Close-up Stereo Triangulation with Application to Sheet Metal Strain Analysis

CLOSE-UP STEREO TRIANGULATION WITH APPLICATION TO SHEET METAL STRAIN ANALYSIS

By

JOHN PHILIP MITCHELL, B.Eng

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Engineering

McMaster University

April 2005

MASTER OF ENGINEERING (2005)	MCMASTER UNIVERSITY
(Electrical and Computer Engineering)	Hamilton, Ontario

TITLE:	Close-up Stereo Triangulation with Application to Sheet Metal Strain Analysis
AUTHOR:	John Philip Mitchell B.Eng
SUPERVISORS:	Dr David W. Capson and Dr Allan D. Spence
NUMBER OF PAGES:	xii, 117

Abstract

This thesis describes an investigation into the experimental accuracy of a close-up stereo vision system. Surfaces are measured from any orientation through the synergy of a Coordinate Measurement Machine (CMM) and a custom made stereo vision head. Calibration of the stereo cameras is achieved using bundle adjustment non-linear optimization, commonly used in photogrammetry. Point set registration techniques are used to calibrate the stereo head with respect to the CMM coordinate system. The fully calibrated stereo vision system is used to provide experimental accuracy results under ideal imaging conditions. The system is then applied to the practical problem of sheet metal strain analysis. This problem involves many non-ideal imaging components that affect system accuracy. Experimental results for this non-ideal situation are provided in the form of three-dimensional strain distribution plots.

Acknowledgements

I would like to thank Dr Spence and Dr Capson for their suggestions and direction. I would like to thank Harley, Dave and Marc for their suggestions and helpful knowledge of the metrology equipment. Finally I would like to thank my friends and family for their support and patience.

Contents

List of Tables

List of Algorithms

List of Figures

List of Abbreviations

CMM	Coordinate Measurement Machine
CCD	Charge Coupled Device
CCS	Camera Coordinate System
SCS	Stereo Coordinate System
RCS	Ram Coordinate System
WCS	World Coordinate System
GCS	Grid Coordinate System
RBT	Rigid Body Transformation
MOD	Minimum Object Distance
DOF	Depth Of Field
SVD	Singular Value Decomposition
FIFO	First In First Out

Chapter 1

Introduction

1.1 Visual Metrology

Research into determining the Three Dimensional (3-D) position of objects within an imaged scene began with the photogrammetry community and the use of film cameras. The 3-D locations of specific points were manually determined within aerial photographs covering large areas. These techniques were also used at close range distances for use in the surveying of land and architectural features. With the development of fast computers and digital cameras the computer vision community has further developed close range visual measurement, applying it to features in room size scenes. In many applications even closer range measurement would be advantageous. By decreasing the object distance, the feature resolution is improved resulting in increased measurement accuracy. This increased accuracy comes with the price of having to process more images for a desired scene. But with the continual increase in computing power the potential for these closer ranges is apparent. Many close up images can be processed in a reasonable amount of time and the measurement data stitched together to provide a final reconstruction of a scene.

Many methods of determining 3-D data from Two Dimensional (2-D) image data exist. One method that uses the principle of triangulation to measure distance is that of binocular stereo vision. In the computer vision community, it is typically applied over distances ranging from 0.5 m to 5 m and used for robotic manipulator positioning or tracking applications. Over this range, system accuracies are usually in the millimeter range. This is reasonable for some positioning applications but is not accurate enough for use in dimensional part inspection. Modern metrology systems have accuracies ranging in the $< 10\ \mu$m for touch probe based systems to one micrometer for some laser based systems. However, these systems are expensive and also have a slow data acquisition rate. Stereo vision has a major advantage in that it is inexpensive and has the potential to gather large amounts of 3-D data relatively quickly. It is also preferable for many applications because of its passive nature. Unlike laser based metrology systems it only requires white light illumination. Despite these benefits stereo vision has many difficulties. The stereo vision paradigm and some of its practical problems are presented in the following section.

1.2 Stereo Vision

In order to relate 3-D points to 2-D image data a model describing the transformation process must be created. The pinhole camera model is the most widely employed. This model assumes that only light rays passing through an infinitesimal pinhole are projected onto the image plane, see Appendix A for details. A binocular stereo model consisting of two pinhole cameras is illustrated in figure 1.1. The cameras are identical except for for a translation in X by a baseline distance of b. Using the two camera centers C_l and C_r, an epipolar plane can be defined for each 3-D point P within the object space. The intersection of an epipolar plane with the image planes define an epipolar line. A point P will be imaged at locations p_l and p_r in the left and right images along the same epipolar line. This is of course not the case in practical systems,

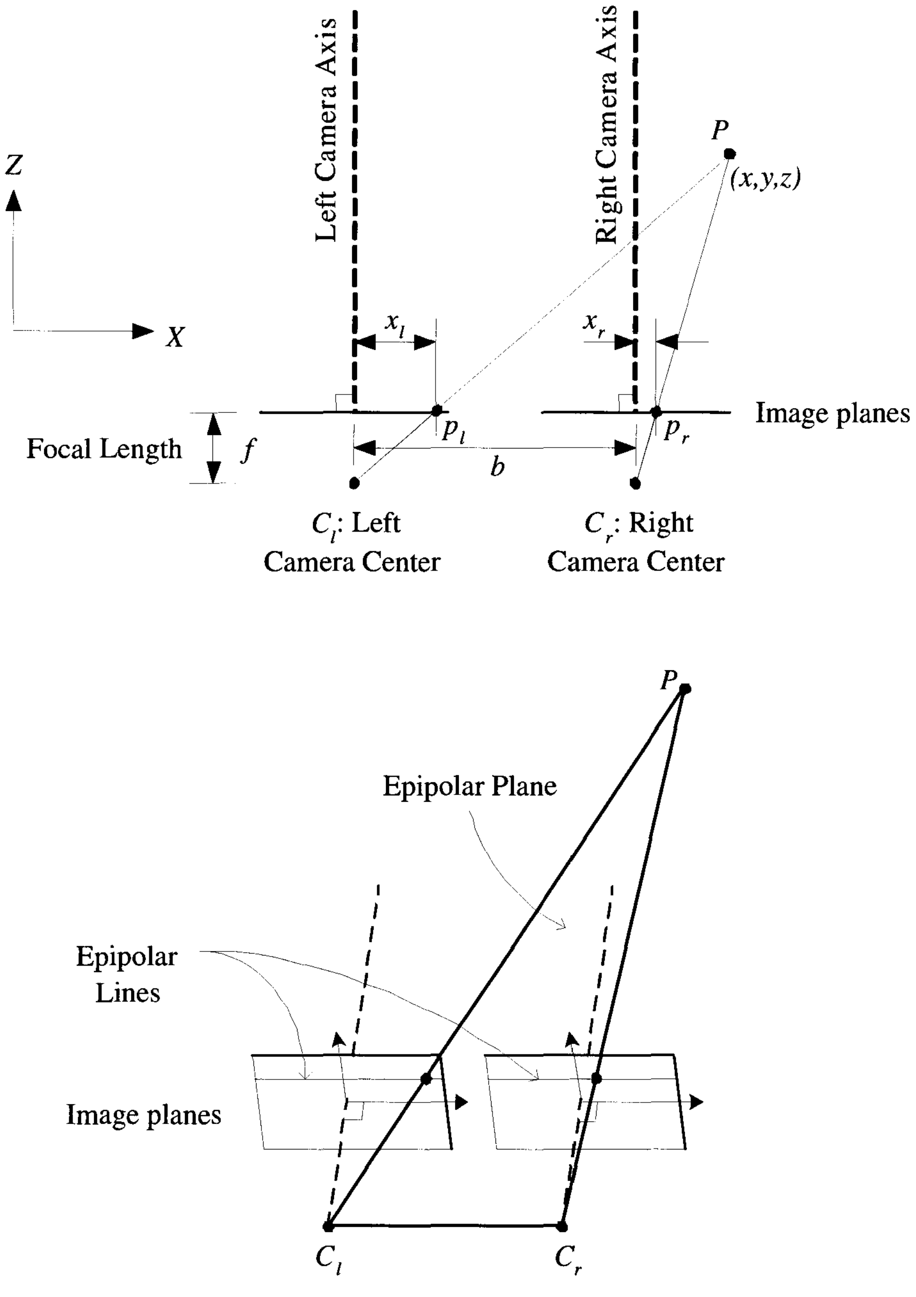


Figure 1.1: Pinhole Camera Stereo Model

even with near perfect camera alignment image noise and differing internal camera parameters contribute to points lying slighty off the epipolar line. The imaged points p_l and p_r define a conjugate pair. With known camera parameters and conjugate pairs, object points can be reconstructed through triangulation. Choosing the origin of the reconstructed coordinate system to coincide with the left camera center and using similar triangles, the following results can be determined:

$$x = \frac{zx_l}{f}, \tag{1.1}$$

$$y = \frac{zy_l}{f}, \tag{1.2}$$

$$z = \frac{bf}{d}, \tag{1.3}$$

where the disparity is defined as $d = (x_l - x_r)$. It can be seen that the depth of the reconstructed point is inversely proportional to the disparity.

The primary problems with stereo vision are calibration, correspondence and triangulation. In practice it is difficult to build stereo systems with nonverged geometry such as that shown in figure 1.1. Internal camera parameters such as focal length are also not known accurately. A calibration must be performed to find the pose and internal parameters of the camera. The pose refers to both the orientation and the position of a coordinate system within a global coordinate system. Calibration is usually done using 3-D control points. The accuracy of the parameters determined through calibration are highly dependent on the accuracy of the control point locations. Assuming a stereo system with a non zero vergence has been calibrated, epipolar geometry can be synthetically achieved through the process of rectification. Using the calibration information and image data, the process of rectification creates a new set of calibration information and image data that conforms to the epipolar geometry.

The correspondence problem consists of detecting corresponding points that form conjugate pairs. Detection and matching of these points can require a great deal of computing power. Early imaging technology required expensive hardware and a dedicated computer for each camera, but this problem has been eliminated in recent years with fast inexpensive computers. However, even with this improved computing power, the matching problem is still formidable. It is complicated by factors such as lighting, occlusion and multiple similar features.

Triangulation is the process of finding the 3-D points that correspond to conjugate point pairs. Whether the epipolar geometry is physically achieved or if it is determined through rectification there will be an difference between the vertical components of conjugate points. This will cause the rays used for triangulation to not intersect.

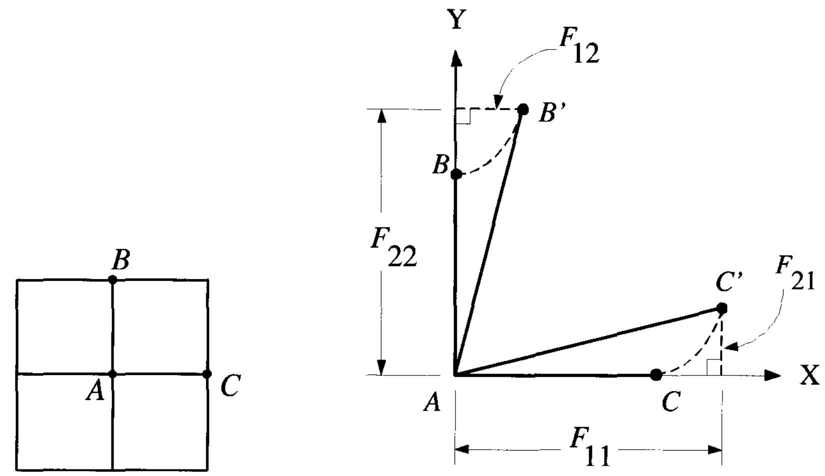
The problems outlined above have made the application of stereo vision to 3-D metrology difficult. One area in which a practical stereo vision system can be developed is for quality control in the sheet metal stamping process. This process and how stereo vision can be applied to quality will be discussed in the next section.

1.3 Sheet Metal Strain Analysis

Sheet metal stamping is an important process for the manufacturing of parts in the automotive, aerospace and consumer product industries. The stamping process consists of a die closing over a sheet metal blank producing the desired shape. On the down stroke the die closes onto the part and the metal first undergoes elastic and then plastic deformation. On the upstroke, only the plastic deformation remains, the loss of the elastic deformation results in a part shape which differs from the die. This is known as springback.

For a sheet metal part to be correct it is required to: 1) have the correct geometry; and 2) be within surface strain limits. Incorrect geometry can occur because of excessive springback and incorrect surface strains can result in part fracture and/or wrinkling. To avoid these defects the stamping process must be evaluated during die setup and also periodically during regular production. This is commonly done with the use of two measurement systems, one for part geometry and the other for part strain distribution. Correct geometry is normally measured using a CMM with an attached touch trigger probe. Surface strain is determined by stamping a specially prepared test part. Prior to stamping, a grid of known size is electrochemically etched onto the flat test part. The change in the grid element dimensions after stamping indicates surface strain over the part. The deformation of each grid element is usually measured manually or using an automated vision system. Two commonly used grid patterns are the circle and square grid patterns. The circle is more prevalent in industry because of ease of measurement but it also has a major disadvantage in that it cannot detect non-proportional deformation [32]. A square grid can be used to detect non-proportional deformation and the grid intersections can also be used for a 3-D reconstruction of the part. Using a square grid the true natural strain can be determined as shown in figure 1.2(a). To calculate the principal strains at point A, each grid element surrounding A is divided into two triangles, e.g. $\triangle ABC$. A strain value is determined for each of the triangles closest to A, the final strain value is the average of these neighborhood strain values. The 3-D points from each triangle surrounding A are transformed to a local 2-D coordinate system with A as the origin (Figure 1.2(b)). The principle strains are then calculated using the method proposed by Sowerby el al. [37]. Under homogeneous deformation, the points B and C are transformed to the new coordinates B', C'. To mathematically describe this deformation, linear functions are defined that map the points from their original coordinates (x, y) to the new coordinates (X, Y) , such that:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{1.4}$$



(a) Group of four grid elements

(b) Homogeneous deformation of one triangle

Figure 1.2: Determining surface deformation from 4 grid elements

where the matrix of F values contains the coefficients of the system equations that represent the deformation gradient tensor. For points B and C, which are assumed to undergo the same homogeneous deformation, the following system can be formed:

$$\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & 0 & 0 \\ X_2 & Y_2 & 0 & 0 \\ 0 & 0 & X_1 & Y_1 \\ 0 & 0 & X_2 & Y_2 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{21} \\ F_{22} \end{bmatrix}, \text{ or } \quad x' = X'F'. \tag{1.5}$$

Rearranging Eq. 1.5, the coefficients of the gradient tensor are defined as $F' = X'^{-1}x'$. The eigenvalues $\lambda_{1,2}$ of F can be computed using Singular Value Decomposition (SVD), and satisfy the relationship $\varepsilon_{1,2} = \ln(\lambda_{1,2})$ where, ε_1 and ε_2 are the true or natural surface strains. For the purposes of evaluating strain accuracy the engineering strain is often used because for small strains it is essentially identical to the true strain. The engineering strain related to the direction defined by two 2-D points (x_1, y_1) and (x_2, y_2) can be stated as:

$$\mathrm{e}\,\% = \frac{L' - L}{L} \times 100\,\%, \tag{1.6}$$

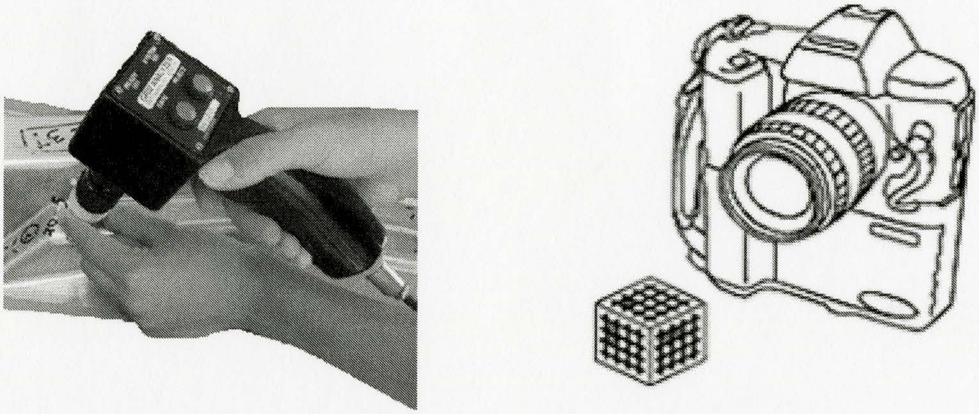
where $L' = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$ and $L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ are the deformed and original lengths respectively. All strain accuracies within this thesis will be stated in percentage form.

The application of stereo vision to sheet metal quality requires a system to accurately detect the 3-D location of square grid intersection points. The 3-D intersection points can be used to give a 3-D reconstruction of the part, providing data to check for the correct geometry. The stereo vision system must also be able to determine the grid structure because adjacent points are necessary to calculate surface strain. Previous approaches to the strain analysis problem using vision are reviewed in the following section.

1.4 Previous Work

The measurement of surface strains was originally performed manually using a ruler or graduated tape. This process is still in use today but it is time consuming and the measurement accuracy is prone to human error. With the appearance of CCD cameras and fast computers, automated strain analysis systems quickly followed. Many of these systems determine surface strains by imaging the actual forming process. They require an elaborate setup with a vision system being integrated into the forming machine. Usually, they are only used in laboratory environments for testing purposes and will therefore not be discussed further. This review will focus on automated surface strain methods that use deformed parts with etched patterns.

The earliest automated strain analysis system was the circle grid analyzer [38]. This hand held device contained a CCD camera with a fixed focus lens. Figure 1.3(a) shows a modern commercial implementation of this device. Grid elements are captured individually by taking an image when the end of the lens assembly is in contact with and covering the desired grid element. The lens assembly contains an internal



(a) FMTI grid analyzer [1]

(b) High resolution camera with reference cube [2]

Figure 1.3: Automated strain analysis systems

light source and is designed so that objects are in focus at its end. The system is calibrated giving an image to object space transformation that allows physical distances to be determined. The grid elements are correctly imaged by viewing the camera output in real time on an attached monitor. The image processing usually consists of binary thresholding to grab the grid element data. The appropriate geometric forms are then fitted to the binary data, circular grid elements are assumed to deform into ellipses and are fitted using a least squares method. The square grid is assumed to deform into quadrilaterals, which are normally modeled using four straight lines and again fitted with a least squares technique. Most commercial systems of this type have very good strain accuracy in the range of ±0.5 to ± 1.5 %. One obvious disadvantage of this system is that it cannot give a 3-D reconstruction of the part. It also has problems with small recessed areas because the lens assembly cannot be placed in direct contact with the part surface.

Almost all other vision based approaches to strain analysis involve using triangulation to locate surface point data. Most of these systems are for use with square grids because they provide specific points which can be detected and refined to sub-pixel accuracy if necessary. A triangulation vision system presented by Vogel et al. [27] uses a stationary CCD camera and a turntable with encoder. The camera is calibrated for the appropriate scene dimensions prior to measurement. A part is etched with a square grid pattern and imaged from two or more turntable positions. The first image processing step uses filters to reduce background noise and equalize image contrast. The grid structure intensity data is segmented from the rest of the image using a binary thresholding algorithm. The user is then required to manually patch any broken grid lines. Broken grid lines are common because of both the etching procedure and the stamping process. The perfect grid structure is then thinned to a single pixel width using morphological operations. Stereo matching is initialized with the user choosing an initial point match, then a mapping algorithm determines corresponding grid intersections by following line segments from one intersection to the next. With known camera parameters, encoder values and corresponding 2-D points between images, the 3-D point data can be found using triangulation. A publication [20] comparing the strain accuracy of various methods provides a strain accuracy of ±2 % for this technique. Some disadvantages of this approach include the fact that it is not very portable and part size is limited by the size of the turntable.

Another recent approach presented by Manthey et al. [19] involves using a high resolution camera with a reference object of known dimension. Figure shows a diagram of the camera and the 25 mm^3 reference cube. With the calibration cube present within each image, no prior calibration of the camera is necessary. Using features on the reference cube, a direct linear transformation [47] can be performed which produces a set of coefficients. The coefficients from two or more images can be used to triangulate 3-D points associated with corresponding 2-D points. Strain accuracy for two images is ±1.5% to ± 2 %, with the strain accuracy increasing with the number of overlapping images. No details on the image processing methods for this technique are given.

A method using a binocular stereo vision approach is presented by Orteu et al. [28]. Two calibrated cameras image a part with a square grid pattern from different positions. Grid intersections within each image are first approximated using a scanning method which the authors do not describe. The grid intersections are then detected to sub pixel accuracy using a model based approach. Each 2-D grid intersection is modeled using a cross consisting of two intersecting lines. Various cross parameters such as line intensity, center position and line direction are adjusted to best fit the image intensity values at each grid intersection. The objective function to be minimized it the correlation of two small windows, one centered at the current grid intersection location and containing image data, the other being the cross model. This continuous function can be minimized using an optimization technique. The matching process is the same as above, with a user defined point, a mapping algorithm determines corresponding points, grid lines are assumed to be intact. The 3-D accuracy of the system is stated as 1 part in 6000, strain accuracy is not stated and cannot be determined without knowledge of the measurement range. One disadvantage of this system is that the model based approach is computationally intensive. The objective function consists of a summation which must be evaluated using the current best parameters for every iteration of the optimization algorithm.

This review focused on recent systems published within the literature. There are various commercial systems available but many of them are not discussed because details of their operation are not available. It is apparent from the material presented above that the image processing techniques require various manual steps. The image processing procedure presented within the next section is completely automated.

1.5 Thesis Objectives

There are four objectives that were to be achieved through the work presented within this thesis; 1) The first objective was to design and build a close up stereo vision system. This system was to be applied to the problem of measuring the dimensions and surface strains of sheet metal parts. 2) The second objective was to develop automated image processing and stereo vision algorithms that accurately measure 3-D grid intersections on the surface of sheet metal parts. 3) The third objective was to experimentally determine the accuracy of the vision system. The two accuracies of interest are the 3-D reconstruction accuracy and the strain accuracy. These results determine whether the system is accurate enough to be used for sheet metal quality analysis. 4) The final objective was to measure sheet metal parts and create 3-D models the parts showing surface strain.

The remainder of this thesis is organized as follows. Chapter 2 describes the image processing procedures that were employed to accurately determine the 2-D grid intersection points. This chapter also describes the algorithm developed to perform stereo matching. Chapter 3 describes the stereo vision system hardware and some of the difficulties with close up vision. The camera and probe head calibration algorithms are also described in this chapter. Chapter 4 describes the experiments that were performed to determine the system accuracy. Both the surface strain and 3-D reconstruction accuracies are presented. This chapter also shows 3-D surface strain plots for two parts that were measured using the vision system. The final chapter states a conclusion and suggests some future avenues of research.

Chapter 2

Image Processing and Stereo Matching

2.1 Introduction

Sheet metal strain analysis and 3-D part reconstruction require the 3-D location of grid intersection points. Surface strain analysis also requires that the structure of the grid be known. The problem is therefore to find the grid intersections as accurately as possible and to determine the structure of the grid. The proposed method is illustrated by the flowchart in figure 2.1. The image processing steps are done on both the right and left images. The first image processing step segments the grid from the background of the image using adaptive binary thresholding. This segmentation technique can effectively deal with specular reflection and uneven lighting. After thresholding, the grid has many breaks, mostly due to problems in the etching of the grid. These breaks are closed by morphological dilation. A circular structuring element thickens the grid causing the breaks to close. After dilation, the grid is eroded using the same structuring element to thin the grid lines back to their original width. The square like grid elements located between grid lines are then incrementally labeled

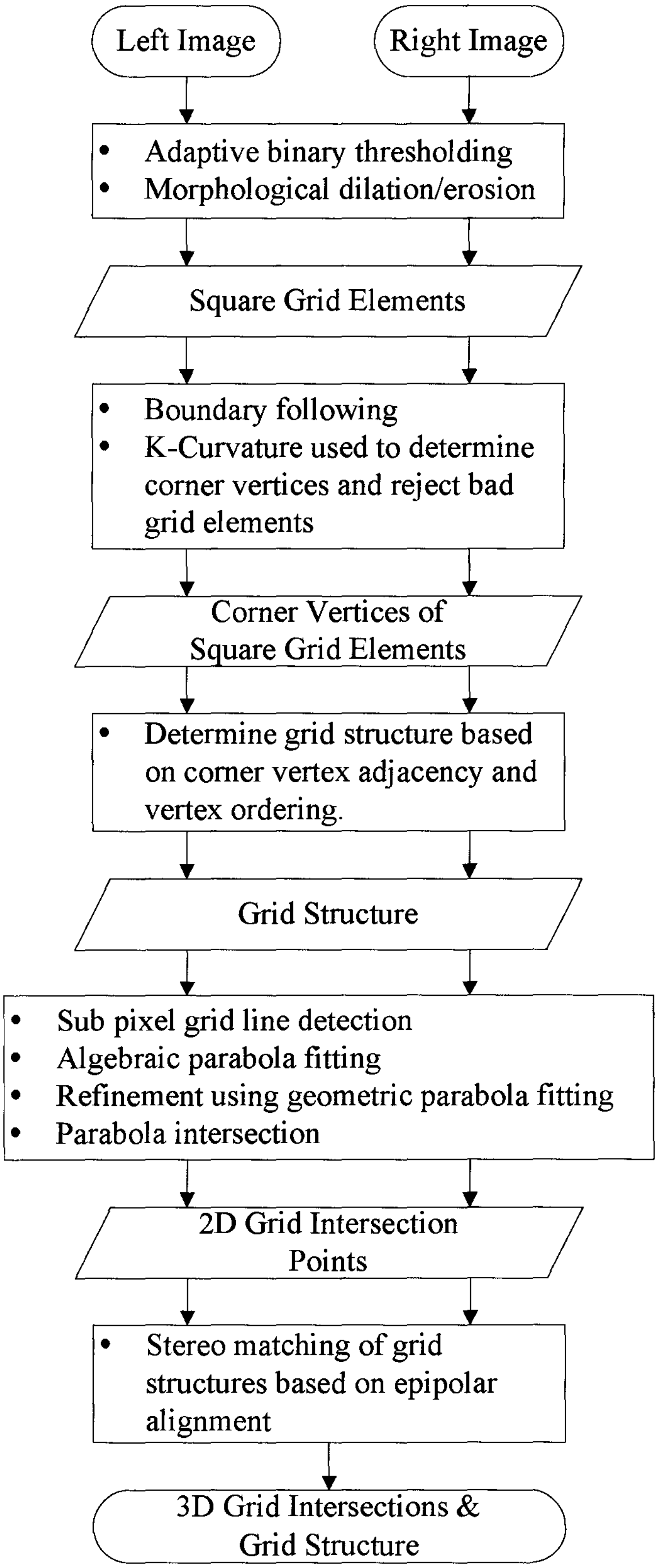

Left Image
Right Image
• Adaptive binary thresholding
• Morphological dilation/erosion
Square Grid Elements
• Boundary following
• K-Curvature used to determine corner vertices and reject bad grid elements
Corner Vertices of Square Grid Elements
• Determine grid structure based on corner vertex adjacency and vertex ordering.
Grid Structure
• Sub pixel grid line detection
• Algebraic parabola fitting
• Refinement using geometric parabola fitting
• Parabola intersection
2D Grid Intersection Points
• Stereo matching of grid structures based on epipolar alignment
3D Grid Intersections & Grid Structure

Figure 2.1: Image Processing Flow Chart

and the boundary of each is determined using a boundary following algorithm. The curvature along the boundary of each element is then estimated using the k-curvature. The four corners of each grid element are then identified as the boundary points of largest k-curvature. The adjacency of corner vertices between grid elements are used to create a reconstruction of the grid structure. A matrix of grid element labels is used to describes the physical grid structure. It is important that this reconstruction is absolutely correct because it is later used to do stereo matching. Within this grid structure grid intersections are associated with square groupings of four grid elements. These square groupings define a crossing of two grid lines. The grid lines between adjacent grid elements are identified in the image and their centerlines detected with a sub pixel edge detector. Two parabolas are then fitted to the sub pixel line points for each square grouping. Parabolas are used because the grid lines between grid elements are assumed to be smooth and of almost constant curvature. This is an acceptable assumption because accurate strain analysis requires that the grid size be small compared with the grid deformation. The first parabola fitting is done using a general least squares approach, this result is then improved using a geometric parabola fitting. The final image processing step is to determine the intersection of the two parabolas for each grouping. With image processing of right and left images done, the grid structure is used to do stereo matching. The alignment of the left and right grid structures that produces the least epipolar error is chosen as the correct stereo matching. These steps are explained in detail in the following sections.

From the details below, it should be noted that the image processing procedure is completely automated. This is essential because a small part may take up to 150 right/left pairs of images to fully capture, tweaking image processing parameters for each individual image to get the best possible result cannot be done.

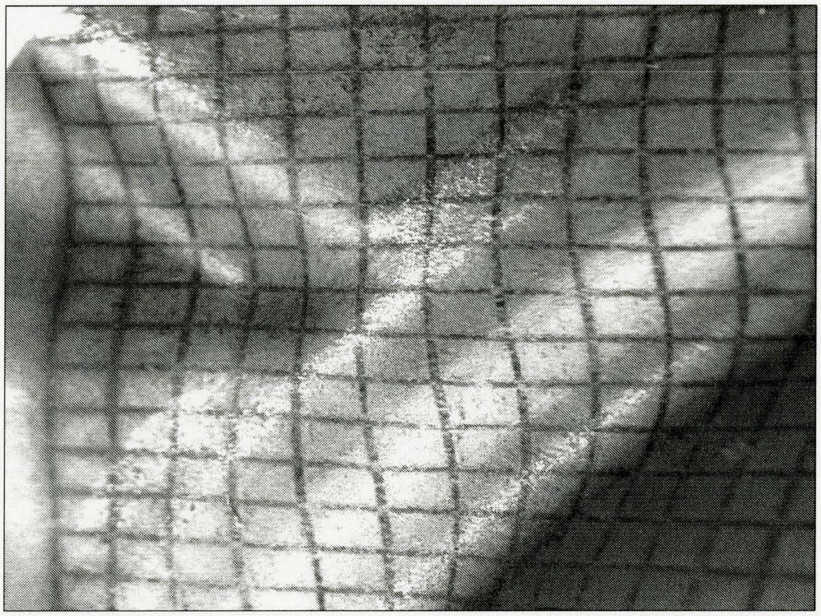


Figure 2.2: Grid image showing specular reflection

2.2 Grid Detection

2.2.1 Adaptive Thresholding

Segmenting the grid from the rest of the image can be accomplished with binary image thresholding. The most basic binary thresholding chooses a single threshold using a histogram of image gray scale values. With an image containing grid and background intensities which are homogeneous and a modest amount of random noise, a histogram would show two distinct peaks, allowing a single threshold value to be chosen. This type of thresholding will not work on typical sheet metal images such as figure 2.2. Varying lighting intensity, specular reflection and problems in the etching of the grid cause both the grid and background to be non homogeneous. These types of problems can be overcome by using an adaptive thresholding technique that varies the threshold value based on local image information.

The grid segmentation problem is almost identical to the well researched problem of enhancing degraded grayscale text documents. Both problems involve images with non-homogeneous background noise, nonuniform illumination and missing information. A parameter that is often used to aid in text document binary thresholding because of its consistency is the stroke or width of characters. One recent algorithm that uses the stroke of a character to help threshold an image was presented by Yang and Yan [46]. The first step in the algorithm is to determine the average stroke width within the image. An automated method involving run-length histograms is described. For this work the stroke width of the grid is known. The grid size does change because of perspective projection, but it does not change significantly because the camera DOF is so small. The stroke width, SW, is used to help determine the binary threshold value, T, for a point P within the image using the following steps:

1. Calculate $I_{Wmax}(u,v) = \max\limits_{(u_i,v_i)\epsilon W} I(u_i,v_i)$ and $I_{Wmin}(u,v) = \min\limits_{(u_i,v_i)\epsilon W} I(u_i,v_i)$, where W is the $(2W+1) \times (2W+1)$ window centered at the point $P = (u,v)$ and I is the gray scale image value or intensity.

2. Calculate $a = |I_{Wmax}(u,v) - ave(P)|$ and $b = |I_{Wmin}(u,v) - ave(P)|$ where $ave(P)$ is the average grayscale value over the $(2W+1) \times (2W+1)$ window centered at point P.

3. If $a > b$ then the W window tends to contain more low gray levels, in this case set $T = \alpha(\frac{2}{3}I_{Wmin}(u,v) + \frac{1}{3}ave(P))$. For text documents the parameter α is chosen to be between 0.3 to 0.8. Lower values are used for images with high noise and low contrast. For the grid images a much smaller value of 0.01 was used. The grid images contain far less contrast than typical degraded text documents and therefore a much smaller value for α is required.

4. If $a < b$ then the window tends to contain more local high gray levels, in this case set $T = \alpha(\frac{1}{3}I_{Wmin}(u,v) + \frac{2}{3}ave(P))$.

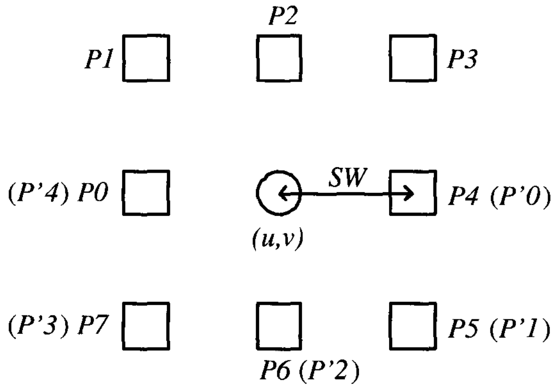


Figure 2.3: Neighborhood points involved with adaptive thresholding

5. If $a = b$ then repeat steps 1 to 5 using a new window size of $(2W + 3) \times (2W + 3)$. If still $a = b$ then $T = \alpha\, ave(P)$.

With the local threshold for P known, the final binary value is determined by simultaneously comparing P with four local averages at a distance of W centered around P. Referring to figure 2.3, the binary image element $B(u, v)$ is determined using:

$$B(u,v) = \begin{cases} 1 & \text{if } \bigvee_{i=0}^{3} \left[L(P_i) \wedge L(P'_i) \wedge L(P_{i+1}) \wedge L(P'_{i+1})\right] \text{ is true,} \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where $P'_i = P_{(i+4)mod8}$ for $i = 0 \ldots 7$ and $L(P) = (ave(P) - g(x, y) > T)$. The result of applying this algorithm to the image in figure 2.2 can be seen in figure 2.4, in this and all following images black=1, white=0.

2.2.2 Morphological post processing

With the adaptive thresholding results from figure 2.4 it is now possible to use morphological operators to tidy up the image. It is desirable to reconnect grid lines that have small breaks in them. The breaks are at areas in the original image where severe noise was present or at areas where the actual grid on the part was missing. The

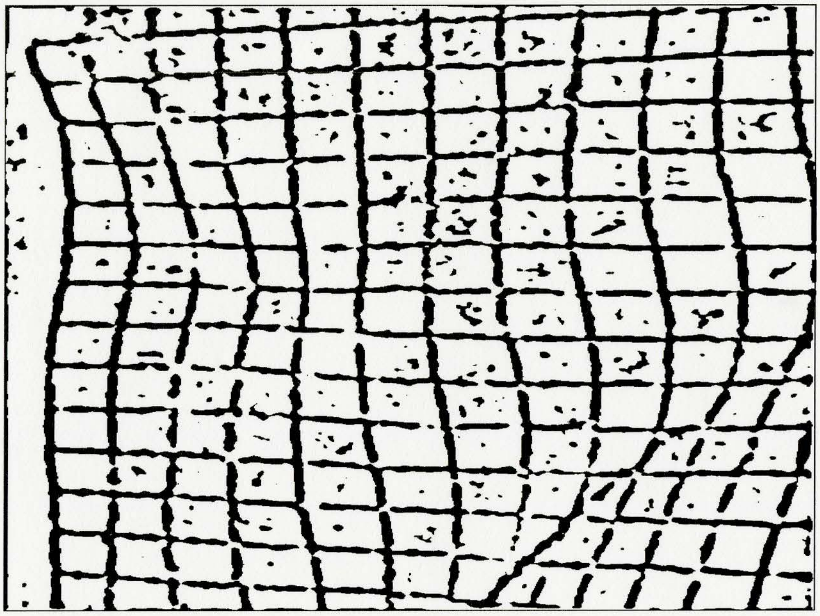


Figure 2.4: Result of adaptive thresholding

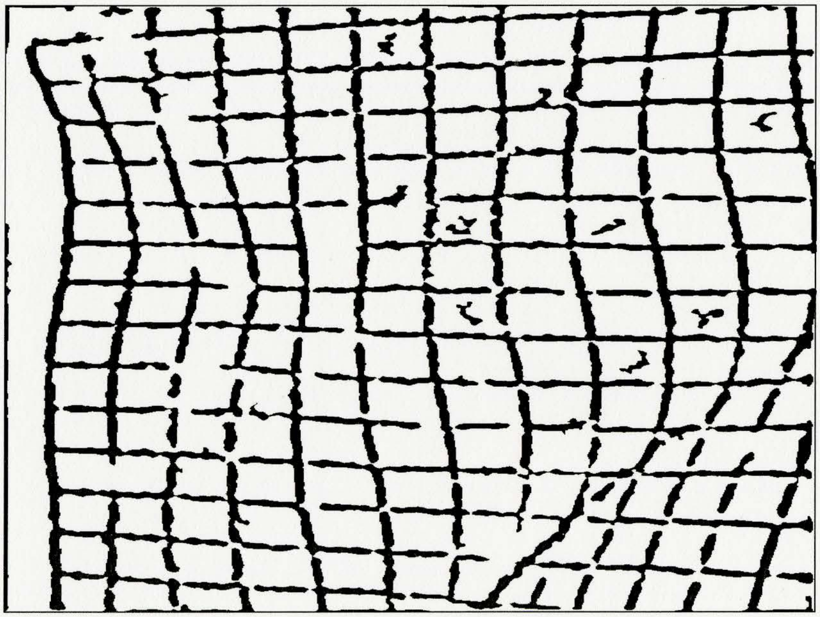


Figure 2.5: Removal of small connected regions

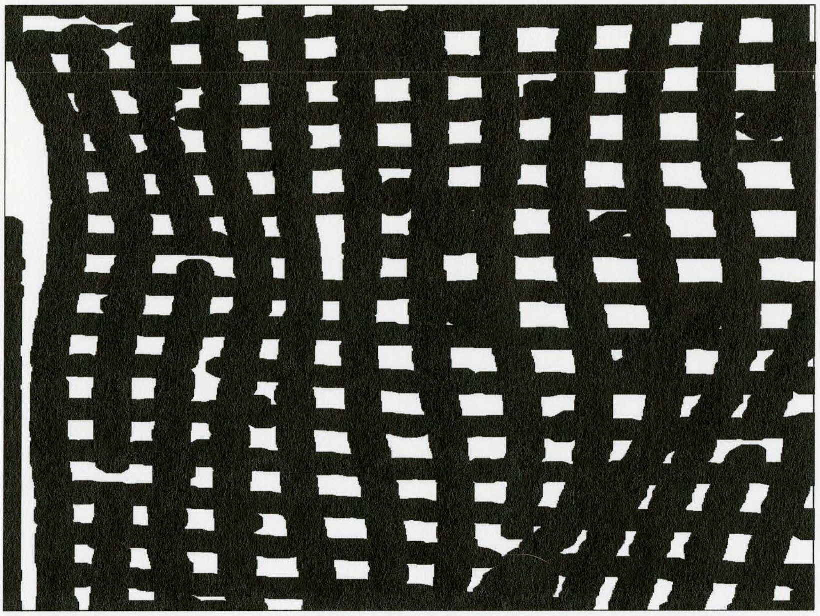


Figure 2.6: Dilation using circular disk

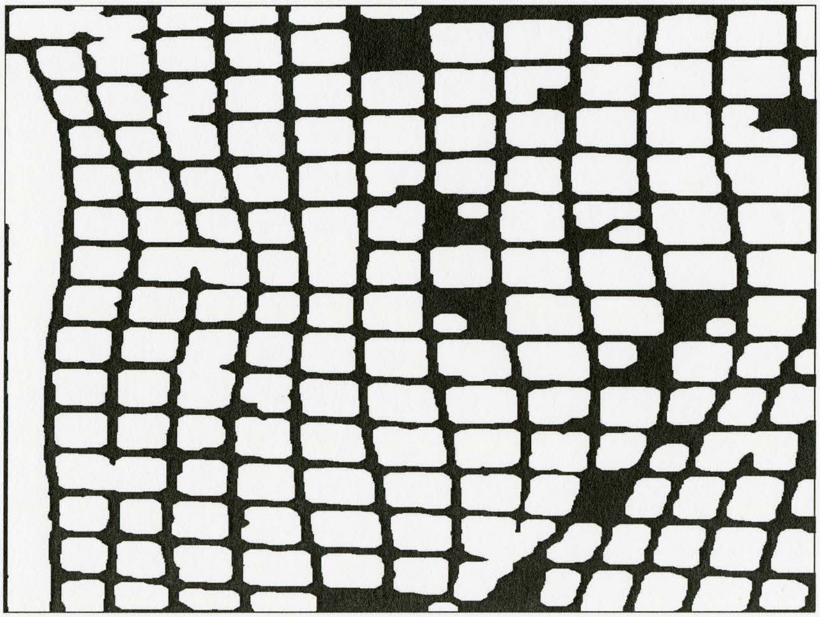


Figure 2.7: Erosion using circular disk

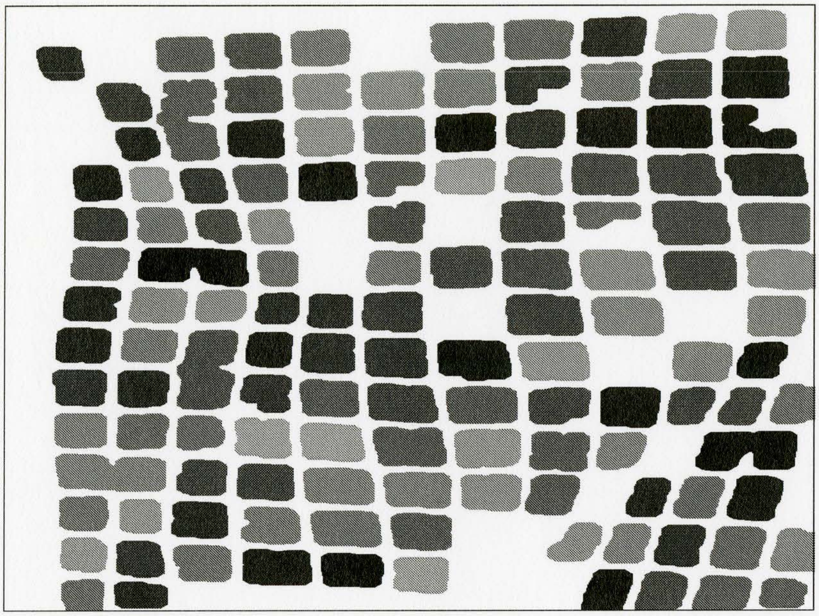


Figure 2.8: Removal of regions too large or too small

reconnection of grid lines can be done with a morphological dilation operation. A disk structured element can be applied that will expand the grid in a breadth wise manner. Before applying the dilation operation, it is important to remove any small connected components that are not part of the grid. If these are not removed, then a dilation operation would also enlarge them, potentially connecting them with the grid. The removal can be simply done by removing any 8-connected component with less than some threshold number of pixels. A value of 300 worked worked well, any larger removed parts of the grid. The thresholded grid image with unwanted components removed can be seen in figure 2.5. Unfortunately this process removes part of the grid in some areas, there is a trade off between removing unwanted components and removing parts of the grid. The dilation operation can now be applied to help relink grid lines. Using a circular disk element of radius 15, the result of a binary dilation operation can be seen in figure 2.6. Many of the broken grid lines are linked but the dilation has also widened the grid. To return it to its normal width, the grid lines must now be eroded using morphological erosion. Using the same circular disk as before, the results of binary erosion of the image can be seen in figure 2.7. The complement of this image gives an image of grid elements, rather than the grid lines. By assigning a unique label to each 8-connected component in the complement image, the grid elements can be segmented. Any of these grid elements that are less than 2000 or greater than 6500 in size are removed because grid element sizes should be within those bounds. The final labeled image consisting of N grid elements will be denoted as L and can be seen in figure 2.8. Each shade of gray within the image denotes a different grid element. This completes the image thresholding operation, the final result is an image of grid elements that are segmented from the surrounding image. The next step in the image processing procedure is to determine the approximate locations of grid intersections.

2.3 Grid Intersection Detection

Determining the approximate location of grid intersections can be accomplished by detecting the four corner vertices of the grid elements. Some of the grid elements are not of use because they have been deformed by the morphological processing. Determining these incorrect grid elements and detecting corner vertices for correct grid elements can be accomplished by using shape descriptors. The boundary of each grid element is a basic shape descriptor and can be determined by using a boundary following algorithm. The shape of the grid element can be discerned from the boundary as well as the four points of highest curvature which indicate the corner vertices.

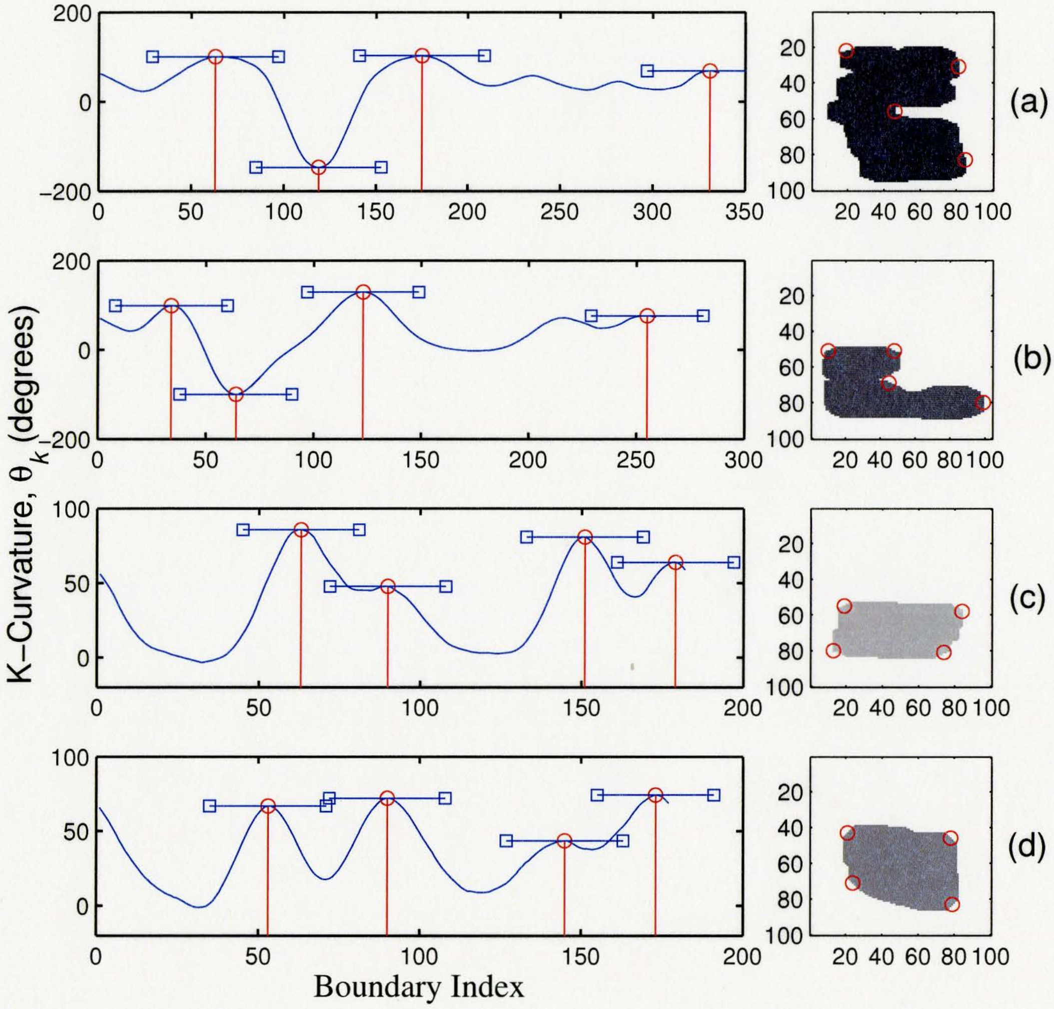


Figure 2.9: K-Curvature vs Boundary index for two incorrect grid elements (a),(b) and two correct grid elements (c),(d).

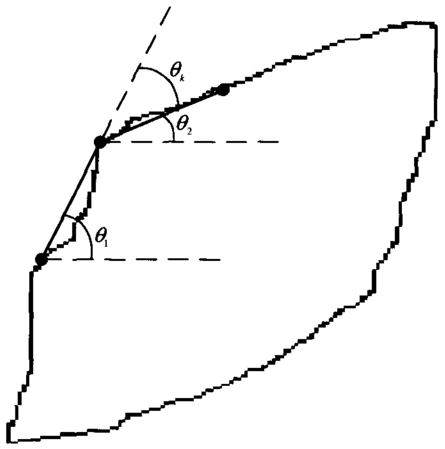


Figure 2.10: k-angles and k-curvature

2.3.1 Corner Vertex Detection using K-curvature

Using a boundary following algorithm, an ordered list describing the boundary of each region can be created. An efficient algorithm is presented in [36], the result is a clockwise list of boundary pixels contained within each region:

$$B^i = \{P^i_1, \ldots, P^i_{M^i}\}, \quad i = 0 \ldots (N-1) \tag{2.2}$$

where M_i is the number of boundary elements for grid element i and $P^i_j = (u_j, v_j)^i$. The k-curvature gives an indication of the curvature at a point on an objects boundary. The k-curvature for boundary list element j is the difference between two k-angles. The first k-angle is the direction vector angle for $(P_j - P_{(j-k)mod(M^i)})$, the second k-angle is the direction vector angle for $(P_{(j+k)mod(M^i)} - P_j)$. Figure 2.10 shows the geometry of the two k-angles, θ_1, θ_2 and the k-curvature, θ_k. The value of k should be large enough to ignore high frequency noise in the boundary but should not be so large as to give incorrect curvatures. Since the desired shape should have four sides that are similar in size, a good value for k is $M^i/10$. The k-curvature for the boundary point j on grid element i can be formally defined as:

$$\theta_k = \begin{cases} \theta_1 - \theta_2 & \text{if } |\theta_1 - \theta_2| \leq \pi \\ (\theta_1 - \theta_2) - sgn(\theta_1 - \theta_2)2\pi & \text{otherwise} \end{cases}, \tag{2.3}$$

where

$$\theta_1 = \arctan\left(\frac{v^i_j - v^i_{(j+k)mod(M^i)}}{u^i_j - u^i_{(j+k)mod(M^i)}}\right), \tag{2.4}$$

$$\theta_2 = \arctan\left(\frac{v^i_{(j+k)mod(M^i)} - v^i_j}{u^i_{(j+k)mod(M^i)} - u^i_j}\right), \tag{2.5}$$

$$sgn(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases}, \tag{2.6}$$

and arctan() is the four quadrant inverse tangent. The condition on θ_k prevents the direction angle from causing k-curvature discontinuities. A k-curvature angle greater than π, is only possible if the value k has been incorrectly chosen. The k-curvature is also smoothed along the boundary with an averaging window of size $\lfloor k/2 \rfloor + \lfloor k/2 \rfloor \, mod \, 2 + 1$. A plot of k-curvature vs boundary index for two incorrect and two correct grid element shapes can be seen in figure 2.9. The boundary index starts at the upper left pixel of each grid element. The circular markers are the detected points of largest absolute curvature and therefore should indicate the four vertices of the grid element. The largest point of absolute curvature is detected by comparison with all others. The second largest point of absolute curvature is detected by comparison with all boundary indices except for the ones within a distance of k from the first point of largest curvature. This process is repeated two more times to detect all four corner vertices. The lines with square bar ends in figure 2.9 indicate the k distance range from each corner vertex, it can seen that vertices should be more than k distance apart to prevent duplicates. After the detection process, the vertices are ordered based on curvature response, they need to be reordered based on boundary index, i.e. clockwise, for later processing steps. Incorrect grid elements

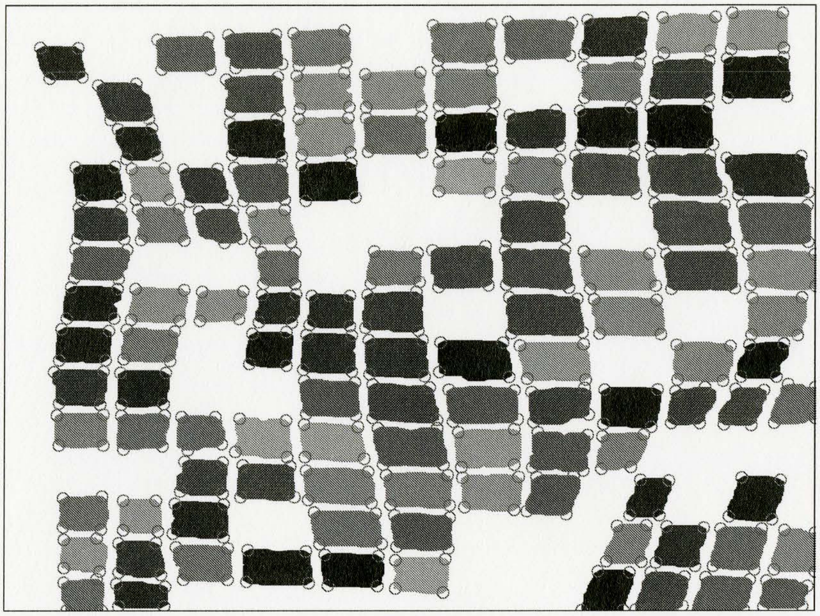
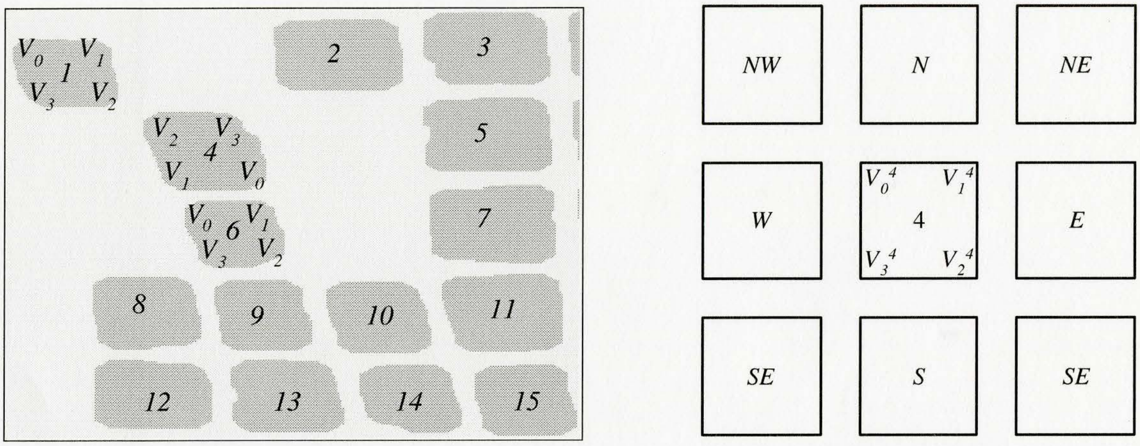


Figure 2.11: Final labeled image with incorrect shapes removed

can be rejected based on their k-curvature plots. Grid elements with any k-curvature values less than 20 degrees can rejected, it is assumed that grid elements are relatively convex in shape. The length of a grid element side is approximately proportional to the number of boundary elements between the vertices that define the side. Any grid element that has a side that is significantly larger than the mean length of sides for that grid element can also rejected. Figure 2.11 shows the results of eliminating grid elements that posses large negative θ_k values or have unequal sides. This method of detecting vertices and rejecting incorrectly shaped grid elements is invariant to changes of rotation, translation and size.

2.3.2 Determining Grid Structure

Determining whether two grid elements are adjacent can be determined by checking the proximity of their corner vertices. Using this observation, a grid structure can



(a) Labeled Grid Image

(b) Grid Structure

Figure 2.12: Grid structure

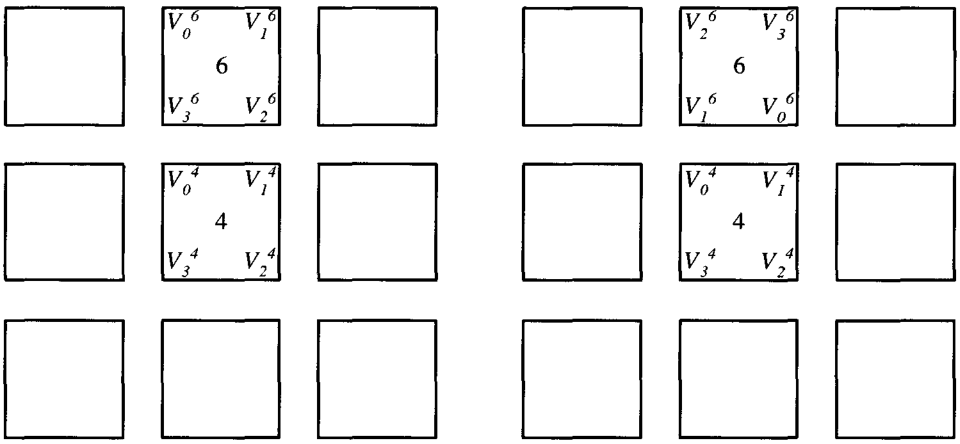
be sequentially formed. Starting with an initial grid element, neighbouring grid elements can be added with their position determined by which corner vertices are close together. This grid structure is used to determine grid intersections and is also used later as the basis for stereo matching.

Determining the grid structure will be described using figure 2.12(a), this cropped image is the top left quadrant of figure 2.11. In this figure the corners of grid elements 1, 4 and 6 are labeled with their corner vertices. The corner vertex lists are ordered in a clockwise direction around the grid element and can be written as $L^m = \{V_0^m, V_1^m, V_2^m, V_3^m\}$, where m is the grid element. The first step in determining the grid structure of figure 2.12(a) is to choose an initial grid element. Any grid element can be used, so 4 is chosen, it now defines the orientation of the entire grid structure. The top left corner of grid element 4 in the grid structure is given by the first corner vertex in the list L^4, the top right corner by the second element in L^4, bottom right by the third element in L^4 and bottom left by the last element in L^4. With this ordering, the structure of figure 2.12(a) will be recreated but with a rotation of 180°. The north direction for grid element 4 is defined by $\{V_0^4, V_1^4\}$, NE by

V_1^4, E by $\{V_1^4, V_2^4\}$, SE by V_2^4 and so on, figure 2.12(b) depicts the situation. Using these definitions, a grid element which has two vertices adjacent to $\{V_0^4, V_1^4\}$ can be placed in the N position relative to grid element 4, and in this way the grid structure can be built. Adjacency between vertices V_i^m and V_j^n can be formally stated using the following Boolean check:

$$dist(V_i^m, V_j^n) = \begin{cases} 1 & \text{if } \sqrt{(u_i^m - u_j^n)^2 + (v_i^m - v_j^n)^2} < D \\ 0 & \text{otherwise,} \end{cases} \tag{2.7}$$

where D is a threshold distance. A D value of 30 pixels was appropriate for grid images. It can be seen from figure 2.12(a) that with an initial grid element of 4, expression 2.7 will be true for grid elements 1 and 6. Firstly considering grid element 6 and incrementing both i and then j from 0 to 3, expression 2.7 is first true for (V_0^4, V_1^6) and then true for (V_1^4, V_0^6), therefore grid element 6 should be placed in a N position relative to grid element 4. Two grid elements that have two corner vertices adjacent will be referred to as *2-adjacent.* Figure 2.13(a) shows the new grid structure containing both 4 and 6. As was the case before, the list L^6 is assumed to start at the top left corner of the grid element in the grid structure, but this is not currently the case because the grid structure orientation was defined by grid element 4. To ensure that placement within the grid structure is continued correctly L^6 must undergo a right circular shift. The correct right shift for a grid element n being entered into the grid structure is given by $(j-i+5)\,mod 4$ where i,j are from the first true value found for $dist(V_i^m, V_j^n)$ and m is the grid element already present within the grid structure. Using this expression, L^6 can be right shifted by $(0-1+5)\,mod 4 = 2$, this correction to the vertex ordering can be seen in figure 2.13(b), the vertices of grid element 6 are now correctly aligned within the grid structure. Grid element 1 could be also be added to the grid structure in a similar manner. When considering the adjacency of grid elements 1 and 4, expression 2.7 is only true for (V_2^4, V_2^1), therefore grid element 1 could be added to the SE position with respect to grid element 4 and a circular



(a) Before right shift

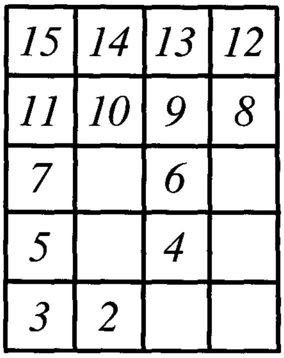
(b) After right shift

Figure 2.13: Grid structure before and after right shift of vertices

shift could then be applied to L^1. In practice it is not advisable to enter grid elements based on just one vertex adjacency. Sometimes grid elements that should have two vertices that are adjacent in fact only have one. This is caused by noise in the grid and the effects of the morphological processing. A grid element that is incorrectly placed would have detrimental effects later on when the grid is used to do automated stereo matching. For this reason a grid element is only entered into the grid structure if it is *2-adjacent* to an element already in the grid structure. The final grid structure for the image in 2.12(a) can be seen in figure 2.14, note that grid element 1 has not been entered because it is not *2-adjacent* to any other grid element. Building the grid can be efficiently accomplished using a First In First Out (FIFO) queue to store grid element numbers. The first grid element number, 4 in the example above, is entered into the FIFO queue to initialize. Each time a grid element label is removed from the queue, its *2-adjacent* neighbors are determined and added to the grid structure, as long as they are not already present. The neighbors that were just added to the grid structure are then added to the FIFO queue and the next queue element is processed. This grows the grid structure in a breadth wise manner.

There will be grid elements that are not adjacent to any others, it is also possible that the grid will be split into two or more pieces. This is why it is necessary to repeat the grid structure determination process until all grid elements in the image have been checked. A grid element not adjacent to any grid element already in one of the grid structures will be used to start a new grid structure. If there is more than one grid structure created, which is often the case, the grid structure with the largest number of elements is retained, all others are ignored. Although this will waste grid elements, it is rare that the grid is largely divided. Usually only one or two grid elements are alone, the rest being adjacent in a large grouping.

Using the grid structure shown in figure 2.14 it is possible to calculate the grid intersection points. Each grid intersection point can be estimated as the centroid of four vertices from four adjacent grid elements. This gives an estimate of where the grid intersection point is located but it is not very accurate. The reason being that the corner vertices were determined from a processed image. Looking at figure 2.7 it can be seen that the morphological dilation and erosion processes have rounded the grid element corners. The corner vertices will always be located at considerable different locations from the corners on the original image. This error in the vertex locations would introduce a large error when using four adjacent vertices to calculate a centroid. For this reason the image processing steps that have been presented so far are just used to determine the grid structure and provide estimates for the grid intersections. The final grid intersections will be calculated using a sub pixel centerline detection algorithm and parabola fitting on original grayscale image data. The sub pixel edge detector can give erroneous point data near grid intersections, for this reason the algorithm is not applied to the entire original image but only to specific grid line regions, these regions are determined in the next section.



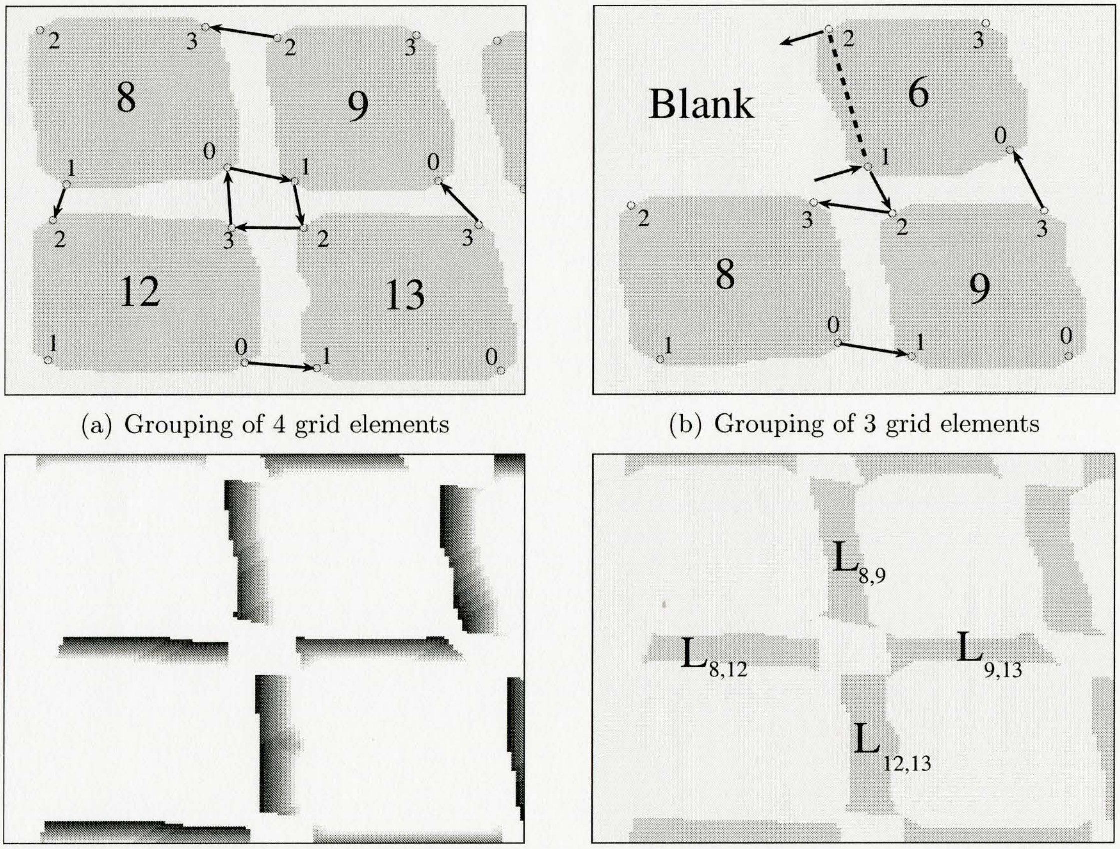
15	*14*	*13*	*12*
11	*10*	*9*	*8*
7		*6*	
5		*4*	
3	*2*		

Figure 2.14: Final grid structure

2.3.3 Determining Grid Line Regions

A method of determining the points along the center of the grid lines to sub pixel accuracy is presented in the next section. This section focuses on determining the small grid line regions between *2-adjacent* grid elements that the sub pixel line detector will operate on. The grid structure can be used to determine sets of 3 and 4 grid elements that define a detectable grid intersection point. For example, two detectable grid intersections from the grid structure in figure 2.14 are $\{9, 8, blank, 6\}$ and $\{8, 9, 13, 12\}$. The grid line regions between the *2-adjacent* grid elements within these sets can be used to define where parabolas should be fitted. For example, points along the center of the grid line between pairs $\{8, 9\}$ and $\{12, 13\}$ can be used to fit one parabola. Points along the center of the grid line between the pairs $\{8, 12\}$ and $\{9, 13\}$ can be used to fit another parabola. The intersection of these two parabolas is the grid intersection point. It is also desirable to determine points on the grid line for the pair $\{blank, 6\}$. Even though only one grid element for this pair is present in the grid structure, the grid line still exists in the image and therefore the center line points can be found.

The grid line regions between *2-adjacent* grid elements can be found using the corner vertices of the grid elements and a localized distance transform. Figure 2.15(a)



(a) Grouping of 4 grid elements

(b) Grouping of 3 grid elements

(c) Localized Distance transform on bounded areas (*DIST*)

(d) Labeled grid line regions (*LABEL*)

Figure 2.15: Determination of grid line areas

shows the grid elements $\{8, 9, 13, 12\}$ with additional boundary rays, the vertices are labeled according to their vertex lists. The boundary rays are defined using adjacent corner vertices. In the case where only one grid element is available, the two bounding rays each consist of a corner vertex and are perpendicular to the line defined by the two corner vertices. For example, the set $\{blank, 6\}$ from figure 2.15(b) will have the two bounding rays defined with the points $(V_1^6, V_1^6 - \overrightarrow{V_2^6V_1^{6\perp}})$ and $(V_2^6 + \overrightarrow{V_2^6V_1^{6\perp}}, V_2^6)$, where $\overrightarrow{V_2^6V_1^{6\perp}} = (v_2^6 - v_1^6, u_1^6 - u_2^6)$. Using one of the grid elements as the object, a distance transform can be applied where the bounding rays enclose the points of interest. Any points that are between the ray boundaries, within a distance of W and are not labeled in the image L (i.e. not part of any grid element), define the grid line area. The value of W is as before, the assumed width of the grid.

This method of extracting the grid line regions ignores the areas around the grid intersections. This is purposely done because the sub pixel edge detector will not always produce the correct center line in this area. This area will also contain sub pixel center line points from the perpendicular grid line, these outlier points would cause a large error when least squares parabola fitting was applied. The distance transform is only necessary because of cases such as $\{blank, 6\}$, where only one grid element is available, but for convenience it is applied in all cases.

Determining whether a point is left of a bounding line can be done with the use of the $area2$ function:

$$area2 = \frac{(u_b - u_a)(v_c - v_a)}{(u_c - u_a)(v_b - v_a)}, \tag{2.8}$$

where $a = (u_a, v_a)$, $b = (u_b, v_b)$, and the function returns negative if the point $c = (u_c, v_c)$ is on left of the vector $\overrightarrow{ab}$, positive if on the right and zero if coincident. A typical distance transform algorithm such as that presented by Borgefors [21] could be applied but after processing it would require a step to search for points within the boundaries. The proposed localized distance transform is applied to the boundary points from one of the grid elements in a *2-adjacent* pair, either can be chosen. For example, the boundary points used for the set {8, 9} from figure 2.15(a) will be:

$$B^8_{(0\to 3)} = \{P^8_i, \dots, P^8_j\} \tag{2.9}$$

where within the circular boundary list i and j are the indices of V^8_3 and V^8_0 respectively. The localized distance transform algorithm uses a FIFO queue which is initialized with the boundary points. The algorithm uses a temporary image called *DIST*, initialized with zeros, to store distance values. The regions labels themselves are stored in another image called *LABEL*. Pseudocode for the detection of one grid line region which will be labeled with the value L is presented below. In the code, the bounding rays $(a1, b1)$ and $(a2, b2)$ are assumed to correspond to the appropriate vertices as shown in figure 2.15(a). After applying the algorithm to the grid elements from figure 2.15(a) the *DIST* and *LABEL* images can be seen in figures 2.15(c) and 2.15(d) respectively. Expanding the grid structure as shown in figure 2.16 allows the accommodation of grid line region labels. The next section gives details on a sub pixel line detection algorithm which is applied to the detected grid line regions.

Algorithm 1 Localized distance transform

```
while $QueueSize > 0$ do
    $P \leftarrow QueueRemove$
    for all $Q$ that are 8-CONN neighbours of $P$ do
        if $((area2(a1, b1, Q) >= 0) \wedge (area2(a2, b2, Q) >= 0) \wedge (L(Q_u, Q_v) = 0) \wedge (DIST(P_u, P_v) < W))$ then
            $LABEL(Q) \leftarrow L$
            $DIST(Q) \leftarrow DIST(P) + 1$
            $QueueAdd \leftarrow Q$
        end if
    end for
end while
```

15		*14*		*13*	$L_{13,12}$	*12*
				$L_{13,9}$	P_1	$L_{8,12}$
11		*10*	$L_{10,9}$	*9*	$L_{9,8}$	*8*
		L_{10}	P_2	$L_{9,6}$		
7			L_6	*6*		
5				*4*		
3		*2*				

Figure 2.16: Expanded grid structure to accommodate grid line region labels

2.4 Sub Pixel Grid Detection

To obtain the most accurate results, sub pixel detection of the original grayscale image grid lines is necessary. In fact the sub pixel line detector is only applied to the centerline regions of the original image. These centerline regions are the areas determined in the previous section.

There are many edge detectors available but a recent method developed by Steger [12], [13] is suited to detecting grid lines. It can be tuned to look for lines of specific width and the colour of detected lines can be specified, bright lines on dark background or dark lines on light background. The determination of sub pixel accuracy for a One Dimensional (1-D) line profile will be presented in the next section. The following section will apply the results of the 1-D line detection to the detection of 2-D lines.

2.4.1 1D Line Detection

The perpendicular cross section of lines can be modeled by a 1-D curve with an asymmetrical bar shaped model:

$$f_a(x) = \begin{cases} 0, & x < -w \\ 1, & |x| \le w \\ a, & x > w \end{cases}, \tag{2.10}$$

where $2w = W$ is the width of the line and $a \epsilon [0, 1]$. This is generally more applicable for image edges because contrast on either side of lines is rarely equal. The basis for the technique is to detect line positions by convolving the image with the derivatives of a Gaussian smoothing kernel. Using a Gaussian to smooth the image reduces the effect of high frequency noise normally problematic in edge detection. The Gaussian smoothing kernel and its derivatives are given by:

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \tag{2.11}$$

$$g'_\sigma(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}}, \tag{2.12}$$

$$g''_\sigma(x) = \frac{x^2 - \sigma^2}{\sqrt{2\pi}\sigma^5} e^{-\frac{x^2}{2\sigma^2}}. \tag{2.13}$$

The responses of the Gaussian kernels to the asymmetrical bar line model are:

$$r_a(x, \sigma, w, a) = \phi_\sigma(x + w) + (a - 1)\phi_\sigma(x - w), \tag{2.14}$$

$$r'_a(x, \sigma, w, a) = g_\sigma(x + w) + (a - 1)g_\sigma(x - w), \tag{2.15}$$

$$r''_a(x, \sigma, w, a) = g'_\sigma(x + w) + (a - 1)g'_\sigma(x - w), \tag{2.16}$$

where

$$\phi_\sigma(x) = \int_{-\infty}^{x} e^{-\frac{t^2}{2\sigma^2}} dt. \tag{2.17}$$

The strength of the line as well as the line colour is given by r''_a, bright lines on dark background are indicated by $r''_a(x, \sigma, w, a) < 0$ and dark lines on bright background are indicated by $r''_a(x, \sigma, w, a) > 0$. The strength of the line is given by $|r''_a(x, \sigma, w, a)|$, where it has been shown [12] that the for the asymmetrical bar shaped lines profile, the maximum response of $|r''_a(x, \sigma, w, a)|$ will occur when:

$$\sigma = \frac{w}{\sqrt{3}}. \tag{2.18}$$

Therefore, the detector can be tuned to a specific line width by choosing an appropriate value for σ.

The center of the line is where the ridges or ravines of the image occur or where $r'_a(x, \sigma, w, a) = 0$. Setting equation 2.15 to zero and solving for the line position l gives:

$$l = -\frac{\sigma^2}{2w} \ln(1 - a). \tag{2.19}$$

From equation 2.19 it can be seen that $a > 0$ causes a bias, i.e. the line is detected off center. However, Steger [12] shows that if the width w and the contrast difference a is known, equation 2.19 can be used to remove the bias from the line. The correction method involves detecting the nearest line edges to each center line point, one on each side, and then adjusting the center line point accordingly.

Up to this point only continuous functions have been used, for the discrete 1D line, $f(x) = f_n$ for $x \epsilon (n - 1/2, n + 1/2]$, the Gaussian convolution masks are given by:

$$g_{n,\sigma} = \phi_\sigma(n + 1/2) - \phi_\sigma(n - 1/2) \tag{2.20}$$

$$g'_{n,\sigma} = g_\sigma(n + 1/2) - g_\sigma(n - 1/2) \tag{2.21}$$

$$g''_{n,\sigma} = g'_\sigma(n + 1/2) - g'_\sigma(n - 1/2), \tag{2.22}$$

where the Gaussian mask size is chosen to give an error of less than 10^{-4}, acceptable for a gray level range of [0, 255]. In the discrete case, detection of the line center to sub pixel accuracy can be determined using the second order Taylor polynomial given by:

$$p(x) = r + r'x + \frac{1}{2}r''x^2. \tag{2.23}$$

The position of the line is where $p'(x) = 0$, or :

$$x = -\frac{r'}{r''}. \tag{2.24}$$

A point is only declared valid if $x \epsilon [-1/2, 1/2]$ and r'' gives the desired color and $|r''|$ is above a user defined threshold.

2.4.2 2D Line Detection

Lines in 2D are modeled as curves that exhibit the line profile 2.10 in the perpendicular direction. If this line is called $n(t)$, then at the center of the line, the first directional derivative of $n(t)$ should vanish and the second directional derivative of $n(t)$ should be of large absolute value. Determining the direction of $n(t)$ for each point within the image can be done using the partial derivatives r_u, r_v, r_{uu}, r_{uv} and r_{vv}, which are estimated by convolution of the image with the appropriate 2-D Gaussian partial derivative kernels. The 2-D convolution can be performed by first convolving the image with a 1-D Gaussian in the u direction, and then convolving with another 1-D Gaussian in the v direction. For example, r_u is determined by convolving the image with the 1-D first derivative Gaussian $g'_\sigma(u)$, and then convolving this result with the 1-D smoothing Gaussian derivative given by $g_\sigma(v)$.

Just as in the 1-D case, the line strength is determined by the value of the second directional derivative. The direction in which the second directional derivative takes on its maximum value can be determined by calculating the eigenvalues and eigenvectors of the Hessian matrix:

$$H(u, v) = \begin{bmatrix} r_{uu} & r_{uv} \\ r_{uv} & r_{vv} \end{bmatrix} \tag{2.25}$$

The eigenvector corresponding to the eigenvalue of maximum absolute value is given by (n_u, n_v), where $||(n_u, n_v)|| = 1$. As in the 1-D case, a Taylor polynomial is used to determine whether the first directional derivative along (n_u, n_v) vanishes within the current pixel. The point (p_u, p_v) is determined by inserting $(p_u, p_v) = (tn_u, tn_v)$ into the Taylor polynomial 2.23 and then setting its derivative along t to zero, giving:

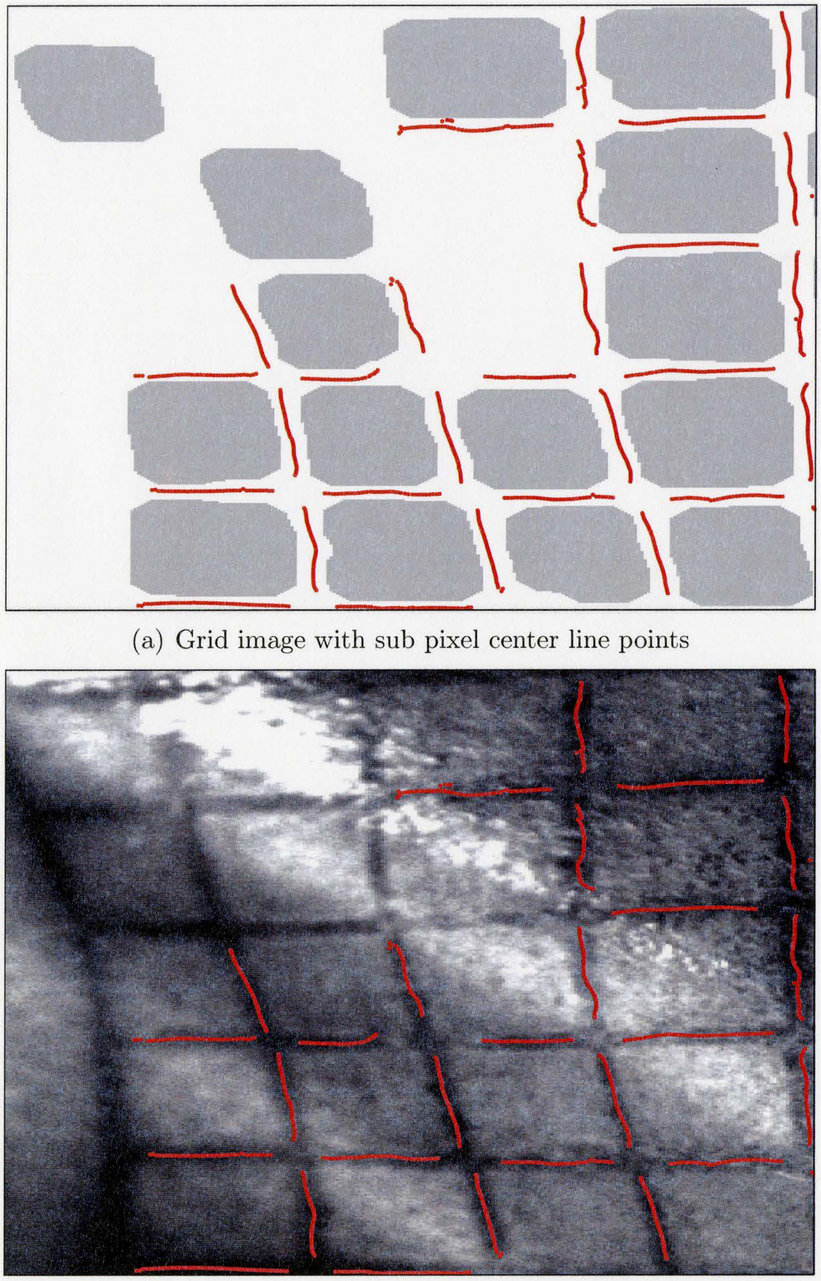
$$t = -\frac{r'}{r''} = -\frac{r_u n_u + r_v n_v}{r_{uu} n_u^2 + 2r_u v n_u n_v + r_v v n_v^2} \, . \tag{2.26}$$

As before in the 1-D case, the point is only declared valid if $(p_u, p_v) \epsilon [-1/2, 1/2] \times [-1/2, 1/2]$ and $|r''|$ is above the threshold value. Points are linked using the hysteresis technique commonly found in other edge detection algorithms.

This line detection technique gives the center of the grid line to sub pixel accuracy with at most one point for each pixel. It has also been shown by Steger [11], that the sub-pixel accuracy on real images is better than one tenth of a pixel for the majority of points. The sub pixel line detection is only applied to the grid line regions determined in the previous section. The centerline points are stored in lists, each label L_{ij} in the grid structure shown in figure 2.16 is associated with a list of centerline points. The results of applying this algorithm to the cropped grid shown in figure 2.12(a) can be seen in figure 2.17(a). The sub pixel points are also overlayed for the same area on the original image in figure 2.17(b). The results for the original image shown in figure 2.2 can be seen in figure 2.18.

2.5 Parabola Fitting

The grid structure contains all the information needed to do parabola fitting. For example, from the grouping $\{8, 9, 12, 13\}$ in figure 2.16, a parabola can be fitted to the points associated with $L_{13,9}$ and $L_{8,12}$ and another parabola fitted to points associated with $L_{13,12}$ and $L_{9,8}$. The intersection of these two parabolas gives the grid intersection point to sub pixel accuracy.



(a) Grid image with sub pixel center line points

(b) Labeled image with sub pixel center line points

Figure 2.17: Sub pixel center line points

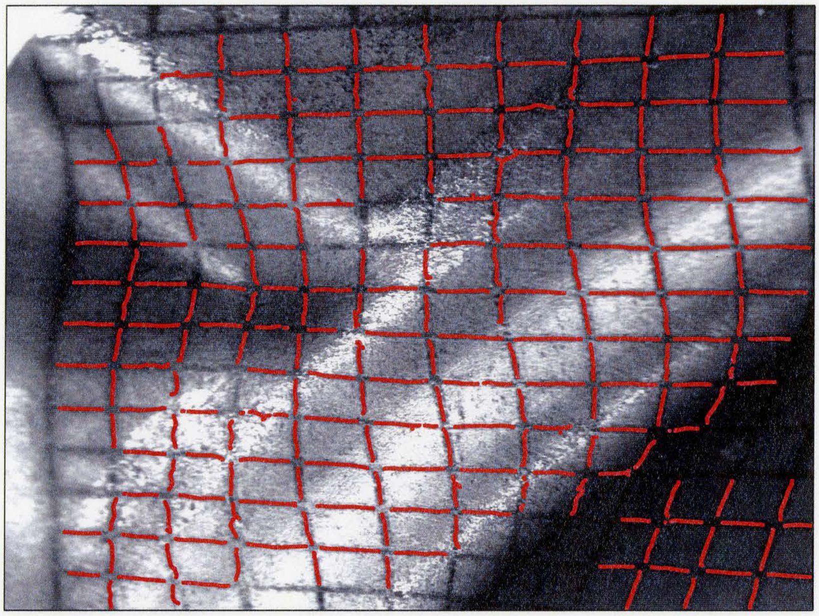


Figure 2.18: Grid Image with sub pixel points onlayed

A parabola can be written in a xy local coordinate system as

$$y^2 = 2gx, \tag{2.27}$$

where g is the focus distance. For free form fitting, the xy parabola coordinate system must be able to rotate and translate within the global XY coordinate system. This is done with the use of an origin $P_c = (X_c, Y_c)$ and a rotation angle α. The rotation is in a counter clockwise direction measured from the x-axis and can be written as a rotation matrix using:

$$R = \begin{bmatrix} C & -S \\ S & C \end{bmatrix} \tag{2.28}$$

where $S = \sin(\alpha)$, $C = \cos(\alpha)$. A point P within the XY global coordinate system can be transformed into the xy local coordinate system with:

$$p = R(P - P_c), \tag{2.29}$$

where $p = (x, y)$. The reverse transformation can also be written:

$$P = R^T p + P_c. \tag{2.30}$$

A parabola is a conic section and can also therefore be written using the standard conic form:

$$0 = p^T A p + B^T p + c \tag{2.31}$$

$$0 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_0 & b_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + c \tag{2.32}$$

where

$$\begin{aligned} a_{00} &= \sin^2 \alpha, \\ a_{01} = a_{10} &= -\sin\alpha\cos\alpha, \\ a_{11} &= \cos^2 \alpha, \\ b_0 &= -2(x_c \sin^2 \alpha + y_c \sin\alpha\cos\alpha - g\cos\alpha), \\ b_1 &= 2(x_c \sin\alpha\cos\alpha - y_c \cos^2\alpha - g\sin\alpha), \\ c &= (x_c \sin\alpha - y_c\cos\alpha)^2 + 2g(x_c\cos\alpha + y_c\sin\alpha). \end{aligned} \tag{2.33}$$

2.5.1 General Least Squares Parabola Fitting

When using general least squares fitting the coordinate system orientation must be fixed, therefore the rotation angle cannot be found. Points that are to be fitted have to be transformed into the parabola coordinate system using an estimate for the rotation angle. For the grid image, the angle can be estimated using the grid structure and corner vertices. Figure 2.19 shows 4 grid elements with the global coordinate system. Using the corner vertices $\{V_3^a, V_2^b, V_1^c, V_0^d\}$ the rotation angle is estimated as:

$$\alpha = \arctan\left(\frac{(v_3^a + v_0^d) - (v_2^b + v_1^c)}{(u_3^a + u_0^d) - (u_2^b + u_1^c)}\right) + \frac{\pi}{2}, \tag{2.34}$$

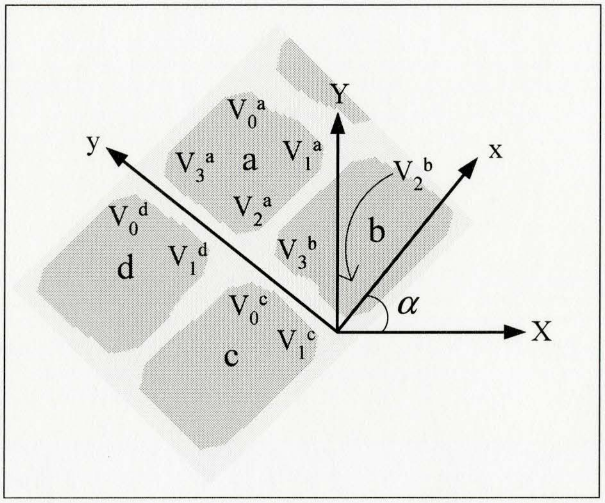


Figure 2.19: Estimation of α using corner vertices

where arctan() is the four quadrant inverse tan. If one of the grid elements such as a is missing then the angle is calculated using just $\{V_1^c, V_0^d\}$. The points to be fitted are transformed into the parabola coordinate system using equation 2.29 with $P_c = (0, 0)$. With an estimate for α, general least squares fitting can be used to solve for the focus distance the origin (x_c, y_c) in the parabola coordinate system. This origin can be added to the parabola as:

$$(y - y_c)^2 = 2g(x - x_c), \tag{2.35}$$

Expanding this and solving for x produces a polynomial function with 3 terms:

$$\begin{aligned} x &= \left(\frac{y_c^2}{2g} + x_c\right) - \left(\frac{y_c}{g}\right) y + \left(\frac{1}{2g}\right) y^2 \\ &= a_0 + a_1 y + a_2 y^2 \end{aligned} \tag{2.36}$$

Least squares algebraic fitting can be applied to solve for the three unknowns in equation 2.36, the SVD technique presented in [44] was used for this work. The

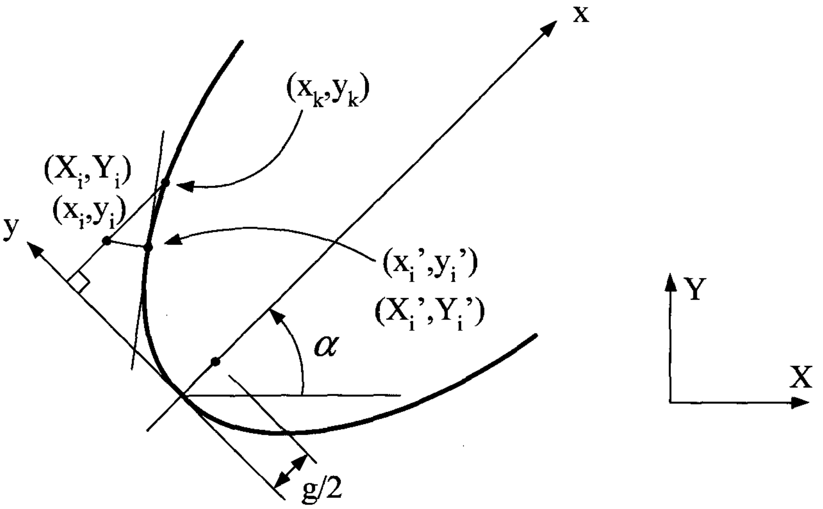


Figure 2.20: Geometric parabola fitting technique

origin in the global coordinate system and the focal length are then calculated using:

$$P_c = R^T \left[\begin{array}{cc} (a_0 - a_1^2/(4a_2)) & -a_1/(2a_2) \end{array} \right]^T \tag{2.37}$$

$$g = 1/(2a_2) \tag{2.38}$$

The general least fitting does not allow for any change in rotation and also only minimizes the algebraic distance between data points and the parabola. The geometric fitting presented in the next section solves these problems, producing a refined fit.

2.5.2 Geometric Parabola Fitting

Geometric parabola fitting allows free motion of the parabola within the coordinate system. A method of fitting free form parabolas proposed by Sung et al. [26] was implemented. The method performs fitting in a two step process. The first step is to find the orthogonal contacting points along the current parabola corresponding to each data point. These points will give the shortest geometric distance to the parabola and indicate the true fitting error. The second step in the process is to use these orthogonal distances to evaluate the current parabola parameters. The minimization of functions to find the orthogonal contacting points and the parabola parameters is done using the Gauss-Newton optimization algorithm. The main optimization routine attempts to find the best parabola parameters based on the orthogonal error. Within each iteration of this optimization, another optimization routine determines the orthogonal contact point and therefore orthogonal error for each data point. The Jacobian matrices used for the Gauss-Newton optimization are described below.

The Jacobian matrix for the orthogonal contacting point problem can be described using contacting conditions. Using equation 2.29, all (X, Y) points can be transformed into the xy parabola coordinate system. Each point (x_i, y_i) has an orthogonal contacting point (x_i', y_i') as indicated in figure 2.20. The line connecting these two points and the tangent line to the parabola at the orthogonal contacting point should be perpendicular:

$$(x_i - x, y_i - y) \cdot (g, y) = 0. \tag{2.39}$$

This condition and the fact that the orthogonal contact point should lie on the parabola can be used to write two orthogonal contact point conditions:

$$f_1(x, y) = \frac{1}{2}(y^2 - 2gx) = 0, \tag{2.40}$$

$$f_2(x, y) = y(x_i - x) + g(y_i - y) = 0. \tag{2.41}$$

Equations 2.40 and 2.41 can be solved simultaneously using Gauss-Newton optimization and the Jacobian matrix:

$$Q = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} -g & y \\ -y & (x_i - x - g) \end{bmatrix}. \tag{2.42}$$

Sung et al. [26] suggest an initial orthogonal contact point guess x_k as:

$$x_k = \begin{cases} 0, & \text{if } x_i < 0 \\ \begin{bmatrix} x_i & sgn(y_i)\sqrt{2gx_i} \end{bmatrix}^T & \text{if } x_i \geq 0 \end{cases}, \tag{2.43}$$

but the parabolas in the grid image are much larger in the y direction, therefore the following initial guess was used:

$$x_k = \begin{bmatrix} y_i^2/(2g) & y_i \end{bmatrix}^T. \tag{2.44}$$

At each iteration of the parabola fitting, the Jacobian given by 2.42 and the initial guess given by 2.44 are used to minimize the orthogonal distance for each data point.

After a given point P_i in the XY system is transformed into x_i and the orthogonal contact point x_i' is found, the point X_i' is determined through the backward transformation given in equation 2.30. If the parameters of the parabola are arranged as $a = (X_c, Y_c, g, \alpha)$ then it can be shown [26] that the Jacobian for the orthonormal point X_i' is given by:

$$J_{X_i',a} = R^{-1}\frac{\partial x}{\partial a}\bigg|_{x=x_i'} + \begin{bmatrix} 1 & 0 & -xS & yC \\ 0 & 1 & xC & -yS \end{bmatrix}\Bigg|_{x=x_i'} \tag{2.45}$$

where

$$\frac{\partial x}{\partial a} = Q^{-1}\begin{bmatrix} 0 & 0 & x & 0 \\ (yC - gS) & (yS + pC) & (y - y_i) & (-yy_i + gx_i) \end{bmatrix} \tag{2.46}$$

and Q^{-1} is the Jacobian matrix from equation 2.45 for the current orthogonal point. With m data points, the Jacobian given in equation 2.45 can be used to form $2m$ linear equations:

$$\begin{bmatrix} J_{X_i',X_c} & J_{X_i',Y_c} & J_{X_i',g} & J_{X_i',\alpha} \\ J_{X_i',X_c} & J_{X_i',Y_c} & J_{X_i',g} & J_{X_i',\alpha} \\ \vdots & \vdots & \vdots & \vdots \\ J_{X_m',X_c} & J_{X_m',Y_c} & J_{X_m',g} & J_{X_m',\alpha} \\ J_{X_m',X_c} & J_{X_m',Y_c} & J_{X_m',g} & J_{X_m',\alpha} \end{bmatrix} \begin{bmatrix} \Delta X_c \\ \Delta Y_c \\ \Delta g \\ \Delta\alpha \end{bmatrix} = \begin{bmatrix} X_c'' \\ Y_c'' \\ g'' \\ \alpha'' \end{bmatrix} \tag{2.47}$$

where $X_i'' = X_i - X_i'$. This system can be solved using iterative Gauss-Newton optimization.

The results of parabola fitting for a few grid intersections can be seen in the right images of figure 2.21.

2.5.3 Parabola Intersection

The intersection of two parabolas can be found using the Bézout determinant. Given two polynomials $f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$ and $g(x) = \beta_0 + \beta_1 x + \beta_2 x^2$ the following system of equations can be written:

$$\begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & 0 \\ 0 & \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_0 & \beta_1 & \beta_2 & 0 \\ 0 & \beta_1 & \beta_2 & \beta_3 \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.48}$$

where two of the rows were created by multiplying the equations by x. From equation 2.48 the Bézout determinant is given by:

$$\begin{vmatrix} \alpha_0 & \alpha_1 & \alpha_2 & 0 \\ 0 & \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_0 & \beta_1 & \beta_2 & 0 \\ 0 & \beta_1 & \beta_2 & \beta_3 \end{vmatrix} = (\alpha_0\beta_2 - \alpha_2\beta_0)^2 - (\alpha_0\beta_1 - \alpha_1\beta_0)(\alpha_1\beta_2 - \alpha_2\beta_1) \tag{2.49}$$

For the two equations to have a common root, the Bézout determinant must equal zero. Two parabolas $Q_{(0)}(x, y)$ and $Q_{(1)}(x, y)$ that are of the form given in equation 2.32 can be rearranged and treated as second degree polynomials where the coefficients are polynomials in y:

$$Q_i(x, y) = (a_{11}^{(i)} y^2 + b_1^{(i)} y + c^{(i)}) + (2a_{01}^{(i)} y + b_0^{(i)})x + (a_{00}^{(i)})x^2. \tag{2.50}$$

Using the previous notation with f corresponding to Q_0 and g corresponding to Q_1 the coefficients

$$\begin{aligned}
\alpha_0 &= a_{11}^{(0)}y^2 + b_1^{(0)}y + c^{(0)}, & \beta_0 &= a_{11}^{(1)}y^2 + b_1^{(1)}y + c^{(1)},\\
\alpha_1 &= 2a_{01}^{(0)}y + b_0^{(0)}, & \beta_1 &= 2a_{01}^{(1)}y + b_0^{(1)},\\
\alpha_2 &= a_{00}^{(0)}, & \beta_2 &= a_{00}^{(1)},
\end{aligned}$$

can be used to produce the Bézout determinant:

$$R(y) = u_0 + u_1 y + u_2 y^2 + u_3 y^3 + u_3 y^4, \tag{2.51}$$

where

$$\begin{aligned}
u_0 &= v_2 v_{10} - v_4^2,\\
u_1 &= v_0 v_{10} + v_2(v_7 + v_9) - 2v_3 v_4,\\
u_2 &= v_0(v_7 + v_9) + v_2(v_6 - v_8) - v_3^2 - 2v_1 v_4,\\
u_3 &= v_0(v_6 + v_8) + v_2 v_5 - 2v_1 v_3,\\
u_4 &= v_0 v_5 - v_1^2,
\end{aligned}$$

and

$$\begin{aligned}
v_0 &= 2(a_{00}^{(0)}a_{01}^{(1)} - a_{00}^{(1)}a_{01}^{(0)}), & v_6 &= 2(a_{01}^{(0)}b_1^{(1)} - a_{01}^{(1)}b_1^{(0)}),\\
v_1 &= a_{00}^{(0)}a_{11}^{(1)} - a_{00}^{(1)}a_{11}^{(0)}, & v_7 &= 2(a_{01}^{(0)}c^{(1)} - a_{01}^{(1)}c^{(0)}),\\
v_2 &= a_{00}^{(0)}b_0^{(1)} - a_{00}^{(1)}b_0^{(0)}, & v_8 &= a_{11}^{(0)}b_0^{(1)} - a_{11}^{(1)}b_0^{(0)},\\
v_3 &= a_{00}^{(0)}b_1^{(1)} - a_{00}^{(1)}b_1^{(0)}, & v_9 &= b_0^{(0)}b_1^{(1)} - b_0^{(1)}b_1^{(0)},\\
v_4 &= a_{00}^{(0)}c^{(1)} - a_{00}^{(1)}c^{(0)}, & v_{10} &= b_0^{(0)}c^{(1)} - b_0^{(1)}c^{(0)},\\
v_5 &= 2(a_{01}^{(0)}a_{11}^{(1)} - a_{01}^{(1)}a_{11}^{(0)}).
\end{aligned}$$

One of the roots of equation 2.51 indicates the y value of intersection for the two parabolas. The desired root is determined using a numerical root finding algorithm. The interval over which the root can be found can be estimated using the center groups of vertices $\{V_2^a, V_3^b, V_0^c, V_1^d\}$, as shown in figure 2.19. The y value for the centroid of these points gives an estimate of the desired root, the interval can be centered at this y value with a width of 10. It is possible that the parabolas will intersect again within this small interval. This should only occur well outside of the image boundaries and can therefore be detected. In this case the interval is narrowed to exclude the incorrect point and the root finding is done again. With the y value known, $Q_{(0)}(x, y)$ can be used to solved for the x value, again using a numerical root finding algorithm in a similar manner.

After the grid intersection point, P_1, is determined it can be stored in the grid structure as, shown in figure 2.16. The final parabola intersections for a few grid intersections can be seen plotted the images in figure 2.21.

2.6 Stereo Matching

Stereo matching is the process of determining conjugate point pairs between left and right images. A recent review of stereo matching techniques by Brown et al. [33] shows that most algorithms focus on dense stereo matching, finding as many conjugate pairs as possible. For this thesis only sparse stereo matching is required, there will generally only be about 70 grid intersections that can be seen by both cameras. All the generic matching algorithms reviewed by Brown et al. [33] also require feature rich images with unique content. The grid images are repetitious and will therefore require a custom matching algorithm.

The two main problems associated with the stereo matching of grid images are the similarity of areas within the same image and the dissimilarity of areas for conjugate

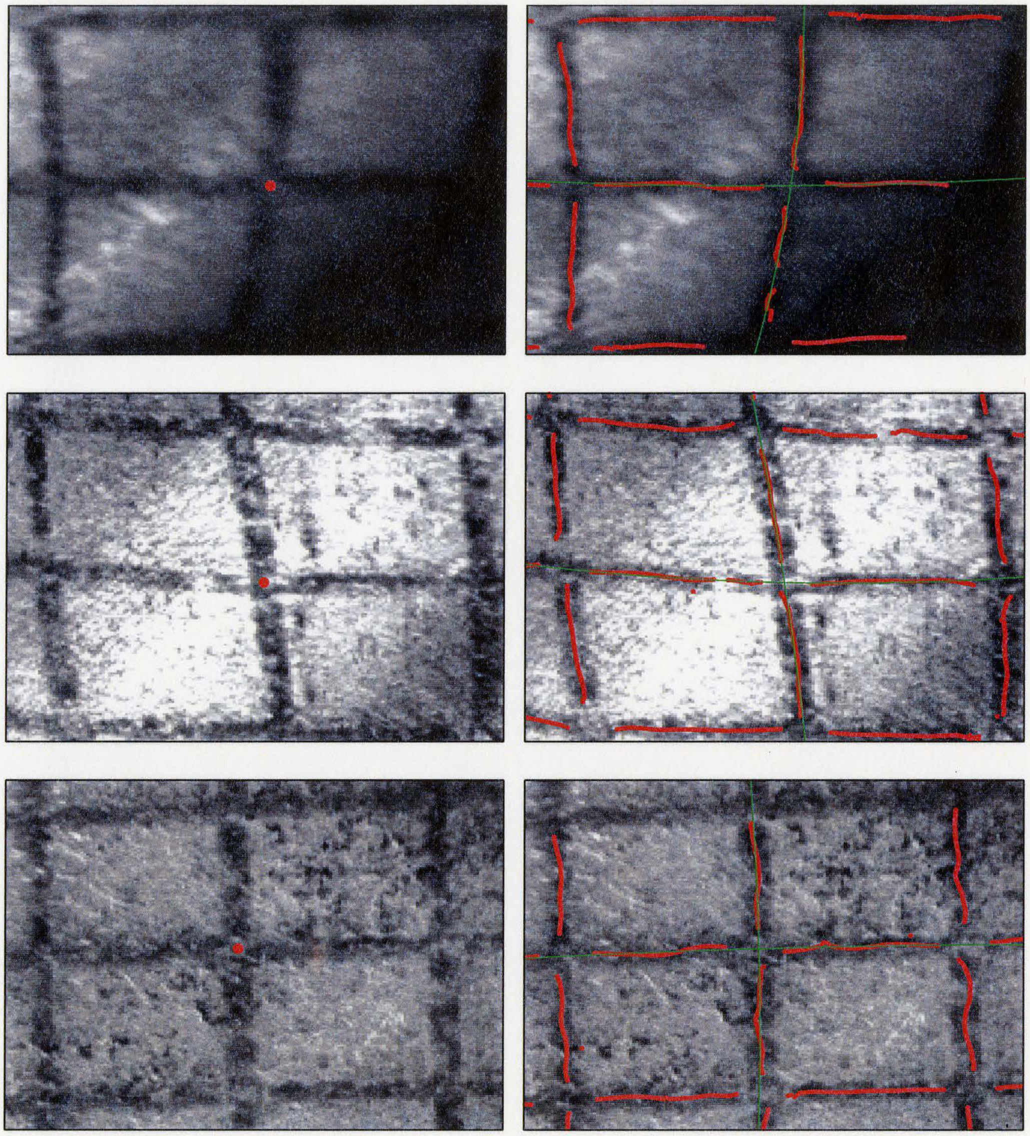
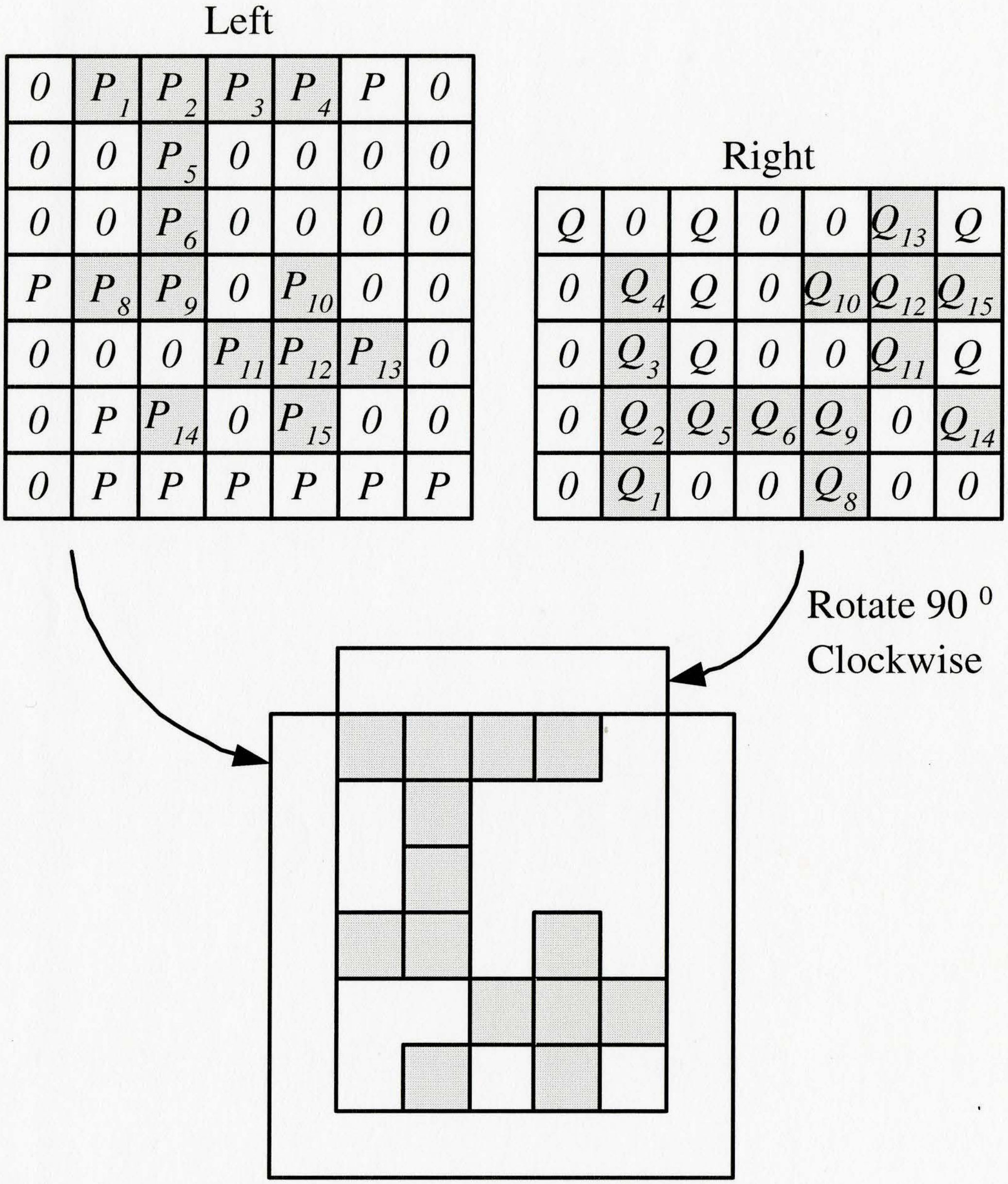


Figure 2.21: Sub pixel center lines and fitted parabolas

points. A grid intersection point from the left image is most likely to be matched with candidate points in the right image that are adjacent. The local regions about the adjacent candidate points are almost identical which makes the correct matching difficult. Due to lighting effects, it is also possible that the wrong candidate may seem to be a better match than the correct one. Therefore a matching algorithm cannot consider just the local information around a grid intersection point, the global image information must be used. The method proposed for this work uses the grid structure and epipolar geometry to match image points.

The epipolar geometry shown in figure 1.1 provides two stereo constraints, the epipolar constraint and the disparity constraint. If a pair of conjugate points, P and Q, are rectified to give P' and Q', then the epipolar constraint states that vertical components P'_v and Q'_v should be equal. As noted in section 1.2 this will not be exact, systematic and processing errors will result in a small epipolar error. This error can be used to validate candidate conjugate points. The stereo depth for epipolar geometry is given in equation 1.3, it shows that the depth of the 3-D point associated with a conjugate pair is directly proportional to the disparity between their horizontal components. If the range of depth over which a 3-D point is known then the disparity range for conjugate points is known. The range of depth for the stereo head used in this thesis is given by the combined DOF for both cameras. This disparity constraint can also be used to validate candidate conjugate points. Figure 2.22 shows a point structure for right and left stereo images. The point structure is the same as the grid structure, but it only holds the intersection points. Correct conjugate pairs P_i, Q_i, are highlighted and given the same subscript label, intersections where no point was found are labeled with zeros. Note that because of the way the grid structure is determined in section 2.3.2 there is a 90^0 rotational difference between the right and left point structures. In general, all the stereo point structures will have a rotational difference of 90^0, 180^0 or 270^0. Using this fact and the two constraints described



Left

0	P_1	P_2	P_3	P_4	P	0
0	0	P_5	0	0	0	0
0	0	P_6	0	0	0	0
P	P_8	P_9	0	P_{10}	0	0
0	0	0	P_{11}	P_{12}	P_{13}	0
0	P	P_{14}	0	P_{15}	0	0
0	P	P	P	P	P	P

Right

Q	0	Q	0	0	Q_{13}	Q
0	Q_4	Q	0	Q_{10}	Q_{12}	Q_{15}
0	Q_3	Q	0	0	Q_{11}	Q
0	Q_2	Q_5	Q_6	Q_9	0	Q_{14}
0	Q_1	0	0	Q_8	0	0

Figure 2.22: Point Structure for right and left images

earlier, the stereo matching algorithm can now be described.

The matching process can be thought of as trying to find the correct rotation and overlap of the point structures that will result in overlapped points being conjugate pairs. This process is displayed in figure 2.22. The matching is done by checking all rotations for all possible overlays of the right and left point structures. The correct matching is the one that matches the most points within acceptable disparity limits and gives the smallest average epipolar error. If the left point structure, P, is of size (M_L, N_L) and the right point structure is given by Q then algorithm 2.6 can be used to determine the correct rotation R and overlay point T. Line 11 and 12 of the algorithm check that the elements of the current pair of candidate points exist and if so, that they are within the correct disparity range. Lines 22 and 23 replace the current best rotation and offset value if the average epipolar error is smaller than the current best. The number of matched points is also required to be at least 80% of current best number of points. This line also ensures that the best alignment is not due to one single matching which could have a very low average epipolar error.

This type of matching would be prohibitive for dense stereo, but for only 70 points it is very fast. The process also assumes that the grid structure is correct, any intersections that are incorrectly placed in the point structure could result in incorrectly reconstructed 3-D points. With conjugate pairs determined the 3-D grid intersection points can be reconstructed using the triangulation equations given in section 3.2.8.

Algorithm 2 Stereo matching algorithm

```
for $rot = 0$ to 3 do
  $Q \leftarrow$ rotate $Q$ by (rot × 90) degrees
  $M_R \leftarrow$ number of rows in $Q$
  $N_R \leftarrow$ number of columns in $Q$
  for $i = 1$ to $M_L$ do
    for $j = 1$ to $N_L$ do
      $Err \leftarrow 0$
      $Num \leftarrow 0$
      for $k = 1$ to $min\{M_R, M_L - i\}$ do
        for $l = 1$ to $min\{N_R, N_L - j\}$ do
          if $(P(i,j) \neq 0) \wedge (Q(k,l) \neq 0)$ then
            if $(D_{LOW} < (P_u(i,j) - Q_u(k,l)) < D_{HIGH})$ then
              $Err \leftarrow Err + \|P_v - Q_v\|$
              $Num \leftarrow Num + 1$
            end if
          end if
        end for
      end for
      if $((Err/Num < E/N) \wedge (Num > 0.8N)) \vee ((N <= 1) \wedge (Num > 1))$
      then
        E ← Err
        N ← Num
        R ← (rot × 90)
        T ← $(i,j)$
      end if
    end for
  end for
end for
```

Chapter 3

System Calibration

3.1 System Hardware

3.1.1 Introduction

Although there are many commercial stereo vision systems available for research purposes, there are none which are designed for close-up measurement. Commercial systems are normally designed to model the human vision system, which is not particularly well suited for close-up imaging. Without a satisfactory commercial system available, a close-up stereo vision system was designed and built.

Being at such close distances with a cameras limited field of view requires many images to cover all of a part surface. To provide movement, the stereo system was attached to a Coordinate Measuring Machine (CMM). The CMM also provides a pan and tilt capability through an articulating motorized probe. The complete system can be seen in Fig. 3.1 and is described in the following sections.

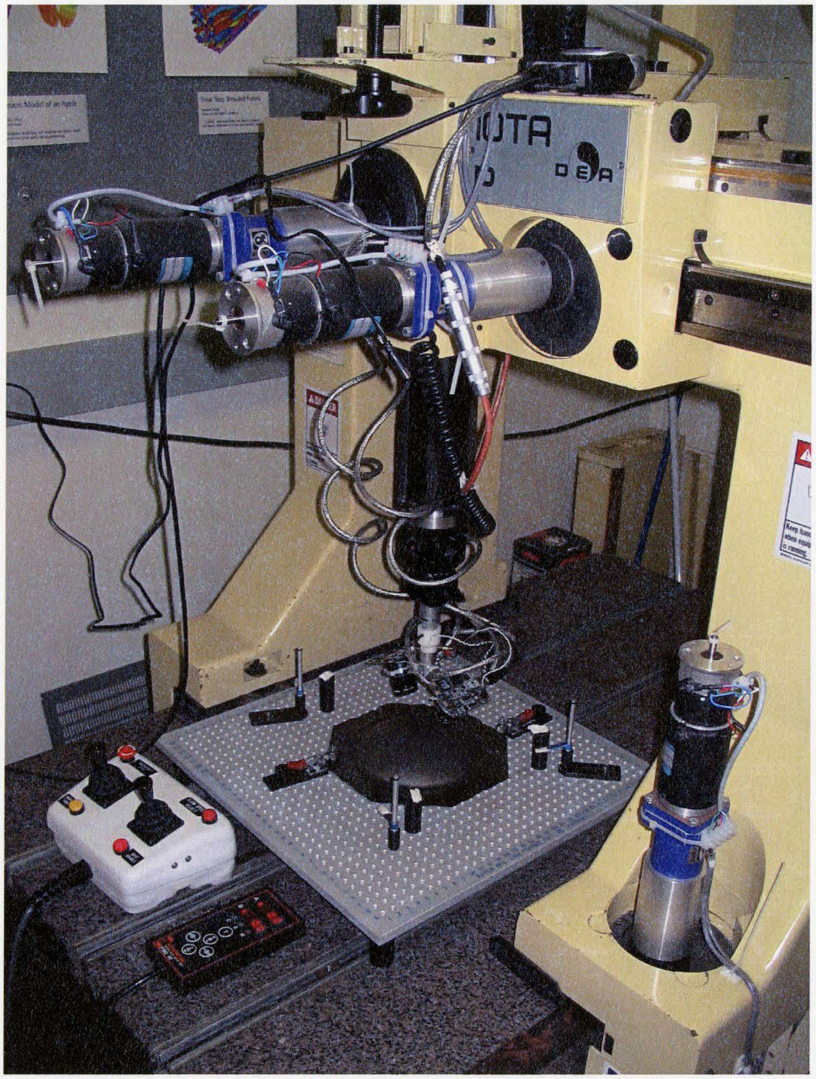


Figure 3.1: Complete vision system consisting of CMM and stereo head.

3.1.2 Cameras and Lenses

The stereo head consists of two Point Grey[3] Dragonfly board mounted cameras. They stream images to a computer using the IEEE-1394 communication protocol. The cameras communicate via a hub that also supplies power. The complete specifications for the camera are listed in Table 3.1.

Imaging Device:	1/3” Sony CCD 1024x768 B&W ICX204 HAD progressive scan image sensor with square pixels
Pixel size:	4.65 μm x 4.65 μm
Frame rates:	5, 7.5, 3.75, 1.875 FPS
Supported formats:	8-bit or 16-bit Mono
Signal to noise ratio:	Greater than 60dB
Power:	Through IEEE-1394, less than 2.0W
Shutter:	Auto/Manual, 1/15s to 1/6000s, Shutter time can be extended up to 60 seconds
Gain:	Auto/Manual, 0-27dB, 0.035dB resolution
Synchronization:	Less than 120 μs
Dimensions:	2.5” x 2.0”

Table 3.1: Point Grey Dragonfly specifications

Each camera of the stereo head uses a 12 mm focal length Cosmicar/Pentax compound lens. This lens is designed for use in metrology applications and has a smaller lens distortion than a standard lens. It can be focused from the Minimum Object Distance (MOD) of 20 cm to infinity. The focus is manually adjusted to the desired object range. For the purpose of describing lens performance, this compound lens system can be adequately modeled using a thin lens. The next two sections use this model to describe issues pertaining to depth of field and minimum object distance.

Depth of Field

One important lens parameter that can be estimated using the thin lens model is the Depth Of Field (DOF). For a fixed focus, this is the range of distance which

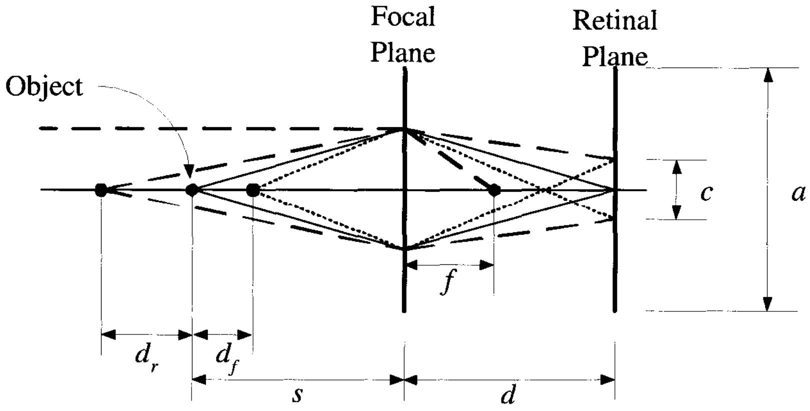


Figure 3.2: Depth Of Field and Circle of Confusion

remains sharp or in focus. All lenses have a limited DOF when focused on a point in space. The human vision system deals with this by constantly changing focal lengths, iris size and eye position to keep objects of interest in focus. Designing an accurate machine vision system to do this has proven difficult. Calibration techniques that provide camera parameters over a range of lens settings are available, but are usually much less accurate than fixed focus calibration.

For a thin lens, the relationship between a point in space, the focal length of the camera and the imaged point is given by Descartes' formula:

$$\frac{1}{s} = \frac{1}{d} + \frac{1}{f}, \tag{3.1}$$

where s is the distance from the lens to the object, d is the distance to the image plane and f is the focal length. This lens formula assumes an aberration free optical system and points lying near the optical axis. It will be sufficient though to provide estimates for lens parameters. The DOF can be described using a rear d_r and front d_f bounding plane that define the volume over which points remain in focus. Figure 3.2 shows the situation where an object in focus is moved to its near and far DOF limits. The aperture diameter is a with the circle of confusion given by c. The circle of confusion must be smaller than the resolution of the imaging sensor for the point to be in focus. For the Dragonfly cameras the maximum circle of confusion where a point remains in focus is approximately equal to the diameter of a CCD element; $1.4142 * 4.5\,\mu\text{m}$. Using Figure 3.2 and simple geometry the DOF parameters can be determined to be:

$$d_f = \frac{scp(s-f)}{f^2 + cp(s-f)}, \tag{3.2}$$

$$d_r = \frac{scp(s-f)}{f^2 - cp(s-f)}, \tag{3.3}$$

where $p = f/a$ is the F-stop number. These equations show that for a fixed focus lens with fixed sensor element size, the DOF can only be increased if the aperture diameter is decreased or object distance s is increased. Although decreasing the F-stop number increases the DOF, it also reduces the amount of light used to image the object and results in increased noise. A compromise must be made between acceptable image noise and DOF.

Minimum Object Distance

One way to decrease the MOD is to use a lens with a smaller focal length. Unfortunately this requires a more powerful lens with an associated greater amount of lens distortion. Although this can be compensated for, as will be seen later, for metrology applications it is better to reduce lens distortion at the source. A 12 mm lens gives a reasonably small MOD without introducing too much lens distortion.

Another method of decreasing the MOD involves adding a macro ring, also known as an extension tube. A macro ring lengthens the effective focal length of a camera and decreases the MOD. Off the shelf lenses are designed to operate within their specifications and adding a macro ring comes with two consequences. The first consequence is that a lens can no longer focus on objects at infinity. Fortunately this is not important for close-up stereo vision. The second consequence is that the Depth Of Field (DOF) is reduced.

Installing a 1 mm micro ring to the Cosmicar/Pentax lens decreased the MOD to approximately 12 cm. The smallest possible aperture setting that gave reasonable image results for close up distances was an F-stop number of 4. Using these settings and equations 3.2 and 3.2, d_r and d_f are shown to be 5 mm and 4.5 mm respectively. This approximation agreed with experimental d_r and d_f values which were approximately measured to be 4 mm and 3.5 mm respectively. For these measurements the determination of an in focus image was done by eye.

A point in space that is imaged out of the depth of field limits will appear with decreased intensity. The COC produced by the point will also be affected by superposition due to diffraction. Although the amount of blur in the image is not important for measuring grid intersections, it can affect the measurement accuracy. If the COC is too big the change in intensity and diffraction effects can lead to later processing problems when performing sub pixel edge detection and image thresholding. Real lenses also have a certain amount of lens distortion, which can be compensated for through calibration. The calibration parameters are determined for images in focus, the compensation parameters will be incorrect for points that are out of focus. A small amount of de-focus is acceptable but too much can affect point detection.

3.1.3 System Kinematics

The DEA-IOTA CMM is a highly accurate metrology device normally used for tactile surface measurement. It provides travel in the three major axis with additional degrees of freedom given by a Renishaw [4] PH10 motorized probe head attached to its vertical column. This system is normally used to orient a touch trigger probe, but for this application it will be used to orient a stereo camera system. The motorized probe can pan 360^0 and tilt from 0^0 to 105^0 in 7.5^0 increments. A locking mechanism results in each of these positions being highly repeatable, $< \pm 0.5\,\mu$m (2σ). The CMM accuracy with touch trigger probe is approximately $\pm 10\ \mu$m.

During normal operation a touch probe is driven into a part surface at slow speeds and its deflection breaks a closed circuit thereby signalling the CMM to record a point. The CMM reads the scales of the machine and, using an error map of the workspace, determines the 3D location. An offset for the current probe orientation is then added giving the final 3D point in the WCS. With a stereo camera system attached, the CMM will not be in motion during image capture. Any motion occurring between right and left image captures would affect the stereo triangulation. A CAD drawing of the custom made stereo mount can be seen in Appendix B. The center axis of the mount provides two contact points which allow hardware triggering of the CMM. Although the mount was designed to allow the cameras to automatically trigger the CMM using this interface, it was not used for the current application. Software triggering of the CMM was acceptable because of the CMM stability in a set position. The automated triggering capability was added for future work involving fast image capture.

The stereo mount was constructed out of aluminum and plastic to reduce weight, essential to be within the torque limits of the motorized probe head. Thermal expansion or any other movement that would cause the stereo pair to be non-rigid was assumed to be minimal. The mount also allows adjustments to the vergence angle, distance between cameras and rotation of the cameras about one axis. These degrees of freedom are necessary to make fine adjustments and to ensure cameras views have the maximum overlap.

3.2 Camera Calibration

3.2.1 Testfield Points

Extracting measurements from 2-D image data requires a relationship between the image space and the object space for each camera. Finding this relationship is commonly referred to as camera calibration and requires a set of known control points within the object space. The exception to this being recent work on self-calibration [34], [35]. These methods only require a sequence of images from a common scene, however they are very sensitive to outliers and are therefore unreliable. The most accurate type of camera calibration involves the use of a 3-D testfield of points. Photogrammetry calibration techniques in room size scenes typically employ theodolites to measure such points. However, at closer distances the measurement of control points becomes extremely difficult and a 2-D testfield is usually adopted. One convenient method of creating a small 2-D test field is to print one using a laser printer. This is common practice within the computer vision community but paper loses its coplanarity at very close distances. For the work presented within this thesis a set of calibration lines was lithographically printed onto a flat aluminum sheet (Figure 3.3). This was done on an imaging machine normally used to create templates for newspapers. The lines in Figure 3.3 define 70 intersection points. The center of each intersection was determined by averaging the four adjacent corners of the background squares. Manual measurement of the corners was performed using an optical microscope of ±2.5 μm resolution.

3.2.2 Interior and Exterior Orientation

Most camera calibration procedures assume the pinhole camera model and try to find physical camera parameters such as position, orientation, etc.. The imaging process

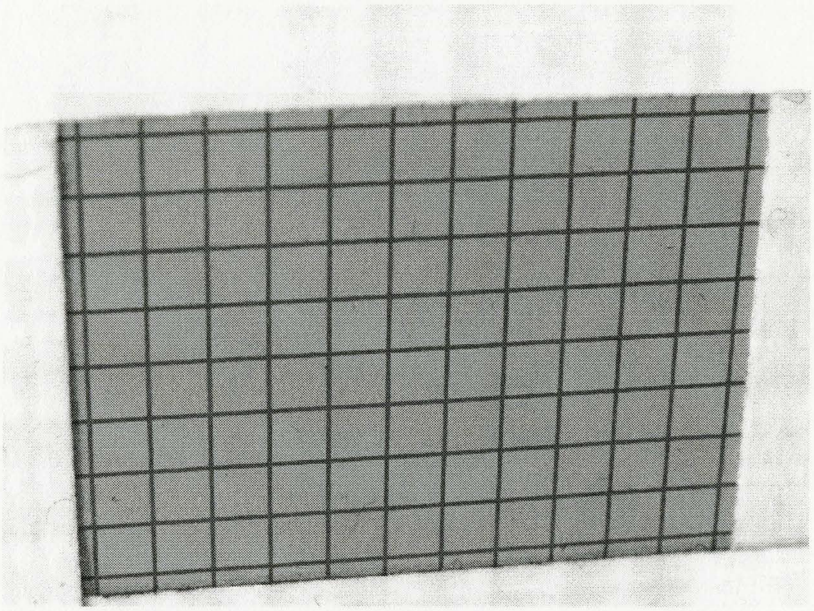


Figure 3.3: Calibration Grid (Approximate line spacing of 2.5mm)

in a pinhole camera model can be mathematically expressed using the collinearity condition. This condition states that a point in object space, the corresponding point in image space and the perspective center are collinear. To apply the collinearity condition, points given in $(row, column)$ computer image coordinates (u, v), must be transformed into the physical image plane coordinates (x, y). Given the computer image size (m, n) and the inter-pixel spacing of the CCD array (d_x, d_y), the transformation is given by:

$$(x, y) = (d_x u, -d_x v),$$

Referring to Figure 3.4, the collinearity condition can be enforced using an orthogonal rotation matrix and scale factor:

$$\begin{bmatrix} x_i - x_p \\ y_i - y_p \\ 0 - f \end{bmatrix} = \lambda R \begin{bmatrix} X_{Ci} - X_0 \\ Y_{Ci} - Y_0 \\ Z_{Ci} - Z_0 \end{bmatrix}, \text{ with } \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \qquad (3.4)$$

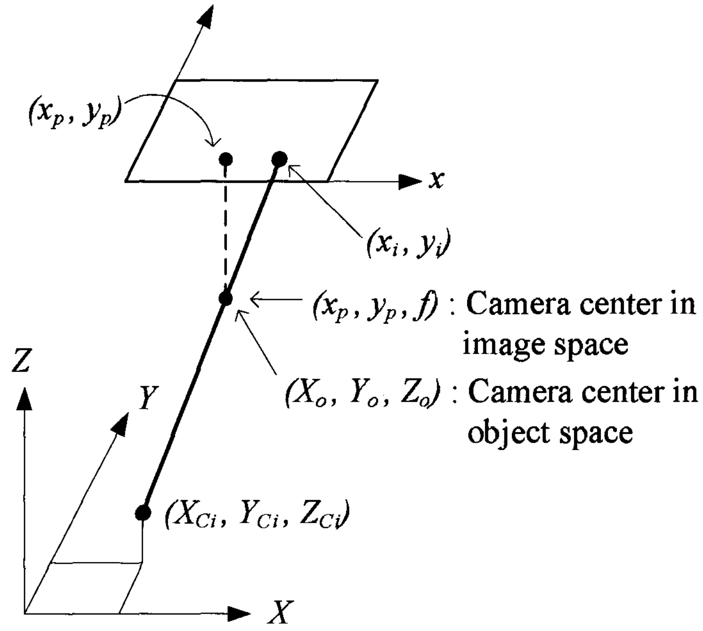

(x_p, y_p)
x
(x_i, y_i)
(x_p, y_p, f) : Camera center in image space
(X_o, Y_o, Z_o) : Camera center in object space
Z
Y
(X_{Ci}, Y_{Ci}, Z_{Ci})
X

Figure 3.4: Pinhole camera model showing image space and object space

and where

(X_{Ci}, Y_{Ci}, Z_{Ci})	object space coordinates of control points
f	focal length
(X_0, Y_0, Z_0)	object space coordinates of camera center
(x_p, y_p)	image plane coordinates of principal point
i	index of object point
λ	scale factor

The scale factor λ is of no interest and can be eliminated by division of the first and second rows with the third:

$$(x_i - x_p) = -f\frac{r_{11}(X_{Ci} - X_0) + r_{12}(Y_{Ci} - Y_0) + r_{13}(Z_{Ci} - Z_0)}{r_{31}(X_{Ci} - X_0) + r_{32}(Y_{Ci} - Y_0) + r_{33}(Z_{Ci} - Z_0)} \tag{3.5}$$

$$(y_i - y_p) = -f\frac{r_{21}(X_{Ci} - X_0) + r_{22}(Y_{Ci} - Y_0) + r_{23}(Z_{Ci} - Z_0)}{r_{31}(X_{Ci} - X_0) + r_{32}(Y_{Ci} - Y_0) + r_{33}(Z_{Ci} - Z_0)} \tag{3.6}$$

Using a sufficient number of known control points, equations 3.5 and 3.6 can be used to determine unknown parameters. For convenience parameters are grouped into exterior and interior orientation sets. The interior orientation defines the internal camera geometry and consists of three parameters, the focal length f and the location of the principal point (x_p, y_p). The exterior orientation consists of six parameters, three defining the location of the camera center (X_0, Y_0, Z_0) and three Euler angles (ω, ϕ, κ) describing the camera orientation. Equations 3.5 and 3.6 were developed assuming the pinhole camera model, additional interior parameters must be added to more accurately represent the imaging process. However, solutions to equations 3.5 and 3.6 are important because they provide initial estimates for more complex iterative methods. One method of determining both interior and exterior orientation is called the Direct Linear Transform and was originally developed by Abdel-Aziz and Karara [47]. It determines a linear 3×4 matrix that directly maps object space points to the image plane. The physical camera parameters f, (X_0, Y_0, Z_0), R and (x_p, y_p) can be determined by various matrix manipulations [43]. However, it requires non coplanar control points and is therefore not suitable for calibration at close ranges. Tsai [39] presented a method to determine f, (X_0, Y_0, Z_0) and R using coplanar control points. The algorithm does not solve for the principal point (x_p, y_p), but assumes it is the same as the image center. It is suggested in [39] that an error in principal point has an insignificant effect on 3D measurement accuracy, although no thorough investigation is done. Tsai's method for coplanar camera calibration is outlined below:

Step 1

Using control points $(X_{Ci}, Y_{Ci}, Z_{Ci})\quad i = 1 \ldots m$, with corresponding image plane points (x_i, y_i), form a matrix A with rows:

$$a_i = \left[\begin{array}{ccccc} y_i X_{Ci} & y_i Y_{Ci} & y_i & -x_i X_{Ci} & -x_i Y_{Ci} \end{array}\right]$$

Form a vector b from the m image plane observations of the calibration points:

$$b = \left[\begin{array}{cccc} x_1 & x_2 & \ldots & x_m \end{array}\right]^T$$

With $m > 5$, the overdetermined system $Au = b$ can be solved using SVD where:

$$u_1 = \frac{r_{11}}{Y_0}, \quad u_2 = \frac{r_{12}}{Y_0}, \quad u_3 = \frac{r_{21}}{Y_0}, \quad u_4 = \frac{r_{22}}{Y_0}, \quad u_5 = \frac{X_0}{Y_0}.$$

Step 2

Compute the magnitude of Y_0:

$$\begin{array}{ll}
\text{if} & (u_1 \neq 0) \wedge (u_2 \neq 0) \wedge (u_3 \neq 0) \wedge (u_4 \neq 0) \\
 & Y_0^2 = \dfrac{U - (U^2 - 4(u_1u_4 - u_2u_3)^2)^{\frac{1}{2}}}{2(u_1u_4 - u_2u_3)^2}, \text{ where } \quad U = u_1^2 + u_2^2 + u_3^2 + u_4^2 \\
\text{else if} & (u_1 \neq 0) \wedge (u_2 \neq 0) \\
 & Y_0^2 = \dfrac{1}{u_3^2 + u_4^2} \\
\text{else} & \\
 & Y_0^2 = \dfrac{1}{u_3^2 + u_4^2}
\end{array}$$

Choose (x_f, y_f), $f = 1 \ldots m$ farthest from the center of image. Compute r_{11}, r_{12}, r_{21},

r_{22}, X_0 and determine the sign of Y_0:

$$\begin{aligned} r_{11} &= u_1 Y_M \\ r_{12} &= u_2 Y_M \\ r_{21} &= u_3 Y_M \\ r_{22} &= u_4 Y_M \\ X_0 &= u_5 Y_M \end{aligned}$$

```
if    $(sgn(r_{11}X_{Cf} + r_{12}Y_{Cf} + X_0) = sgn(xd_f))\wedge$
      $(sgn(r_{21}X_{Cf} + r_{22}Y_{Cf} + Y_M) = sgn(yd_f))$
      $Y_0 = Y_0$
else
      $Y_0 = -Y_0$
```

Step 3

Compute remaining rotation matrix values:

$$\begin{aligned} r_{13} &= \sqrt{1 - r_{11}^2 - r_{12}^2} \\ r_{23} &= \sqrt{1 - r_{21}^2 - r_{22}^2} \\ r_{31} &= \frac{1 - r_{11}^2 - r_{12}r_{21}}{r_{13}} \\ r_{31} &= \frac{1 - r_{21}r_{12} - r_{22}^2}{r_{23}} \\ r_{23} &= \sqrt{1 - r_{31}r_{13} - r_{32}r_{23}} \end{aligned}$$

If $(sgn(r_{11}r_{21} + r_{12}r_{22}) > 0)$ then negate the sign of r_{21}. The signs of r_{31} and r_{32} may still need to be adjusted in the next step.

Step 4

Compute initial estimates for camera constant c and Z_0 translation. Using control points, their corresponding image plane points and the vertical size of a CCD element d_y, form a matrix A with rows:

$$a_i = \begin{bmatrix} (r_{21}X_{Ci} + r_{22}Y_{Ci} + Y_0) & -y_i \end{bmatrix}$$

Form a vector b with elements $b_i = (r_{31}X_{Ci} + r_{32}Y_{Ci})y_i$. Solve the overdetermined system $Av = b$ using SVD. The camera constant and translation parameter Z_0 are then given by:

$$v = \begin{bmatrix} c & Z_0 \end{bmatrix}$$

If $(c < 0)$, then negate r_{31} and r_{32}.

Step 5

Determine final values for c, Z_0 using equation 3.6 and an optimization technique such as simplex minimization. Initial estimates for c and Z_0 are from step 4.

3.2.3 Additional Calibration Parameters

The collinearity equations are based on the pinhole camera model. To comprehensively model the imaging process certain systematic errors must be added. These errors are added in the form of additional parameters. They are considered part of the interior orientation and correct the image space coordinates to more accurately represent physical reality.

It has long been known [29] that lens distortion must be included in the calibration parameter set if accurate stereo measurement is to occur. Brown [17] successfully modeled radial distortion using a power series with coefficients K_1, K_2, K_3. Tangential distortion parameters, P_1, P_2 can also be added to produce the following correction equations:

$$\begin{aligned} x &= x' + \bar{x}(r^2K_1 + r^4K_2 + r^6K_3) + [(P_1(r^2 + 2\bar{x}^2) + 2P_2\bar{x}\bar{y}], && (3.7) \\ y &= y' + \bar{y}(r^2K_1 + r^4K_2 + r^6K_3) + [2P_1\bar{x}\bar{y} + P_2(r^2 + 2\bar{y}^2)], && (3.8) \end{aligned}$$

where (x', y') is the distorted image plane coordinate, (x, y) is the corrected image plane coordinate and $\bar{x} = (x' - x_p)$, $\bar{y} = (y' - y_p)$, $r = \sqrt{\bar{x}^2 + \bar{y}^2}$. Brown [17] noted that if tangential lens distortion was included during calibration, it could absorb some of the effect of the principal point (x_p, y_p). Tsai [39] noted through personal experience that tangential lens distortion can be ignored without problem, and if included in a model can even cause numerical instability during calibration. Tangential lens distortion is usually caused by incorrect lens assembly. The lenses used for this thesis are high quality and designed for use in metrology applications, therefore tangential lens distortion was ignored.

A common additional calibration parameter that can be ignored in this work is the timing parameter s_x. It was originally included to account for a frame grabber sampling mismatch. A camera signal could be sampled at a rate differing from the horizontal spacing between sensor pixels, creating a horizontal shift. Modern cameras such as the Point Grey Dragonfly used for this thesis communicate digitally and therefore avoid this problem.

3.2.4 Bundle Adjustment

There are many calibration methods available using coplanar test fields, for example [39], [48] and [22]. However, the most accurate type of camera calibration known to date is that of bundle adjustment with self calibration [24]. It is an iterative technique involving a least squares solution that was first introduced by Brown [16]. The name "Bundle" refers to the group of light rays entering the lens and the "Adjustment" refers to change in exterior and interior orientation parameters. The technique has been successfully used in photogrammetry for many years and is receiving more attention within the computer vision [9] community. It has given experimental 3D accuracies exceeding 1 part in 1000000 for film based cameras [15] and 1 part in 30000 for CCD cameras with off the shelf components [24]. The general bundle adjustment technique can be used to model any number of parameters. A problem involving m non linear independent observation equations, n unknowns and r measured parameters can be functionally written as:

$$F_i(a_1, a_2, .., a_n, b_1, b_2, .., b_r) = 0, \qquad i = 1 \ldots m \tag{3.9}$$

Bundle adjustment is applied by first linearizing F using a Taylor series expansion, only the first order term is retained. The resulting linear observation equations are formulated in the Gauss-Markov model:

$$-V = A\Delta - C \tag{3.10}$$

where the design matrix A contains the partial derivatives of F with respect to the unknown parameters Δ, with the partial derivatives being evaluated at the current approximation for Δ. The difference between the measured and approximated values is stored in the absolute vector C, and V represents the true error. It is assumed that the errors in the observational equations have a normal distribution, mean of zero, are uncorrelated and have equal variance. The estimation of parameters is given by the least squares solution:

$$\tilde{\Delta} = (A^T PA)^{-1} A^T WC; \quad N = A^T WA, \tag{3.11}$$

with

$$W = \sigma_0^2 \Sigma^{-1}, \tag{3.12}$$

being the weight matrix and Σ the covariance matrix of observations. The variance of unit weight a priori is σ_0^2 and N is the matrix of normal equations common to least squares data fitting. At each iteration the design matrix A and the absolute vector C are evaluated using the current approximation and a priori value, a solution for the current iteration is then produced using 3.11. The process is iterated until the desired tolerance is met. The next two sections describe bundle adjustment methods of determining interior and exterior orientation parameters.

3.2.5 Determining Interior Orientation

A bundle adjustment method of determining (x_p, y_p), K_1, K_2, K_3, P_1 and P_2 is described by Brown [17]. It is based on the fact that a pinhole camera will project a straight line in object space to a straight line on the image plane. Any deviation from straightness can be used to account for radial distortion, tangential distortion and the principal point. This is more accurate than other methods of interior camera calibration [48], [39] that use discrete calibration points. Not only does it provide more data for calibration but it also guarantees more data coverage near to image edges where the strongest lens distortion occurs.

A straight line L can be expressed as:

$$x \sin\theta + y \cos\theta = \rho \tag{3.13}$$

where ρ is length of a vector that is extended from the origin until it touches L orthogonally. The angle between this vector and the y-axis is given as θ. A nonlinear observation equation can be created by substituting the lens distortion correction equations 3.7 and 3.8 into equation 3.13. If a set of straight lines are imaged and the jth point along the ith line is given by (x'_{ij}, y'_{ij}), a set of nonlinear observational equations can be formed:

$$F_i(x_{ij}, y_{ij}; x_p, y_p, K_1, K_2, K_3, P_1, P_2, \theta_i, \rho_i) = 0, \qquad i = 1 \ldots m \tag{3.14}$$

Taking the first term of a Taylor series expansion results in the following observational equations:

$$A_{ij}v_{ij} + \dot{B}_{ij}\dot{\delta} + \ddot{B}_{ij}\ddot{\delta}_i = \epsilon_{ij} \tag{3.15}$$

in which

$$\epsilon_{ij} = -F(x_{ij}^0, y_{ij}^0; x_p^k, y_p^k, K_1^k, K_2^k, K_3^k, P_1^k, P_2^k, \theta_i^k, \rho_i^k) \tag{3.16}$$

$$v_{ij} = \begin{bmatrix} v_{x_{ij}} \\ v_{y_{ij}} \end{bmatrix}, \qquad \delta = \begin{bmatrix} \delta_{x_p} \\ \delta_{y_p} \\ \delta_{K_1} \\ \vdots \\ \delta_{P_2} \end{bmatrix}, \qquad v_{ij} = \begin{bmatrix} v_{\delta\theta_i} \\ v_{\delta\rho i} \end{bmatrix} \tag{3.17}$$

with the A, $\dot{B}$ and $\ddot{B}$ being the following Jacobian matrices:

$$\begin{aligned} A_{ij} &= -\frac{\partial \epsilon_{ij}}{\partial(x_{ij}^0, y_{ij}^0)} \\ \dot{B}_{ij} &= -\frac{\partial \epsilon_{ij}}{\partial(x_p^k, y_p^k, K_1^k, , P_2^k)} \\ \ddot{B}_{ij} &= -\frac{\partial \epsilon_{ij}}{\theta_i^k, \rho_i^k} \end{aligned} \tag{3.18}$$

The superscripts "0" and "k" refer to evaluation using the measured values and current approximations respectively. Normal equations for the ith line can be formulated as:

$$\begin{bmatrix} \ddot{N}_i & \bar{N}_i \\ \bar{N}_i^T & \ddot{N}_i \end{bmatrix} \begin{bmatrix} \dot{\delta} \\ \ddot{\delta}_i \end{bmatrix} = \begin{bmatrix} \dot{c}_i \\ \ddot{c}_i \end{bmatrix} \tag{3.19}$$

where

$$\begin{aligned} \ddot{N}_i &= \sum \ddot{N}_{ij}, & \dot{c}_i &= \sum \dot{c}_{ij}, \\ \bar{N}_i &= \sum \bar{N}_{ij}, & \ddot{c}_i &= \sum \ddot{c}_{ij}, \\ \ddot{N}_i &= \sum \ddot{N}_{ij}, \end{aligned} \tag{3.20}$$

and in which

$$\begin{array}{rclcrcl} \ddot{N}_{ij} & = & p_{ij}\dot{B}_{ij}^T\dot{B}_{ij}, & \quad & \dot{c}_{ij} & = & p_{ij}\dot{B}_{ij}^T\epsilon_{ij}, \\ \bar{N}_{ij} & = & p_{ij}\dot{B}_{ij}^T\ddot{B}_{ij}, & & \ddot{c}_{ij} & = & p_{ij}\ddot{B}_{ij}^T\epsilon_{ij}, \\ \ddot{N}_{ij} & = & p_{ij}\ddot{B}_{ij}^T\ddot{B}_{ij}. & & & & \end{array} \tag{3.21}$$

In equations 3.21, p_{ij} is the scalar

$$p_{ij} = (A_{ij}\Lambda_{ij}A_{ij}^T)^{-1} \tag{3.22}$$

where Λ_{ij} is the covariance matrix of x_{ij}^0, y_{ij}^0. It is shown in [17] and [14] that the normal equations from all m lines can be combined to from the following system of normal equations:

$$\left[\begin{array}{c|cccc} \dot{N}+\dot{W} & \bar{N}_1 & \bar{N}_1 & \ldots & \bar{N}_m \\ \hline \dot{N_1}^T & \ddot{N}_1 & 0 & \ldots & 0 \\ \dot{N_1}^T & 0 & \ddot{N}_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \dot{N_m}^T & 0 & 0 & \ldots & \ddot{N}_m \end{array}\right] \left[\begin{array}{c} \dot{\delta} \\ \hline \ddot{\delta}_1 \\ \ddot{\delta}_2 \\ \vdots \\ \ddot{\delta}_n \end{array}\right] = \left[\begin{array}{c} \dot{c}-\dot{W}[\epsilon^{(k-1)}-\epsilon^k] \\ \hline \ddot{c}_1 \\ \ddot{c}_2 \\ \vdots \\ \ddot{c}_m \end{array}\right] \tag{3.23}$$

in which

$$\dot{N} = \sum_{i=1}^{m} \dot{N}_i, \quad \dot{c} = \sum_{i=1}^{m} \dot{c}_i,$$

and $\dot{W}$ is the inverse of the covariance matrix for the parameters. The discrepancy between the a priori estimate and the current approximation is the vector $[\epsilon^{(k-1)} - \epsilon^k]$. This system can be solved using equation 3.11. With a large number of lines the order of equation 3.23 grows too large to be solved using direct matrix techniques. Fast techniques exploiting the block diagonality of 3.23 are available [17], [14] that have no bound on m.

This technique was applied to the same type of lithographically printed grid as shown in figure 3.3. Center line points were determined using line detection technique discussed in section 2.4. Due to ambiguity in the hysteresis procedure, points detected

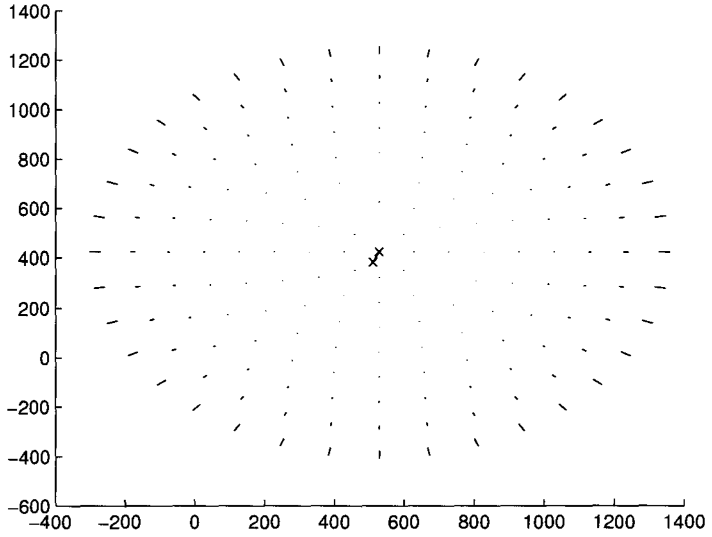


Figure 3.5: Radial lens distortion and principal point

in the local area around grid intersections were ignored. A Hough line transform was applied to determine the set $\{\theta, \rho\}$ for each line. Multiple instances of the same line were dismissed by choosing the line with the strongest response. The change in principal point and the effect of radial distortion for the left camera can be seen in figure 3.5. The lenses have very low distortion so the normal image plane size of 1024 × 768 is widened in the figure to effectively show the radial distortion. Note that the principal point marked with the × moves from the assumed position in the center of the image, to the right and upwards. Both right and left cameras exhibited barrel distortion, the internal parameters for each camera are shown in table 3.2.5, where (u_p, v_p) are the pixel coordinates of the principal point (x_p, y_p).

After this bundle adjustment algorithm is applied, the only unknown interior orientation parameter is the focal length f of the camera. This is determined using the calibration procedure from Tsai [39].

	Left Camera	Right Camera
x_p	2.434315	2.276582
y_p	1.972908	1.893317
u_p	523.5087	489.5876
v_p	424.2813	407.1650
K_1	0.001158592	0.001198772
K_2	-0.000013361	-0.00000915
K_3	0.000000710	0.000000774

Table 3.2: Internal parameters for right and left cameras

3.2.6 Determining Exterior Orientation

Exterior orientation parameters consist of the camera position (X_0, Y_0, Z_0) and the camera orientation (ω, ϕ, κ). With known control points these parameters can be solved using the following bundle adjustment algorithm.

Using the jth control point from the ith image, the collinearity equations 3.5 and 3.6 can be written in a functional form as:

$$F_{x_{ij}} = (x_{ij} - x_p) - f\frac{r_{11}(X_{Cj} - X_0) + r_{12}(Y_{Cj} - Y_0)r_{13}(Z_{Cj} - Z_0)}{r_{31}(X_{Cj} - X_0) + r_{32}(Y_{Cj} - Y_0)r_{33}(Z_{Cj} - Z_0)} \tag{3.24}$$

$$F_{y_{ij}} = (y_{ij} - y_p) - f\frac{r_{21}(X_{Cj} - X_0) + r_{22}(Y_{Cj} - Y_0)r_{23}(Z_{Cj} - Z_0)}{r_{31}(X_{Cj} - X_0) + r_{32}(Y_{Cj} - Y_0)r_{33}(Z_{Cj} - Z_0)} \tag{3.25}$$

Taking the first term of a Taylor series expansion of equations 3.24 and 3.25 results in the following observational equations:

$$\begin{array}{ccccccccc} V_{x_{ij}} & + & b_1\Delta\omega_i & + & b_2\Delta\phi_i & + & b_3\Delta\kappa_i & + & \\ & & b_4\Delta X_{0i} & + & b_5\Delta Y_{0i} & + & b_6\Delta Z_{0i} & + & \\ & & b_7\Delta X_i & + & b_8\Delta Y_i & + & b_9\Delta Z_i & + & F^k_{x_{ij}} = 0, \end{array} \tag{3.26}$$

$$\begin{array}{ccccccccc} V_{y_{ij}} & + & b_{11}\Delta\omega_i & + & b_{12}\Delta\phi_i & + & b_{13}\Delta\kappa_i & + & \\ & & b_{14}\Delta X_{0i} & + & b_{15}\Delta Y_{0i} & + & b_{16}\Delta Z_{0i} & + & \\ & & b_{17}\Delta X_i & + & b_{18}\Delta Y_i & + & b_{19}\Delta Z_i & + & F^k_{y_{ij}} = 0, \end{array} \tag{3.27}$$

where $V_{x_{ij}}$, $V_{y_{ij}}$ represent the residuals between the measured image values and their true value. The partial derivatives b_i evaluated at the kth estimate are given by:

$$\begin{aligned}
b_1 &= \left(\frac{\partial F_x}{\partial \omega_i}\right)^k, & b_{11} &= \left(\frac{\partial F_y}{\partial \omega_i}\right)^k, \\
b_2 &= \left(\frac{\partial F_x}{\partial \phi_i}\right)^k, & b_{12} &= \left(\frac{\partial F_y}{\partial \phi_i}\right)^k, \\
b_3 &= \left(\frac{\partial F_x}{\partial \kappa_i}\right)^k, & b_{13} &= \left(\frac{\partial F_y}{\partial \kappa_i}\right)^k, \\
b_4 &= \left(\frac{\partial F_x}{\partial X_{0i}}\right)^k, & b_{14} &= \left(\frac{\partial F_y}{\partial X_{0i}}\right)^k, \\
b_5 &= \left(\frac{\partial F_x}{\partial Y_{0i}}\right)^k, & b_{15} &= \left(\frac{\partial F_y}{\partial Y_{0i}}\right)^k, \\
b_6 &= \left(\frac{\partial F_x}{\partial Z_{0i}}\right)^k, & b_{16} &= \left(\frac{\partial F_y}{\partial Z_{0i}}\right)^k, \\
b_7 &= \left(\frac{\partial F_x}{\partial X_i}\right)^k, & b_{17} &= \left(\frac{\partial F_y}{\partial X_i}\right)^k, \\
b_8 &= \left(\frac{\partial F_x}{\partial y_i}\right)^k, & b_{18} &= \left(\frac{\partial F_y}{\partial y_i}\right)^k, \\
b_9 &= \left(\frac{\partial F_x}{\partial Z_i}\right)^k, & b_{19} &= \left(\frac{\partial F_y}{\partial Z_i}\right)^k,
\end{aligned} \tag{3.28}$$

This can be written in matrix notation as:

$$\begin{bmatrix} V_{x_{ij}} \\ V_{y_{ij}} \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\ b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \end{bmatrix} \begin{bmatrix} \Delta\omega_i \\ \Delta\phi_i \\ \Delta\kappa_i \\ \Delta X_{0i} \\ \Delta Y_{0i} \\ \Delta Z_{0i} \end{bmatrix}$$

$$+ \begin{bmatrix} b_7 & b_8 & b_9 \\ b_{17} & b_{18} & b_{19} \end{bmatrix} \begin{bmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \end{bmatrix} = \begin{bmatrix} -F^k_{x_{ij}} \\ -F^k_{y_{ij}} \end{bmatrix}, \tag{3.29}$$

or more simply as

$$\begin{array}{ccccccc} V_{ij} & + & \dot{B}_{ij}\dot{\Delta}_i & + & \ddot{B}_{ij}\ddot{\Delta}_i & = & \epsilon_{ij}. \\ (2,1) & & (2,6)(6,1) & & (2,3)(3,1) & & (2,1) \end{array} \tag{3.30}$$

With control point i being present in each of the m images, the complete set of collinearity equations for j can be written as:

$$\underset{(2m,1)}{\begin{bmatrix} V_{1j} \\ V_{2j} \\ \vdots \\ V_{mj} \end{bmatrix}} + \underset{(2m,6m)}{\begin{bmatrix} \dot{B}_{1j} & & & \\ & \dot{B}_{2j} & & \\ & & \ddots & \\ & & & \dot{B}_{mj} \end{bmatrix}} \underset{(6m,1)}{\begin{bmatrix} \dot{\Delta}_1 \\ \dot{\Delta}_2 \\ \vdots \\ \dot{\Delta}_m \end{bmatrix}} + \underset{(2m,3)}{\begin{bmatrix} \ddot{B}_{1j} \\ \ddot{B}_{2j} \\ \vdots \\ \ddot{B}_{mj} \end{bmatrix}} \underset{(3,1)}{\ddot{\Delta}_j} = \underset{(2m,1)}{\begin{bmatrix} \epsilon_{1j} \\ \epsilon_{2j} \\ \vdots \\ \epsilon_{mj} \end{bmatrix}}, \tag{3.31}$$

which may also be presented more simply as:

$$V_j + \dot{B}_j\dot{\Delta} + \ddot{B}_j\ddot{\Delta}_j = \epsilon_j. \tag{3.32}$$

All points in the calibration grid (Figure 3.3) are obtained from each view, therefore equations 3.32 can be formed for each point and expressed as:

$$\underset{(2mn,1)}{\begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix}} + \underset{(2mn,6m)}{\begin{bmatrix} \dot{B}_1 \\ \dot{B}_2 \\ \vdots \\ \dot{B}_n \end{bmatrix}} \underset{(6m,1)}{\dot{\Delta}} + \underset{(2mn,3n)}{\begin{bmatrix} \ddot{B}_1 & & & \\ & \ddot{B}_2 & & \\ & & \ddots & \\ & & & \ddot{B}_n \end{bmatrix}} \underset{(3n,1)}{\begin{bmatrix} \ddot{\Delta}_1 \\ \ddot{\Delta}_2 \\ \vdots \\ \ddot{\Delta}_n \end{bmatrix}} = \underset{(2mn,1)}{\begin{bmatrix} \ddot{\Delta}_1 \\ \ddot{\Delta}_2 \\ \vdots \\ \ddot{\Delta}_n \end{bmatrix}}, \tag{3.33}$$

or more simply expressed as

$$V + \dot{B}\dot{\Delta} + \ddot{B}\ddot{\Delta} = \epsilon. \tag{3.34}$$

With all the collinearity equations created, a set of equations for the measured control point values must be created. The residual error $(V_{X_C}, V_{Y_C}, V_{Z_C})$ between a measured control point (X_C^0, Y_C^0, Z_C^0) and its true value (X_C, Y_C, Z_C) can be stated as:

$$\begin{aligned} X_C &= X_C^0 + V_{X_C} \\ Y_C &= Y_C^0 + V_{Y_C} \\ Z_C &= Z_C^0 + V_{Z_C} \end{aligned} \tag{3.35}$$

Ideally, the current correction $(\Delta X_C, \Delta Y_C, \Delta Z_C)$ to an approximated control point (X_C^k, Y_C^k, Z_C^k) should be equal to the true value (X_C, Y_C, Z_C), or:

$$\begin{aligned} X_C &= X_C^k + \Delta X \\ Y_C &= Y_C^k + \Delta Y \\ Z_C &= Z_C^k + \Delta Z \end{aligned} \tag{3.36}$$

Combining and rearranging equations 3.35 and 3.36, a matrix equation for control point j can be created:

$$\begin{bmatrix} V_{X_{Cj}} \\ V_{Y_{Cj}} \\ V_{Z_{Cj}} \end{bmatrix} - \begin{bmatrix} \Delta X_{Cj} \\ \Delta Y_{Cj} \\ \Delta Z_{Cj} \end{bmatrix} = \begin{bmatrix} X_{Cj}^k & - & X_{Cj}^0 \\ Y_{Cj}^k & - & Y_{Cj}^0 \\ Z_{Cj}^k & - & Z_{Cj}^0 \end{bmatrix} \quad \text{or,} \quad \begin{array}{ccccc} \ddot{V}_j & - & \ddot{\Delta}_j & = & \ddot{C}_j. \\ (3,1) & & (3,1) & & (3,1) \end{array} \tag{3.37}$$

All n control points can then be expressed as:

$$\underset{(3n,1)}{\begin{bmatrix} \ddot{V}_1 \\ \ddot{V}_2 \\ \vdots \\ \ddot{V}_n \end{bmatrix}} - \underset{(3n,1)}{\begin{bmatrix} \ddot{\Delta}_1 \\ \ddot{\Delta}_2 \\ \vdots \\ \ddot{\Delta}_n \end{bmatrix}} = \underset{(3n,1)}{\begin{bmatrix} \ddot{C}_1 \\ \ddot{C}_2 \\ \vdots \\ \ddot{C}_n \end{bmatrix}} \quad \text{or,} \quad \ddot{V} - \ddot{\Delta} = \ddot{C}. \tag{3.38}$$

The final set of collinearity 3.32 and observation 3.38 equations are:

$$\begin{array}{lcl} V + \dot{B}\dot{\Delta} + \ddot{B}\ddot{\Delta} & = & \epsilon, \\ \ddot{V} \qquad\quad - \quad \ddot{\Delta} & = & \ddot{C}. \end{array} \tag{3.39}$$

Combining the two system of equations yields:

$$\begin{bmatrix} V \\ \ddot{V} \end{bmatrix} + \begin{bmatrix} \dot{B} & \ddot{B} \\ 0 & -I \end{bmatrix} \begin{bmatrix} \dot{\Delta} \\ \ddot{\Delta} \end{bmatrix} = \begin{bmatrix} \epsilon \\ \ddot{C} \end{bmatrix} \quad \text{or,} \quad \bar{V} + \bar{B}\Delta = \bar{C}. \tag{3.40}$$

This final set of equations is of the from given by the Gauss-Markov model 3.10 and a bundle adjustment solution is therefore given by equation 3.11 as:

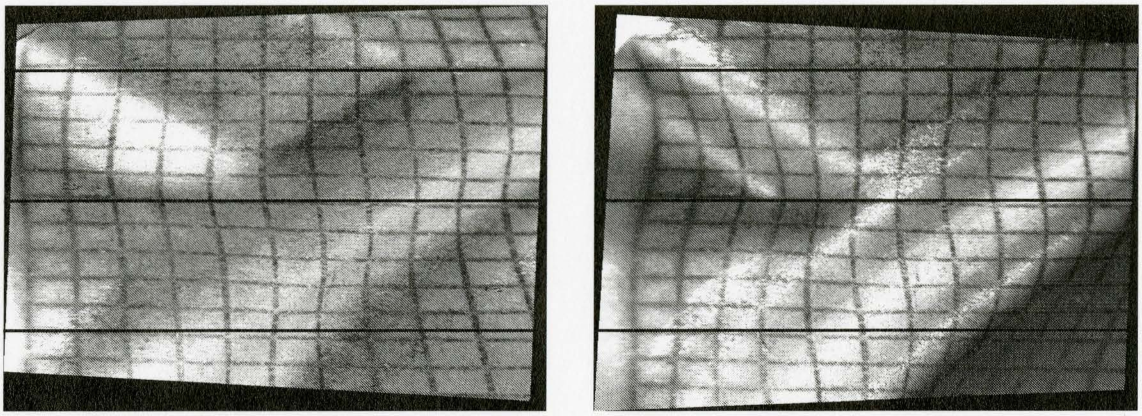
$$\Delta = (\bar{B}^T \bar{W} \bar{B})^{-1} \bar{B}^T \bar{W} \bar{C} \tag{3.41}$$

where W is the weight matrix. Iteration can be performed to reach the desired tolerance.

This method of exterior orientation is applied in the following manner. Firstly, the calibration grid is imaged from at least one pose. The calibration error should decrease as the number of poses is increased. Grid intersections points are determined using the methods described in section 2. With the lens distortion known, the distorted image coordinates (x', y') are corrected using equations 3.7 and 3.8. An initial approximation for the exterior orientation and the focal length is determined using the calibration method presented by Tsai [39], described in section 3.2.2. Normally the focal length can be added as an unknown parameter in the bundle adjustment algorithm described above, but this is not possible for a coplanar set of control points. A 3-D test field of control points is required in order to formulate a problem with a defined minimum using the collinearity conditions. For this reason the focal length is solely determined using the calibration method from Tsai [39]. A 3-D test field of points would allow the focal length to be added in the bundle adjustment, resulting in a more accurate calibration.

3.2.7 Rectification of Stereo Pairs

The stereo matching procedure described in section 2.6 assumes the pinhole stereo camera setup as shown in figure 1.1. Two identical cameras that have the same



(a) Left Image

(b) Right Image

Figure 3.6: Rectification of stereo pair

orientation but with a difference of position in the x direction. This configuration ensures that conjugate points lie along the same epipolar line. The results from interior calibration have shown that the two cameras used in this thesis have a slightly different set of parameters. The stereo head also has a vergence angle of about 30 degrees and therefore does not satisfy the desired setup. This problem can be overcome with the process of image rectification.

The process of rectification can be thought of as acquiring a new pair of images of the same scene by rotating the cameras about their optical centers and changing their interior parameters. The process aligns image planes so that they have the same orientation, are coplanar and are separated by a baseline distance. The end result is a new set of exterior and interior parameters and a planar homography, a 2D to 2D mapping that transforms image points from the old image plane to the new image plane. Stereo reconstruction can be done using this new set of camera parameters, but results are of lesser accuracy than using original parameters. The reason is that the homography maps points into non integer positions in the new image plane. Image processing algorithms generally require regularly spaced images and therefore the rectified image must be re-sampled using bilinear interpolation. For this thesis,

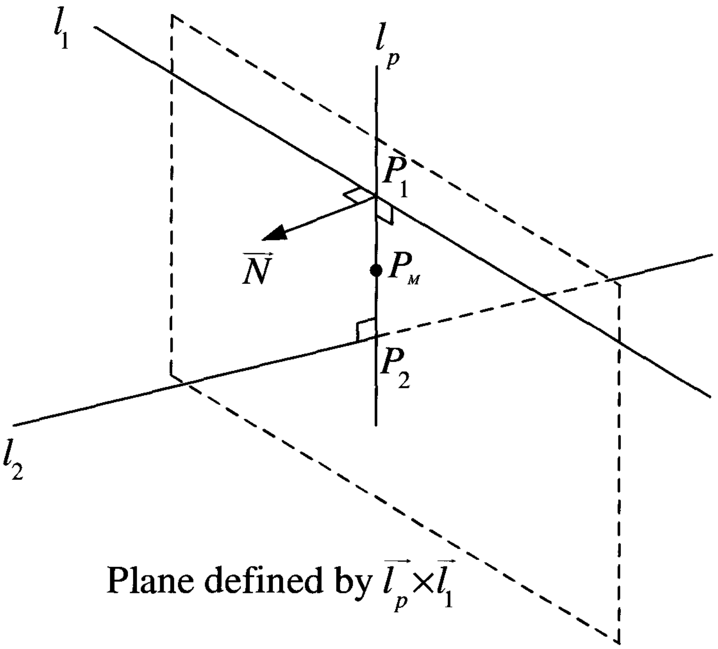


Figure 3.7: Triangulation using two lines

only the rectification homography is used, it provides the epipolar constraint needed for the matching process from section 2.6. Reconstruction of 3D points is done using the original images and parameters. A compact algorithm for the rectification of a pair of stereo images has been published by Fusiello et al. [8]. The details will not be presented but the result of rectifying a pair of stereo images can be seen in figure 3.6. The three horizontal lines overlayed on the image emphasize the epipolar geometry.

3.2.8 Stereo Triangulation

A 3D point imaged in both the right and left cameras can be reconstructed through triangulation. A conjugate pair of 2D points can be used to create two rays which should ideally intersect at the 3D point. Due to sensor noise, sensor quantization and calibration error these rays are unlikely to intersect. One way of approximating the intersection is to use the mid-point method. Referring to figure 3.7, this method assumes that the correct point is the point closest to both rays. This point is the mid point of the line segment perpendicular to both rays and of shortest distance. The collinearity condition expressed in equation 3.4 can be rearranged to give:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \frac{1}{\lambda} R^{-1} \begin{bmatrix} x_i - x_p \\ y_i - y_p \\ 0 - f \end{bmatrix} \tag{3.42}$$

The camera center and rotation matrix are chosen to be one of the camera poses determined from the exterior orientation calibration. Any pose will suffice but all stereo points must be reconstructed using that pose. Using the 2D left and right points from a conjugate pair, equation 3.42 can be used to the write the equations of two lines in parametric form:

$$\begin{aligned} l_1(X,Y,Z) &= (X_{0,L}, Y_{0,L}, Z_{0,L}) + \frac{1}{\lambda_L} R_L^{-1} (x_L - x_{p,L}, y_L - y_{p,L}, -f), \\ &= t_L + k_1 \overrightarrow{l_L},, \qquad (3.43) \\ l_2(X,Y,Z) &= (X_{0,R}, Y_{0,R}, Z_{0,R}) + \frac{1}{\lambda_R} R_R^{-1} (x_R - x_{p,R}, y_R - y_{p,R}, -f), \\ &= t_R + k_2 \overrightarrow{l_R},, \qquad (3.44) \end{aligned}$$

where k_1 and $k_2 \in [-\infty, \infty]$. The cross product of vectors $\overrightarrow{l_L}$ and $\overrightarrow{l_R}$ can be used to form the unit vector:

$$\overrightarrow{l_p} = \frac{\overrightarrow{l_L} \times \overrightarrow{l_R}}{||\overrightarrow{l_L} \times \overrightarrow{l_R}||} \tag{3.45}$$

The cross product of $\overrightarrow{l_p}$ with $\overrightarrow{l_L}$ defines the normal vector, $\overrightarrow{N}$, of the plane in which the line l_1 and vector $\overrightarrow{l_p}$ lie:

$$\overrightarrow{N} = \frac{\overrightarrow{l_L} \times \overrightarrow{l_p}}{||\overrightarrow{l_L} \times \overrightarrow{l_p}||} \tag{3.46}$$

The point t_L resides in this plane and therefore the plane can be completely described as:

$$N_X X + N_Y Y + N_Z Z + d = 0, \tag{3.47}$$

where

$$d = \overrightarrow{N} \cdot t_L.$$

The intersection point P_2, of the line l_2 with the plane can be calculated by substituting the components of equation 3.44 into equation 3.47 and solving for the value of k_2:

$$k_2 = \frac{-\overrightarrow{N} \cdot t_R - d}{\overrightarrow{N} \cdot \overrightarrow{l_R}}. \tag{3.48}$$

P_2 is then given by:

$$P_2 = t_R + k_2 \overrightarrow{l_R}. \tag{3.49}$$

Using this point the line l_p can now be defined as:

$$l_p(X, Y, Z) = P_2 + k_p \overrightarrow{l_p}, \tag{3.50}$$

where $k_p \in [-\infty, \infty]$. The point P_1 is the intersection of the lines l_1 and l_p. Equating the components of the parametric line equations gives:

$$P_{2_X} + k_p l_{pX} = t_{L_X} + k_1 l_{L_X}, \tag{3.51}$$

$$P_{2_Y} + k_p l_{pY} = t_{L_Y} + k_1 l_{L_Y}, \tag{3.52}$$

$$P_{2_Z} + k_p l_{pZ} = t_{L_Z} + k_1 l_{L_Z} \tag{3.53}$$

Using equations 3.51 and 3.52 to solve for k_1 gives:

$$k_1 = \frac{l_{pX}(P_{2_Y} - t_{L_Y}) - l_{pY}(P_{2_X} - t_{L_X})}{l_{L_Y} l_{pX} - l_{L_X} l_{pY}}. \tag{3.54}$$

The point P_1 can be now be calculated using equation 3.43, thus the midpoint P_M is:

$$P_M = 0.5(P_1 + P_2). \tag{3.55}$$

3.3 Probe Head Calibration

3.3.1 Introduction

At each stereo head pose within the CMM workspace any measured 3-D data will be relative to a unique Stereo Coordinate System (SCS). As the CMM is moved around the workspace and the probe head pan/tilt angles are changed, the 3-D measurements between the different SCSs are not related. A Rigid Body Transformation (RBT) must be determined from each SCS into the CMM coordinate system. For convenience the CMM coordinate system will be referred to as the World Coordinate System (WCS). The unknown transformation from SCS to the WCS is comprised of two separate coordinate transformations. One is the transformation from the SCS to the coordinate system at the end of the CMM ram and the other is from the ram coordinate system to the WCS. The Ram Coordinate System (RCS) is approximately centered at the end of the boom arm, its exact position and orientation is determined by the CMM scale values and CMM error map.

Figure 3.8 describes the situation, it includes a calibration object that will be used to link coordinate systems together. The RBT from the RCS to the WCS for pose number i is given by H_{Ri}. The CMM used in this thesis was of a small size and its error compensation map showed no rotational error. This means that the H_R is purely a translation for each CMM position within the workspace. However, larger CMMs typically have a rotational error unique to each position and this would therefore introduce a rotational component into H_R. Referring to the figure again, the RBT from the SCS to the RCS for position i is given by H_{CRi}. The H_{CR} is almost purely rotational, the rotation being centered at a probe head pivot point, but due to small machining errors in the probe head locking mechanism, each H_{CR} contains a unique small translational component. This means that every pan/tilt combination necessary to image a scene must be calibrated individually. Calibration

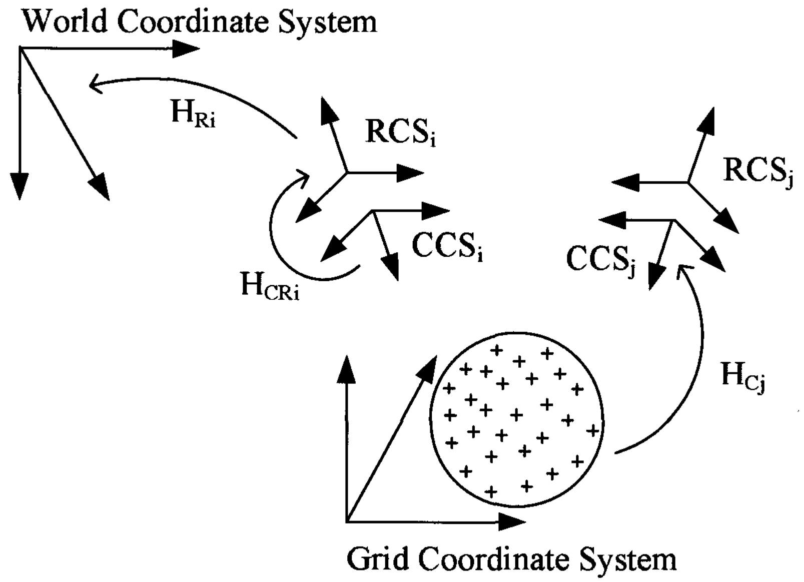


Figure 3.8: RBTs involved with probe head calibration

of pan angles and then tilt angles separately is not possible. To help link coordinate systems together a calibration object with control points on it can be used. The RBT from SCS i to a coordinate system defined by the control points will be referred to as H_{Ci}.

Normally the PH10 probe head must be calibrated for use with a touch trigger probe. This is performed using a very accurate 1" calibration sphere. Using an initial guess for each probe tip offset, sphere points can be gathered from all necessary probe head orientations. By performing geometric fitting to the sphere points it is possible to refine the unique tip offset for each probe head position. This process is repeated until sufficient accuracy is obtained. Unfortunately this type of calibration is not useful because it only determines a translational offset. The stereo camera system requires a rotational component because it acquires data by line of sight not direct contact.

The probe head camera calibration problem is similar to the "Hand/Eye" calibration problem in robotics. This problem involves determining the RBT between the gripper at the end of a robot and a camera rigidly mounted on it. Calibration of one of the probe-head/camera orientations is equivalent to the hand/eye problem. So, calibration of M probe head positions is equivalent to M hand/eye calibration problems. Unfortunately, solutions to the hand/eye problem such as that given by Tsai [40] cannot be used because they require the gripper/camera rigid body to rotate and view a calibration object from different orientations. A minimum of three different orientations creates a locked system and a solution can be determined. For a fixed PH10 probe orientation, only translation is available for the CMM used in this thesis. A larger CMM would have some rotational component to each H_{Ri}, but it would be so small that the locked system would be unreliable.

One method of solving the probe calibration problem is to use point set registration techniques. Registration in this context refers to finding the optimal linear transform that maps 3-D points in one set to corresponding points in another set. It is also referred to as the absolute orientation problem and as the orthogonal procrustes problem. The two-view point set regsitration problem and the N-view point set registration problem are described in the next two sections.

3.3.2 2-View Point Set Registration

Assuming independent zero mean Gaussian error for point data, the 2-View registration problem can be stated as:

$$e(R,T) = \frac{1}{n}\sum_{i=1}^{n} ||y_i - (Rx_i + T)||^2, \tag{3.56}$$

where $P_1 = \{x_1, x_2, \ldots, x_n\}$ and $P_2 = \{y_1, y_2, \ldots, y_n\}$ are two corresponding point sets in 3-D space, $n > 2$, R and T belong to the optimal RBT and $||x||$ is the usual Euclidean norm. There are two main approaches to solve this problem, comparisons have shown both give almost identical results [18]. The first approach uses unit quaternions and was presented by Horn [10]. The second approach which uses the preferred rotation matrix notation was presented by Arun et al. [30] and then further refined by Umeyama [41]. The solution by Umeyama requires the mean vectors μ_{P_1}, μ_{P_2} and variances around mean vectors $\sigma^2_{P_1}$, $\sigma^2_{P_2}$ for P_1 and P_2, as well as the covariance matrix of P_1 and P_2, $\Sigma_{P_1P_2}$:

$$\mu_{P_1} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.57}$$

$$\mu_{P_2} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3.58}$$

$$\sigma^2_{P_1} = \frac{1}{n}\sum_{i=1}^{n} ||x_i - \mu_{P_1}||^2 \tag{3.59}$$

$$\sigma^2_{P_2} = \frac{1}{n}\sum_{i=1}^{n} ||y_i - \mu_{P_2}||^2 \tag{3.60}$$

$$\Sigma_{P_1P_2} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \mu_{P_2})(x_i - \mu_{P_1})^T \tag{3.61}$$

With the singular value decomposition of $\Sigma_{P_1P_2}$ given by UDV^T, $(D = diag(d_i), \quad d_1 \geq d_2 \geq \ldots \geq d_m \geq 0)$, the final RBT parameters are given by:

$$R = USV^T, \tag{3.62}$$

$$T = \mu_y - R\mu_x, \tag{3.63}$$

where S is the identity matrix I except in the cases where $\Sigma_{P_1P_2}$ is of full rank and $\det(\Sigma_{P_1P_2})$ is less than zero or when the rank of $\Sigma_{P_1P_2}$ is $m-1$ and $\det(U)\det(V)$ is equal to -1. In these cases S is equal to $diag(1, 1, \ldots, -1)$. If the rank of $\Sigma_{P_1P_2}$ is less than $m-1$ then the points in both sets are collinear and a unique solution is not possible.

Moving back to the problem of probe head calibration, the registration technique described above can be used to determine H_C for all poses. Referring to figure 3.8, the calibration object can be imaged from all necessary poses and the 3-D locations of the control points determined. Choosing the control point coordinate system to be the same as the first SCS, H_{Ci} can be determined by registration of control points common to both the 1st and ith pose. Any pose that does not share $n > 2$ control points with the 1st pose can be registered in a sequential manner using a pose that has control points which overlap with both. This process works well with small registration sequences, but a large mismatch error is usually present between any SCSs that were registered in long sequences. This mismatch problem can be overcome with N-View point set registration algorithms. These algorithms register all point sets together simultaneously, distributing measurement error globally.

3.3.3 N-View Point Set Registration

Before the N-View registration problem can be stated, some point set notation must be defined. Assume that there are M points sets each taken from a different pose, $P^1 \dots P^M$, where $P^\alpha = \{x_1^\alpha, \dots, x_{N^\alpha}^\alpha\}$ and N^α is the number of points for pose α. For ease of notation, let a rigid body transformation $H^\alpha = \{R^\alpha, T^\alpha\}$ be applied to a 3-D point x using $H^\alpha * x = R^\alpha x + T^\alpha$. Let the overlap of P^α with P^β be denoted as $O^{\alpha\beta} \subset P^\alpha$ where $O^{\alpha\beta} = \{x_1^{\alpha\beta}, \dots, x_{N^{\alpha\beta}}^{\alpha\beta}\}$. Each point $x_i^{\alpha\beta}$ is matched with $x_i^{\beta\alpha} \subset P^\beta$, therefore $N^{\alpha\beta} = N^{\beta\alpha}$. The global point set registration problem can now be stated as minimization of:

$$E(H^1, \dots, H^M) = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{\imath=1}^{N^{\alpha\beta}} \|H^\alpha * x_i^{\alpha\beta} - H^\beta * x_i^{\beta\alpha}\|^2 \tag{3.64}$$

where the first RBT, H^1, is fixed as the identity matrix to properly define the function.

A recent comparison of three solutions to the global registration problem was given by Cunnington et al. [42]. It concluded that the mean shape method presented by Pennec [45] consistently out performed others for both speed and accuracy of registration. Pennec gives a lengthy formal definition for the mean point set, for this application it can be defined using all the control points:

$$P^\mu = \{x_1^\mu, \dots, x_{N^\mu}^\mu\} \tag{3.65}$$

where N^{μ} is the total number of control points,

$$x_r^{\mu} = \sum_{r=1}^{N^{\mu}} \frac{\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} w(i,\alpha,\beta,r) H^{\alpha} * x_i^{\alpha\beta}}{\sum_{\alpha=1}^{M}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} w(i,\alpha,\beta,r)}, \tag{3.66}$$

$$w(i,\alpha,\beta,r) = \begin{cases} 1 & \text{if } x_r^{\mu} \text{ and } x_i^{\alpha\beta} \text{ correspond} \\ 0 & \text{Otherwise} \end{cases},$$

and H^{μ} initially consists of the zero rotation and zero translation. The global point set registration problem is solved in an iterative manner by registering each point set to the mean point set using Umeyama's [41] 2-View point set registration technique from above. Each iteration will minimize the function:

$$E^{\alpha}(f^{\alpha}) = \sum_{r=1}^{N^{\mu}}\sum_{\beta=1}^{M}\sum_{i=1}^{N^{\alpha\beta}} w(i,\alpha,\beta,r) \| x_r^{\mu} - H^{\alpha} * x_i^{\alpha\beta} \|^2 \tag{3.67}$$

After equation 3.67 is minimized for each pose, a new mean point set is defined using the new estimates for H^{α} and equation 3.66. This process is repeated until E is sufficiently small. The sequential probe calibration method initially suggested used the first SCS as the control point coordinates system, the global registration method defines the control point coordinate system with the use of the mean point set. Although there is no guarantee of convergence for the N-View point set registration algorithm, Cunnington et al. [42] states that only cases with extreme point errors result in non-convergence, this was also observed in this work.

3.3.4 Probe Head Calibration Algorithm

With the N-view point registration method outlined, the probe head calibration can now be described by the following steps.

1. Gather 3-D control points from the calibration object for all necessary poses. There must be at least 3 poses α, β, γ, in which the SCS position was changed but not its orientation. These poses should also have a minimum of three control points common to all of them, i.e. $N^{\alpha\beta\gamma} > 3$. The H_B for all poses is given by the CMM controller.

2. Use the N-View registration method to determine H_{Ci} for all poses. The control point coordinate system is now defined by the mean point set.

3. Define two sets of points $P_1 = \{H_{R\alpha} * [0,0,0]^T, H_{R\beta} * [0,0,0]^T, H_{R\gamma} * [0,0,0]^T\}$ and $P_2 = \{H_{C\alpha} * [0,0,0]^T, H_{C\beta} * [0,0,0]^T, H_{C\gamma} * [0,0,0]^T\}$ which are the origins of the RCS and SCS respectively for poses α, β and γ. Use the 2-View registration method to register P_1 and P_2 together. Denote the solution to this registration problem as $H_G = \{R_G, T_G\}$. This step aligns the CMM and mean point set coordinate systems, the rotation of H_{CRi} is now determined, $R_{CRi} = R_G$. This rotation is the same for all poses because the CMM contains no rotational error. The only remaining unknown is the translational component of H_{CRi}.

4. Determine the translational component of H_{CRi} using: $T_{CRi} = H_G * H_{Ci} * [0,0,0]^T - H_{Ri} * [0,0,0]^T$.

A point x from probe orientation i can now be transformed into the WCS using:

$$x'_i = H_R * H_{CRi} * H_{Ci}x_i, \tag{3.68}$$

where H_R was given by the CMM at the time of measurement. This calibration technique would not work if the CMM had rotational error. The calibration is also weak at Step 3 because it only uses three points to align the CMM and mean point set coordinate system.

Chapter 4

Results and Discussion

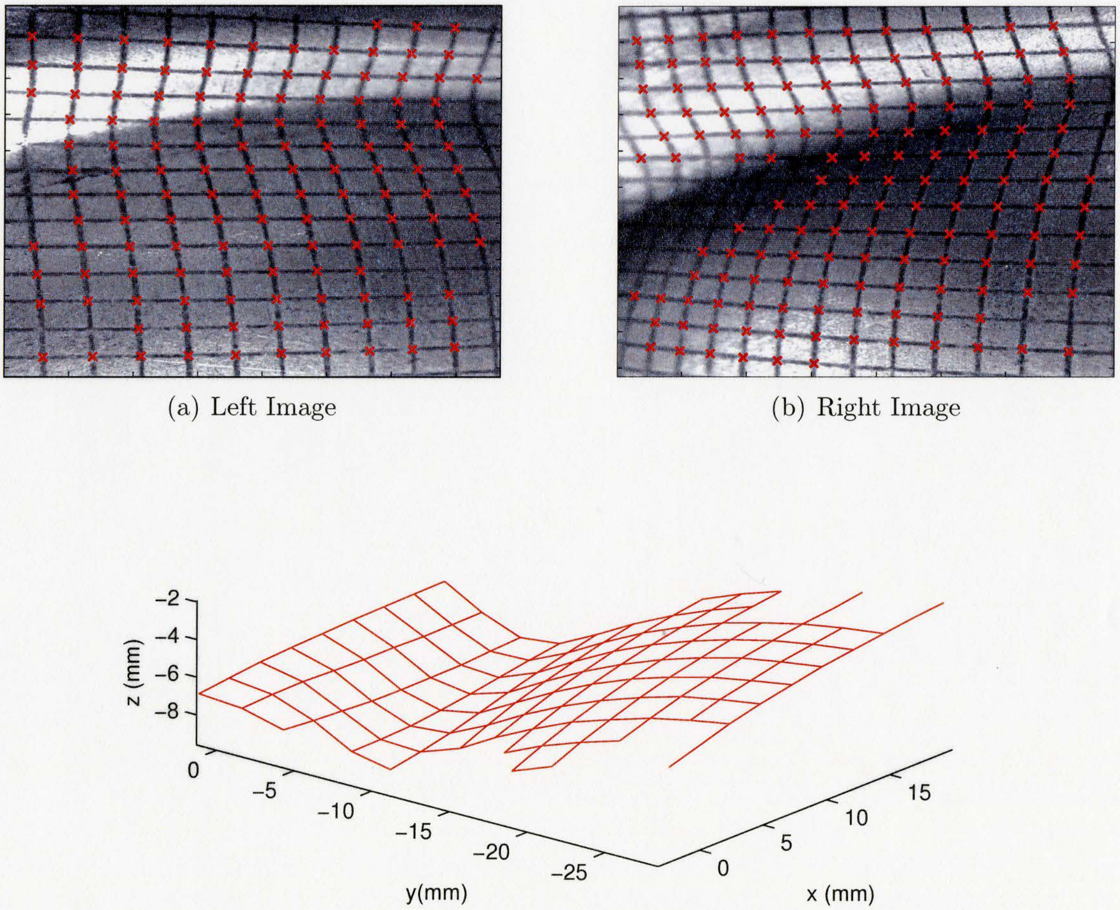
This chapter will present the accuracy of the system and discuss error sources. The final accuracy results are summarized in table 4. Figure 4.1 shows a typical stereo pair of images and their corresponding 3-D reconstruction.

Stereo Head Accuracy	$\pm 1.1\ \mu\text{m}\ (2\sigma)$
Strain Accuracy	$\pm 0.064\%\ (2\sigma)$
3-D Reconstruction Accuracy	$\pm 17.1\ \mu\text{m}\ (2\sigma)$

Table 4.1: Accuracy results

4.1 System Accuracy

Surface strain values are calculated using points from the same pose. Therefore the accuracy of surface strain measurements is associated with the stereo head accuracy, not the accuracy of the entire system. However, the 3-D reconstruction is comprised of points from many poses and therefore its accuracy is given by the system accuracy.



(a) Left Image

(b) Right Image

(c) Wire mesh reconstruction

Figure 4.1: Stereo pair and 3-D reconstruction

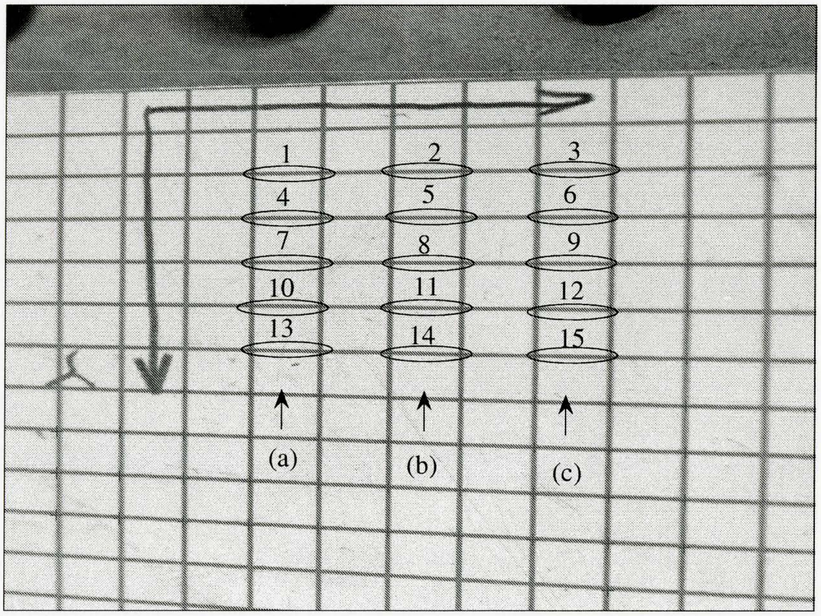


Figure 4.2: Grid links used to determine strain and system accuracy

The following sections describe the experiments used to determine the stereo and system accuracy.

4.1.1 Stereo Head Accuracy

The flat piece of sheet aluminum shown in figure 4.2 was imaged from 12 different poses. This is the same type of lithographically printed grid as used for the camera calibration. The grid lines have an approximate spacing of 2.5 mm. The relative accuracy of the stereo head was determined by measuring the Euclidean lengths of the 15 circled line segments. Four images were taken with the circled line segments in the image corners, the other 8 images containing the circled line segment's were taken at 15 deg tilt, and at 45 degree pan increments. Each line segment is defined by two grid line intersection points. These intersection points were determined by manually measuring the four points at which the grid line edges intersect and then calculating the centroid of these points. The edge intersection point measurements

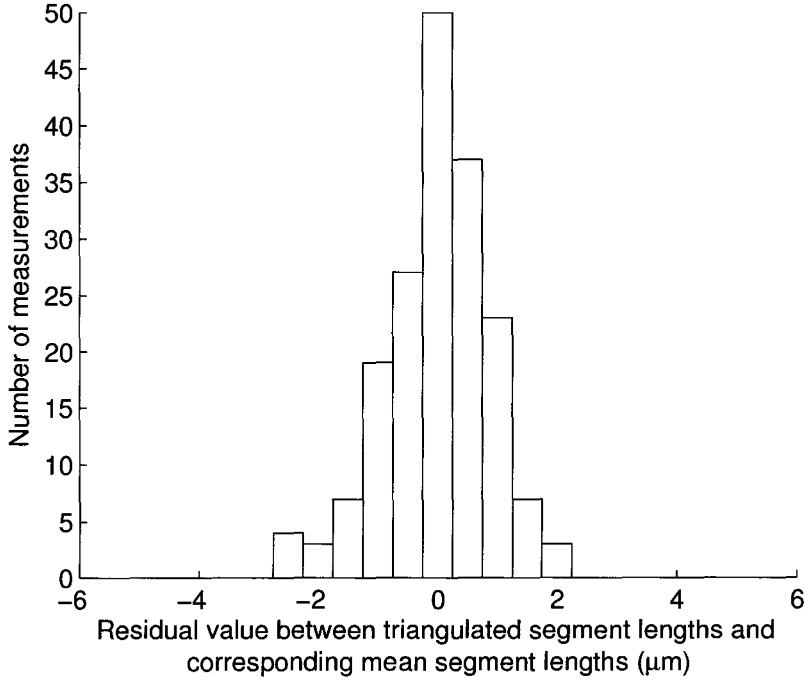


Figure 4.3: Histogram of residuals between stereo lengths and their corresponding means

were performed using an optical microscope of $\pm 2.5\,\mu$m resolution. In order to not bias the result in any way, each measured grid intersection point was only used in one line segment. Bias due to the direction of the line segments should not matter because the stereo head was oriented at different angles. The optical microscope is assumed to have no bias in any direction.

Figure 4.4 shows the segment lengths for the microscope and stereo vision measurements. The y scale indicates the length of the line segments and the x scale values are the labels for the line segments corresponding to those in figure 4.2. The red points marked with a cross are the lengths determined by the optical microscope. The blue points are the individual measurements for the line segments, each line segment has 12 measurements corresponding to the 12 different views. The blue points marked with a cross are the mean of the lengths determined from the 12 different views.

The red error bars encompass a confidence interval of two for the lengths measured with the optical microscope. A confidence interval describes a one standard deviation region where measurements lie 68% of the time. A confidence interval of two represents the region over which 95% will lie and is the norm for describing uncertainty in industrial metrology. The confidence interval of a microscope measurement can be determined using the microscope resolution and a conversion factor defined by the International Organization for Standardization (ISO) [7]. For a symmetric rectangular distribution such as the scale on a microscope, the ISO defines one standard deviation to be the resolution of the measurement device divided by $\sqrt{3}$. Using this conversion, the confidence interval for a point measured by the optical microscope is given by $2.5/\sqrt{3} = 1.44\,\mu$m (1σ). A confidence interval of two for the length of a line segment should be no greater than $4 \times 1.44 = 5.76\mu$m (2σ). It can also be argued that the optical microscope would have a Gaussian error distribution, which would result in a smaller confidence interval. For a Gaussian error distribution the ISO defines one standard deviation to be the resolution of the measurement device divided by 2. To determine the confidence interval for the summation of two Guassian errors, the confidence intervals are added in quadrature:

$$\sigma_l = \sqrt{\sigma_a^2 + \sigma_b^2}, \tag{4.1}$$

where σ_l, σ_a and σ_b are the confidence intervals for the segment length, first point and second point respectively. Using equation 4.1 the accuracy of a measured segment is $(\sqrt{2} \times 2.5) = 3.54\mu\text{m}$ (2σ). This lower bound on the microscope accuracy is displayed using red error bars in figure 4.4.

The blue error bars show a confidence interval of two and are uniquely determined using the stereo data associated with each line segment. It can be seen that the error bars for all microscope and stereo measurements overlap for a confidence interval of two, or 95% of the time. Figure 4.3 shows a histogram plot of the all the differences between stereo length data shown with blue points in figure 4.4 and their respective means shown with blue crosses in figure 4.4. It can be seen that the distribution is Gaussian in nature. It is assumed that this is the result of the quadrature addition of two Gaussian distributions, corresponding to the error from two points. The confidence interval of the distribution shown in figure 4.3 is the standard deviation for all length measurements and is equal to $0.8\,\mu\text{m}$ (1σ). Using the quadrature equation the accuracy of a stereo point is determined as $2 \times 0.8/\sqrt{2} = 1.1\,\mu\text{m}$ (2σ).

4.1.2 Full System Accuracy

The accuracy of the entire system was determined using the same poses and points as described in section 4.1.1. Using the pan/tilt and CMM information, all points were reconstructed in the WCS creating 30 point clouds. Each of the 30 point clouds containing 12 points corresponding to the 12 different views. Figure 4.5 shows the segment lengths with the microscope measurements in red. The black points are the

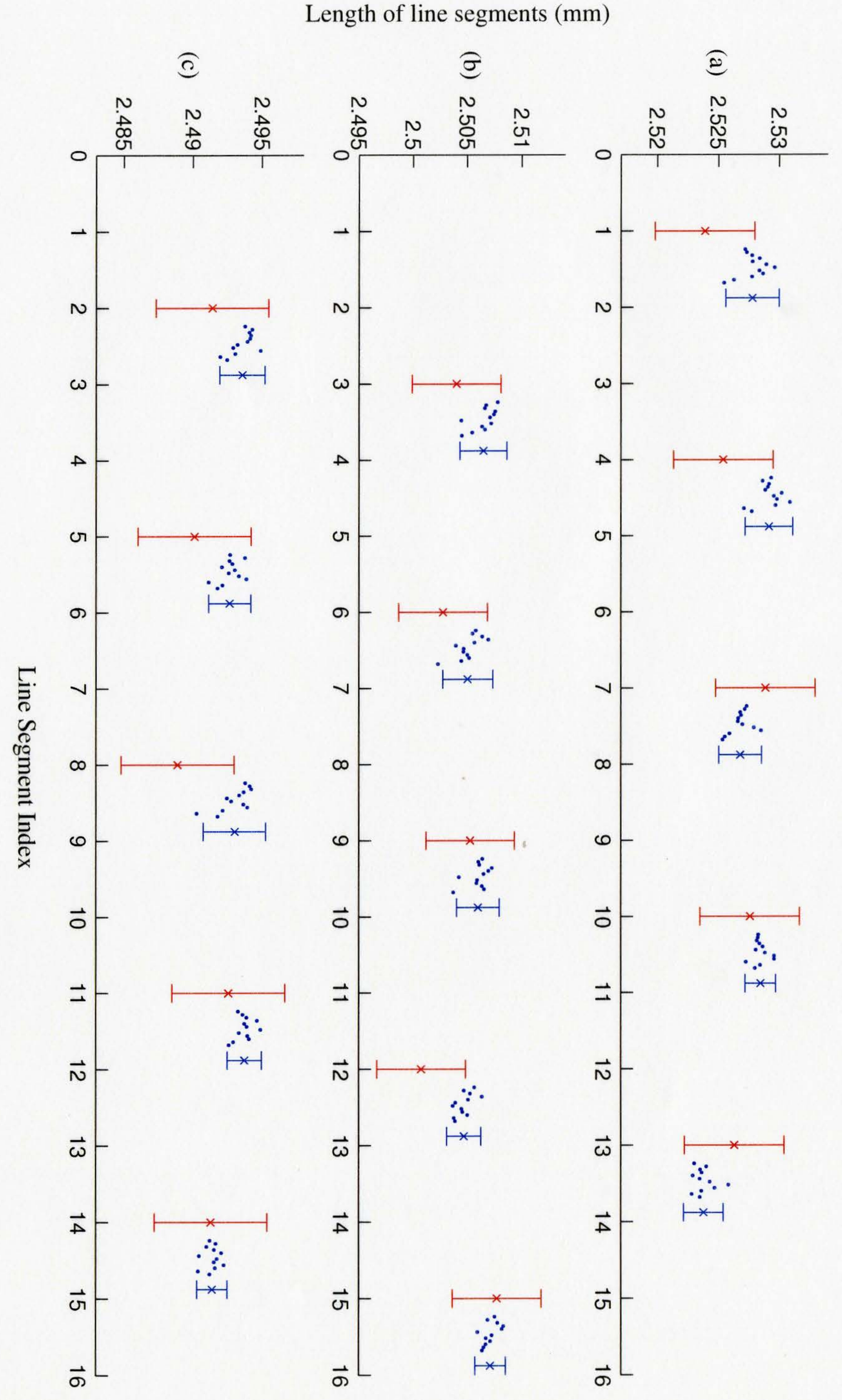


Figure 4.4: Segment length measurements using microscope and stereo system

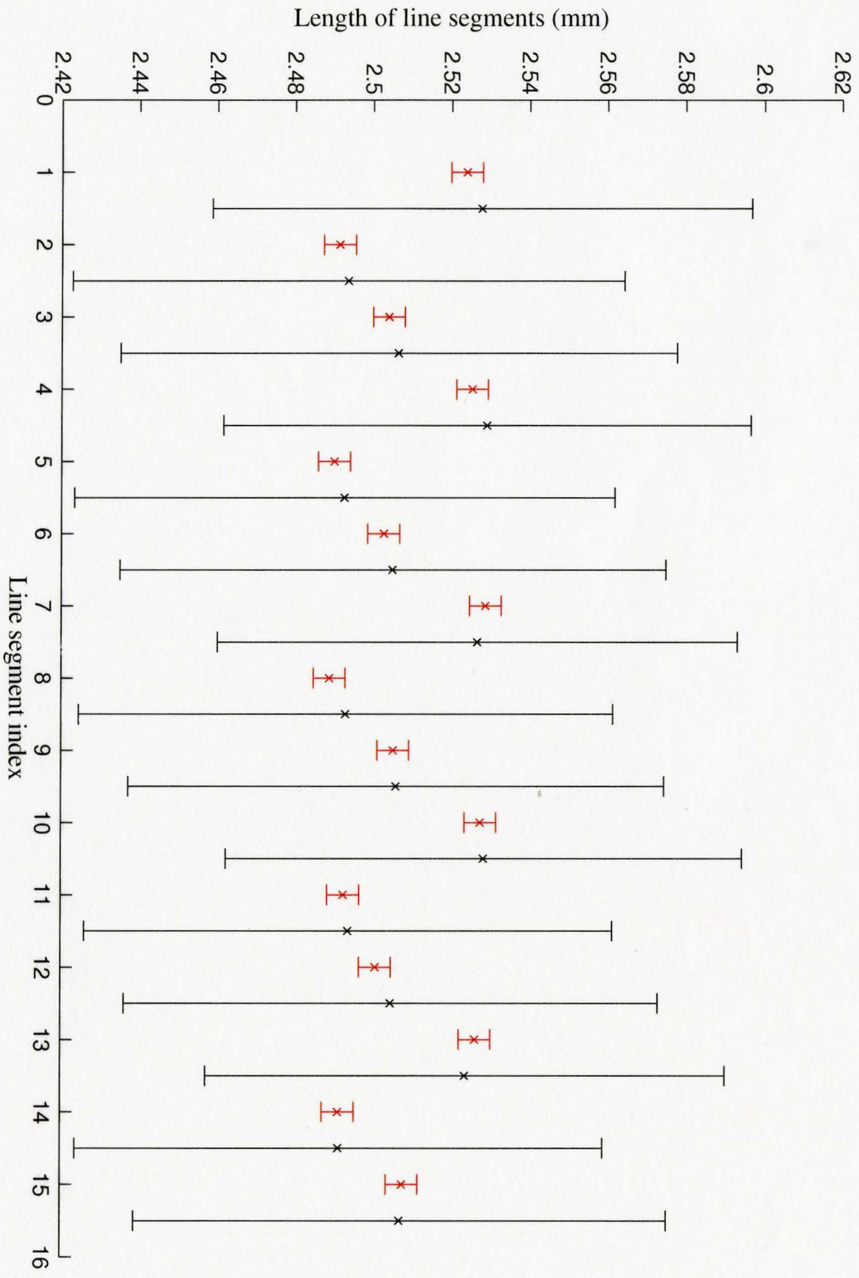


Figure 4.5: Microscope, stereo head and system measurements with error bars

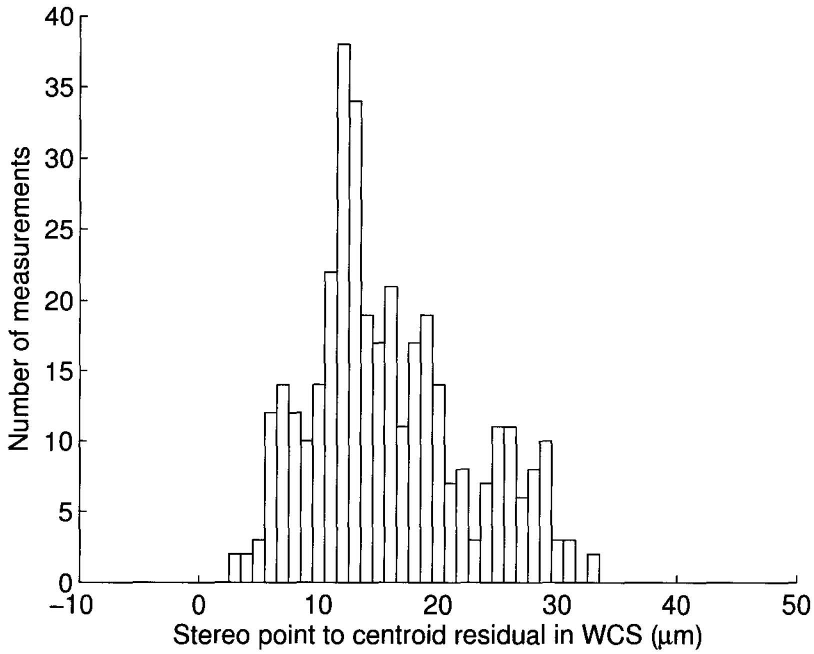


Figure 4.6: Histogram of residuals between stereo points and their corresponding means in WCS

segment lengths calculated using the centroid of the point clouds. The error bars in figure 4.5 show confidence intervals of two and were calculated using the standard deviation of the point clouds in the WCS. These confidence intervals illustrate that the system accuracy is much worse than the stereo head accuracy.

Figure 4.6 shows a histogram of residuals between stereo points and their corresponding means for all point clouds. The histogram is one sided because 1-D plots of 3-D residuals can only be stated with a magnitude not a direction. It can be seen that the distribution is biased which indicates a systematic error. The error is largely due to the probe calibration procedure, especially the final step which performed weak calibration between the mean point set and the CMM coordinate system. Using the distribution shown in figure 4.6, the accuracy of the 3-D stereo reconstruction was calculated to be 17.1 μm (2σ).

4.2 Strain Accuracy

This section presents the results of measuring two sheet metal dome parts. The first dome had a low level of surface strain whereas the second dome had a high level. The 3-D reconstruction of the dome surfaces consists of points from many different camera poses. An intersection point that was detected in more than one pose creates a small cloud of points. The centroid of this cloud is displayed in the 3-D reconstruction. Correspondingly, any strain values for a particular grid intersection that were calculated from more than one pose are averaged to give the final strain value.

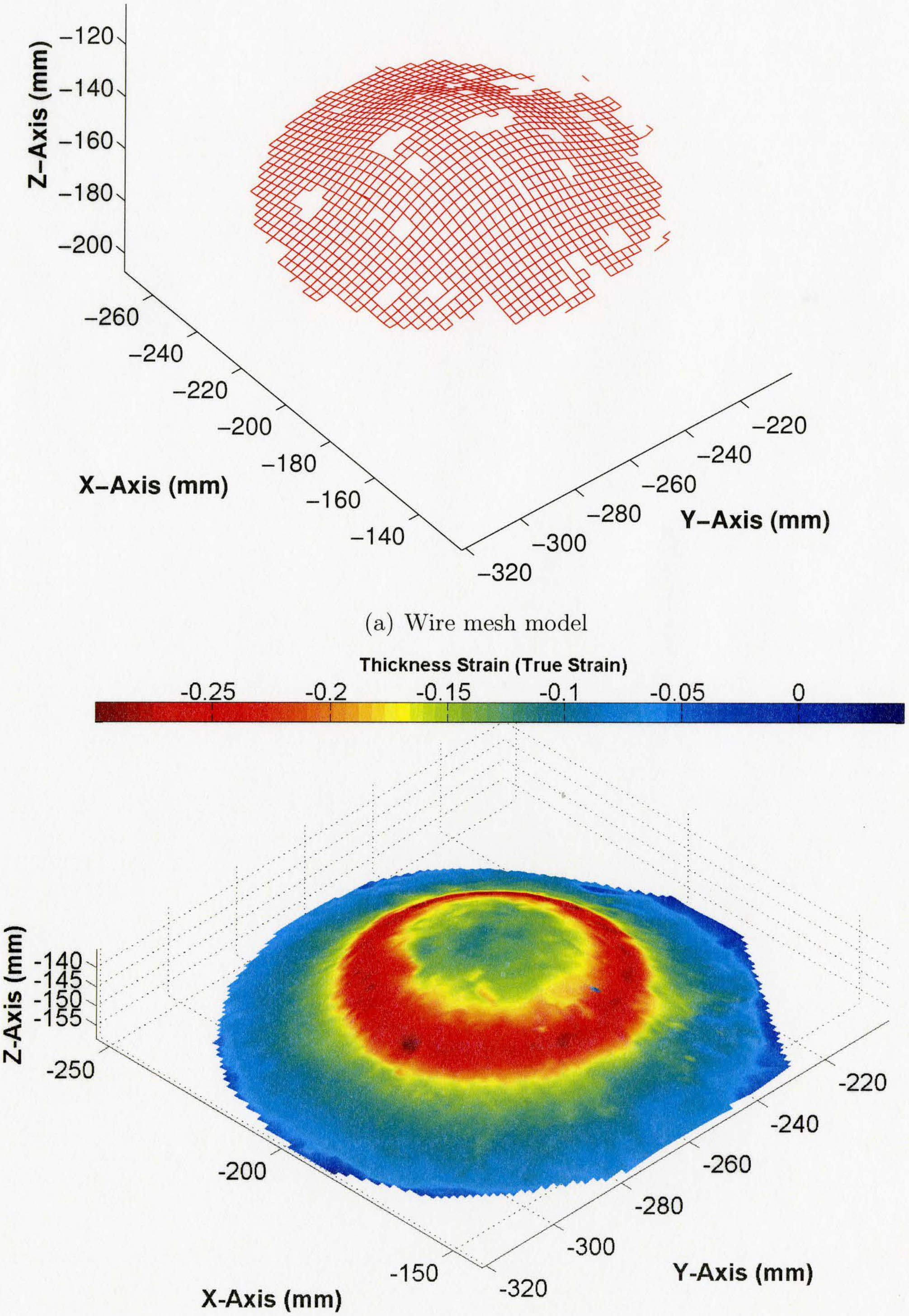
In order to properly determine strain accuracy of the stereo vision system it would be necessary to examine the effect of the 3-D point accuracy on the true strain calculations described in section 1.3. This is beyond the scope of this work and therefore it is reasonable to use the true strain equation:

$$\varepsilon' = \int_{L}^{L'} \frac{dL}{L} = ln\left(\frac{L'}{L}\right), \tag{4.2}$$

where L and L' are respectively, the original and final lengths. The error for the length of a line segment was determined from above to be 1.6 μm (2σ), therefore the true strain accuracy is calculated to be $ln(2.5016/2.5) \times 100 = \pm 0.064\,\%\ (2\sigma)$. This result is the strain accuracy for an ideal grid. Real grid measurements will result in far less accuracy because of the etching process. Another error source is the dimension of the grid itself, it is assumed that the grid is etched onto the part with perfect accuracy, obviously this will also introduce strain measurement error.

4.2.1 Sheet metal dome Parts

A high strain aluminium dome was created using a press and hemisphere punch. The grid lines were placed on the part using a silk screening process that results in very



(a) Wire mesh model

(b) Strain intensity plot

Figure 4.7: 3-D Models for high strain part

sharp edges. The 3-D reconstruction in wire mesh form can be seen in figure 4.7(a). Using the major ε_1 and minor ε_2 true strains, the thickness strain ε_3 is calculated as:

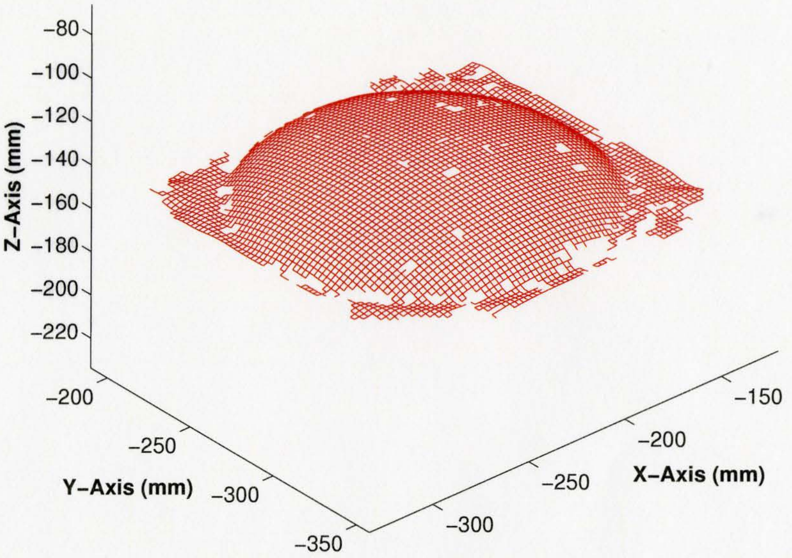
$$\varepsilon_3 = -\varepsilon_1 - \varepsilon_2 \tag{4.3}$$

A 3-D colour plot showing the thickness strain distribution over the part can be seen in figure 4.7(b). The strain distribution over this part shows the expected result, high strain around the dome sides and lower strain on top. The low strain on top is a result of the friction between the hemisphere press and sheet metal.

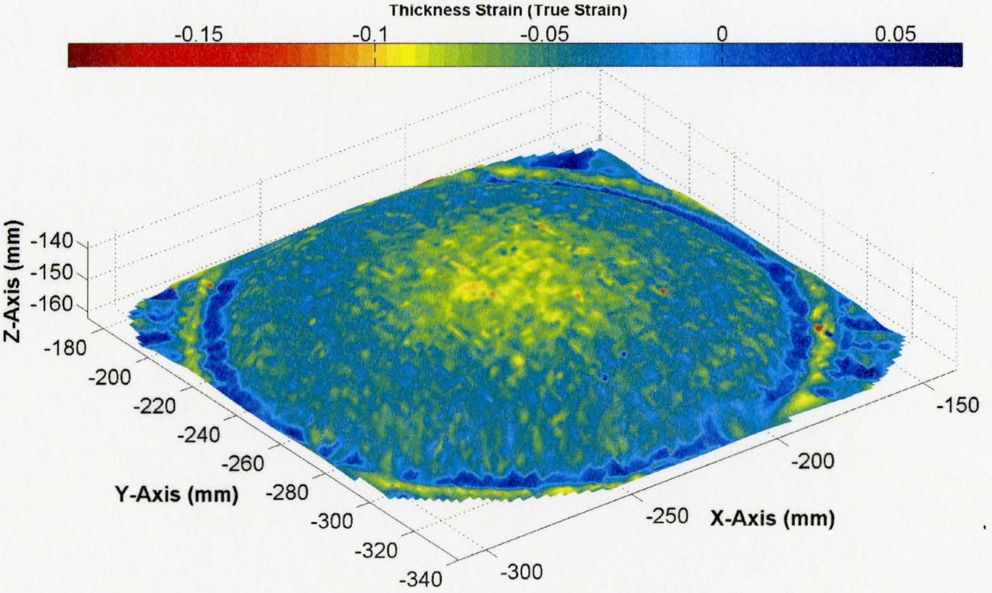
A low strain steel dome was created using a hydraulic forming machine. This grid was etched using an electrochemical process that does produce quite as sharp edges as the silk screening method. The resulting wire mesh and colour plots can be seen in figures 4.8(a) and 4.8(b). The strain distribution is as expected, with the strain increasing towards the top of the dome. There is a lot of noise in the colour plot for the large dome, this is mostly due to the etching process. It has been stated [32], that a square grid electrochemically etched into the part can introduce as much as ± 2% strain error. The holes in the two mesh grid images are due to missing grid lines on the part surface.

4.3 Processing Time

Each image of the stereo pair takes approximately 9 minutes to process. Binary thresholding, morphological operations and sub pixel grid line detection take approximately 45 seconds. The majority of processing time, approximately 8.25 minutes, involves fitting parabolas to grid line points. Finding between 60-90 grid intersection points per image, 120-180 parabolas have to be fitted. The parabola fitting is very computationally intensive because each iteration of the main optimization routine involves an optimization process to find the orthogonal contact point for each data



(a) Wire mesh model



(b) Strain intensity plot

Figure 4.8: 3-D Models for low strain part

point. In order to get the maximum possible accuracy, all points detected on the grid line were used to fit the parabolas. For most grid lines this usually involves 100 data points. This number could be substantially reduced without too much of an effect on the accuracy, resulting in a reduced processing time. The parabola parameters were determined with a tolerance of 1 e^{-3}, this could also be adjusted to improve processing time. Stereo matching takes less than 1 second per image pair.

To completely image the surfaces of the small and large dome parts shown in figures 4.8 and 4.7, required 60 and 170 image pairs respectively. This results in a final processing time of 18 and 49 hours for the small and large domes.

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

This thesis investigated the accuracy of a custom built close-up stereo vision system. The pose of the stereo head could be changed using an articulating probe head attached to a coordinate measurement machine. Image processing and stereo matching algorithms were presented and successfully used to measure points on the surface of two sheet metal parts.

In the proposed system the surface strain is determined using data from individual poses. This is beneficial because there is no error introduced into the strain measurement from movement of the imaging system. The accuracy of a 3-D point with respect to a single stereo head pose was found to be ±1.1 μm (2σ). Strain accuracy was not directly measured but was calculated using the stereo accuracy and found to be ±0.064% (2σ). This accuracy is more than adequate for most strain analysis. The best commercial system currently available is the ARGUS [5] system and its data sheet states a strain accuracy of ±0.2%. It has also been shown that the electrochemical process used to etch the grid can introduce a significant amount of strain error up to ±2%. Therefore the system presented within this thesis is probably more accurate than is necessary to perform sheet metal strain analysis. However, this good strain accuracy comes with the price of a very lengthy image processing time, approximately 9 minutes per image. The 3-D reconstruction of a part imaged by the system includes data from many different stereo head poses. The accuracy of a 3-D point from one pose with respect to 3-D points from other poses was found to be $\pm 17.1\,\mu$m (2σ). This should be adequate for checking the dimensions of most sheet metal products. Measurement accuracies of other 3-D metrology systems range from $\pm 50\,\mu$m for FARO [6] arms to $\pm 1\,\mu$m for a bridge type CMM.

The 3-D vision system presented in this thesis is currently not practical because of the lengthy image processing time. However, the system has demonstrated the feasibility of using close-up stereo vision to produce very accurate 3-D measurements.

5.2 Future Directions

Camera calibration

Bundle adjustment camera calibration cannot be utilized to its full potential using a 2-D testfield of points. A 3-D testfield of points would give improved calibration accuracy and will also allow the focal length to be determined. This could probably be best achieved using a planar test grid mounted to a micrometer attached to an optical bench. Assuming the motion can be carefully restricted to a translation in one direction, a set of 3-D points could be produced.

Physical problems

Defocus is a problem at very close distances. This is a physical limitation of optical lenses which has to this point in time not been overcome. To determine the effect of defocus on point accuracy, an investigation into the theoretical aspects of this problem could be undertaken. This is a complicated endeavor because compound lens systems consist of two or more lenses with variable focus and aperture settings. A better approach may be to experimentally determine the effects of defocus on 3-D point accuracy.

Sheet metal strain analysis

The image processing algorithms presented within this thesis are not currently suited for use with sheet metal strain analysis. They are too computationally intensive to be considered practical. The bottleneck in the process is the geometric parabola fitting which takes approximately 8.25 minutes per image. This processing time could be reduced by only performing geometric parabola fitting in cases where the algebraic fitting results in a large fitting error. The processing time could also be reduced with smaller tolerances on the geometric parabola fitting without too much of an effect on strain accuracy. Even with a reduction in processing time it is doubtful that the image processing could be done in real time. It may be more beneficial to develop a real time system that provides approximate stereo data so that the user can view which sections of the part have been covered. This real time processing could be achieved by using the centroid of adjacent corner vertices to approximate the grid intersections. The stereo accuracy could then be improved in an off line process. This processing could be done in parallel using a network of computers. The parabola fitting lends itself well to parallel processing because only 2-D data points must be sent for processing and a few parabola parameters received.

Surface measurement

The system presented in this thesis measures surface strain using an etched grid. This system could also be used to measure a plain surface with a grid projected on it. Some changes to the image processing would be necessary because grid elements could be very different in size and shape. However, the projected grid would be of better contrast than the etched grid and more easily detected. Assuming parabolas can be fitted, this system would give very good reconstruction accuracy.

Appendix A

Pinhole camera model:

1. All rays eminating from a single point in space must converge on a single point in the image plane (definition of focus).
2. Any ray entering the lens parallel to the axis on one side goes through the focus point on the other side.
3. Any ray entering the lens from the focus point on one side emerges parallel to the axis on the other side.

Appendix B

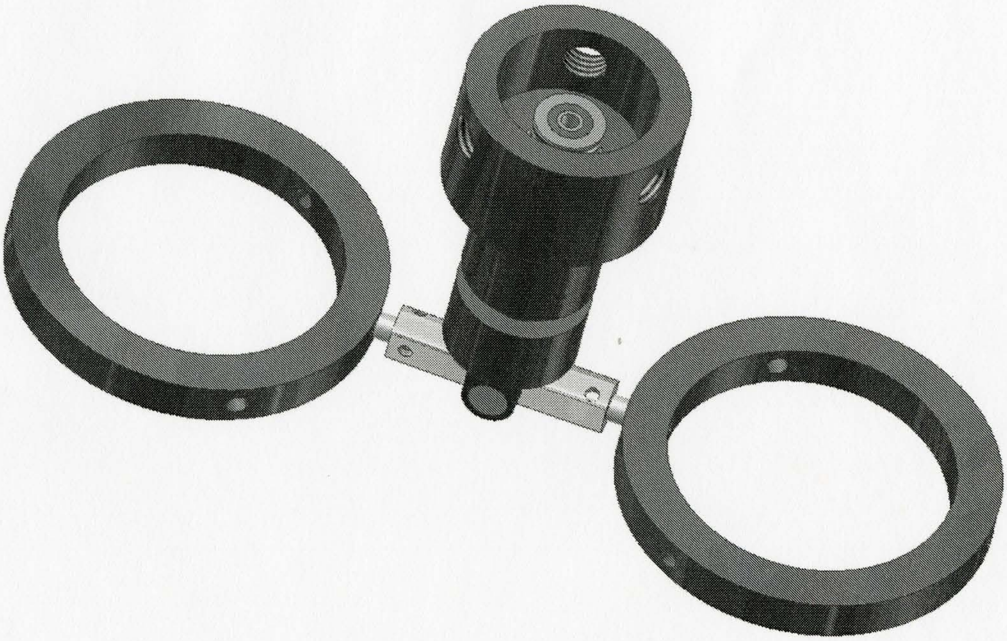


Figure B.1: Stereo head CAD drawing

Bibliography

[1] FMTI Systems Inc., Hamilton, Canada, http://www.fmtisystems.com.

[2] Hitchcock-Manthey, LLC, Troy, NY, U.S.A., http://www.camsysinc.com.

[3] Point Grey Research Inc., Vancouver, BC, Canada, http://www.ptgrey.com.

[4] Renishaw plc, Gloucestershire, United Kingdom, http://www.renishaw.com.

[5] GOM, Braunschweig, Germany, http://www.gom.com.

[6] FARO Technologies Inc, Lake Mary, Florida, U.S.A, http://www.faro.com.

[7] *ISO/TS14253-2: Geometrical Product Specifications (GPS) Inspection by measurement of workpieces and measuring equipment.* International Organization for Standardization, Case postale 56, CH-1211 Geneva 20, 1st edition, 1999.

[8] Fusiello A., Trucco E., and Verri A. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12:16–22, 2000.

[9] Triggs B., McLauchlan P.F., Hartley R.I., and Fitzgibbon A.W. Bundle adjustment a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms, Corfu, Greece*, pages 298–372, 1999.

[10] Horn B.K.P. Closed-form solution of absolute orientation using unit quarternions. *Journal Optical Society Amer. Vol.*, 4(4):629–642, 1987.

[11] Steger C. Analytical and empirical performance evaluation of subpixel line and edge detection. In Kevin J. Bowyer and P. Jonathon Phillips, editors, *Empirical Evaluation Methods in Computer Vision*, pages 188–210, Los Alamitos, California, 1998. IEEE Computer Society Press.

[12] Steger C. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, February 1998.

[13] Steger C. Subpixel-precise extraction of lines and edges. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIII, part B3, pages 141–156, 2000.

[14] Slama C.C. *Manual of Photogrammetry.* American Society of Photogrammetry and Remote Sensing, Virginia Ave, Falls Church, Va., fourth edition, 1980.

[15] Fraser C.S. Photogrammetric measurement to one part in a million. *Photogrammetric Engineering and Remote Sensing*, 58:305–310, March 1992.

[16] Brown D.C. A solution to the general problem of multiple station analytical stereotriangulation. *RCA Data Reduction Technical Report No.43, Patrick Air Force Base*, 1958.

[17] Brown D.C. Close-range camera calibration. *Photogrammetric Engineering*, 37:855–866, 1971.

[18] Eggert D.W., Lorusso A., and Fisher R.B. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6):272–290, 1997.

[19] Manthey D.W. and D. Lee. Recent developments in a vision-based surface strain measurement system. *The Journal of The minerals, Metals, and Materials Society*, 47(7):46–49, 1995.

[20] Manthey D.W., Bassette R.M., and Lee D. Comparison of different surface strain measurement techniques used for stamped sheet metal parts. *Proceedings of International Body Engineering Conference: Body Assembly and Manufacturing*, pages 106–111, September 21-23 1993.

[21] Borgefors G. Distance transforms in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[22] Wei G.Q. and Ma S.D. Implicit and explicit camera calibration: Theory and experiments. *PAMI*, 16(5):469–480, May 1994.

[23] Wei H., Yu D., Ruan X., and Wang Y. Deforming analysis of sheet metal based on stereo vision and coordinate grid. *Journal of University of Science and Technology Beijing*, 11(2):178–182, 2004.

[24] Beyer H.A. Accurate calibration of ccd-cameras. In *CVPR92*, pages 96–101, 1992.

[25] Harara H.M. *Handbook of Non-Topographic Photogrammetry.* American Society of Photogrammetry, Virginia Ave, Falls Church, Va., 1st edition, 1979.

[26] Sung J.A., Wolfgane R., and Warnecke H. Least-squares orthogonal distance fitting of circle, sphere, ellipse, hyperbola and parabola. *Pattern Recognition*, 34(12):2283–2303, 2001.

[27] Vogel J.H. and D. Lee. An automated two-view method for determining strain distributions on deformed surfaces. *Journal of Materials Shaping Technology*, 6:205–216, 1989.

[28] Oreau J.J., Garric V., and Devy M. Camera calibration for 3d reconstruction: application to the measure of 3d deformation on sheet metal parts. *Proceedings of SPIE - New Image Processing Techniques and Applications: Algorithms, Methods, and Components II*, 3101:252–263, 1997.

[29] Luh J.Y. and Klassen J.A. A three-dimensional vision by off-shelf system with multi-cmaeras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 7(1):35–45, 1985.

[30] Arun K.S., Huang T.S., and Blostein S.D. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987.

[31] Sid-Ahmed M.A. and Boraie M.T. Dual camera calibration for 3-d machine vision metrology. *IEEE Trans. Instrumentation and Measurement*, 39(3):512–516, June 1990.

[32] Sklad M.P. Aspects of automated measurement of proportional and non-proportional deformation in sheet metal forming. *Journal of Materials Processing Technology*, (145):377–384, 2004.

[33] Brown M.Z., Burschka D., and Hager G.D. Advances in computational stereo. *Trans. on pattern analysis and machine intelligence*, 25(8):993–1008, 2003.

[34] Faugeras O.D., Luong Q.-T., and Maybank S.J. Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision*, pages 321–334, 1992.

[35] Luong Q.-T. and Faugeras O.D. Self-calibration of a moving camera from point correspondences and fundamental matrices. *Int. J. Comput. Vision*, 22(3):261–289, 1997.

[36] Jain R., Kasturi R., and Schunck B.G. *Machine Vision*. McGraw-Hill, Inc., New York, 1st edition, 1995.

[37] Sowerby R., Duncan J.L., and Chu E. The modelling of sheet metal stampings. *International Journal of Mechanical Sciences*, A28(7):415–430, 1986.

[38] Ayres R.A., Brewer E.G., and Holland S.W. Grid circle analyzer: Computer aided measurement of deformation. *SAE Transactions*, 88(3):2630–2634, 2001.

[39] Tsai R.Y. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *RA*, 3(4):323–344, 1987.

[40] Tsai R.Y. and Lenz R.K. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, 1989.

[41] Umeyama Shinji. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380, 1991.

[42] Cunnington S.J. and Stoddart A.J. N-view point set registration: A comparison. In *BMVC*, 1999.

[43] Melen T. Geometrical modelling and calibration of video cameras for underwater navigation. *Dr. ing thesis, Norges tekniske hgskole, Institutt for teknisk kybernetikk*, 1994.

[44] Press W.H., Flannery B.P., Teukolsky S.A., and Vetterling W.T. *Numerical Recipes in C*. Cambridge University Press, New York, NY., 2nd edition, 1992.

[45] Pennec X. Multiple registration and mean rigid shape - application to the 3d case. In K.V. Mardia, C.A. Gill, and Dryden I.L., editors, *Image Fusion and Shape Variability Techniques (16th Leeds Annual Statistical Workshop)*, pages 178–185. University of Leeds, UK, July 1996.

[46] Yang Y. and Yan H. An adaptive logical method for binarization of degraded document images. *Pattern Recognition*, 33(5):787–807, 2000.

[47] Abdel-Aziz Y.I. and Karara H.M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Proceedings of the Symposium on Close-Range Photogrammetry*, pages 1–18, 1971.

[48] Zhang Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.