UNIVERSITY OF AGDER

Development of a Vision-Based Measurement System for Relative Motion Compensation

Johan Lindal Haug

Supervisors

Geir Hovland
Morten Ottestad

This master's thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

University of Agder, 2014
Faculty of Engineering and Science
Department of Engineering

Abstract

Motion compensation is used in many applications where no relative motion between two objects is desired. In the offshore industry, motion compensation systems are used to increase the operational window during load transfers like crane operations or positioning of gangways for personell transfer. In most systems one or more motion reference units (MRU) are used to measure motion with respect to a world reference frame and compensate for this motion. For motion compensation between two independently moving objects, a MRU-based solution becomes complex when measurements must be exchanged and combined. Vision systems have existed for a long time and are utilized in a wide specter of functions. In this thesis a vision-based system for measuring the relative motion between objects is developed. A camera is fixed to one object, while a pattern is fixed to the other object. Images are acquired of a scene containing the pattern, and the relative motion between the camera and the pattern is computed. Experiments have shown that under static conditions the vision based measurement system detects the heave distance between pattern and camera with less than 0.5 mm deviation at a distance of 2.3 m, and roll and pitch angles of the pattern is computed with less than 0.2° deviation.

Preface

This thesis is written as a part of the Master's Programme in Mechatronics at the University of Agder. It has been an interesting and challenging process, where a solution to a relevant problem has been looked into. The author is grateful for the opportunity to work with this subject.

The author would like to thank the supervisors, Professor Geir Hovland and Assistant Professor Morten Ottestad, for their invaluable assistance and guidance. Through the work with the thesis, the supervisors have been available both in meetings and whenever the author has requested assistance. The help provided by Morten Ottestad with his knowledge of LabVIEW has helped the author greatly in developing the software.

Thanks are also extended to PhD candidates Øyvind Berg Magnussen and Magnus Berthelsen Kjelland for allowing the author to borrow functions from their LabVIEW-libraries to communicate with the equipment used in the experiments. Lastly, having access to the Stewart-platforms in the Norcowe Motion Lab has allowed for experiments under controlled conditions where the positions to be measured are known with a high degree of accuracy.

Grimstad, June 2, 2014

Johan Lindal Haug

Johan Lindal Haug

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

When a vessel transfers a load to a fixed structure in an offshore environment, active heave compensation systems are often used to increase the operational window. One or more motion reference units (MRU) are used to measure the vessel's movement. The kinematics of the lifting structure is used in combination with the information from the MRU to find the relative motion, providing information to the system to be compensated how much movement needs to be compensated for. However, this is measured with respect to the world coordinate system, and will only provide information enough to maintain a fixed position of the load in relation to the seabed or a structure fixed to the seabed.

When transferring a load between vessels moving independently of each other, the situation becomes more complicated. For the vessel transferring the load to be able to perform this in a safe manner, it will need the MRU signals from the second vessel to be transferred either by cable between vessels, or wirelessly. MRU signals from, and kinematics of, both vessels have to be used to compute relative motion between the two vessels, which is the information the motion compensation system needs to be able to land the load safely. Neither using a wireless connection to exchange safety critical information, or going through the hassle of placing a physical cable between vessels is desirable, but being able to transfer loads vessel to vessel is.

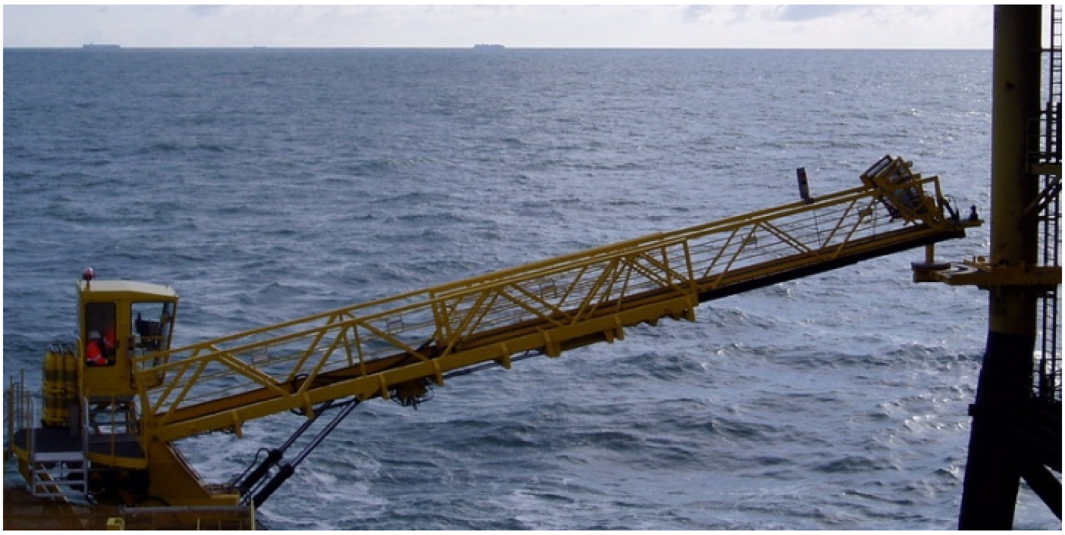


Figure 1.1: Load transfer example [1]

Figure 1.2 show an example of a load transfer where a gangway is extended from a vessel to a fixed structure. This gangway could be compensated for the motion of the ship by utilizing information from an MRU on the vessel. If the gangway is extended to a vessel moving freely in the waves, the information from the MRU information from the vessel with the gangway would not be enough to keep relative motion between a landing point on the structure and the tip of the gangway close to zero. The same problem would apply to a conventional crane, or any other operation where no relative motion between vessels is required.

1.2 Problem Statement

When transferring a load from vessel to vessel, and the vessels are moving independently of each other, finding the relative motion to be used in the motion compensation system can be challenging.

The necessary information can be found by combining MRU signals with knowledge about the kinematics of both vessels to find the the relative motion between a landing point and the point which should be kept at a constant distance. This requires the signals to be exchanged over a wireless connection or a cable physically connected between the vessels, and combined on one of the vessels to compute the relative motion between them.

1.3 Key Assumptions and Limitations

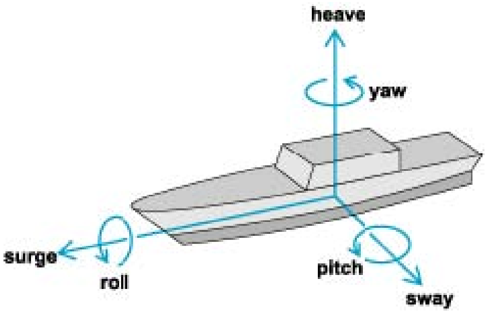


Figure 1.2: Degrees of freedom for vessel [3]

The vessel movement is limited to ±15° roll and ±15° pitch and wave frequency is defined as 0.05 to 0.125 Hz [2], giving waves with 8 to 20 seconds from peak to peak. The minimum update frequency of system is set to 10 Hz, approximately two orders of magnitude higher than highest occuring frequency in waves.

The operational range for the test system is limited by operational range of test equipment and ±400 mm heave is the highest achievable heave motion. During dynamix experiments the test platform will receive sinusoidal setpoints within its operational limits.

The system should be able to estimate the pose of the test platform with an accuracy of ±25 mm heave and ±0.5° roll and pitch. System should calculate all 6 degrees of freedom in the rotation matrix and 3d translational vector.

1.4 Problem Solution

Table 1.1 shows a simple ranking of a vision based solution versus a MRU based solution. MRU scores higher on robustness as it will always provide signals, and is a well proven technology. Points could however have been given to a vision based system, as the MRU system relies on the signals being transferred between vessels in this situation.

A vision-based system will most likely be simpler and cheaper. It may also be more accurate as it is not subject to drift in position measurements, unlike the MRU.

Table 1.1: Solution ranking

	Vision	2xMRU
Simple	1	0
Cheap	1	0
Robust	0	1
Accurate	1	0
Proven	0	1
Sum	3	2

A camera, positioned on the structure to be compensated for the relative motion between vessels that has the landing zone in its field of view and is able to find enough features, can establish the pose of the landing area. From this the relative motion between vessels can be computed without taking into account signals from motion reference units or transferring these signals between vessels.

1.5 Report Outline

Chapter 2 looks into the camera model and coordinate systems used. Chapter 3 explains the image processing, which is the first part of the sensor system. Chapter 4 shows the process of estimating the pose of the pattern, with features from the image known. Chapter 5 goes through the equipment used during testing, and chapter 6 shows the results from experiments performed with the system.

CHAPTER 2

Definitions

2.1 Camera Model

The frontal pinhole camera model is used to describe the parameters of the camera. Figure 2.1 shows an illustration of the model. The image plane (uv-plane) where the image sensor is located is parallell to the xy-plane of the camera coordinate system. Note that the planes have opposite positive directions for their coordinates. This is because the imaging sensor is in fact located at $z = -f$, but placing the image plane at $z = f$ will not mirror the image. Coordinate systems used are looked into later in this chapter.

The optic centre is where the infinitely small pinhole is assumed to be, and all lines from features in the scene will pass through this pinhole before being projected onto the imaging sensor at a point (u, v).

Focal length is the distance between the optical centre of the lens and the image sensor, often given in pixels. The principal point of the image plane is where the optical axis of the camera system intersects the image plane. This point is often slightly offset from the centre of the image sensor due to imperfections in the optics used.

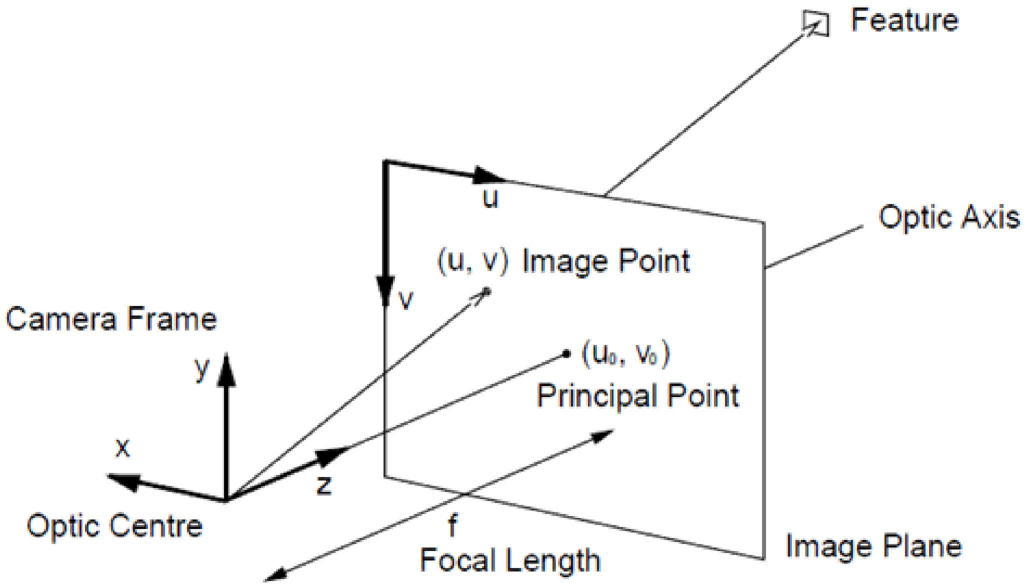


Figure 2.1: Illustration of frontal pinhole camera model [6]

2.1.1 Intrinsic Parameters

The intrinsic parameters describe the internal parameters of a camera and are listed in table 2.1.

Table 2.1: Intrinsic parameters

Focal length, u-direction	f_u
Focal length, v-direction	f_v
Principal point, u-direction	u_0
Principal point, v-direction	v_0
Distortions	k_1,k_2,k_3

Where:

- Focal length describes the distance between image plane and principal point
- Principal point is the point on the image plane where the optical axis intersects the image plane
- Distortion is a dimensionless vector describing the distortion in the image

The parameters focal length and principal point are given in pixels here. They make up the upper triangular matrix shown in equation (2.1), referred to as the *Camera Calibration Matrix* or K-matrix. A skewing factor α may be inserted in k_{12} if the pixels on the imaging sensors are not square. The camera used has an imaging sensor where the pixels are square, thus this value is set to zero.

$$K = \begin{bmatrix} f_u & 0(\alpha) & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

Depending on the quality of the optics used, radial and tangential distortion will be introduced in the image. Figure 2.2 shows the amount of radial distortion in an image acquired by the camera system used. The circles in the figure have numbers along their edges indicating how many pixels the image is distorted by, and the arrows show the direction of the distortion.

Radial distortion is corrected using the distortion coefficients k_1, k_2 and k_3. Equation (2.2) shows how the distortion is corrected when coordinates are normalized as shown in section 4.1.1, around the principal point of the image[8]. Subscript d indicates distorted coordinate, i indicates that coordinate is in the normalized image plane. Variable r is the distance between the point to be corrected and the principal point, computed from equation (2.3).

$$\begin{aligned} x_i &= \frac{x_{id}}{1 + k_1 r^2 + k_2 r^4 + k_3 r^6} \\ y_i &= \frac{y_{id}}{1 + k_1 r^2 + k_2 r^4 + k_3 r^6} \end{aligned} \tag{2.2}$$

$$r^2 = x_{id}^2 + y_{id}^2 \tag{2.3}$$

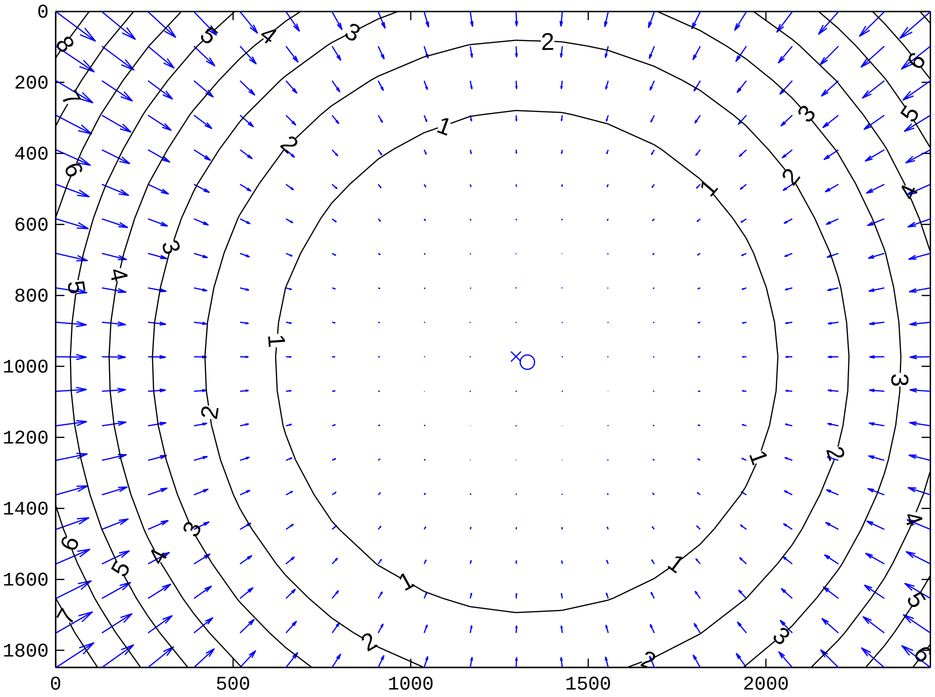


Figure 2.2: Radial distortion from *Camera Calibration Toolbox for Matlab*

2.1.2 Extrinsic Parameters

The extrinsic parameters describe the position of the camera reference frame with respect to a world reference frame. The transformation between these two reference frames consists of the 3x3 rotation matrix $\boldsymbol{R}$, and the 3x1 translation vector $\boldsymbol{T}$. These two variables are collected in a matrix referred to as the ($\boldsymbol{RT}$)-matrix which represents the extrinsic parameters, shown in equation (2.4).

$$(\boldsymbol{RT}) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

The function of the extrinsic parameters are illustrated in figure 2.3. The camera coordinate system is shown on the left, and a world coordinate system is shown on the right. The translation vector $\boldsymbol{T}$ will describe the translation between the origins of the two coordinate systems. The rotation matrix $\boldsymbol{R}$ describe the rotation that is required to bring the axes into alignment with each other. The angles used to represent the rotations are looked into in section 2.3.

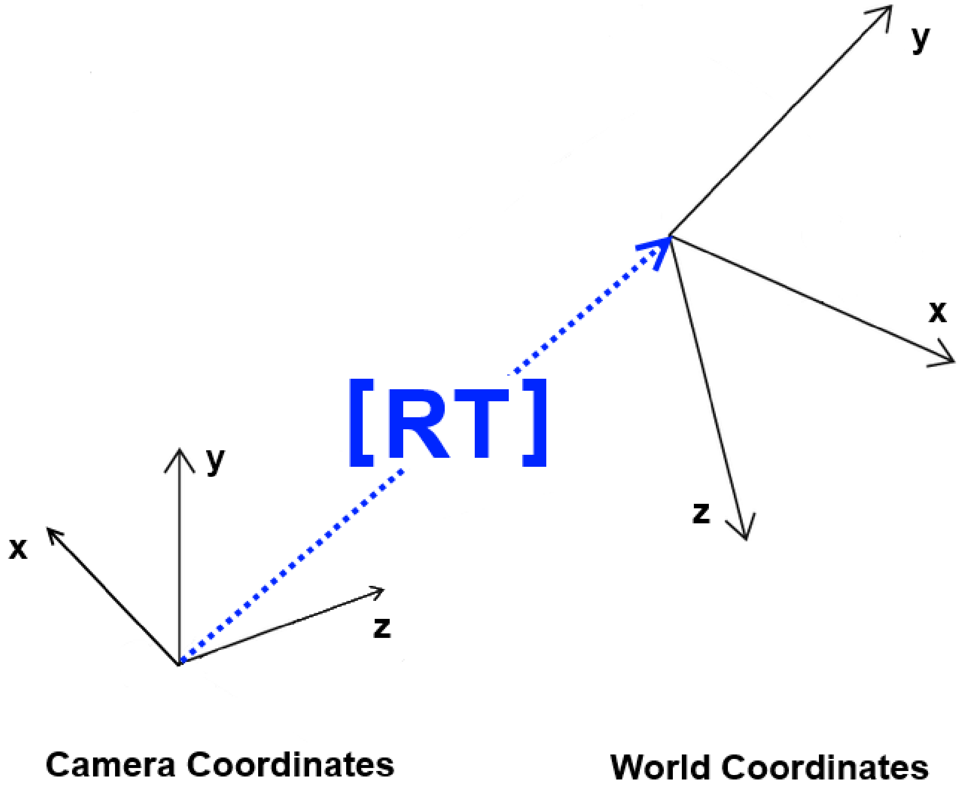


Figure 2.3: Extrinsic parameters

2.1.3 Camera Calibration

The extrinsic parameters of a camera will change when the position and attitude of the camera is changed, but the intrinsic parameters will stay the same if the optics stay the same. Therefore the camera can be calibrated in a known environment to find the intrinsic parameters, and use these in other settings. To find the calibration data, the *Camera Calibration Toolbox for Matlab* [7] was used. Several images of a chessboard pattern of known size were acquired, and the toolbox calculated the intrinsic parameters of the camera by analysing these images.

For the camera used, the objective lenses can be changed, the objective lens used has a focal length of 12 millimeter. The camera calibration matrix acquired is shown in (2.5) and the distortion coefficients are shown in table 2.2.

$$K = \begin{bmatrix} 5562 & 0 & 1327 \\ 0 & 5548 & 987 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

Table 2.2: Distortion coefficients

Coefficient	Value
k_1	-0.09195
k_2	0.36222
k_3	-0.00302

2.2 Pattern

For the software to detect features correctly, it is important to select something in the image that can easily be located and matched to known coordinates. The corners of a chessboard pattern were chosen because this pattern provides clear corners that can easily be detected in the image and measured on the real-world pattern.

Depending on where in the chessboard the corner is located, an L- or X-corner can be found. Where two black squares meet there will be a X-corner, while at the outer border the black squares will produce L-corners. A chessboard pattern containing only X-corners has been created to give the software better indications on where the interesting features are located [9]. This has been achieved by rounding off all black fields bordering the pattern as shown in figure 2.4.

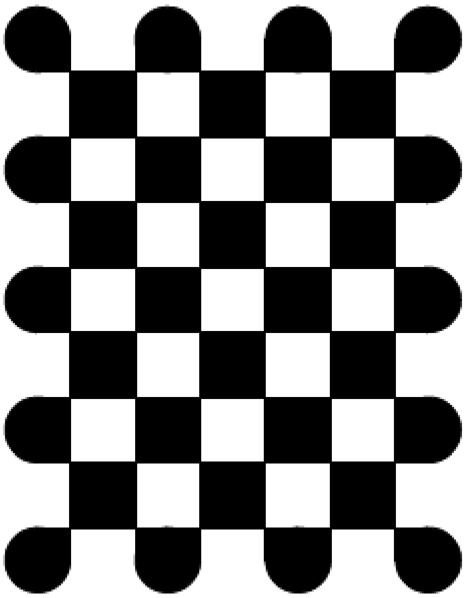


Figure 2.4: Chess pattern with 48 X-corners

2.3 Coordinate Systems

SNAME (Society of Naval Architects and Marine Engineers) notation is used for the orientation and position, pose, of the pattern. X-axis is along the long side of the pattern and y-axis is along the short side of the pattern. Z-axis is normal to the pattern and has its positive direction away from the camera.

Figure 2.5 illustrates the world and pattern coordinate systems, located in the center of the pattern with x-axis parallell to the longest side of the pattern. As the pattern moves, so does the world coordinate system. The extrinsic parameters will then describe the relative motion between camera and pattern.

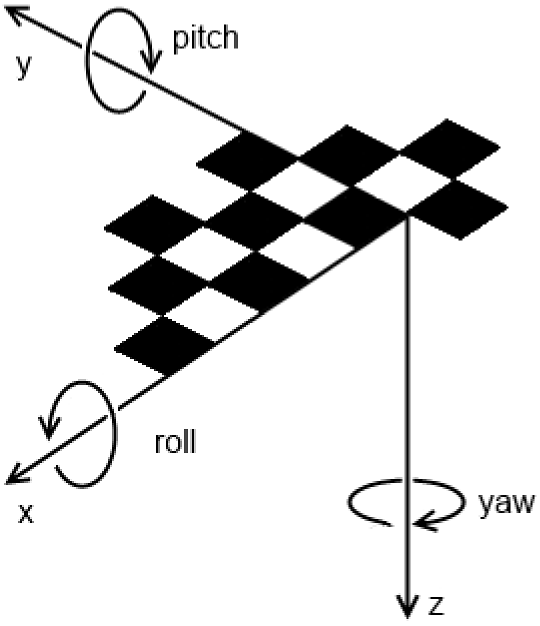


Figure 2.5: World coordinate system

Tait-Bryan angles are used for the three rotations representing the patterns orientation. The Tait-Bryan angles describe three successive rotations about fixed axes. The XYZ order is chosen here, with the first rotation as roll about the x-axis, the second rotation is pitch about the y-axis, and the last rotation is yaw about the z-axis.

Angles and positions of the pattern will have the camera origin as the reference point, with translations in x- and y-directions at zero when the center of the patterns representation in pixel coordinates is located at the principal point of the imaging sensor. Higher z-value in the translation vector means pattern is further away from the camera.

When using Tait-Bryan angles, the rotations are measured at the local coordinate system of the pattern. If pattern is rotated around the z-axis, the x-axis will follow the rotation, and roll will be measured about the new orientation of the x-axis.

A drawback when using Tait-Bryan angles is that there exists a singularity caused by gimbal lock at the second rotation when its angle reaches 90°. In this case the second rotation is the pitch-angle, and 90° is outside the range, thus this will not become a problem.

The degrees of freedom for the pattern with their names and notations are shown in table 2.3.

Table 2.3: Degrees of freedom

Degree of freedom	Name	Notation
Translation along x-axis	Surge	x_w
Translation along y-axis	Sway	y_w
Translation along z-axis	Heave	z_w
Rotation about x-axis	Roll	θ_{roll}
Rotation about y-axis	Pitch	θ_{pitch}
Rotation about z-axis	Yaw	θ_{yaw}

For practical reasons, only heave, roll and pitch will be looked thoroughly into, as these are the degrees of freedom easiest to compare against the position of the Stewart platform.

To summarize, there are two 3D coordinate systems and two 2D coordinate system used. The world coordinate system which is located at the pattern, and the camera coordinate system, located at the optical centre of the camera (the objective lens) are the three-dimensional systems. The two-dimensional coordinate systems are the pixel coordinates at the imaging sensor, and the homogeneous image plane at f=1. Table 2.4 lists the notations used for coordinates in the different systems.

Table 2.4: Coordinate systems

World Coordinates	x_w, y_w, z_w
Camera Coordinates	x_c, y_c, z_c
Homogeneous Image Coordinates	x_i, y_i
Pixel coordinates	u, v

CHAPTER 3

Image Processing

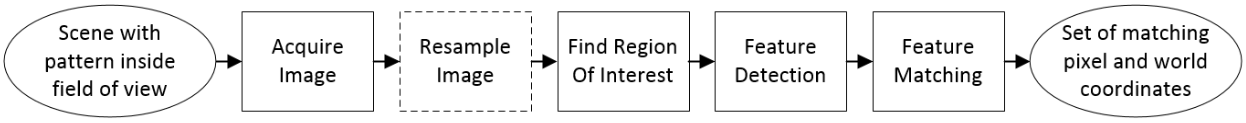


Figure 3.1: Image processing sequence

Figure 3.1 shows the sequence of operations in the image processing. An image is acquired of a scene where the pattern is inside the field of view for the camera system. If required, the image is resampled, before the pattern is located, and features of the pattern are found. The pixel coordinates of the features are matched to world coordinates, and a set of matching pixel and world coordinates is known.

3.1 Acquire Image

An image of the scene is acquired from the buffer of the camera at the beginning of each cycle of the software. The function to acquire images can wait for a new image to be available or grab the image that at any given time is the most recent image in the buffer. The camera used has a framerate of 14 frames per second (fps), and delivers a new image of the scene to its buffer approximately every 70 ms.

If the total cycle time of the software is less than this, there is a chance that the same image is processed twice. However, if the software is to wait for a new image, the total cycle time is increased while waiting for a new image without performing any computations. While the software is processing the image, a new image is acquired and stored in the buffer, and as soon as the current image has been processed, the next image is retrieved from the buffer.

3.2 Image Resampling

An image with full resolution of the camera system used, 2592x1944 pixels (5 MP), may put a heavy load on the computer when all operations listed in this chapter are performed on it, therefore the first operation is to resample the image according to an adjustable scaling value. This will still provide accurate measurements. However, more noise has been observed in the results when the image is resampled. When the pattern is a small part of the total image, it is not neccessary to resample the image in order to maintain cycle time below 100 ms.

3.3 Region of Interest

To further ease the CPU load, a region of interest (ROI) is found and used to narrow down which part of the image contains useful information.

A binary image is created to separate the useful information from the background by applying a threshold to the original image. A threshold operation uses an intensity level, k, to determine whether a pixel belongs to class 0, where intensities are lower than the threshold intensity, or class 1, where intensities are higher, as shown in figure 3.2. This threshold operation is looking for the dark areas of the image (black areas of the chessboard), and therefore pixels in class 0 are set to $I(u, v) = 1$ and pixels in class 1 are set to $I(u, v) = 0$.

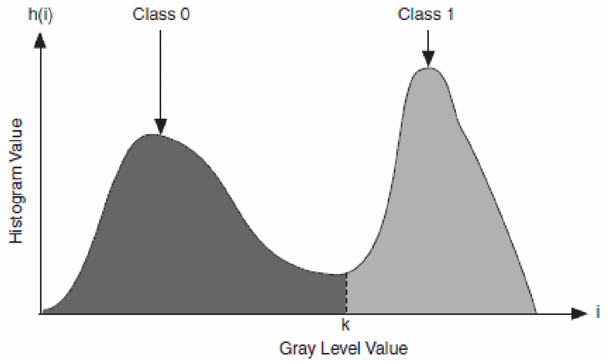


Figure 3.2: Histogram used for thresholding [11]

An automatic thresholding function in LabVIEW is used. This function is using an inter-variance technique, Otsu's method [10], to find the optimal threshold level to segment the foreground from the background. Equation (3.1) shows how the function computes the between-class variance for a threshold intensity level k.

$$\sigma_B^2(k) = \frac{(\mu_T \omega(k) - \mu(k))^2}{\omega(k) - (1 - \omega(k))} \tag{3.1}$$

Where

$$\mu(k) = \sum_{i=0}^{k} ip(i) \tag{3.2}$$

$$\mu_T = \sum_{i=0}^{N-1} ip(i) \tag{3.3}$$

$$\omega(k) = \sum_{i=0}^{k} p(i) \tag{3.4}$$

$$p(i) = \frac{h(i)}{\sum_{i=0}^{N-1} h(i)} \tag{3.5}$$

and

- k = Gray level threshold intensity
- i = Gray level intensity
- h(i) = The number of pixels at each gray level
- N = The total number of gray levels in the image
- n = The total number of pixels in the image

The histogram is normalized, and regarded as a probability distribution. Function $p(i)$ in (3.5) is then the probability of a gray level i in the image.

Equation (3.3) is the total mean gray level in the image, (3.4) and (3.2) are the zeroth- and first-order cumulative moments of the histogram. The optimal threshold level, k_{opt}, is found when the between-class variation is highest, see (3.6).

$$\sigma_B^2(k_{opt}) = \max_{0 \leq k < (N-1)} \sigma_B^2(k) \tag{3.6}$$

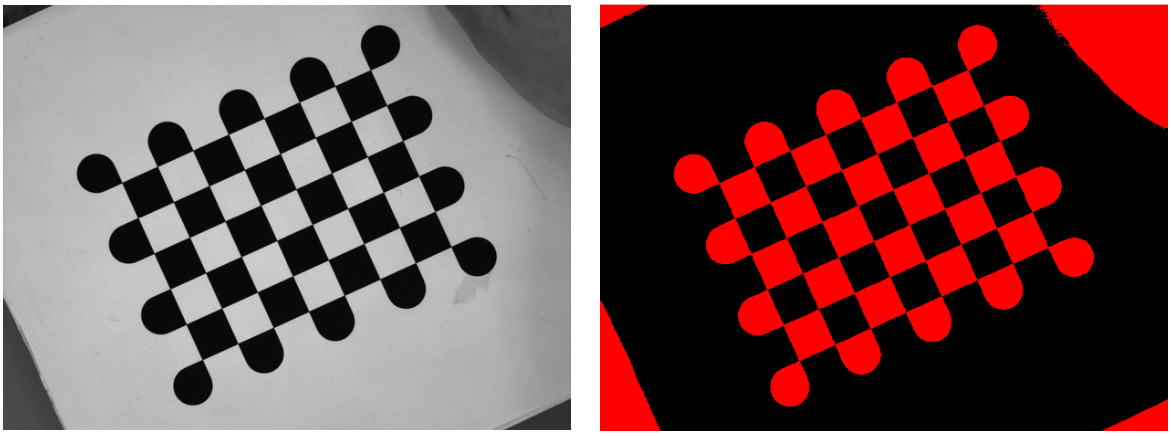


Figure 3.3: Input image and thresholded image for ROI

Figure 3.3 shows the input image on the left side, and the thresholded image on the right side. The black area of the chessboard has been clearly marked in red, but the same has happened to parts of the background.

To remove the parts of the background that have been determined as dark enough to be part of the chessboard, an operation that removes all regions bordering the image is applied, leaving only the black regions of the chessboard and objects not on the border in the resulting image.

The image is then dilated to ensure that all black regions in the chessboard are connected. In a dilation operation performed on a binary image, all pixels with $I(u,v) = 1$ will grow into neighbouring pixels with $I(u,v) = 0$, enlarging the area of touching pixels with intensity $I(u,v) = 1$.

A blob analysis is performed, and the largest object found is assumed to be the black regions of the chess pattern. Blob is short for Binary Large OBject, and is defined as *"an area of touching pixels with the same logical state"* [12]. LabVIEWs' blob analysis can provide a vast amount of information about the blobs, but only the following information is acquired for each blob in this step:

- Area in pixels
- Pixel coordinates for upper left corner of a rectangle encompassing the blob
- Pixel coordinates for lower right corner of a rectangle encompassing the blob

The blob with the largest area is selected, and the corners defining a rectangle around it is used to create a template for cropping the original image to the minimal image size containing the pattern.

3.4 Feature Detection

To find the pixel coordinates (u, v) of the corners, a Harris-Stephens corner detector[14], commonly referred to as a Harris detector, is used. When a Harris detector is applied to an image, the corners will be highlighted as bright spots, and other parts of the output image becomes black.

A window function is passed over the image to find the intensity in u- and v-direction for each shift the window does through the image. Equation (3.7) shows how the change of intensity for a shift $[u, v]$ is computed.

$$E(u,v) = \sum_{x,y} w(x,y)[I(u+x, v+y) - I(u,v)]^2 \tag{3.7}$$

Where:

- $w(x, y)$ = Windowing function
- $I(x + u, y + v)$ = Shifted intensity
- $I(u, v)$ = Intensity

Using a Taylor expansion the approximation in equation (3.8) is produced.

$$E(u,v) \cong \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{3.8}$$

Where M is the Harris matrix shown in equation (3.9). I_u represents the gradient in u-direction and I_v represents the gradient in v-direction found using the Sobel-operators.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix} = \begin{bmatrix} [I_u^2] & [I_u I_v] \\ [I_u I_v] & [I_v^2] \end{bmatrix} \tag{3.9}$$

From the eigenvalues of the Harris matrix, information about features in the image can be found. If $\lambda_1 \approx \lambda_2 \approx 0$ the pixel contains no useful feature. If one of the eigenvalues has a large positive value the pixel is part of an edge. When both eigenvalues have large positive values, the pixel is part of a corner as can be seen in the green area of figure 3.4.

On this figure the value of λ_1 is along the x-axis, and the value of λ_2 is along the y-axis. Red areas indicate where only one eigenvalue has a large value, the grey area is the areas of the image where no gradients are found.

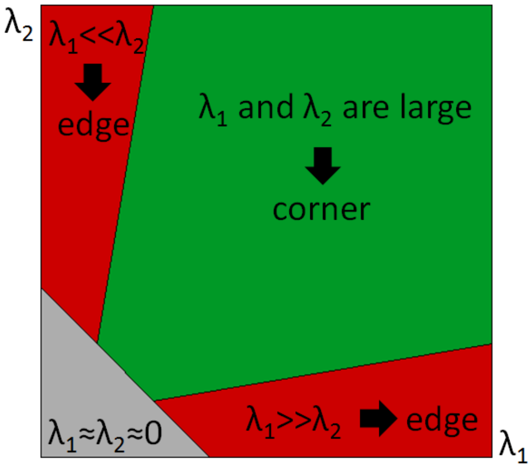


Figure 3.4: Graphic describing eigenvalues of the Harris matrix [14]

Finding eigenvalues for all pixels in an image is computationally expensive. Harris-Stephens [14] suggested the function M_c shown in equation (3.10) where k is a tunable sensitivity parameter and a corner response for the pixel is acquired by computing determinant and trace of the matrix M. Figure 3.5 shows the output image after the Harris-Stephens corner detector has been applied to the region of interest in the input image.

Before determinant and trace of M are computed, a Gaussian smoothing filter is applied to I_u^2, I_v^2 and I_uI_v to remove noise from the image derivatives. An image is a large matrix of intensity values, and to increase the speed of the computations, M_c is computed with vectors representing columns of the image matrix. Instead of 2592 pixels * 1944 pixels ≈ 5 million operations, the computation is reduced to 1944 operations (the number of colums) to compute corner responses for the full image. This number is further reduced by the ROI-operation.

$$M_c = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 = det(M) - k \cdot trace^2(M) \tag{3.10}$$

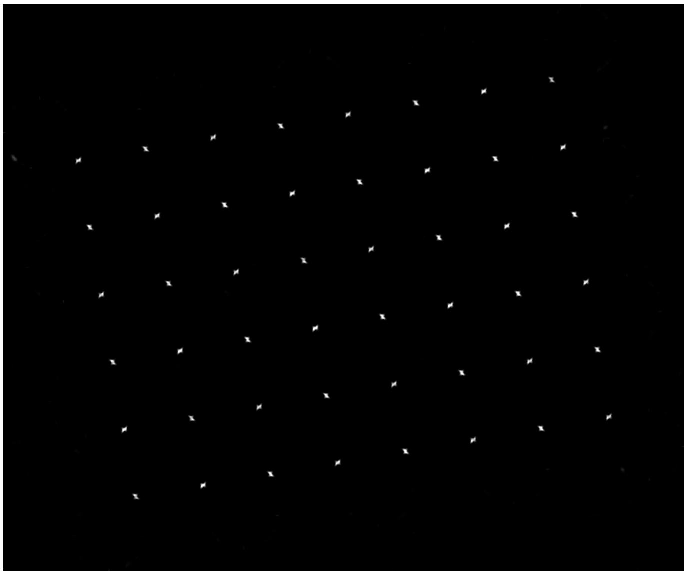


Figure 3.5: Output image with responses from Harris corner detector

3.5 Feature Localization

The output from the Harris corner detector contains bright areas where there are corners located in the original image. This image with corner responses is thresholded in the same type of operation as described in section 3.3. However, in this operation the objective is to find the bright regions of the image, thus all pixels in class 0 are set to $I(u,v) = 0$ and all pixels in class 1 are set to $I(u,v) = 1$.

Figure 3.6 shows four steps of the process of finding the features in the image. From left to right the figure shows a corner in the original image, the response from the Harris detector, the thresholded image, and at last a dilated image.

The thresholded image shows that the corner detector response may be picked up as several blobs. To ensure that the indications for a corner is detected as one blob, the image is dilated.

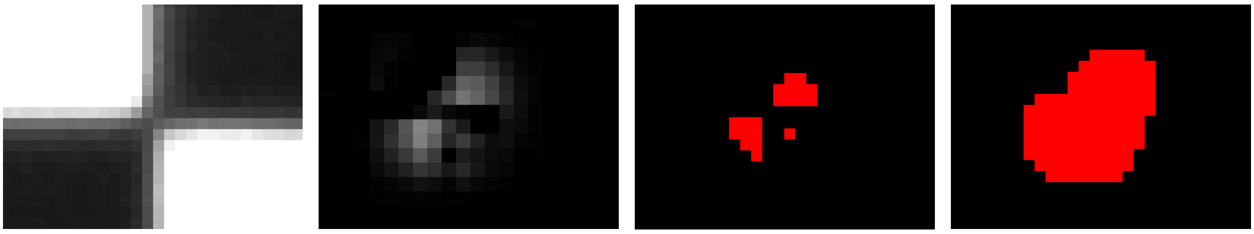


Figure 3.6: Feature localization

From the image produced by these operations, a blob analysis is performed. For all blobs found in the image the following information is acquired:

- Area in pixels
- Pixel coordinates for center of mass

The center of mass for each blob is assumed to be the pixel coordinates where a corner is located.

3.6 Feature Correspondences

3.6.1 Finding Axes of Pattern

A set of coordinates where corners are located in an image must be matched to a set of known world coordinates to be interesting for further investigation. The distance from each point to all other points are computed, and the 4 smallest distances are selected and stored as the point's neighbours.

Figure 3.7 illustrates the two possible outcomes from this operation. On the left hand side a point located somewhere inside the pattern is shown, with its 4 closest corners marked with green distributed around it. On the right hand side a point located along an edge of the pattern is shown, two of the points are marked in faded green to indicate that one of them will be selected as the 4th closest point. In this case the neighbours will not be distributed around the centre point.

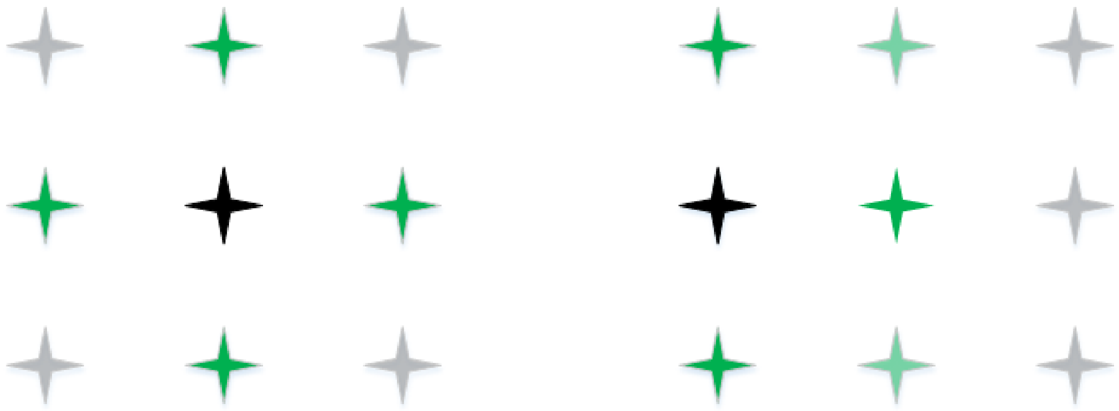


Figure 3.7: Neighbour points inside and along edge of pattern

Each point that has its 4 closest neighbours distributed around itself, as shown on the left hand side in figure 3.7, can be used to find the axes of the pattern. Each neighbour will have a relative position to the the center point assigned as *north*, *south*, *east* and *west*.

The neighbour with lowest v-coordinate is defined as *north*, and highest v-coordinate is defined as *south*. Similarly the neighbour with lowest u-coordinate is defined as *west*, and highest u-coordinate is defined as *east* as shown in figure 3.8.

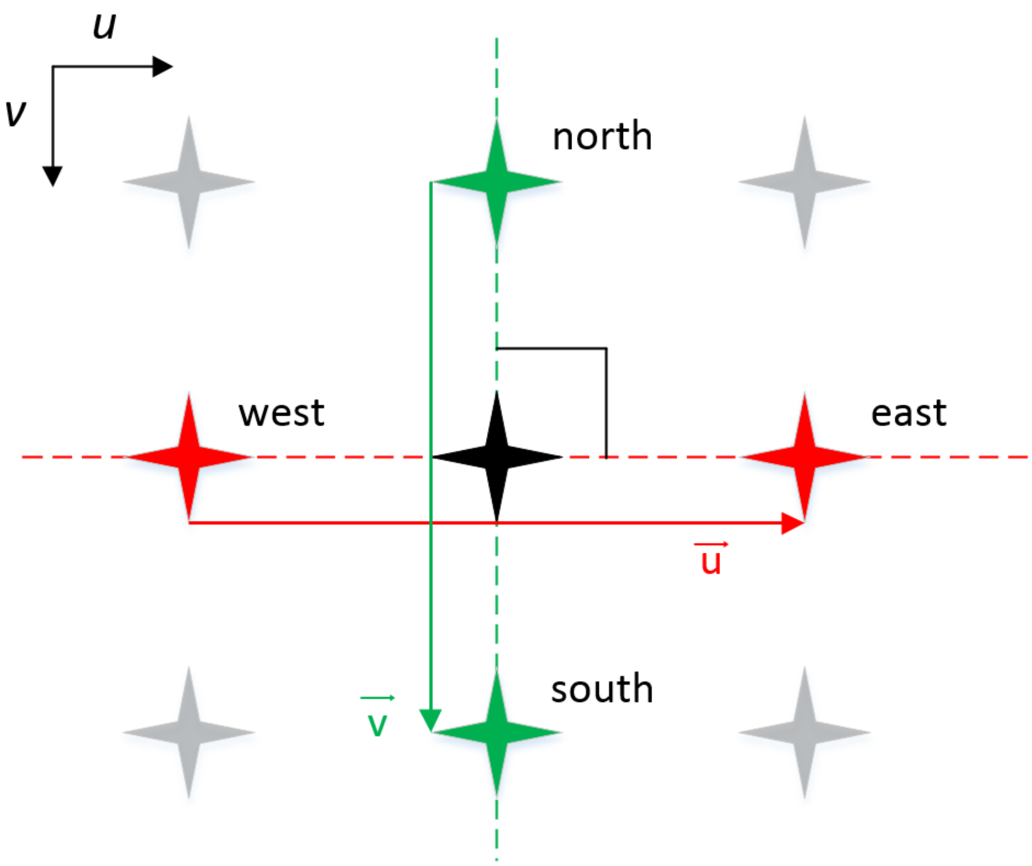


Figure 3.8: Points used to find axis lines

If the dot-product of the vector $\vec{u}$ from *west* to *east* and the vector $\vec{v}$ from *north* to *south* is approximately zero, they are normal to each other and it is assumed that these vectors are parallell to the axes of the pattern indicated with stapled lines.

Equation (3.11) shows the expression used to filter out the blobs with neighbours supporting the axes. Unit vectors are used to keep the expression invariant to changes in distances between pattern and camera.

$$\frac{\vec{u}}{||\vec{u}||} \bullet \frac{\vec{v}}{||\vec{v}||} < 0.15 \tag{3.11}$$

As the pattern is rotated, this angle will not appear as 90° because of the perspective change. The limit 0.15 has been determined experimentally as the threshold to approve the angles as normal to each other, as this will accept the correct points within the working range of ±15° pitch and roll.

To find the axes, the first point found that has its 4 closest neighbours distributed as shown on the left hand side in 3.7, and in addition has an eastern neighbour fulfilling these criteria is selected. The reason for choosing a point that has an eastern neigbour is explained further in section 4.2. The point chosen is used to draw lines making up the local axes of the pattern.

The RANSAC (RANdom Sample And Consensus) algorithm [13] is used in many vision applications to differentiate between inliers and outliers to a line. It can be used to find the points that lies along a line. An important input to the RANSAC algorithm is the threshold, t, to approve or reject a point as an inlier. The threshold is the distance from a point to the line being checked, the distanse used is a quarter of the distance between east and west as shown in (3.12) .

$$t = \frac{||\vec{u}||}{4} \tag{3.12}$$

By using the RANSAC algorithm but modifying it so that the line to search along is not selected at random, but instead uses the neighbours of the chosen point, the number of corners along the lines are counted. The line with highest number of inliers is defined as being parallell to the x_w-axis in the pattern. Since these lines are normal to each other, the line with the lowest number of inliers is then parallell to the y_w-axis.

3.6.2 Stacking Features

With the direction of the pattern known, a systematic way of stacking the the corners must be used, such that each pixel-coordinate of a corner can be linked to the corresponding corner in the actual pattern.

At this point the directions of the axes are known, and this information can be used to collect the corners in an orderly fashion that match a set of coordinates for the corner positions in the actual chessboard.

The line parallell to the y-axis of the pattern, indicated as a green line in figure 3.8, is used to create an array of points by counting inliers using the modified RANSAC algorithm and sorting points by lowest v-value.

For each point in the array, a line passing through the point and in parallell to the x_w-axis is created. For each of these lines the inliers are found, and stacked from lowest u-value to highest. In figure 3.9 the order the coordinates of the corners are stacked is indicated with red numbers.

Corner 1 through 8 are the corners found when looking for inliers to the line parallell to the x_w-axis and going through the first point of the array. Corner 9 through 16 are the corners found when looking for inliers to the line parallell to the x_w-axis and going through the second point of the array.

This sequence is repeated for as many entries the array has to stack all coordinates in a matrix containing u- and v-coordinates for all points in the image (in pixels). A list of world coordinates of the corners in the actual pattern stacked in the same order is combined with the pixel coordinates to give a set of matching pixel and world coordinates.

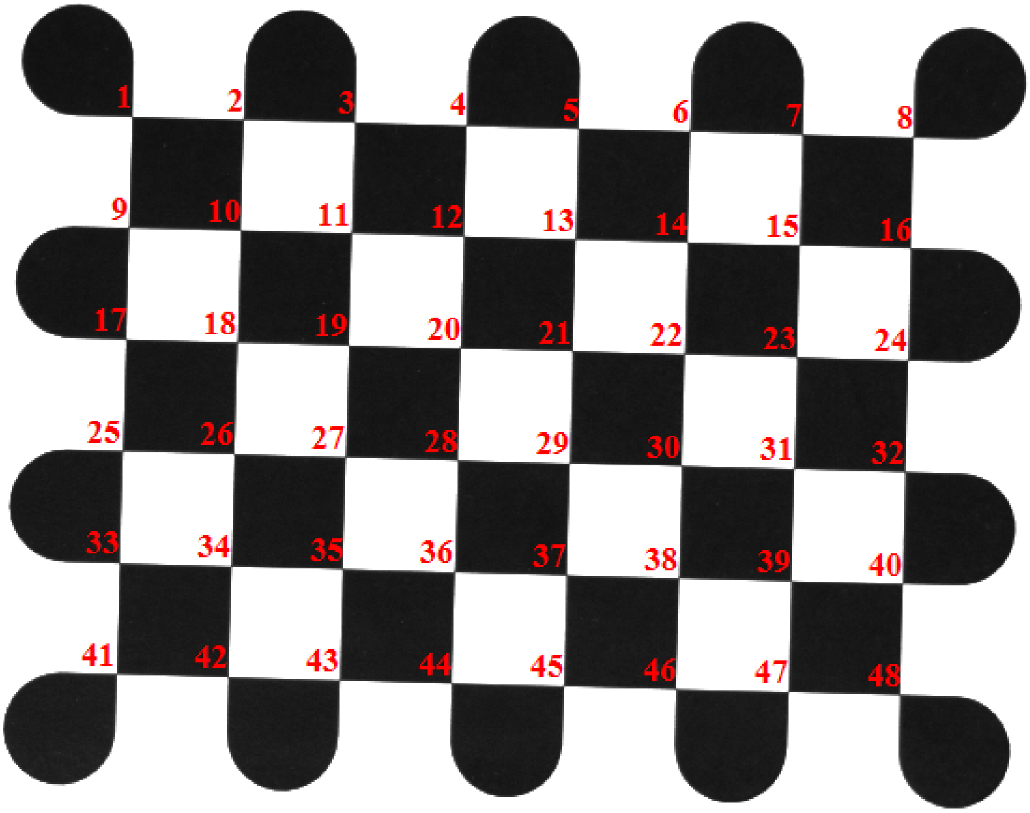


Figure 3.9: Numbered corners

CHAPTER 4

Pose Estimation

4.1 Homography

Homography describes the relation between a pixel coordinate and a world coordinate of the same point. The equations used are explained in this section.

Figure 4.1 illustrates how the known information is used to find the extrinsic parameters which are the unknowns here. The pixel coordinates, world coordinates and the intrinsic parameters of the camera are combined to compute the pose of the pattern from the extrinsic parameters.

The world coordinates of each corner in the pattern is known by controlling the size of each square when printing, and measuring the pattern to find the x_w- and y_w-coordinates. If z_w is assumed to be zero for all points in the world coordinate system, the pattern is assumed to be located at the xy-plane of the camera coordinate system.

Pixel coordinates of corners are known and correspondences to world coordinates of corners have been found in the previous chapter.

The extrinsic parameters will then describe the position of the camera in relation to the pattern, providing information about the relative motion between origin of the camera coordinate system and the center of the pattern.

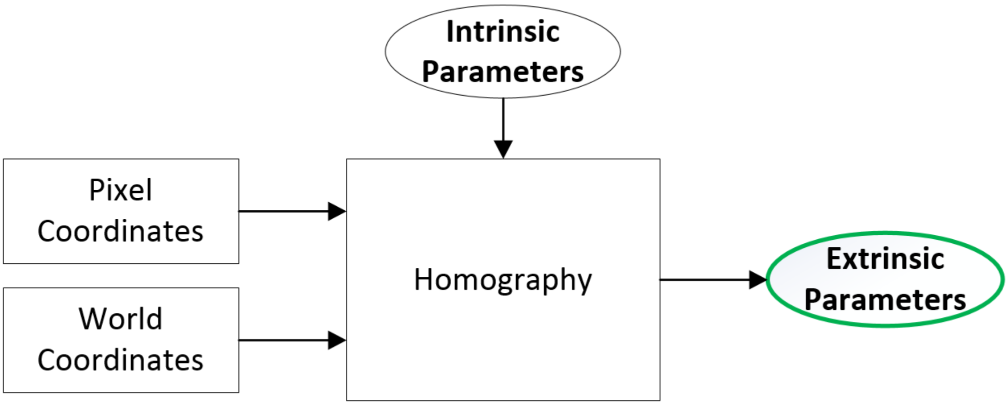


Figure 4.1: Computing extrinsic parameters from homography

The intrinsic parameters describe the transformation between camera coordinates and pixel coordinates by equation (4.1) where λ is a dimensionless scaling value and subscript c denotes that the coordinate is given in the camera coordinate system.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{4.1}$$

The extrinsic parameters describe the transformation between camera coordinates and world coordinates by equation (4.2). Subscript w denotes world coordinate.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{4.2}$$

Collecting these expressions by inserting (4.2) into (4.1) yields equation (4.3) which describes the transformation between pixel coordinates and world coordinates of a point.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{4.3}$$

As mentioned earlier in this section, z_w is set to zero for all points. Equation (4.4) shows that the third column of the rotation matrix will disappear, leaving the compressed equation shown in (4.5) if the identity matrix is multiplied with the $\boldsymbol{RT}$-matrix.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & 0 & t_x \\ r_{21} & r_{22} & 0 & t_y \\ r_{31} & r_{32} & 0 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix} \tag{4.4}$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \tag{4.5}$$

The homography-matrix $\boldsymbol{H}$, shown in equation (4.6), is introduced and inserted into (4.5) to give (4.7).

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \tag{4.6}$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} \tag{4.7}$$

From (4.7) the following set of equations exist.

$$\lambda \cdot u = h_{11} \cdot x_w + h_{12} \cdot y_w + h_{13} \tag{4.8}$$

$$\lambda \cdot v = h_{21} \cdot x_w + h_{22} \cdot y_w + h_{23} \tag{4.9}$$

$$\lambda = h_{31} \cdot x_w + h_{32} \cdot y_w + h_{33} \tag{4.10}$$

Inserting (4.10) into (4.8) and (4.9) and reorganizing gives equations (4.11) and (4.12). These equations describe the relation between the two representations (u,v) in pixel coordinates and (x_w,y_w) in world coordinates of the same point.

$$h_{11} \cdot x_w + h_{12} \cdot y_w + h_{13} - h_{31} \cdot x_w \cdot u - h_{32} \cdot y_w \cdot u - h_{33} \cdot u = 0 \tag{4.11}$$

$$h_{21} \cdot x_w + h_{22} \cdot y_w + h_{23} - h_{31} \cdot x_w \cdot v - h_{32} \cdot y_w \cdot v - h_{33} \cdot v = 0 \tag{4.12}$$

Stacking these equations in matrix form and collecting h_{xx} in a vector, the expression in (4.13) is obtained. Subscript n denotes the number of matching points in image coordinates, (u, v), and world coordinates, (x_w, y_v).

$$\begin{bmatrix} x_{w1} & y_{w1} & 1 & 0 & 0 & 0 & -x_{w1} \cdot u_1 & -y_{w1} \cdot u_1 & -u_1 \\ 0 & 0 & 0 & x_{w1} & y_{w1} & 1 & -x_{w1} \cdot v_1 & -y_{w1} \cdot v_1 & -v_1 \\ x_{w2} & y_{w2} & 1 & 0 & 0 & 0 & -x_{w2} \cdot u_2 & -y_{w2} \cdot u_2 & -u_2 \\ 0 & 0 & 0 & x_{w2} & y_{w2} & 1 & -x_{w2} \cdot v_2 & -y_{w2} \cdot v_2 & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{wn} & y_{wn} & 1 & 0 & 0 & 0 & -x_{wn} \cdot u_n & -y_{wn} \cdot u_n & -u_n \\ 0 & 0 & 0 & x_{wn} & y_{wn} & 1 & -x_{wn} \cdot v_n & -y_{wn} \cdot v_n & -v_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \end{bmatrix} \tag{4.13}$$

$$\boldsymbol{A} \cdot \boldsymbol{h} = \mathbf{0} \tag{4.14}$$

Equation (4.13) is compressed in (4.14), where $\boldsymbol{A}$ is a [9x2n] matrix and $\boldsymbol{h}$ is the vector containing elements h_{xx} from the matrix $\boldsymbol{H}$ introduced in (4.6).

The unknown in this situation is the values of the $\boldsymbol{h}$-vector, from which the extrinsic parameters can be computed. The $\boldsymbol{h}$-vector can be found by singular value decomposition of $\boldsymbol{A}$.

$$\boldsymbol{A} = \boldsymbol{U\Sigma V}^T \tag{4.15}$$

Equation (4.15) shows the general expression for singular value decomposition of a matrix, the $\boldsymbol{h}$-vector is retrieved from the last column of $\boldsymbol{V}$, and reshaped into the [3x3] matrix $\boldsymbol{H}$ [15].

Earlier the $\boldsymbol{H}$-matrix replaced matrix $\boldsymbol{K}$ multiplied with the $\boldsymbol{R}|\boldsymbol{T}$-matrix. The operation is reversed in (4.16), and the first two columns of the rotation matrix and the translation vector has been found.

$$\begin{bmatrix}\boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t}\end{bmatrix} = \boldsymbol{K}^{-1} \cdot \boldsymbol{H} \tag{4.16}$$

Translation Vector

The translation vector that comes out from these operations will have $t_z \approx 1$, meaning the values are given in homogeneous coordinates, the reason for this is looked into in section 4.1.1. A property of the columns in the rotation matrix is that they must have a length of 1. To satisfy this criterion the current length of a vector is found, and used to divide the vector by to get the correct scale.

The lengths of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are approximately equal. To scale the translational vector to its correct size, $\boldsymbol{t}$ is divided by the average length of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ as shown in (4.17). This will give the translation vector scaled to world coordinates between origin of the camera system and center of the pattern. From this the relative position between camera and pattern is known.

$$\boldsymbol{T} = \frac{\boldsymbol{t}}{\frac{||\boldsymbol{r}_1||+||\boldsymbol{r}_2||}{2}} = \begin{bmatrix}t_x\\ t_y\\ t_z\end{bmatrix} \tag{4.17}$$

Rotation Matrix

It has already been mentioned that the column vectors in a rotation matrix must have a length of 1. This is because the rotation matrix is an orthonormal matrix, and this property can be used to find the last column vector, $\boldsymbol{r}_3$ [16].

The known elements $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are first normalized by dividing them by their lengths. To compensate for numerical errors, the Gram-Schmidt theorem is applied to $\boldsymbol{r}_2$ to ensure that $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are orthogonal.

When all vectors in the rotation matrix are orthonormal, cross-multiplying $\boldsymbol{r}_1$ with $\boldsymbol{r}_2$ will yield $\boldsymbol{r}_3$ as shown in (4.18). After this operation all columns in the rotation matrix are known. From the rotation matrix the roll, pitch and yaw angles can be computed.

$$\boldsymbol{r}_1 \times \boldsymbol{r}_2 = \boldsymbol{r}_3 \tag{4.18}$$

The rotation matrix consists of three rotation matrices representing each elemental rotation as shown in compressed form in (4.19), and expanded where the matrix for each rotation is shown in (4.20). As stated in section 2.3, XYZ order is used for the angles, therefore the matrix consists of the roll rotation about the x-axis, then pitch rotation about the y-axis and at last the yaw rotation about the z-axis.

$$\boldsymbol{R} = \boldsymbol{R}_x(\theta_{roll}) \cdot \boldsymbol{R}_y(\theta_{pitch}) \cdot \boldsymbol{R}_z(\theta_{yaw}) \tag{4.19}$$

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_{roll}) & -sin(\theta_{roll}) \\ 0 & sin(\theta_{roll}) & cos(\theta_{roll}) \end{bmatrix} \begin{bmatrix} cos(\theta_{pitch}) & 0 & sin(\theta_{pitch}) \\ 0 & 1 & 0 \\ -sin(\theta_{pitch}) & 0 & cos(\theta_{pitch}) \end{bmatrix} \begin{bmatrix} cos(\theta_{yaw}) & -sin(\theta_{yaw}) & 0 \\ sin(\theta_{yaw}) & cos(\theta_{yaw}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.20}$$

In (4.21) the matrices are shown multiplied together. Instead of writing out the full length of the expression in each element, $cos(\theta_{yaw})$ is written as c_y, $sin(\theta_{pitch})$ is written as s_p and so on. Equations (4.22) through (4.24) show the equations used to retrieve roll, pitch and yaw angles from the rotation matrix. These equations are the same as used in the *Robotics toolbox* for Matlab by Peter Corke [17].

$$\boldsymbol{R} = \begin{bmatrix} c_y c_p & c_y s_p s_r - s_y c_r & c_y s_p c_r + s_y s_r \\ s_y c_p & s_y s_p s_r - c_y c_r & s_y s_p c_r - c_y s_r \\ -s_p & c_p s_r & c_p c_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4.21}$$

$$\theta_{roll} = atan2\left(\frac{-\ r_{23}}{r_{33}}\right) \tag{4.22}$$

$$\theta_{pitch} = atan2\left(\frac{r_{13}}{cos(\theta_r) \cdot r_{33} - sin(\theta_r) \cdot r_{23}}\right) \tag{4.23}$$

$$\theta_{yaw} = atan2\left(\frac{-\ r_{12}}{r_{11}}\right) \tag{4.24}$$

4.1.1 Pre-Normalization of Pixel Coordinates

The method described here to find the homography matrix $\boldsymbol{H}$, is also known as the Direct Linear Transform, DLT. Hartley and Zissermann [15] strongly recommend to normalize the pixel coordinates before the DLT-operation to condition the matrices. Without normalization the different elements of $\boldsymbol{A}$ may be of different magnitudes and increasing one term by a factor of 100 could lead to a huge change, while increasing another term by a factor of 100 could lead to a minimal change.

The normalization is related to the condition number of the equations used to create $\boldsymbol{A}$, a high condition number will increase the effect of noise in the measurement data, and by reducing the condition number, the effect of noise is also reduced. The method recommended by Hartley and Zissermann is to calculate a new set of coordinates with the centroid off all points as origin and where the average distance from origin is $\sqrt{2}$.

In the *Camera Calibration Toolbox for Matlab* [7] this is solved differently, and that method is implemented here. Coordinates are scaled and normalized around a centrepoint, but this is achieved by moving $\boldsymbol{K}$ to the left hand side of equation (4.3). Multiplying the pixel coordinates (u, v) with

$\boldsymbol{K}^{-1}$ as shown in equation (4.25) produces the new coordinates (x_i, y_i) centered around the principal point of the image plane and scaled to a fraction of the focal length $f = 1$.

$$K^{-1}\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\frac{1}{f_x} & 0 & \frac{-O_x}{f_x}\\ 0 & \frac{1}{f_y} & \frac{-O_y}{f_y}\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\frac{u-O_x}{f_x}\\ \frac{v-O_y}{f_y}\\ 1\end{bmatrix} = \begin{bmatrix}x_i\\y_i\\1\end{bmatrix} \tag{4.25}$$

Equation (4.26) shows the homography expression when pixel coordinates are normalized to a homogenous image plane at $f = 1$. Equations (4.7) through (4.15) are still valid, but when vector $\boldsymbol{h}$ is reshaped to matrix $\boldsymbol{H}$, the $\boldsymbol{RT}$-matrix is available without any further operations as shown in equation (4.27). The accuracy of the measurements are significantly increased by using this method.

$$\lambda\begin{bmatrix}x_i\\y_i\\1\end{bmatrix} = \begin{bmatrix}r_{11} & r_{12} & t_x\\ r_{21} & r_{22} & t_y\\ r_{31} & r_{32} & t_z\end{bmatrix}\begin{bmatrix}x_w\\y_w\\1\end{bmatrix} \tag{4.26}$$

$$\boldsymbol{H} = \begin{bmatrix}\boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t}\end{bmatrix} \tag{4.27}$$

4.1.2 Inhomogenous Solution for Homography

In *Robotics, Vision and Control* by Peter Corke [17], a different solution is proposed to solve the homography. Element h_{33} is set to 1 and the equation shown in (4.28) is obtained when inserting (4.10) into (4.8) and (4.9).

$$\begin{bmatrix}x_{w1} & y_{w1} & 1 & 0 & 0 & 0 & x_{w1}\cdot u_1 & y_{w1}\cdot u_1\\ 0 & 0 & 0 & x_{w1} & y_{w1} & 1 & x_{w1}\cdot v_1 & y_{w1}\cdot v_1\\ x_{w2} & y_{w2} & 1 & 0 & 0 & 0 & x_{w2}\cdot u_2 & y_{w2}\cdot u_2\\ 0 & 0 & 0 & x_{w2} & y_{w2} & 1 & x_{w2}\cdot v_2 & y_{w2}\cdot v_2\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ x_{wn} & y_{wn} & 1 & 0 & 0 & 0 & x_{wn}\cdot u_n & y_{wn}\cdot u_n\\ 0 & 0 & 0 & x_{wn} & y_{wn} & 1 & x_{wn}\cdot v_n & y_{wn}\cdot v_n\end{bmatrix}\begin{bmatrix}h_{11}\\h_{12}\\h_{13}\\h_{21}\\h_{22}\\h_{23}\\h_{31}\\h_{32}\end{bmatrix} = \begin{bmatrix}u_1\\v_1\\u_2\\v_2\\ \vdots\\u_n\\v_n\end{bmatrix} \tag{4.28}$$

$$\boldsymbol{A}\cdot\boldsymbol{h} = \boldsymbol{f} \tag{4.29}$$

Equation (4.28) is compressed in (4.29). A simple way of finding $\boldsymbol{h}$ is by multiplying with $\boldsymbol{A}^{-1}$, but this will only work for 4 correspondences, as this is the only situation where $\boldsymbol{A}$ is square and the inverse can be computed. The pseudo inverse is used, as shown in (4.30). Before the $\boldsymbol{h}$ -vector can be reshaped back to a matrix, H_{33} is set to 1 and added to the end of the vector. From this point the same operations are applied as when using DLT to find the $\boldsymbol{RT}$-matrix.

$$\boldsymbol{h} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{f} \tag{4.30}$$

This method is referred to as the inhomogeneous solution by Hartley and Zissermann [15], and they generally do not recommend using it due to the fact that for $z_w = 0$ there exists no solution. Experiments have shown that the inhomogenous solution provides the same results as the DLT-method in the operational area used while testing in the laboratory.

4.2 Error Handling

In section 3.6 the function to collect features and stack them in an orderly fashion was described. If a point that is not a corner is detected (noise), it will be collected as well. If kept in the array of features this will destroy the results, as the pixel coordinates and world coordinates will no longer be matched.

For each operation where the corners along a line are counted and stacked, an operation is performed to verify that only the correct features are stored before the next sequence of the program starts. The expected number of features to find is known and checked against the number of stacked features. If more corners have been found, the excess number is used to remove the same number of features from the stack. The noise is assumed to always have a smaller area than the corner-indications, and therefore the features with smallest area are removed.

When the homography is calculated, an initial homography, $\boldsymbol{H}$, is computed from 4 points around the point used to draw the axis lines shown in figure 3.8. This homography is multiplied with the known coordinates (x_w, y_w), to create an array of which image coordinates, (x_i, y_i), where correspondences should be found. The distance between current and expected image coordinates is used to discard points not supporting the homography. With a new array of correspondences, the final homography is computed.

CHAPTER 5

Test setup

5.1 Equipment

5.1.1 Camera System

During testing and development of the software, a Basler ace acA2500-14um [4] USB3 camera has been used. This is a 2592x1944 USB3.0 monochrome camera with a CMOS image sensor and rolling shutter. The camera is rated with 14 fps image acquisition rate, images are are acquired with 8-bit depth.

Brackets for mounting the Basler camera on a desktop stand and for general mounting has been produced by modeling the holder according to drawings from Basler, and using 3D-printers to produce the plastic brackets. This allows the camera to be mounted in different settings with the camera pointing down, allowing reference patterns to be positioned in the camera's field of view.

A Computar M1214-MP objective lens with a focal length of 12mm is fitted to the camera. Aperture opening can be adjusted from $f = 1.4$ to fully closed.

5.1.2 Inertial Measurement Unit

A MicroStrain 3DM-GX3® -25-OEM inertial measurement unit (IMU) is used to verify the angle of the pattern in tests performed in a desktop setting.

The 3DM-GX3® -25-OEM uses MEMS sensor technology to provide a miniature Attitude Heading Reference System [5].

5.1.3 Computer and Software

The software is developed in LabVIEW by National Instruments, see appendix A for code. The software is executed on a regular computer with the following relevant specifications:

- Intel® Core™ i5-660 CPU @ 3.33 GHz
- 8.00 GB DDR 1333 MHz RAM
- Icy-box PCI-e USB 3.0 expansion card
- LabVIEW 2012 with Vision 2013 package
- Windows 7 64-bit

5.2 Desktop Setup

Having a simplified setup for testing in a desktop environment allows for rapid testing of software and troubleshooting problems instantly. The software has been developed and initial tests performed on the desktop setup.

Figure 5.1 shows the system used for verifying the software in a desktop environment. The pattern is printed on a regular laser printer and glued to a MDF-board together with the IMU in the lower left corner, providing accurate readouts of the angle of the board that can be used for initial verifications of the roll and pitch angle measurements from the camera system.

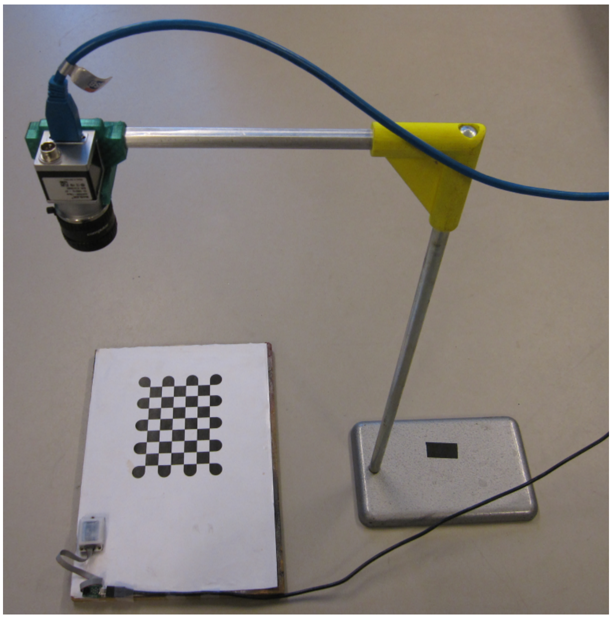


Figure 5.1: Desktop test setup

5.3 Stewart Platform

A Stewart platform is a type of parallel robot with six degrees of freedom. It is named after D. Stewart who published the design of a six-axis platform in 1965 [18]. The original layout was first used by V. E. Gough in 1954 [19]. It is sometimes referred to as a Gough-Stewart platform, but is commonly known as a Stewart platform.

Six actuators are fixed in pairs to a fixed base frame. Figure 5.2 shows how the actuators cross to three fixation points on a movable frame. A pattern positioned on the movable frame can be moved freely in all 6 degrees of freedom, and the patterns current position and angles can be retrieved from the control system of the Stewart platform. Information about the Stewart platform is retrieved over UDP, allowing the computed pose to be compared directly to the actual pose. In previous tests at the University of Agder, the heave position received has been verified to be accurate within 10 μm.

Two Stewart platforms are available, both delivered from Bosch Rexroth. The Stewart platform in figure 5.2 is an EMotion-1500, capable of a 1500 kg load on the movable frame. In addition there is an EMotion-8000 availabe, capable of a 8000 kg load on the movable frame.

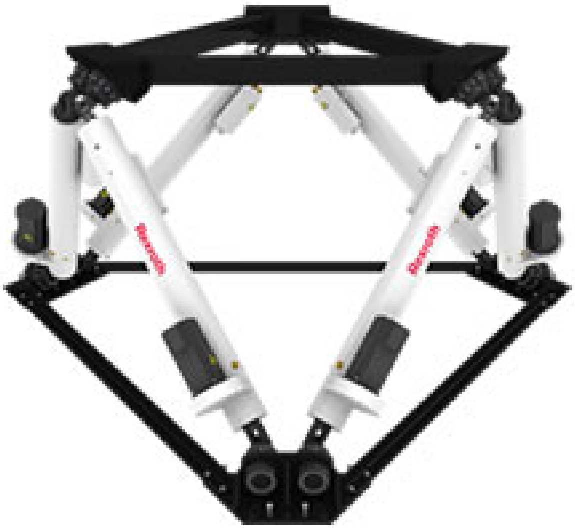


Figure 5.2: EMotion-1500 Stewart platform [20]

The two Stewart platforms are mounted next to each other facing the same direction. An anglebar is mounted along one of the sides of the EMotion-8000 platform and a 3D-printed bracket holding the camera is fastened to the end of the angle bar over the EMotion-1500 platform where the pattern has been glued to a piece of aluminium sheet metal. The plate used in the desktop tests is also fastened to the Stewart platform to get IMU readings of the roll and pitch angles. Figure 5.3 shows the setup.

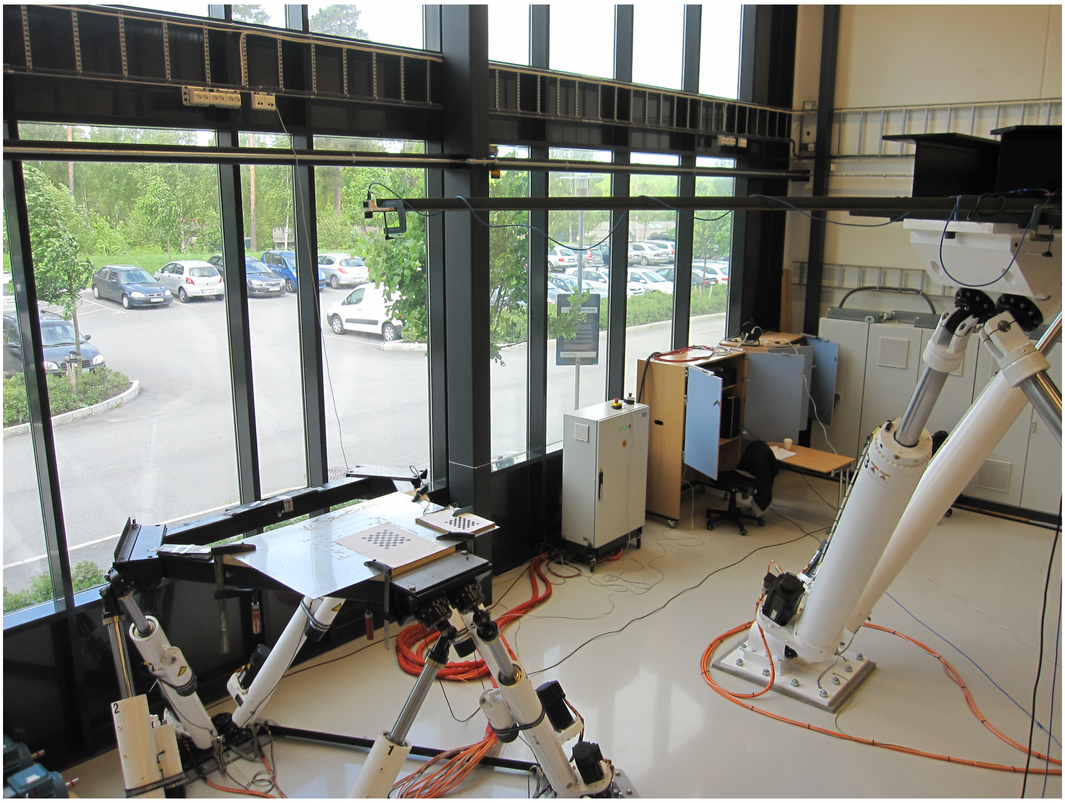


Figure 5.3: Stewart test setup

To position the camera over the pattern, the EMotion-8000 is shifted slightly in its x- and y-direction until the pattern is centered in the field of view for the camera. The platform is then elevated until the first actuator reaches its mechanical endstop to maximize the distance between camera and pattern.

Figure 5.4 shows an image acquired in this setup. A blue square indicate the ROI where the software has detected the pattern, red and green lines indicate the axis lines of the pattern. Between the aluminium sheet and the Stewart platform, 2"x8" pieces of wood has been fitted to keep the sheet level with the platform. Aluminium sheet, wood pieces and mdf-plate with IMU is all kept in position by clamping them to the Stewart platform.

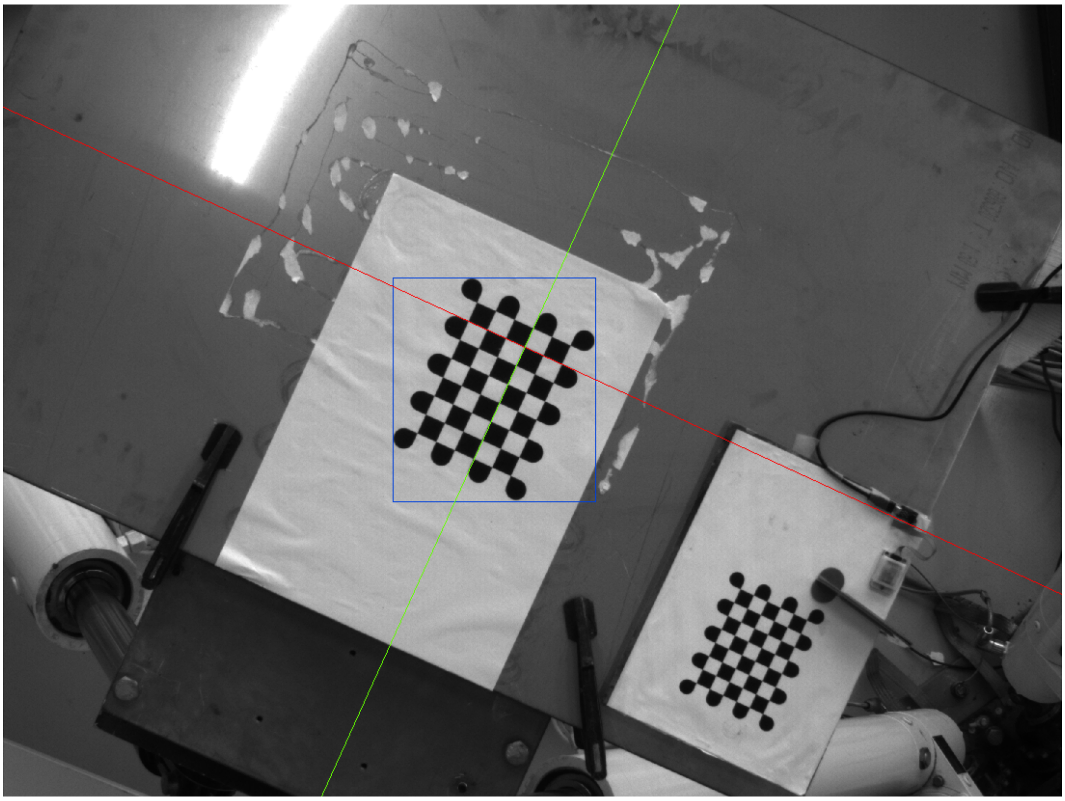


Figure 5.4: Input image in the Stewart setup

CHAPTER 6

Experiments

The software has been developed using a desktop test system, and initial experiments performed on it. With functioning software the Stewart platforms have been used to create more accurate tests where the position and orientation of the pattern is known at all times.

6.1 Desktop Tests

6.1.1 Impact of Unfocused Image

For a pinhole camera model all objects are imaged focused regardless of their distance to the image sensor. When a lens is used to collect more light, a problem may arise with the focus depth of the lens. The focus depth is the range of distances at which objects are acceptably focused.

For a large opening/aperture, the focus depth will be small, but the amount of light reaching the sensor is high, which allows for higher framerates/shorter acquisition time. To increase the focus depth, the aperture opening must be reduced, approximating a pin-hole camera, but at the cost of reducing the amount of light reaching the sensor.

The goal is to find the balance between enough light reaching the sensor, and achieving a focus depth large enough that features of the pattern can be detected over the operational range of the Stewart platforms. A test has been performed on the desktop to see what effect an unfocused image has on the measurements.

Figure 6.1 shows the variation in measured distance between camera and pattern when the focus varies as shown in the two images in figure 6.2. The test starts with a focused image containing 48 corners at 15 mm distance in the pattern. The image is brought out of focus and back again by using the focus adjustment ring on the lens. At approximately 450 mm distance between camera and pattern, there is a 1.6 mm change in measured z-distance when the image comes out of focus.

The change is 0.4% of the distance, and from this it can be said that it is not crucial to have a focused image to be able to compute the distance between camera and pattern reasonably accurately. This allows for using a larger aperture opening and shorter acquisition time, which reduces the effect of image skewing.

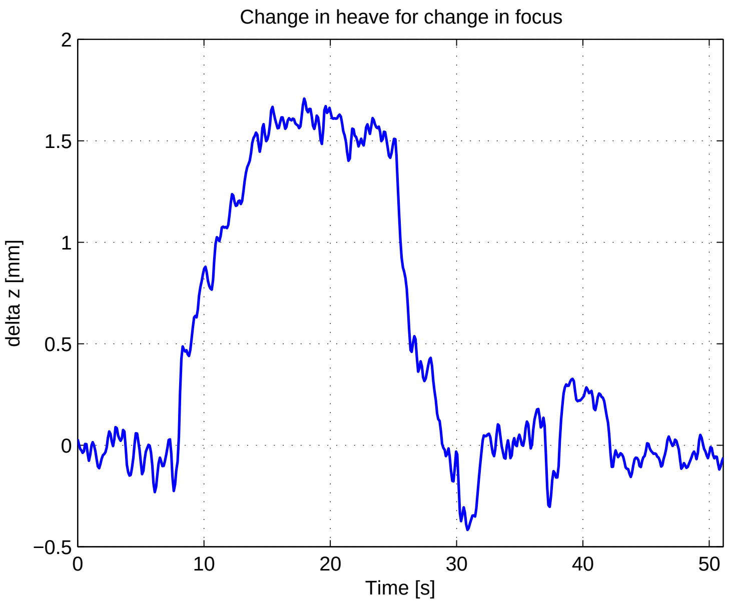


Figure 6.1: Change in z-distance for focused and unfocused image

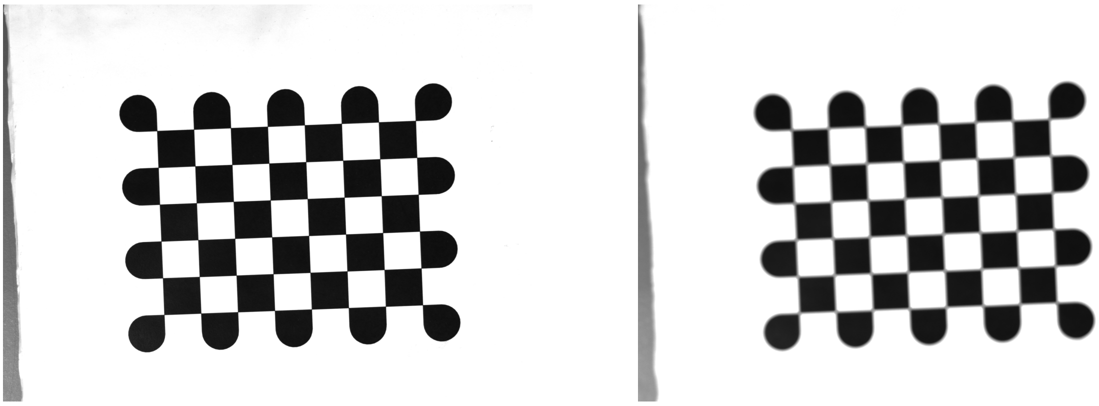


Figure 6.2: Focused and unfocused image

6.1.2 Image Skew

In the desktop experiments, some problems have been observed when the pattern is moved quickly. Figure 6.3 shows a snapshot of the image acquired by the camera while a pattern is moved from right to left in the image. The pattern is not the same as used previously (only 8 corners, but with the same dimensions) allowing for an image to be acquired more easily while pattern is being moved.

As expected, the horizontal lines are fairly lined up with the edges of the image. However, this is not the case for the vertical lines which are expected to be normal to the horizontal lines; here they are skewed. Also, the horizontal lines are sharp, whereas the vertical are not; this is caused by motion blur.

The reason for the skewing of the image is that the camera used has a CMOS-sensor with a rolling shutter. A rolling shutter will not read all pixel intensities in the same operation, but it will roll through lines of pixels in the sensor, allowing the scene to change before all pixels have been captured to form an image. Using shorter exposure time can reduce the skewing.

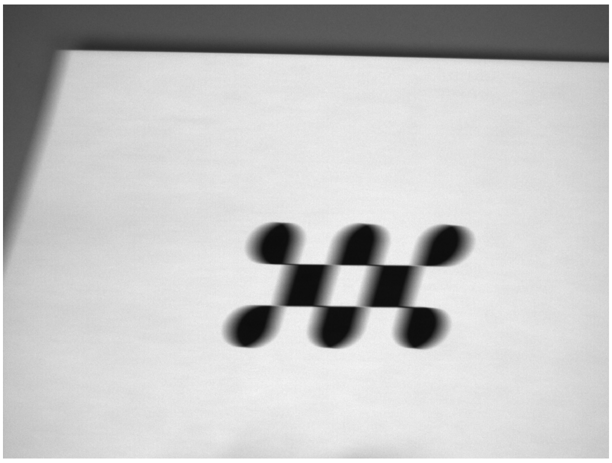


Figure 6.3: Skewed image

6.1.3 Accuracy

Figure 6.4 shows two charts with the roll and pitch angles recorded by the IMU, roll and pitch angles computed by the vision system and the deviation when pattern is moved by hand to ±15° for one degree of freedom at the time.

At 0° the board is resting on the desk while the camera is mounted in the test-jig. One edge of the board is kept in contact with the desk at all times while the board is rotated. The IMU provides reliable and accurate feedback on the actual angle of the board. For the roll test the highest recorded deviation is 0.26°, while for the pitch test the highest recorded deviation is 0.82°. For both instances the highest deviation occurs as the angle is being changed.

Translations have been verified using a ruler and moving the pattern along an edge for x and y, and by moving the camera fixture a measured quantity up/down for z to give an indication of the quality of the computed results. Both angles and translations have been found to be reasonably accurate in the desktop tests, and more accurate comparisons are performed on the Stewart platforms.

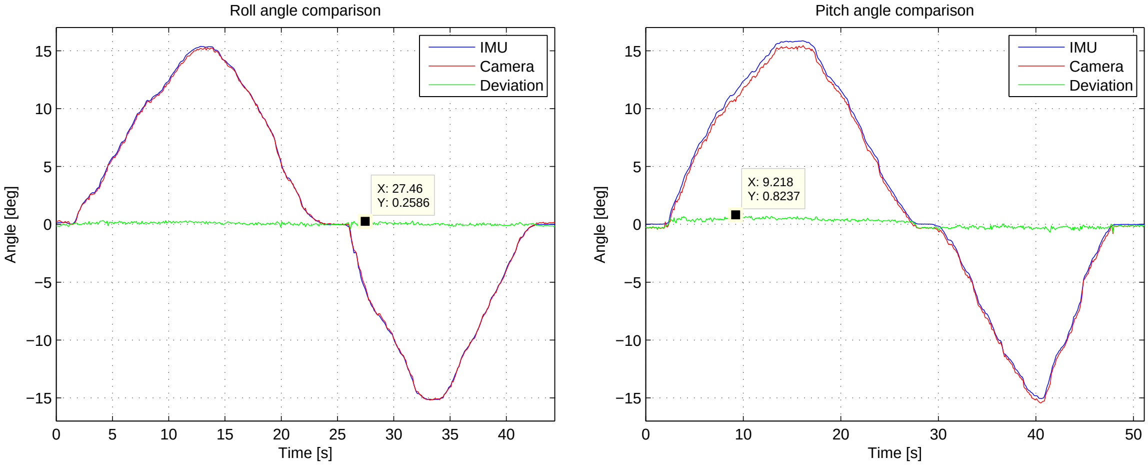


Figure 6.4: Angle measurements in desktop setup

6.2 Stewart Platform Tests

Table 6.1 and 6.2 show an overview of the experiments that have been performed to test the accuracy of the vision-based measurements. In a measurement system there will be static errors, dynamic errors and stochastic errors. Static or systematic errors are the repeatable errors which can be calibrated for. Dynamic errors are the errors occuring when the system is not able to pick up what is happening in the scene, and are linked to the bandwidth of the system. The stochastic errors are the noise in the signal.

Tests for heave accuracy and roll/pitch accuracy are done separately as the Stewart platform will rotate around its centre of rotation, located at the plane intersecting the couplings between actuators and top frame. If these tests were to be performed simultaneously the Stewart platform would either have to rotate around the center of the pattern, or the center of the pattern would have to be positioned at the centre of rotation for the Stewart platform. It is far more practical to test the quality of the rotation matrix and the translation vector separately.

Data is read from the computer controlling the Stewart platform over a user-defined protocol (UDP), and compared directly to the values computed from the acquired images. Prior to the experiments, the camera and IMU measurements are reset to measure from zero. For the heave experiments, positive position means higher distance between pattern and camera.

Table 6.1: Static experiments

Experiment #	Heave position [mm]	Roll position	Pitch position
1	0	0°	0°
2	400	0°	0°
3	-400	0°	0°
4	0	15°	0°
5	0	-15°	0°
6	0	0°	15°
7	0	0°	-15°

Table 6.2: Dynamic experiments

Experiment #	Heave amplitude [mm]	Roll amplitude	Pitch amplitude	Frequency [Hz]
1	400	0°	0°	0.050
2	400	0°	0°	0.125
3	0	15°	0°	0.050
4	0	15°	0°	0.125
5	0	0°	15°	0.050
6	0	0°	15°	0.125

6.2.1 Static Experiments

In all the static experiments the Stewart platform is moved between the positions defined in table 6.1. The platform is left in each position until 1000 samples have been recorded and mean value and variance are computed. Mean value and sample variance are found using the sample variance function in LabVIEW, which uses (6.1) and (6.2) to find these values.

$$\mu = \sum_{i=0}^{n-1} \frac{x_i}{n} \tag{6.1}$$

Where:

- μ is the mean value
- n is the number of samples
- x_i is a datapoint

$$\sigma = \sqrt{\sum_{i=0}^{n-1} \frac{(x_i - \mu)^2}{n-1}} \tag{6.2}$$

Where:

- σ is the sample variance

The heave, roll and pitch measurements are looked into one at the time. When both Stewart platform are in their neutral positions with platforms elevated to the center of their working areas, there is approximately 2.3 meters between camera and pattern.

Heave

In figure 6.5 the raw measurements when pattern is translated along the z-axis are shown on the left. Experiment starts at 0 mm heave, and progresses through the 400 mm (away from camera) and -400 mm positions (towards camera).

Deviation between Stewart platform position and position computed by the measurement system is shown on the right. The peaks when platform changes position comes from the dynamic errors and is explained in section 6.2.2.

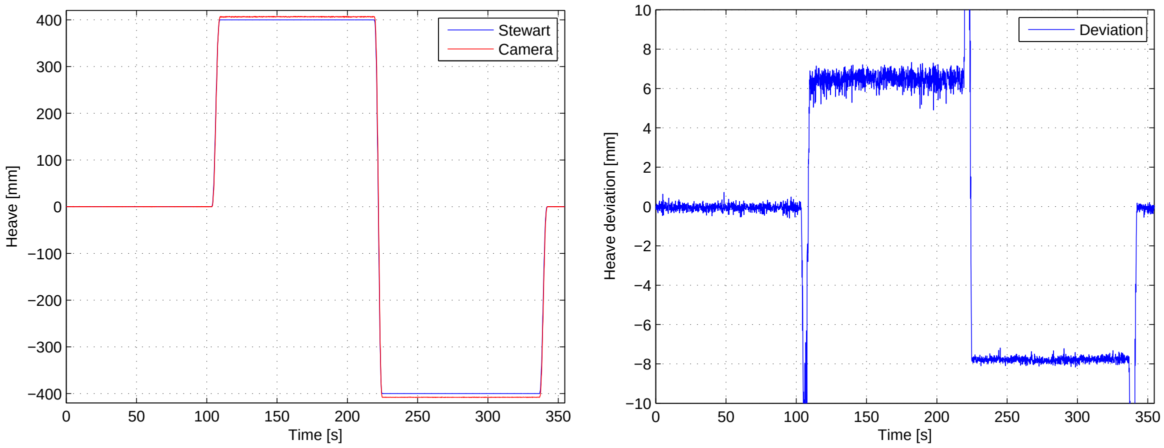


Figure 6.5: Raw heave measurements and deviation

Table 6.3 shows the raw results from the heave experiments. Note that the variance seems to be connected with the distance between camera and pattern. For a lower distance the variance is decreased and vice versa for a higher distance. At these distances the region of interest varies between approximately 425x455 pixels and 600x650 pixels, the resolution of the ROI seems to have an impact on the variance in the measurements.

Table 6.3: Static results - raw heave

Experiment #	Position [mm]	Heave mean [mm]	Heave variance [mm]
1	0	-0.01	0.148
2	400	406.5	0.356
3	-400	-407.7	0.127

Deviation seems to be approximately symmetrical around zero. By multiplying the heave value with a calibration factor, k_{heave}, the heave measurement is corrected for systematic errors. To find the calibration factor, the position of the Stewart platform is divided by the mean value of the measurements for the two offsets. The average value of these are used as shown in equation (6.3).

$$k_{heave} = \frac{\frac{400}{406.5} + \frac{-400}{-407.7}}{2} \approx 0.983 \tag{6.3}$$

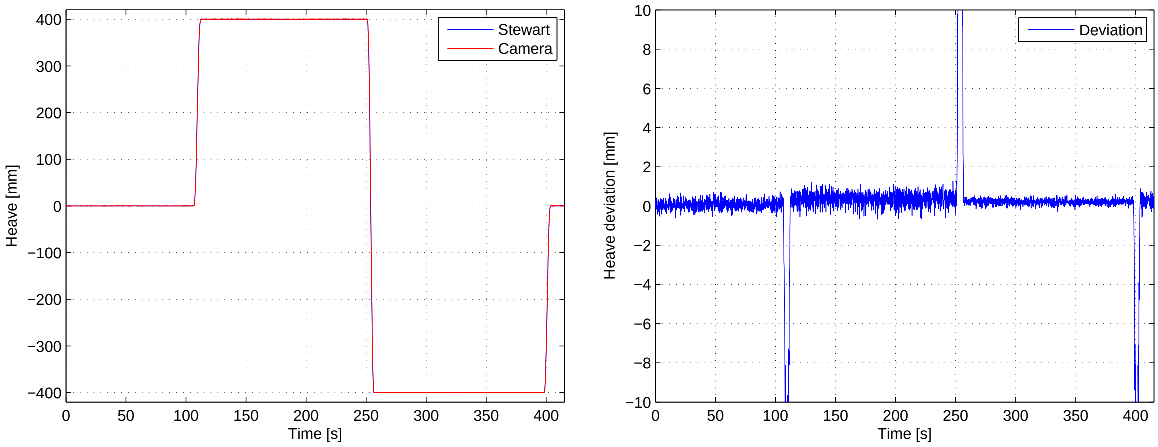


Figure 6.6: Calibrated heave measurements and deviation

Figure 6.6 shows the measurements from a repetition of the experiments where the heave measurement is calibrated. Results are shown in table 6.4, mean values deviate 0.4 mm for the 400 mm position and 0.2 mm for the -400 mm position.

The variances measured are slightly different from when the raw measurements were taken, the light conditions changed between experiments and seems to have affected the variance. The link between variance and distance between camera and pattern can still be observed both in the deviation graph and in the variance in the results.

Table 6.4: Static results - calibrated heave

Experiment #	Position [mm]	Heave mean [mm]	Heave variance [mm]
1	0	-0.07	0.203
2	400	400.4	0.298
3	-400	-399.8	0.122

Roll

In figure 6.7 the raw measurements when pattern is rotated around its x-axis is shown on the left. Deviation between Stewart platform angle and angle computed by the measurement system is shown on the right.

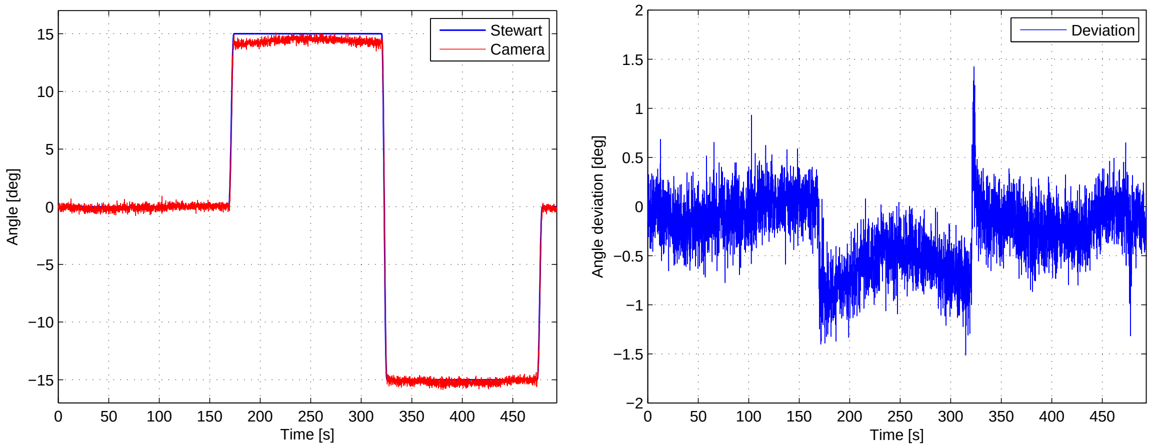


Figure 6.7: Raw roll measurements and deviation

Table 6.5 shows the raw results from the experiments where the pattern is rotated around its x-axis. The mean measurements deviate from the Stewart platform with -0.53° for the 15° position and -0.22° at the -15° position. Deviations appear to be slightly shifted in the negative direction. Equation (6.4) shows how the calibration factor, k_{roll}, is computed.

The pattern is mounted on the Stewart platform in a position where the pattern and Stewart platform will roll around approximately the same axis. Because of this the size of the ROI stays approximately the same from position to position and the variance changes little from position to position.

Table 6.5: Static results - raw roll

Experiment #	Position	Camera mean	Camera variance
1	0	-0.08°	0.228°
4	15°	14.47°	0.225°
5	-15°	-15.22°	0.225°

$$k_{roll} = \frac{\frac{15}{14.47} + \frac{-15}{-15.22}}{2} \approx 1.011 \tag{6.4}$$

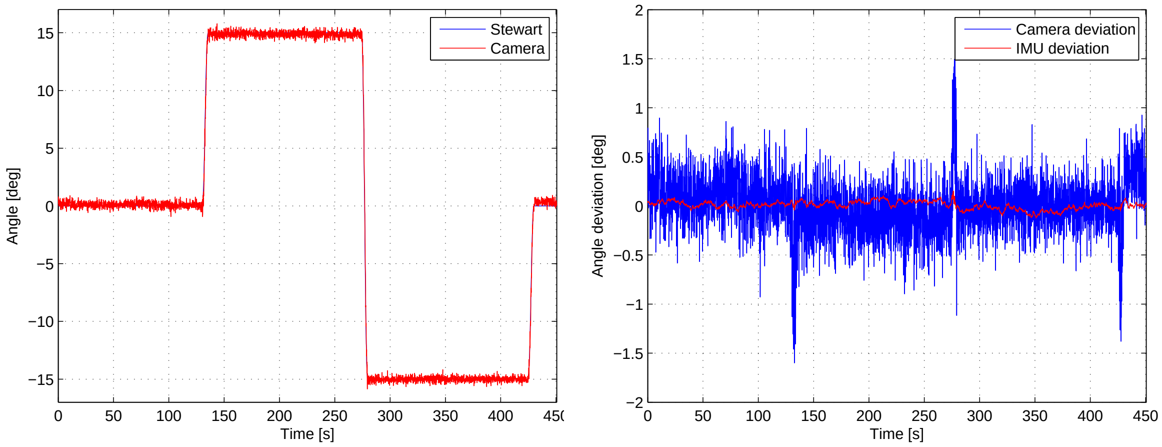


Figure 6.8: Calibrated roll measurements and deviation

Figure 6.8 shows the measurements from a repetition of the experiments where the roll measurement is calibrated, deviation between Stewart-platform and IMU has been added in the left side.

The mean values and variances for both camera and IMU measurements are shown in table 6.6. As with the heave experiments, some differences are seen in the variance between experiments where the raw and calibrated measurements were taken, as already mentioned likely caused by differing light conditions. Mean values after calibration are closer to the Stewart-platform position, deviating -0.11° for the 15° position and -0.01° for the -15° position.

The IMU measurements show mean values deviating 0.04° from the Stewart platform position, and significantly lower variance.

Table 6.6: Static results - calibrated roll

Experiment #	Position	Camera mean	Camera variance	IMU mean	IMU variance
1	0	-0.11°	0.243°	0.03°	0.026°
4	15°	14.89°	0.239°	15.04°	0.028°
5	-15°	-15.01°	0.204°	-15.04°	0.037°

Pitch

In figure 6.9 the raw measurements when pattern is rotated around its y-axis are shown on the left. Deviation between Stewart platform angle and angle computed by the measurement system is shown on the right.

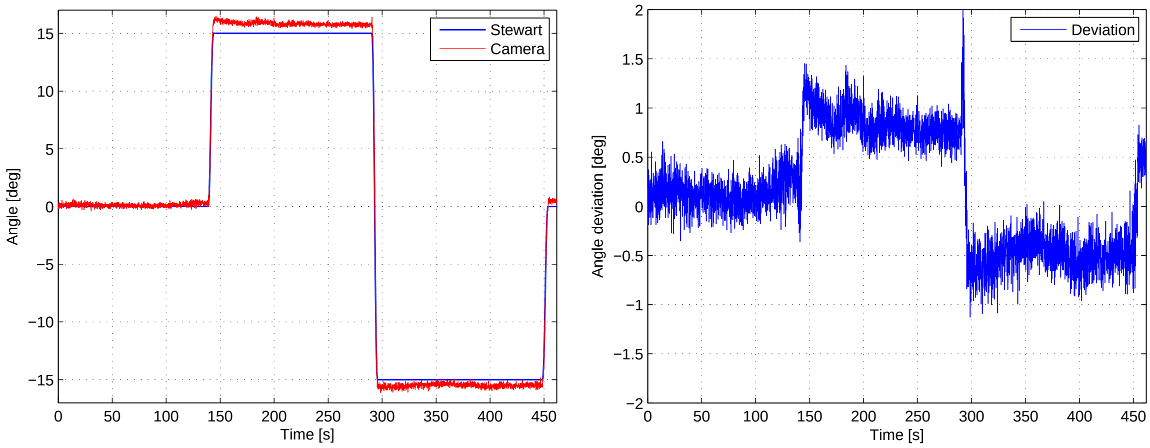


Figure 6.9: Raw pitch measurements and deviation

Table 6.7 shows the raw results from the experiments where the pattern is rotated around its y-axis. The mean measurements deviate from the Stewart platform with 0.8° at the 15° position and -0.47° at the -15° position.

Note that the variance here is lower than in the raw roll experiments. Equation (6.5) shows how k_{pitch}, used to calibrate the pitch measurements is computed.

Table 6.7: Static results - raw pitch

Experiment #	Position	Camera mean	Camera variance
1	0	0.10°	0.143°
6	15°	15.80°	0.148°
7	-15°	-15.47°	0.167°

$$k_{pitch} = \frac{\frac{15}{15.80} + \frac{15}{15.47}}{2} \approx 0.959 \tag{6.5}$$

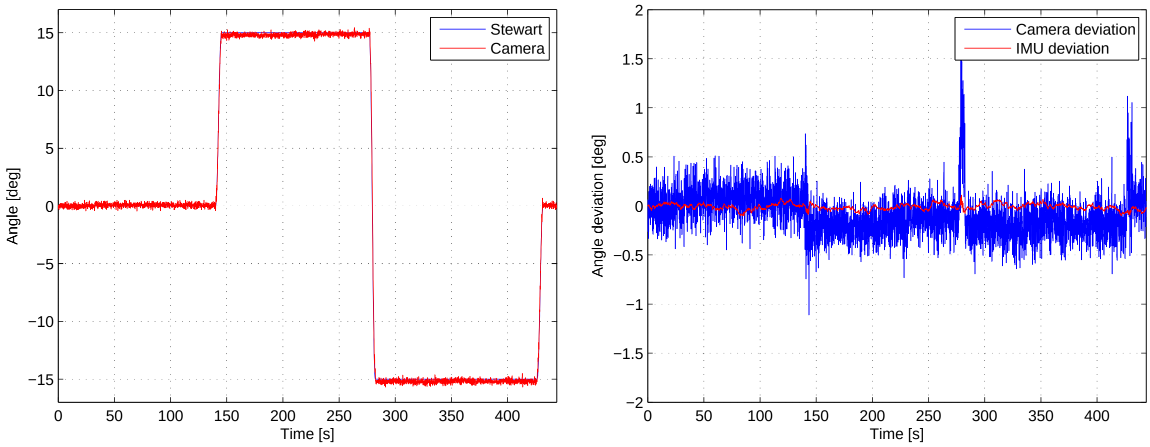


Figure 6.10: Calibrated pitch measurements and deviation

Figure 6.10 shows the measurements from a repetition of the experiments with calibrated pitch measurements on the right side. The left side shows the deviation between Stewart platform and the two sensors, camera measurements and IMU.

The mean values and variances for both camera and IMU measurements are shown in table 6.8. It can be seen that in general, the pitch variance is lower than the roll variance, likely caused by the fact that there are more points in the x-direction of the pattern. Mean values after the calibration deviate from the Stewart-platform position with 0.18° for the 15° position and -0.19° for the -15° position.

IMU measurements have mean values very close to the Stewart platform position and as for the roll experiments, a significantly lower variance.

Table 6.8: Static results - calibrated pitch

Experiment #	Position	Camera mean	Camera variance	IMU mean	IMU variance
1	0	0.05°	0.159°	-0.01°	0.033°
6	15°	14.82°	0.148°	14.98°	0.029°
7	-15°	-15.19°	0.157°	15.00°	0.029°

6.2.2 Dynamic Experiments

In the dynamic experiments the Stewart platform starts at rest and is commanded to follow a sinusoidal curve with the amplitude and frequency specified in table 6.2. Motion on the Stewart platform is stopped after the platform has done two full cycles at 0.05 Hz and three full cycles at 0.125 Hz. Measurements are calibrated for systematic errors.

Deviations found during the dynamic experiments will be part stochastic errors and part dynamic errors linked to the bandwidth of the measurement system.

Heave

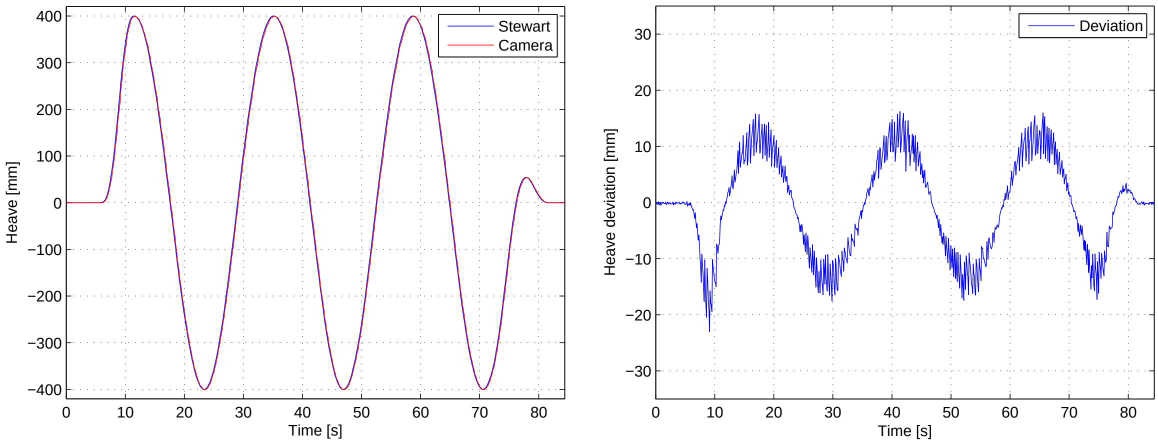


Figure 6.11: Dynamic heave measurements and deviation at 0.05 Hz

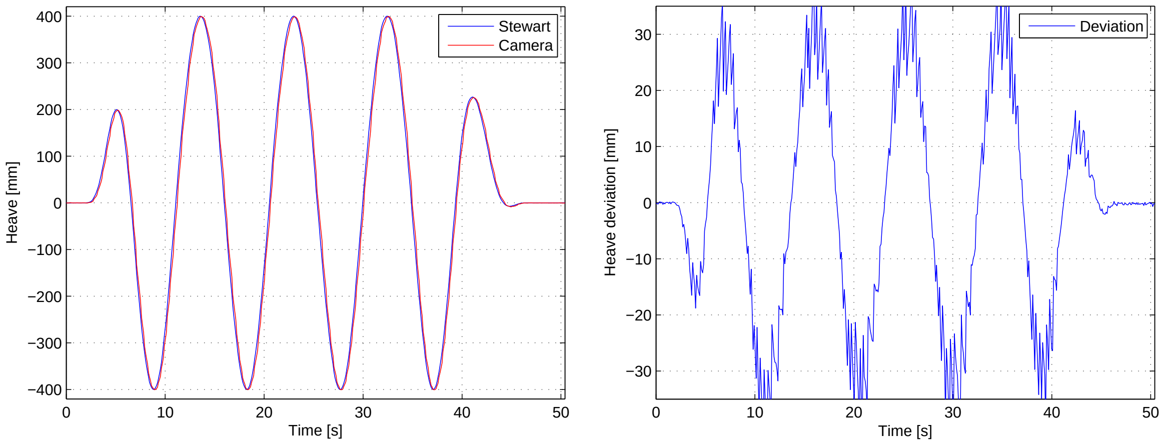


Figure 6.12: Dynamic heave measurements and deviation at 0.125 Hz

Figure 6.11 and 6.12 show measurements and deviation between Stewart and camera system for a sinusoidal heave motion with 400 mm amplitude and frequencies of 0.05 Hz and 0.125 Hz.

It is clear from the deviations that under dynamic conditions there is a large error. The largest error in the 0.05 Hz experiment is approximately -15 mm, occuring every time the position passes 0 mm when moving from -400 mm to 400 mm (moving away from the camera).

In the 0.125 Hz experiment the largest error is just below -30 mm, also occuring every time the position passes 0 mm when moving from -400 mm to 400 mm. For both experiments the dynamic error is highest when the platform has its highest velocity.

From the measurements on the left side of the figures, it seems like the measurement system is following the position of the Stewart platform accurately, the amplitude is correct, but there is a delay introduced, which leads to deviation actually being as large as can be seen in the figures when the velocity is high.

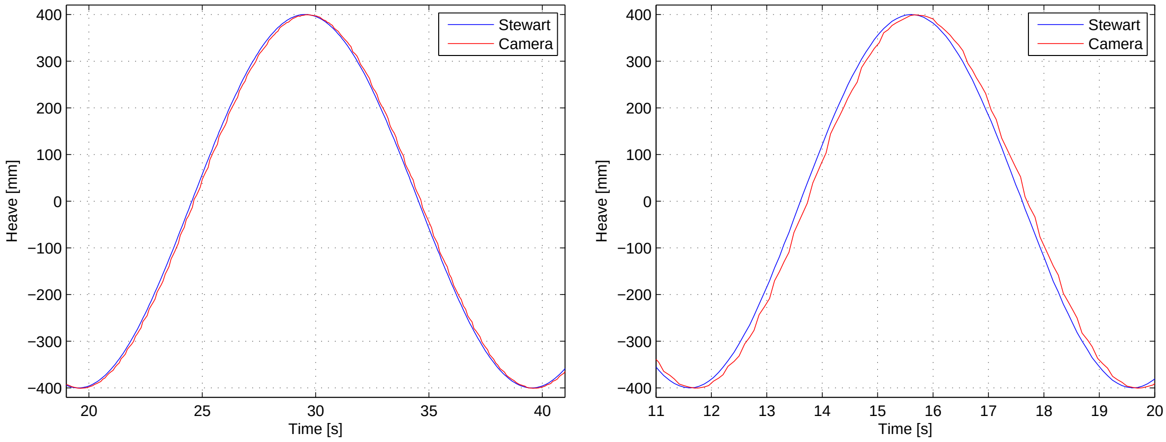


Figure 6.13: Zoomed in dynamic heave measurements - 0.05 Hz and 0.125 Hz

In figure 6.13 the measurements are zoomed in to show the lag for 0.05 Hz on the left side and 0.125 Hz on the right side. Time delay between Stewart platform and camera measurements are found at each zero crossing. The average of the delay is used to compute the phase-shift in the 0.05 Hz experiment in equation (6.6) and the 0.125 Hz experiment in equation (6.7).

$$\phi_{0.05Hz} = \frac{360^\circ}{20s} \cdot 0.092s = 1.66^\circ \tag{6.6}$$

$$\phi_{0.125Hz} = \frac{360^\circ}{8s} \cdot 0.106s = 4.77^\circ \tag{6.7}$$

The cycletime of the software is in the area of 80 ms=0.080 s. The delay between Stewart platform and camera measurements during the zero crossings are just above the cycletime. In dynamic heave experiments at 0.01 Hz the delay at zero crossing has been observed to also be in the range of the cycletime of the software, indicating that this might be caused by the update frequency of the software.

Roll

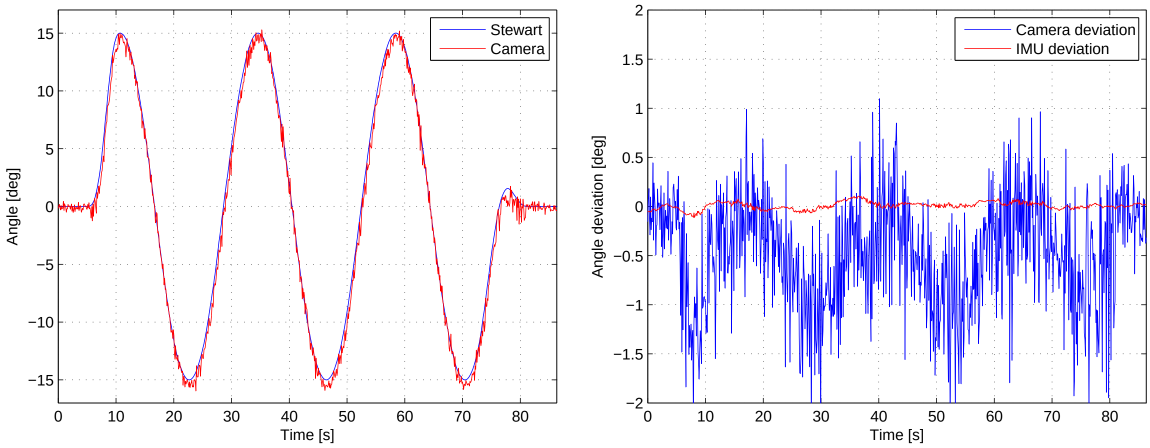


Figure 6.14: Dynamic roll measurements and deviation at 0.05 Hz

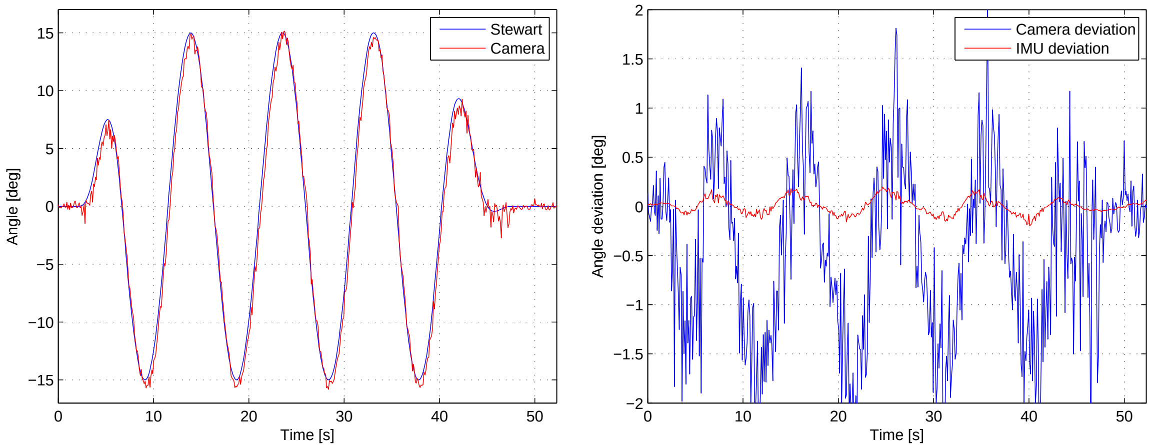


Figure 6.15: Dynamic roll measurements and deviation at 0.125 Hz

Figure 6.14 and 6.15 show measurements from the dynamic roll experiments on the right, and deviation between Stewart and the two sensors, camera measurements and IMU on the left for a sinusoidal roll motion with 15° amplitude and frequencies of 0.05 Hz and 0.125 Hz.

For both experiments there is a static deviation of approximately -0.5° in the camera measurements when the platform is moving. When platform is not moving the measured position is at zero.

Deviations indicate a connection to the velocity of the platform here also, in the 0.05 Hz experiment the error looks centered around the -0.5° deviation, and has an amplitude of approximately 1°. In the 0.125 Hz experiment the error also looks centered around the static deviation, with an amplitude increased to approximately 1.5°. Peaks in deviations occur as platform passes 0°, where it will have its highest velocity.

In the 0.125 Hz experiment the IMU error increases show similar behavoiour as the camera measurements. When the velocity is high there is a deviation with the same shape, but lower amplitude.

Pitch

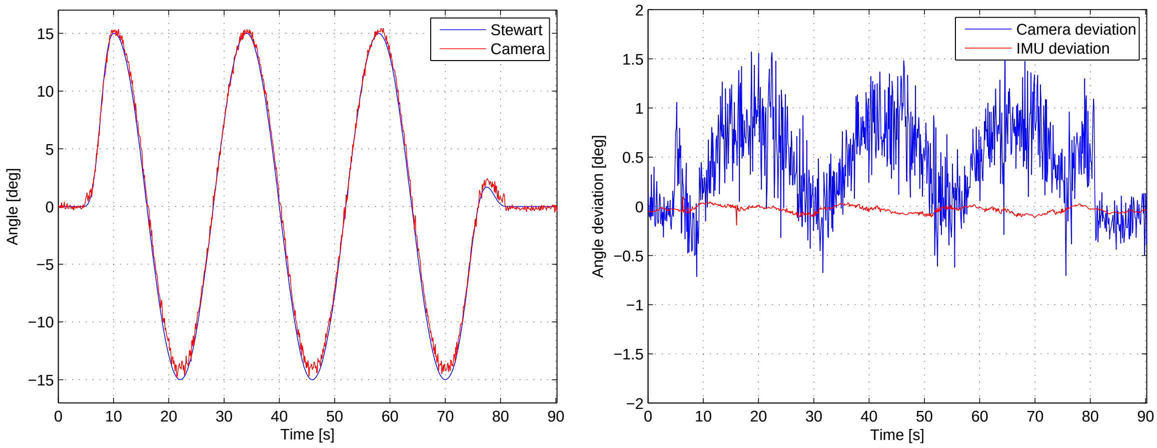


Figure 6.16: Dynamic pitch measurements and deviation at 0.05 Hz

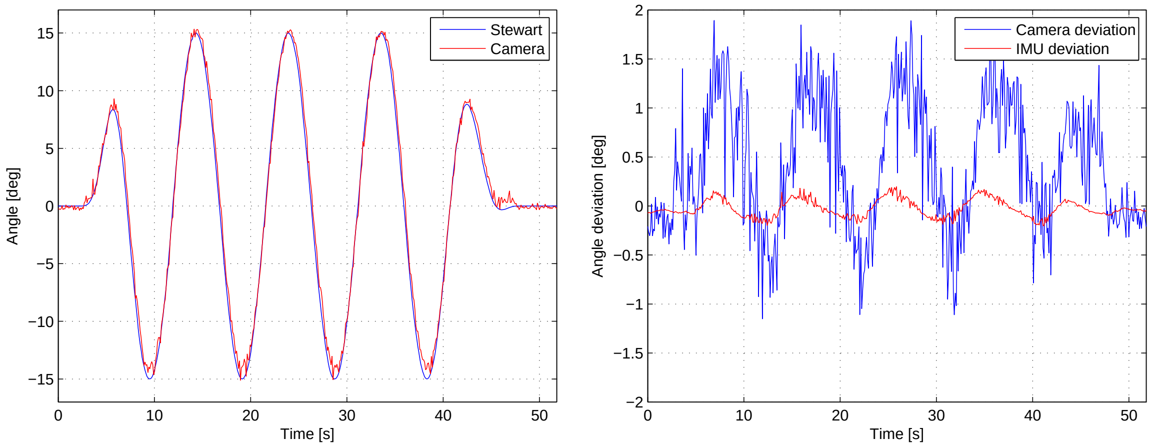


Figure 6.17: Dynamic pitch measurements and deviation at 0.125 Hz

Figure 6.16 and 6.17 show measurements from the dynamic pitch experiments on the right, and deviation between Stewart and the two sensors, camera measurements and IMU on the left for a sinusoidal pitch motion with 15° amplitude and frequencies of 0.05 Hz and 0.125 Hz.

For both experiments there is a static deviation of approximately 0.5°, similar to the offset for roll but with opposite sign.

As in the roll experiments, camera measurement deviation follows the velocity of the platform. In both the 0.05 Hz and 0.125 Hz experiment deviation is centered around the 0.5° static deviation, with an approximate amplitude of 0.5° at 0.05 Hz, and 1° at 0.125 Hz.

IMU deviation displays the same behaviour as in the roll experiments, especially in the 0.125 Hz experiments where the IMU measurements show the same sinusoidal deviation, only with a smaller amplitude.

6.2.3 Stochastic Errors

In both the static and dynamic experiments the measurements show large stochastic errors or noise, especially compared to the IMU measurements. Filtering the measurements with a low-pass filter will remove the noise, but at the cost of introducing a lag. In the previous figures for 0.125 Hz experiments it is possible to see that the camera measurements are already lagging behind the Stewart platforms position. Introducing a filter would further increase this lag, leading to larger deviations between Stewart position and camera measurement.

Figure 6.19 shows the Stewart and camera roll measurements when a Butterworth low-pass filter is applied to the 0.125 Hz roll experiment. Filter is applied offline, so results can be compared directly to figure 6.15. Because of the lag in the measurements, deviation between the two is large.

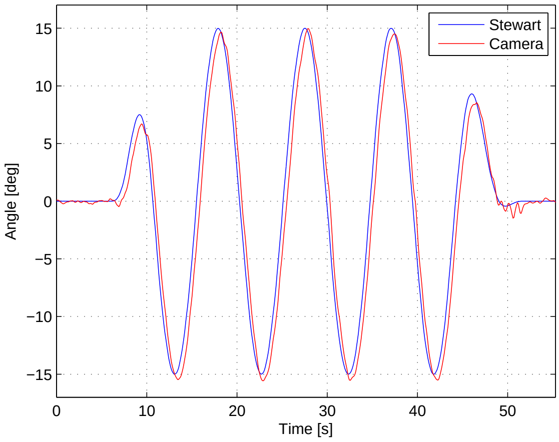


Figure 6.18: Filtered roll measurements

In figure 6.19 the camera measurement has been shifted with -0.5 seconds to reduce/remove the lag. Compared with 6.15 the same pattern can be seen in the deviation, a steady offset/deviation of 0.5°. The amplitude of the sinusoidal deviation is not correct because the measurements has been shifted, but it shows that the measurements from the camera has potential to be as accurate as the IMU if signal is filtered and dynamic errors are removed.

To be able to filter the signal without introducing as much lag as in this example, the sampling frequency of the camera should be increased and general cycletime of the software should be brought down. To reduce the dynamic errors a camera with a global shutter should be used.

Sample rate of the sensors in the IMU is 30 kHz, giving the IMU time to filter the signals properly, as can be seen in the figures with camera and IMU deviation to Stewart platform. In the current setup the camera provides new images at 14 Hz and the software executes at approximately 10-12 Hz.

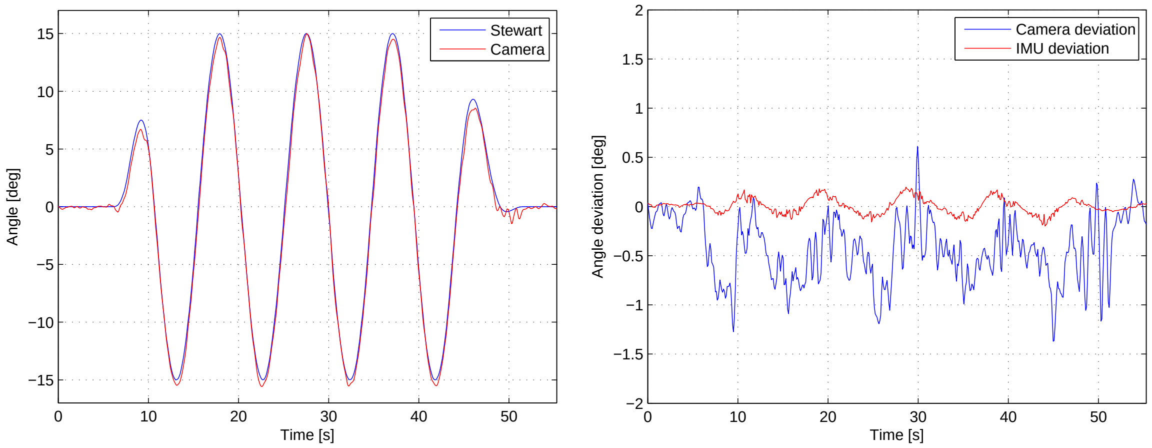


Figure 6.19: Filtered and phase-shifted roll measurements

6.2.4 Image Skew Impact on Homography

The skew effect caused by the rolling shutter will make the pixel coordinates of a corner be in a different position from where it is expected to be in relation to corners imaged early in the acquisition process. This causes problems computing all 6 degrees of freedom accurately under dynamic conditions. Figure 6.20 illustrates this by showing the deviation from the camera and IMU to the Stewart platform pitch angle during the 0.125 Hz dynamic heave experiment.

The IMU deviation shown in red indicate that there might be a slight movement in the pitch angle, but not nearly as much as the camera deviation, which follows a sinusoidal curve, centered around approximately 0.3° and an amplitude of close to 1°.

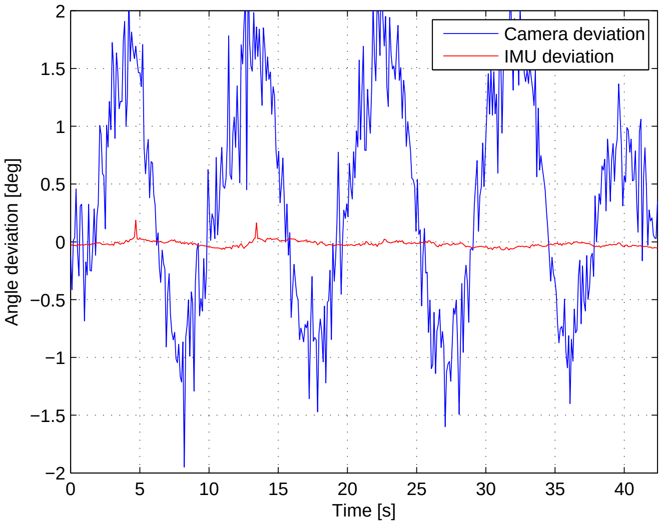


Figure 6.20: Pitch angle during 0.125 Hz dynamic heave experiment

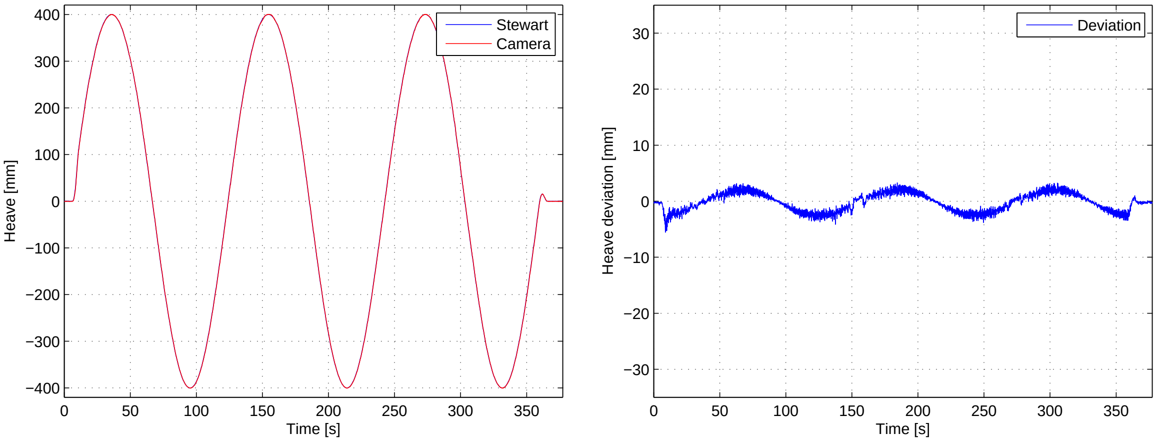


Figure 6.21: Dynamic heave measurements and deviation at 0.01 Hz

Figure 6.21 show the measurements from a dynamic measurement at 0.01 Hz, a frequency below the chosen interval. Compared to the dynamic heave measurements for 0.05 Hz and 0.125 Hz, the deviation is reduced as the velocity of the platform is reduced.

The skew impact on the homography is reduced considerably, figure 6.22 show the pitch angle deviations during the 0.01 Hz dynamic heave experiment. The deviation signal is dominated by noise, but a sinusoidal deviation pattern can be observed. The amplitude is lower than in figure 6.20 for the 0.125 Hz dynamic heave experiment.

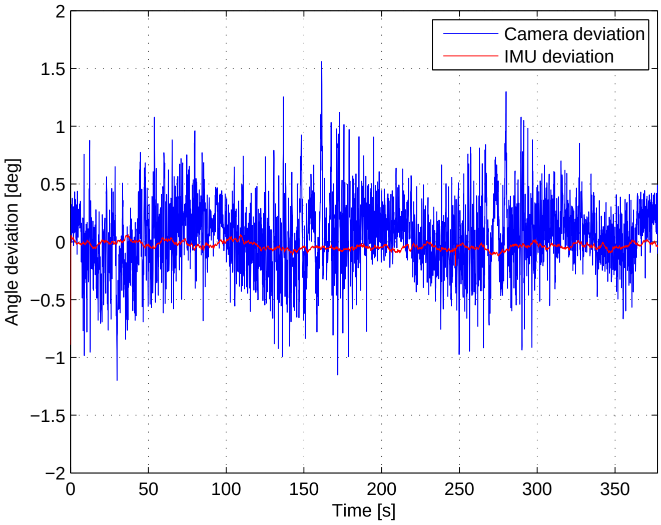


Figure 6.22: Pitch angle during 0.01 Hz dynamic heave experiment

6.2.5 DLT vs Inhomogeneous Solution

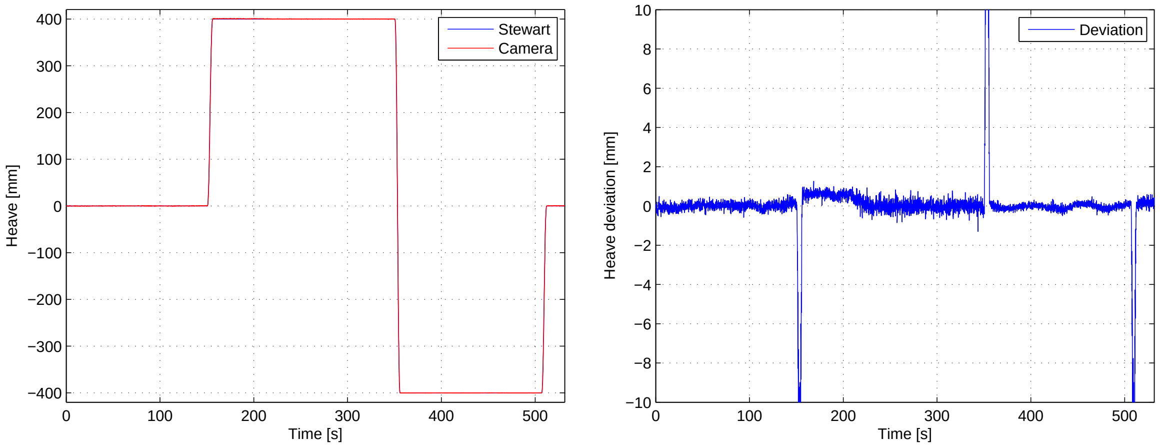


Figure 6.23: Static heave measurement solved with DLT

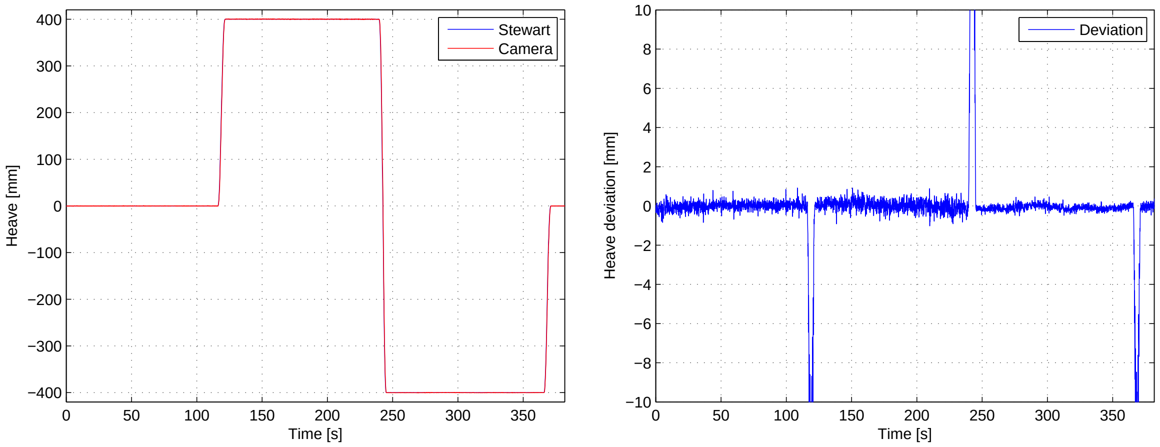


Figure 6.24: Static heave measurement solved with inhomogeneous solution

Figure 6.23 and 6.23 shows the measurements from two repetitions of the static heave experiments. In the first experiment the homography is solved using the direct linear transform, DLT. Homography of the second experiment is solved using the inhomogeneous solution. Mean and variance values are listed in table 6.9. Differences in mean and variance values are very small, and credited to differing light conditions.

Table 6.9: Static results - solving with DLT and inhomogeneous equations

Experiment #	Position[mm]	DLT mean[mm]	DLT var.[mm]	Inhom. mean[mm]	Inhom. var.[mm]
1	0	0.01	0.17	-0.02	0.19
2	400	400.03	0.27	400.05	0.28
3	-400	-400.06	0.14	-400.09	0.14

6.3 Analysis of Results

The vision-based measurement system show good accuracy under static conditions, but suffers from large deviations under dynamic conditions. The desktop tests has showed that all 6 degrees of freedom are computed fairly accurate when compared to ruler measurements for the translations, and uncalibrated roll and pitch measurements are compared to IMU readings.

Angle measurements in the desktop setup indicate problems when pattern is moved, these problems are verified by the dynamic experiments on the Stewart platform where there is a strong correlation between deviations and pattern velocity.

Deviations seen in the dynamic experiments would likely be reduced if a camera with a global shutter was used, ensuring that the scene is captured at once, not capturing the movement in the scene one row of pixels at the time, as is the case in the current setup with a rolling shutter. The effect of this is that the pattern will appear to the measurement system as bent, and not as a planar surface which it actually is. Figure 2.4 show the corners of the pattern oriented approximately the same as it appears to the camera in the Stewart platform experiments.

When an image is acquired, the top row of pixels is aquired first, then the second and so on. In a dynamic heave experiment with the pattern moving away from the camera the second row of corners which are acquired after the first row will be acquired when the pattern is further away from the camera than where the first row was acquired. This applies to all corners, and leads to the corners being acquired as if they were located on a curved plane. The effect from this is that the pattern appears at an angle while it is in fact at 0° during the dynamic heave experiments.

It has been observed that the variance changes with the lighting conditions, the reason for this has not been investigated thorougly, but it is thought to be related to the dilation of thresholded Harris response and the subpixel accuracy of the blob detection. How focused an image is rely on many factors, among them is how much the pattern is illuminated, the aperture opening and the exposure time of the pixels. Exposure time and aperture opening can easily be controlled, illumination of pattern can be more challenging. If illumination of the pattern changes, the aperture opening might need adjustment to change the amount of light reaching the imaging sensor. This will in turn lead to the exposure time and focus requiring readjustment as well to keep the pattern focused.

For a focused corner there will be a very clear point where the threshold k is applied when dividing pixels into class 0 and class 1 to separate corner responses from background. As the image becomes more unfocused the corner responses are not as bright, and some pixels might have the same intensity as the threshold level, thus flickering between the classes.

If a pixel flickers between being a part of a corner indication or not during the thresholding operation, the centerpoint of that blob will move accordingly. The dilation applied to ensure all indications of a corner is detected as one corner might enhance this effect, because one pixel flickering between class 0 and class 1 will be several pixels after dilation, moving the centerpoint of the blob more than one pixel would.

Variance is also linked to the resolution of the region of interest, ROI, as it decreases with increased resolution of ROI. This might also be linked to the previously mentioned effect of pixels flickering between class 0 and class 1. With a lower resolution of the ROI, each corner response will consist of fewer pixels, and a pixel flickering between classes will have a larger impact on the position of the centerpoint.

CHAPTER 7

Discussion

The Basler ace camera used in this setup has a 2592x1944 pixel imaging sensor with 2.2 μm high and wide pixels, and a rolling shutter. It can acquire images at 14 frames per second, although 15 frames per second has been observed. Quality of the measurement data would be improved if the current camera was replaced with a camera with higher performance. The software that has been developed for the measurement system can be used on any camera system, as long as the intrinsic parameters for the system are known. This camera was chosen because it provided high resolution at a price of 4000 NOK.

Point Grey delivers several cameras that would be interesting to test in this vision-based measurement system. The Grasshopper USB3 camera [22] can be delivered with a 2048x2048 pixel imaging sensor with 5.5 μm high and wide pixels, global shutter and a framerate of 90 fps which pushes the 350 MB/s bandwidth boundary of the USB3 Vision standard [23]. With physically larger pixels, less light is required for each pixel, allowing for shorter exposure time and higher framerates. The global shutter will ensure that the scene is captured in the same instance, not allowing it to change during the image acquisition. This camera costs 12 000 NOK, 3 times more expensive than the Basler ace camera currently used.

If the framerate of the camera is increased, the demands to the software computing the measurements is also increased. The system is in its current state fairly well balanced with image acquisition frequency at 14 Hz and software update frequency at approximately 10-12 Hz.

As shown in the dynamic heave experiment, a delay between Stewart platform position and measurements from the camera system that might be linked to the cycletime exist. If the camera frame rate is increased to 90 Hz, the software will still be the bottleneck at its 10-12 Hz. All functions are currently running in the same loop, dividing the image processing and homography computation into two separate loops might reduce the cycletime of the software.

The most time consuming operations are the creation of the ROI and the Harris corner detector. When looking for the ROI, the whole image is searched. The information about where the ROI was in the previous cycle of the program could be used to reduce the area to look for the pattern in. The coordinates of the previous ROI and an added boundary that would allow the pattern to move reasonably between cycles could be used to decrease the area to search for the pattern in and thereby reducing the cycletime.

The Harris corner detector has already been optimized by computing the Harris responses, M_c, column-wise instead of pixel-wise. An interesting approach to further improve the speed of the Harris corner detector is to look into using a graphics processing unit, GPU, to perform these computations. Any improvements from this might be overshadowed by something as simple as using an adequately powerful multi-core CPU.

With a faster camera and software divided in several loops, the update frequency of the measurement system could be improved a lot. A natural division would be image processing operations executed in one loop, encompassing the image acquisition and necessary processing until an array of coordinates is available. This array is used in a second loop where all operations regarding axis lines, finding feature correspondences and computing homography is done. The pose of the pattern is then computed simultaneously as the next image is being processed.

Another interesting approach to finding correspondences is to create a grid that support the corners found, and use intersecting lines of this grid as corner coordinates for the homography. The corners in the pattern are on straight lines and the RANSAC algorithm can be used to find the line that is the best fit for a line passing through the corners found that should be on a line. This way a grid of lines that is the best fit for lines passing through the corners could be used to create correspondences between world coordinates and pixel coordinates of the intersections [21].

CHAPTER 8

Conclusion

A vision-based measurement system that will measure relative motion with an update frequency of approximately 10 Hz between a camera and a reference pattern has been developed. Accuracy during static experiments show that measurements are well within the target of achieving a heave accuracy of ±25 mm, with mean values deviating less than 0.5 mm from the position of the Stewart platform at a distance of approximately 2.3 m. The target for ±0.5° angular accuracy is also achieved, with mean values for roll angle deviating less than 0.15° and mean values for pitch angle deviating less than 0.2°.

Larger deviations are seen during the dynamic experiments with deviations increasing with the velocity of the pattern. The error source for these deviations is assumed to be way the images of the scene are formed, with an imaging sensor using a rolling shutter that captures the scene one row of pixels at the time, not capturing the whole scene at the same time as an imaging sensor with a global shutter would.

The measurement system has been realized utilizing off the shelf components, and utilizing the USB3 Vision standard. Camera and lens costs 6000 NOK and the software is executed on a computer with an Intel i5-660 CPU which has two cores at 3.33 GHz. The software that has been developed can be executed on any fairly up to date computer with a USB3 interface, and any camera system can be used as long as the intrinsic parameters of the camera system are known.

Many vision-based measurements exists for a vast range of applications, but few systems exist for this application. As this thesis show, some further work is required, but the results achieved at this stage are very promising. Further work stemming from this thesis could be looking into optimization of the code to reduce the demands to the system it is executed on, improve the correction for distortion, use other techniques for feature correspondence, improve the error handling and include velocity measurements.

Bibliography

[1] Seatools *Offshore Access System*, 31.05.2014
http://www.seatools.com/portfolio/heave-and-motion-compensation-systems/oas-heave-compensation

[2] D. E. Hasselman et al., Directional Wave Spectra Observed during JONSWAP 1973, *Journal of Physical Oceanography*, vol.10, pp. 1264-128, 1980

[3] Byoung-Ho Song et al., "Case based reasoning system based risk context awareness in the digital marine vessels", the Information and Communication Technology Convergence 2010 International Conference, 2010, Jeju, 443-447

[4] Basler, *Product information for acA2500-14um camera*, 29.04.2014
http://www.baslerweb.com/products/ace.html?model=387

[5] Microstrain, *Product information for 3DM-GX3® -25-OEM IMU*, 30.04.2014
http://www.microstrain.com/inertial/3DM-GX3-25-OEM

[6] G. Bruneau et al. (2012) "MATLAB Implementation of MonoSLAM", 26.04.2014
http://perso.ensta-paristech.fr/~filliat/Courses/2011_projets_C10-2/BRUNEAU_DUBRAY_MURGUET/monoSLAM_bruneau_dubray_murguet_en.html

[7] Jean-Yves Bouguet, "Camera Calibration Toolbox for Matlab", 02.02.2013
http://www.vision.caltech.edu/bouguetj/calib_doc/

[8] Heikkilä J. & Silvén, O. "A Four-step Camera Calibration Procedure with Implicit Image Correction" IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, San Juan, 1106-1112.

[9] F. Zhao et al, "An Automated X-corner Detection Algorithm (AXDA)", *Journal of Software* Vol 6 no 5, pp.791-797, 2011.

[10] N Otsu, "A threshold selection method from gray-level histograms" *IEEE Transactions on systems, man, and cybernetics*, vol 9 no 1, pp 62-66, 1979

[11] National Instruments *Vision Concepts Manual*, 26.02.2014
http://www.ni.com/pdf/manuals/372916e.pdf

[12] National Instruments White Papers *Image Analysis and Processing*, 25.04.2014
http://www.ni.com/white-paper/3470/en/#toc3

[13] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, Vol 24 Issue 6, pp 381-395, 1981.

[14] C. Harris and M. Stephens ”A Combined Corner And Edge Detector”, 1988, Alvey vision conference. Vol. 15. 147-151
http://courses.daiict.ac.in/pluginfile.php/13002/mod_resource/content/0/References/harris1988.pdf

[15] R. Hartley and A. Zisserman *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004

[16] Michael E. Pique, *Graphics Gems*, Academic Press, 1990

[17] P. Corke *Robotics, Vision and Control*, Springer, 2011

[18] D. Stewart, "A Platform with Six Degrees of Freedom", *Proceedings of the Institution of Mechanical Engineers.* Vol 180 pt 1 no 15, pp 371-386, 1965

[19] V. E. Gough, "Contribution to discussion of papers on research in Automobile Stability, Control and Tyre performance". *Proc. Auto Div. Inst. Mech. Eng.*,pp 392-394, 1957

[20] Rexroth, *EMotion-1500 Specifications* 02.06.2014
http://www.boschrexroth.de/business_units/bri/subwebsites/systems_engineering/en/Industry_Segments/Motion_Simulation_Technology/applications_motion_new/e_motion1500/emotion1500_specs/index.jsp

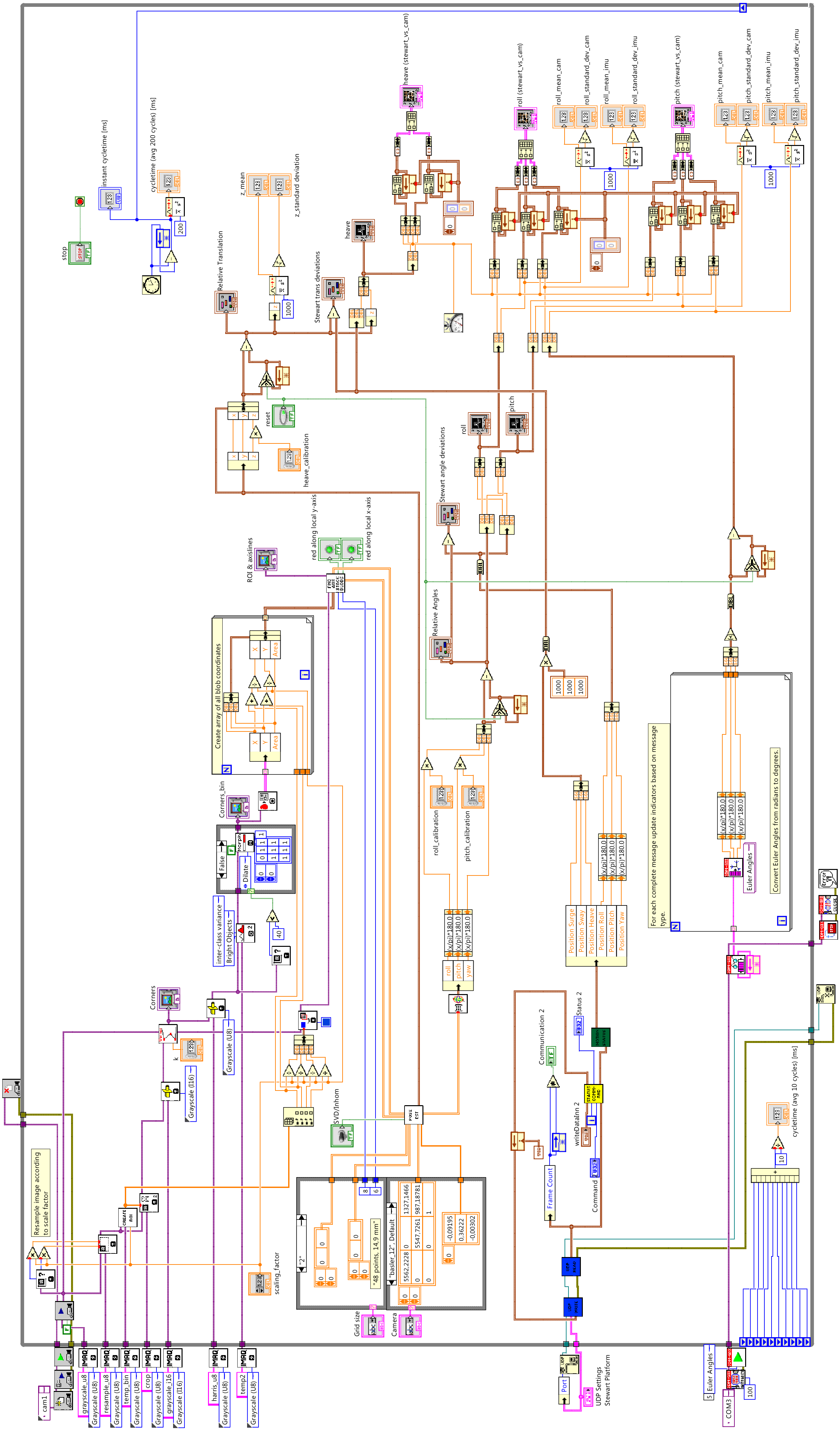
[21] M. Ottestad, University of Agder 2013

[22] Point Grey, *Product information for Grasshopper3 camera*, 29.05.2014
http://grasshopper3.ptgrey.com/USB3/Grasshopper3

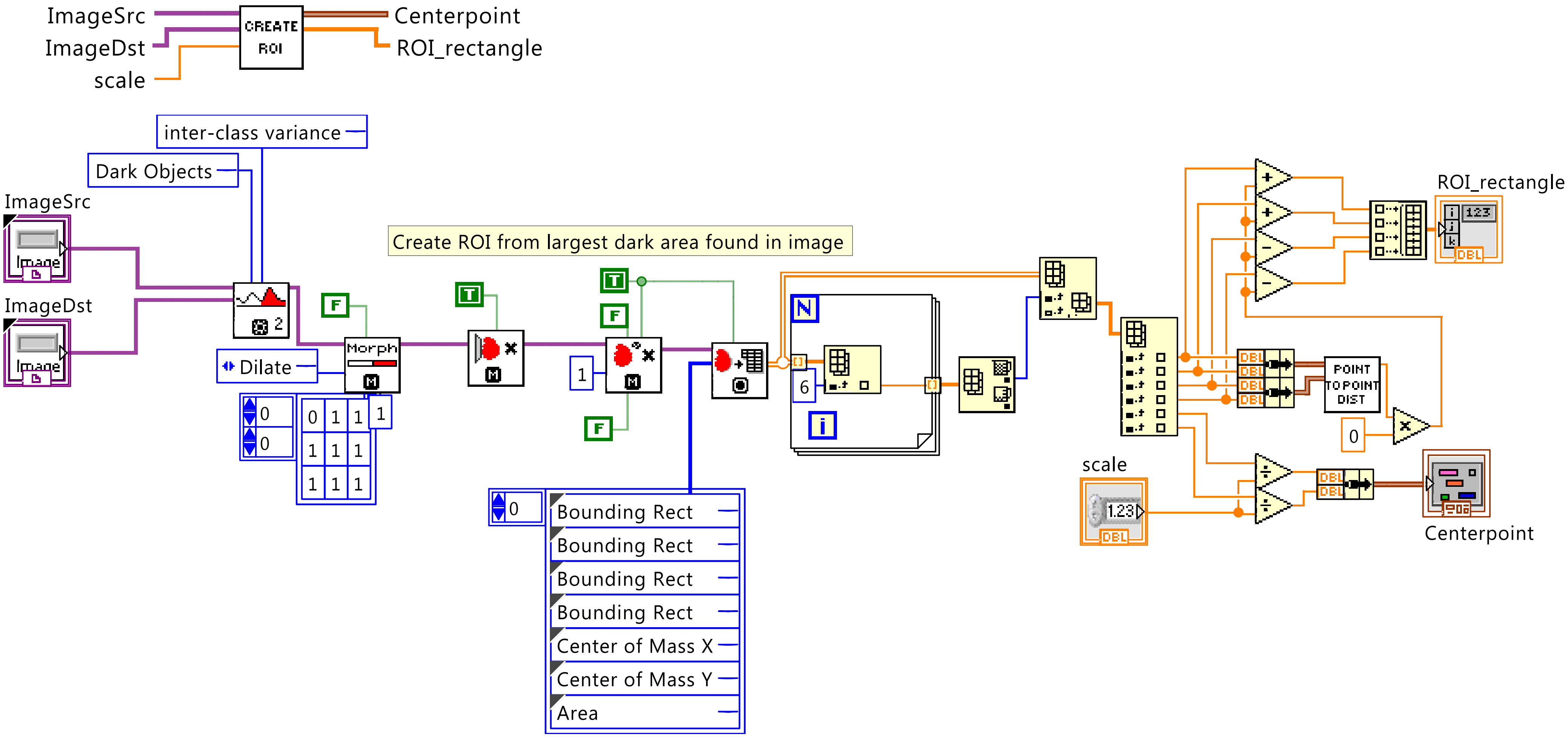
[23] *USB3 Vision Standard* AIA 2013
http://www.visiononline.org/vision-standards-details.cfm?type=11, 29.05.2014

APPENDIX A

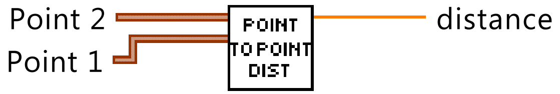
Labview Software

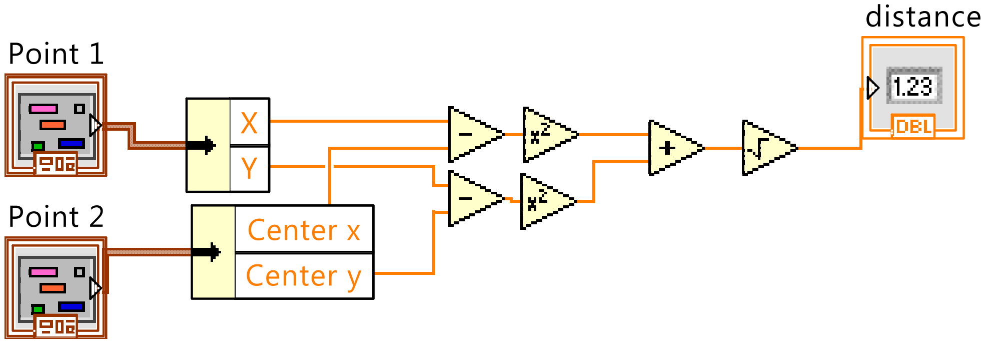

stop
instant cycletime [ms]
cycletime (avg 200 cycles) [ms]
200
z_mean
z_standard deviation
1000
Relative Translation
Stewart trans deviations
heave
heave (stewart_vs_cam)
roll (stewart_vs_cam)
roll_mean_cam
roll_standard_dev_cam
roll_mean_imu
roll_standard_dev_imu
pitch (stewart_vs_cam)
pitch_mean_cam
pitch_standard_dev_cam
pitch_mean_imu
pitch_standard_dev_imu
reset
heave_calibration
roll
pitch
Stewart angle deviations
Relative Angles
ROI & axislines
red along local y-axis
red along local x-axis
Create array of all blob coordinates
X
Y
Area
Corners_bin
Morph
Dilate
False
inter-class variance
Bright Objects
40
Corners
k
Grayscale (U8)
Grayscale (I16)
SVD/Inhom
roll_calibration
pitch_calibration
roll
pitch
yaw
(x/pi)*180.0
Position Surge
Position Sway
Position Heave
Position Roll
Position Pitch
Position Yaw
For each complete message update indicators based on message type.
Convert Euler Angles from radians to degrees.
Euler Angles
Communication 2
Status 2
writeDataIn 2
Frame Count
Command
cycletime (avg 10 cycles) [ms]
10
Resample image according to scale factor
scaling_factor
Grid size
Camera
"2"
"48 points, 14,9 mm"
"basler_12", Default
8
6
5562,2228
1327,1466
5547,7261
987,18781
-0,09195
0,36222
-0,00302
cam1
grayscale_u8
resample_u8
temp_bin
crop
grayscale_i16
harris_u8
temp2
Grayscale (U8)
Grayscale (I16)
Port
UDP Settings Stewart Platform
COM3
5
100

create_ROI.vi



distance_point_to_point.vi





mocorner.vi

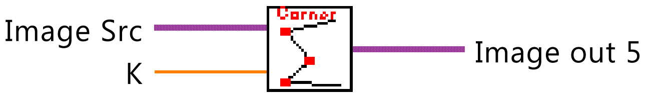
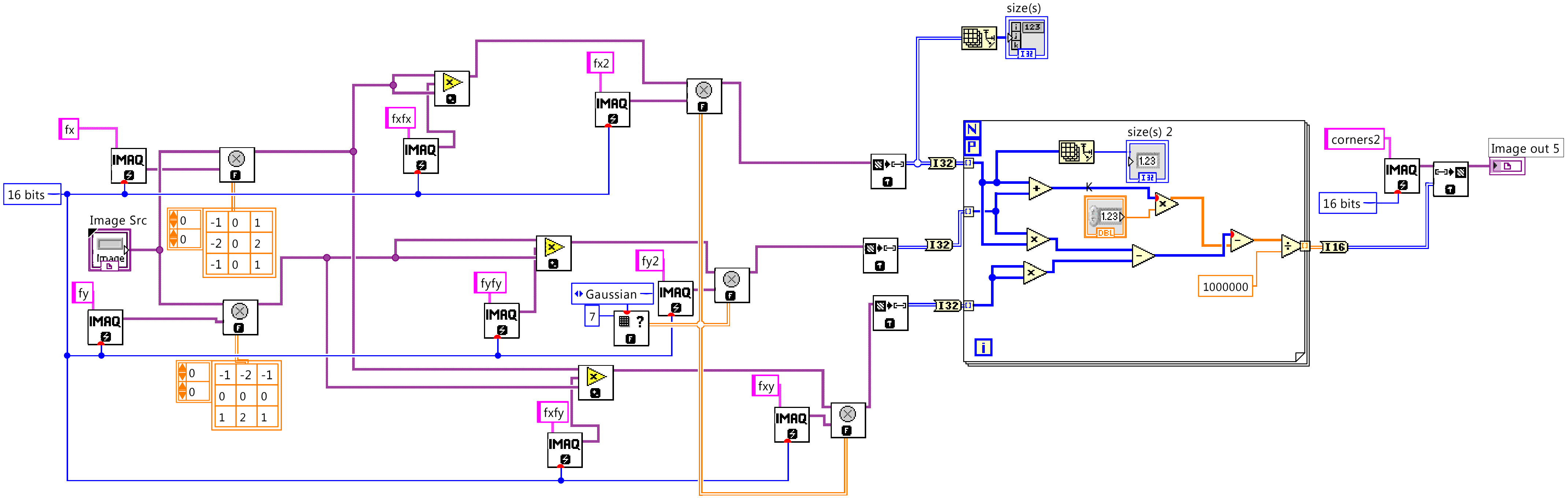
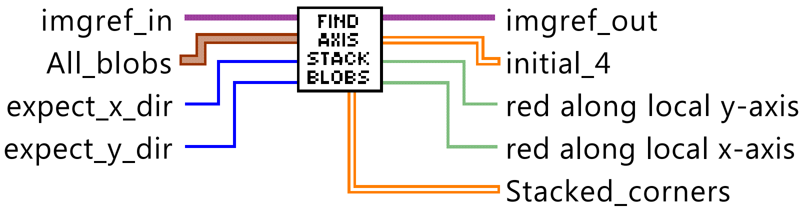


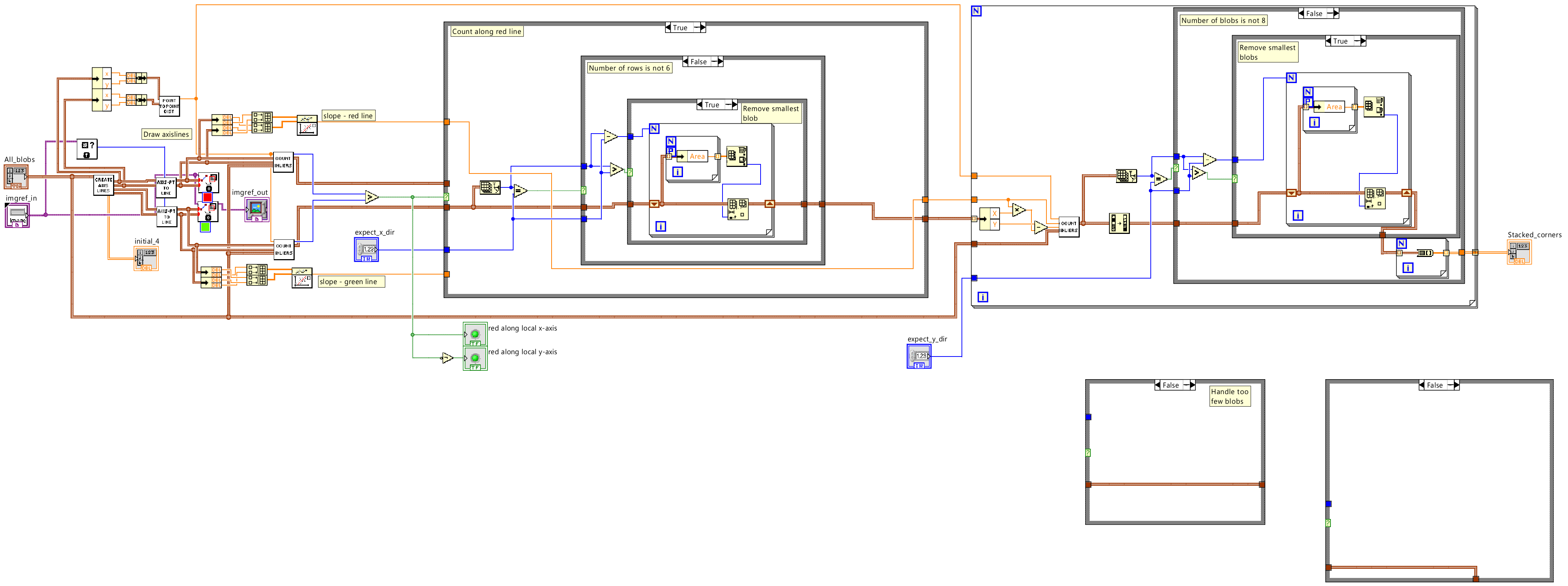
Image Src
K
Corner
Image out 5



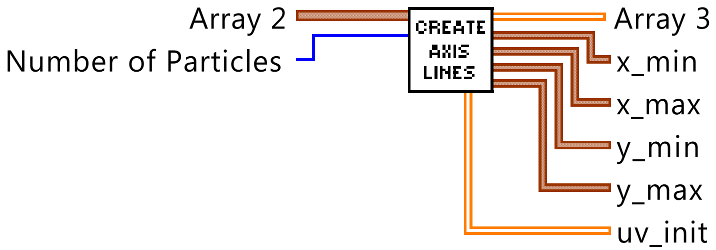
fx
16 bits
IMAQ
Image Src
Image
0
0
-1 0 1
-2 0 2
-1 0 1
fy
-1 -2 -1
0 0 0
1 2 1
fxfx
fx2
fyfy
fy2
Gaussian
7
fxfy
fxy
size(s)
size(s) 2
I32
I32
I32
N
P
K
1.23
DBL
1000000
I16
i
corners2
16 bits
Image out 5

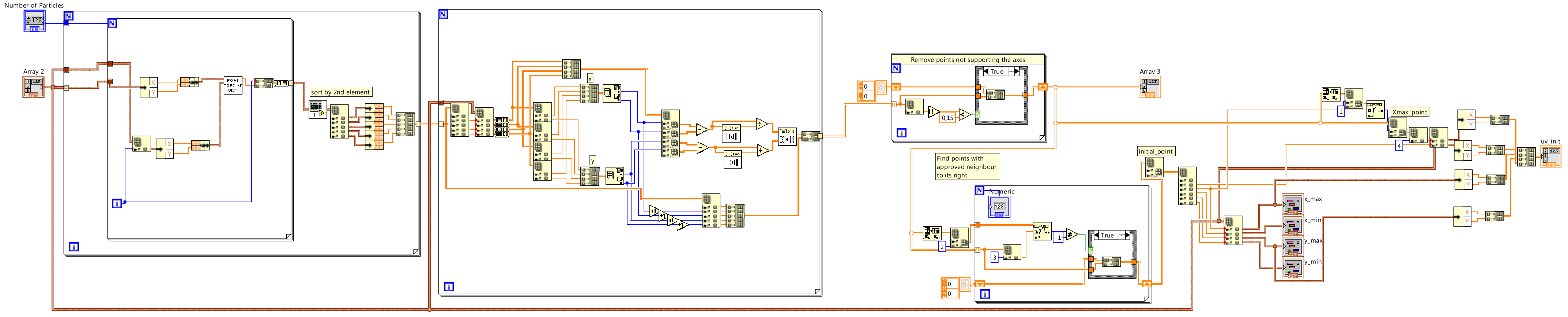
find_axis_and_count_blobs.vi



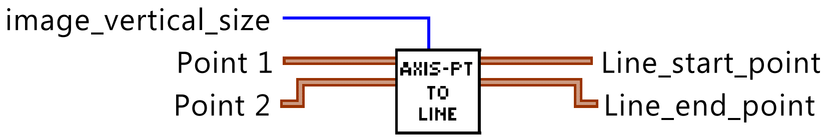


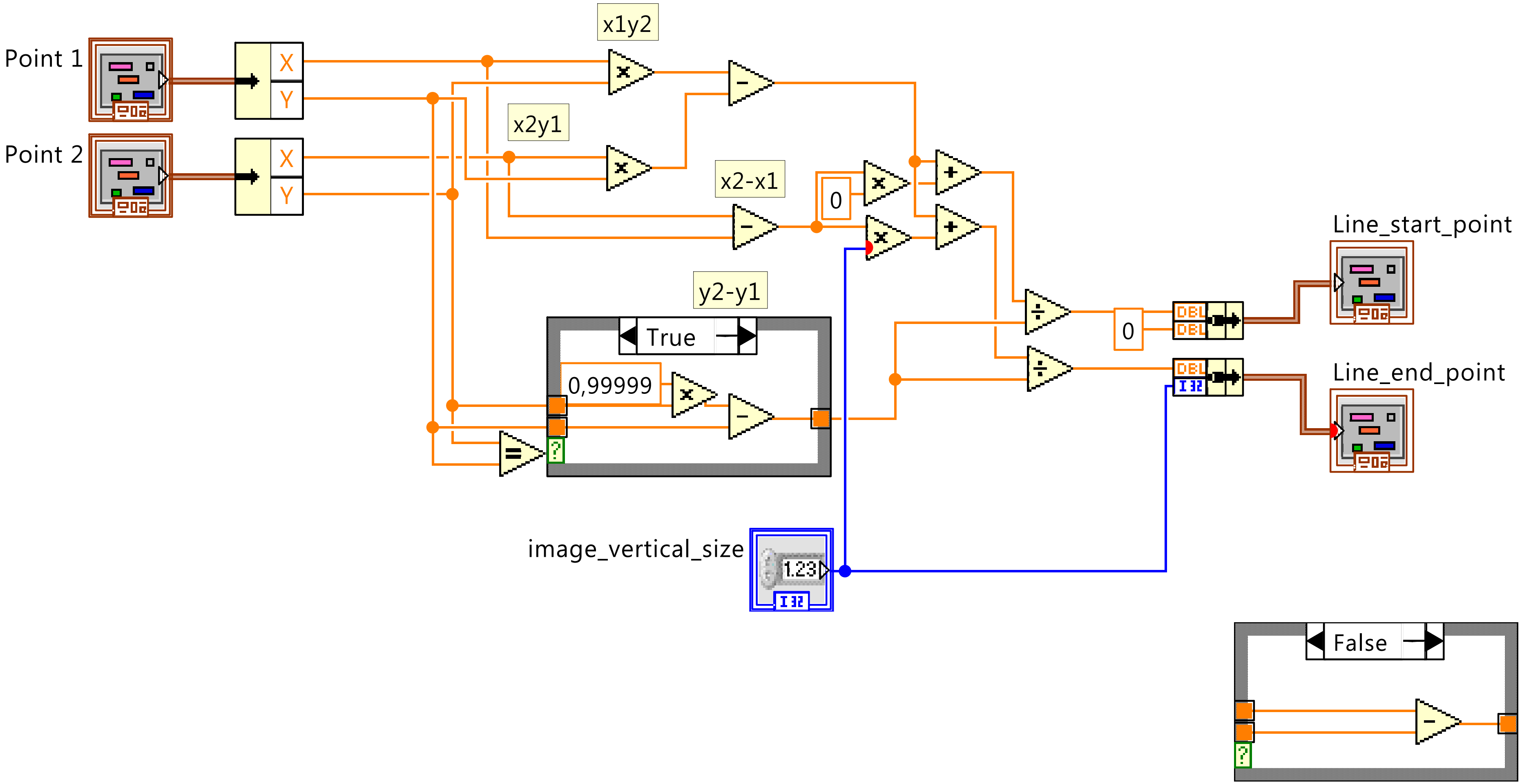
find_axis_lines_rev4.vi



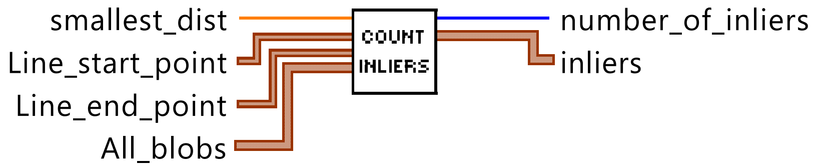


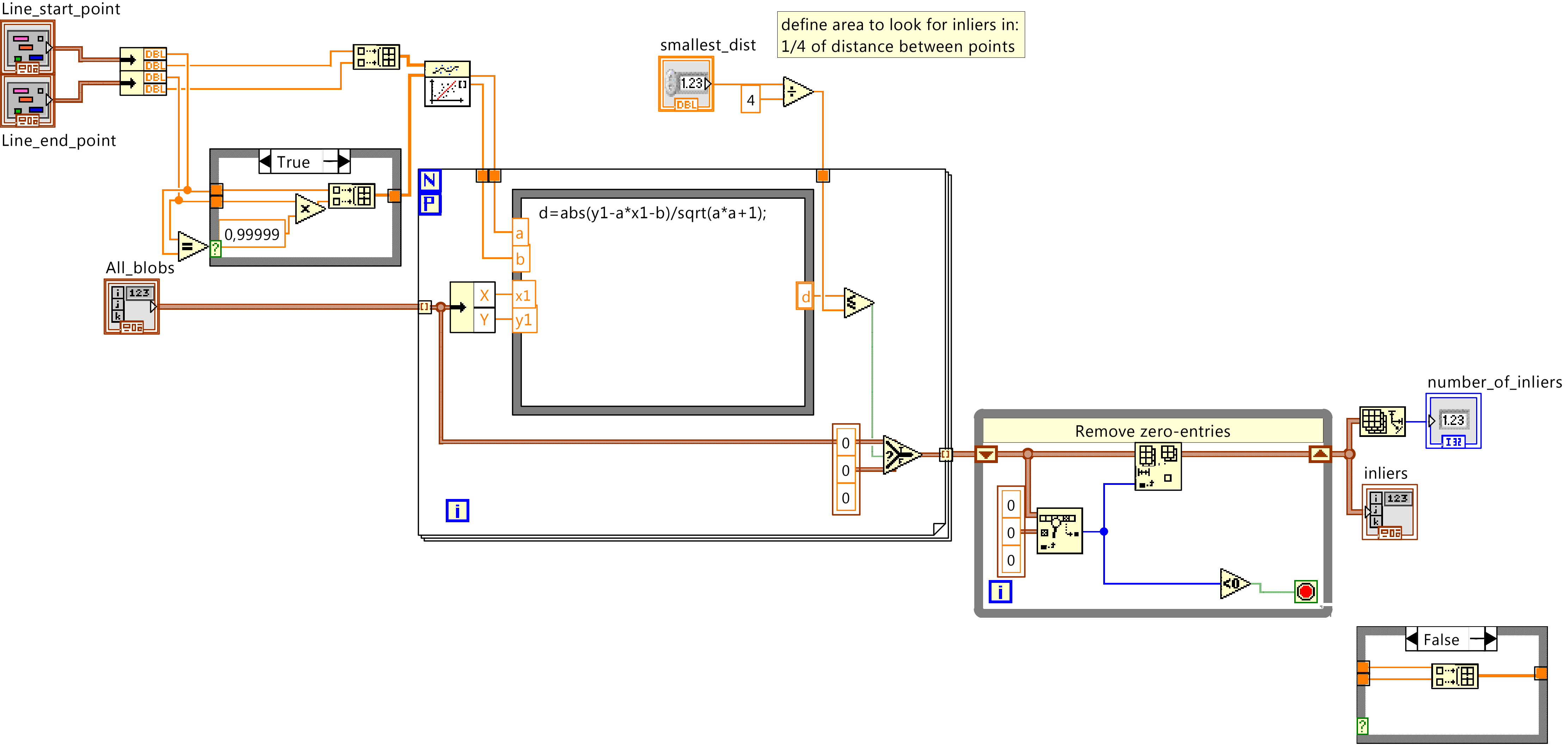
convert_axis_points_to_line_points.vi





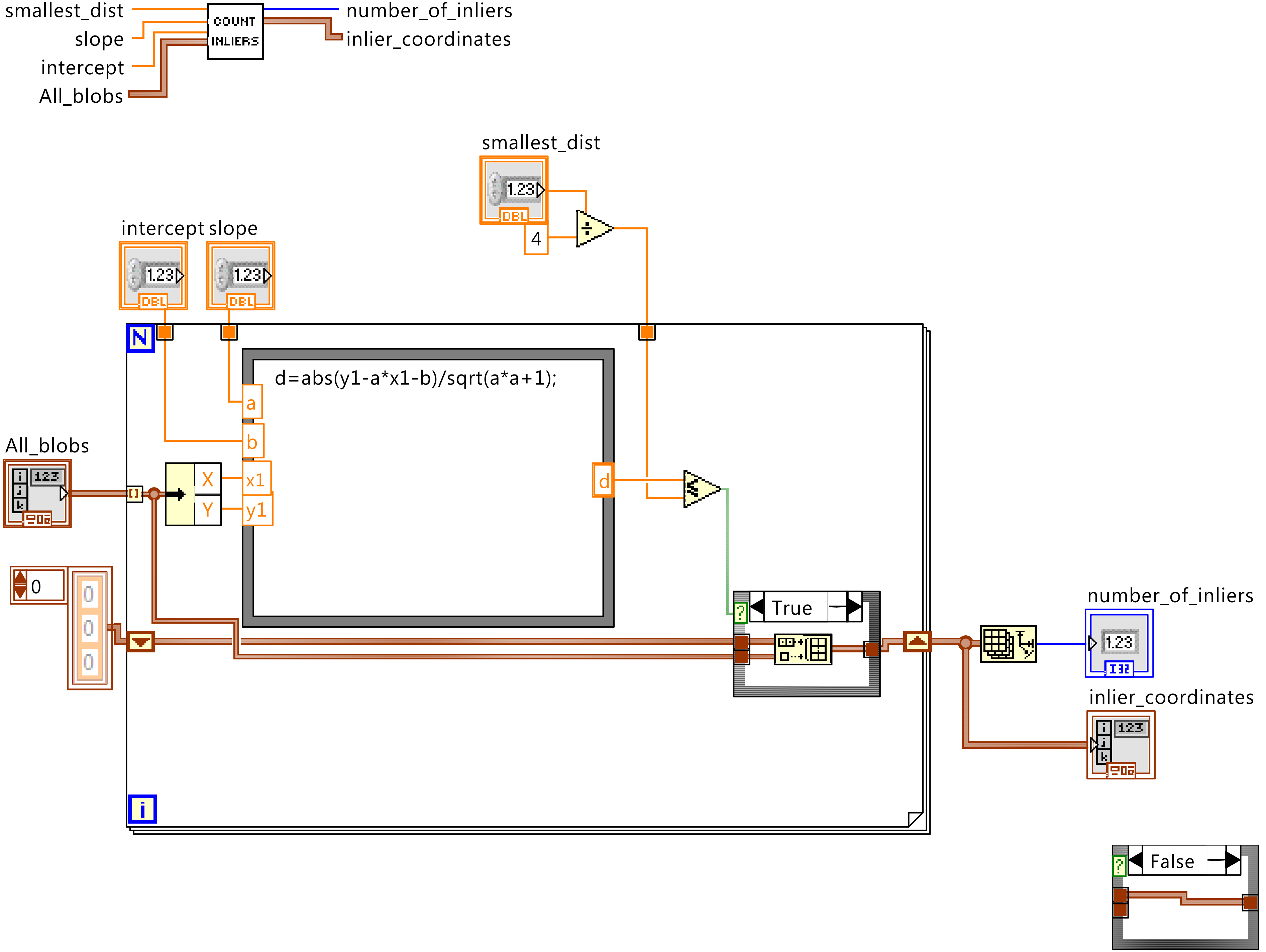
count_inliers_point_to_point_rev2.vi





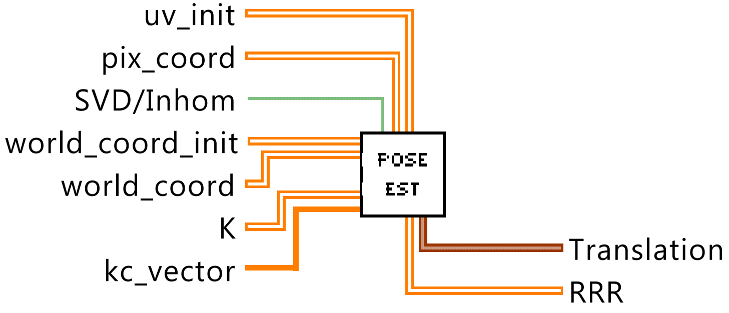
```
d=abs(y1-a*x1-b)/sqrt(a*a+1);
```

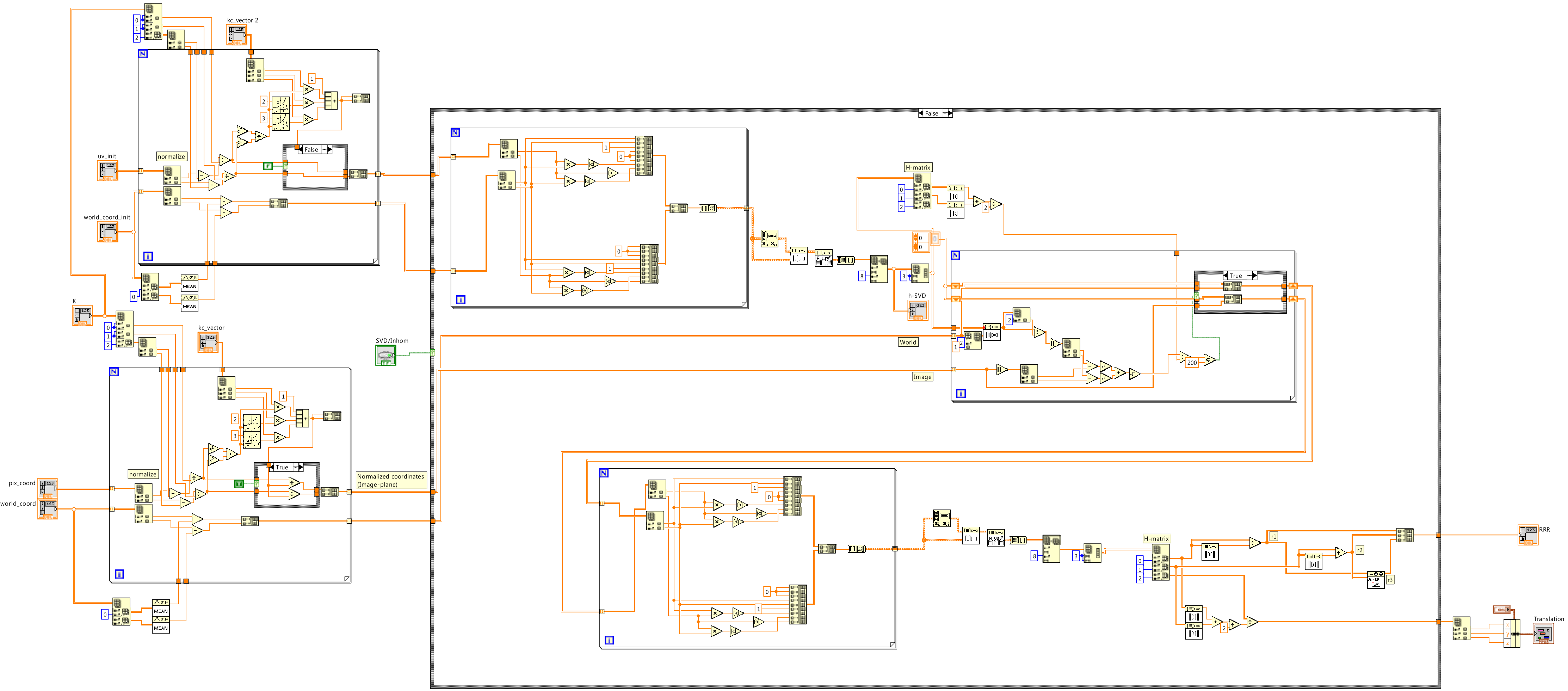
count_inliers_slope_intercept.vi



```
d=abs(y1-a*x1-b)/sqrt(a*a+1);
```

pose_estimation_w_svd_rev8.vi





APPENDIX B

Equipment Datasheets

TECHNICAL DETAILS

Specifications

Basler ace	NEW acA1920-25um/uc	acA2500-14um/uc	NEW acA3800-14um/uc*	NEW acA4600-10uc*
Camera				
Resolution (H x V pixels)	1920 x 1080	2592 x 1944	3856 x 2764	4608 x 3288
Sensor	Aptina MT9P031	Aptina MT9P	Aptina MT9J003	Aptina MT9F002
Sensor Size (optical)	1/3.7“	1/2.5“	1/2.3″	1/2.3″
Sensor Technology	CMOS, rolling shutter			
Pixel Size (µm)	2.2 x 2.2	2.2 x 2.2	1.67 x 1.67	1.4 x 1.4
Frame Rate	25	14	14	10
Mono / Color	Mono/Color			Color
Video Output Format	Mono 8, Mono 12, Mono 12 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed, Bayer BG 8, Bayer BG 12, Bayer BG 12 Packed			
Interface	USB 3.0			
Synchronization	Via external trigger or free-run			
Exposure Control	Via external trigger or programmable via the camera API			
Mechanical / Electrical				
Housing Size (L x W x H)	29.3 mm x 29 mm x 29 mm			
Housing Temperature	Up to 50 °C			
Lens Mount	C, CS	C, CS	C, CS	C, CS
Digital I/O	1 opto-isolated input + 1 opto-isolated output + 2 Fast-GPIO (configurable as In/Out)			
Power Requirements	Via USB 3.0 interface			
Power Suspend Mode	Yes, less than 0.02 W, configurable			
Power Consumption (typical)	2.2 W	2.2 W	2.2 W (preliminary)	2.2 W (preliminary)
Weight (typical)	<80 g			
Conformity	CE, FCC, IP30, RoHS, UL (in preparation), USB3 Vision, USB-IF (in preparation)			
Software / Driver				
Driver	Basler pylon Camera Software Suite or 3rd party USB3 Vision Software			
Operating System	Windows 32 bit and 64 bit			
Conformity	USB3 Vision, GenICam			

Specifications are subject to change without prior notice. Latest specifications can be found on our website. Please visit www.baslerweb.com/manuals for the detailed camera User's Manual and www.baslerweb.com/thirdparty for information on third party software.

* Available Q2/2014

computar

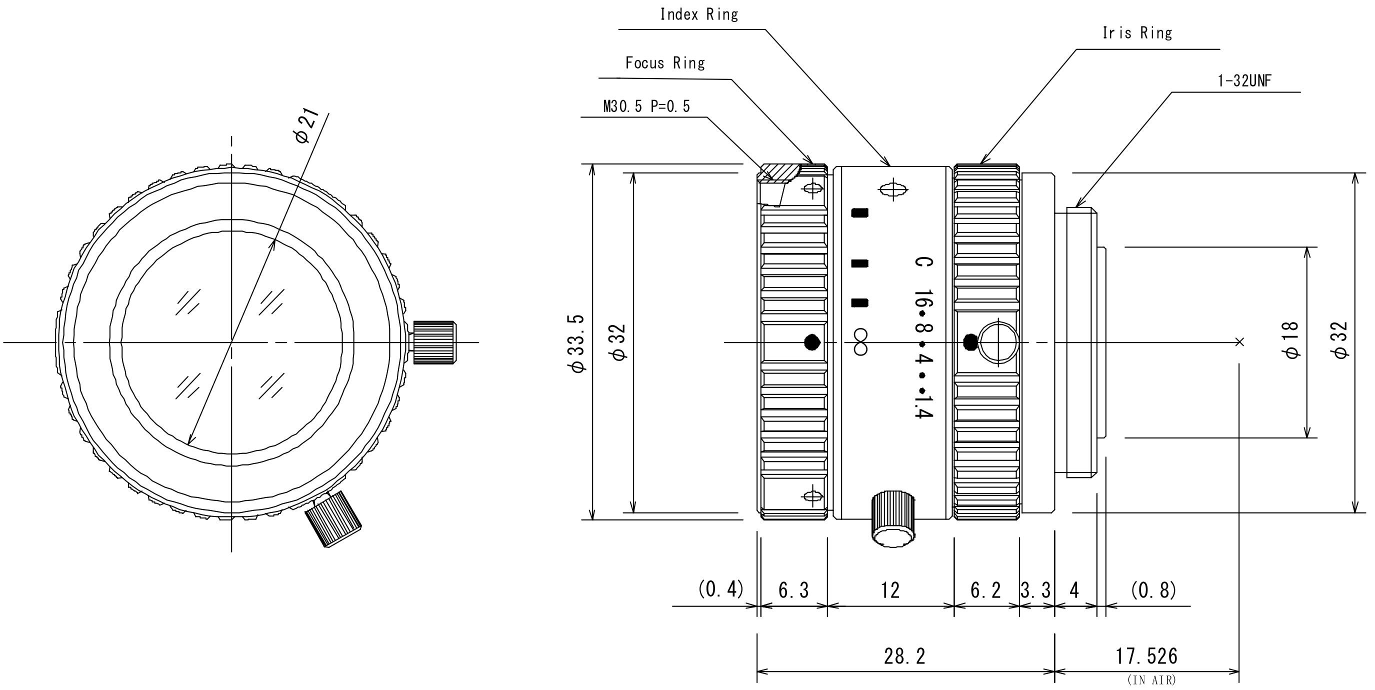
M1214-MP2

f=12mm F1.4
for 2/3 type Megapixel Cameras
C-Mount

Model No.		M1214-MP2		Effective	Front	φ21.0mm	
Focal Length		12mm		Lens Aperture	Rear	φ13.0mm	
Max. Aperture Ratio		1:1.4		Distortion	2/3type	-0.1%(y=5.5)	
Max. Image Format		8.8mm x 6.6mm(φ11mm)			1/2type	-0.35%(y=4.0)	
Operation Range	Iris	F1.4 - F16C		Back Focal Length		13.1mm	
	Focus	0.15m - Inf.		Flange Back Length		17.526mm	
Control	Iris	Manual		Mount		C-Mount	
	Focus	Manual		Filter Size		M30.5 P=0.5mm	
Object Dimension	2/3 type	12.1cm x 9.0cm		Dimensions		φ33.5mm x 28.2mm	
at M.O.D	1/2 type	8.8cm x 6.5cm		Weight		61.9g	
Angle of View	D	2/3 type	49.2°	1/2 type	37.0°		
	H		40.4°		30.0°		
	V		30.8°		22.7°		
Operating Temperature		-10℃ - +50℃					

M.O.D : Minimum Object Distance

Dimensions



Specification is subject to change without any notice.

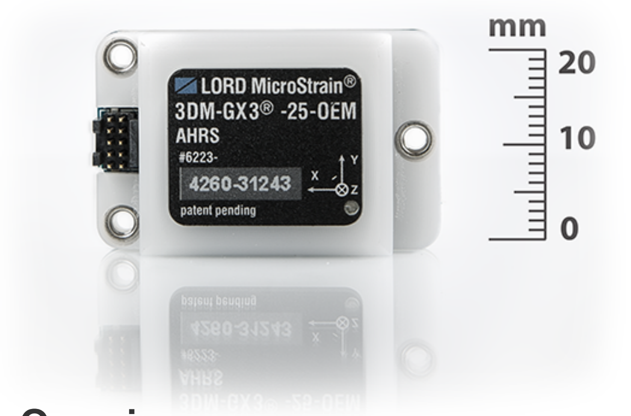
2008.05

LORD PRODUCT DATASHEET

3DM-GX3® -25-OEM

Miniature Attitude Heading Reference System

The **3DM-GX3®** -25-OEM is a high-performance, miniature Attitude Heading Reference System (AHRS), utilizing MEMS sensor technology. It combines a triaxial accelerometer, triaxial gyro, triaxial magnetometer, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm to provide static and dynamic orientation, and inertial measurements. Its form factor is ideally suited for OEM applications.



Features & Benefits

Easiest to Integrate

- smallest, lightest industrial OEM AHRS available
- simple integration supported by SDK and comprehensive API

Best in Class

- precise attitude estimations
- high-speed sample rate & flexible data outputs
- high performance under vibration

Cost Effective

- reduced cost and rapid time to market for customer's applications
- aggressive volume discount schedule

Applications

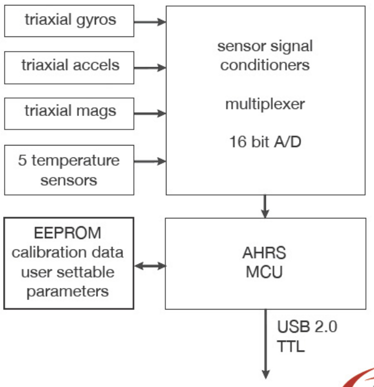
Accurate navigation and orientation under dynamic conditions such as:

- Inertial Aiding of GPS
- Unmanned Vehicle Navigation
- Platform Stabilization, Artificial Horizon
- Antenna and Camera Pointing
- Health and Usage Monitoring of Vehicles
- Reconnaissance, Surveillance, and Target Acquisition
- Robotic Control
- Personnel Tracking

System Overview

The **3DM-GX3®** -25 OEM offers a range of fully calibrated inertial measurements including acceleration, angular rate, magnetic field, deltaTheta and deltaVelocity vectors. It can also output computed orientation estimates including Euler angles (pitch, roll, and heading (yaw)), rotation matrix and quaternion. All quantities are fully temperature compensated and are mathematically aligned to an orthogonal coordinate system. The angular rate quantities are further corrected for g-sensitivity and scale factor non-linearity to third order. The **3DM-GX3®** -25 OEM architecture has been carefully designed to substantially eliminate common sources of error such as hysteresis induced by temperature changes and sensitivity to supply voltage variations. Gyro drift is eliminated in AHRS mode by referencing magnetic North and Earth's gravity and compensating for gyro bias. On-board coning and sculling compensation allows for use of lower data output rates while maintaining performance of a fast internal sampling rate.

The **3DM-GX3®** -25 OEM is initially sold as a starter kit consisting of an AHRS module, USB communication and power cable, software CD, user manual and quick start guide. The circuit board form-factor provides thru-holes for mounting on larger circuit assemblies and custom TTL communication and power cables can be user fabricated or purchased from the factory.



best of sensors expo 2009
Silver Level Winner

LORD MicroStrain®
SENSING SYSTEMS

E-Motion-1500

Specifications

General	
Payload:	1500 Kg (2204lbs)
Ixx	2023Kgm²
Iyy	3713Kgm²
Izz	3611Kgm²
Settled height	1.392 m.

Relative to COG of the Payload

System limits (Operational)

Performance	pos.	neg
Surge	0.716m	-0.602m
Sway	0.603m	-0.603m
Heave	0.407m	-0.422m
Roll	27.45deg	-27.45deg
Pitch	27.10deg	-24.35deg
Yaw	39.20deg	-39.20deg

Velocity

Performance	
Surge	+/-0.8m/s
Sway	+/-0.8m/s
Heave	+/-0.6m/s
Roll	+/-40.0deg/s
Pitch	+/-40.0deg/s
Yaw	+/-50.0deg/s

Accelerations

Performance	non sim	onset
Surge	+/-0.65g	+/-8g/s
Sway	+/-0.60g	+/-8g/s
Heave	+/-0.60g	+/-8g/s
Roll	+/-300deg/s²	+/-3000deg/s²/s
Pitch	+/-300deg/s²	+/-3000deg/s²/s
Yaw	+/-350deg/s²	+/-3000deg/s²/s

EMotion-8000

Specifications

General	
Payload:	8000 Kg (17600lbs)
Ixx	21274Kgm^{2}*
Iyy	24193Kgm^{2}*
Izz	28197Kgm^{2}*
Settled height	1.370 m.

*Relative to COG of the Payload

System limits (Operational)

Performance*	pos.	neg.
Surge	1.333m	-1.110m
Sway	1.149m	-1.149m
Heave	0.885m	-0.955m
Roll	26.10deg	-26.10deg
Pitch	33.40deg	-25.55deg
Yaw	31.10deg	-31.10deg

Velocity

Performance	
Surge	+/-0.711m/s
Sway	+/-0.711m/s
Heave	+/-0.610m/s
Roll	+/-20.0deg/s
Pitch	+/-20.0deg/s
Yaw	+/-20.0deg/s

Accelerations

Performance	non sim	onset
Surge	+/-0.6g	+/-3g/s
Sway	+/-0.6g	+/-3g/s
Heave	+/-0.8g	+/-6g/s
Roll	+/-100deg/s^{2}	+/-300deg/s^{2}/s
Pitch	+/-100deg/s^{2}	+/-300deg/s^{2}/s
Yaw	+/-100deg/s^{2}	+/-300deg/s^{2}/s