Validation of Stereo Matching for Robot Navigation

Jörgen Lidholm Giacomo Spampinato Lars Asplund
School of Innovation, Design and Engineering, Mälardalen University, Sweden
E-mail: {jorgen.lidholm,giacomo.spampinato,lars.asplund}@mdh.se

Abstract

This paper presents results from experiments on visual stereo matching for robot navigation. Visual features are stereo paired with respect to their pixel position. Stereo triangulating all paired visual features results in a set of landmarks whereof a subset are true landmarks. Constraining the horizontal disparity limits the amount of spurious matches. The stereo matching is validated by finding which landmarks survives short motions measured with a complementary navigation system, like odometry, thus transferring the stereo matching problem from two to three dimensional space and robot motion is estimated from the landmarks surviving the motion. The results from our experiments show that the spurious matching algorithm for stereo matching validation works and that the system is able to estimate the motion.

1. Introduction

Encoder based odometry is probably the most common way to measure the robot motion. On omnidirectional and differential drive robots, slip is a problem that affects the estimation of the robot motion. Over time, encoder based odometry is therefore not a reliable method for determining the robot motion. Because of the simplicity and precision for short motions, encoders are, however, still used as a complementary sensor.

Other sensors like LIDAR (LIgth Detection And Ranging) and Laser scanners are also used to solve the localization problem. A LIDAR uses the time of flight to measure the distance to an object with high precision. A laser scanner on the other hand is typically dependent on special reflective surfaces that are triangulated providing fairly good accuracy of the localization. A LIDAR also enables mapping of the environment.

With increasingly powerful computers, vision has become a more commonly used sensor for detecting objects or features in the environment which can form landmarks. The landmarks can be used to build a map and recovering the robots current position in the map. This is known as SLAM, simultaneous localization and mapping.

Vision has both the drawback and benefit of being information dense. The information density provides the opportunity to select different properties from an image to extract, e.g. corners, edges and intensity regions.

Because of the density it also require substantial processing of the image before the data can be used, this can be compared with the human brain where-of visual cortex takes a large part.

General purpose computers have become increasingly powerful with a multitude of special purpose instructions allowing for parallel operations, the software compiler is responsible for optimizing the code to utilize the CPU in the best way possible. The main benefit of using general purpose computers is the simplicity to implement and test the code.

There are a number of toolkits that allows for utilizing the Graphical Processing Units (GPU) commonly found in regular desktop computers. GPUs are specifically made for matrix operations that are common in image processing [8][12][4].

Historically DSPs (Digital Signal Processing units) have been used for image processing and still are. These devices are equipped with special instructions that they do very fast which makes them suitable for image processing.

In robotics a general purpose computer fast enough to perform image processing as well as all other task related to robot behaviour is rarely available, in [13] an implementation of Harris and Stephens corner and edge detector is tested on an 2.6 GHz Opteron computer unable to perform in real-time on approximately 11 mega pixel data per second. And furthermore a GPU is almost never available.

A Field Programmable Gate Array (FPGA) makes it possible to construct pipelined processing for a very specific task. In this paper we present experiments on a stereo matching algorithm for robot navigation based on an implementation of Harris and Stephens corner and edge detector [2] on an FPGA which performs at about 9 mega pixel per second performance with the ultimate goal to also implement the spurious matching [6] and pose estimation, presented in this paper, on the FPGA.

978-1-4244-2728-4/09/$25.00 ©2009 IEEE

2. Theory

2.1. Definitions

Camera parameters

A camera consists of an image sensor and a lens. An image acquired from a camera is distorted mainly because of the properties of the lens. The distortion can in most cases be modeled by a radial and a tangential component. The center of the distortion is called the principal point, $\overline{PP}$. The location of $\overline{PP}$ in a camera is affected by how the lens is mounted with respect to the image sensor and would be the same as the center of the image in a perfect camera. The focal length f of a camera is defined as the distance from the optical center of the lens to the focal point and it determines the viewing angle, a short focal length results in a greater field of view and a long focal length results in a smaller field of view.

Stereo cameras

A stereo camera provides the ability to triangulate an object without moving the camera, given that the object correspondence between the stereo image pair is known. Each pixel on the camera sensor corresponds to a vertical and horizontal direction to the object from which the light is projected onto the image sensor. The camera separation is known as *baseline*.

Coordinate system

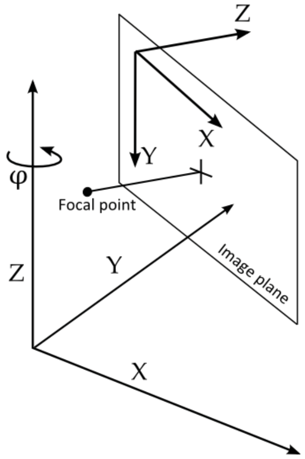

Figure 1. Coordinate system with respect to the camera.

When working with vision one have to consider a number of different coordinate systems. Each camera has a coordinate system, see figure 1. Transforming a coordinate from one cameras reference frame to the others consists of a rotation $\mathbf{R}$ and a translation $\overline{T}$, $\overline{X}_1 = \mathbf{R}\overline{X}_2 + \overline{T}$. The camera has a position and orientation with respect to the robots reference frame, with six degrees of freedom.

Considering an indoor environment with no slopes the robots location in the world can be described by (x, y, φ). Landmarks in the environment has a position with three degrees of freedom, not considering orientation.

Visual features and landmarks

A visual feature is a property of an object that can be found in an image. A common method for extracting visual feature from an image is to use a corner or edge detector. Visual features will be referred to as features from this point on. There are a number of different feature extractors with different properties [2][10][15]. In a stereo camera system the visibility of a feature depends on occlusion and the orientation of the cameras and the cameras field of view.

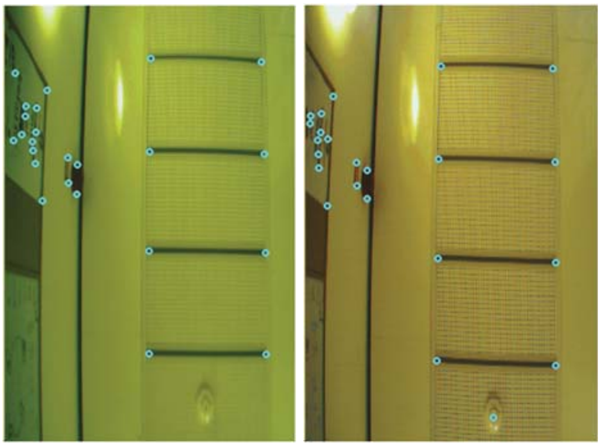

Figure 2. A photo of the ceiling and corner features extracted using Harris and Stephens combined corner and edge detector, the corners are clustered with a 10 pixel radius.

Figure 2 shows images from a stereo camera with corner features extracted using Harris and Stephens corner detector. A corner feature have a strength associated with it called cornerness value, only the strongest corner features are used and here clustered with a 10 pixel radius. Not all features are located in both frames, in this case a result from the difference in field of view.

A landmark is a point in object space which can be an object, part of an object or a property of an object. To make a feature or landmark more distinctive from other features or landmarks a *local descriptor* can be associated with it. A local descriptor characterizes the region of the feature and some examples are Spin-Images, SIFT and SURF [5][7][9]. Local descriptors simplifies the correspondence problem, but increases processing requirements.

2.2. Navigation process overview

Vision based navigation is a process which consists of finding features in a set of images from at least two different points of view (usually solved with a stereo camera), determining the correspondence and constructing landmarks that are used to determine the egomotion.

Finding the stereo correspondence of features is not a trivial problem, many solve this in the two dimensional plane using a statistical method like *sum of squared difference* or *cross-correlation* as a measure of similarity. Local descriptors simplifies the stereo matching problem, but does not solve it completely. Instead of solving the problem in two dimensional space, we have chosen to solve it in three dimensional space.

Landmarks are triangulated from extracted features, F_l and F_r, from left and right camera, only considering geometric constraints. The correspondence problem is not solved here, resulting in spurious matches and invalid landmarks.

Two sets of potential landmarks, called the spurious sets L_s, are evaluated with respect to the delta motion measured with odometry. Given the delta motion a subset of the landmarks in the spurious set should correspond to that motion, L_e.

The egomotion is estimated from the landmark set L_e with respect to landmarks in the map, that are already validated and found valid.

The map is updated with newly discovered landmarks when the new position is acquired. See figure 3.

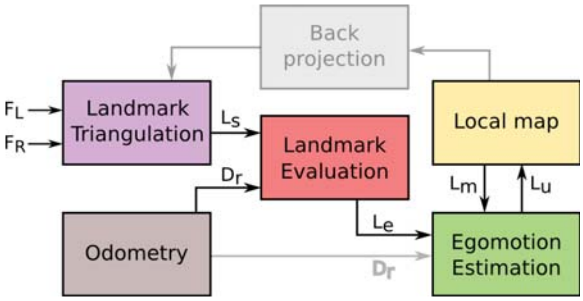

Figure 3. Navigation process

2.3. Stereo triangulation

A landmark $\overline{L} = [x, y, z]^T$ is projected at pixel $[p_x, p_y]^T$ on the image sensor. By using the pinhole camera model, a known focal length, and pixel separation it is possible to find a pointing vector from the camera towards the landmark. With a stereo camera two of these vectors give all necessary information to triangulate the landmark location in object space. The point closest to both lines formed by the pointing vectors is the estimate of the exact location. The point closest to both lines is on the middle of the the shortest line perpendicular to both lines. Under perfect conditions, this line should of length zero, thus the point would be where these lines cross, see figure 4. This method was inspired by [14].

The pointing vector $\overline{P}$ is normalized so that a constant α multiplied with the vector gives us the location of the landmark with respect to the camera. In a stereo pair we have two cameras with a displacement $\overline{T}$ and a rotation $\mathbf{R}$. Given these properties we can form the following equation.

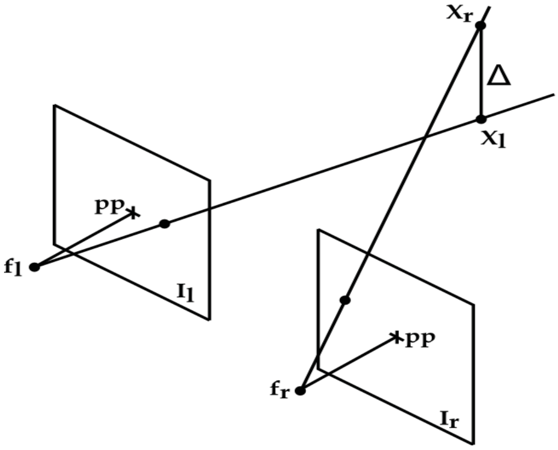

Figure 4. Stereo triangulation

$$\overline{P}_r = \mathbf{R}\overline{P}_l + \overline{T} \tag{2.1}$$

$$\alpha\overline{P}_r = \beta(\mathbf{R}\overline{P}_l) + \overline{T} \tag{2.2}$$

Because of discretization and calibration imperfection the equation 2.2 is incomplete. A vector, Δ formed by the shortest distance between the lines added to the right hand side gives a complete equation.

$$\alpha\overline{P}_r = \beta(\mathbf{R}\overline{P}_l) + \overline{T} + \overline{\Delta} \tag{2.3}$$

$\mathbf{R}, \overline{T}, \overline{P}_r$ and $\overline{P}_l$ are all known. By using the property that $\overline{\Delta}$ is perpendicular to both vectors $\overline{P}_r$ and $\overline{P}_l$ we can build an equation system to find α and β, for simplicity we say that $\overline{U}$ is $\overline{P}_l$ rotated to the same reference frame as $\overline{P}_r$.

$$\overline{U} = \mathbf{R}\overline{P}_l \tag{2.4}$$

$$\overline{P}_r^T \cdot \overline{\Delta} = 0 \tag{2.5}$$

$$\overline{U}^T \cdot \overline{\Delta} = 0 \tag{2.6}$$

$$q = \overline{P}_r^T \cdot \overline{U} \tag{2.7}$$

$$t_u = \overline{T}^T \cdot \overline{U} \tag{2.8}$$

$$t_r = \overline{T}^T \cdot \overline{P}_r \tag{2.9}$$

Since the vectors $\overline{P}_r$ and $\overline{U}$ are normalized we also have:

$$\overline{P}_r^T \cdot \overline{P}_r = 1 \tag{2.10}$$

$$\overline{U}^T \cdot \overline{U} = 1 \tag{2.11}$$

By multiplying equation 2.3 with $\overline{P}_r$ and $\overline{U}$ the following two equations are obtained:

$$\alpha = \beta q + t_r + 0 \tag{2.12}$$

$$\beta = \alpha q - t_u \tag{2.13}$$

By inserting 2.13 into 2.12 and the other way around equations 2.14 and 2.15 are obtained:

$$\alpha = \frac{-t_u q + t_r}{1 - q^2} \tag{2.14}$$

$$\beta = \frac{t_r q - t_u}{1 - q^2} \tag{2.15}$$

Once we have both α and β we can calculate the relative vector from each camera to the landmark in each cameras reference frame:

$$\overline{X}_r = \alpha \overline{P}_r \tag{2.16}$$

$$\overline{X}_l = \beta \overline{P}_l \tag{2.17}$$

$$\overline{X}_{landmark} = 0.5(\overline{X}_r + \overline{X}_l) \tag{2.18}$$

2.4. Back projection of landmarks onto the image sensor

By utilizing the properties of the triangles formed by the focal length and pixel coordinates and the relative landmark location, it is possible to find the corresponding pixel coordinate of a landmark and the opposite [16].

$$d = f\frac{b}{z} \tag{2.19}$$

$$p_x = f\frac{x}{z} \tag{2.20}$$

$$p_y = f\frac{y}{z} \tag{2.21}$$

Where b is the baseline, or camera separation, d is the pixel separation and x, y, z is the space location. This is only valid for the pin-hole camera model. The rectification should, however give a camera model close to the pin-hole camera. Back projecting stable landmarks allows for removing features before the spurious matching step, this way the landmark set that requires evaluation is reduced without reducing landmarks used in egomotion estimation.

This method could also be used to calculate the location of a landmark in space. For reasons of robustness we have chosen the stereo triangulation method described in the previous section.

2.5. Planar egomotion estimation

In order make the robot localization robust to the incremental errors coming from odometry, the stereo matching information from the stereo camera system has been used to estimate the robot motion from a different perspective. More specifically, the robot movement has been calculated based on two sets of 3D relative landmarks positions with respect to the robot in two subsequent instants of time. The egomotion problem is known in literature as the robot motion estimation based on video feedback information [1][11]. In this paper a planar egomotion problem has been addressed assuming that the robot motion is constrained to move in a plane and completely described by three degrees of freedom (two for position and one for the orientation). The planar ego-motion estimation is completely described by three parameters affecting the rototranslation of the robot in the motion plane: $\delta\overline{T} = [\delta t_x, \delta t_y]^T$, and $\mathbf{R}_z(\delta\varphi)$. The three dimensional position of the i-th feature with respect to the robot in two subsequent instants of time are expressed in (2.23).

$$\overline{P}_t^i = \mathbf{R}_z(\delta\varphi)\overline{P}_{t+1}^i + \delta\overline{T} \tag{2.22}$$

$$\begin{bmatrix} p_{x_t}^i \\ p_{y_t}^i \end{bmatrix} = \begin{bmatrix} b & -a \\ a & b \end{bmatrix} \begin{bmatrix} p_{x_{t+1}}^i \\ p_{y_{t+1}}^i \end{bmatrix} + \begin{bmatrix} \delta t_x \\ \delta t_y \end{bmatrix} \tag{2.23}$$

Where $a = sin(\delta\varphi)$ and $b = cos(\delta\varphi)$.

Rearranging the relation (2.23), the parameters vector $\overline{X} = [\delta t_x, \delta t_y, a, b]$ can be calculated by a least square method as shown in (2.25)

$$\overline{B} = \mathbf{A}\overline{X} \tag{2.24}$$

$$\begin{bmatrix} \vdots \\ p_{x_t}^i \\ p_{y_t}^i \\ \vdots \end{bmatrix} = \begin{bmatrix} & & \vdots & \\ 1 & 0 & -p_{y_{t+1}}^i & p_{x_{t+1}}^i \\ 0 & 1 & p_{x_{t+1}}^i & p_{y_{t+1}}^i \\ & & \vdots & \end{bmatrix} \begin{bmatrix} \delta t_x \\ \delta t_y \\ a \\ b \end{bmatrix} \tag{2.25}$$

$$\overline{X} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\overline{B} \tag{2.26}$$

The more features tracked during the motion, the more accurate the ego-motion estimation will be.

3. Experiments

3.1. Experimental Platform

Our camera system is modularized with a main board, an FPGA board and camera boards. See block diagram in figure 5.

Main board

The main board has connectors for four cameras and carries the FPGA board, it contains a USB to parallel converter, TTL level serial communication connection, a program selector, control pins for program selection, and hardware reset. The control pins overrides the program selector, enabling a microcontroller to select and load a new configuration into the FPGA from flash in about one second.

Camera

In the experiments for this paper we have used two OmniVision OV7610 CMOS camera sensors with a 640×480 pixel array with 8.4×8.4μm pixel separation. The image sensor is capable of 30Hz frame rate in progressive mode. The lens has fixed focus and a focal length of 6.5 mm.

FPGA board

The FPGA board incorporates an Xilinx XC2V8000 FPGA (Field Programmable Gate Array), 512 Mbit of SDRAM, 256 Mbit of flash memory (for holding up to 8 configurations), a CPLD (Complex Programmable Logic Device) for loading a configuration into the FPGA from flash memory. The board has 20 LVTTL pins that can be used to connect an expansion board.

The FPGA comprises eight million gate equivalents, up to 168, 18×18 multipliers and 3'024 Kbits of block ram. The FPGA currently runs at 50MHz and can be increased up to 200MHz.

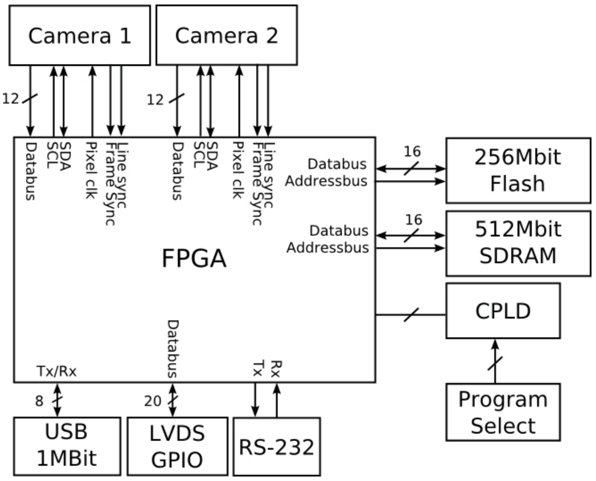

Figure 5. Camera system with two Omni-Vision OV7610 CMOS sensors and a Xilinx XC2V8000 FPGA

3.2. Stereo Camera Calibration and rectification

There are several methods for finding the intrinsic parameters of a camera, a common method is to take a number of photos from different views of a checkered pattern. By extracting the corners in the checker pattern and knowing the actual inter-distance of the corners, it is possible to find focal length, the actual camera center, radial and tangential distortion. These are known as the intrinsic parameters. Using a similar method it is also possible to find the extrinsic parameters of a stereo camera. The extrinsic parameters consists of a rotation and a translation from one of the cameras coordinate system to the others. See [3][17] for more information on calibration.

Calibration of a camera system is often performed once off-line. But to use the image data in a proper way the complete image has to be rectified for each frame. To correct the whole raw image requires quite many operations to both correct the position of the pixels according to the distortions factors but also to interpolate image data to the empty pixels. A look-up table speeds up the rectification but requires a lot of memory for larger pictures.

For a small part of the image the distortion is relatively small, therefore we do not need to rectify the image before extracting the features, instead the rectification is performed on each pixel qualified as a feature.

3.3. Experimental setup

We performed two different experiments. The first experiment is a controlled environment with artificial landmarks consisting of five LEDs mounted on a board in the shape of an X. In this experiment the vision system was moved horizontally along the x-axis with respect to the cameras reference frame in 10 mm steps so that $\overline{Pos}_n = \overline{Pos}_0 + [10n, 0, 0]^T$ where n is 1 to 36. At each location a stereo pair of images was taken and corner features were extracted using Harris and Stephens combined corner and edge detector implemented in the FPGA. In figure 6 three images are displayed where the left one shows the camera view in the first position with features plotted, the middle one shows the same view in the center position and the right one in the final position. All artificial landmarks were visible in all locations in both cameras.

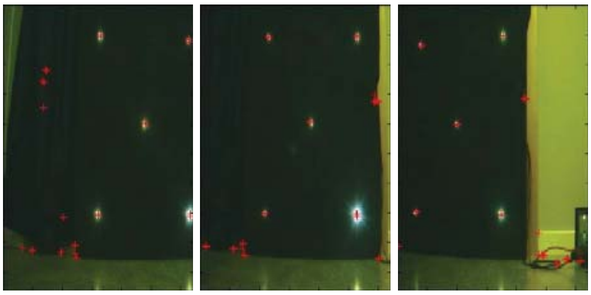

Figure 6. Views from the first experiment from three different locations with extracted corners.

In the second experiment we put the system in a vertical position facing the ceiling, images from this view is presented in figure 2. In this experiment the distance between two subsequent positions was 100 mm so that $\overline{Pos}_n = \overline{Pos}_0 + [0, 100n, 0]^T$, where n is 1 to 7.

4. Results

All measurements presented here is measured with respect to the camera reference frame, thus measuring the horizontal position in one axis and the vertical position in one axis in the first experiment. In the second experiment the horizontal positions in two dimensions are measured.

4.1. Stereo matching

Due to the relative orientation and translation of the cameras the horizontal and vertical position of the features in the images do not correspond. Therefore we allow stereo matching of features that have a difference in vertical position on the image sensor of up to 12 pixels. In figure 7 extracted features from left and right camera and their stereo matched correspondence are plotted, not all matches are correct as expected.

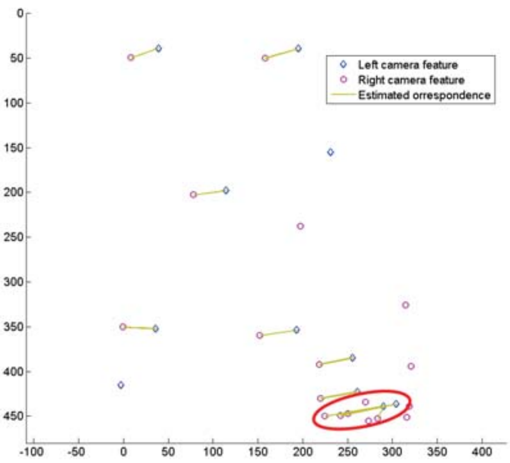

Figure 7. Stereo matched features, not all are correctly matched (spurious matches).

In these experiments we aim only at finding the egomotion of a robot, no considerations to obstacles are taken. Therefore we are strictly limiting the disparity in the stereo matching to be no greater than 120 pixels, only allowing objects further than 396 mm away to be detected. By restricting the maximum disparity the amount of spurious matches is decreased significantly.

4.2. Landmark location

In figure 8 we show all landmarks, from the first experiment. Landmarks are plotted with respect to the position of each measurement. Landmarks that appears from correctly stereo matched features appear in groups, where as the spurious matches appears at more isolated locations.

The resulting landmarks from the second experiment is presented in figure 9. Spurious landmarks appear even more isolated in this experiment.

4.3. Egomotion estimation

In the first experiment the vision system was moved by hand on a horizontal line along the x-axis and its position measured with respect to its start position. The position and orientation precision was similar at each location. The egomotion at each location n was calculated from the landmark set at $\overline{Pos_0}$ and $\overline{Pos_n}$. As can be seen in figure 10 the position is underestimated.

The estimation error of the vision system location with respect to the origin, in the first experiment, is presented in figure 11. The maximum error in position was 10.2 mm and orientation 0.65 degrees. All five landmarks were visible at each location in both cameras, but not necessarily stereo matched.

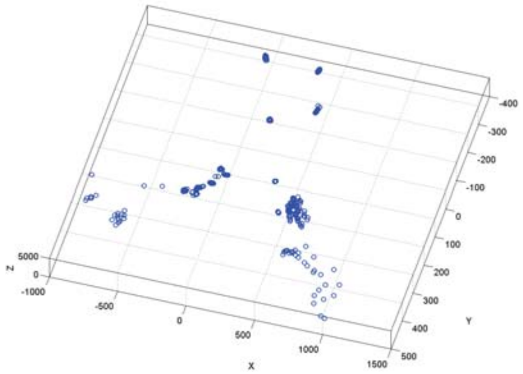

Figure 8. 3D landmarks from the first experiment calculated from the spurious set, translated the estimated distance.

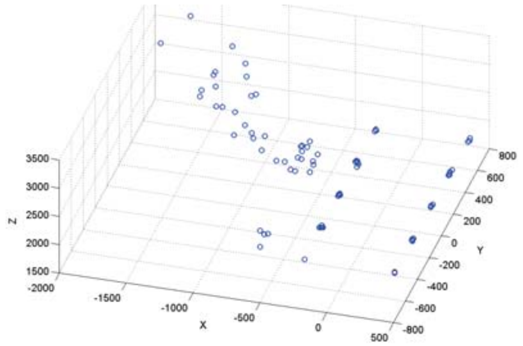

Figure 9. 3D landmarks from the second experiment calculated from the spurious set, translated the estimated distance.

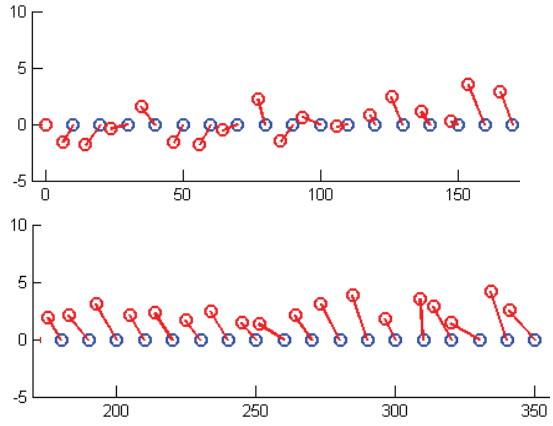

Figure 10. Egomotion estimation results from the first experiment with the orientation indicated.

The average delta motion was 9.7 mm in x and 0.04 mm in y against the expected 10 mm and 0 mm respectively. As seen in figure 12 the distribution around the average delta motion in both axis is quite even.

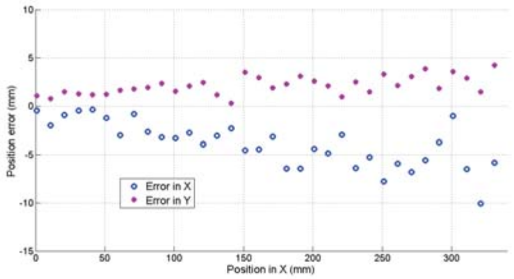

Figure 11. Position estimate error in X and Y, at each position in X, from the first experiment.

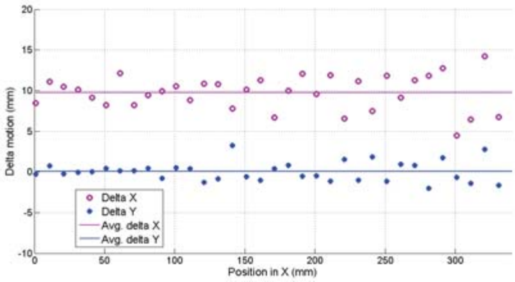

Figure 12. Delta motion between each location in X, from the first experiment.

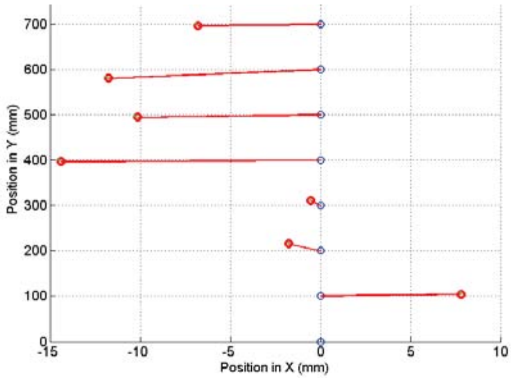

Figure 13. Egomotion estimation results from the second experiment using landmarks in the ceiling.

In the second experiment the vision system was tilted facing the ceiling, the distance to the landmarks was approximately 2200 mm, thus we expected less accuracy localizing the position of the landmarks, compared to the first experiment.

For the second experiment the egomotion estimation was slightly worse, the maximum error in position was 22.6 mm and orientation 2.41 degrees. Accuracy was, as expected, lower compared to the first experiment see figure 14.

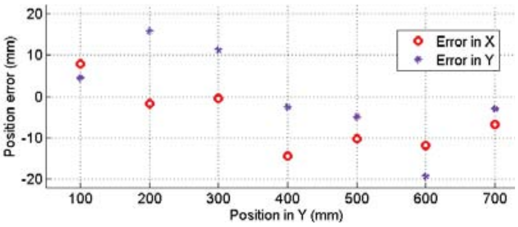

Figure 14. Position estimate error in X and Y, at each position in X, from the second experiment.

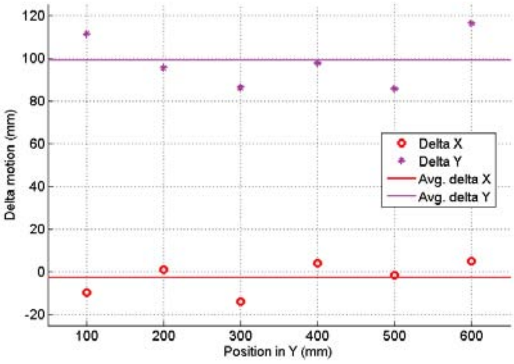

Figure 15. Delta motion between each location in Y, from the second experiment.

The average delta motion in the second experiment was 98.7 mm in y and -2.4 mm in x and the expected motion 100 mm and 0 mm respectively, see figure 15.

5. Conclusions

In the presented experiments we have shown that the system finds most of the correct stereo matches. The spurious matching step manages to find the true correspondence of landmarks from one position to another given that the delta motion is short enough and the landmark density is not to high. The underestimation of the delta motion noticed in the experiments is most likely caused by imperfection in the stereo calibration, this has to be further investigated. We have in our experiments not inferred any intelligence with respect to how landmark matching is performed. A few incorrect landmark matches does not affect the egomotion estimation that much. We have also experienced that if the features are too cluttered, the amount of landmarks grows rapidly, affecting the landmark matching so that almost any motion, within some range, can be supported by the spurious landmark set.

6. Future work

We intend to develop an automated on line calibration method for both intrinsic and extrinsic parameters that runs in parallel with the navigation, continuously improving the parameters. Furthermore we intend to implement a filter like EKF or similar to predict the location of a landmark given a measured delta motion and implement the back projection described in this paper to simplify the spurious matching process. The ultimate goal is to implement full navigation on an FPGA for on line calculations and to measure the performance.

Acknowledgments

This project is supported by Robotdalen. The authors would also like to acknowledge: *Xilinx*, for their kind donation of our FPGAs and design software tools. *Hectronic*, for the design and manufacturing of our FPGA boards.

References

[1] M. A. García and A. Solanas. 3d simultaneous localization and modeling from stereo vision. In *ICRA*, pages 847–853. ICRA, IEEE, 2004.

[2] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[3] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1106, 1997.

[4] J. Hong and M. Wang. High speed processing of biomedical images using programmable gpu. *Image Processing, 2004. ICIP '04. 2004 International Conference on*, 4:2455–2458, 24-27 Oct. 2004.

[5] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[6] J. Lidholm, F. Ekstrand, and L. Asplund. Two camera system for robot applications; navigation. In *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, pages 345–352, September 2008.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[8] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard. Cg: a system for programming graphics hardware in a c-like language. *ACM Trans. Graph.*, 22(3):896–907, 2003.

[9] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[10] H. Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, page 584, August 1977.

[11] C. Olson, L. Matthies, M. Schoppers, and M. Maimone. Stereo ego-motion improvements for robust rover navigation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1099–1104 vol.2, 2001.

[12] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1):80–113, 2007.

[13] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006.

[14] A. Ryberg. *Camera Modeling and Calibration for Machine Vision Applications*. Licentiate thesis, Department of Signals and Systems, Chalmers University of Technology, 2008.

[15] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, Vol. 37(2):151–172, June 2000.

[16] S. Takezawa, T. Ishimoto, and G. Dissanayake. Optimal control for simultaneous localisation and mapping problems in indoor environments with stereo vision. In *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, pages 4749–4754, Nov. 2006.

[17] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *Computer Vision, IEEE International Conference on*, 1:666, 1999.