

Range determination for mobile robots using an omnidirectional camera

Ola Millnert

February 27, 2006

Abstract

This master thesis presents a method to determine distances in a scene using only one omnidirectional camera. The algorithm will be integrated with the navigation system for a robotic wheel chair. In contrast to prior work, our method is able to build absolute scale 3D without the need of a known baseline length, traditionally acquired by odometers. Instead we use the ground plane assumption together with the camera system's height to determine the scale factor.

Using only one omnidirectional camera our method is proven to be cheaper, more reliable and more compact than the current methods for distance determination. It is cheaper since it only uses one sensor instead of having to rely on laser scanners or other expensive range detectors. It is more reliable since it can determine distances in a 3D space instead of in a plane. The experiments conducted here show promising results. The algorithm is indeed capable determine the distances in meters to features and obstacles and is able to located all major obstacles in the scene.

Contents

1	Introduction	3
1.1	Background	3
1.2	Aim of the thesis	3
1.3	Related work	4
1.4	Organisation of the thesis	6
1.5	Theoretical background	7
1.5.1	Mathematical model of the camera - the pinhole camera model	7
1.5.2	Panoramic camera	8
1.5.3	Epipolar geometry, basic properties	10
1.5.4	Calibration matrix	11
1.5.5	The Essential matrix	12
1.5.6	Epipolar geometry using hyperbolic mirrors	12
1.5.7	Basics of Kalman filtering	13
2	Methods	16
2.1	Overview of the problem	16
2.2	Camera calibration	16
2.3	Feature matching	16
2.4	Mirror projection	16
2.5	Generate and select - calculating the E-matrix	18
2.5.1	Introduction	18
2.5.2	Rank and sort correspondences	18
2.5.3	Construct sample $\hat{\mathbf{E}}_k$ -matrices	19
2.5.4	Calculating the $\hat{\mathbf{E}}$ -matrix	19
2.5.5	Choosing the best $\hat{\mathbf{E}}_k$ -matrix.	19
2.6	Retrieval of the rotation and translation	20
2.7	Determining 3D coordinates by triangulation	21
2.8	Determining the scale	22
2.8.1	Evaluating the smallest values	23
2.8.2	Determining the scale using a Kalman filter	23
3	Results	24
3.1	Input sequence	24
3.2	Feature matching	24
3.3	Mirror projection	24
3.4	Generate and select	24
3.5	Translation and rotation	28
3.6	Triangulation	30
3.7	Scaling	30
3.8	Final results	30
4	Conclusions	35

1 Introduction

1.1 Background

At K.U. Leuven the Visics group has, in cooperation with the Mechanical Engineering department, developed an automatic navigation algorithm for a robotic wheel chair, see fig. 1. The application, see [2], uses a camera system for navigation and a few ultrasound sensors together with a laser range scanner for obstacle detection. This thesis extends the application with a range determination algorithm based on the camera images. The range scanner is meant to be used as the obstacle detector and to replace the laser range sensor. The use of a laser range sensor has several drawbacks, as discussed in the next section.



Figure 1: The wheel chair setup.

1.2 Aim of the thesis

The goal of this thesis is to develop an algorithm that determines the distance to features in the scene using only one omnidirectional camera. A feature is any area in the image where there is a sharp change in intensity, for example a table leg, chair, but also a dark shadow. The viewing system is an omnidirectional camera, presented in sec. 1.5.2. An omnidirectional camera is a camera with 360° field of view. This is obtained by using a camera together with a mirror, as shown in fig. 2.

The range algorithm is meant to replace the laser range scanner the application has used for obstacle detection up to this point. The laser range scanner has a major drawback since it only allows distances in one plane to be measured. For an obstacle detection algorithm this is a great draw back. An algorithm using an omnidirectional camera is able to determine distances in the cameras entire field of view. Additionally since the camera system is already used for path finding this approach also reduces cost, the ultrasound sensors are expensive, and makes the application more compact.

The biggest challenge in developing the range detection algorithm based on only one camera comes in determining the scale in a reliable way. Relative 3D coordinates is straight forward to compute but the classical problem in computer vision is that the scale factor needed to express distances in units as meters is unknown. Other applications, see sec. 1.3, gives examples of how this can be



Figure 2: The camera setup and an image from the omnidirectional camera.

done. This thesis introduces one additional method.
 To summarise; The aim of the thesis is to create a range detecting algorithm using one omnidirectional camera.

1.3 Related work

There are several examples of work based on omnidirectional cameras. Tomas Svoboda provides a good overview of methods and theory using panoramic cameras in his Ph.D. dissertation [3]. Robot navigation, obstacle detection and range determination using omnidirectional cameras are all examples of work that incorporates some way of locating features in a 3D-space and determining their location. A survey of different computer vision approaches to range determination is given in [1]. One example of obstacle detection is the work by Koyasu, Miura and Shirai, [4]. They use a stereo camera system, two vertically aligned omnidirectional cameras, see figure 3.

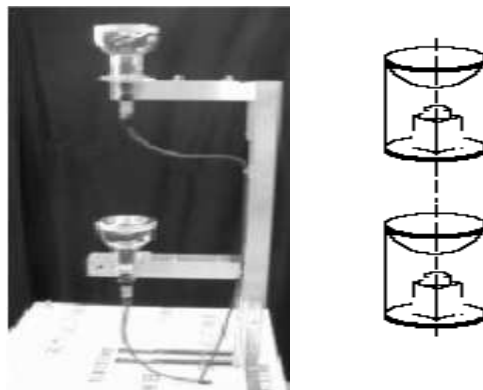


Figure 3: Image from an omnidirectional camera.

Using two cameras provides one simplification for determining the range. They have one known distance between the two mirrors and can use this to determine the scale.
 Another application where the goal is range determination has been proposed

by J.S. Chahl and M.V. Srinivasan in [15]. They compute range based on the nature of deformation in images, They use a CCD video camera positioned under a polished cone, see fig. 4. They use the fact that when the camera system

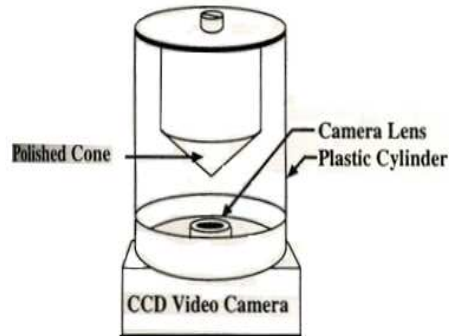


Figure 4: The vision sensor used for the range determination application.

moves the omnidirectional view deforms. In the direction of motion there is an expansion of the features, in the opposite direction a contradiction and in the perpendicular directions to the motion direction there is a plain translation of features. To determine range they move the panoramic sensor a predefined distance, h , and measure the image deformation that occurs along the azimuthal direction. They then compare this deformation to the deformation that would have occurred at a known distance. This method uses the known translation distance as reference frame, it also needs knowledge about the translation in advance. The requirements for our system is to use neither known translations nor to use a reference distances in the scene.

1.4 Organisation of the thesis

The thesis is divided into four parts;

- *Introduction*, section 1, describes the background, the aim of the thesis and covers the basic theory which the application is built upon.
- *Methods*, section 2, describes the different steps of the range algorithm, outlined in the flow chart below.
- *Results*, section 3.
- *Conclusion*, section 4.

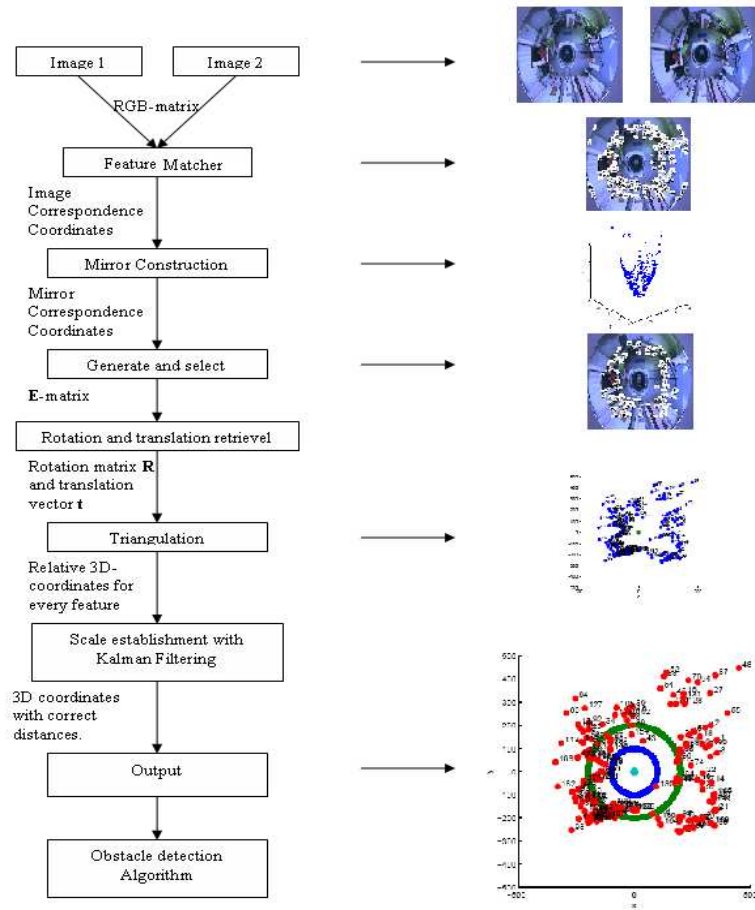


Figure 5: Outline of the Range determination algorithm.

1.5 Theoretical background

This section will give an overview of the theory which this thesis is build upon. The section starts with the basics of the camera model in 1.5.1 and the panoramic camera system in 1.5.2. Then the epipolar geometry for normal cameras is presented in 1.5.3 as well as for panoramic camera systems 1.5.6. The overview is concluded with sections on the calibration matrix in 1.5.4, the essential matrix in 1.5.5 and Kalman filtering in 1.5.7.

1.5.1 Mathematical model of the camera - the pinhole camera model

Let us start the theoretical overview with an overview of camera models and the camera system. Simplified an ordinary camera is made up of a lens system, an aperture and an image board, see fig. 6 The image board is usually a CCD-chip. It registers the incoming light and produces voltage representative to the light intensity. Based on this simplified model different mathematical models

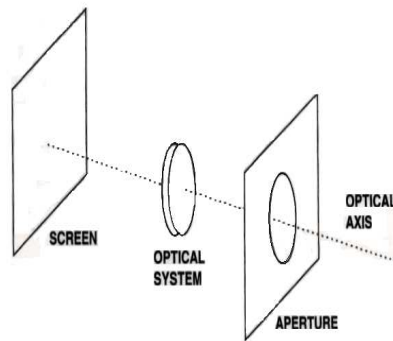


Figure 6: Simplified camera model.

are derived. The model used in this thesis, the pinhole camera model, is an approximation of the perspective projection, see eq. 1. The pinhole camera model consists of the following entities, which are all depicted in fig. 7 and listed below.

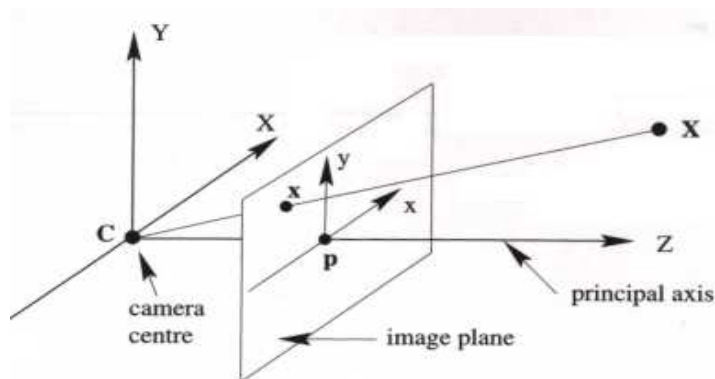


Figure 7: The pinhole camera model.

- The *Reference coordinate system* denoted as $[X\ Y\ Z]$ in figure. This is the coordinate system that defines all other coordinate systems.
- The *Image plane* is the plane where the 3D-points are projected onto. It is located at a distance f , the focal length, from the camera centre.
- The *Image plane coordinate system*, $[x\ y]$, is the reference system that defines a point on the image plane.
- The *Camera centre* C is the centre of projection. The origin of the camera's reference coordinate system.
- The *Principal axis* is the axis through the camera centre which is perpendicular with the image plane. It is aligned with the Z -axis.
- The *Focal length*, f , is the distance along the optical axis between the image plane and the camera centre.

Assume a 3D-point $P = [X, Y, Z]^T$. The projection of P onto the image plane, p , is described by

$$p = [f \frac{X}{Z}, f \frac{Y}{Z}, f]^T. \quad (1)$$

From image 8 the perspective equation 1 can be intuitively understood based on congruent triangles.

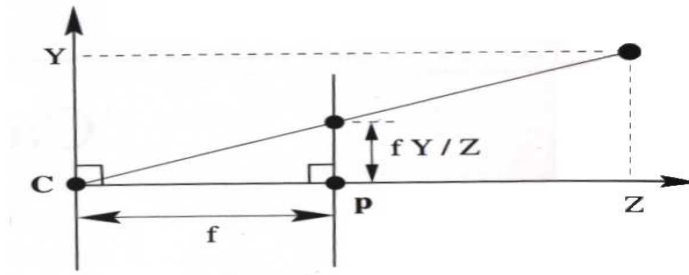


Figure 8: The congruent triangles in the pinhole camera model.

1.5.2 Panoramic camera

There are several ways to obtain a panoramic view. One is to rotate an ordinary camera round a vertical axis. This greatly reduces the frame rate and increases the reaction time. Another method that doesn't reduce speed is to have multiple cameras looking in different directions. This is an expensive and less compact option. To get both fast image acquisition and a wide field of view an ordinary video camera can be used together with a mirror. The most common combinations are an ordinary camera with one of the following mirror shapes; spherical, hyperbolic or parabolic, see figure 9. In this thesis a hyperbolic mirror is used together with a video camera. The hyperbolic shape of the mirror is essential

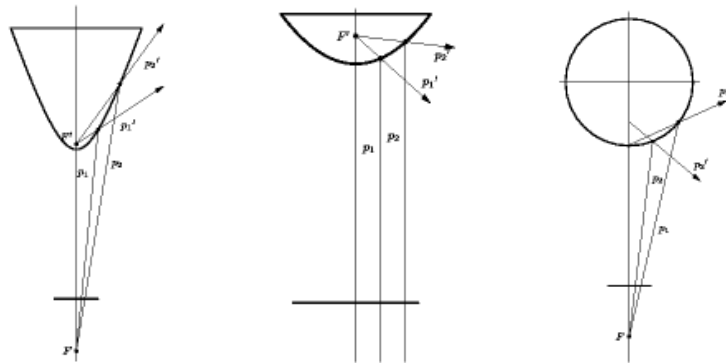


Figure 9: A hyperbolic, parabolic and spherical mirror.

if the perspective camera equations, eq. 1 are used. Among the different mirror shapes in fig. 9 the hyperbolic shape is the only one where the reflected rays intersect in the second focal point. In the figure it looks as though this would be true also for the spherical mirror shape but that is an approximation.

The hyperbolic mirror has two focal points F' and F , as can be seen in figure 10. The first focal point, F' , is where the extension of all incoming rays intersect. The second focal point, F , is where the rays intersect after they have been reflected by the mirror. The camera is mounted in the second focal point of the mirror, see fig. 10 This is, as mentioned above, very essential if the pinhole

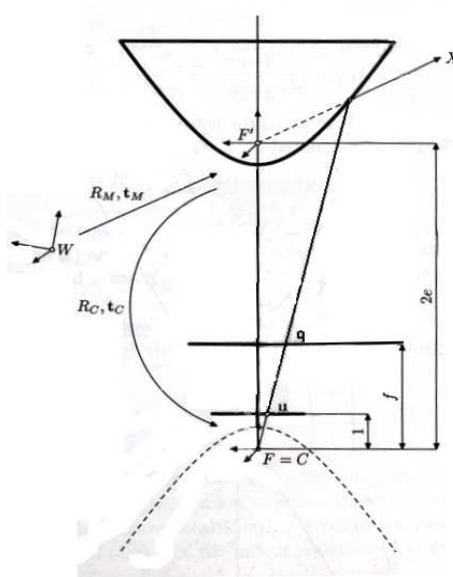


Figure 10: The epipolar geometry using ordinary cameras.

camera model is to be used, because it requires that all rays intersect in the camera centre, i.e. central projection. The shape of the hyperbolic mirror is

defined by equation,

$$\frac{(z + \sqrt{a^2 + b^2})^2}{a^2} - \frac{a^2 + b^2}{b^2} = 1, \quad (2)$$

where a,b are parameters of the mirror.

1.5.3 Epipolar geometry, basic properties

In order to process the information in two views, to search for correspondences between the views and to estimate translation and rotation between two views, a geometry known as *Epipolar geometry* has been developed. The epipolar geometry describes the geometry of relations between corresponding points in two or more views of a scene. The epipolar geometry simplifies the search for correspondences between images and is also used to calculate translation and rotation between two views. The following will give a summary of the basics of epipolar geometry. For a more extensive text refer to [5].

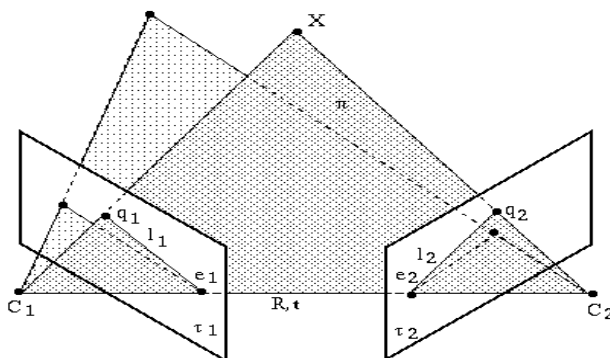


Figure 11: The epipolar geometry using panoramic cameras.

Assume a normal camera and a 3D-point \mathbf{X} in the scene, see fig. 11. There are two different views of \mathbf{X} with image centres C_1 and C_2 respectively. In each of these views there is a projection of \mathbf{X} onto the respective image plane denoted q_1 and q_2 respectively. As can be seen in fig. 11, q_1 , q_2 , \mathbf{X} and image centres C_1 , C_2 are coplanar. This coplanarity is an essential characteristic, which the epipolar geometry is built upon. The plane formed is called the *epipolar plane* and usually denoted as π . The epipolar plane intersects both images and the lines that form at the intersection between the two planes are called epipolar lines, see fig. 11, and denoted as l_1 and l_2 . The points where the epipolar lines intersect the baseline is called *epipole*, denoted as e_1 and e_2 in fig. 11. The following list summarises the basic entities and properties of the epipolar geometry,

- The *Baseline* is the line joining the camera centres of the two views. The baseline has the same direction as the translation vector \mathbf{t} .
- The *Epipolar plane* is the plane defined by the base line and the two lines joining the 3D world point \mathbf{X} with the two camera centres C_1 and C_2 .

- An *Epipolar line* is the intersection between the epipolar plane and an image plane. Denoted l_1 and l_2 in the figure.
- An *Epipole* is the point where the baseline intersects with an image plane. With the exception of the epipole, every image point is intersected by only one epipolar line. The epipole is also the projected image of the other views camera centre. The epipoles are denoted as e_1 and e_2 in the fig. 11. This is also the point where all epipolar lines intersect.
- The *Epipolar constraint* says that corresponding points must lie on conjugated epipolar lines.

As \mathbf{X} moves in space, the epipolar planes form a pencil of planes passing through the baseline. The epipolar planes in turn form a pencil of epipolar lines in both image planes which go through each epipole. This knowledge greatly reduces the search for correspondences, if the epipolar geometry is known, from the whole second image to the corresponding epipolar line.

1.5.4 Calibration matrix

In order to relate the pixel values of features to the 3D world reference frame the external and internal parameters of the camera needs to be determined. This is done with a calibration process. In [10] the external parameters, also known as the extrinsic parameters, are defined as the orientation and location of the camera reference frame with respect to the world reference frame. The external parameters are the rotation matrix \mathbf{R} and the translation matrix \mathbf{T} . The internal parameters, also known as the intrinsic parameters, are defined as the parameters that are necessary to link the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame. These parameters are the focal length f , the location of the image centre in pixel coordinates, o_x , o_y , the effective pixel size in horizontal and vertical direction s_x , s_y and the radial distortion coefficient. Having defined both the extrinsic and intrinsic parameters two matrices \mathbf{M}_{ext} and \mathbf{M}_{int} can be created. \mathbf{M}_{ext} performs the transformation between the world and the camera reference frame and \mathbf{M}_{int} performs the transformation between the camera reference and the image reference frame;

$$\mathbf{M}_{int} = \begin{pmatrix} \frac{-f}{s_x} & 0 & o_x \\ 0 & \frac{-f}{s_y} & o_y \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{M}_{ext} = \left(\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & -\mathbf{R}_1^T \mathbf{T} \\ r_{21} & r_{22} & r_{23} & -\mathbf{R}_2^T \mathbf{T} \\ r_{31} & r_{32} & r_{33} & -\mathbf{R}_3^T \mathbf{T} \end{array} \right).$$

The perspective projection, eq. 1, can then be written as,

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{M}_{int} \mathbf{M}_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = \mathbf{K} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix}. \quad (3)$$

In the equation above it can be seen that first the projection between world and camera reference frame is performed and the transformation in the camera between the pixel reference frame and image reference frame.

1.5.5 The Essential matrix

The fundamental matrix \mathbf{F} is the algebraic representation of the epipolar geometry, it relates points in two views as,

$$\mathbf{x}_r^T \mathbf{F} \mathbf{x}_l = 0, \quad (4)$$

where \mathbf{x}_r and \mathbf{x}_l is the respective coordinates of the right and left view. The fundamental matrix is used when working with uncalibrated cameras. When the cameras are calibrated, i.e. the camera calibration matrix \mathbf{K} is known, the essential matrix \mathbf{E} is used. The matrices \mathbf{F} and \mathbf{E} relate to each other by the calibration matrix \mathbf{K} as

$$\mathbf{E} = \mathbf{K}^T \mathbf{F} \mathbf{K}. \quad (5)$$

The essential matrix contains information about the translation and rotation between two views. The essential matrix \mathbf{E} is only defined up to some nonzero scale and therefore can only the direction of the translation \mathbf{t} be determined [3]. The translation vector \mathbf{t} plays a very essential part in this application. It defines the baseline of our stereo geometry.

The essential matrix can be written as the matrix product between the matrices \mathbf{R} and \mathbf{S} , i.e.

$$\mathbf{E} = \mathbf{R} \mathbf{S}$$

where

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (6)$$

and

$$\mathbf{S} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}.$$

The matrix \mathbf{R} is known as the rotation matrix and contains the rotation between the two views and the matrix \mathbf{S} contains the translation between the two views.

Having two normalised image points in two different views \mathbf{p}_r and \mathbf{p}_l in the right and left view respectively, the essential matrix satisfies the equation

$$\mathbf{p}_r^T \mathbf{E} \mathbf{p}_l = 0. \quad (7)$$

1.5.6 Epipolar geometry using hyperbolic mirrors

As our system uses a panoramic camera a few differences in the epipolar geometry are introduced. The geometry is defined by the mirror coordinates instead for the image plane coordinates.

Assume a 3D world point \mathbf{X} and two views of this with camera centres \mathbf{C}_1 , \mathbf{C}_2 and mirror focal points \mathbf{F}'_1 , \mathbf{F}'_2 . The vectors \mathbf{X}_{h1} , \mathbf{X}_{h2} joining \mathbf{X} with respective mirror focal point intersect the mirror in \mathbf{X}_{h1} and \mathbf{X}_{h2} . The baseline is between the two focal points \mathbf{F}'_1 and \mathbf{F}'_2 . The epipolar plane is spanned by the baseline, \mathbf{x}_{h1} and \mathbf{x}_{h2} , as can be seen in the fig. 12. The epipolar constraint now reads as,

$$\mathbf{X}_{h2}^T \mathbf{E} \mathbf{X}_{h1} = 0 \text{ where } \mathbf{E} = \mathbf{R} \mathbf{S}, \quad (8)$$

with \mathbf{S} and \mathbf{R} from eq. 6.

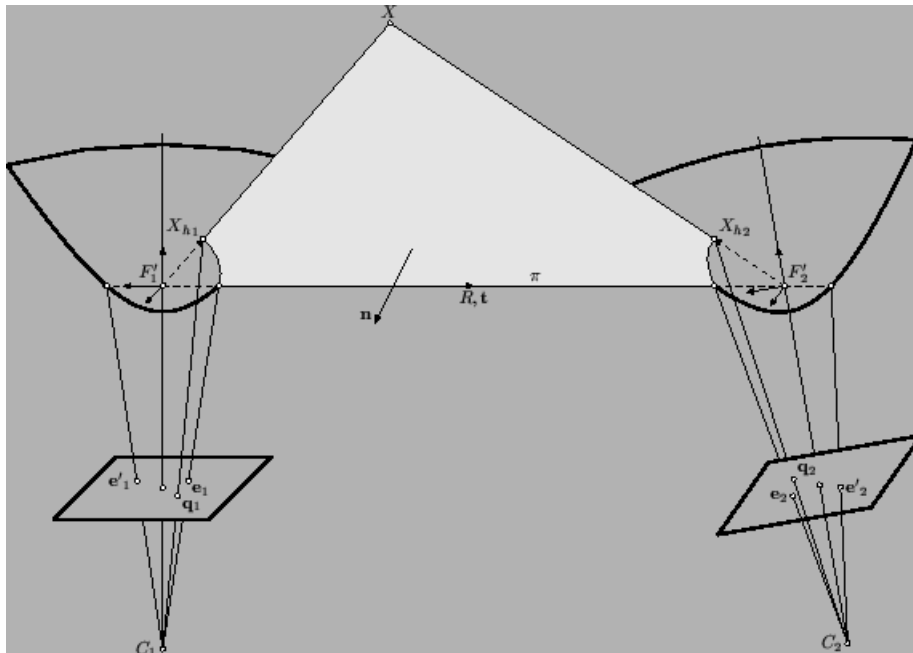


Figure 12: Epipolar geometry for panoramic cameras.

1.5.7 Basics of Kalman filtering

We continue this theory section with an overview of the Kalman filter. The Kalman filter will be used to stabilise the scale through the image sequence. The Kalman filter is used to estimate the state of a process. The estimation process can be thought of as a predict and correct cycle. First the state is predicted based on past experience and then the prediction is corrected based on a new measurement. For a more extensive text on Kalman filter, refer to [8] and [9].

The state of the process or system \mathbf{X}_k that the Kalman filter tries to estimate is a discrete-time controlled process that can be expressed by the following stochastic difference equation;

$$\mathbf{X}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad (9)$$

with measurement \mathbf{z}_k as

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \quad (10)$$

where \mathbf{v}_k and \mathbf{w}_k represent the measurement- and process-noise respectively and \mathbf{u}_k is the process input. \mathbf{H} relates the measurement \mathbf{z}_k to the present state \mathbf{x}_k .

An overview of the equations and concepts of the predict- and correct-step of the filter will now be given. Introducing $\hat{\mathbf{x}}_k^-$ as the past state estimation and $\hat{\mathbf{x}}_k$ as the present state estimation. The estimation error, of the respective state

estimations then follows as,

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-, \quad (11)$$

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k. \quad (12)$$

The estimation error covariance for both state estimations is given by,

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}], \quad (13)$$

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T]. \quad (14)$$

The equation for the estimation of the next state from the measurement \mathbf{z}_k is,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-), \quad (15)$$

where \mathbf{K}_k is the gain factor that weighs the importance of the residual, $\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-$.

The gain factor \mathbf{K}_k is chosen so it minimises the error covariance of the present state, eq. 14. For more details refer to [9]. The expression for \mathbf{K}_k is

$$\mathbf{K}_k = \frac{\mathbf{P}_k^- \mathbf{H}^T}{\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k}, \quad (16)$$

for complete derivations see [7].

From eq. 16 it can be seen that when \mathbf{R}_k , the measurement covariance error, approaches zero \mathbf{K}_k is bigger. Looking at eq. 15 this means that the residual will have a more significant impact on the state prediction. This makes sense since the measurement error is low. Also when \mathbf{P}_k^- , the estimation error covariance of the prior state, approaches zero \mathbf{K}_k will approach zero. The decrease of \mathbf{P}_k^- means that the prediction of the past state was accurate therefore it is trusted more.

To summarise the two groups of equations are given below.

$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$	(17)
$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$	(18)
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k)^{-1}$.	(19)
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-)$	(20)
$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H})\mathbf{P}_k^-$	(21)

Table 1: Predict and Correct equations.

The list below summarises the terms in the above equation;

- $\hat{\mathbf{x}}_k^-$ is the state estimate before the measurement has been made and $\hat{\mathbf{x}}_k$ is the state estimate after the measurement and \mathbf{u}_k is the systems input signal.

- \mathbf{A} is a weight function relating $\hat{\mathbf{x}}_k^-$ to $\hat{\mathbf{x}}_k$. \mathbf{B} is weight function determining the importance of the system input \mathbf{u}_k .
- \mathbf{P}_k^- is the covariance of the error function, see eq. 11, of the state before the measurement $\hat{\mathbf{x}}_k^-$.
- \mathbf{P}_k is the covariance if the error function, see eq. 12, of the state after the measurement $\hat{\mathbf{x}}_k$.
- \mathbf{Q} is the process noise covariance.
- \mathbf{K}_k is a weight function that determines the importance of the measurement \mathbf{z}_k .
- \mathbf{H} originates from eq. 10, where it relates the state to the measurement.
- \mathbf{R}_k is the process measurement noise.

2 Methods

2.1 Overview of the problem

This section contains a description of the methods used in this thesis. To be able to compute the range to all features the epipolar geometry needs to be established. This is first done by defining a stereo system, which is done by capturing two images with a small delay, the movement the robot has done during the delay is the translation defining the baseline in our stereo system. After the two images have been acquired correspondences between them needs to be found. This is done with a feature matcher. From the features the essential matrix \mathbf{E} can be computed. Then the rotation matrix \mathbf{R} and translation vector \mathbf{t} can be extracted from the essential matrix \mathbf{E} . When \mathbf{R} and \mathbf{t} have been determined the triangulation can locate all features at their respective 3D-position. Since the essential matrix only can be determined up some nonzero scale the next step is to determine that scale. This is done by using a known distance, the height of the camera, and searching for correspondences on the floor. In order to stabilise the scale a Kalman filter is employed.

2.2 Camera calibration

The first step is to calibrate the cameras. The camera was calibrated using the "Omnidirectional Calibration Toolbox Extension" which is based on the "Caltech Calibration Toolbox" by Jean-Yves Bouguet [6].

2.3 Feature matching

The tracker used to establish correspondences in views is the widely used KLT-tracker of Kanade, Lucas, Shi, and Tomasi [17]. KLT starts by identifying interest points (corners), which then are tracked in a series of images. The basic principle of KLT is that the definition of corners to be tracked is exactly the one that guarantees optimal tracking. A point is selected if the matrix

$$\begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}, \quad (22)$$

containing the partial derivatives g_x and g_y of the image intensity function over an $N \times N$ neighbourhood, has large eigenvalues. Tracking is then based on a Newton-Raphson style minimisation procedure using a purely translational model. This algorithm works surprisingly fast: we were able to track 100 feature points at 10 frames per second in 320×240 images on a 1 GHz laptop. The number of feature points to track is a parameter of the system.

2.4 Mirror projection

In sec. 1.5.6 an overview of the epipolar geometry for hyperbolic mirrors was given. In order to use these equations the features that are captured on respective views image plane needs to be back projected onto the mirror. This thesis uses the projection model of Svoboda [3], which is less general but simpler than that from Geyer and Daniilidis [16]. Figure 13 depicts the geometry of the cam-

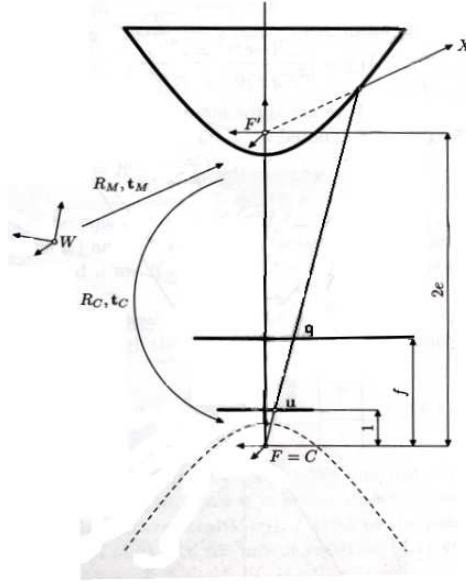


Figure 13: The epipolar geometry using ordinary cameras.

era and mirror. The equation, [3], for back projecting the image plane features onto the mirror shape is,

$$\mathbf{X}_h = \mathcal{F} \left(\mathbf{R}_C^T \mathbf{K}^{-1} \mathbf{q} \right) \mathbf{R}_C^T \mathbf{K}^{-1} \mathbf{q} + \mathbf{t}_c, \quad (23)$$

where

$$\mathcal{F} \left(\mathbf{v} = \mathbf{R}_C^T \mathbf{K}^{-1} \mathbf{q} \right) = \frac{b^2 (e v_1 + a \|\mathbf{v}\|)}{b^2 - v_1^2 - a^2 v_2^2 - a^2 v_3^2}. \quad (24)$$

- Matrix \mathbf{R}_c is the rotation matrix between the mirror reference frame and the image plane reference frame. In this application the camera and mirror are mounted so that $\mathbf{R}_c = \mathbf{I}$
- Matrix \mathbf{K} is the camera calibration matrix.
- Vector \mathbf{q} is the pixel coordinates of the feature.
- Vector \mathbf{t}_c is the translation between the mirror reference frame and the camera reference frame. It needs by construction to be $\mathbf{t}_c = [0, 0, -2e]^T$.
- Scalars a,b and e are mirror parameters.
- Vector is given by $\mathbf{v} = \mathbf{R}_C^T \mathbf{K}^{-1} \mathbf{q}$.

The camera calibration, feature matching and mirror projection were the steps required to establish the epipolar geometry. This will be done in the next section by a Method called *Generate and Select* by Tomas Svoboda [3].

2.5 Generate and select - calculating the \mathbf{E} -matrix

2.5.1 Introduction

When the features in the two images have been paired up the \mathbf{E} -matrix can be computed. As mentioned above the method used is *Generate and Select*.

- Sort and rank every correspondence based on their outlierness.
- Create sample $\hat{\mathbf{E}}$ -matrices from the correspondences, starting with the 9 best correspondences and then add the rest of the correspondences step by step.
- Choose the best $\hat{\mathbf{E}}$ -matrix based on a quality measure.

2.5.2 Rank and sort correspondences

The first step of the generate and select process is to rank the correspondences. This greatly reduces the number of iterations and correspondences needed to calculate the best \mathbf{E} -matrix. Only the combinations of the best ranked correspondences are needed to be tried.

To rank the correspondences eq. 8 is rewritten first. Introduce $\mathbf{u}_i = \mathbf{X}\mathbf{h}_{1i}$ and $\mathbf{v}_i = \mathbf{X}\mathbf{h}_{2i}$ to be the i th correspondence in the first and second mirror respectively. Then eq. 8 becomes

$$\mathbf{v}_i^T \mathbf{E} \mathbf{u}_i = 0. \quad (25)$$

Carrying out the vector multiplication and rearranging the terms results in,

$$\begin{aligned} &v_{1i}u_{1i}e_{11} + v_{1i}u_{2i}e_{12} + v_{1i}u_{3i}e_{13} + v_{2i}u_{1i}e_{21} + \\ &v_{2i}u_{2i}e_{22} + v_{2i}u_{3i}e_{23} + v_{3i}u_{1i}e_{31} + v_{3i}u_{2i}e_{32} + v_{3i}u_{3i}e_{33} = 0. \end{aligned} \quad (26)$$

The terms in eq. 26 can then be arranged in the following way,

$$\mathbf{A}\mathbf{e} = 0, \quad (27)$$

where the rows of \mathbf{A} are equal to

$$\mathbf{a}_i = [v_{1i}u_{1i}, v_{1i}u_{2i}, v_{1i}u_{3i}, v_{2i}u_{1i}, \dots, v_{3i}u_{3i}]$$

and

$$\mathbf{e} = [e_{11}, e_{12}, e_{13}, e_{21}, \dots, e_{33}]^T.$$

Generate and select uses the \mathbf{A} -matrix to rank the correspondences. When the correspondences have been ranked and sorted only the best features are needed to determine the epipolar geometry. To measure the outlierness value of the correspondences the hat matrix \mathbf{H} is used [12]. The hat matrix \mathbf{H} is given by

$$\mathbf{H} = \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T. \quad (28)$$

The idea is to measure each row's, \mathbf{a}_i in \mathbf{A} , distance from the rest of the rows in

\mathbf{A} . If the distance is large it is assumed that the correspondence is an outlier. According to [12] the hat matrix is a way of measuring that distance. The outlierness measure o_i for the i th row, i.e the i th correspondence, is obtained from the diagonal element in the i th row,

$$o_i = h_{ii} = a_i^T (\mathbf{A}^T \mathbf{A})^{-1} a_i, \quad (29)$$

where all h_{ii} s are diagonal terms of \mathbf{H} . The correspondences are then sorted in ascending order based on their outlierness value, o_i .

The next step is to generate sample $\hat{\mathbf{E}}$ -matrices from an increasing numbers of rows in \mathbf{A} , starting with the 9 best features.

2.5.3 Construct sample $\hat{\mathbf{E}}_k$ -matrices

In the section above the correspondences were ranked and sorted based on their outlierness measure o_i . The idea is that the best paired correspondences have the lowest o_i values and are sorted in to the top rows of the \mathbf{A} -matrix. The sample $\hat{\mathbf{E}}_k$ -matrices are calculated from sample sets \mathbf{S}_k , which are generated from the \mathbf{A} -matrix. The first sample set \mathbf{S}_0 contains the top 9 features of \mathbf{A} and each subsequent set \mathbf{S}_{k+1} is the union between \mathbf{S}_k and the next still not used row in \mathbf{A} . There can be maximally $N - 9$ number of sample sets \mathbf{S}_k , where N is the total number of correspondences.

This method will generate better sample $\hat{\mathbf{E}}_k$ -matrices for every feature added until the first row that contains an outlier is added. The next step is to construct sample $\hat{\mathbf{E}}_k$ -matrices and evaluate them based on different quality measures. When the \mathbf{E} -matrix is determined the translation vector \mathbf{t} and rotation matrix \mathbf{R} can be extracted.

2.5.4 Calculating the $\hat{\mathbf{E}}$ -matrix

In the previous section sample sets \mathbf{S}_k were created from the ranked correspondences. The \mathbf{E} -matrix is calculated for every correspondence set by a method known as the *8-point algorithm* introduced by Longuet-Higgins [13].

Rewriting eq. 27, with \mathbf{S}_k instead of \mathbf{A} gives,

$$\mathbf{S}_k \mathbf{e} = 0. \quad (30)$$

The sample $\hat{\mathbf{E}}$ is then, with the *8-point algorithm*, computed by taking the singular value decomposition of \mathbf{S}_k . It should be noted that eq. 30 only has a nontrivial solution if \mathbf{S}_k is singular and \mathbf{e} then lies in the null space of \mathbf{A} [3]. This is the reason why \mathbf{e} only can be recovered up to a non-zero scale. When the singular value decomposition has been taken of \mathbf{A} , the solution to \mathbf{e} is given by the right singular vector corresponding to the smallest singular value [14].

2.5.5 Choosing the best $\hat{\mathbf{E}}_k$ -matrix.

The different $\hat{\mathbf{E}}_k$ -matrices has now been calculated. It now remains to select the best one. This can be done by comparing the the residual,

$$\mathbf{v}_i^T \mathbf{E}_k \mathbf{u}_i = \mathbf{r}_i, \quad (31)$$

but this is according to [3], not a robust enough measurement. The reason for this is that there are often some \mathbf{E} -matrices that produce small residuals for

both good correspondences and outliers. Instead another method is used to measure the quality of the \mathbf{E} -matrix proposed by [12]. This method uses the variance of the residuals of the correspondences to evaluate the quality.

The first step of the selection process is to evaluate the correspondences by its residual r_i and compare it to a threshold value,

$$|r_i| < 2.5 \left(1.4826 \left(1 + \frac{5}{N-p} \right) \right) \sqrt{\text{med}(r_i^2)}, \quad (32)$$

where N is the number of features and $p=8$, the number of parameters estimated. For those correspondences that remain after eq. 32 the variance v_i of the r_i s is calculated. The variance v_i for that $\hat{\mathbf{E}}_k$ -matrix is then also compared to a threshold value,

$$v_i < 2.5 \left(1.4826 \left(1 + \frac{5}{N-p} \right) \right) \text{med}(v_i). \quad (33)$$

Among those $\hat{\mathbf{E}}_k$ -matrices that pass under the threshold value the $\hat{\mathbf{E}}_k$ with the lowest q_E value is chosen. The q_E value is the difference between the two biggest singular values of $\hat{\mathbf{E}}_k$,

$$q_E = \sigma_1 - \sigma_2, \text{ where } \mathbf{D} = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \text{ and } \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{E}. \quad (34)$$

If the essential matrix was perfect the difference would be zero.

The next step is to extract the rotation matrix \mathbf{R} and translation vector \mathbf{t} from the \mathbf{E} -matrix.

2.6 Retrieval of the rotation and translation

In order to determine the features 3D-position by triangulation the second views reference frame must be defined in the first views reference frame. The two views reference frames relate to each other by

$$\mathbf{C}_1 \cdot \mathbf{R} + \mathbf{t} = \mathbf{C}_2, \quad (35)$$

where \mathbf{R} is the rotation matrix and \mathbf{t} is the translation vector, as seen in the figure. \mathbf{R} and \mathbf{t} are determined by extracting from the \mathbf{E} -matrix. In sec. 1.5.6 it was stated that $\mathbf{E} = \mathbf{R}\mathbf{S}$, with

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \text{ and } \mathbf{S} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}. \quad (36)$$

To compute \mathbf{R} and \mathbf{S} a method using singular value decomposition introduced by Hartley in [11] has been used. The decomposition is as follows,

$$\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^T. \quad (37)$$

Then \mathbf{R} and \mathbf{S} are computed by,

$$\mathbf{S} = \pm \mathbf{V}\mathbf{Z}\mathbf{V}^T \text{ and } \mathbf{R}_1 = \mathbf{U}\mathbf{Y}\mathbf{V}^T \text{ or } \mathbf{R}_2 = \mathbf{U}\mathbf{Y}^T\mathbf{V}^T, \quad (38)$$

where

$$\mathbf{Z} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ and } \mathbf{Y} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (39)$$

This gives four different combinations for the correct \mathbf{R} and \mathbf{t} .

1. $\mathbf{R1}$ and \mathbf{t}
2. $\mathbf{R1}$ and $-\mathbf{t}$
3. $\mathbf{R2}$ and \mathbf{t}
4. $\mathbf{R2}$ and $-\mathbf{t}$

The correct combination will be determined first after the triangulation, which will be covered in the section below.

2.7 Determining 3D coordinates by triangulation

The triangulation determines where the features are situated in the 3D-space. The name comes from the fact that the method uses triangles to determine the features location. The positions are determined by finding where two rays, back projected from each image centre going through respective image correspondence, intersect see fig. 14. The triangulation formula derived in this thesis is

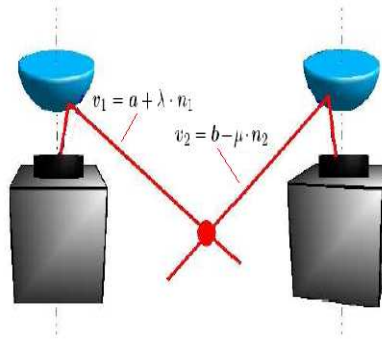


Figure 14: The feature is located at the intersection at the intersection between the two rays v_1 and v_2 .

based on the notion that the smallest distance between two rays has a direction perpendicular to both rays. Assume two rays $\mathbf{v}_1 = \mathbf{a} + \lambda \cdot \mathbf{n}_1$ and $\mathbf{v}_2 = \mathbf{b} + \mu \cdot \mathbf{n}_2$, where \mathbf{a} and \mathbf{b} are the respective centre of projections in the two views and λ and μ are scale factors. \mathbf{n}_1 and \mathbf{n}_2 are the two rays respective direction, as shown in fig. 14.

The shortest distance between \mathbf{n}_1 and \mathbf{n}_2 , noted d is given by,

$$\mathbf{d} = (\mathbf{b} - \mathbf{a}) \cdot \frac{\mathbf{n}_1 \times \mathbf{n}_2}{\|\mathbf{n}_1 \times \mathbf{n}_2\|}, \text{ where } \mathbf{n}_3 = \frac{\mathbf{n}_1 \times \mathbf{n}_2}{\|\mathbf{n}_1 \times \mathbf{n}_2\|}.$$

Knowing that $\mathbf{v}_1 + \mathbf{d} \cdot \mathbf{n}_3 = \mathbf{v}_2$ a formula for λ and μ can be derived. The derivation of λ is shown below,

$$\begin{aligned}
\mathbf{a} + \lambda \cdot \mathbf{n}_1 + \mathbf{d} \cdot \mathbf{n}_3 &= \mathbf{b} + \mu \cdot \mathbf{n}_2 \Rightarrow \\
(\mathbf{a} + \lambda \cdot \mathbf{n}_1 + \mathbf{d} \cdot \mathbf{n}_3) \times \mathbf{n}_2 &= (\mathbf{b} + \mu \cdot \mathbf{n}_2) \times \mathbf{n}_2 \Rightarrow \\
\mathbf{a} \times \mathbf{n}_2 + \lambda \cdot \mathbf{n}_1 \times \mathbf{n}_2 + \mathbf{d} \cdot \mathbf{n}_3 \times \mathbf{n}_2 &= \mathbf{b} \times \mathbf{n}_2 + \mu \cdot \mathbf{n}_2 \times \mathbf{n}_2 \Rightarrow \\
\mathbf{a} \times \mathbf{n}_2 + \lambda \cdot \mathbf{n}_1 \times \mathbf{n}_2 + \mathbf{d} \cdot \mathbf{n}_3 \times \mathbf{n}_2 &= \mathbf{b} \times \mathbf{n}_2 \Rightarrow \\
\lambda \cdot \mathbf{n}_1 \times \mathbf{n}_2 &= \mathbf{b} \times \mathbf{n}_2 - \mathbf{a} \times \mathbf{n}_2 - \mathbf{d} \cdot \mathbf{n}_3 \times \mathbf{n}_2,
\end{aligned}$$

rewriting the vector on the right hand side as \mathbf{n}_r and the vector on the left hand side as \mathbf{n}_l the equation becomes,

$$\lambda \cdot \mathbf{n}_l = \mathbf{n}_r.$$

Then it is clearly seen that the two vectors \mathbf{n}_l and \mathbf{n}_r have the same direction and only differ by the scale factor λ . λ is now given by,

$$\lambda = \frac{\mathbf{n}_r[x]}{\mathbf{n}_l[x]} = \frac{\mathbf{n}_r[y]}{\mathbf{n}_l[y]} = \frac{\mathbf{n}_r[z]}{\mathbf{n}_l[z]}.$$

μ is derived with the analog steps.

When λ and μ have been determined for all four combinations mentioned in sec. 2.6 the right combination needs to be determined. Depending on how the reference frames are oriented, how R and t are chosen, four different combinations for μ and λ can occur,

1. $\lambda > 0$ and $\mu > 0$,
2. $\lambda > 0$ and $\mu < 0$,
3. $\lambda < 0$ and $\mu > 0$,
4. $\lambda < 0$ and $\mu < 0$.

The correct one is when they are both positive. This is because the intersection must take place in front of the cameras and not behind.

The last step in the range determination is to establish the actual range to all features. To do this the scale must be determined. This is covered in the next section.

2.8 Determining the scale

In the previous section all features relative 3D-coordinates were determined. To determine their absolute distance the scale factor needs to be established. This is done by using the ground plane assumption, assuming that the robot is moving in one plane, and the height of the camera system. More precisely it is the distance between the ground plane and the first focal point of the mirror that is used. The relative positions are scaled to their absolute positions.

2.8.1 Evaluating the smallest values

To compute a coherent set of ground points an evaluation step is needed. The first step to select the k lowest relative features. Some of these features might be outliers, a feature might be higher or lower than the ground plane, therefore it is needed to filter these out. The evaluation process then computes the Euclidean distance in the z -coordinate between each feature and the average of the k features. If the distance is bigger than a threshold t the feature is discarded. This step is then repeated for the $k-1$ remaining features and is iterated until all features are within the predefined threshold t or we have a minimum set of features. This evaluation step prevents features that due to noise, bad triangulation etc. affect the scale. To have a robust scale through the images a Kalman filter is used. This is described in the next section.

2.8.2 Determining the scale using a Kalman filter

Since the scale is very essential and there is great risk it may fluctuate greatly a Kalman filter is used to stabilise the scale. Section [8] covered the basics of the Kalman filter, eqs. 40-44, reviews the equations controlling the process.

For a summary of the terms refer to the list at the end of sec. 1.5.7.

$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \quad (40)$
$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (41)$
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k)^{-1}. \quad (42)$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \quad (43)$
$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (44)$

Table 2: Predict and Correct equations.

To define the kalman filter the parameters needs to be set. The list below explains the values chosen;

- The scalar x is the scale of that is being filtered.
- The scalar $A = 1$, the state of the process is unchanged through the different images.
- The scalar B is not assigned to any value since there is no input signal to the system.
- The scalar Q is set to $1e^{-5}$, assuming a small process invariance.
- The scalar R , different R 's was tried and the results will be seen under Results.
- The scalar x_0 was initially set to 20, which was the average for this specific set of test images.
- The scalar P_0 was initially set to 0, according to [7] this is not a crucial choice since the filter will converge anyway.

3 Results

In this section the results obtained with the previously describes methods will be presented. Section 3.1 explains the input sequence which the algorithm was tested upon. Section 3.2 shows the result of the feature matching and sec. 3.3 presents the results of the mirror projection. Then the result from the Generate and Select process is shown followed by the translation and rotation vector which is extracted from the Essential matrix \mathbf{E} . The section is concluded with the results from the triangulation and scale determination process.

3.1 Input sequence

The algorithm is tested on a video sequence captured when the camera was moved in a straight line in the camera's Y-axis direction. The typical example used the algorithm is tested on is two images from this sequence. The images used are shown in fig. 15.

Then the whole video sequence is used to test the variation of the distances as the camera moves and how consistent the scaling and feature matching is.

3.2 Feature matching

The determination and matching of features in two views is, as said before, the foundation which the epipolar geometry is build upon. It is therefore important to have a high amount of qualitative correspondences. This is a function of the translation, rotation and it is also dependent on the richness of feature points in the scene. The graph below shows how the number of features varies as a function of the delay between two images. When the delay is greater the baseline will also be greater, as the robot is moving at a constant velocity.

Figure 15 shows the two input images and fig. 17 shows the correspondences between them.

As can be seen in fig. 17 there are a high amount of correspondences covering all important features.

3.3 Mirror projection

In sec. 2.4 the method for mirror projection was given. The mirror projection is needed to use the epipolar geometry using the hyperbolic mirror. Figure 18 shows the results after the tracked features in fig. 17 have been back projected onto the mirrors. The hyperbolic shape of the mirrors is clearly seen.

3.4 Generate and select

The generate and select process calculates the \mathbf{E} -matrix by generating a number of $\hat{\mathbf{E}}$ -matrices, see sec. 2.5. The $\hat{\mathbf{E}}$ -matrices are evaluated with different measures and finally the best one is chosen.

The first step in the process is to sort the correspondences based on their "outlierness measure", see eq. 29. To measure the outlierness the hat-matrix based on \mathbf{A} -matrix is used, see sec. 2.5.2. Figure 19 shows the sorted outlierness values.

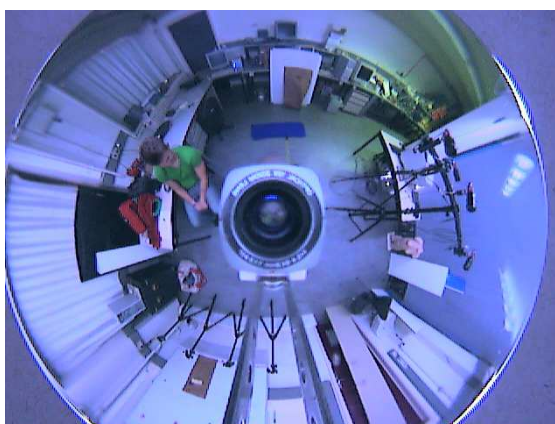


Figure 15: The two input images.

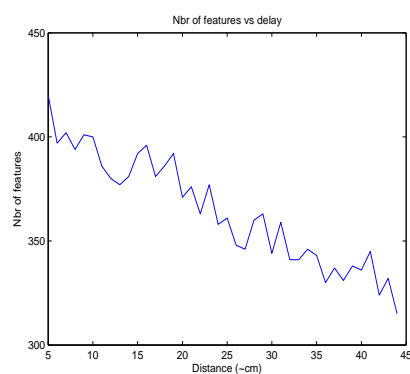


Figure 16: Number of features as a function of the delay.

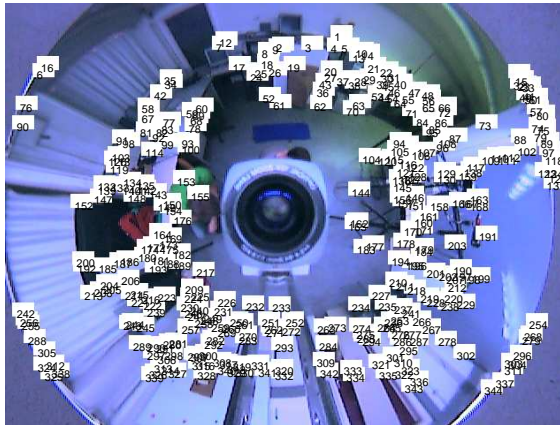


Figure 17: The correspondences found in each of the images.

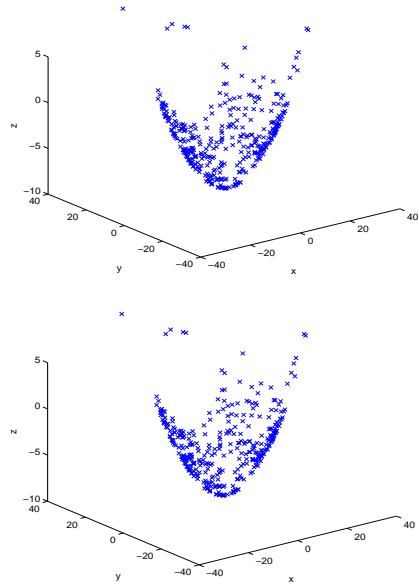


Figure 18: The features back projected onto the mirrors.

After the outlieriness of every feature have been determined the generation of

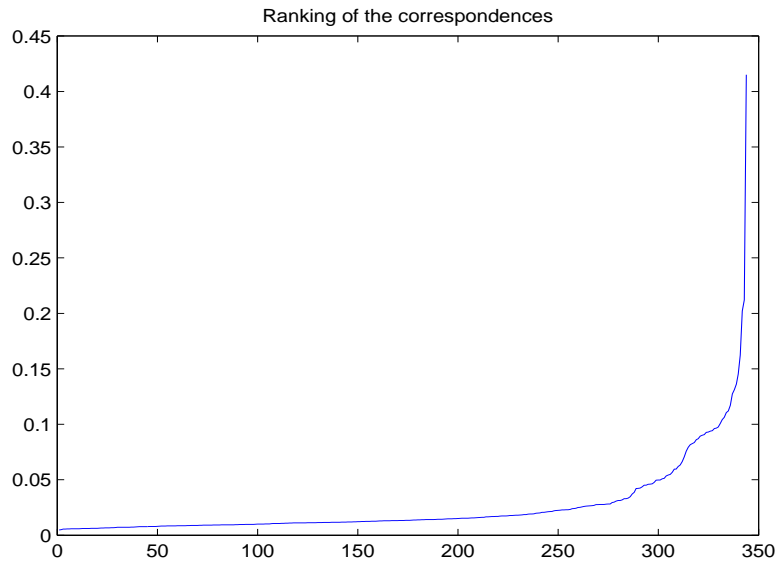


Figure 19: The ranking of the correspondences.

\hat{E} -matrices can begin. Totally $N-9$ \hat{E} -matrices will be generated. Each of them

is evaluated and the one with best q_E -value is chosen. The q_E -value is given by,

$$q_E = \sigma_1 - \sigma_2, \text{ where } \mathbf{D} = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \text{ and } \mathbf{UDV}^T = \mathbf{E}. \quad (45)$$

The q_E -value varies with each added feature, which fig. 20 shows.

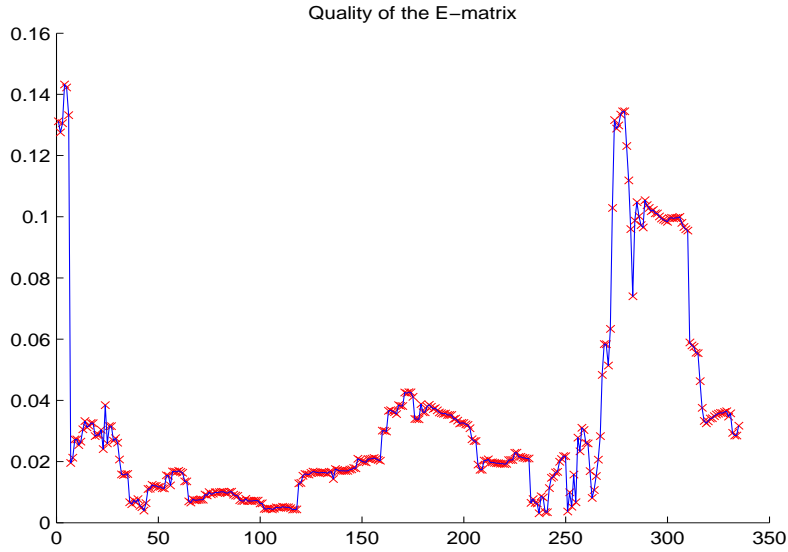


Figure 20: The quality of the $\hat{\mathbf{E}}$ -matrix.

Figure 20 shows the q_E values, i.e. the quality of the $\hat{\mathbf{E}}$ -matrix. In the figure can be seen that it is fluctuating but still good for every feature added until the first outliers are included. The $\hat{\mathbf{E}}$ -matrix with the lowest q_E is then chosen as the \mathbf{E} -matrix as long it's variance v_i passes under the threshold, eq. 33. After Generate and Select the suspected features is reduced. The result is shown in fig. 21

3.5 Translation and rotation

In sec. 2.6 methods for calculating the translation and rotation between the two views was given. The rotation matrices \mathbf{R} and \mathbf{S} is computed from the \mathbf{E} -matrix with equations 37 - 39. The translation vector \mathbf{t} is then given from \mathbf{S} by,

$$\mathbf{S} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}. \quad (46)$$

In the tests, fig. 15, the camera is moved along the Y-axis of the camera's coordinate system with no rotation. The \mathbf{S} -matrix and rotation matrix \mathbf{R} , which are calculated from the \mathbf{E} -matrix, get values,

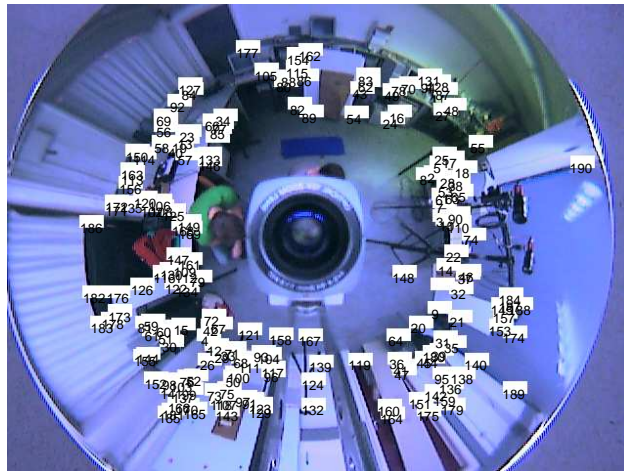
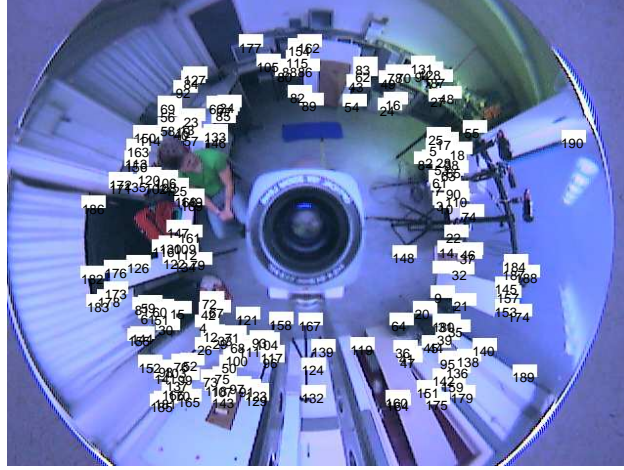


Figure 21: The images with the suspected outliers filtered out.

$$S = \begin{pmatrix} 0 & -0.02624 & 0.9995 \\ 0.0262 & 0 & -0.0148 \\ -0.9995 & 0.0148 & 0 \end{pmatrix} \quad R = \begin{pmatrix} 0.9995 & -0.0099 & 0.0008 \\ 0.0099 & 0.9999 & -0.0027 \\ -0.0009 & 0.0027 & 0.99999 \end{pmatrix}.$$

The translation vector then becomes,

$$\mathbf{t} = \pm \begin{pmatrix} 0.0148 \\ 0.9995 \\ 0.0262 \end{pmatrix}.$$

An interesting test would of course be to rotate the camera and then see if the correct rotation was measured. The current algorithm works well for small rotations. For greater rotations the feature matcher is not suited, it has therefore problems following the features. This is not an intrinsic problem. The best solution would be to use a different tracker, even though it might be slower. Modifications are suggested in the conclusion section.

The next section shows the triangulation for the different combinations.

3.6 Triangulation

The prior section gave four combinations of translation vector and rotation matrix. The triangulation for each of those were carried out and tested. The respective result is given in fig. 22. It is clearly seen that the two upper images are much better than the lower ones. The triangulation resulting in the upper images use the correct rotation matrix. The right upper figure shows the correct triangulation result. The difference between the two upper images is the sign of the translation vector \mathbf{t} . The upper left image is the right image mirrored in the origin.

Figure 23 shows the correct result again.

3.7 Scaling

To test the consistency of the scaling between the images the whole image sequence is used. Since the scaling is used to determine the distances in units as meters it is important it is consistent. Figure 24 shows how the scale varies with different images.

3.8 Final results

All the pieces have now been put together and the distance to the tracked features can now be estimated in meters. Figure 25 shows the final result from different angles. It can clearly be seen that the geometry of the features correspond to that in fig. 15. The green circle indicates two meters and the blue one meter.

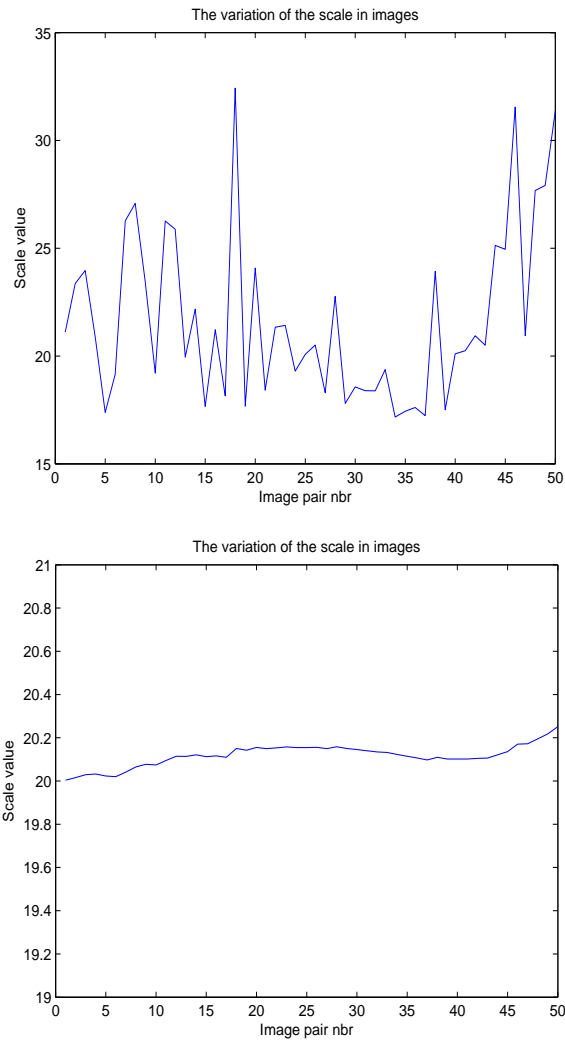


Figure 24: Top: The variation of the scale between images. Bottom: The scale after employing a Kalman Filter

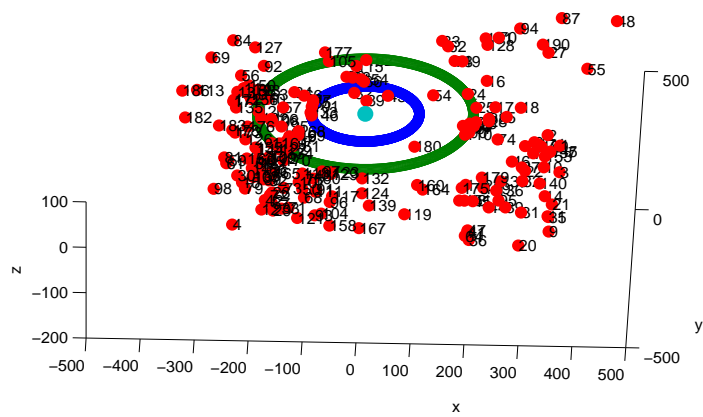
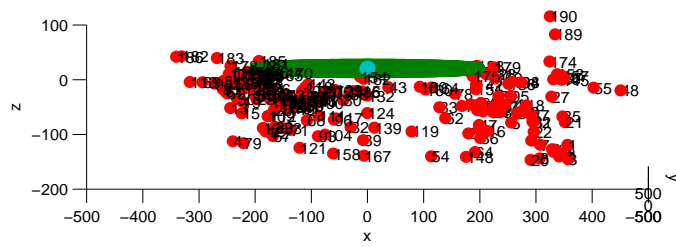
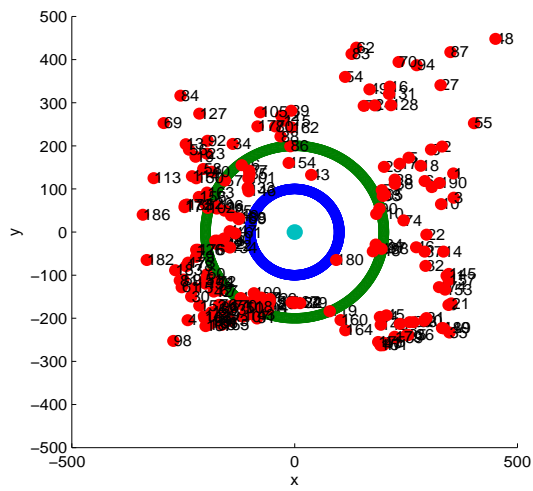


Figure 25: Different views of the final result.

4 Conclusions

Based on the results presented in this thesis we conclude the method using only one camera as a sensor in robot navigation is promising. The results are a good indication that this algorithm will be able to give greatly more distance information than the laser range scanner.

The contributions given in this work are; The thesis has showed that it is possible to determine distances in two views captured by one camera with only prior knowledge about the camera's height. This thesis also derived a successful triangulation formula and employed a Kalman filter in order to obtain a consistent scale, even though the features vary greatly between images. For the algorithm to be successfully used in more general applications a few items need to be refined;

- First, more features need to be detected in order to have more correspondences. This would in turn produce a more reliable epipolar geometry and enable more distances to be computed.
- Second, more features would enable a different method used to track the features between images. In the present algorithm the features are captured in every image pair. If it would be possible to follow features in images the movement of the robot would not be limited to slow turns and slight rotations. This is of course essential to work out in future work.
- Finally, a more sophisticated method for tracking the lowest point could be developed. If the features would be followed through the images the lowest values would be more consistent through out the images.

Acknowledgements

I thank my supervisor Toon Goedemé for his guidance, patience and insightful suggestions. I thank my supervisor prof. Kalle Åström at Lund's Institute of Technology for his comments, great care in reading through the material and his guidance of the work. Without my promoter prof. Luc Van Gool this thesis would never have been materialised.

References

- [1] R.A Jarvis 1983 *A perspective on Range Finding Techniques for Computer Vision*. Transaction on pattern analysis and machine intelligence, VOL. PAMI-5, No. 2, March 1983
- [2] T. Goedemé, M. Nuttin, T. Tuytelaars, L. Van Gool, *Omnidirectional Vision based Topological Navigation*, 15th International Symposium on Measurement and Control in Robotics, ISMCR 2005, 2005.
- [3] Tomas Svoboda, *Central Panoramic Cameras Design, Geometry, Egomotion*. Ph.D. dissertation at Centre for Machine perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University. Sept 1999.
- [4] Hiroshi Koyasu, Jun Miura and Yoshiaki Shirai. *Realtime Omnidirectional Stereo for Obstacle Detection and Tracking in Dynamic Environments*. Proc. 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems pp. 31-36, Maui, Hawaii, Oct./Nov. 2001.
- [5] Richard Hartley and Andrew Zisserman. *Multiple View Geometry*, Cambridge University Press, ISBN 0521540518, 2004.
- [6] Christoffer Mei URL: <http://www-sop.inria.fr/icare/personnel/..Christopher.Mei/index.html>, cited 2006/12/10.
- [7] Greg Welch and Gary Bishop, *An introduction of the Kalman filter* TR 95-041 Department of Computer Science, University of North Carolina at Chapel Hill, April 2005
- [8] Kalman R.E. *A new approach to linear filtering and prediction problems*, transaction of the ASME - Journal of Basic Engineering, pp. 35-45, March 1960.
- [9] Jacobs, O.L.R., *Introduction to Control Theory*, 2nd Edition, Oxford University Press, 1993.
- [10] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, Prentice Hall, Inc. ISBN 0-13-261108-2, 1998.
- [11] Richard I. Hartley, *Estimation of relative camera positions for uncalibrated cameras*, in 2nd European Conference on Computer vision, pp. 579 - 587, Springer - Verlag, LNCS 588, May 1992.
- [12] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. John Willey and Sons, 1987.
- [13] H.C. Longuet-Higgins. *A computer algorithm for reconstructing a scene from two projections*. Nature, 293:133 - 135, 1981.
- [14] Sabine Van Huffel and Joos Vandewalle. *The total Least Squares Problem: Computational Aspects and Analysis*, volume 9 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia. 1991.

- [15] J.s. Chahl and M.V. Srinivasan, *Range estimation with a panoramic visual sensor*. 2144 J. Opt. Soc. Am. A/Vol. 14, No. 9/Sept 1997.
- [16] C. Geyer and K. Daniilidis, *Mirrors in motion: Epipolar geometry and motion estimation*, ICCV, p. 766, Nice, 2003.
- [17] J. Shi and C. Tomasi *Good Features to Track*, Computer Vision and Pattern Recognition, Seattle, pp. 593-600, 1994