5
Results

5.1.
Introduction

The objective of this chapter is to implement the presented theory in practical experiments. The theory was first simulated in Matlab in order to eliminate errors in the mathematic models and to make it easier to identify potential anomalies in the experiments. These experiments were performed in the robotic laboratory at PUC-Rio.

5.2.
Laboratory Experiments

These experiments were carried out using an x-y table with a movement range of 200mm in x and y directions and a resolution of 0.1mm. A rotary stage is mounted on top of the x-y table. The rotation axis is perpendicular to the x-y table and the angular resolution is 0.1°. A camera was mounted on the rotary stage. The table with the mounted web camera is shown in Figure 40.

The camera calibration was performed using a 2D calibration rig with two perpendicular planes with 7 x 7 squares in chess pattern. The chess boards had dimensions 197mm x 197mm. The calibration rig is shown in Figure 41.

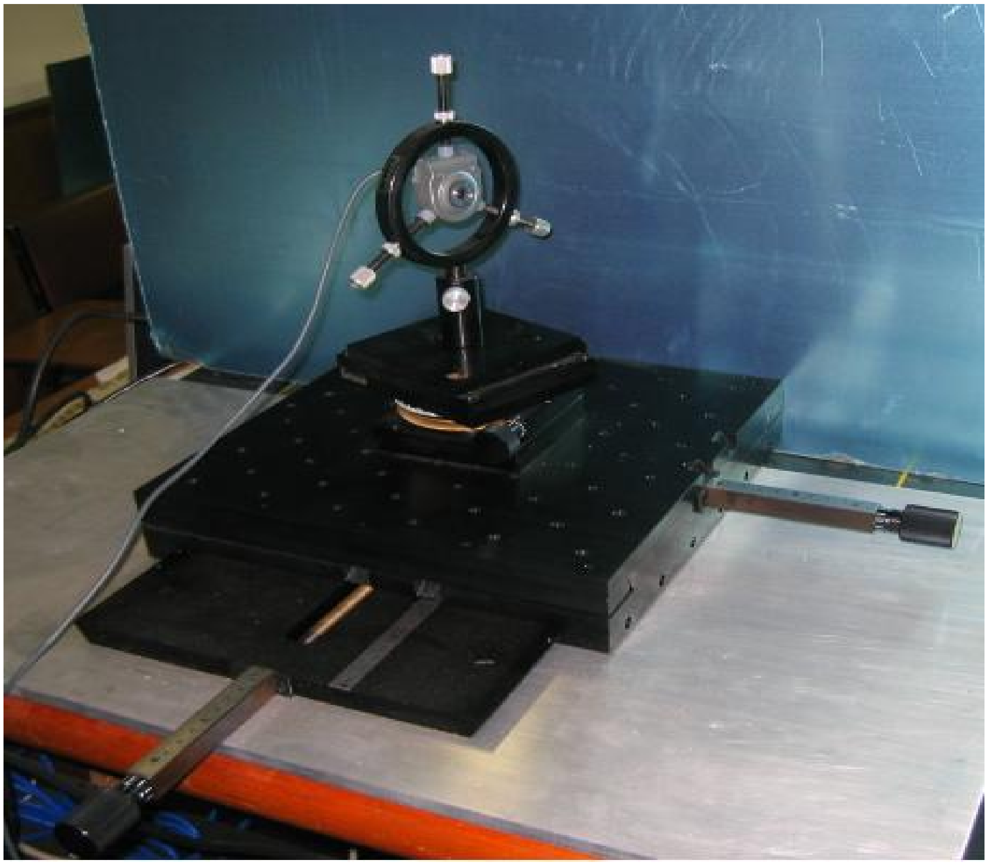

Figure 40 – x-y table in the Robotics laboratory at PUC

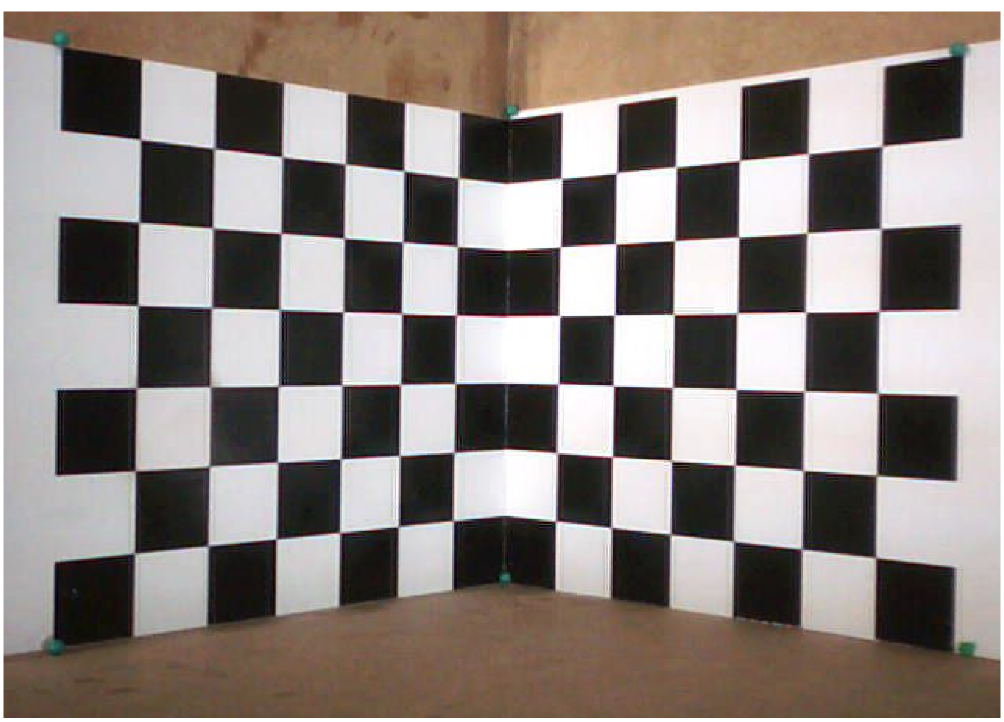

Figure 41 – Calibration rig

5.2.1.
Camera Calibration

The first experiments were carried out using a Logitec webcam with resolution 640 x 480 pixels. The images were given in Jpeg format. Before any experiments could be performed, the camera needed to be calibrated. Seven photos of the calibration rig were taken. The coordinates of the corners were found by using non maximum suppression and k-means line fitting, described in Chapter 3.3.4 and 3.3.5. The output of the non-maximum suppression algorithm and the estimated coordinates are shown in Figure 42.

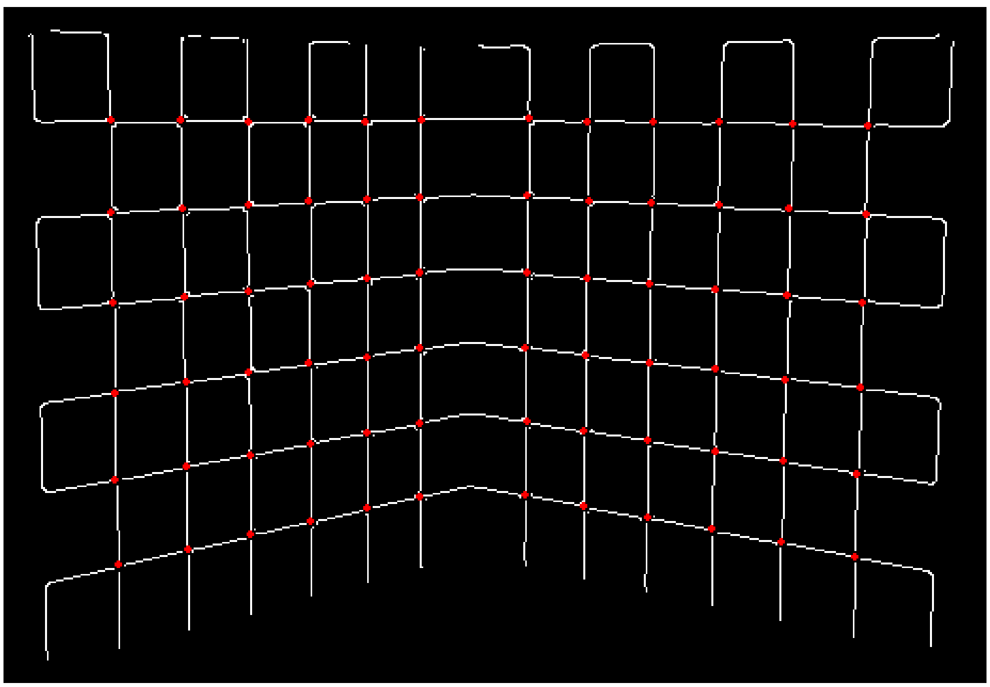

Figure 42 – Edges of the calibration rig. The estimated corners are marked in red.

The actual coordinate of each corner was found by estimating the intersection between each pair of estimated lines. Figure 43 shows how the coordinates were found through only a few iterations.

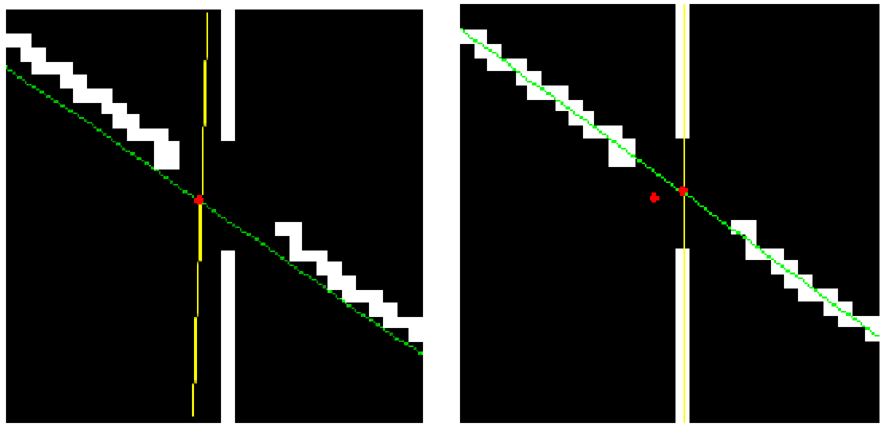

Figure 43 – Figure showing the edges of the image in white. The image to the left shows the initial k-mean line parameters. The image on the right shows the improved estimate after only three iterations. The estimated coordinate of the corner is marked in red. The yellow and green lines show the estimated lines.

The calibration procedure used was based on the pin-hole model and it did not take into account any contribution from lens distortion. To verify whether the pin-hole model was sufficient to model the camera, a test was performed comparing the extracted image coordinates with the projected coordinates using the estimated projection matrix. Ideally, the extracted image coordinates corresponding to the respective 3D coordinate on the chess board should correspond to the projected 3D coordinate using the estimated projection matrix from each calibration. This means that if there is a good correspondence between the projected image coordinates, $\left(u_p, v_p\right)$, and the extracted coordinates, (u_e, v_e), from the image, the linear pin-hole model is a sufficient model for the camera. The projected coordinates are given by the following formula:

$$\begin{bmatrix} u_{p,i} \\ v_{p,i} \\ 1 \end{bmatrix} = \frac{1}{z} M \, {}^{W}P_i \tag{195}$$

where M is the estimated projection matrix and ${}^{W}P_i$ is the i-th world coordinate from the calibration rig. A measure of the accuracy of the calibration is the RMS error between the projected coordinates and their corresponding extracted coordinates, (u_e, v_e). The average error will then be given by the following equation:

$$\chi = \sqrt{\frac{1}{N_w}\sum_{i=1}^{N_w}\left(\left(u_{p,i}-u_{e,i}\right)^2+\left(v_{p,i}-v_{e,i}\right)^2\right)} \qquad (196)$$

where N_w is the number of extracted coordinates from the calibration rig. In these experiments, 72 coordinates were extracted from the rig. The results from the calibration of the Logitec webcam are given in Table 3.

Param/pict	1	2	3	4	5	6	7	Average
θ	90.3	90.1	90.0	90.0	89.8	90.1	90.0	90.0
α	977.7	969.4	984.0	973.6	972.3	974.3	975.7	975.3
β	974.5	969.7	987.2	971.2	969.4	978.7	976.4	975.3
u_0	327.3	325.9	330.0	328.7	331.6	330.0	331.4	329.3
v_0	232.9	233.0	237.6	238.0	241.0	235.0	240.7	236.9
χ	0.463	0.397	0.457	0.466	0.449	0.417	0.475	0.443

Table 3 – Camera calibration parameters

The results in Table 3 show that the estimated parameters varied depending on which image was used to calibrate the camera. The final parameters of the calibration matrix, *K*, were the estimated average of the seven calibrations. The fact that the average error χ was relatively low, indicated that the pin-hole model was adequate to model the camera used in this experiment.

If the error χ is caused by lens distortion, the vectors between the coordinates p_e and p_p will form a uniform pattern. For example, if effects of radial distortion are dominant, the deviation between the coordinates will give vectors along lines crossing the image center. To get a visual interpretation of the deviation, the vectors formed by subtracting the coordinate sets, p_e - p_p, were plotted in the calibration image together with the coordinate set p_e. Since the coordinate sets were almost identical, the direction vector was amplified 50 times in order to give a visual impression.

Figure 44 shows the relative movement between the extracted coordinates used in the calibration and the estimated projected coordinates. The red lines show the 50 x amplified error vectors, p_e - p_p. The figure shows that the movements are rather random indicating that a compensation for lens distortion does not serve any purpose for this camera.

Figure 44 – Relative movement between extracted coordinates and the projected coordinates.

5.2.2.
Experiments with the X-Y Table

To investigate the accuracy of the vision based positioning techniques developed, an experiment was performed using the x-y table. A robotic manipulator (MA 2000 model) was used as reference object and placed 700mm in front of the web camera mounted on the table. The origin of the reference frame was defined as the camera center when the camera was positioned in the center of the x-y table. The axes of the reference frame were defined by the movement axes of the x-y table. Figure 45 shows the robot used as reference object in the experiment.

Figure 45 - The robot used as reference object in the experiment

The axes of the reference frame are shown in Figure 46. The x-y tables' movements were in x and z direction. The rotary stage on top of the table rotated around the y axis.

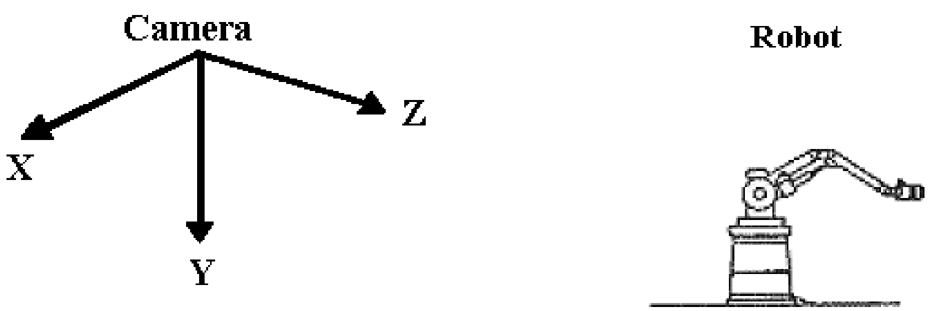

Figure 46 – Coordinate system of the x-y table

The purpose of the experiment was to investigate how accurate it is possible to estimate the cameras' movement relative to the chosen reference position using SIFT matches. Two methods were compared. The first method used quaternions to estimate the relative rotation matrix between the keypoint coordinates. The second method used a least square (LMS) estimate deduced in [21].

21 images of the robot were taken from different positions. The configurations of the different positions are show in Table 4. The SIFT keypoints from all the images were calculated and image number 11 was chosen as the reference image, since it was taken at the center of the reference frame with an angle of 0°. The relative positions of the other camera configurations were then estimated relative to the origin, which was the camera center when image 11 was taken. Image 11 had 1056 keypoints so there were 1056 possible matches. By using this image as reference and finding the respective keypoint matches in all the other images, a 21 x 1056 match matrix (nn) was formed. The i-th column in the matrix represents the keypoint number in each image corresponding to the i-th keypoint in the reference image. Since image 11 was the reference image, line 11 in the matrix is the vector [1:1056].

position	θ_x [°]	θ_y [°]	θ_z [°]	Δx [mm]	Δy [mm]	Δz [mm]
1	0	6	0	-100	0	0
2	0	5	0	-90	0	0
3	0	4	0	-80	0	0
4	0	4	0	-70	0	0
5	0	3	0	-60	0	0
6	0	3	0	-50	0	0
7	0	2	0	-40	0	0
8	0	1	0	-30	0	0
9	0	1	0	-20	0	0
10	0	0	0	-10	0	0
11	0	0	0	0	0	0
12	0	-1	0	10	0	0
13	0	-2	0	20	0	0
14	0	-2	0	30	0	0
15	0	-3	0	40	0	0
16	0	-4	0	50	0	0
17	0	-5	0	60	0	0
18	0	-5	0	70	0	0
19	0	-6	0	80	0	0
20	0	-7	0	90	0	0
21	0	-8	0	100	0	0

Table 4 – Position parameters for the experiment

A corresponding matrix (nn_o), of the same size as nn, contained the orientation of every keypoint.

To estimate one camera's position relative to the reference camera, two sets of corresponding 3D coordinates need to be generated. ${}^{1}p$ gives the keypoint coordinates relative to the reference image, and ${}^{2}p$ gives the same coordinates relative to the chosen camera position to be estimated.

The orientation of the camera relative to the axes of the reference frame was difficult to estimate accurately. Ideally, the cameras optical axis should correspond to the z-axis of the x-y table, but this was hard to measure visually. Therefore a rotation matrix, R_0, was introduced to denote the camera orientation deviation relative to the reference frame of the table:

$$R_0 = \cdot \begin{bmatrix} \cos\theta_{y_0} & 0 & \sin\theta_{y_0} \\ 0 & 1 & 0 \\ -\sin\theta_{y_0} & 0 & \cos\theta_{y_0} \end{bmatrix} \begin{bmatrix} \cos\theta_{x_0} & -\sin\theta_{x_0} & 0 \\ \sin\theta_{x_0} & \cos\theta_{x_0} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{z_0} & -\sin\theta_{z_0} \\ 0 & \sin\theta_{z_0} & \cos\theta_{z_0} \end{bmatrix} \qquad (197)$$

where θ_{x_0}, θ_{y_0} and θ_{z_0} represents the camera orientation deviation around axes x, y and z respectively. The true orientation of i-th camera configuration was then:

$$R_i = R_0 \cdot \begin{bmatrix} \cos\theta_{y_i} & 0 & \sin\theta_{y_i} \\ 0 & 1 & 0 \\ -\sin\theta_{y_i} & 0 & \cos\theta_{y_i} \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_{x_i} & -\sin\theta_{x_i} & 0 \\ \sin\theta_{x_i} & \cos\theta_{x_i} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{z_i} & -\sin\theta_{z_i} \\ 0 & \sin\theta_{z_i} & \cos\theta_{z_i} \end{bmatrix} \qquad (198)$$

To estimate the three rotation angles of R_0, their respective effects on the estimated coordinates needed to be interpreted. Since the translation between the cameras was along the x-axis, a change in θ_{y_0} meant that a coordinate set estimated by triangulation between the reference image and image number 1 would give a different average position in z direction than the same coordinates estimated by triangulation with image number 21. The three images 1, 11 and 21 had only six good keypoint matches in common. By changing θ_{y_0} in steps of 0.5°, the correct value was found when the coordinate sets gave the same average distance relative to the origin of the reference frame. The value of θ_{y_0} was found to be 6°. Since there was no movement in the y direction, θ_{x_0} did not have any apparent effect on the estimated coordinates and was therefore set to 0°.

The offset angle in the image plane θ_{z_0} influences how the estimated lines of sight through the keypoints coincide with their corresponding lines in the reference image. By assuming that choosing the right θ_{z_0} the number of lines coinciding within 1mm would reach a maximum, θ_{z_0} was found by counting the total number of coinciding lines between the reference image and all the images. θ_{z_0} was changed in steps of 0.1° until a maximum was reached at 2.0°. The offset configuration was then $\begin{bmatrix}\theta_{x_0} & \theta_{y_0} & \theta_{z_0}\end{bmatrix}^T = \begin{bmatrix}0 & 6° & -2°\end{bmatrix}^T$.

To estimate the 3D coordinate set, the procedure deduced in Section 3.9 was used. Since the accuracy of the estimated position depends highly on the intraocular distance, the coordinate set ${}^2p$ was estimated using relative intraocular distances from 20 mm to 110 mm. Having 21 images taken every 10mm along the x axis of the x-y table, many different possible combinations were obtained to achieve the desired relative distances. Images had to be chosen to estimate the 3D coordinate sets ${}^1p$ and ${}^2p$. First, the two camera positions used to estimate ${}^2p$ were chosen. The reference image defined the reference frame of the 3D coordinate set, ${}^1p$, so it could not be used to estimate the coordinate set, ${}^2p$. All the other remaining images could then be used to estimate ${}^1p$. However, the images taken close to the reference image didn't offer a sufficient depth resolution. The SIFT keypoints found in these experiments have an accuracy of approximately one pixel. If a depth resolution of e.g. 10 mm is required, the necessary intraocular distance can be calculated. Rearranging Eq.(121) gives:

$$d_{\text{int}} = -\left(\frac{1}{\alpha \Delta z}\right) z^2 \Delta s \tag{199}$$

By inserting Δz=1 pixel, α=975 pixels and z=700mm the required intraocular distance becomes 49 mm. This means that a limited number of images achieved the necessary intraocular distance relative to the reference image. Therefore only the images at least 49 mm from the reference image could be used to estimate ${}^1p$. Also, the two images used to estimate ${}^2p$ were not used to estimate ${}^{1}p$. For an intraocular distance of 100mm, 8 possible configuration pairs were considered, 2-12, 3-13, …, 10-20.

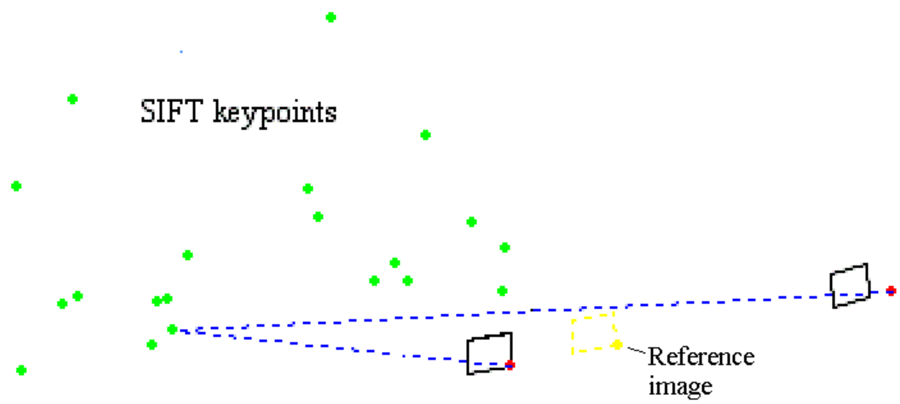

Figure 47 – Method to estimate the set of coordinates, ${}^{2}p$. The reference image is marked in yellow. The position of one of the two cameras is to be estimated relative to the reference image.

For example, if images 3 and 13 were used to estimate, ${}^{2}p$, these two images could not be used to calculate ${}^{1}p$. Therefore, the images [1 2 4 5 6 16 17 18 19 20 21] were used to estimate, ${}^{1}p$. This is shown in Figure 48.

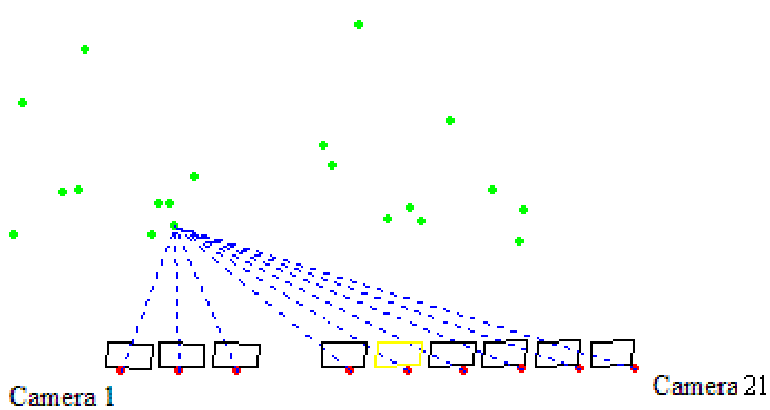

Figure 48 – Triangulation to estimate the coordinate set ${}^{1}p$ relative to the origin. The reference image is marked in yellow. The images closest to the reference image and the two cameras used to estimate the coordinate set ${}^{2}p$ were not used.

The estimated position error relative to the origin was then estimated for all possible combinations for every intraocular distance between 20mm and 110mm.

In this experiment, the keypoint orientation filter was set to eliminate any matches with an orientation deviation greater than 20°. All keypoints that did not coincide within 5 mm were also eliminated. After the coordinate sets ${}^{1}p$ and ${}^{2}p$ were found, RANSAC was run for 100 iterations to find the best matching model. The maximum distance to be accepted by the model, d_{lim}, was 20mm. The rotation matrix between ${}^{1}p$ and ${}^{2}p$ was estimated using, quaternions. The results from the experiments are shown in Table 5 and in Figure 49. The position errors are given in RMS.

d_o [mm]	20	30	40	50	60	70	80	90	100	110
Combinations	17	16	15	14	13	12	11	10	9	9
Error x [mm]	44.0	28.5	29.2	26.8	17.2	17.9	14.1	17.7	9.7	12.6
Error y [mm]	25.4	19.1	18.4	13.7	11.1	11.9	8.9	9.8	8.1	6.0
Error z [mm]	15.6	13.4	6.7	7.8	4.3	4.9	4.7	4.9	2.1	3.6

Table 5 – Position error as a function of intraocular distance using RANSAC

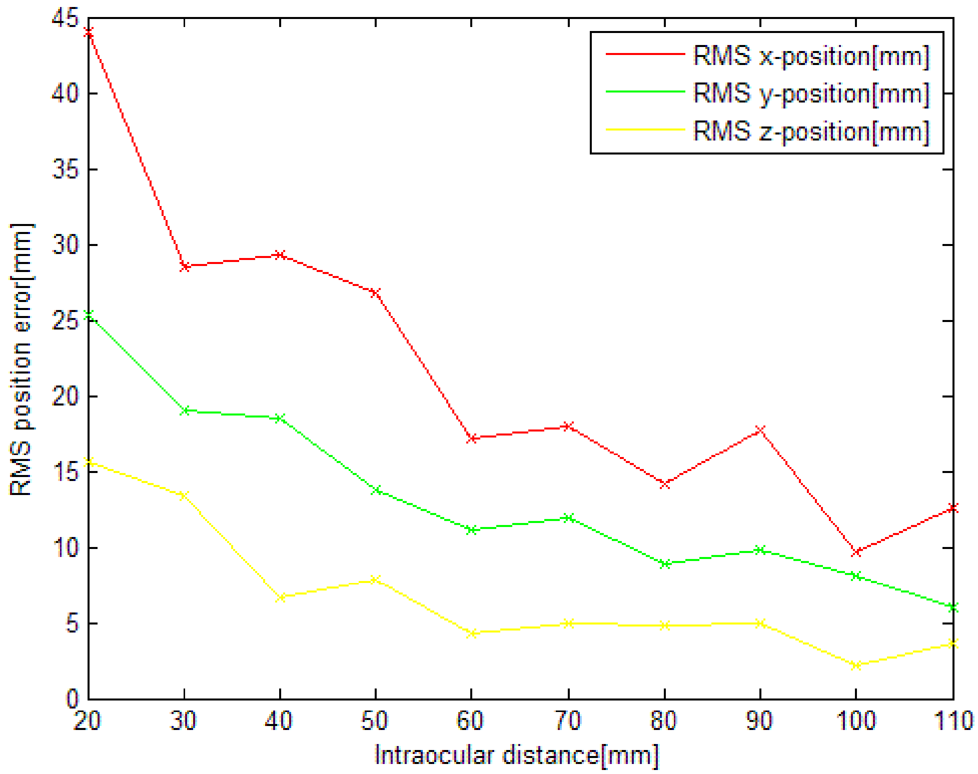

Figure 49 – RMS position error as a function of intraocular distance using RANSAC and quaternion rotation estimation.

The results show that the accuracy of the camera position improved as the intraocular distance increased. Achieving an accuracy of close to 10 mm in either direction was a satisfactory result, but in order to improve the estimated position further, a second procedure was used.

To improve the estimated rotation matrix, the coordinate pairs with the highest error ratio were eliminated. The error ratio limit was set to r_{lim}=0.1. The results can be seen in Table 6 and in Figure 50.

d_o [mm]	20	30	40	50	60	70	80	90	100	110
Combinations	17	16	15	14	13	12	11	10	9	9
Error x [mm]	42.6	24.7	23.3	12.1	7.6	9.7	8.9	4.8	8.1	9.3
Error y [mm]	26.2	22.9	14.3	9.7	10.8	9.7	7.8	7.9	4.9	5.4
Error z [mm]	15.0	11.4	6.7	6.2	4.5	3.5	5.0	4.4	2.0	3.3

Table 6 – Position error as a function of intraocular distance using RANSAC and error ratio elimination.

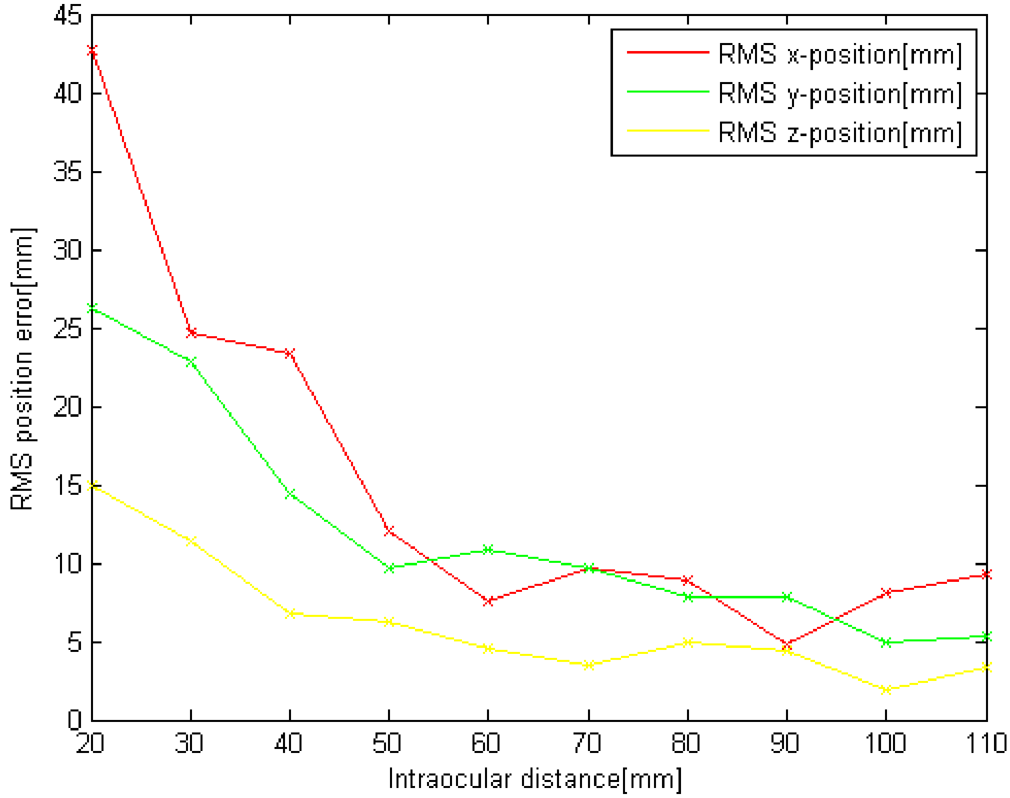

Figure 50 – Graph showing the average RMS position error as a function of the intraocular distance after the samples with the worst error ratio had been removed.

Figure 50 shows that the average estimated position error was below 10 mm in any direction, which was a slight improvement from the estimated position using only RANSAC.

The positioning experiments were then repeated using the least mean square(LMS) estimate of the rotation matrix deduced in [21], instead of the quaternion estimate. The same parameters were used. The results using RANSAC can be seen in Table 7 and Figure 51.

d_o [mm]	20	30	40	50	60	70	80	90	100	110
Combinations	17	16	15	14	13	12	11	10	9	9
Error x [mm]	38.9	37.7	26.0	21.0	21.1	17.5	17.9	14.2	12.7	14.4
Error y [mm]	19.5	12.7	15.8	10.7	11.9	9.5	7.8	4.4	7.1	6.4
Error z [mm]	15.8	12.8	6.9	7.8	4.4	4.6	4.8	4.7	2.3	3.3

Table 7 - Position error as a function of intraocular distance using RANSAC together with the least mean square estimated rotation matrix.

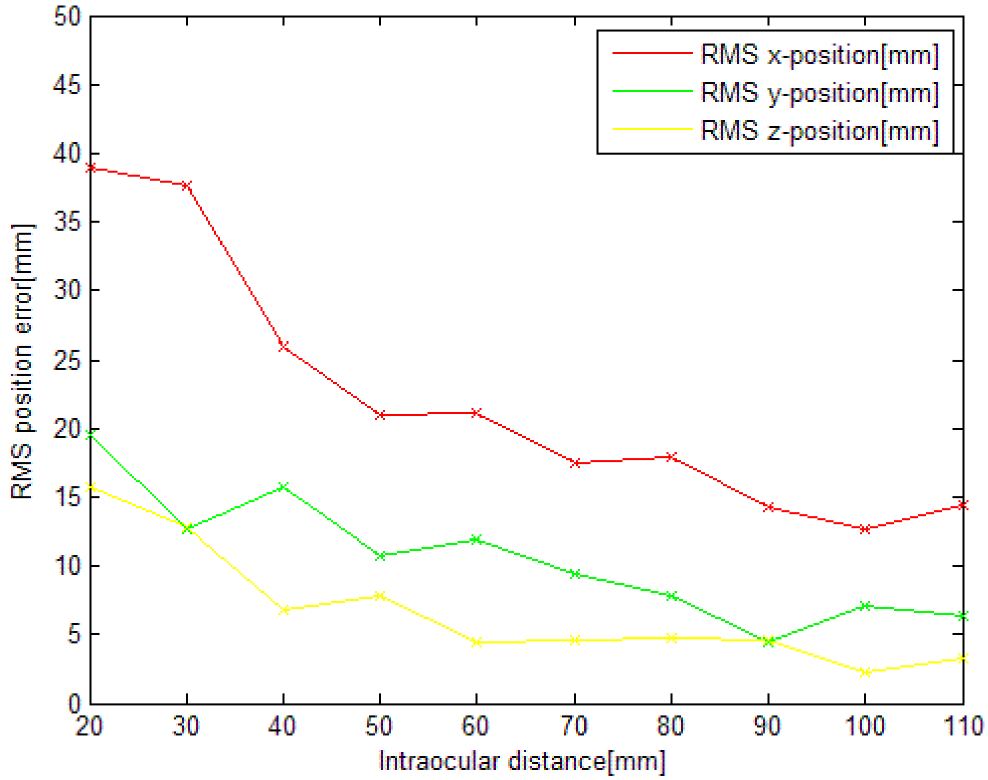

Figure 51 - RMS position error as a function of intraocular distance using RANSAC with LMS rotation estimate.

Figure 51 shows that by using RANSAC, the estimated position did not depend much on which algorithm was used to estimate the rotation matrix.

Further, the coordinates with the highest error ratio were eliminated using the same error limit as in the previous experiment. The results can be seen in Table 8 and in Figure 52.

d_o [mm]	20	30	40	50	60	70	80	90	100	110
Combinations	17	16	15	14	13	12	11	10	9	9
Error x [mm]	135.1	12.0	5.0	4.5	4.5	4.1	4.9	2.3	4.2	3.7
Error y [mm]	36.0	6.6	4.6	4.2	5.1	2.6	3.5	3.5	2.7	3.6
Error z [mm]	491.2	12.3	7.1	7.3	4.3	3.4	5.0	4.3	1.5	2.6

Table 8 - Position error as a function of intraocular distance using RANSAC together with the least mean square estimated rotation matrix and eliminating the coordinates with error ratio.

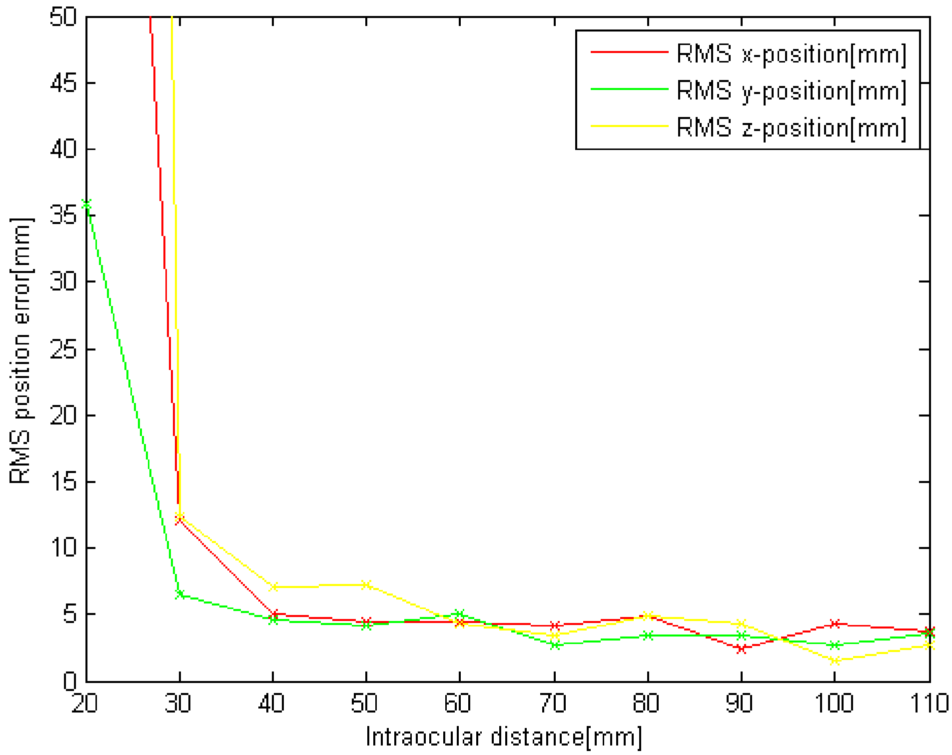

Figure 52 - RMS position error as a function of intraocular distance using RANSAC with LMS rotation estimate and eliminating the coordinates with a high error ratio.

Figure 52 shows that the LMS algorithm gave a more accurate estimate of the rotation matrix when the less accurate coordinate samples were removed. Achieving accuracy better than 5 mm in any direction was a better result than expected. Generally, the LMS algorithm is more sensitive to noise than the quaternion algorithm. Therefore, the quaternion algorithm usually gives a more accurate result when outliers are present. However, by managing to eliminate all the outliers and inaccurate coordinates, the LMS algorithm proved to give a better estimate than the quaternion algorithm.

5.3.
Calibration of an Underwater Camera

An underwater camera used in subsea interventions at great sea depths needs to be contained inside a thick rigid housing. This means that a thick glass separates the camera from the environment. It is therefore difficult to avoid the effects of lens distortion. The camera used in these experiments is the MCH-3000P which is an analog underwater video camera. Since the camera is analog the video signals were converted to digital format using a Pinnacle PCTV video card. The converter card had a resolution of 240 x 320 pixels. In order to achieve the same resolution as the web camera, the images extracted from the camera were interpolated using a cubic spline function in Matlab. The images were highly distorted. Since the camera was not aligned with the glass of the camera housing, the distortion was not symmetrical. This means that a radial distortion model would not be sufficient to model the projection errors. For this camera, a more sophisticated model needed to be used.

An appropriate model was found in [17]. The theory of this calibration procedure is explained in section 3.3.2. Both the radial and tangential components are estimated using a fifth degree polynomial. The parameters are estimated using the Levenberg-Marquart algorithm. The Matlab software calibration toolbox was used. It contains an implicit image correction that simulates the image coordinates of an ideal pinhole camera. From these coordinates, the true direction of the objects in the images can be estimated using the pinhole model. The 2D calibration rig was used to perform the calibration. The same coordinate extraction algorithm was used to find the coordinates of the corners as in the previous experiments with the webcam. Six corners of the rig were found manually and the exact positions of the remaining 72 corners were extracted using k-means line fitting. The calibration algorithm was iterative and therefore needed an initial guess for some of the camera parameters in order to converge. The resolution and size of the image sensor and the focal length had to be given. Since the image was interpolated to give a resolution of 480 x 640 pixels, the size of the sensor was given as 4.8 x 6.4 mm, giving 100 pixels/mm. The initial focal length was given different values between 4 and 6 mm, corresponding to 400 and 600 pixels. Every experiment, the algorithm converged to the same calibration parameters. The parameters are given in Table 9. The image distortion can then estimated using Eqs.(75) and (76).

s_u	f [pix]	u_o [pix]	v_o [pix]	k_1	k_2	T_1	T_2
1.061	510.06	298.45	229.20	$1.38 \cdot 10^{-2}$	$1.45 \cdot 10^{-3}$	$9.20 \cdot 10^{-5}$	$2.28 \cdot 10^{-4}$

Table 9 –Calibration parameters for the underwater camera in air, where s_u is the aspect ratio, f is the focal length, k_1 and k_2 are the radial distortion coefficients, T_1 and T_2 are the tangential distortion coefficients and (u_o,v_o) denotes coordinates of the image center.

Figure 53 shows the corrected image coordinates in green. The red lines show the location of the corresponding image coordinates. It is obvious from the figure that the images from the camera suffered from large barrel distortion.

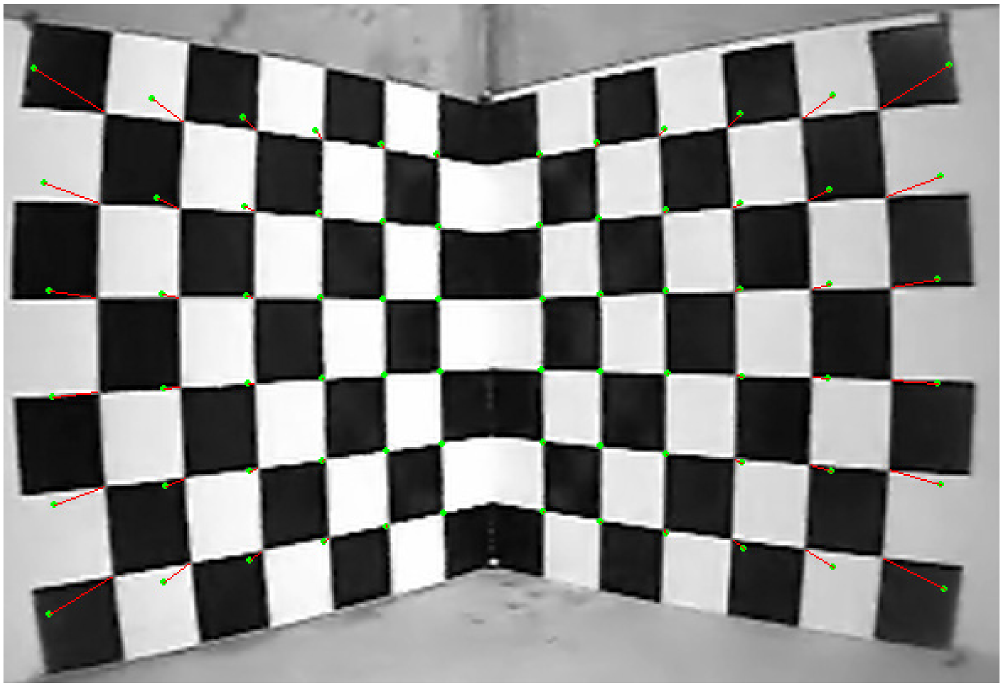

Figure 53 –The corrected image coordinates relative to their respective extracted image coordinates

5.4.
Position Estimation using the Underwater Camera

After the underwater camera had been calibrated, the camera could be used in experiments with the x-y table. Since the projection geometry of the camera was very complex, the experiments were not carried out using SIFT. In order to avoid any uncertainties regarding false keypoint matches, the calibration rig was used as a reference object. The semi automatic algorithm used in the camera calibration was used to extract the coordinates of the corners. The coordinates were verified manually so that there would be 72 correct matches to estimate the movement of the camera. The underwater camera had a shorter focal length than the web camera. This meant that it had to be placed fairly close to the reference object in order to be able to extract the image coordinates. The calibration rig was placed 500 mm from the center of the x-y table. The camera was mounted on a support delrin structure on top of the x-y table, see Figure 54. The support was originally made to mount two underwater cameras side by side with an intraocular distance of 200mm. Since the cameras´ optical axes were not aligned with the camera housing, the cameras were not used in a stereo cam configuration. The angle deviation of the two cameras would have been too big, meaning that the cameras would not be able to focus on the same object when mounted side by side in a parallel structure. The camera was mounted at the left aperture in the support. Since the camera was mounted 100mm from the rotation center of the table, the movement of the camera was more complex than the movement of the web camera mounted at the rotation axis of the table. The different camera positions used in the experiment were put in table 10.

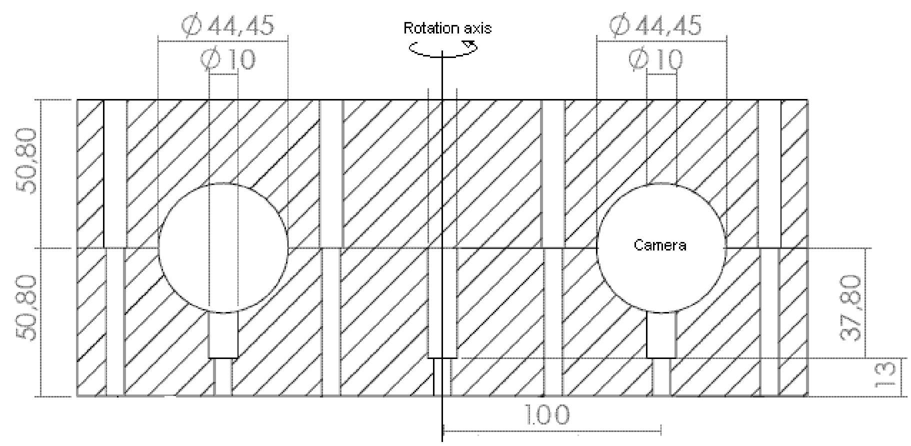

Figure 54 – Attachment support for the underwater camera.

position	θ_x [°]	θ_y [°]	θ_z [°]	Δx [mm]	Δy [mm]	Δz [mm]
1	0	6	0	-79.5	0	10.4
2	0	5	0	-69.6	0	8.7
3	0	4	0	-59.8	0	7.0
4	0	3	0	-49.9	0	5.2
5	0	2	0	-40.0	0	3.5
6	0	2	0	-29.9	0	3.5
7	0	1	0	-20.0	0	1.7
8	0	0	0	-10.0	0	0
9	0	0	0	0	0	0
10	0	0	0	10	0	0
11	0	-2	0	20.1	0	-3.5
12	0	-3	0	30.1	0	-5.2
13	0	-4	0	40.2	0	-7.0
14	0	-6	0	50.5	0	-10.4
15	0	-8	0	61.0	0	-13.9
16	0	-10	0	71.5	0	-17.4
17	0	-12	0	82.2	0	-20.8

Table 10 – Position parameters for the experiment with the underwater camera.

The experiments followed the same procedure as the experiments with the web camera. Image 9 was chosen as the reference image. The images closest to the reference image, 7, 8, 10 and 11 were not used to estimate ${}^{1}p$. The cameras' orientation deviation was found using the same procedure as in the previous experiments. The fixed orientation deviation converged to:

$$\begin{bmatrix}\theta_{X0} & \theta_{Y0} & \theta_{Z0}\end{bmatrix}^T = \begin{bmatrix}0^\circ & 7.5^\circ & -1.3^\circ\end{bmatrix}^T.$$

Experiments were performed using different intraocular distances to estimate ${}^{2}p$. The desired intraocular distance gave different number of possible camera combinations. An intraocular distance of 10mm gave 14 different camera combinations, while an intraocular distance of 100mm gave only 7 combinations.

The average RMS position error as a function of the intraocular distance is shown in Table 11 and Figure 55. In this experiment, all the 72 coordinates were used. The rotation matrix was estimated using quaternions.

d_o [mm]	10	20	30	40	50	60	70	80	90	100
Combinations	14	13	12	11	10	9	8	7	8	7
Error x [mm]	60.6	38.2	39.8	36.2	29.4	27.6	17.0	16.4	13.8	10.9
Error y [mm]	22.5	13.5	10.1	7.9	6.0	4.4	3.9	4.2	3.6	2.7
Error z [mm]	12.1	7.5	7.7	8.7	8.8	9.8	9.8	10.4	11.8	12.2

Table 11 – Position error as a function of intraocular distance using the underwater camera, all 72 coordinates, and quaternion rotation estimation.

Figure 55 shows that the estimated position errors were much smaller when the intraocular distance was 100 mm. The fact that the depth error (z direction) did not decrease when the intraocular distance increased might be caused by the fact that it was difficult to estimate the exact position of the camera center. When the camera rotated on the support, this would affect the estimated relative intraocular distance between the views.

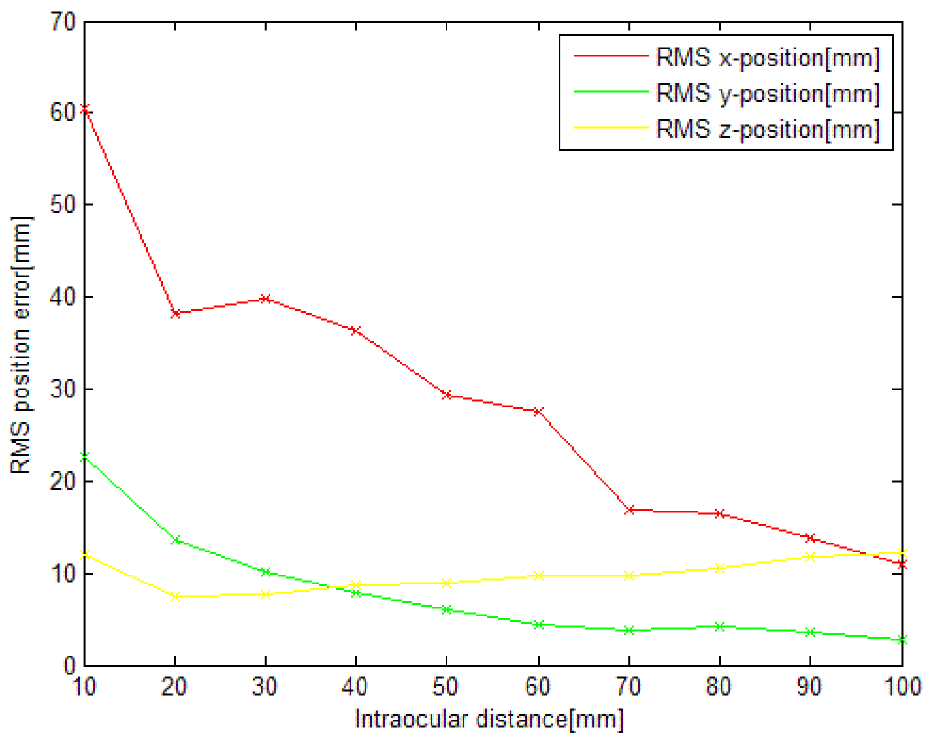

Figure 55 – Position accuracy as a function of the intraocular distance using all 72 coordinates and quaternion rotation estimation.

In the experiments with the web camera, the elimination of coordinates with a high error ratio improved the estimated rotation matrix between the coordinate sets. To see if this had the same effect when using the underwater camera, the samples were eliminated sequentially until all the coordinates had ratio $r_{\lim} = 0.1$. The results can be seen in Table 12 and Figure 56

d_o [mm]	10	20	30	40	50	60	70	80	90	100
Combinations	14	13	12	11	10	9	8	7	8	7
Error x [mm]	65.3	42.1	35.2	24.5	21.7	23.9	16.8	12.5	8.7	8.6
Error y [mm]	54.2	25.1	24.2	10.8	9.6	9.8	5.3	6.7	5.8	4.1
Error z [mm]	16.5	7.7	9.3	10.4	9.8	8.6	8.7	10.5	10.9	12.1

Table 12 - Position accuracy as a function of the intraocular distance after the coordinates with a large error ratio had been eliminated.

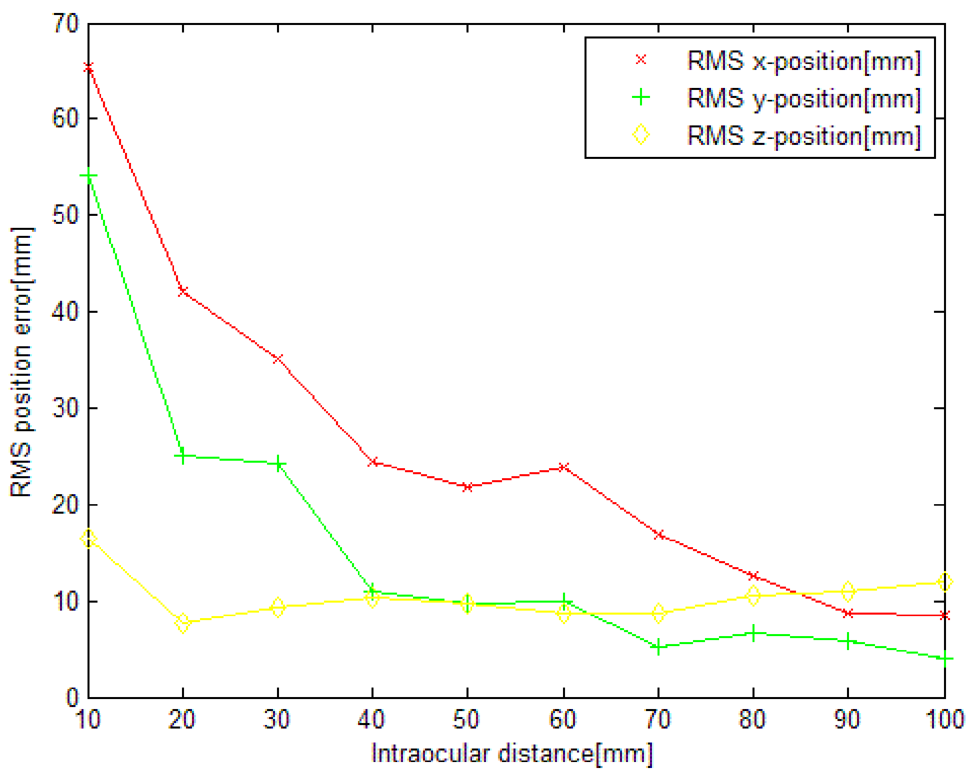

Figure 56 - Position accuracy as a function of the intraocular distance after the coordinates with a large error ratio had been eliminated.

Figure 56 shows that eliminating the coordinates with the highest error ratio did not improve much the estimated position. All the coordinates were verified manually in the photos of the calibration rig, so there were no outliers. There were many coordinates and they were also evenly spread, so any bad measurements would not influence too much the result.

The same experiments were carried out again, but substituting the quaternion rotation estimate with the LMS algorithm. First, all the 72 coordinates were used. The results are shown in Table 13 and Figure 57.

d_o [mm]	10	20	30	40	50	60	70	80	90	100
Combinations	14	13	12	11	10	9	8	7	8	7
Error x [mm]	11.2	6.9	5.9	4.5	3.0	2.5	1.8	1.2	1.5	1.9
Error y [mm]	17.2	8.2	7.3	5.5	4.4	3.7	2.8	2.9	2.7	3.0
Error z [mm]	9.9	6.8	7.1	7.8	7.8	8.5	8.6	8.9	10.3	11.2

Table 13 – Position error as a function of intraocular distance using the underwater camera using all 72 coordinates and LMS rotation estimation

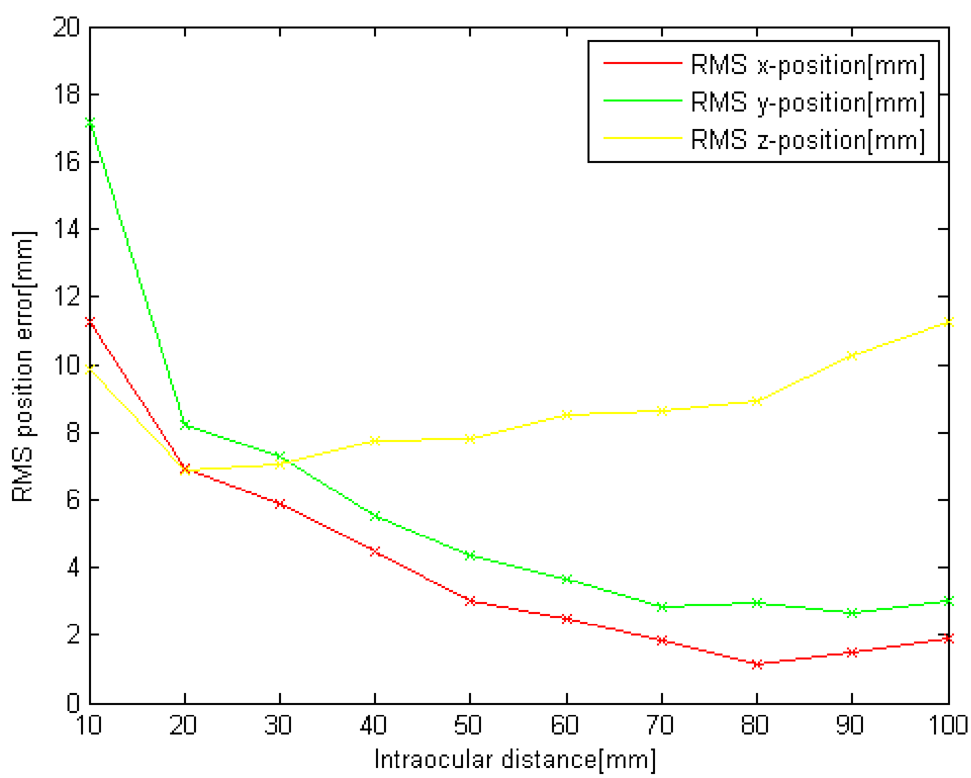

Figure 57 - Position error as a function of intraocular distance with the underwater camera using all 72 coordinates and LMS rotation estimation

The graph shows that the LMS algorithm worked much better than the quaternion algorithm in this experiment also. The estimated positioning error in the z direction remained virtually the same as the previous experiment since the rotation matrix does not affect the depth estimate. Achieving a position estimate in x and y direction smaller than 4 mm better result than expected.

Again, the coordinates with a high error ratio were eliminated. The results can be seen in Table 14 and Figure 58.

d_o [mm]	10	20	30	40	50	60	70	80	90	100
Combinations	14	13	12	11	10	9	8	7	8	7
Error x [mm]	13.6	7.1	5.0	4.8	3.6	3.7	3.1	3.0	2.5	3.1
Error y [mm]	6.7	4.2	3.5	2.5	2.9	3.5	2.5	2.1	2.0	2.5
Error z [mm]	6.8	8.2	7.3	7.4	8.2	7.6	8.2	8.8	10.2	11.2

Table 14 - Position accuracy as a function of the intraocular distance after the coordinates with a large error ratio had been eliminated.

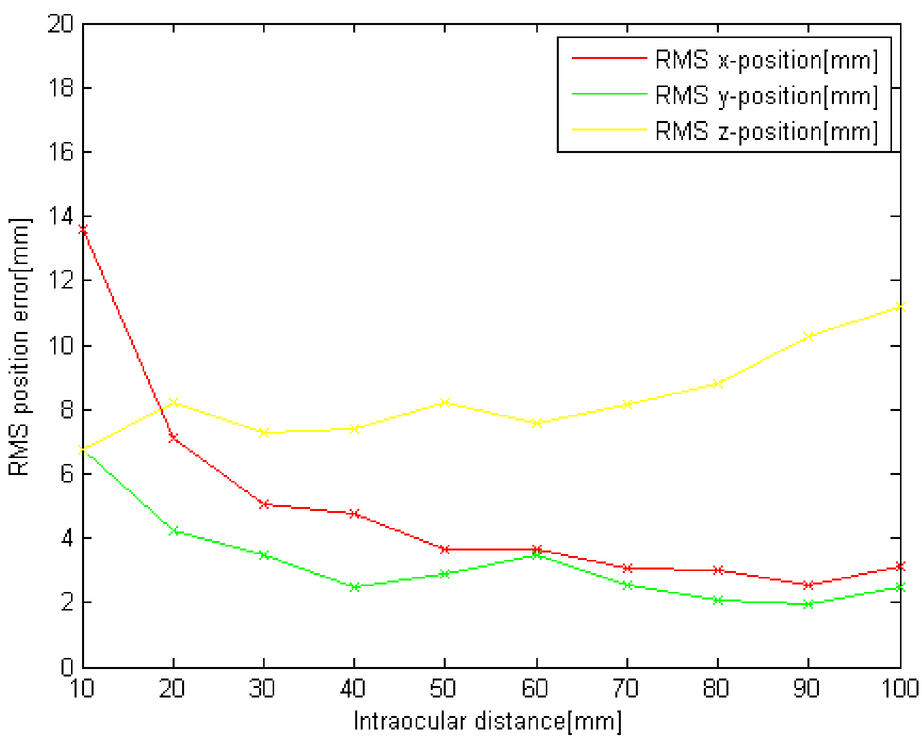

Figure 58 - Position accuracy as a function of the intraocular distance after the coordinates with a large error ratio had been eliminated

These experiments show that the underwater camera manages to estimate its position with a high accuracy despite the huge distortion. Due to the wide angle lens of the camera, the size of the reference object can be large relative to the distance between the camera and rig. This improves the accuracy of the estimated rotation matrix. The fact that the depth estimate was not very accurate indicates that the camera calibration might have some discrepancies. The exact position of the camera center was also difficult to estimate and might be the cause of this error.

Since all the coordinates were verified manually, these experiments only served to demonstrate the potential ability of the underwater camera to be used in position estimation. To find out how well it works with SIFT keypoints in sub sea interventions, tests need to be performed under water with objects similar to those that will be encountered in the robots' work environment.

5.5.
Camera Calibration performed Underwater

Since the developed methods are to be used in underwater interventions, the underwater camera used in the previous section was calibrated in a water tank. The calibration rig was lowered into a container filled with water. An image was then taken of the rig with the camera submerged. The image can be seen in Figure 59. The coordinates of the corners were extracted and the calibration parameters were estimated. The calibration parameters are given in Table 15.

s_u	F [pix]	u_o [pix]	v_o [pix]	k_1	k_2	T_1	T_2
1.064	662.6	299.24	239.45	$4.36 \cdot 10^{-3}$	$7.04 \cdot 10^{-4}$	$6.86 \cdot 10^{-4}$	$6.62 \cdot 10^{-4}$

Table 15 – Calibration parameters for the underwater camera in air, where s_u is the aspect ratio, f is the focal length, k_1 and k_2 are the radial distortion coefficients, T_1 and T_2 are the tangential distortion coefficients and (u_o, v_o) denotes coordinates of the image center.

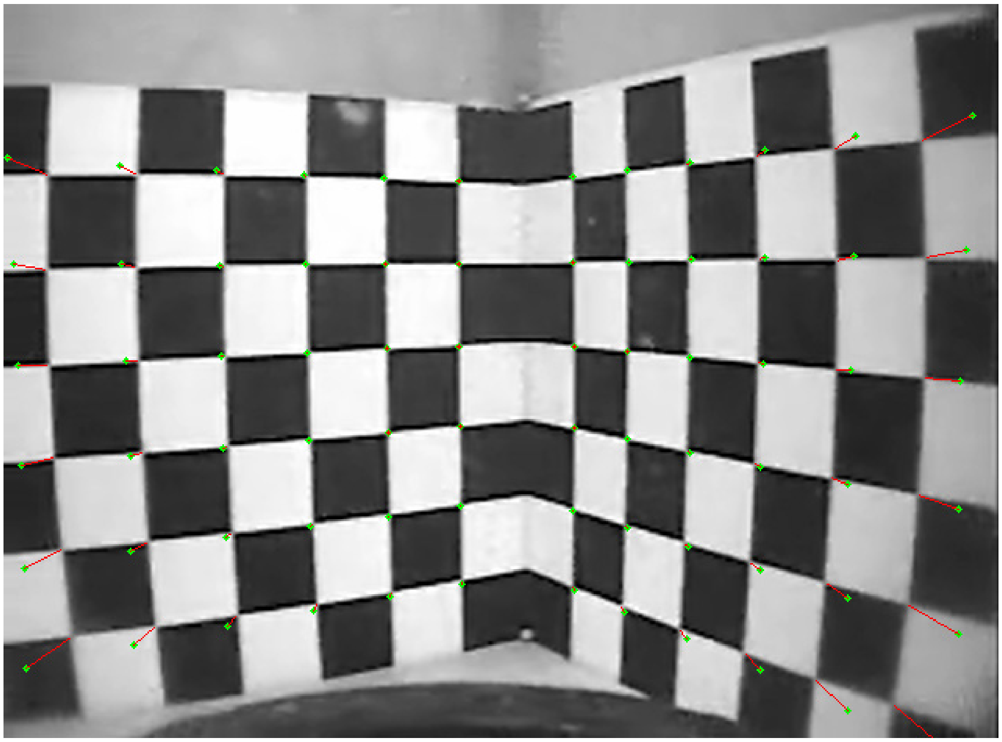

Figure 59 – Underwater calibration. The green points are the corrected image coordinates and the red lines show their respective image coordinates

It is apparent that the camera suffered from large distortion even when submerged. From Table 15 it can be seen that the nominal focal length increased compared to the calibration performed in air and the distortion coefficients are smaller. Since the camera calibration algorithm managed to cope with the huge distortion in air, there is a high probability that the camera can be used in positioning under water, however with the accuracy limitations shown in the previous tables.