udpsrc port=5006
caps="application/x-rtp, media=(string)video, clock-rate=(int)90000,
encoding-name=(string)RAW, sampling=(string)YCbCr-4:2:0,
depth=(string)8, width=(string)360, height=(string)180,
colorimetry=(string)BT601-5" ! rtpbin.recv_rtp_sink_1 rtpbin. !
rtpvrawdepay ! ffmpegcolorspace ! ximagesink sync = false udpsrc
port=5007 ! rtpbin.recv_rtcp_sink_1 rtpbin.send_rtcp_src_1 !
udpsink port=5009 sync=false async=false
```

D USB 1.1 Webcam

Better not to use a USB1.1 camera for the Overo COM's without a good USB2.0 hub that has proper transaction translators.

Much work has been done to get a small webcam from an endoscope (BS-15) to work with the Overo Water Com. It is found that this had possibly (several) causes. This section is copied from logs that were made during testing.

D.1 Trial and Error

A backslash \ is used to note that the line continues. This is a lab-log that might or might not be useful in future work.

```
#v4l2-ctl --set-fmt-video=width=320,height=240,pixelformat=YUYV
#gst-launch-0.10 --gst-debug-level=2 v4l2src num-buffers=1 ! \
video/x-raw-yuv,format=\(fourcc\)YUY2,width=320,height=240 ! \
ffmpegcolorspace ! jpegenc ! filesink location=test.jpg
```

With BS-15 on kernel 2.x this gives the following error:

```
Additional debug info:
gstv4l2object.c(1983): gst_v4l2_object_start_streaming ():\
/GstPipeline:pipeline0/GstV4l2Src:v4l2src0:
system error: No space left on device
```

This is probably a bug in the usb driver. The driver is not able to find an endpoint for the ISO transfer. (fifo configuration is not right?)

also dmesg | grep uvc gives:

```
uvcvideo 2-2.2:1.0: usb_probe_interface
uvcvideo 2-2.2:1.0: usb_probe_interface - got id
uvcvideo: Found UVC 1.00 device Digital_Camera (093a:2900)
usbcore: registered new interface driver uvcvideo
uvcvideo: Failed to submit URB 0 (-28).
uvcvideo: Failed to submit URB 0 (-28).
uvcvideo: Failed to submit URB 0 (-28).
```

With kernel 3.x:

```
ram@gumstix:~$ dmesg | grep uvc
[    8.828002] uvcvideo 1-2.2:1.0: usb_probe_interface
[    8.828063] uvcvideo 1-2.2:1.0: usb_probe_interface - got id
[    8.828308] uvcvideo: Found UVC 1.00 device Digital_Camera
                (093a:2900)
[    9.070312] usbcore: registered new interface driver uvcvideo
```

Seems to work, however, still can't receive images. Try BS-15 on kernel 2.x old usb (1.0) hub:

```
#gst-launch-0.10 --gst-debug-level=2 v4l2src num-buffers=1 ! \
image/jpeg,width=160,height=120 ! jpegdec ! ffmpegcolorspace \
! jpegenc ! filesink location=test.jpg
```

This works, but ... 640x480 still fails with no space error.

v4l2-ctl check supported formats for webcam:

```
#v4l2-ctl --list-formats-ext
v4l2-ctl set pixelformat and resolution:
#v4l2-ctl --set-fmt-video=width=176,height=144,pixelformat=MJPG
v4l2-ctl set framerate:
#v4l2-ctl --set-parm=1

# gst-launch-0.10 --gst-debug-level=2 v4l2src num-buffers=1 ! \
image/jpeg,width=176,height=144,framerate=1/1 ! jpegdec ! \
ffmpegcolorspace ! jpegenc ! filesink location=test.jpg
```

SUCCESS... but 176x144 is maximum resolution for BS-15 on the Gumstix Overo

D.2 OMAP3 Silicon Errata

When an USB 1.1 device is used, the EHCI controller (for USB 2.0) should detect this and pass the data to the USB 1.1 companion controller (the OHCI in the OMAP).

A problem with the USB host subsystem is mentioned in the silicon errata (TI, 2013) of the OMAP3 processor. When connected to a PHY/Transceiver all ports must use the same configuration, i.e. high speed mode or full/low speed mode.

On the Gumstix Overo the USB port is connected to a HS USB 2.0 PHY through the ULPI interface, therefore the port must be configured in high speed mode and use the EHCI controller.

D.3 Problem

USB1.1 devices interact whith the EHCI controller through a Transaction Translator (TT), which turns low- and high speed transactions into split transactions. https://www.kernel.org/doc/Documentation/usb/ehci.txt

The OMAP3530 does not have a EHCI controller with LS and FS support (through a hardware TT) It does it in software and TT scheduling seem to be problematic. the TT scheduler in the EHCI-HCD seems sub-optimal for full speed. The probably low amount of RAM (cheap) is too low for high resolution images since the peak bus speed (because of the bad scheduling) is too slow.

E First Prototype: Beam Splitter

A laser triangulation setup using a beamsplitter is tested as shown in Figure E.1.

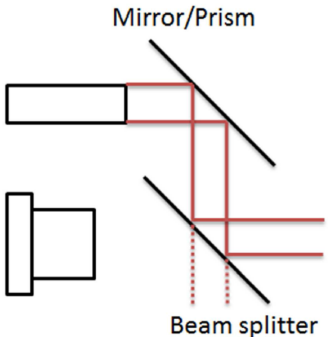


Figure E.1: Sketch of a posible arrangement using a beamsplitter (Brilman, 2011)

There were a few problems encountered with this design. The main problem was that it is sensitive to dust. Even little dust particles on the beamsplitter causes problems because the scattering is close to the camera. This partially saturated the sensitive image sensor.

Another issue is that the laser projector must be placed close to the camera. The laser cone would otherwise become too large and a larger beamsplitter would be necessary. This means that this setup has a small base distance for triangulation.

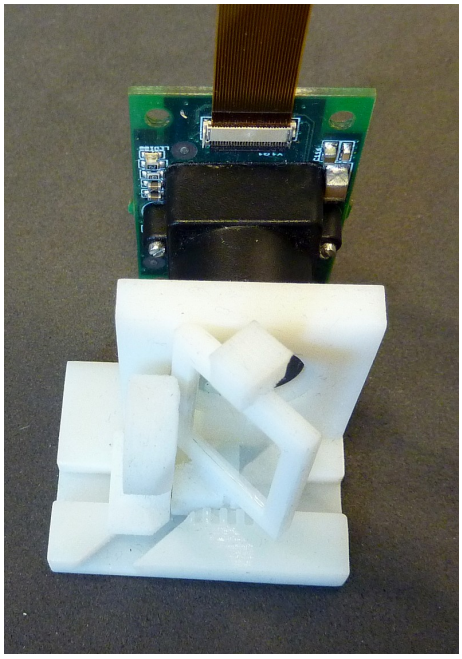


Figure E.2: Experimental setup to test the beamsplitter

F OpenCV on the Gumstix Overo COM

F.1 OpenCV patch

```
Description: Fix incorrect opencv2 include path in opencv.pc template
 pkg-config --cflags opencv showed an incorrect include path for the
 new header files.
 Could not find opencv .hpp headers during install with pkg-config.
Author: Mark Reiling

--- opencv-2.4.8/cmake/templates/opencv-XXX.pc.in
        2013-12-27 18:09:18.000000000 +0100
+++ opencv-2.4.8-fix/cmake/templates/opencv-XXX.pc.in
        2014-02-19 05:57:49.385772093 +0100
@@ -4,7 +4,7 @@
 exec_prefix=@exec_prefix@
 libdir=@libdir@
 includedir_old=@includedir@/opencv
-includedir_new=@includedir@
+includedir_new=@includedir@/opencv2

 Name: OpenCV
 Description: Open Source Computer Vision Library
```

F.2 Compile OpenCV for hardfp ABI and with NEON Support

This manual follows the OpenCV installation for Linux on (OpenCV, 2014) with a few optimizations.

On the Gumstix, follow these steps (only tested with OpenCV 2.4.8 and emdebian wheezy) but should work for other versions of OpenCV and other distros than emdebian wheezy as well:

```
:~mkdir OpenCV
:~cd OpenCV/
:~wget https://github.com/Itseez/opencv/archive/2.4.8.tar.gz
:~tar zxvf 2.4.8.tar.gz
```

Now make the patch and copy the patch to the newly created OpenCV dir, then go on:

```
:~cd opencv-2.4.8
:~patch -p1 < ../opencv-XXX.pc.in.patch
:~cd ..
:~mkdir release
:~cd release
:~cmake \
-DCMAKE_CXX_FLAGS="-march=armv7-a -mcpu=cortex-a8 -ffast-math" \
-DENABLE_NEON=ON \
-DCMAKE_BUILD_TYPE=RELEASE \
-DBUILD_FAT_JAVA_LIB=OFF \
-DBUILD_JASPER=OFF \
-DBUILD_JPEG=OFF \
-DBUILD_OPENEXR=OFF \
```

```
-DBUILD_PACKAGE=OFF \
-DBUILD_PERF_TESTS=OFF \
-DBUILD_PNG=OFF \
-DBUILD_SHARED_LIBS=OFF \
-DBUILD_TBB=OFF \
-DBUILD_TESTS=OFF \
-DBUILD_TIFF=OFF \
-DBUILD_WITH_DEBUG_INFO=OFF \
-DBUILD_ZLIB=OFF \
-DBUILD_opencv_apps=OFF \
-DBUILD_opencv_calib3d=OFF \
-DBUILD_opencv_contrib=OFF \
-DBUILD_opencv_core=ON \
-DBUILD_opencv_features2d=OFF \
-DBUILD_opencv_flann=OFF \
-DBUILD_opencv_gpu=OFF \
-DBUILD_opencv_highgui=ON \
-DBUILD_opencv_imgproc=ON \
-DBUILD_opencv_legacy=OFF \
-DBUILD_opencv_ml=OFF \
-DBUILD_opencv_nonfree=OFF \
-DBUILD_opencv_objdetect=OFF \
-DBUILD_opencv_ocl=OFF \
-DBUILD_opencv_photo=OFF \
-DBUILD_opencv_stitching=OFF \
-DBUILD_opencv_superres=OFF \
-DBUILD_opencv_ts=OFF \
-DBUILD_opencv_video=OFF \
-DBUILD_opencv_videostab=OFF \
-DWITH_1394=OFF \
-DWITH_CUBLAS=OFF \
-DWITH_CUDA=OFF \
-DWITH_CUFFT=OFF \
-DWITH_EIGEN=OFF \
-DWITH_FFMPEG=OFF \
-DWITH_GIGEAPI=OFF \
-DWITH_GSTREAMER=OFF \
-DWITH_GTK=ON \
-DWITH_JASPER=ON \
-DWITH_JPEG=ON \
-DWITH_LIBV4L=ON \
-DWITH_NVCUVID=OFF \
-DWITH_OPENCL=ON \
-DWITH_OPENCLAMDBLAS=OFF \
-DWITH_OPENCLAMDFFT=OFF \
-DWITH_OPENEXR=ON \
-DWITH_OPENGL=OFF \
-DWITH_OPENMP=OFF \
-DWITH_OPENNI=OFF \
-DWITH_PNG=ON \
-DWITH_PVAPI=OFF \
```

```
-DWITH_QT=OFF \
-DWITH_TBB=OFF \
-DWITH_TIFF=ON \
-DWITH_UNICAP=OFF \
-DWITH_V4L=ON \
-DWITH_XIMEA=OFF \
-DWITH_XINE=OFF \
-DCMAKE_INSTALL_PREFIX=/usr/local ..
```

Finish with:

```
:~make
:~sudo make install
```

Explanation for some settings that were used:

- GTK is used for GUI output,
- JPEG/JASPER is used for encode/decode jpeg/jpeg2k images,
- LIBV4L is used to grab frames from uvc webcams,
- SHARED_LIBS=OFF is to build for static libraries (= faster),

Only the three basic libs are compiled. Building more libraries is simply done, for example by turning DBUILD_opencv_calib3d=OFF to DBUILD_opencv_calib3d=ON.

F.3 Gtk-Message: Failed to load module "canberra-gtk-module"

When functions from the huighui API in OpenCV are used on the Overo COM, this error may occur. Somehow, this does not impact functionality but is annoying. Solve this issue by:

```
:~sudo apt-get install libcanberra-gtk-module
```

F.4 Compiler Optimization Flags

For best performance on a ARM Cortex-A8, use the following flags:

-O3 -march=armv7-a -mcpu=cortex-a8 -mfpu=neon -ffast-math

-O3 turns on the ftree-vectorize optimization. Vectorization allows the compiler to optimize code for the NEON SIMD (Single Instruction, Multiple Data) coprocessor. One single instruction on a SIMD processor can ad four parallel additions of 8-bit. When -O2 is used, this flag has to be set manually.

-ffast-math can be unsafe! Do not use for debug.

G Calibration

G.0.1 Camera calibration

The implemented code available on OpenCV (OpenCV dev. team, 2014) is adjusted to work with images in YUY2 colorformat. Calibratin is performed using the maximum resolution of the camera for best results. Lens distortion parameters are independent on resolution but the camera parameters like focal point distance and principal point coordinates must be scaled to the correct resolution.

G.0.2 Mirror calibration

The goal is to parrallel align the optical axis of the camera and projector. To realize this, the optical axis of the camera is aligned perpendicular to a plane and then the mirror is adjusted such that a perfect circle is projected on that plane.

Software is written, using a combination of GStreamer and OpenCV, to measure the orientation and position of the camera with respect to a plane with a calibration pattern. A topview is given in Figure G.1. As illustrated, for parallel optical alignment, the laserprojector and camera does not have to share the same principle axis. The plane is supported at three points; one pivot point and two adjustment screws. With these adjustment screws, the plane can be accurately aligned with the camera.

The four markers are spaced far apart for good accuracy and such that they do not interfere with the laser projection. The calibration pattern has a high contrast and uses circular markers. The circular markers are easy to detect and the centres can be accurately found using a least squares fitting of an ellipse.

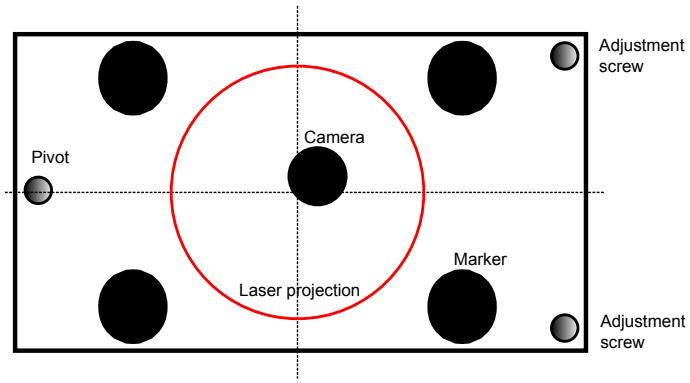


Figure G.1: Mirror calibration top view

The following steps are used to calculate the position and orienation of the camera and projection plane with respect to each other.

- undistortion
- canny edge detection
- find contours
- find marker by checking shape and size
- fit ellipse and find centres
- solve Perspective-n-Point problem

- display orientation and position

When the camera and plane are aligned, the mirror can be calibrated. It is difficult to determine the circularity of the projection with a ruler or slide caliper. Software is written to accurately measure the major and minor radii using the following steps:

- undistortion
- variable threshold using Otsu's method
- canny edge detection
- find contours
- find laser
- fit ellipse
- display major and minor radii

G.0.3 Sensor Characterization

Accurate position of the focal point of the projector and orienation of the optical axis of the projector must be found. This is achieved by measuring 'cross-cuts' of the laser cone by changing the height of camera with respect to the projection plane. The realized setup is shown in Figure G.3. See Figure G.2 for a schematic illustration. The position is then calculated by extrapolation and the orientation is determined by fitting a line through the centres.

It is noted that the accuracy of the final measurement, which is the position and orientation of the projecter with respect to the camera, is dependend on many variables. Therefore, each measurement of a 'cross-cut' is taken multiple times and all data is used in a least squares fitting algorithm.

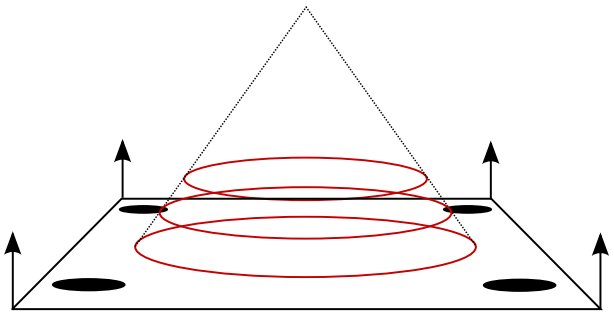


Figure G.2: Sensor characterization

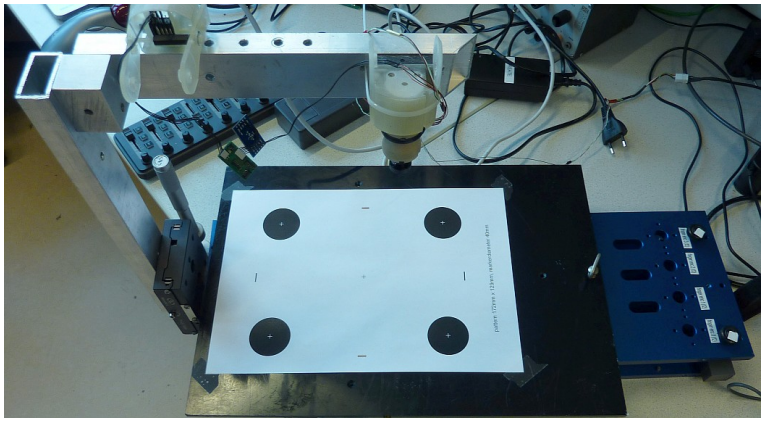
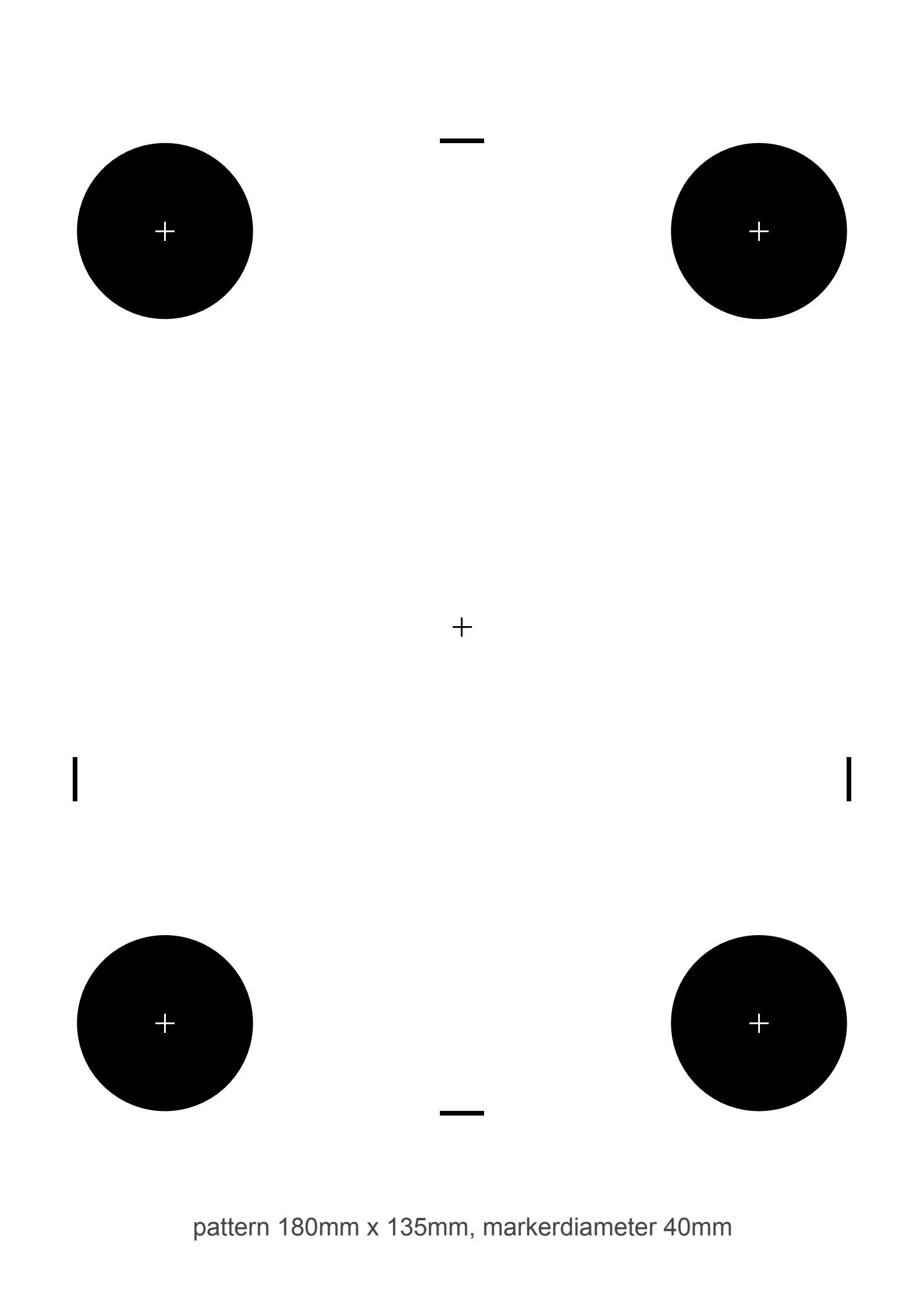


Figure G.3: Calibration setup



pattern 180mm x 135mm, markerdiameter 40mm

Bibliography

Brilman, M. F. (2011), Embedded Vision System for PIRATE.

Dertien, E. C. (2014), *Design of an inspection robot for small diameter gas distribution mains*, Ph.D. thesis, Enschede.
http://doc.utwente.nl/91336/

Doggen, C. J. M. (2010), Wireless state feedback and control of a pipe inspection robot.

Drost, E. H. (2009), Measurement system for pipe profiling.

GStreamer (2014), GStreamer.
http://www.gstreamer.com/

Gumstix (2013), Overo Water COM.
http://www.gumstix.com

Mennink, T. (2010), Self Localization of PIRATE Inside a Partially Structured Environment.

OpenCV (2014), Installation in Linux.
http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html

OpenCV dev. team (2014), Camera calibration With OpenCV.
http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

Petr Smolik, Pavel Pisa (2008), ORTE: The Open Real Time Ethernet.
http://orte.sourceforge.net/

Texas Instruments (2014), GStreamer Plug-ins for TI hardware.
http://processors.wiki.ti.com/index.php/GStreamer

TI (2013), Omap3530.
http://www.ti.com/product/omap3530

Wim Taymans, Steve Baker, Andy Wingo, Ronald S. Bultje, Stefan Kost (2014), GStreamer manual 1.4.0.
http://gstreamer.freedesktop.org/documentation/