2009:MR01

Machine vision for finding a joint to guide a welding robot

Mathias Larsson

THESIS WORK
Mechatronics Engineering, Robotics and Embedded system
Department of Engineering Science

BACHELOR'S THESIS

Machine vision for finding a joint to guide a welding robot

Summary

This report contains a description on how it is possible to guide a robot along an edge, by using a camera mounted on the robot. If stereo matching is used to calculate 3D-coordinates of an object or an edge, it requires two images from different known positions and orientations to calculate where it is.

In the image analysis in this project, the Canny edge filter has been used. The result from the filter is not useful directly, because it finds too many edges and it misses some pixels. The Canny edge result must be sorted and finally filled up before the final calculations can be started. This additional work with the image decreases unfortunately the accuracy in the calculations. The accuracy is estimated through comparison between measured coordinates of the edge using a coordinate measuring machine and the calculated coordinates. There is a deviation of up to three mm in the calculated edge.

The camera calibration has been described in earlier thesis so it is not mentioned in this report, although it is a prerequisite of this project.

Author:	Mathias Larsson		
Examiner:	Anna-Karin Christiansson		
Advisor:	Anders Rydberg, HV@PTC		
Programme:	Mechatronics Engineering, Robotics and Embedded Systems,180 Credits		
Subject:	Mechatronics	**Level**:	Basic level
Date:	July 7, 2009	**Report Number**:	2009:MR01
Keywords	edge detection, stereo vision, image analysis, canny,		
Publisher:	University West, Department of Engineering Science, S-461 86 Trollhättan, SWEDEN Phone: + 46 520 22 30 00 Fax: + 46 520 22 32 99 Web: www.hv.se		

Contents

1 Introduction

To increase the productivity and lower the cost in today's welding operations, the manufacturer introduces more robots to do the welding. The robots weld the parts at the same coordinates and welding parameters over and over with a high repeatability, but the robots have not the welders experience or possibility regarding when it is time to adjust some parameter in the welding process which is a very complex procedure or compensate for measurements variations in the work piece. The requirements in the weld seams are only increasing, so this is a problem that brings more research into vision sensors, which can detect errors in the weld seam during the welding, so the robot path and welding parameters can automatically be adjusted during welding [1].

The research of updating the robot path due to variations in the work pieces has reached a very high accuracy and it is now possible to detect edges in subpixel level as Chen et al., [1] proposed by using the Zernik moment. Gao et al., [2] proposed the same edge detection method, but they used structured light to amplify the joint between the work pieces.

All stereo vision methods are very similar. The differences are in how the images are captured. The most common is when the camera is mounted on the end effector and capture two images from different positions. Placing two cameras on a rig and they capture one image each is also a possibility and this decreases the time for capturing the images. The pseudo stereo method which Pachidas et al., [3] have studied makes it possible to calculate 3D-coordinates from one image without loosing any accuracy. With the help of mirrors the camera which is mounted on the end effector captures a complex image in a single shot and the result is a faster image processing.

The method which the images are captured in this project is similar to that method Chen et al., [1] used, one camera mounted on the robot captures two images in different positions. In the image processing the similarity stops with Chen et al., [1] then they uses the Zernik moment to detect edges in subpixel level. The edge detection method which is used in this project is the less advanced Canny method [4] and it only detects edges in pixel level.

1.1 Background

This project is a part of a project, where Volvo Aero is interested in decreasing their costs by welding small cast parts together instead of using large cast parts. To achieve this, the welding should be performed by a robot and the path must be determined before each weld, due to that the tolerances in the parts are bigger than the welding process allows. Otherwise the weld will not manage the high demands.

1.2 Purpose and aim

The purpose with this project is to see how accurately the 3D-coordinates of a joint between two metallic sheets could be calculated by using images from a camera mounted on the robot. The robot path will be determined by the result of the calculations and a welding will be simulated with a tool which is mounted on the end effector and directed on the backside of the metallic sheet with certain accuracy.

2 Prerequisites

2.1 Equipment description

Existing equipment at University West@Production Technology Centre shall be used.

- ABB robot IRB 2400 with control system S4
- Vision computer with Matlab and the image processing toolbox
- Calibrated vision camera with frame-grabber card mounted in vision computer
- Serial communication between vision computer and the robot control system
- Experimental geometry

2.2 Limitations

- The tests will be done on the experimental geometry, see *Figure 2.1,* where the metallic sheets are glued together
- The camera is mounted on the robot
- The robot path will not be updated with any orientation in the final simulated welding
- This report will not mention anything about camera calibration, camera parameters and the serial communication, since these are described in detail in earlier thesis works [5, 6].

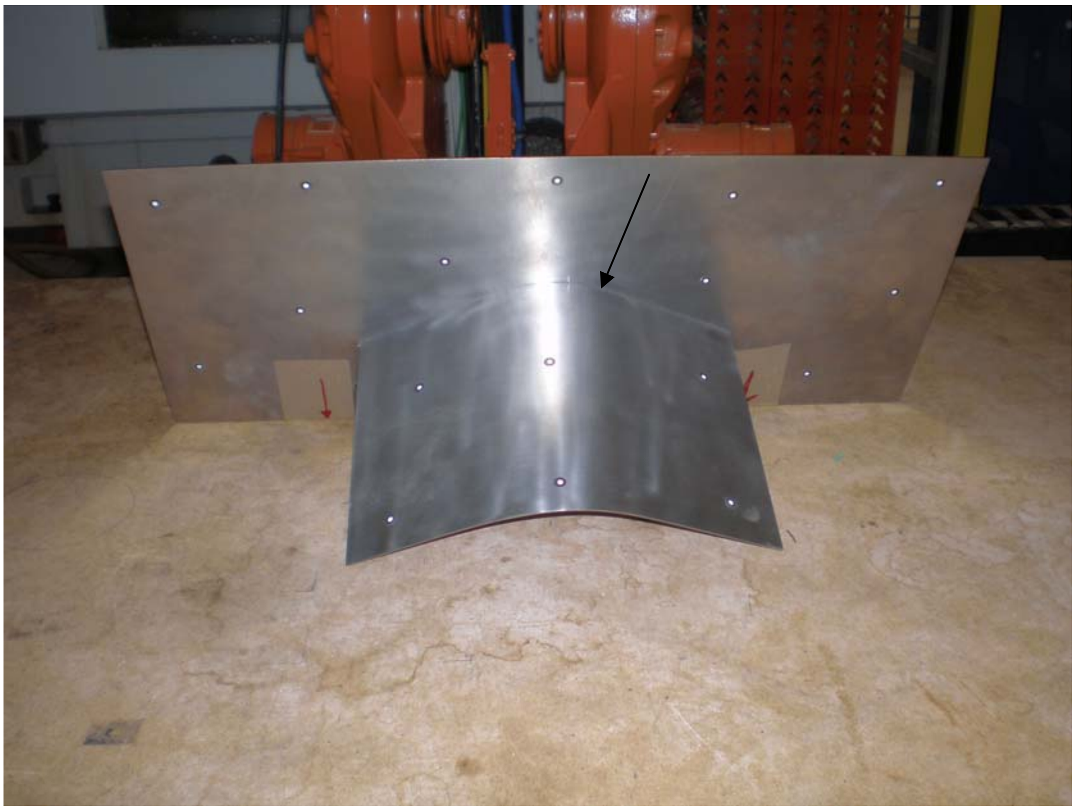

Figure 2.1 Experimental geometry with the joint marked with an arrow

3 Realisation

The approach in this test is to capture images of the experimental geometry from two different positions, with the camera that is mounted on the robot. The images will then be processed in Matlab so only the joint between the metallic sheets remains. When the image analysis is done, the stereo matching starts and the result of that is the 3D-coordinates of the joint and these will be sent to robot over the serial communication. The robot adds an offset to the coordinates of three mm in the tool's direction to avoid that the tool bumps into the experimental geometry before the simulated welding can start.

3.1 Method

- Acquire images of the experimental geometry is done from two different positions, see *Figure 3.1* and *Figure 3.2*.
- When the camera captures the images, save the robot pose (position and orientation) and send these together with the images to Matlab.
- Perform an image analysis of both images, to obtain the pixel coordinates in 2D of the joint in each image. The image analysis is done with a Canny filter [7] and a polyfit function [8].

- Calculate the camera pose when it captures the images by adding the transformation matrix between the tool and the camera to the robot pose. The transformation matrix is received from the camera calibration.
- Perform stereo matching of the 2D-images, to receive the 3D coordinates of the joint.
- Send the coordinates to the robot and offset those before the robot determines its path. The offset is done by defining and using a false tool which is three mm longer than the real tool.
- Run the robot along the determined path and study the result. To see how accurate the calculated coordinates are they will be compared with coordinates received from a measurement with a CMM (coordinate measuring machine).

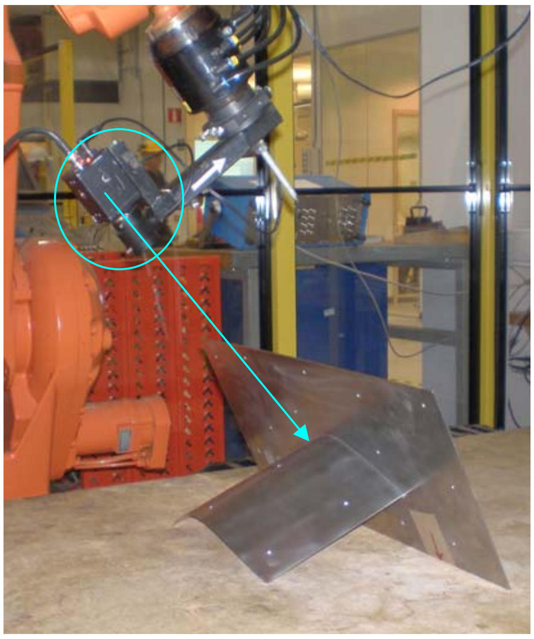
Figure 3.1 Camera position of image1

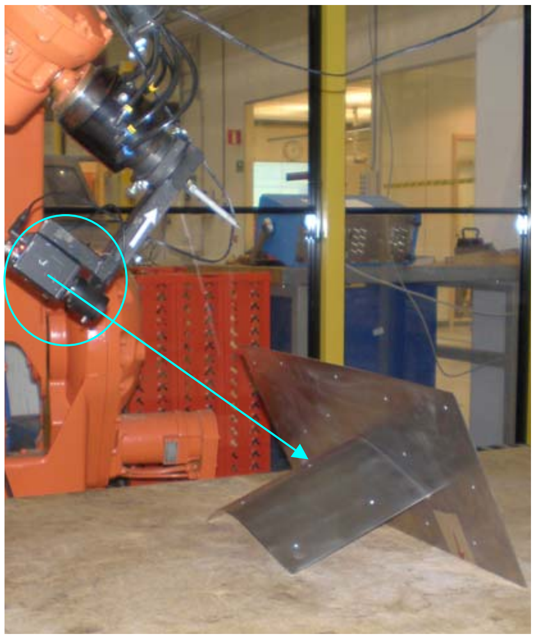
Figure 3.2 Camera position of image2

4 Image analysis

The 2D-images which will be analysed are acquired from two different camera positions and they will be saved in BMP format. Images in the format GIF, JPEG, PNG and TIFF can be analysed in Matlab with the same instructions as the BMP. To get a high accuracy in the stereo matching the angle between the positions should be as close to 90 degrees as possible. Unfortunately the camera position of image2 can not be lower than it is, see *Figure 3.2*, because the Canny filter does not register the joint if the position is any lower. All the steps in the image analysis of image1 [7], see *Figure 4.1*, will be explained here. For the image2 only the result will be shown, because the steps are almost identical with the steps in the image analysis of image1. The difference between the both images analyses are only the parameters in the Canny filter.

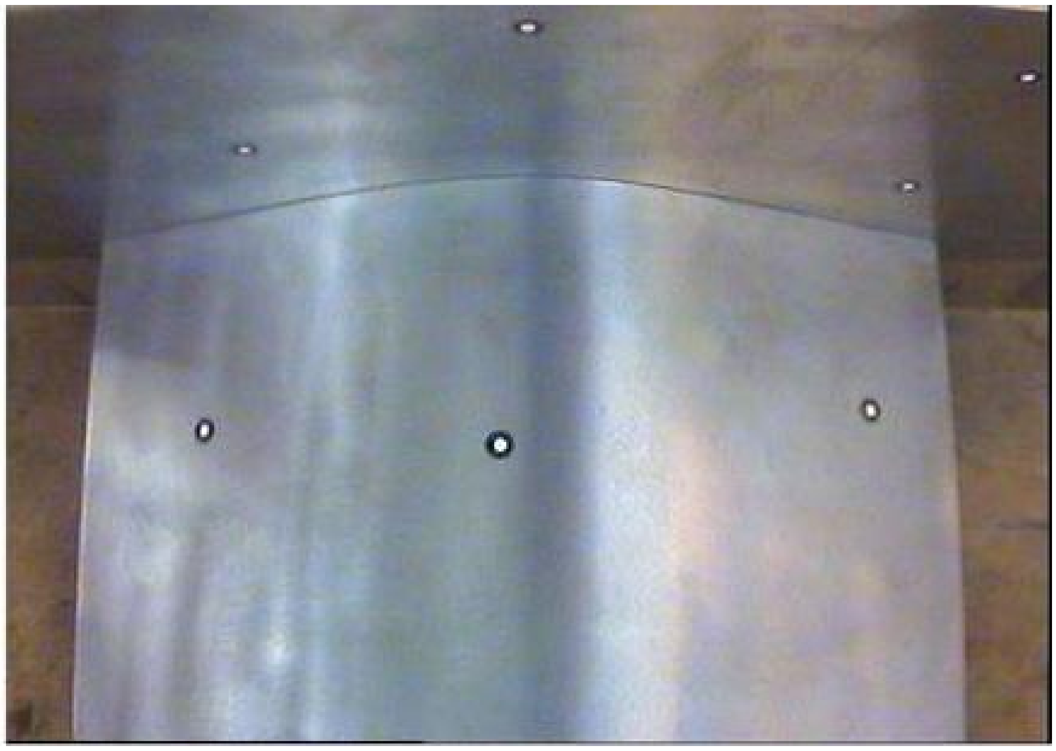

Figure 4.1 Image1 as acquired by the camera (the black and white spots are markers used by a 3D-scanning system outside the scope of this project)

4.1 Step 1: Greyscale

The first step in the image analyses is to read the image into Matlab and make a greyscale transformation of the image. The images are presented in the RGB colour system, this means that each pixel consists of three values, the intensity of red, green and blue. The values can range from 0 to 255 and if the intensity of red, green and blue is same, the pixel has a greyscale. The value which is used in every pixel to create the greyscale is the average of the red, green and blue in just that pixel.

Matlab commands [7, 8] are in the following shown with the Matlab prompt >>.

```
>>temp = imread('image1.bmp');
>>Greyscale = rgb2gray(temp);
```

4.2 Step 2: Edge detection

To be able to find the joint between the sheet metals an edge detection filter is used. This looks for differences in intensity between the pixels in the greyscale image. Matlab has several different edge detection functions in the image processing toolbox, like Sobel, Roberts and Canny filter [7, 9]. In this project the Canny filter is used, because it is the most capable.

The Canny filter has two adjustable parameters:

- The high threshold value, which together with the low threshold value controls how the difference should be between two pixels to be detected as an edge. The low threshold value is automatically calculated to 0.4 times the high threshold value.
- The sigma value, which decides the size of the window the filter uses when it tries to detect edges.

To understand the meaning of the parameters, see *Figures 4.2-4.5*, where the same image is shown with different settings in the Canny filter.

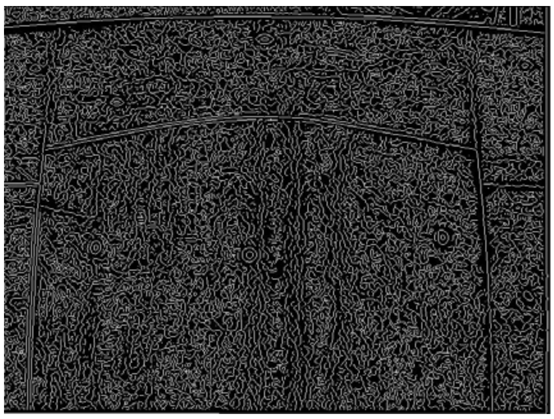
Figure 4.2 Low threshold and standard sigma

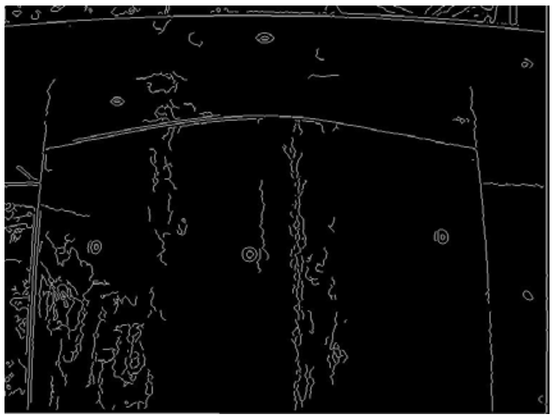
Figure 4.3 High threshold and standard sigma

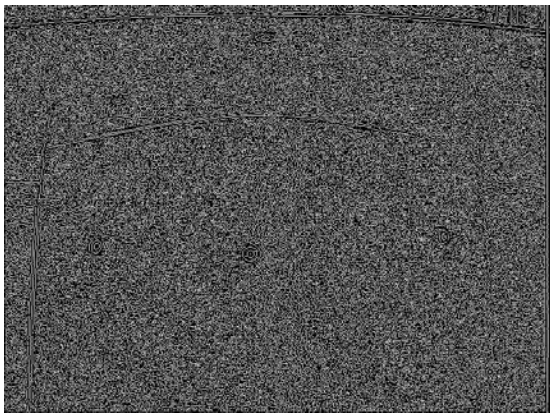
Figure 4.4 Low sigma and low threshold

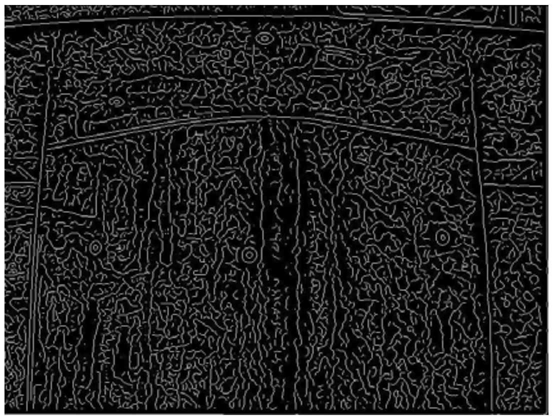
Figure 4.5 High sigma and low threshold

To receive a good result from the Canny filter, a low threshold value and almost a standard sigma are used. The result from the Canny edge filter, see *Figure 4.6*, shows that the edge is found but unfortunately it also has found a lot of useless information due to that the threshold values are set very low. If the threshold values had been higher it would easily missed to detect some part of the joint.

```
>>Canny = edge(Greyscale,'canny',0.01,1.2);
```

In the enlargement of the left corner of the joint, see *Figure 4.7* the pixels are very clearly seen.

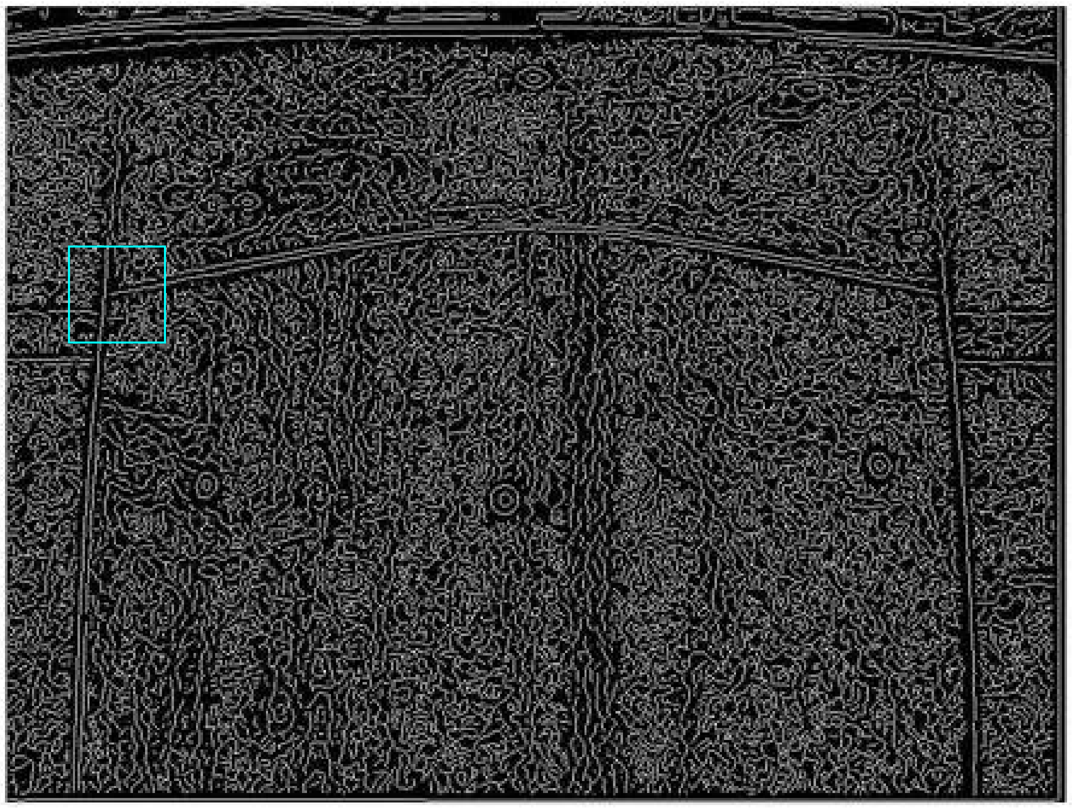

Figure 4.6 Image1 after the Canny edge filter

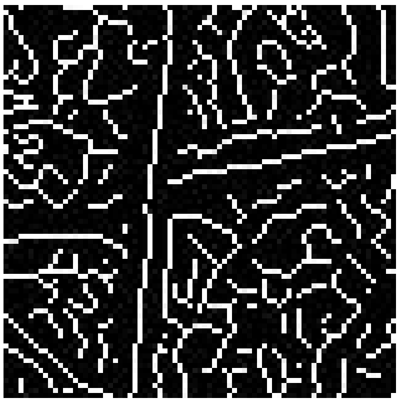

Figure 4.7 Enlargement of the left corner of the joint

4.3 Step 3: Connected components

After the filter the pixels in the image consists of zeros and ones, if it is a zero in the pixel it interprets as black and an one interprets as white. If two white pixels are beside each other, the pixels displays the same object and to be able to control all the found objects it is suitable to give all pixels which display the same object (in this case a change in intensity) a unique number, see *Figure 4.8*. All pixels with the same number can now be treated as an object and now it is possible to decide the size and position of the found edges.

```
>>Components = bwlabel(Canny,8);
```

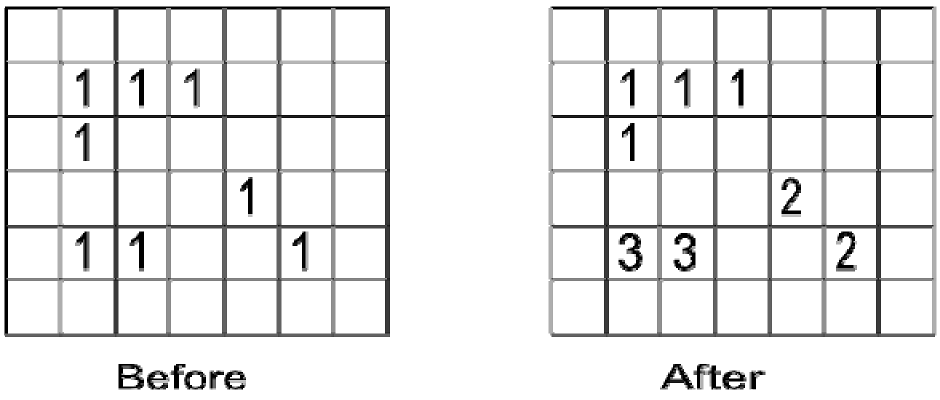

Figure 4.8 An image before and after analysis of connected components. Each square represents a pixel

4.4 Step 4: Removal of unnecessary information

The edge between the sheet metals consists of over one hundred pixels and it doesn't contain any vertical parts which are more than two pixels long, see *Figure 4.6*, so all small objects and all objects which contain long vertical parts will be deleted. Finally there are some horizontal objects in the top of the image and these are very similar to the edge, see *Figure 4.6*. One way to remove these are to create a corridor along the topside and delete all objects which are partly inside this area. After that only the edge remains, see *Figure 4.9*.

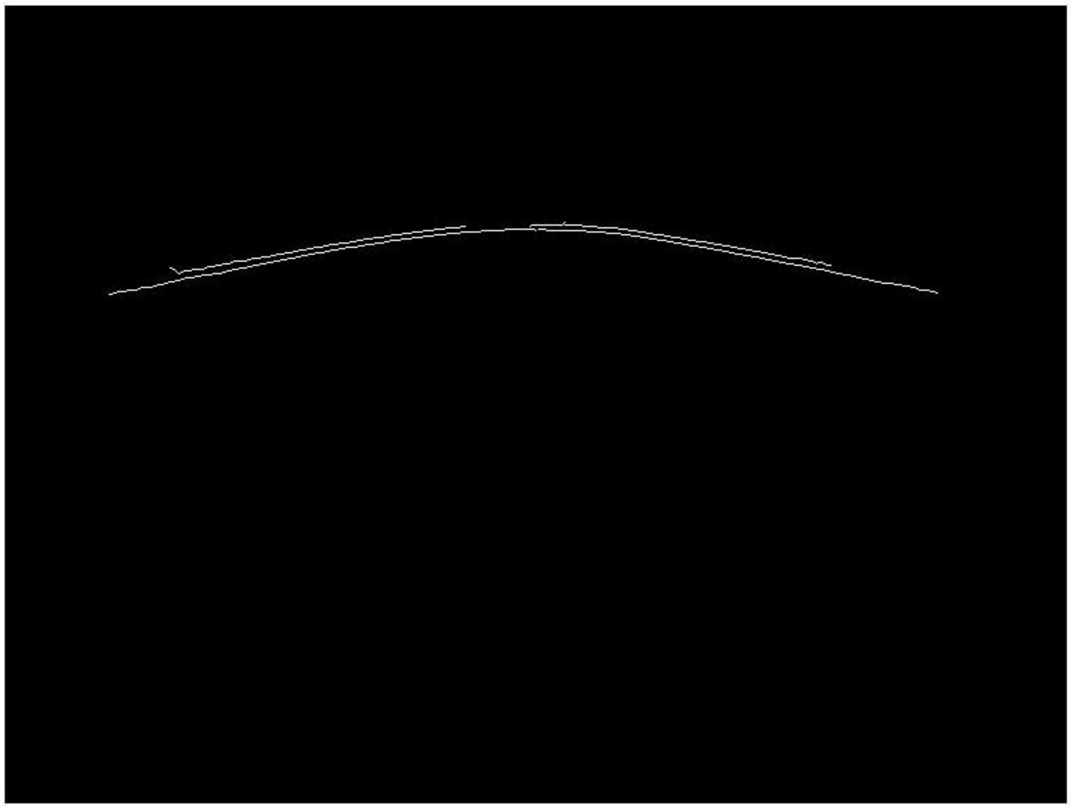

Figure 4.9 Image1 after removal of unnecessary information

4.5 Step 5: Polyfit

The Canny edge filter has found pixels along two lines, which belong to the edge, see *Figure 4.9*. This depends on that the filter reacts on both sides of the edge. To get some kind of average of the remaining pixels and to fill up the curve if it is not complete, a polyfit [8] function is used on the remaining pixels. The polyfit function creates a smooth mathematical curve and the result of the polyfit can be influenced by setting a higher degree to the mathematical curve, which gives it a better accuracy.

```
>>xCurve=xmin:1:xmax;                    %decides where the curve will start and stop
>>coefficients = polyfit(x,y,7);         %decides which degree the curve shall have
% when it tries to find an average along all pixels
>>yCurve = polyval(coefficients,xCurve);  %calculate y values to all x values
```

4.6 Result from image analysis

To see how good the polyfit created curve is relative to the original images, they will be displayed in the same figure, see *Figure 4.10* and *Figure 4.11*, and this comparison shows that the result of image1 is very good, but for image2 the polyfit curve does not match completely.

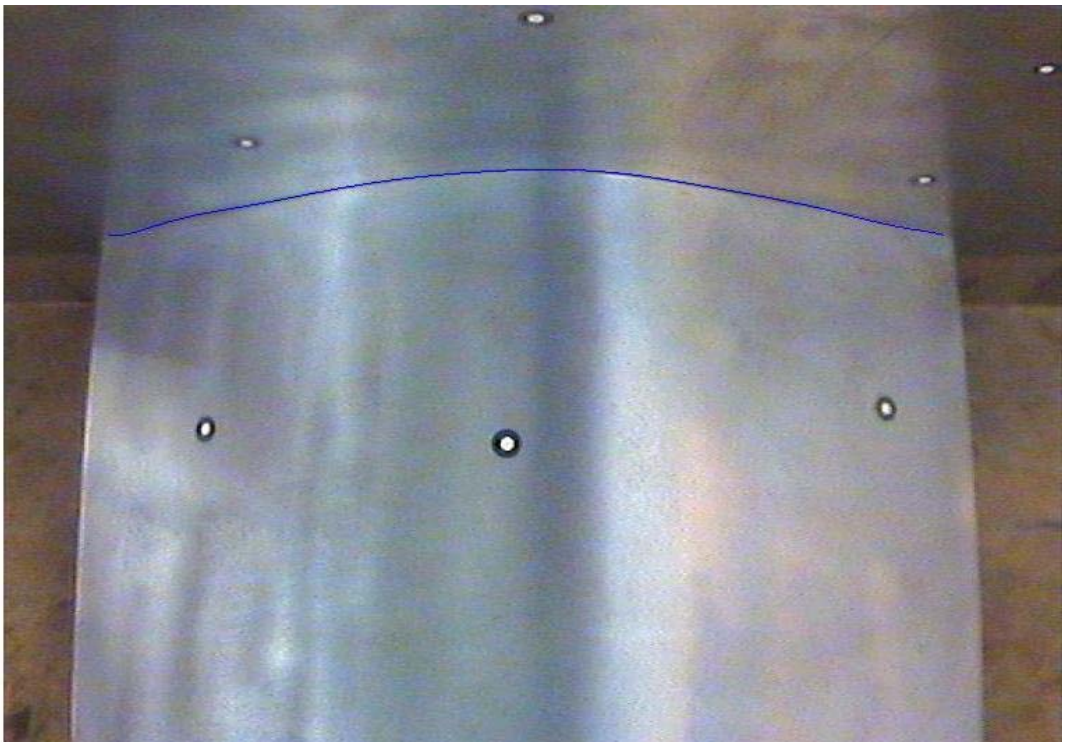

Figure 4.10 Polyfit created curve compared with original image 1

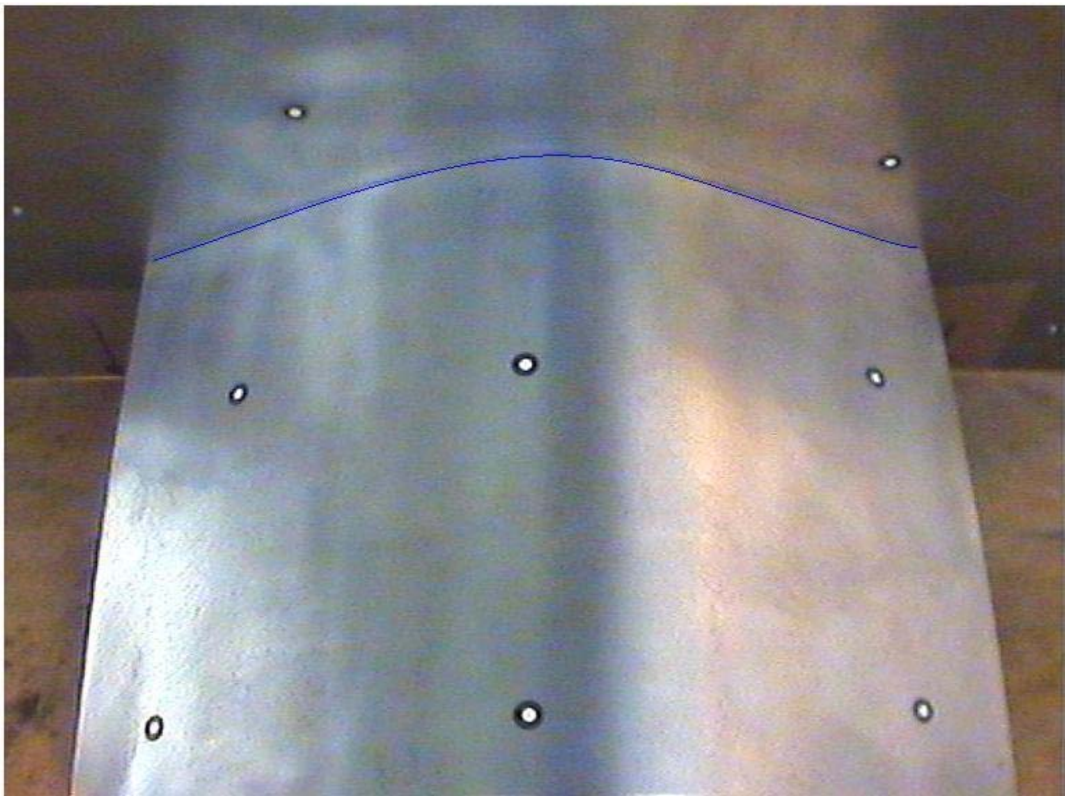

Figure 4.11 Polyfit created curve compared with original image 2

5 Stereo matching

Performing stereo matching of the two images which has been acquired with the camera is done by creating lines from each pixel which are remaining after the image analysis. The line from a pixel in image1 is expected to intersect with a line from a pixel in image2 because they represent the same point in 3D. To be able to calculate where two lines intersect in space, it requires one known point along each line and the direction vector of both lines. The known point in all lines from each image will be the position of the origin in the cameras coordinate system when the camera captures that image. The direction vectors for each line are calculated with parameters received from the camera calibration.

5.1 Camera points

The camera is mounted on the camera tool, see *Figure 5.1,* and the pose the robot is tracking is the origin in the camera tool coordinate system. The origin in the camera tool coordinate system is placed at the tip. To calculate where the origin of the camera coordinate system is in space when the camera captures the images, it is necessary to know the transformation matrix between the camera tool coordinate system and the camera coordinate system. The transformation matrix is received from the camera calibration [5] and it must be updated every time the camera tool has been removed from the robot or it has bumped into something.

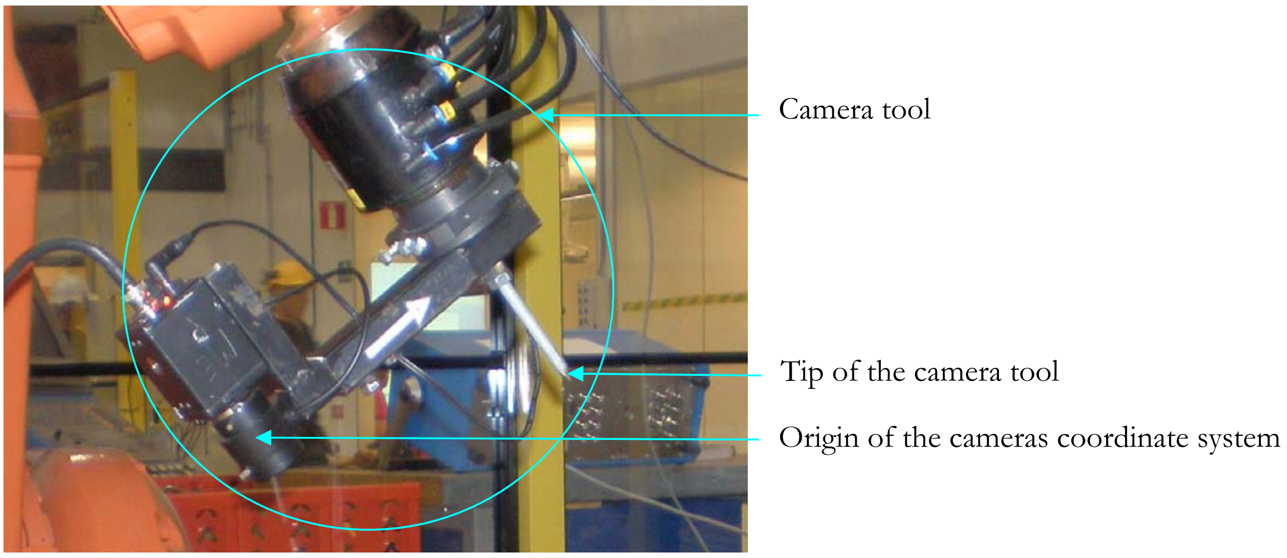

Figure 5.1 The camera tool

5.2 Direction vector

The mathematical model of a camera [10], see *Figure 5.2,* represents the camera coordinate system relative to the image. In the image there is also placed two coordinate systems, which are marked with prim and bis. The x and y axes of the camera coordinate system are placed parallel to the image and the z axis, also called the optical axis, is going through the image in the principal point. The prim coordinate system is placed in the image with its origin in the right upper corner and it is scaled in pixels. The bis coordinate system is also in the image, but its origin is placed at the optical axis and it is rotated like the camera coordinate system and it is also scaled in pixels. The distance between the origin of the camera coordinate system and the principal point is called the focal distance. The focal distance and the coordinates of the principal point relative to the prim coordinate system are three camera parameters which are received from the camera calibration [5].

After the image analysis is performed, the coordinates of all pixels in the prim coordinate system are known and those together with the coordinates of the optical axis give the coordinates of all pixels in the bis coordinate system.

The coordinates of a pixel in the bis coordinate system together with the focal distance, which is given in pixel are the direction vector from that pixel.

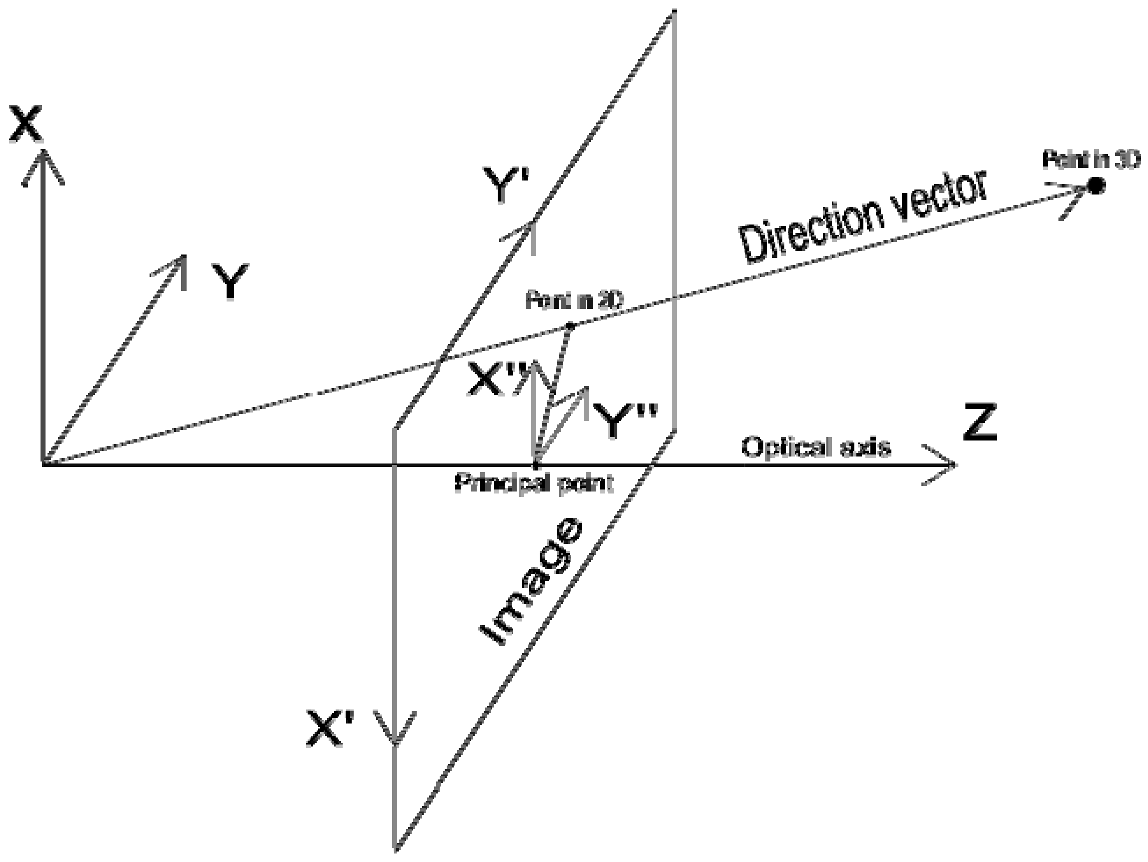

Figure 5.2 Mathematical model of a camera

5.3 3D coordinates

Now it is possible to create lines along the direction vectors from each pixel in both images which also cross the origin of the camera coordinate system. In theory every line from image1 would intersect with a line from image2 as mentioned before, but due to disturbances and resolutions they will normally not, see *Figure 5.3*. So instead of searching for intersections point, the lines from image1 have to search for the shortest distance to a line from image2.

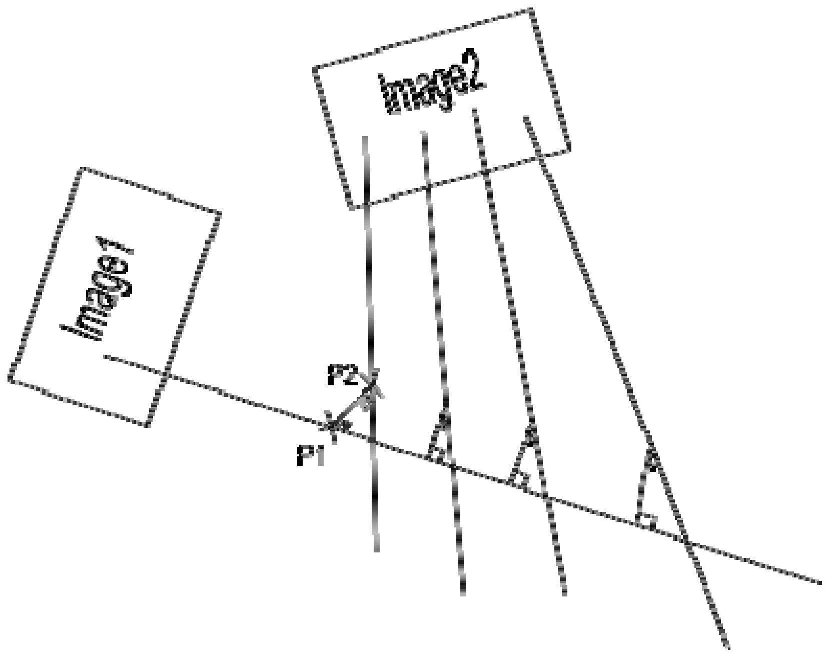

Figure 5.3 Line from image1 searching for line from image2 with shortest distance

Let Cpose$_1$ and Cpose$_2$ be the camera pose for each image, V$_1$ and V$_2$ be the direction vectors for different lines, S$_1$ and S$_2$ be the scaling parameters of these lines and P$_1$ and P$_2$ be the X, Y and Z coordinates along these lines. It gives the line equation (5.1) from the first pixel in image1 and the line equation (5.2) from the first pixel in image2.

$$P_1 = Cpose_1 + S_1 V_1 \tag{5.1}$$

$$P_2 = Cpose_2 + S_2 V_2 \tag{5.2}$$

Putting the line equation (5.1 and 5.2) to equal gives the equation (5.3).

$$Cpose_1 + S_1 V_1 = Cpose_2 + S_2 V_2 \tag{5.3}$$

The equation (5.3) written in matrix form (5.4)

$$\begin{bmatrix} V_{1X} & -V_{2X} \\ V_{1Y} & -V_{2Y} \\ V_{1Z} & -V_{2Z} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} Cpose_{2X} - Cpose_{1X} \\ Cpose_{2Y} - Cpose_{1Y} \\ Cpose_{2Z} - Cpose_{1Z} \end{bmatrix} \tag{5.4}$$

The equation (5.4) has no solution when the lines do not intersect, so when Matlab solves the system of equation it is searching for the values on S$_1$ and S$_2$ which have the closest solution. The line from the first pixel in image1 does this with all lines from image2 and when it has found the line with the shortest distance to (closest solution), the P$_1$ and P$_2$ are calculated. The 3D-coordinates from the stereo match of the first pixel in image1 are the average of P$_1$ and P$_2$. When all lines from image1 have done this, the stereo match is completed.

6 Result

The 3D-coordinates received from the stereo match are sent to the robot, so the robot can determined its simulated welding path. The robot follows the joint from the other side of the metallic plate with quite god accuracy. To see how good the accuracy is, the sheets of metallic is measured with a coordinate measuring machine so it is possible to create a 2D curve of the joint between the sheet metals. The 3D-coordinates are then projected into the same coordinate system, see *Figure 6.1*, and this comparison shows that it is a max deviation on 3 mm on the right side between these curves. In the middle the curves are identical and on the left side there are small deviations.

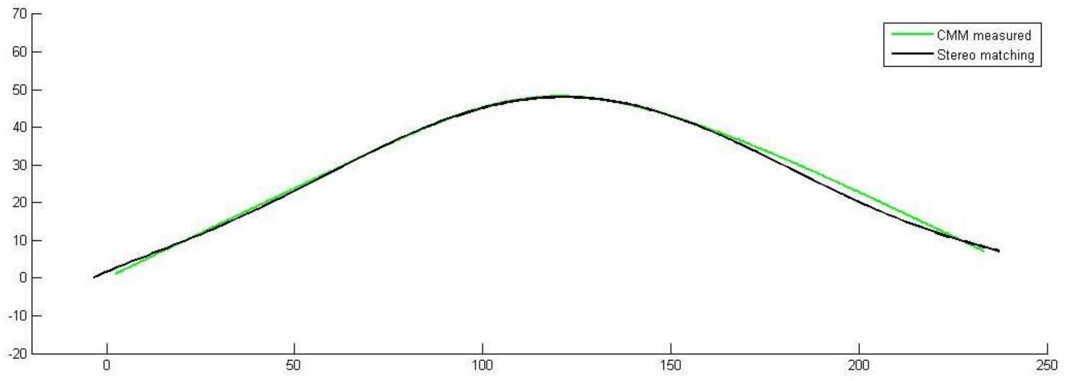

Figure 6.1comparison of coordinates from CMM measuring and stereo matching

7 Conclusion

7.1 Analysis of result

The fact that the deviation varies along the curve has its cause in the polyfit function in the image processing, the polyfit function tries to create a mathematical curve of all remaining pixels. In the image processing result it shows very clearly in image2, see *Figure 4.11*, that the curve wants to turn up at the ends of the joint, mostly in the right side, this tendency exists even in image1, see *Figure 4.10*.

After removal of unnecessary information, see *Figure 4.9*, the joint is marked of two parallels lines. This depends on that the canny edge filter reacts on both sides of the joint and the real coordinates are in the middle. When the camera position is lower as in image2 the filter only finds one line, which also decreases the accuracy. The second camera position should be higher up to increase the accuracy in the canny edge filter, but this doesn't the stereo matching allow, so the second camera position is a compromise between the accuracy in the mathematical calculations in stereo matching and the edge detection function.

7.2 Recommendations to continued work

The repeatability in the canny edge filter output has been a small problem during this project. This depends on that the lighting conditions have varied during the spring. Putting a tarpaulin over the robot cell has decreased the variation in the result of image analysis, but not enough, so some kind of controlled illumination is necessary. The problem with lightning of this experimental object is that it gives huge reflections in the image, which lead to disturbances in the image analysis. At tests with ordinary lamps to illuminate the experimental geometry, the problem with reflections did it impossible to continue with that kind of illumination.

To increase the accuracy it is important to get a better result from the image analysis, the polyfit function works fine on the top where the curve is bent more, but not on the sides there the curve is mainly straight. Perhaps it possible to use some other method to create a curve of the remaining pixels after the image analysis or split the curve in sections, which later can be put together to a complete curve.

In this project only the 3D coordinates have been updated, in welding process the angle between welding tool and the metal should be perpendicular. It would be interesting to include the orientation in the update.

The comparison, see *Figure 6.1*, only compares the 2D-coordinates from the stereo matching, a comparison could also include the third coordinates

References

1. S. B. Chen, X. Z. Chen & T. QIU, "Acquisition of Weld Seam Dimensional Position Information for Arc Welding Robot Based on Vision Computing," *Journal of Intelligent and Robotic System*, vol. 43, pp 77-97, 1986.
2. Shiyi Gao, et al., " Dual-Beam Structured Light Vision System for 3D Coordinates Measurment," *Intelligent Control and Automation, 7th World Congress*, pp. 3687-3691, 2008.
3. Theodore P. Pachidis, John N. Lygouras, "Vision-Based Path Generation Method for a Robot-Based Arc Welding System," *Journal of Intelligent and Robotic System*, vol. 48, pp 307-331, 2007.
4. J. Canny, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence*, vol. PAM-8, pp. 679-698, 1986.
5. Nilsson, Jim (2008). *Acquiring and Correction Weld Paths by Means of Machine Vision.* Unreleased manuscript. Trollhättan: University West.
6. Stark, Per (2007). *Machine vision camera calibration and robot communication.* [Electronic]. Trolhättan: University West. Available:<http://urn.kb.se/resolve?urn=urn:nbn:se:hv:diva-1351> [2009-05-19].
7. The MathWorks (2004). *Image Processing Toolbox*. ver.5
8. Pärt-Enander, Eva & Sjöberg, Anders (2003). *Användarhandledning för MATLAB.* Stockholm: Elmanders Gotab.
9. Parker, James R., *Algorithms for Image Processing and Computer Vision,* New York, John Wiley & Sons, Inc., 1997, pp. 23-29.
10. Hagnelius, Anders (2005). *Visual Odometry.* [Electronic]. University of Umeå. Available:http://umu.se/education/examina/Rapporter/AndersHagnelius.pdf> [2009-06-23].