Relative Measurement for Kinematic Calibration Using Digital Image Processing

A.A. Fratpietro, M.J.D. Hayes
Department of Mechanical & Aerospace Engineering, Carleton University,
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada,
afratpie@mae.carleton.ca
jhayes@mae.carleton.ca

An algorithm is developed for the purpose of extracting metric information from a Thermo CRS A465 manipulator using a camera-based data acquisition system and a measurement artifact. The algorithm uses common image processing techniques to extract the location of a point in the X-Y plane of the robot workspace. This measurement point will be used in the future to calibrate the Thermo CRS manipulator. The process used to calibrate the algorithm and a validation of its resolution are also presented. A characterization of the manipulator repeatability using this system is performed and the results presented in this paper.

1 Introduction

The kinematic calibration[1] of a serial manipulator with six revolute axes generally requires a system for measuring the error in the position and orientation (pose) of the robot end-effector (EE). This error refers to the difference between the pose of the robot EE as determined by some external measurement system and the pose of the robot EE as determined by the robot joint encoders. By implementing a kinematic model based on the Denavit-Hartenberg parameters, robot kinematic calibration can be achieved requiring only the external measurement of relative positional error in the robot EE. The measurement of this positional error in three directions is obtained through the implementation of a CCD camera and a laser distance sensor mounted to the robot EE. The distance sensor is oriented parallel to the camera optical axis.

The CCD camera captures images of a precision ruled surface that must be digitally processed in order to extract accurate metric information. The acquisition of these measurements requires that the precise location, orientation and representative equations of all objects in the image be determined in the image pixel coordinate system. The scale of an image is determined by estimating the pixel distance between adjacent lines in any single image. A comparison of images taken before and after a linear robot motion parallel to the ruled surface produces the relative positional error resulting from that motion. It is these positional errors that are used to calibrate the kinematic model of the robot.

In this paper we develop the algorithm for extracting accurate metric information from images of a measurement artifact. This algorithm will be applicable to the previously outlined calibration system. There are a multitude of techniques[2] in the realm of image processing that might be used for this purpose. The core image processing algorithm that we employ consists of noise compensation, thresholding, edge-detection, edge-scanning and linear regression techniques. A comparison is presented between the measurement accuracy achieved through the use of several different methods of noise compensation and several different parameter values within the edge-scanning algorithm. The results of this comparison are used to determine the best algorithm for this calibration system. A characterization of the repeatability of the Thermo CRS A465 manipulator is also presented.

2 Equipment

The Thermo CRS A465 robotic manipulator is a robot arm consisting of 6 serially connected revolute joints and a total arm span of $710mm$. This robot is rated as having a repeatability of $\pm 0.05mm$. The data presented in this paper is produced using the A465.

The measurement head consists of a camera, lens, mounting structure and appropriate power/data cabling. The combined mass of these devices does not

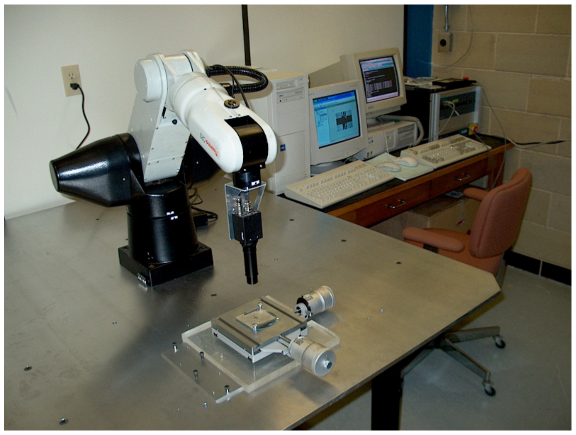

Figure 1: A465 robot and data acquisition system

exceed the $2.0kg$ payload of the robot.

The camera is a Pulnix TM-200. This camera has horizontal and vertical resolutions of 768 and 494 pixels, respectively, and a minimum signal to noise ratio of $50dB$. Mounted to the camera through a C-mount interface is a Rodenstock Macro 1X lens. This lens has a field of vision of $6.4 \times 4.8mm$ at a working distance of $68mm$.

The camera and lens are mounted to the robot using a stainless steel measurement head designed and manufactured at Carleton University. The complete camera/lens/head system integrated with the robot end effector can be viewed in Figure 1. Images produced by the camera are captured using the National Instruments PCI-1409 Image Acquisition Card integrated into a personal computer. The program IMAQ is implemented for the simple capture and saving of images.

All measurements are performed through the extraction, processing and comparison of images depicting a measurement artifact. The artifact in question is a thin block of aluminum with a pattern precision-machined onto its surface. The pattern on the artifact consists of a set of intersecting perpendicular grooves of depth $2mm$ and a line width of $0.794mm$ ($\frac{1}{16}$"). The accuracy on these dimensions has an upper limit of $+0.000$" and a lower limit of -0.002". The lengths of the grooves are approximately $2cm$ each, much larger than the field of view of the camera and lens. The grooves are partially filled with black enamel paint to provide sharp contrast between the grooves and surface of the artifact. All processing of images and data is performed in the MATLAB environment using custom developed high-level image processing software and pre-packaged MATLAB library functions.

The calibration and validation of results is performed using the components described above, excluding the A465 robotic manipulator, and including the Mitutoyo Vernier X-Y Table $0-2$", with a resolution of 0.0002".

3 Image Processing

The A465 robotic manipulator is programmed to manipulate a measurement head into a pose such that an image of a measurement artifact can be captured. A typical pose can be viewed in Figure 1. This image indicates the initial position that the robot will attempt to return to throughout the repeatability test. After the image is captured, the robot moves through a single motion or series of motions and then attempts to return to the initial position. Another image of the artifact is captured. This new image indicates the projection in the plane of the camera of the positional error that the manipulator has produced while attempting to return to the original pose. This process is repeated multiple times to produce sets of images. These raw images captured by the data acquisition system contain the information that is used to characterize the repeatability of the robot. Digital image processing is required to extract the metric information from these sets of images.

3.1 Image Characteristics

The selection of an image processing algorithm is based on the type of application in which it will be used. In this paper, the algorithm is required to extract precise metric information from an image captured by a data acquisition system. Figure 2 shows a typical image captured by the data acquisition system in question. This image clearly displays the measurement artifact, including surface irregularities on the aluminum and reflected light from the black enamel paint located in the groove of the artifact. The position of the robot during image acquisition is indicated by the crossing of the exact centreline of the horizontal and vertical lines located on the artifact. The image processing algorithm is required to extract the location of this point from each image.

The images in question are 8-bit monochrome images. Each pixel contains a possible value in the range 0 to 255, where a value of 0 represents the shade black and 255 represents the shade white. The values between 0 and 255 are shades of gray.

The size of an image is $640(H) \times 480(V) pixels$. Since the field of view of the lens is $6.4(H) \times 4.8(V)mm$, each pixel represents a field of approximately $10(H) \times 10(V)um$. It can be noted that the pixel length is an order of magnitude smaller than the repeatability of the robot.

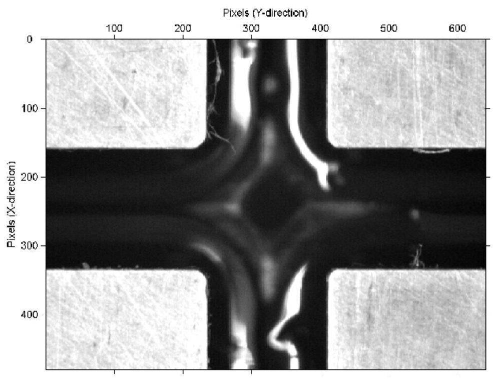

Figure 2: Raw image captured for processing

In order to determine the location of the artifact centrepoint, both lines can be described in terms of their two binding edges. The vertical line, when described from the left to the right side of the image, is bound by a light-to-dark edge and, subsequently, a dark-to-light edge. Since these two edges were machined to be completely linear, the first-order equations of each of these edges can be extracted and then combined to produce the equation of the line directly between them. This same procedure can then performed on the horizontal line resulting in the first order equation of its centreline. The solving of two first-order functions can result in only one solution and, in the case of the crossing lines in question, this solution is the point of intersection.

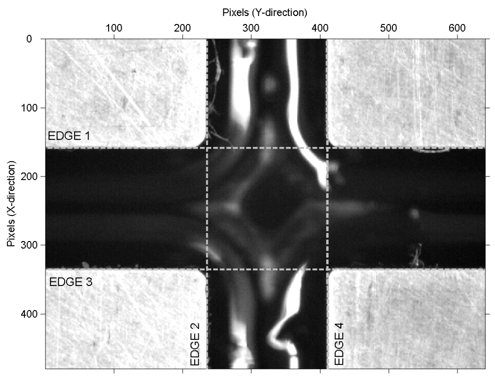

Figure 3: Image with four edges labelled

The processing of the images consists of extracting the first-order equations of the 4 edges located in each image. These four edges are labelled in Figure 3. The algorithm used to extract this information is described in the following section.

3.2 Algorithm Selection

The first step in the processing of an image is obtaining the image data. For the 8-bit monochrome images used in this paper, the data is represented by a 640×480 element matrix where each element in the matrix represents the intensity observed by the corresponding pixel in the camera.

3.2.1 Noise Filtering

The image data contains at least 2 types of noise of varying effects which are inherent in digital CCD cameras. Salt and pepper noise is the random inclusion of peaks and valleys across the image data. These peaks and valleys appear as a distribution of lighter and darker pixels which can skew calculations involving pixel intensities. Gaussian noise affects the intensities of pixels in a manor proportional to the Gaussian distribution. There are several different means of dealing with noisy data. This paper explores the results of applying two different convolution masks to the data.

A convolution mask is an $n \times m$ dimensional window that is centered on each element in the image data matrix. The elements in the mask can be weighted, as in the Gaussian filter, or not, as in the Mean filter. The elements in the mask are then combined with the corresponding elements in the image data matrix to produce a weighted or non-weighted sum of the image data which represents the value of the filtered data pixel at that location in the matrix.

The Mean filter implemented in this paper is a 3×3 matrix with all elements equal to a value of one.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{1}$$

This mask sets the value of each element in the data matrix equal to the mean value of its surrounding pixels. The benefit of implementing this filter is that local peaks and valleys in pixel intensity caused by noise will be reduced. The possible problem with implementing this filter is the smoothing of possible edges resulting in a reduction of contrast used to position edges.

The Gaussian filter implemented in this paper is a 7x7 Gaussian mask with elements chosen from the two-dimensional zero-mean Gaussian distribution,

$$g[i,j] = ce^{-\frac{i^2+j^2}{2\sigma^2}} \tag{2}$$

A typical convolution mask described by this distribution is used in the development of this algorithm.

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 3 & 2 & 2 & 1 \\ 2 & 4 & 8 & 16 & 8 & 4 & 2 \\ 2 & 2 & 4 & 8 & 4 & 2 & 2 \\ 1 & 2 & 2 & 4 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 & 2 & 1 & 1 \end{bmatrix} \tag{3}$$

The benefits of this filter are similar to those of the Mean filter, but the Gaussian filter is suited to filtering out noise with a Gaussian distribution. The partial processing of one image using both Mean and Gaussian filtering can be viewed in Figure 4.

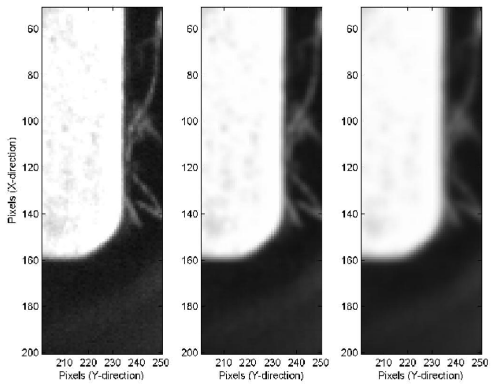

Figure 4: Comparison of raw image, mean-filtered image, and gauss-filtered image

3.2.2 Edge Detection

After filtering, the image contains data with most of the noise-related spikes in pixel intensity removed. The next step of the processing is the accentuation of possible edge pixels through the use of an edge detection operator. An edge detection operator is another type of convolution mask with its elements weighted in such a way that the weighted sum of the elements in the data matrix produce larger intensities in the proximity of sharp transitions from one pixel intensity to another. The elements of these convolution masks can be weighted in such a way as to accentuate pixels containing horizontal or vertical edges with transitions from high to low or low to high intensity. The edge detection mask implemented in this algorithm is known as the Prewitt operator,

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

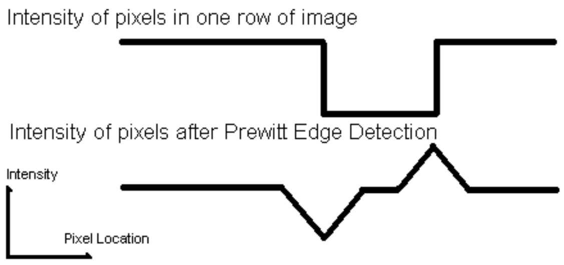
Intensity of pixels in one row of image
Intensity of pixels after Prewitt Edge Detection
Intensity
Pixel Location

Figure 5: Basic result of edge-detection

This operator is a 3×3 element convolution mask that uses pixels in the two columns adjacent to the current pixel under investigation and along the length of the edge being detected to determine whether or not an edge is present. Figure 5 shows a graphic description of the edge-detection by such an operator. As the Prewitt operator passes over pixels of equal intensity, the resulting pixels are nearly unchanged. When the operator passes over a sharp transition from high to low or low to high intensity, the calculated resulting pixel is either attenuated or accentuated depending on the bias of the operator being used. Four distinctly weighted Prewitt operators are required to distinguish between the two possible horizontal transitions and two possible vertical transitions. The mask shown in Equation 4 is used to reveal horizontal edges that transition from low-intensity pixels located above the edge to high-intensity pixels below the edge. The 3-dimensional inverted view of the filtered image is shown in Figure 6. This figure illustrates how the transition in the X-direction from pixels of low intensity to pixels of high intensity is accentuated using the Prewitt operator in Equation 4. The three other Prewitt operators will have a similar effect on the other 3 edges. In the case of this algorithm, each of the four edges is processed separately.

One possible benefit to using the Prewitt edge-detector is that this operator also acts as a partial noise filter by including a 3×3 pixel area in all calculations. Using the pixels located in adjacent columns will reduce the effect of noise in the calculation. This reduces the necessity of using a separate noise filter.

3.2.3 Applying a Threshold

After the edge-detection operator is applied, it is possible to eliminate many of the pixels in the image as possible edge locations by applying a threshold limit. The elements in the data matrix with larger values are more likely to belong to the edge being characterized. The elements with lower values are not located on the visible edge. The application of a threshold limit in-

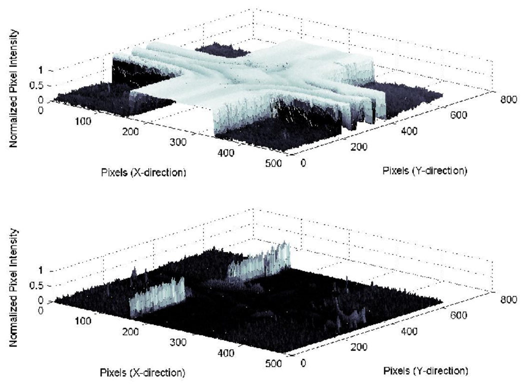

Figure 6: Results of Prewitt edge detection

sures that only the most likely pixels are considered as possible edge locations. After experimentation, it was determined that a threshold of approximately 30 percent of the maximum pixel intensity works to retain the information required for the line-scanning algorithm described in the following section.

3.2.4 Segmentation

The line-segmentation algorithm makes use of both the processed and unprocessed image data in determining the most likely location of the artifact edge currently being processed. The result of this segmentation is a slope/y-intercept form of the equation that best describes the edge. Several considerations must be taken in order to achieve this result.

The processed data at this point in the algorithm is a matrix containing elements whose intensity values are larger at coordinates where a certain transition in a certain direction (an edge) is located. Following the example in the Figure 6, the transition in question occurs from a low to a high intensity (dark to light) in the vertical direction (from the top to the bottom of the image). The edge under segmentation is primarily horizontal. The pixels along this edge must be extracted so that they can be used in determining the location of the edge. The algorithm is able to focus on extracting a maximum of one pixel for each column along the edges length. This will result in a maximum of 640 pixels used in the computation of any horizontal edge and 480 pixels used in the computation of any vertical edge. These values represent the maximum possible elements used for each edge computation, but in practice there is a certain class of pixel which is excluded from any line computation. These pixels are located in the neighborhood of the intersection between the edge currently being segmented and any perpendicular edge. At these locations, the measurement artifact displays a gradual round between the two perpendicular edges, as illustrated in Figure 7. Any pixels located on this transition are not located along the linear edge being segmented. Special consideration is given to the removal of these pixels.

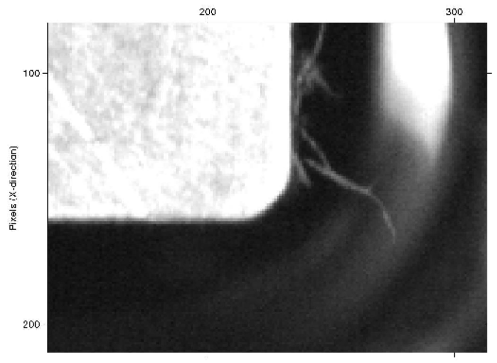

Figure 7: Round between horizontal and vertical edges

The segmentation of the edge is performed as a series of scans across the width of the artifact edge with a window referred to as the $TestBox$. This window starts from the upper-left corner of the processed image and travels downwards in search of pixels belonging to the edge currently being segmented. When the window detects a pixel belonging to the edge, the location of this pixel is recorded, the window shifts one pixel to the right and several pixels upwards,and the scanning then continues downwards in the image until another pixel is recorded. The scanning stops when, column by column, the entire length of the edge has been scanned.

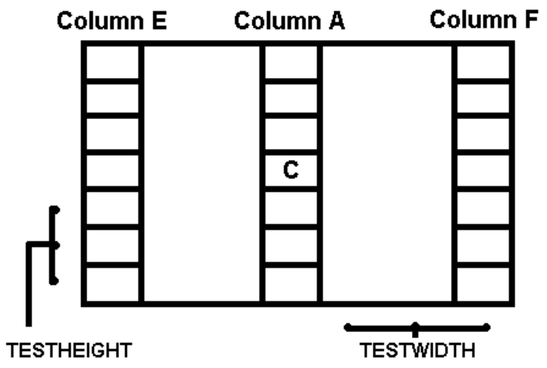

Figure 8: $TestBox$ used in segmenting edges

The window, illustrated in Figure 8, is used to extract pixels that meet a predefined set of requirements.

As the window scans the processed image, point C is centered over the pixel under investigation. There are several requirements that must be met in order for this pixel to be considered part of the edge. First, the pixel must have an intensity greater then zero. All pixels that contain intensity levels below the threshold limit explained in the previous section are not considered to belong to edge. This requirement insures that they are not considered. Second, the intensity at pixel C must be the maximum intensity of all the pixels in the $1 \times m$ column referred to in Figure 8 as column A. This requirement insures that pixel C is the most likely pixel in column A to belong to the edge. The length of column A is a variable defined as,

$$m = (2 \times TESTHEIGHT) + 1 \tag{5}$$

where $TESTHEIGHT$ is the number of pixels above or below the pixel under consideration that should be used in determining its edge-worthiness. An image containing lines located closer together might use a smaller $TESTHEIGHT$ value to be able to distinguish between adjacent lines. The third requirement is that the two $1 \times m$ matrices referred to in Figure 8 as columns E and F must both contain at least one non-zero element. These two columns, spaced a distance defined by,

$$n = (2 \times TESTWIDTH) + 1 \tag{6}$$

pixels apart, are used to exclude the pixel at C if it is located close to any edges perpendicular to the edge being segmented. Figure 9 illustrates that, after the Prewitt line detection and threshold are applied, the transitions that do not occur in the direction being detected by the line detector are no longer visible in the image data. The rounded transitions between perpendicular edges are present, but by scanning a distance $TESTWIDTH$ ahead and behind the pixel under consideration and searching for the region where all vertical edges have disappeared, pixels located on the rounded transitions can be excluded. With these three requirements met, the pixel located at C is added to a data matrix containing all pixels considered part of the edge.

A parameter referred to as $SKIP$ is used to select the spacing of pixels along the length of an edge that are used in determining its location in the image. This parameter can be set to a value of 10 allowing only every tenth pixel along the length of the edge to be used in subsequent calculations. A $SKIP$ value of 10 will provide a faster result since less calculations will be performed by the computer, but it is expected that the use of more pixels in calculating the edge location will provide a more accurate result.

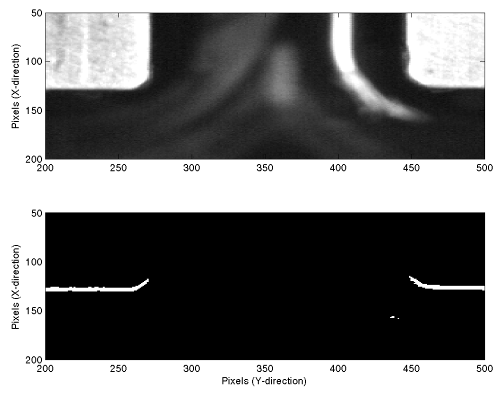

Figure 9: Off-edge pixels near vertices

After the segmentation of one edge is complete, an array of data is created containing the i and j coordinates of all pixels in the image that are considered to belong to the edge. Since each pixel represents an area of $10 \times 10um$, using these pixels in the linear regression computation of the edge will result in a possible $10um$ error in both the i and j coordinates. This error can be reduced through the use of a moment calculation in two directions[3]. Figure 10 illustrates pixelated representation of an edge in an image. By applying the moment calculation, one can observe that the coordinates are effectively shifted based on the intensities of the surrounding pixels and a sub-pixel accuracy is achieved. The moment calculations are implemented using a convolution mask and the following equations,

$$x = \frac{\sum\sum x \times I^P(x,y)}{\sum\sum I^P(x,y)} \tag{7}$$

$$y = \frac{\sum\sum y \times I^P(x,y)}{\sum\sum I^P(x,y)} \tag{8}$$

where $I^P(x,y)$ is the intensity of the pixel at the location (x,y). These values are summed over a 3×3 area. The pixel intensity I^P is taken from the raw image data so that any bias introduced by the processing is removed. A calculation is performed for each of the two coordinates at the location specified by the coordinates in the segmented data array.

After the moment calculation, the segmented data array contains a list of the pixel coordinates, at sub-pixel accuracies, determined by processing to belong to the edge under segmentation. An attempt is then made to optimize the linearity of this set of data points using a recursive function. The data matrix is sent to the recursive function and linear regression is used to fit a line through the data. Using this 'best fit' line,

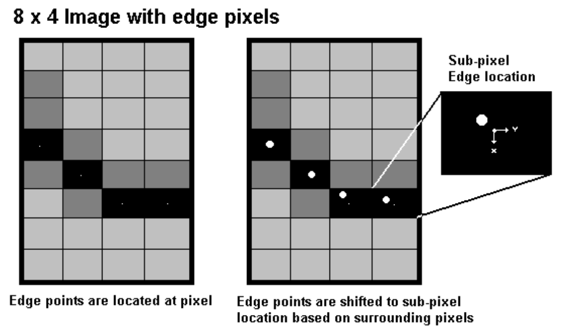
8 x 4 Image with edge pixels
Sub-pixel Edge location
Edge points are located at pixel
Edge points are shifted to sub-pixel location based on surrounding pixels

Figure 10: Shift caused by the moment calculation

the perpendicular distance between the line and each of the sets of coordinates is calculated. Another set of data is created containing all coordinates from the original data set excluding the one set of coordinates with the largest perpendicular distance. The linear correlation coefficient is calculated for both sets of data, resulting in the current linear correlation coefficient and the potential difference in the coefficient if the specified coordinate is removed. If either the coefficient or the difference are below a pre-determined tolerance then the smaller data set is used in the subsequent recursion of the function. The recursion is set to continue until both of the requirements are met or a maximum of 50 coordinates are removed. The final result of this function is the calculation of the slope/y-intercept form of the 'best fit' line through the optimized data.

This same segmentation continues for all four edges located in each image. The algorithm for segmenting each edge is very similar with only a few exceptions; when segmenting the lower horizontal edge in an image, the $TestBox$ scans the edge from the lower-left corner travelling upwards; when scanning the vertical edges, the entire image is transposed and the same algorithm is used as that for the horizontal edges (this results in the slope/x-intercept of the lines).

3.2.5 Projective Transformation

The image data at this point in the algorithm consists of four first-order slope/intercept equations describing the four edges in the image. By solving the appropriate equations together, the coordinates of the four points of intersection can be calculated. These four coordinates are described in terms of pixels rather then an applicable unit such as millimeters, and they contain distortion caused by the inclusion of perspective in the images. The removal of distortion and the proper scaling of the data is performed through the use of a projective transformation.

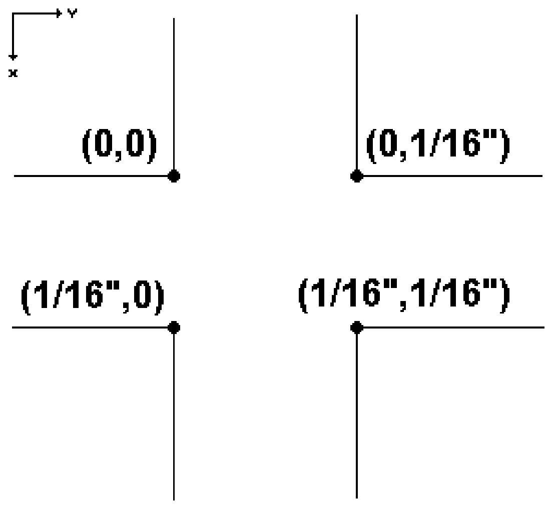

Figure 11: Four points used to calculate **T** matrix

The measurement artifact was machined to a width of $1/16$" along both the horizontal and vertical grooves. Using each of the four vertices located at the intersection of the edges, four points can be characterized with the coordinates shown in Figure 11. The pixel-based location of these four points is extracted from the image of the measurement artifact, although the location of these points is distorted by perspective and given in pixel units. Using these eight coordinates, a solution can be found for the following set of equations,

$$\rho W = \mathbf{T}w \tag{9}$$

$$\rho X = \mathbf{T}x \tag{10}$$

$$\rho Y = \mathbf{T}y \tag{11}$$

$$\rho Z = \mathbf{T}z \tag{12}$$

where W,X,Y, and Z are each one of the four scaled coordinate sets and w,x,y, and z are the corresponding image coordinates. ρ is a scaling factor set equal to one. The solution to these equations, a transformation matrix **T**, can be constructed to transform points from the image space to a scaled world coordinate system. The origin of the image ($pixel(1,1)$) is then transformed to the world coordinate system and shifted to its origin. This transformation shifts the location of the measurement artifact vertices to a location in the positive x and y directions.

3.2.6 Centre Point

At this point in the algorithm, the data consists of the four coordinates of the measurement artifact scaled to the proper dimensions and comprising the vertices of a perfect square. A simple mean-value calculation in both the x and y directions results in the centre coordinate of the artifact with respect to the image origin using the units millimeters.

4 Validation and Calibration

The algorithm is able to extract measurements characterizing the position in the X-Y plane of the robot end-effector with respect to the location of the measurement artifact. The resolution of these measurements is not known. In order to determine and improve this resolution, a calibration and validation procedure is performed.

The Vernier X-Y table is a displacement device that controls the position of an object in the X-Y plane. The device, displayed in Figure 12, has two perpendicular axes. The displacement along each axis is manipulated by one manually controlled actuator with a resolution of $5.08um$ (0.0002"). The validation procedure uses this device to determine the resolution of the image processing output.

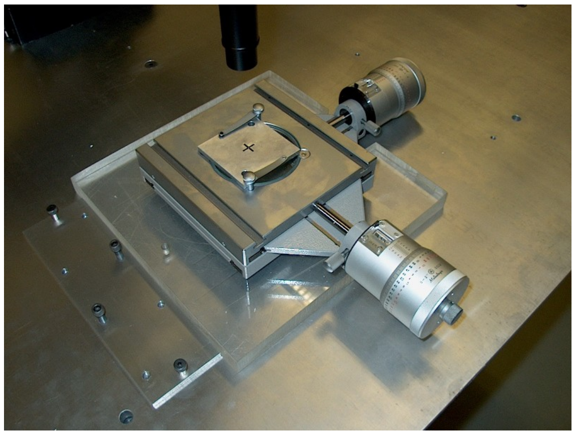

Figure 12: The Mitotoyo Vernier X-Y table

The measurement artifact is fixed onto the Vernier X-Y table and placed into the robot workspace. The robot and measurement head are positioned such that the camera can clearly extract images of the measurement artifact for processing. Both the robot and measurement head remain stationary for the calibration procedure. The Vernier X-Y table is used to displace the measurement artifact within the field of view of the camera, into 40 positions. At each position, an image of the artifact is extracted and processed. These 40 positions construct a square pattern of dimensions $50.8x50.8um$. This pattern can be viewed in Figure 13. The displacement of the measurement artifact is also extracted from the images taken at each location. By comparing the displacement from the images with that of the Vernier X-Y table, a reasonable estimate of the resolution of the image processing algorithm can be obtained.

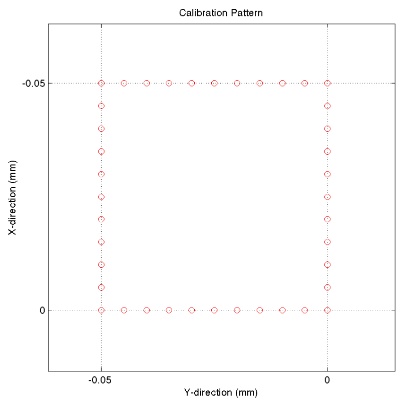

Figure 13: The displacement pattern plotted by the X-Y table

After this test was performed several times, it was determined that the two axes of the Vernier X-Y table might not be positioned exactly perpendicular to each other. As a result, the comparison of image-extracted displacements to the displacements measured using the Vernier X-Y table uses a 'well-fit' parallelogram. The use of this parallelogram eliminates most of the error that would be incurred from the use of the offset Vernier X-Y table.

A diagram of a typical comparison of image data to actual data can be viewed in Figure 14. In this figure, data points extracted from an image are marked with a plus $(+)$ sign. The points as measured using the Vernier X-Y table are marked with small circles (o). The lines between the markers represent the error between that image point and the actual point. For the data in Figure 14, the calculated mean displacement error is determined to be $0.0170mm$ and the maximum displacement error is measured at $0.0372mm$. A calibration exercise is used to improve this result.

The discrepancy in values between the image data points and Vernier Table points is partially a result of the resolution of the Vernier Table, but mostly a result of poor optimization of the image processing algorithm. There are several parameters within the body of the algorithm must be properly selected in order to increase the measurement accuracy of the camera.

The parameter names are $FILTER$, $TESTHEIGHT$, $TESTWIDTH$, and $SKIP$. The parameter $FILTER$ refers to the use of one of the two filters mentioned in section 3.2.1. The

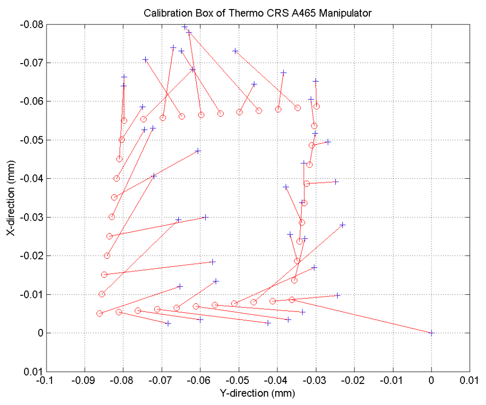

Figure 14: Comparison between actual and measured data (Trial 1)

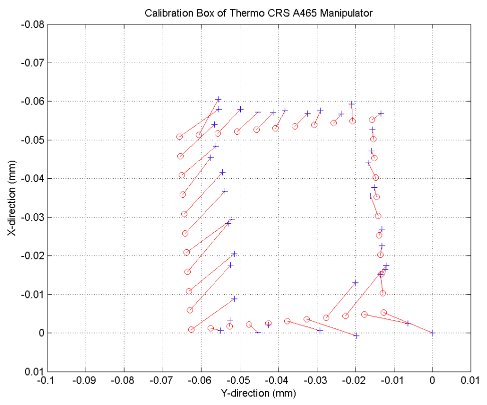

Figure 15: Comparison between actual and measured data (Trial 7)

Table 1: Parameter optimization table

Trial	Filter	SK	TW	TH	MaxE	Mean E
1	None	20	5	10	0.0372	0.0170
2	None	10	5	10	0.0197	0.0106
3	None	1	5	10	0.0170	0.0081
4	Mean	1	5	10	0.0150	0.0081
5	Gauss	1	5	10	0.0160	0.0083
6	None	1	25	10	0.0177	0.0085
7	None	1	5	5	0.0168	0.0080
8	None	1	5	15	0.0171	0.0081
9	None	1	5	20	0.0171	0.0081

parameters $TESTHEIGHT$, $TESTWIDTH$, and $SKIP$ are all explained in section 3.2.4. The proper selection of these parameters will result in a higher resolution of the measurements produced by the algorithm.

The calibration procedure consists of setting each of the four parameters and then performing the validation procedure to establish the mean and maximum error in the readings. The algorithm is optimized by selecting parameters that produce a lower mean and maximum error. Table 1 documents several results of this optimization where the values SK, TW, TH refer to the parameters $SKIP$, $TESTHEIGHT$, and $TESTWIDTH$ respectively.

The parameters that produce the best combination of mean and maximum error correspond to Trial 7 in Table 1. Figure 15 is a plot of the error associated with each measurement during Trial 7.

5 Repeatability Test

The algorithm developed in this paper will be used in the camera-based robot calibration system. The algorithm is able to extract measurements to a maximum resolution of $0.0168mm$ with an average resolution of $0.0080mm$. Since both of these values are well below the $0.05mm$ repeatability of the Thermo CRS A465 manipulator, this algorithm, along with the measurement head and data acquisition equipment, can be used to characterize the repeatability of this device.

The repeatability of a robot is established as its ability to reproduce certain measurements under the repeated application of a stated value and under certain conditions[4]. In terms of calculations the repeatability is given by[5],

$$Repeatability = \bar{l} + 3\sigma \tag{13}$$

where σ is the standard deviation and $\bar{l}$ is the mean attained position. The following equations are used to calculate σ and $\bar{l}$,

Attained Position i: X_{ai}, Y_{ai}

Mean Attained Position:

$$\overline{X} = \frac{1}{N} \times \sum X_{ai} \tag{14}$$

$$\overline{Y} = \frac{1}{N} \times \sum Y_{ai} \tag{15}$$

$$l_i = \sqrt{(X_{ai} - \overline{X})^2 + (Y_{ai} - \overline{Y}-)^2} \tag{16}$$

$$\bar{l} = \frac{1}{N} \times \sum l_i \tag{17}$$

$$\sigma = \sqrt{\frac{\sum(l_i - \bar{l})^2}{N-1}} \tag{18}$$

where l_i is the attained position. The experiment will test the manipulator at approximately 50 percent of its $2kg$ rated load and approximately 80 percent of its maximum velocity. The repeatability tests consist

Table 2: Results of joint repeatability test

Joint	Repeatability (mm)
1	0.01664
2	0.01813
3	0.02321
4	0.01671
5	0.05299
6	0.01601

of separate actuation of each of the robots six joints followed by two sequences of arbitrary robot motion using all six joints. The results of the repeatability test can be viewed in Table 2 and in Figure 16 and Figure 17.

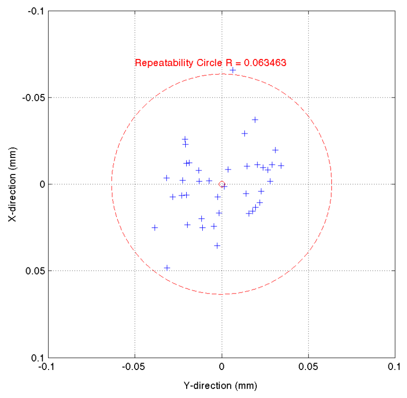

Figure 16: Repeatability of A465 using arbitrary sequence of motion (R = 0.06346)

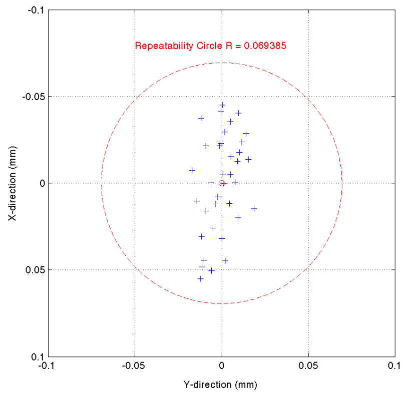

Figure 17: Repeatability of A465 using second arbitrary sequence of motion (R = 0.06939)

6 Conclusions and Future Work

The algorithm for extraction of metric information has been designed and validated. According to the validation procedure, the algorithm produces a maximum error of $0.0168mm$ and a mean error of $0.0080mm$. This error is well below the repeatability of the A465 manipulator (havinf a range of $0.1mm$) and can be used in its calibration. Future work on this project will involve the integration of a laser displacement sensor for the measurement of displacement in the Z-direction. This reading can also be included in the calculation of the repeatability of the manipulator, as it is in standard practice. The algorithm must also be tailored to extract information from a second measurement artifact that resembles a ruler. This second artifact will be used in the manipulator calibration. A thorough optimization of the four parameters will be performed in an attempt to further reduce the measurement error.

References

[1] Andrew Fratpietro, Nick Simpson, and M. John D. Hayes. Advances in Robot Kinematic Calibration. Technical report, Carleton University, 2003.

[2] Ramesh Jain, Rangachar Kasturi, and Brian G. Schunck. *Machine Vision*. MIT Press and McGraw-Hill, Inc., 1995.

[3] Ronald Ofner, Paul O'Leary, and Markus Leitner. A Collection of Algorithms for the Determination of Construction Points in the Measurement of 3d Geometries via Light-Sectioning. *Institute for Automation at the University of Leoben*, 1998.

[4] A. T. J. Hayward. *Repeatability and Accuracy*. Mechanical Engineering Publications Limited, 1977.

[5] Nicholas G. Dagalakis. Industrial Robot Standards. *Chapter 27 in the Handbook of Industrial Robots*, 1998.