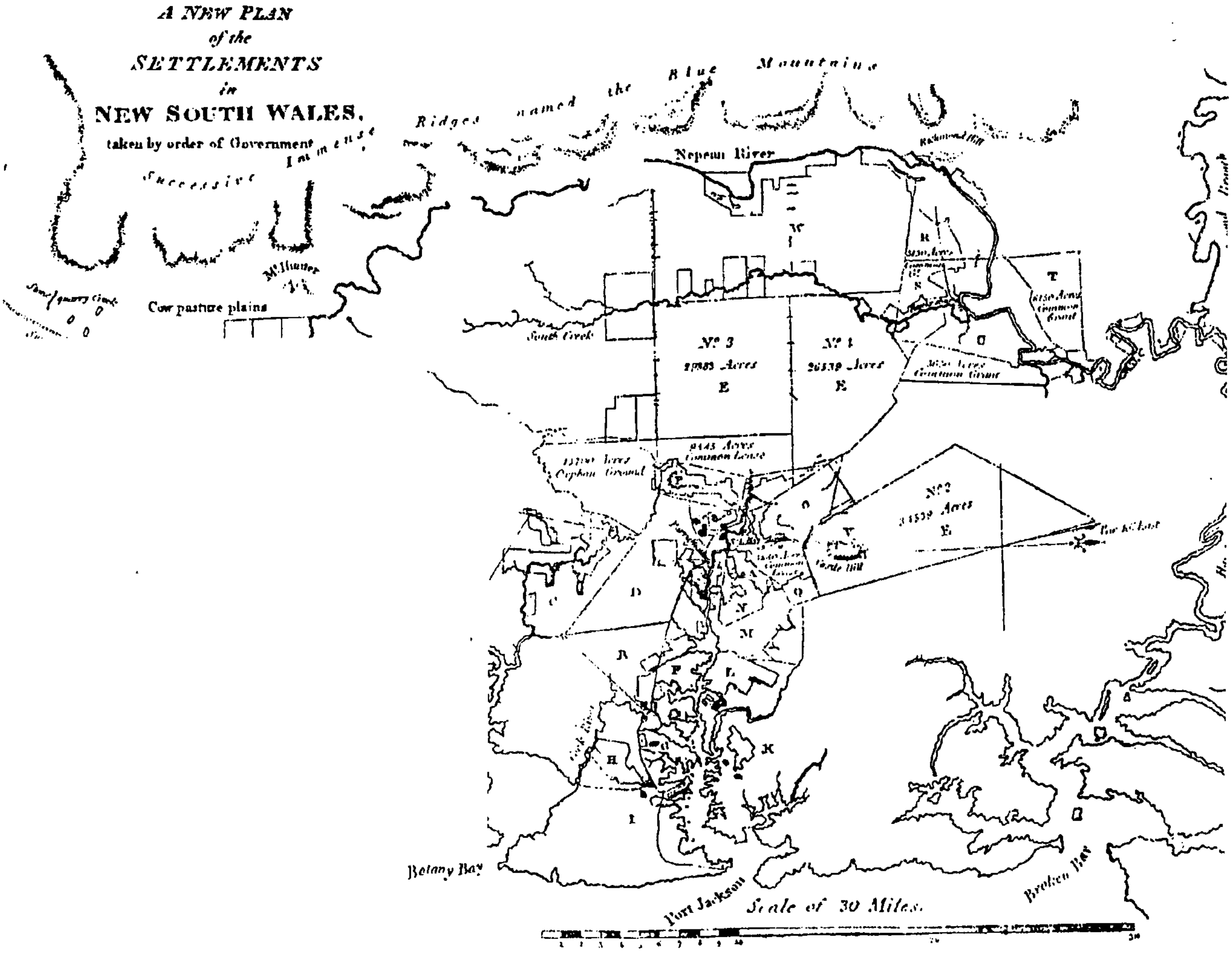
A DIGITAL PHOTOGRAMMETRY SYSTEM FOR INDUSTRIAL MONITORING

YONGRU HUANG



UNISURV S-47, 1997

Reports from

SCHOOL OF GEOMATIC ENGINEERING

THE UNIVERSITY OF NEW SOUTH WALES SYDNEY NSW 2052 AUSTRALIA

UNISURV REPORT S-47, 1997

A DIGITAL PHOTOGRAMMETRY SYSTEM FOR INDUSTRIAL MONITORING

YONGRU HUANG

Received: June, 1996
Accepted: August, 1997

SCHOOL OF GEOMATIC ENGINEERING
UNIVERSITY OF NEW SOUTH WALES
SYDNEY NSW 2052
AUSTRALIA

National Library of Australia

Card No. and ISBN 0 85839 072 8

Acknowledgments

I would like to thank my adviser Prof. John Trinder for introducing to the field of digital photogrammetry, and for carefully reading the thesis and marking valuable comments. His suggestions and advises on the various aspects of the work have been helpful.

I also thank:

- Prof. Bruce Forster, my co-advisor, for serving my examination committee and offering helpful comments regarding this research.
- Mr Brian Donnelly for his technical help and for providing me with access to any available resource that could advance my research.
- Dr. Joseph Jansa for his moral support and useful advice during his staying at School of Geomatic Engineering, University of NSW, Australia. His software and program technique are valuable for me to develop the digital photogrammetry system.

Finally, I wish to express my deep gratefulness to my wife Jianmin and my daughter Jing for their love and support throughout my life.

This research was performed while I was a recipient of an Overseas Postgraduate Research Scholarship. Other financial support for the work was provided by the Australian Research Council.

Abstract

Digital photogrammetry has been a very active and productive research field for years. Great strides have been made possible by the recent developments in digital sensors, advanced microprocessors, array processors, computer vision technique, and the decrease in software and hardware costs. Its application areas in a close range field have been extended to architecture, medicine and industry.

The purpose of this research is to study methods used in digital photogrammetry and machine vision, and develop them into a measuring system for industrial monitoring. The digital photogrammetry system includes steps of camera orientation, feature extraction, 3D object construction and the application of the recognition of regular objects.

Images of industrial components are generally characterised by sharp discontinuities, which represent features on the object. The accurate extraction of these discontinuities, which are characterised by the peak of the intensity change in the image, is an important step of machine vision and digital photogrammetry. A method has been developed, which detects edges with subpixel accuracy. Edge detection results in edge points chained along object boundaries projected in the images. The line segmentation method is applied to slip and merge edges into straight lines or regular curves in terms of the local direction of edges. These lines can be linked at terminals by the constraints of distance and orientation to form surface patches. The image processing transfers raw image data into a geometric graph. The data to be processed are reduced, but the extracted features become more meaningful and easier to calculate.

To transform image coordinates into object space or match image features in stereo images, it is necessary to establish the relation between the 3D object coordinate system and the 2D image coordinate system. An automatic or semi-automatic procedure of camera orientation has been developed for the digital photogrammetric system. In the application, small bright balls mounted on a calibration frame serve as control points. The edges along the contour of targets are detected, which determine the central location and size of targets. The precise camera parameters are then calculated by a bundle adjustment, while the initial approximations of the parameters are transformed from DLT computation.

Using two or more CCD cameras, the 3D surfaces of objects in the scene can be constructed by matching the same geometric properties in different images with the addition of the epipolar line constraints. A 3D straight line is simply presented by 3D coordinates of its two terminals, while a 3D regular curve can be described by a plane in object space and 2D curve parameters on the plane.

In order to recognise objects in the scene, a CAD system is used to design models which are output in DXF files. An inference system is then applied on DXF files to create graphic presentation. The descriptions of models in the database is the same as an object in the scene. The process of object recognition is performed by a detailed comparison between the potential matching graphs of a model and an object, and the 3D transformation between them. The object recognition process results in the identification of the sensed object, its position and orientation.

The methods described above were investigated, and software was developed and tested on several industrial components. The results obtained indicate the feasibility and applicability of digital photogrammetry for industrial monitoring.

Table of Contents

List of Tables

Table Page

List of Figures

Figure Page

Chapter 1 Introduction

1.1 General

Automatic measurement and interpretation of the physical world from remote sensing data is one of the most challenging problems in many engineering and physical science fields. A typical example is work transfer robots, which are required to recognise objects, as they pass along the assembly line, and to determine the next appropriate action which should be taken on them. The task of quality control, on the other hand, involves amongst other requirements, the precise checking of dimensions of the manufactured parts to ensure that they satisfy the specifications of the product.

Many machine vision systems developed so far have been based primarily on range images which contain direct measurements of the 3D properties of objects. Each pixel of the image represents a depth value of the object surface, giving relative distances between the surface point and the viewer. Using range images, the ambiguities of the feature interpretation which usually occurred in an intensity image, such as shadows, surface markings or illumination, are eliminated. The method can determine 3D geometry of objects directly with precision as high as 50 μm (Littlehales and Rioux, 1992), a precision that is difficult to achieve with a digital photogrammetric system. However, an intensity-based vision system is still acceptable not only because of its relevance to biological vision but also because of the robustness of passive sensing for industrial and other applications. There are a number of advantages in the use of intensity imaging systems, including: the intensity data is viewable by an operator and can reveal more than geometric information, eg. colour, texture, blemishes; features such as edges and faces can be extracted from the object by image processing, provided that these features are apparent in the image; lighting can be varied to accentuate various elements in the object. The facilities of the intensity imaging systems are more flexible to set up in an industrial environment and cheaper to purchase. The algorithms extracting shape information from intensity images can be classified as the following:

- **Shape from Shading** : The method exploits the changes in the image intensity (shading) to recover surface shape information. This is done by calculating the orientation of the scene surface corresponding to each point in the image. In addition to the constraint imposed by the radiometric principles, the method requires that the surfaces are smooth in order to calculate surface orientation parameters. Clearly, the smoothness constraint is not satisfied at all points and the surface reflectance is not always known accurately, resulting in inaccurate reconstructions (Horn and Brooks, 1989).

- **Shape from Focus** : Due to the finite depth of field of optical systems, only objects which are at a proper distance appear focused in the image, whereas those at other depths are blurred in proportion to their distances. Algorithms to exploit this blurring effect have been developed. The image is modelled as a convolution of focused images with a point spread function determined by the camera parameters and the distance of the object from the camera. The depth is recovered by estimating the amount of blur in the image, using the known or estimated line spread function. If the estimation of line spread function or the amount of blur is not correct, however, there exists a large error in determining the shape of a scene object.

- **Shape from Stereo** : Stereo disparity refers to the shift in the image position of a 3D localised entity in the scene when two or more cameras are used. The shape and depth can be derived by ray intersection in the use of image disparity information. This method is well established in photogrammetry or stereo procedures. The main problem in the method is to detect conjugate pairs of points in the stereo images.

- **Shape from Motion** : When images of a stationary scene are acquired using a moving camera, the displacement of the image plane coordinate of a scene point from one frame to another depends on the distance of the scene point from the camera. This is thus similar to the problem of shape from stereo. Alternatively, a moving object also produces motion disparity in an image sequence captured by a stationary camera. Such a disparity also depends upon the position and velocity of the object point.

- **Shape from Texture** : Image plane variations in the texture properties such as density, size, and orientation are the cues exploited by shape from texture algorithms. For example, the texture gradient, defined as the magnitude and direction of maximum change in the primitive size of the texture elements, determines the orientation of the surface. Quantifying the change in the shape of texture elements is also useful to determine surface orientation. From images of surfaces with textures made up of regular grids of lines, possibly due to structured lighting, orientation may be uniquely determined by finding the vanishing points. Besides being an indirect method for depth computation, this method also suffers from difficulties in accurately locating and quantifying texture primitives and their properties.

There are advantages and disadvantages in the use of each of these methods, depending on the type of applications, their environments and conditions. The application of industrial monitoring relates to dimensional inspection, positioning and tracking of objects, and reverse engineering, which usually requires high accuracy, automation and real time processing. **Digital photogrammetry,** which is a "shape from stereo" method, is the technology of obtaining reliable information about physical objects through processes of recording, measuring, and interpreting digital images, based on two or more overlapping images. Its emphasis is placed on accuracy and reliability. Therefore, digital photogrammetry, among the above technologies, is best suited to the application of industrial monitoring.

Digital photogrammetry has developed from analogue and analytical photogrammetry over the last two decades, but great strides have been made recently due to the developments in digital sensors, advanced microprocessors, array processors, machine vision techniques, and the decrease in software and hardware costs. Without doubt, digital photogrammetry has become a part of the broad area of machine vision and image understanding. Its application areas in a close range field have been extended to architecture, medicine and industry.

Like other technologies, digital photogrammetry is based on certain fundamental principles and techniques. To design successful practical digital photogrammetry systems requires a thorough understanding of all system aspects, from initial image formation to final scene interpretation. To convert raw data acquired by the sensors to meaningful information about objects in the scene being interpreted for industrial monitoring purpose, typical digital photogrammetry systems use the following processing-step sequence:

- Image capture;
- Camera calibration;
- Feature extraction;
- Construction of 3D objects from stereo images;
- Application of domain knowledge to recognise or inspect objects in the scene.

In each of these steps, many factors influence the choice of algorithms and techniques. In traditional close range photogrammetry, most industrial applications are mainly for metrology. Close range photogrammetry systems are most successful when measuring well targeted points in a controlled environment.

Machine vision is a relatively new and rapidly changing field, which pursues the same goal as human vision: to generate descriptions about the scene from images. The emphasis of machine vision is on extracting and grouping features, image segmentation, associating meaning to groups of symbols, geometric modelling, matching and hypothesis verification. The enrichment of the tool box of image analysis in machine vision opens the door to digital photogrammetry for full or at least partial automation of image interpretation. It inserts a new spirit into photogrammetry, leading to the redefinition of the basic concepts in the field. The image formation process can no longer be reduced to simply the perspective geometry of the images including the necessary calibration procedures, but it has to be modelled from the image acquisition to the stage of image understanding, thus closing the research gap between digital photogrammetry and machine vision.

Digital photogrammetry and machine vision are two different disciplines. The former is more concerned with accuracy, while the latter is more concerned with automatic recognition, so that they can be complementary each other. However, the technologies in machine vision are still developing and not general enough to solve the problems in digital photogrammetry. For example, the application of digital photogrammetry usually requires the features in an image to be measured with subpixel accuracy, while most methods in machine vision extract image features in pixel units. The technologies used in machine vision cannot be simply moved to the digital photogrammetry system. Therefore, the combination of the methods in both areas is a non-trivial task.

This thesis aims to study some methods used in digital photogrammetry and machine vision, and investigate their applications in digital photogrammetry for industrial monitoring. The main task is to develop a complete system which automatically describes 3D regular objects, using the methods of description to represent both models of objects and the objects themselves, and which recognises each scene object by identifying it with one of the models. The research pays attention not only to the measurement of industrial components with high resolution, but also the interpretation of objects in the scene. It is believed that this combination of digital photogrammetry with machine vision will lead to an improvement in the knowledge about the tools required for the development of digital photogrammetry systems for industrial monitoring.

1.2 The Scope of This Research

3D object recognition consists of identifying an object in a scene with a model in a database, to determine its correspondences, and the 3D information of the object, such as the orientation and position. Two major factors condition the performance and competence of a recognition system. One is the method used to describe the objects and models, the other is the method used to establish the correspondences between objects and models.

It is essential to develop an appropriate method of describing objects, in order to retrieve significant information from scenes of 3D objects and effectively represent these objects.

Objects can be described at different levels as shown in figure 1.1. Lower level descriptions such as pixel and edges are easier to compute, but they are not stable with respect to viewing directions. Higher level descriptions on the other hand, maintain their invariance with respect to the change of environments, but the algorithms to compute them are very expensive (Rao and Nevatia, 1986). In the research, a graphic representation based on boundary description is developed because of the following reasons:

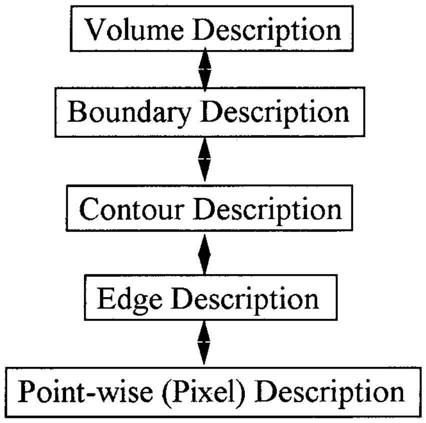


Figure 1.1 : Level of descriptions

- Boundary descriptions are richer than edge and pixel description. The computation of boundary representation is much more robust than that of volume description (Fan, 1990).

- Images of industrial components are generally characterised by sharp discontinuities. The accurate extraction of these discontinuities results in features of boundaries on the objects. Therefore, the information of object boundaries is related to feature extraction.

- The system is designed not only for object recognition, but also object inspection, in which object boundaries are the main elements measured.

A boundary representation can be viewed as a triple {S, I, G}, where S is a set of surfaces of the object, I a set of space curves describing the intersections between the surfaces in S, and G a graph describing the surface connectivity (Jain et. al., 1995). A graphic representation designed for object recognition is transformed from boundary representation and presented by planar surfaces which are bounded by straight lines or regular curves. These surfaces are grouped in terms of their normal directions and are stored together with their areas and perimeters. The topological relations between surfaces are constructed in terms of the centre of each surface and the common edges of two surfaces.

An object in the scene and the models in the database are presented in the same manner. The former is reconstructed from the scene, while the latter is created by a CAD system. Figure 1.2 illustrates a schema of the procedure by means of which objects in the scene can be constructed and recognised from digital stereo images.

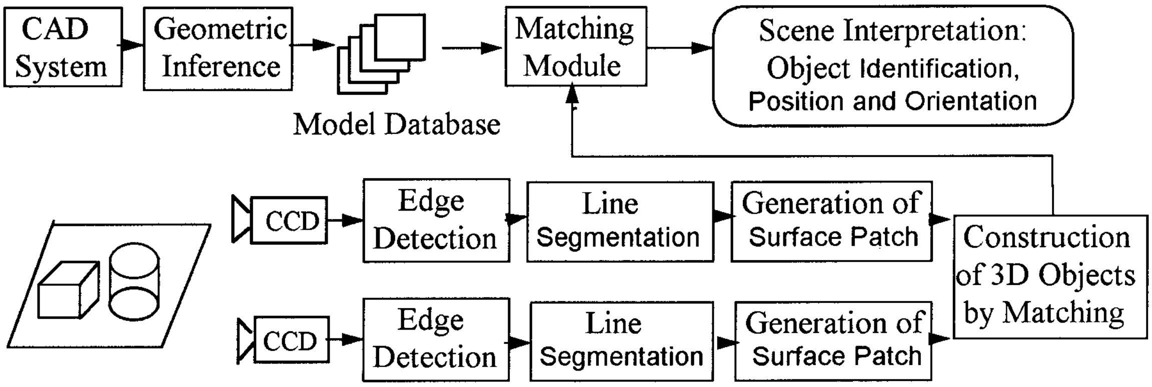

CAD System
Geometric Inference
Model Database
Matching Module
Scene Interpretation: Object Identification, Position and Orientation
CCD
Edge Detection
Line Segmentation
Generation of Surface Patch
CCD
Edge Detection
Line Segmentation
Generation of Surface Patch
Construction of 3D Objects by Matching

Figure 1.2 : Components of an object recognition system

The application of object inspection involves the precise checking of dimensions of the manufactured parts to ensure that they satisfy the specifications of the manufactured product. The difference between object recognition and object inspection is that the latter assumes that the objects under inspection are already known and they are usually placed in stable environments. However, the basic problems such as object construction and model description remain the same. The following subsection will outline the main issues in the thesis.

1.2.1 Reconstruction of Objects in the Scene

Physical boundaries of objects captured by CCD cameras are likely to result in the generation of edges during the imaging process. To describe features by their boundaries or to use feature based image matching in digital photogrammetry, it is necessary to obtain complete linear features of the object from its edges and represent them with a suitable data structure such as straight lines and smooth curves. A method has been developed, which detects edges with subpixel accuracy. Edge detection results in edge points chained along object boundaries in the images.

Edge detection is referred to as "low-level" image processing. A simple chained edge has no geometric meaning, because the boundaries of industrial components are often presented as straight lines and regular curves. To construct objects in the scene, it is generally more relevant to present the boundaries of objects revealed in the images in geometric form, since simple geometric functions provide more reliable information and are easy to calculate. A suitable approach to line segmentation has been developed, based on the analysis of the local direction of edges. The detected straight lines and regular curves are then linked at their terminals to form surface patches.

Digital photogrammetry enables object boundaries to be constructed using stereo images. The process of line segmentation results in the extracted features in the form of straight lines and regular curves. Therefore, matching on these meaningful features supplies reliable and stable results.

1.2.2 Generation of Models in Database

In an industrial environment, CAD systems are usually used to design objects for the manufacturing task. They should therefore be explored for representing the model database. AutoCAD is a general purpose Computer Aided Design program for preparing two dimensional drawings and three dimensional models. The speed and ease with which a drawing can be prepared and modified using a computer offer a significant advantage over hand preparation. Using the AutoCAD system, models can be created and output in DXF file, a drawing interchanging file. CAD models serve as a basic description of object geometry. Inference procedures of various sorts are applied to the CAD models to produce a graphic presentation in the database for object recognition.

1.2.3 Matching Primitives and Algorithms

The efficiency and robustness of a recognition system is crucially dependent on the primitives and algorithms used in the matching of the measured object with the models in the database. To recognise an object in the scene, it is necessary to match the extracted primitives against the design dimensions of the model in the database. One important characteristic of the system is that the dimensions of objects are measured accurately. As a first stage of model matching, most models can be discarded as not matching the scene object, by checking their size against the size of the object. In the second stage, all visible graphic elements of an object should exist and satisfy those in the corresponding model. Their topological relations should also be identical with the relations between the graphic elements in the model. One condition not being satisfied will indicate that the model does not correspond. The object recognition process results in the identification of the sensed object, its position and orientation.

1.3 Organisation of This Study

- Chapter 2 presents a method developed for edge detection with subpixel accuracy. It is an extension of Förstner operator which is an efficient operator for detection and precise location of feature points. In the chapter, the Förstner operator is first introduced in terms of a linear model, which is then extended by adding an additional constraint to locate step edges. To improve the accuracy of the edge location, an edge point is determined by a weighted average of two points derived from both sides of the edge. Another implementation is to use a round operator window instead of a square window, so that the result will not be influenced by the difference of edge orientations. The direction of the edge is attached to each edge point as a primitive for succeeding processes.

- Chapter 3 presents the line segmentation methods, describing how straight lines are detected by grouping edge points with approximately the same local direction, and how regular curves are detected by checking the sign of the difference of edge local directions. These geometric features are then linked in their terminal points to form surface patches.

- Chapter 4 presents a procedure of camera orientation. In a digital photogrammetry system, the step of camera orientation is normally performed in advance of image capture of the objects in the scene. Since the procedure of camera orientation applies the method of edge detection described in chapter 2, this portion is placed after image processing. The procedure includes recognising targets as control points in images, calculating the central coordinates of targets, and obtaining the camera orientation by bundle adjustment.

- Chapter 5 presents an automatic procedure of 3D object construction in terms of epipolar geometry and feature geometry. Epipolar geometry is first introduced to give the relation between image coordinates in the overlapping stereo images. Straight lines and regular curves are then matched and intersected in object space. A 3D straight line is simply presented by 3D coordinates of its two terminals, while a 3D regular curve can be described by a plane in object space and 2D curve parameters on the plane.

- Chapter 6 presents the application of object recognition. In the recognition procedure, matching between an object in the scene and models in the database is performed by two modules: the screener, in which most models are deleted due to their size being beyond the range of the scene object; and the graph matcher, which performs a detailed comparison between the potential matching graphs and computes the 3D transformation between them.

- Chapter 7 offers conclusions. The main points from the thesis are summarised. Contributions are discussed with suggestions for future research.

Chapter 2 Edge Detection

2.1 Introduction

Edge detection is low level image processing, which serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information in images. Physical object boundaries are very important descriptors of objects in machine vision and recognition systems. In order to describe a feature by its boundaries or to use feature-based image matching in photogrammetry, it is necessary to extract the complete details of linear features in the object from edges, and to represent them in a suitable data structure such as straight lines and smooth curves. For this reason, edge detection plays an important role in the early stages of image processing. Due to the complexity of the physical world and multiple sources of noise, the signal to be processed is complex, and the detection of such edges is therefore non trivial.

In the early stages of edge detection, most approaches tried to focus on slope analysis of the grey values. These straight forward methods, which made good sense, certainly had the advantage of being very fast to compute. Roberts operator (Roberts, 1965) requires 2×2 window and the slope is calculated at the central pixel corner in the two orthogonal directions. The Sobel's operator (Hall, 1979) is performed with a 3×3 window by a simple convolution. It operates as a low pass filter followed by the computation of the derivative. However, these simple operators, even though fast to calculate, cannot give comprehensive results when used on a variety of images. A good edge detection operator is expected to satisfy the following criteria (Canny, 1986) when applied to an image corrupted by noise:

- Good detection: The operator must be able to mark the edge points, while reducing the probability of falsely marking non-edge points.
- Good location: Location of the marked edges should be as close as possible to the true edges.

- Robustness: Performance of the algorithm should be robust to noise and should be consistent for a variety of images.

In recent years, many researchers have worked on this area for various applications and created many powerful algorithms. The major approaches chosen to tackle this challenging problem could be classified into the following categories: optimal operator, adaptive algorithms, multiscaling approaches, morphological operators, statistic tools, linear filtering, neural networks and subpixel analysis.

2.1.1 Optimal Operator

The optimal operator aims to find the optimal filter (in terms of signal to noise ratio) for edge detection. In this group, a single edge is considered. Marr and Hildreth (1980) convolve the signal with a rotationally symmetric Laplacian of Gaussian mask and locate zero-crossings in the resulting output. Canny (1986) proposed to solve the problem by deriving, using variational methods, an optimal operator which can be approximated by a derivative of Gaussian mask. He defined a set of heuristic criteria for the integration of multiple size masks, and showed some promising results for two scales of edges. Canny Operator is a well known operator, which has been extensively used in image processing, since it was proposed, Nalwa and Binford (1986) proposed an edge detector which fits, at each point, a set of surfaces within a window and accepts the best surface, in the least squares sense, which has the fewest parameters.

2.1.2 Adaptive Algorithms

The general concept is to apply a versatile operator which adapts itself to the local topography of the signal to smooth. It is mainly a relaxation method, whose principal design is suited to iterative algorithm implementations. Philippe et al.(1991) presented a method which convolves the signal to be smoothed with a very small averaging mask whose coefficients reflect, at each point, the degree of continuity of the signals. The weights vary at each pixel and at each iteration, being computed as a function of the gradient. After a few iterations edges are then very simply singled out. Parvin and

Medioni (1987) proposed a method to extract meaningful features from range images. Their strategy consists of automatically selecting an adequate kernel size for the detection and location of such features. Although the method requires the non-obvious setting of five parameters, it directly provides features at different scales and is applicable to curves.

2.1.3 Multiscaling Approaches

In a gray level image, changes in light intensity reflect many spatial scales at which visible edges are formed. In order to describe the signal changes at different scales, the edge operator should be applied on an image varying in size according to the size of an edge. This is called the multiscale or multiresolution approach. A classical approach is to construct image pyramids: the initial image is subsampled a number of times, until one reaches an appropriate size of an image.

Many different approaches have been recently studied based on multiscaling. Williams and Shah (1990) focus on a multiscaling approach for improving weak parts in contour lines. The processing then undertakes pixel chaining. Zhang et al. (1988) summarise the basic principle of relaxation methods and presents an improved method based on a multiscaling approach, by propagating the weighted coefficients from the top of the pyramid to the lower layers. Jeong and Kim (1992) extend the optimal-filter concept of Poggio et al. (1984) and the scale space concept of Witkin (1983), to an adaptive scale parameter determined for each pixel before detecting final edge maps. Lastly, the method introduced by Lu and Jain (1992) is a multiscale reasoning algorithm for edge recovery called reasoning about edges in scale space.

Included in this topic is also an original method, Wavelet Transform, whose framework is introduced by Meyer (1990) and Mallat (1989). Mallat and Zhong (1990) showed that the sharper variation points of a signal can be detected from the local maxima of its wavelet transform. By proving that the evolution across scales of the wavelet maxima characterises the local shape of the signal shape variations, they showed the method not only detects edges but also classifies them. Barlaud et al. (1991) presented an edge detection implementation based on a Recursive BiOrthogonal Wavelet Transform. The algorithm uses a filter bank composed of a highpass filter for the wavelet and a lowpass filter for the scaling function.

2.1.4 Morphological Operators

Morphological filters are based on erosion, dilation, opening and closing operations. Yu and Yla-Jaaski (1991) applied a method to segment magnetic resonance images for subsequent 3D visualisation. The technique combines region growing, edge detection and a novel edge preserving smoothing algorithm. Pavlidis and Liow (1990) combined region growing and edge detection for image segmentation. Cheng and Venetsanopoulos (1992) presented a new type of opening operators (NOP) and closing operators (NCP). An adaptive morphological filter is then developed on the basis of the NOP and NCP. It is said that the filter can preserve the significant details in any shape, does not create artificial patterns, and does not require much computation. Boulanger (1994) developed a multiscale edge detection algorithm for range and intensity images. The algorithm is based on a new nonlinear filter, which produces a scale-space filtering analogous to Gaussian filtering, but has several interesting properties, since the algorithm uses a multidimensional morphological operator to compute the position of edges. The morphological operators can be used efficiently to compute these edges at pixel precision with a good noise immunity.

2.1.5 Statistical Tools

The method has been used in medical imagery for a long time. The input signal is modelled by statistical assumptions. Cristi (1989) presented an algorithm which efficiently filters the data without degrading the boundaries. The technique is based on the assumption that data are doubly stochastically modelled: local behaviour is modelled by auto regressive equations, while regions are modelled by a Markov random field. Statistically based tools can also be applied for automatically making decisions. For instance, Samadani and Vesecky (1990) presented a method using maximum a posteriori estimation, together with statistical models for the speckle noise and for a curve-generation process, to find the most probable estimate of features in a given image.

2.1.6 Linear Filtering

For this popular research sector, infinite impulse response filters are implemented with iterative algorithms like filter banks. Sakar and Boyer (1991) described an optimal infinite impulse response zero-crossing based edge detector. With the implementation they suggested, the computing time required is independent of the scale factor chosen. Ziou's work (1991) derives an optimal line detector from Canny's approach. An efficient implementation using an infinite impulse response filter is provided.

2.1.7 Neural Networks

Neural networks are based on connected neurones, which imitate human performance to model and integrate fairly complex rules and handle exceptions. The large variety of networks existing relates to all the different types of design layers, the interconnection model chosen between each neurone, and the way the corrective coefficients are propagated. The training phase for the network remains the crucial part. Along with the long computation time required, a major problem is the difficulty to exactly control the prediction of the results. In a recent publication, Paik and Kataggelos (1990) presented an algorithm using multistage adaptive linear neurones. The method requires the use of predefined masks but is much faster than the Marr and Hildreth's (1980) edge detector. Xue and Breznik (1990) use a Hopfield-type neural network with one layer per direction studied, which implies unfortunately that the number of directions considered has to be preset. The same problem occurs with the algorithm of Liao et al. (1991), where one network per direction studied is require in the first prospective edge selection pass. A second pass is then performed for the final edge selection module.

2.1.8 Subpixel Analysis

Two main edge models have been proposed for the measurement of edges at subpixel accuracy. In the first technique, subpixel analysis is based on using a perfect step edge as the underlying edge model. MacVicar-Whelan et al. (1981) used the gradient operator to determine the pixel location of zero crossings and then linearly interpolated the location. Tabatabai et al. (1984) fitted the first three statistical moments to a step edge model by determining the optimal values of the moments. Huertas et al. (1986) implemented LOG masks combined with a facet model followed by interpolation to detect edges at subpixel accuracy. Lyvers et al. (1989) developed a subpixel edge operator that locates edges by fitting the spatial moments of a step edge model to the data. Gruen and Agouris (1994) proposed a globally enforced least squares template matching method, constrained by internal shape forces, for automatic precise geometric identification and registration of object outlines. Edge locations are identified in an image by matching windows of this image to ideal edge models (templates).

The second subpixel technique is based on energy models to determine the localisation of step, peak and roof edges. Kisworo et al.(1991) used a parameterized function to model an edge. The initial model is determined by a decision process that is based on the response of the signal to the local energy filters. The parameters describing the best fit of the model to the data define the position of the edge to subpixel accuracy. Morrone et al.(1987) proposed an alternate method for feature extraction based on discerning how features are built up in an image, rather than by considering differential properties. The norm of local energy, defined by a quadrature pair of functions, is computed as the square root of the sums of squares of the functions obtained by convolving the image profile with a set of quadrature masks.

2.1.9 A New Method for Subpixel Edge Detection

The main purpose of edge detection in this research is for object inspection and recognition, in which subpixel measurement is highly desirable. The methods discussed above give the various approaches to this goal. However, it is not clear from the literature which of these methods will work in practice and what performance and accuracy can be expected from a given solution. Hence, to extract edges with high accuracy is still an open problem, especially in industrial applications. In this chapter, a new algorithm has been proposed for edge detection with subpixel precision. It is an extension of the Förstner Operator (1987) which is an efficient operator for detection and precise location of feature points. The algorithm is supported by the fact that most edges of close range images of the real world will be composed of sharp changes over a small number of pixels. The distribution of gradient magnitude forms a shape like mountain. The peak of the intensity changes, therefore, can be optimally determined from the signal in the neighbourhood of the gradient magnitudes.

In the following sections 2.2 and 2.3, the Förstner operator is first introduced in terms of a linear model which precisely determines the position of corners and then the model is extended by adding an additional constraint to locate step edges. In section 2.4, some problems are discussed, which influence the accuracy of edge location. To improve the accuracy of edge points, two implementations of the model are presented in section 2.5: one is to overcome the deficiency in the use of a fixed window size for the operator to extract edges at different scales; the other is to solve the practical problem that a square window gives a non identical solution, due to the different orientations near the edges. The edge detection procedure is given in section 2.6, based on the linear model and several examples of the detector performance on real images are presented in section 2.7. Finally, a summary is concluded in section 2.8.

2.2 Förstner Operator

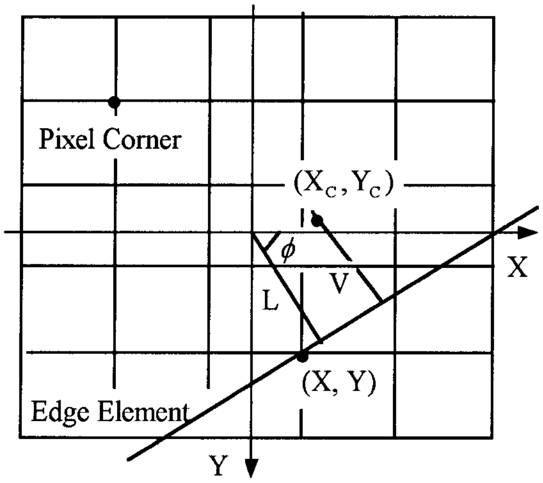
The Förstner operator has its basis in extracting feature points for digital image matching. The distinct points, such as corners, can be located by least squares estimation using a linear model, which computes the intersection point of all edge elements inside the window. The edge element at a pixel corner is defined as a straight line passing through the pixel corner with an orientation derived from the gradient as showed in figure 2.1. The corresponding weights are added in the least squares operation to reflect the strength of gradient in each pixel corner. The error equation for each pixel corner can be written as:

$$V = \cos\phi X_C + \sin\phi \; Y_C - L \qquad (2.1)$$

Where L = cosϕ X + sinϕ Y, the distance of the line from the origin; X, Y are the coordinates of the pixel corner in the window; ϕ is the angle of the gradient direction from the X-axis; X_C, Y_C are the coordinates of the estimated location of a feature corner. The weight P is proportional to the absolute gradient square:

$$P = G_x^2 + G_y^2 \qquad (2.2)$$

The gradient is obtained by Roberts operator as showed in figure 2.2.



[BD1]Figure 2.1: Edge element at position (X, Y) for determining the estimated point (Xc, Yc)

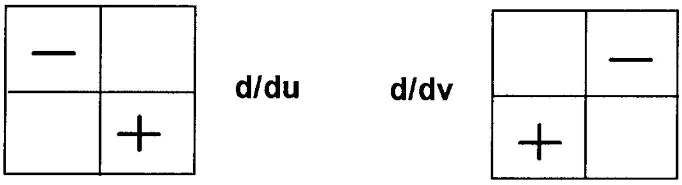


Figure 2.2: Roberts gradient in U and V direction

$$\begin{aligned} G_u &= g(i+1, j+1) - g(i, j) \\ G_v &= g(i+1, j) - g(i, j+1) \end{aligned} \qquad (2.3)$$

Where i, j = 0, 1, ··· ··· , n; n is number of the rows and columns of the image; g is the intensity of a pixel in the image. Then the gradient is translated to X, Y direction by:

$$\begin{aligned} G_x &= \frac{\sqrt{2}}{2}(G_u - G_v) \\ G_y &= \frac{\sqrt{2}}{2}(G_u + G_v) \end{aligned} \qquad (2.4)$$

$\cos\phi$ and $\sin\phi$ can be represented by gradient value:

$$\cos\phi = \frac{G_x}{|G|} \qquad \sin\phi = \frac{G_y}{|G|} \tag{2.5}$$

Where $|G| = \sqrt{G_x^2 + G_y^2}$.

According to principle of least squares adjustment $\sqrt{PVV} = \min$, we obtain the normal equation from Eq[2.1]:

$$\begin{pmatrix} \sum G_x^2 & \sum G_x G_y \\ \sum G_x G_y & \sum G_y^2 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \end{pmatrix} = \begin{pmatrix} \sum (G_x^2 X + G_x G_y Y) \\ \sum (G_x G_y X + G_Y^2 X) \end{pmatrix} \tag{2.6}$$

The Förstner operator has the advantage that any number of edge elements can intersect on the unknown point. The residual V of the edge elements from the pixel corners in Eq[2.1] could be very large, up to a few pixels, depending on the weight and position of these pixel corners in the window. According to Förstner (1987), the internal precision of the estimated point can be measured by an error ellipse, which illustrates the error of the point in different directions. The roundness of the error ellipse can be estimated by

$$q = \frac{4\det(N)}{\mathrm{tr}^2 N} \tag{2.7}$$

Where

$$N = \begin{pmatrix} \sum G_x^2 & \sum G_x G_y \\ \sum G_x G_y & \sum G_y^2 \end{pmatrix} \tag{2.8}$$

$$\det(N) = \sum G_x^2 G_y^2 - \left(\sum G_x G_y\right)^2 \tag{2.9}$$

$$\mathrm{tr}N = \sum G_x^2 + \sum G_y^2 \tag{2.10}$$

The roundness measure q lies in the range between 0 and 1. In the case of a right angled corner, where the two intersection directions determine the position, the error ellipse will tend to a circle and q to 1. When a straight edge passes through the window, however, all edge elements, except those without the magnitude of gradient, lie approximately in the same direction, ie. q tends towards 0. The error along the edge direction is infinite and the location of an edge is unstable. Therefore the Förstner operator cannot locate the position of a single edge.

2.3 A Linear Model for Edge Detection

In an image, an edge point is a point where a 'sharp' intensity change takes place. An edge can be described by discrete edge points which chain together. Since edge points can be defined anywhere on the edge, it is not necessary to consider the error of an edge point along the edge direction, because any constraint on the edge location in edge direction does not violate the definition of an edge point. In order to implement the model Eq[2.1] for edge detection, an additional constraint is added, which is based on the introduction of the line passing through the origin and which meets the edge at right angles as showed in figure 2.3. The constraint limits the edge position along the edge direction to the point which is nearest to the origin in the window coordinate system. The error of the edge point along the edge direction, under the constraint, becomes zero. The limiting condition can be given by

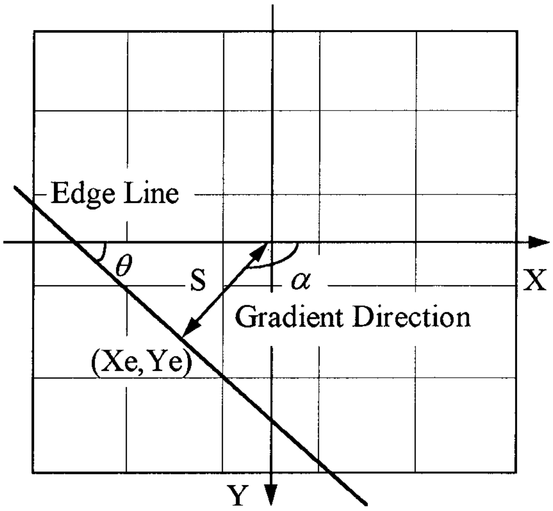


Figure 2.3: The edge position limited by a constraint along the edge direction.

$$S = \frac{X_e}{\sin\theta} = -\frac{Y_e}{\cos\theta} \tag{2.11}$$

Where θ is the angle of the edge direction to the X-axis, which is derived from the weighted average of gradients in the window as follows:

$$\begin{aligned} \overline{G}_x &= \frac{1}{n}\sum_{i=1}^{n}(G_i^2 \cos\phi_i) \\ \overline{G}_y &= \frac{1}{n}\sum_{i=1}^{n}(G_i^2 \sin\phi_i) \end{aligned} \tag{2.12}$$

Where $G^2 = G_x^2 + G_y^2$. The gradient direction is given by

$$tg\alpha = \frac{\overline{G}_y}{\overline{G}_x} \qquad (2.13)$$

The gradient direction is defined from the low intensity towards the high one. To define the edge orientation in the image, we assume that the intensity of the right side along the edge direction has a higher value than that of the left side. Therefore the edge direction is given by:

$$\theta = \alpha - 90^0 \qquad (2.14)$$

Substituting Eq[2.11] into Eq[2.1], we obtain:

$$V = (\cos\phi \, \sin\theta - \sin\phi \, \cos\theta\,) \times S - L \qquad (2.15)$$

In the linear model, only one parameter S is estimated. The result of least square adjustment is given by:

$$S = \frac{\sin\theta\,(\sum(G_x^2 X) + \sum(G_x G_y Y)) - \cos\theta\,(\sum(G_x G_y X) + \sum(G_y^2 Y))}{\sin^2\theta \sum G_x^2 - 2\cos\theta\sin\theta \sum G_x G_y + \cos^2\theta \sum G_y^2} \qquad (2.16)$$

and

$$\begin{aligned} X_e &= \sin\theta \times S \\ Y_e &= -\cos\theta \times S \end{aligned} \qquad (2.17)$$

The value of S gives the distance from the origin to an edge in the window, while the subpixel coordinates (X_e, Y_e) form a vector from the pixel centre to an edge point. If $|X_e| < 0.5$ pixel and $|Y_e| < 0.5$ pixel, the edge passes through the central pixel and the pixel is defined as an edge pixel which is used in the subsequent definition of edges.

2.4 Problems of Edge Location Using the Model

The linear model Eq[2.15] employed to estimate the peak of intensity changes characterises S by the weighted average coordinates in each pixel corner. Edge location determined by the linear model is equivalent to the centre of gravity in the operator window. In general, the centre of gravity of a physical object is estimated by the whole mass of an object without being affected by the mass of other objects. In a similar manner to the situation for the precise location of edges, the operator window should cover the whole range of intensity changes associated with an edge, but not include other unwanted signals. Since there is no a priori knowledge of the scale of intensity change in a local area of an image, it is difficult to determine the optimal size of the operator window. In this section, the problems of spatial scale are presented, which exist in the location error of edges by using this linear model. Other problems are also discussed such as curved edges and the asymmetric distribution of gradient magnitude.

2.4.1 Scale of Intensity Change

In the real world, edges rarely occur in a single step, but exist over a range of different scales. It is possible that some intensity changes for one edge may cover only 2 or 3 pixels, while the intensity changes associated with another may extend over a large number of pixels in the same image. Figure 2.4 illustrates a 1-D signal presented as the first derivative of intensity, covered by a window (from A to C). As there is no significant gradient value in the asymmetrical part between line A and line B, the symmetrical part in the window (from B to C) determines the peak of intensity change. In the case of figure 2.5, however, where the range of intensity change is larger than the window size, the gradient value in the asymmetrical part (from A to B) influences the result of the least squares adjustment, which makes the position of the edge point biased towards the window centre. The truncation error increases with the increase of the range of the intensity change for a fixed sized window. This means that the value of S defined by Eq[2.16] will be shorter than expected. The truncation error is also dependent on the

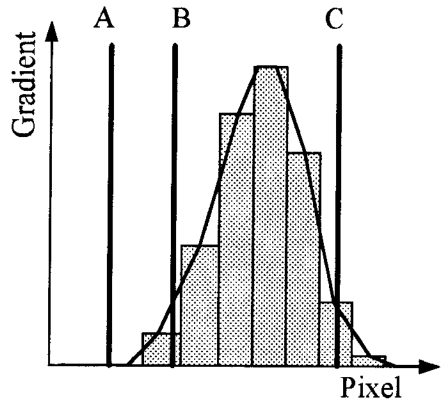


Figure 2.4: The operator window covers the narrow range of intensity change

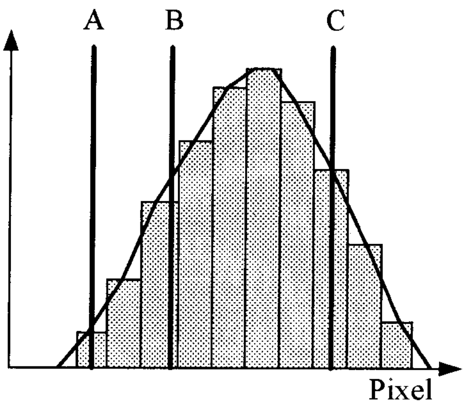


Figure 2.5: The operator window covers the wide range of intensity change

distance between the window centre and the peak of an intensity change. In figure 2.5, the area of the asymmetrical part increases with the increase of the distance, so that the largerS, the greater the error in the value of S. When an edge covers a small amount outside a pixel, as shown in figure 2.6 by pixel 1, the pixel is falsely defined as an edge pixel. However, when an edge just lies on the centre of operator window, no asymmetrical part exists and the truncation error becomes zero.

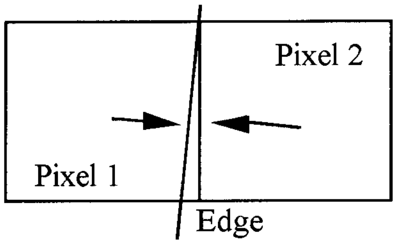


Figure 2.6: A false edge pixel 2 created due to its short vector length.

In general, a large window will precisely locate edges, even for the narrow range of intensity changes, but the gradient at the pixel corners not included in the range of intensity change associated with the edge should be zero. This condition is usually not satisfactory for complex images, where additional unwanted information such as neighbouring edges may be included into the operator window, if it is too large, This results in the edge location being biased towards or away from other neighbouring edges (Lu, 1989). In addition, the calculations associated with a large window is time consuming and may create a location errors in curved edges. Therefore, it is not appropriate to use large windows for the edge detection.

2.4.2 Curved Edges

The linear model Eq[2.15] is designed for straight edge detection. For a curved edge covering the operator window as showed in figure 2.7, there exists an error in edge location due to the variation in the direction of edge elements. The diameter of the circle in this example is 7 pixels and the operator window size is 5×5. In order to make the example simple, three pixel corners are considered to contribute to the location of the edge point. The edge element in each pixel corner is assumed to be of the same direction as the tangent in the point of the circle, and has the same weight as other pixel corners. The three edge elements form a triangle, and determine the location of the edge point, which is biased away from the circular centre. More tests on the different sizes of computer generated circles are showed in figure 2.8. Four operator window sizes are used, 3×3, 5×5, 7×7, 9×9. The edge points on the circles are extracted by Eq[2.17] with different window sizes, and they are fitted to a circular function. The estimated diameters are distorted due to operator window sizes. The magnitude of the distortion being dependent on the diameter of the circle and window size. The larger the diameter or the smaller the window size, the less the distortion. If the diameter of a circle is over 60 pixels, a piece of arc on the circles can be regarded as a straight line, and the influence of edge being curved can be ignored, especially for the small window size.

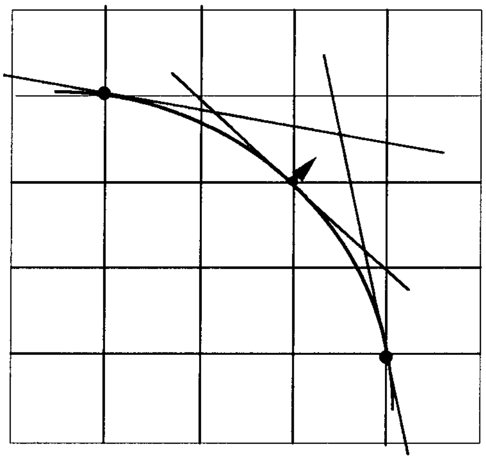


Figure 2.7: Distortion of edge location due to the edge being curved

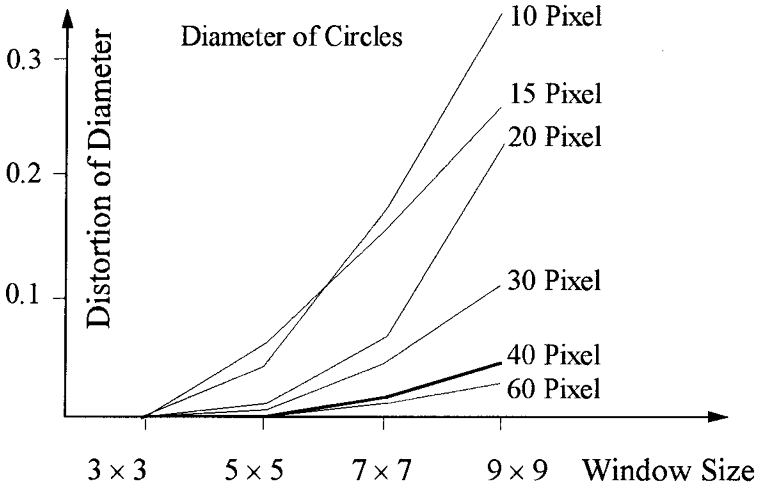


Figure 2.8: Distortion of the circle diameter for different circle sizes and window sizes

The situation of edge displacement in a corner is similar to that in curved edges. In a real image, a corner will not occur as a sharp angle, but as a smoothed arc in the local area. One reason is that the process of image acquisition and digitisation blurs corners in an image. The other reason is that the image preprocessing prior to edge detection, such as image smoothing, also blurs corners. The curvature in a corner is dependent on the shape of the corner and the degree of its smoothness. For the precise location of curved edges, therefore, the size of the operator window should be made as small as possible.

2.4.3 Asymmetric Distribution of Gradient Magnitude

The distribution of gradient magnitude in the above examples is assumed as symmetrical with respect to the peak of intensity changes. In fact, an asymmetric distribution of gradient magnitudes usually occurs in real images. Figure 2.9 displays a white strip on a black background. The grey values and gradient magnitude on the black line shown in the figure are listed in table 2.1. The slope of the distribution on the white side is steeper than that in the black. If the peak of intensity changes is in the centre of an operator window, as shown in figure 2.10, the edge location will be biased towards the black side due to the asymmetrical distribution of gradient magnitudes. The error increases with increases in the operator window size. Hence a small operator window is required to achieve high precision of edge location.

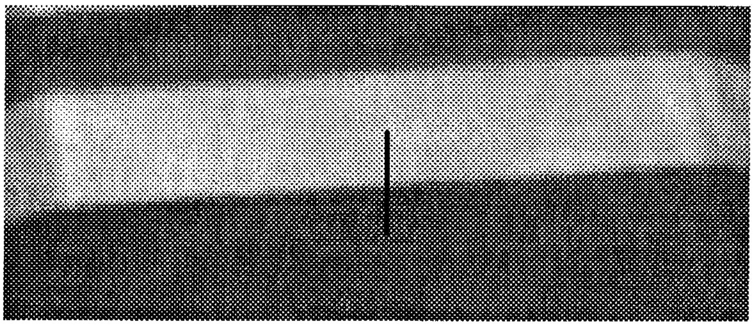


Figure 2.9: A white strip under the black background

Pixel	1	2	3	4	5	6	7	8	9
Intensity	188	186	185	178	167	136	79	72	72
Gradient	2	1	7	11	31	60	7	0	

Table 2.1: Intensity and gradient on a line

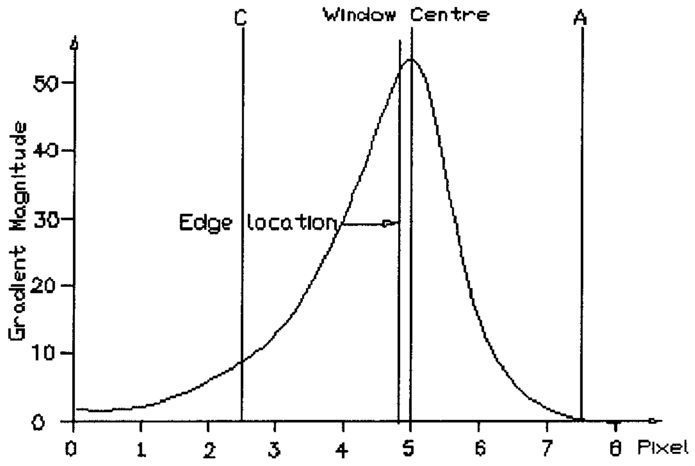


Figure 2.10: The shift of edge location due to the asymmetrical distribution

2.5 Implementations of the Model

High accuracy of edge location is essential in the research for the purpose of object inspection and recognition. To locate edges precisely, the problems discussed above should be solved or limited to a certain magnitude so that their influences on final results are minimised. In this section, a weighting function is developed to overcome the deficiency of using a fixed window size for the operator to extract edges at different scales. As discussed above, the length S in Eq[2.16] calculated using a small operator window is shorter than its correct dimension in the case of an intensity change across a broad edge. Fortunately, this situation will occur on both sides of an edge as shown in figure 2.6, ie. the lengths S_1 and S_2 of the pixels in each side are shorter than their correct lengths. This makes it possible to delete the false edge pixels and to improve the accuracy of edge location from the adjoining edge pixels by a weighting function. Another implementation is to solve the practical problem that the same signal in a square window may have different contents due to different edge orientations. A round window operator, instead of a square window operator, is performed on edges, in order to give an identical result.

2.5.1 Weighting Function

The vectors (Xe, Ye) of the edge pixels occurring on opposite sides of an edge will differ. If a false edge pixel exists, the vector of the true edge pixel must exist in the direction to that shown by the false edge pixel, as shown in figure 2.11. By comparing the lengths of their vectors, the false edge pixels (pixel 3 and 4) attached to the longer vector can be rejected.

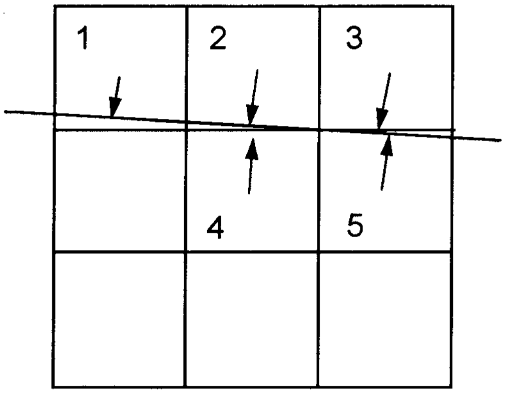


Figure 2.11: Edge pixels occur on opposite sides of an edge.

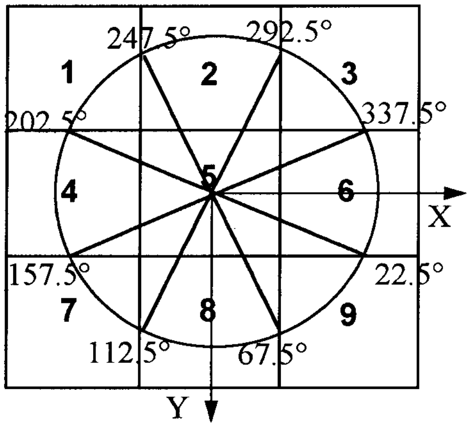


Figure 2.12: The direction of a vector assigned to one of eight sectors.

To find the adjoining pixel to which the vector of a candidate points, the orientation of the vector should be defined within one of eight sectors (each 45°) as shown in figure 2.12, since there are the eight pixels around a given pixel. The section associated with an adjoining pixel is determined by reference to its orientation. The precise position of the edge point can be determined from the weights as a function of the two vector lengths. The shorter the vector, the higher the weight. The empirical weighting function can be written as follow:

$$
\begin{aligned}
X &= \frac{1}{P_1 + P_2}(P_1 X_1 + P_2 X_2) \\
Y &= \frac{1}{P_1 + P_2}(P_1 Y_1 + P_2 Y_2)
\end{aligned}
\qquad (2.18)
$$

Where (X_1,Y_1), (X_2,Y_2) are the coordinates of two edge points in the image coordinate system, determined from both sides of an edge. P_1 and P_2 are the weights determined by the lengths of the two vectors from the centres of two pixels respectively. To choose the best weighting function, tests have been performed on the image of a computer generated circle with a diameter of 20 pixels, and blurred by a Gaussian smoothing filter. The edge points are fitted to a circular function using a least square adjustment. The precision of edge points can be assessed by the RMS (root mean square error) of the distance from each edge point to the assessment circle. Table 2.2 illustrates the precision of the edge points with different weights. Type 1 with 5 × 5 window size provides the best result. The weighting function in Eq[2.18] greatly improves the precision of edge points as shown in table 2.3. In the table, line 1 presents the precision of edge points located in the pixel centres; line 2 presents the precision of subpixel location determined by Eq[2.17]; and line 3 presents the precision of subpixel location after improvement by the weighting function (type 1). Figure 2.13 illustrates the edge location of a circle in three resolutions. It is interesting to note that the error of edge points in line 1 and 2 decreases with the increase of operator window size, while the best precision in number 3 is calculated by 5 × 5 window.

	Weight		3 X 3	5 X 5	7 X 7	9 X 9
	P2	P1				
1	S_1	S_2	0.022	0.015	0.040	0.056
2	$\sqrt{S_1}$	$\sqrt{S_2}$	0.124	0.065	0.045	0.058
3	S_1^2	S_2^2	0.084	0.041	0.041	0.055

Table 2.2: The precision of edge points for different weighting functions

Diameter		3 × 3	5 × 5	7 × 7	9 × 9
10 Pixels	1	0.322	0.322	0.337	0.254
	2	0.187	0.086	0.047	0.031
	3	0.037	0.012	0.030	0.025
20 Pixels	1	0.343	0.343	0.325	0.236
	2	0.128	0.128	0.061	0.054
	3	0.022	0.015	0.040	0.056
30 Pixels	1	0.360	0.348	0.332	0.240
	2	0.130	0.130	0.060	0.046
	3	0.016	0.016	0.030	0.042

Table 2.3: The precision of edge points in (1) Pixel location; (2) Subpixel location; (3) Subpixel location after improved by the weighting function.

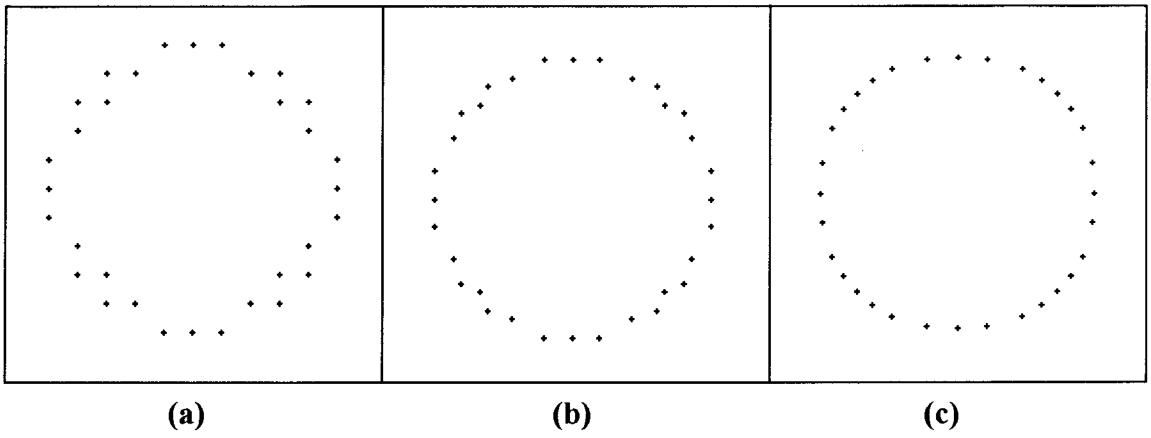


Figure 2.13: The edge points in a circle (a) Pixel location; (b) Subpixel location; (c) Subpixel location after improved by the weighting function

2.5.2 Round Window

The normal shape of the operator window is square. An area covered by a square window is not invariant with the orientation of the window. The same edge with the different orientations can exist in the square window, resulting in different contents as displayed in figure 2.14, leading to the different location of the edge. To overcome the deficiency in the use of a square window, the operator window is made round as shown in figure 2.15. Tests have been performed by the operator with round and square windows on the computer created circles whose diameters are 10, 20 and 30 pixels. Table 2.4 illustrates the test results obtained with the window size of 5×5, 7×7 and 9×9 pixels. The precision of edge points is assessed by the RMS of the distance from each edge point to the circular function. It is obvious that the precision is greatly improved by using the round window, particularly for 5×5 pixels.

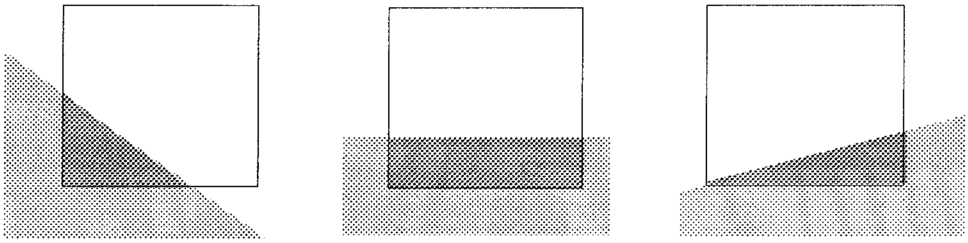


Figure 2.14: A square window covers on the area with different orientations

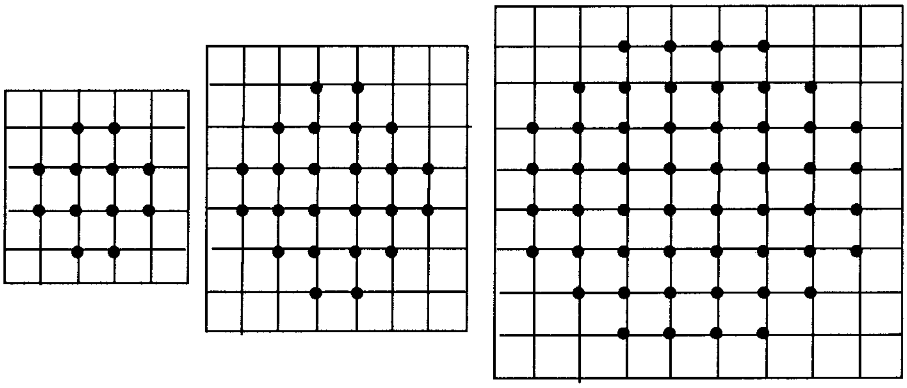


Figure 2.15: Round window in 5 × 5, 7 × 7 and 9 × 9

Diameter	5 × 5		7 × 7		9 × 9	
	Square	Round	Square	Round	Square	Round
10 Pixels	0.059	0.012	0.049	0.031	0.039	0.025
20 Pixels	0.043	0.014	0.053	0.040	0.078	0.056
30 Pixels	0.031	0.016	0.053	0.030	0.075	0.042

Table 2.4: The precision of edge points performed by square and round window on the different size of circles.

2.6 Process of Edge Detection

The process of edge detection has been implemented in five steps: ***image smoothing, pixel screening, subpixel location of edge points, edge thinning*** and ***edge chaining.*** A image smoothing function is first applied on the input images to reduce the effect of noise distortion. To eliminate unnecessary processing, the image is then screened in terms of gradient magnitude in each pixel to determine which pixel should be processed in the succeeding step. The third step is to calculate subpixel location of edge points using Eq[2.17]. The edge thinning checks false edge points and improves the accuracy of edge location. Finally, edge points are chained by following pixels, based on the minimum local distance. These steps are discussed below.

2.6.1 Image Smoothing

Natural images taken by a camera usually contain some noise. A differential operator, such as Roberts operator, usually enhances noise, which will result in jagged or discontinuous edge lines. The linear model (Eq[2.15]) based on a round window using the least squares method of edge detection has some influence on reducing the effect of the noise. However, such influences will degrade when the window size is smaller. In order to reduce the effect of noise in the intensity changes without changing the operator window size of 5×5, a smoothing function must be applied to the image intensities. The Gaussian filter is a well known image smoothing function, which is defined as:

$$G(x,y) = \frac{1}{\sigma} \exp\left(- \frac{x^2 + y^2}{2\sigma^2} \right) \qquad (2.19)$$

where x, y are the coordinates in the window coordinate system. σ is the scale factor of the Gaussian function, which determines the size of the smoothing window. A coarse scale of the smoothing function not only filters out noise but also blurs the image. If one edge with the symmetrical distribution of gradient magnitude occurs in the window during smoothing, coarse scale smoothing will not influence the edge location (Lu, 1989). However, if an edge with asymmetrical distribution of gradient magnitude is smoothed, the peak of intensity change will be shifted as shown in figure 2.16. A large smoothing window may also include more unwanted signals than a small window. Therefore, the window size must be considered in regard to both image smoothing and accuracy of edge location.

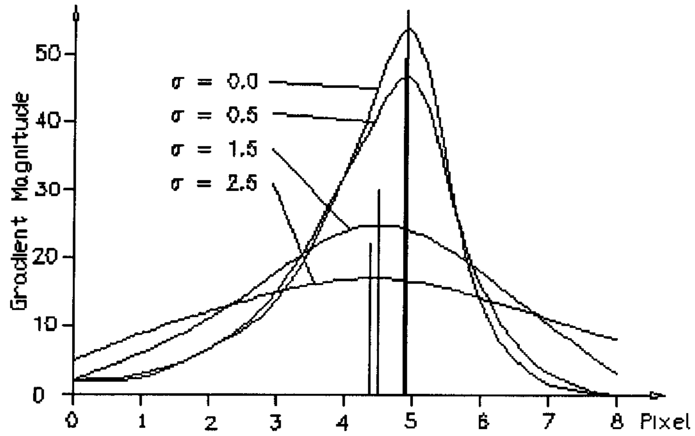


Figure 2.16: Shift of the peak of intensity change due to image smoothing

2.6.2 Pixel Screening

An edge corresponds to an intensity change which has a higher gradient magnitude than for neighbouring parts of an image. To avoid estimating edge points over all pixels in the image using Eq[2.16], because it is too time consuming, pixel screening is used to locate those edges with gradient magnitude above a certain threshold. Edge gradient magnitude is calculated by Roberts gradient in a 3 × 3 window from:

$$\mathrm{trN} = \sum G_u^2 + \sum G_v^2 \qquad (2.20)$$

The threshold is determined by the analysis of the histogram of gradient magnitude over the whole image. If an image consists mainly of two classes of gradient levels, the histogram will have two peaks separated by a valley. The histogram-based method, proposed by Otsu (1979), can find the midway point between the mean of the two classes. In this way, most pixels in an image are screened out, and only those gradient magnitudes which are above the threshold will reveal the pixels which must be located by the succeeding step.

2.6.3 Subpixel Location of Edge Points

The position of the pixels screened by the above step are calculated using Eq[2.16] in 5 × 5 windows. If the coordinates of a pixel are within the area of the central pixel in the operator window, the pixel is labelled on an edge support image with a value of 1, while 0 means non-edge pixel. In practice, the threshold for the absolute coordinates is set to 0.75 pixel. Those pixels, whose coordinates are larger than 0.5 pixel, are used to improve the accuracy of edge location, based on the function in Eq[2.18]. It has been shown by tests that the absolute coordinates of an edge point greater than 0.75 pixel have little effect on accuracy improvement.

2.6.4 Edge Thinning

As discussed in section 2.4.1, the vector length S of an edge pixel obtained by Eq[2.17] will be shorter than expected when the range of intensity changes is wide, resulting in the generation of false edge pixels and an error in the edge location. The purpose of edge thinning is to overcome the deficiency in the linear model. In the process, each edge pixel has its neighbours determined by the orientation of its vector. If the neighbouring pixel is indicated as an edge pixel as shown in figure 2.6, the correct edge pixel is determined from the two by comparing their vector lengths. The pixel with the shorter length is defined as an edge pixel. The edge location is then calculated by the weighting function Eq[2.18]. If the neighbouring pixel is not an edge pixel, the pixel is defined as a true edge pixel and its coordinates are not changed.

2.6.5 Edge Chaining

After edge thinning, extracted pixels within which edge points have been determined correspond to the peak of intensity changes. Edge chaining is needed to connect these scattered pixels into edge chains which represent the required structural information of the image features. Generally, the edges which correspond to boundaries of objects should be long and of high gradient magnitude, while the edges created by the other factors, such as illumination of light source and reflection of surface, are usually jagged, discontinuous or of low gradient magnitude. By analysing these properties, we can classify features in images and limit the influences of other factors.

To chain edges, the edge pixels should be tracked in the proper order. Since each edge pixel has eight neighbouring pixels, the deterministic rule for pixel tracking is to find the nearest candidate to the current edge point by local minimal distance. The distances between the chained edge pixel and neighbouring edge pixels are calculated using their subpixel coordinates. Figure 2.17 shows that the edge has been chained along the edge points from pixel 1 to pixel 4. In order to determined the next tracked pixel in the 3 × 3 window, the distances from edge point 4 to point 5 and to point 6 are calculated (point 2 and 3 have already been tracked). By comparing the two distances, the edge pixel, which is clearly point 5, with shorter distance can be determined.

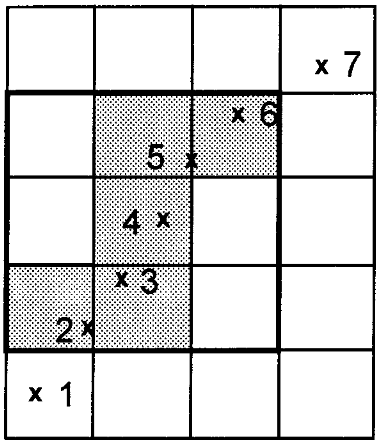


Figure 2.17: Edge chaining in minimal distance

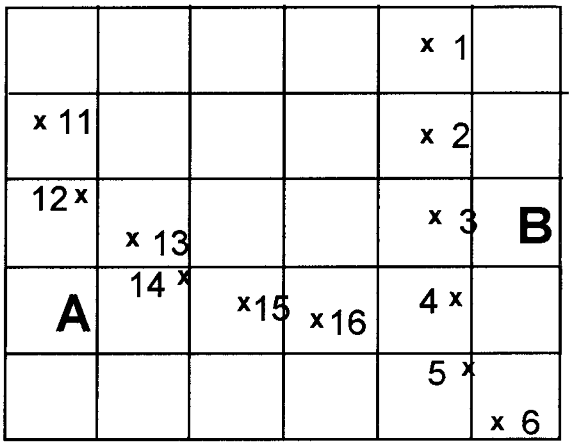


Figure 2.18: Edge chaining in a junction constraint

However, simply using the minimal distance rule will not handle the complicated situation of junctions. In figure 2.18, if the first tracked edge point is number 1, the edge can be chained from 1 to 6. If the first tracked edge point is number 11, the edge can be chained from 11 to 16 and then chained to 4. Different edge chaining exists in the same edge map, due to different starting points. In order to give identical results, it is necessary to check if the next candidate is appropriate for tracking. In the above example, point 4 is the nearest to point 16, but the distance $D_{4,16}$ is larger than the distance $D_{4,5}$ and $D_{3,4}$. This means that the point 4 is more properly chained between point 3 and point 5. This condition determines whether the edge tracking should continue or not. Each candidate (pixel 4) is placed in the centre of the 3×3 window. The distances from the candidate point to its neighbouring points in the window are calculated and listed in terms of their magnitudes ($D_{4,5}$, $D_{4,3}$ and $D_{4,16}$). If the distance ($D_{4,16}$) from the candidate point (point 4) to the current chained point (point 16) is after the second one ($D_{4,3}$) in the list of the distances, the edge chaining stops. This method provides an identical result of edge chaining, regardless of starting chain points.

The direction of the edge chaining corresponds to the local direction of the edge. If a closed edge chains clockwise, the intensity inside the edge should have a higher grey value than that on the outside. The local direction of an edge derived from Eq[2.14] is attached to each edge point as a primitive for line segmentation in the next step. In the process, an edge line is also given the properties of length and average gradient magnitude which are important information for distinguishing boundaries of objects and noise. If an edge is short and has a small gradient magnitude, it is regarded as a falsely marked edge and rejected. On the other hand, by ranking the properties of edges, they can be listed in the order of importance. More important edges can be processed first, and they will in turn provide some knowledge for processing less important edges. A demonstration of the application of these principles is given in chapter 3.

2.7 Experiments

Initial experiments have been carried out on real images of industrial components. The images of an industrial part and its edges are displayed in figure 2.19 and 2.20 respectively. From the illustrated subpixel locations of the edges, an analysis will reveal which factors influence the precision of the edge location. The intensity on one plane surface is often non homogeneous. It will change depending on the direction of imaging, the location of light source and the reflection of an object surface. Marks on the surface are also factors which change the intensity in local areas of the image. For a boundary to appear identical from different image view points, it must be the intersection of two surfaces. This condition may not be satisfied in the real world due to the smoothness of boundaries, causing edge shifts in different images. Illumination of the light source is particularly important in the reflection of intensity change. High contrast in images reduces the effect of noise, so that edges are chained more stably as showed in figure 2.22, which is superior to figure 2.21.

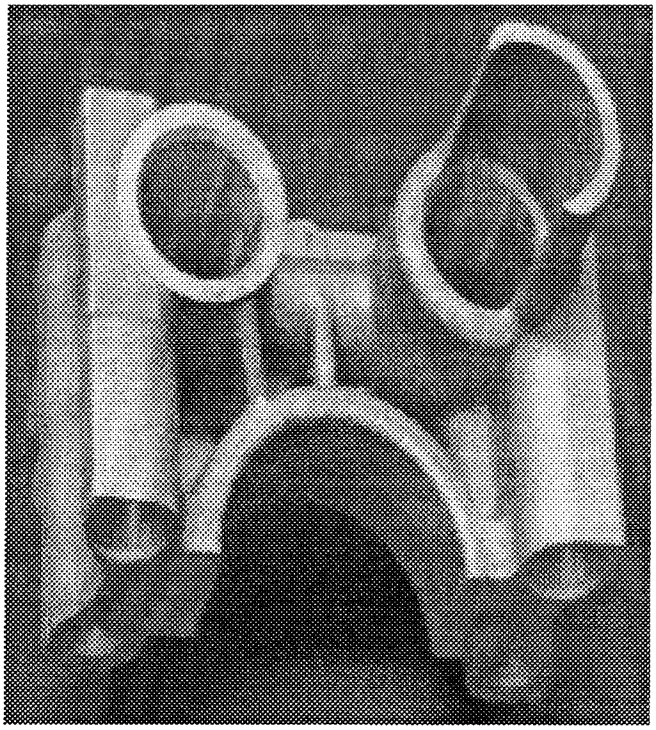


Figure 2.19 : An industrial part

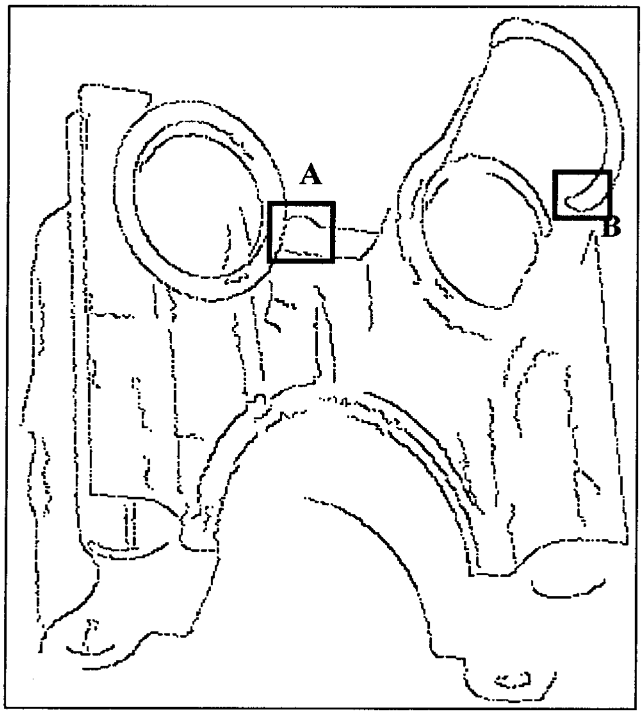


Figure 2.20 : Edges detected from the original image

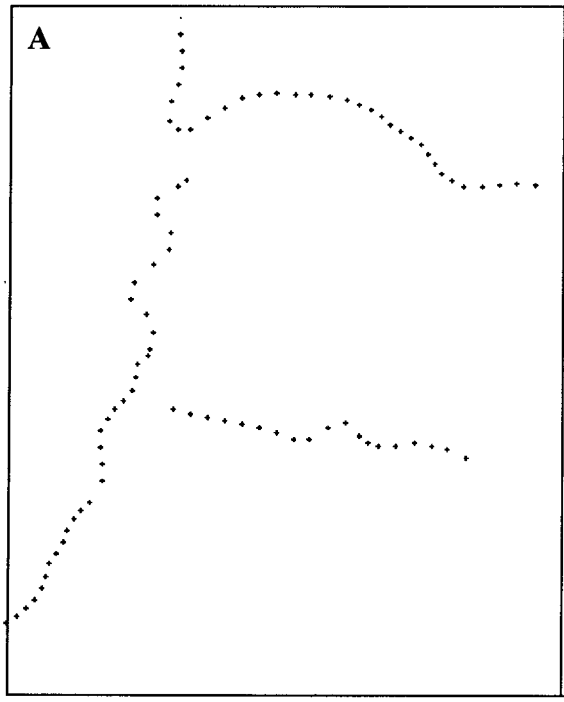


Figure 2.21: Low resolution with low contrast

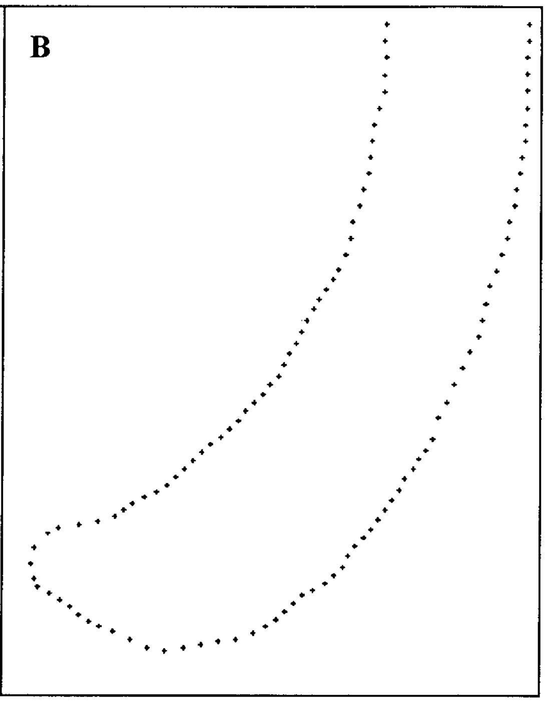


Figure 2.22: High resolution with high contrast

In order to investigate the precision of edges derived from the method described in this chapter, a test was carried out on a block , as illustrated in figure 2.23. This block is used as baseblock for an interferometer. The boundaries of the block, which had been tested by autocollimation equipment, are sufficiently close to straight lines, so that the precision of edge location can be estimated by the RMS of residuals of edge points from a straight line fitted to the edge points. A digital CCD camera (720 × 575 pixel) was first calibrated by a procedure, which will be discussed in chapter 4. The block was then imaged by the camera. The edge along the boundary of the block, whose size is about 58 × 25 × 5 cm^3, was extracted by using edge detection method as described above. The error of the edge points created by lens distortion and image shear was corrected in terms of the parameters of camera calibration. The edge points were

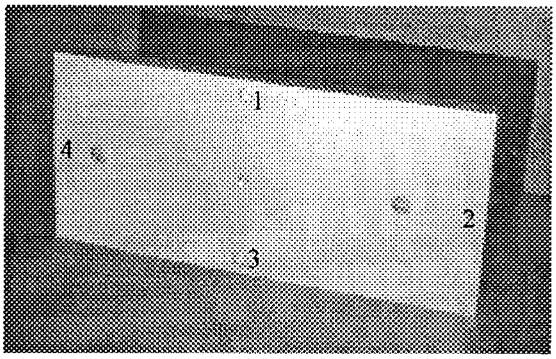


Figure 2.23: A image of a block

segmented into 4 parts, which were fitted to straight line functions. The precision of edge points presented as the RMS error derived from the straight line is showed in Table 2.5. The average precision of the edge points achieved in this example is 0.046 pixel. More than 6 images were tested with the different orientations of the block. The results are approximately the same as the result in Table 2.5.

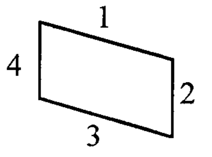
Line No.	1	2	3	4	Average	
No. of Edge Points	321	126	345	146		
Precision (pixel)	0.050	0.040	0.052	0.043	0.046	

Table 2.5: Precision of edge points in the segmented lines

2.8 Summary

An edge detection algorithm has been presented based on a linear model which locates an edge point in the operator window with sub-pixel accuracy. The linear model comprises two aspects: one is to determine the peak of the intensity change in gradient direction, which is performed by the Förstner Operator (Förstner, 1987); the other is to limit the unstable edge location in the edge direction by the introduction of a linear constraint which passes through the window centre and meets the edge at right angles. To improve the accuracy of the edge location, an edge point is determined by a weighted average of two points derived from both sides of the edge. Another implementation is to use a round operator window instead of a square window, so that the result will not be influenced by the difference of edge orientations. Tests show that 5 × 5 window size is the most appropriate choice for edge location, considering the effect of the weighting function, the use of round window and the influence of curved edges. The scattered edge points are then chained by following neighbouring pixels, based on the minimum local distance. The direction of the edge chaining corresponds to the local direction of the edge, which is attached to each edge point as a primitive for the succeeding process.

Chapter 3 Line Segmentation

3.1 Introduction

3-D surfaces can be reconstructed by matching corresponding extracted edges from two or more overlapping images. Close range images taken from different camera positions, however, will reveal different views. One surface of an object occurring in one image may not be imaged on one or more of the other images. In addition, the features derived from the chained edge pixels in chapter 2 will differ in their geometric properties on the overlapping images obtained from different view points. An edge in one image may correspond to parts of two edges or more in another image. Therefore, a procedure of construction of objects in a close range image based on edge matching by simply determining conjugate edges will fail. In order to create conjugate edges in stereo images, it is necessary to segment edges into distinct edge features. Furthermore, a simple chained edge has no geometric meaning, since in industrial environments regular shapes, such as ellipses and straight lines, often occur as elements of object boundaries. For the purpose of object recognition and inspection, therefore, it is generally more relevant to present the boundaries of objects revealed in the images in geometric form, because simple geometric functions provide more reliable information and are easier to calculate.

The line segmentation problem has been addressed extensively in computer vision literature. The major approaches could be classified into the two categories: those focusing on detecting corners of intersecting edges; and those which detect straight lines and curves from a global view. These two approaches will be reviewed below.

3.1.1 Corner Detection

Many algorithms have been developed to detect dominant corners on object boundaries. Grimson and Pavlidis (1985) found the breakpoints of a curve by comparing the original and a smoothed version of the curve. A discontinuity is then easily detected and regular curve fitting is performed only between the discontinuities. Meer et. al (1986) suggested a method to detect dominant points by first determining the optimal scale of Gaussian-like convolutions of multiple-scale representations of the boundary. Subsequently, corners are detected at this optimal scale. The method uses a single scale that best represents the object. Asada and Brady (1988) proposed the curvature primal sketch to represent changes in boundary curvature. Boundary symbolic features are each detected at a different scale based on some preconceived notions of a best scale for a particular feature. The resulting representation is a multiple-scale interpretation of the boundary. Rattarangsi and Chin (1990) analysed the scale space of isolated simple and double corners, which is transformed into a tree that provides a simple but concise representation of corners at multiple scales. A coarse-to-fine tree parsing technique is used to detect corners at the maximum observable curvature at a majority of scales. Philippe et al (1991) detect corners of planar curves by repeatedly convolving the signal with a very small averaging mask weighted, by a measure of the signal continuity at each point.

The techniques of line segmentation by detecting corners on an edge will result in a small number of edge pieces which can be fitted to different geometrical shapes. However, they have a fundamental short coming: there is no difference between corners and points on high curvature curves corrupted by noise. It is possible that a well-defined corner, which is the intersection of two straight lines with slightly different directions, or a tangent point between a straight line and a regular curve, will not be detected. Some points on a small circle will be regarded as corners based on the same criterion, due to errors in the edge detection process. This ambiguity is caused by the analysis being carried out only on very local areas. A global view is necessary to segment the lines by analysing the characteristics of the edges.

3.1.2 Global Line Segmentation

Global line segmentation aims to directly segment digital lines into straight lines and circular arcs under some criteria. Ramer (1972) presented a simple algorithm to approximate planar curves by polygons, based on a maximum offset from straight lines. Fischer and Bolles (1986) described two methods, one of them passed a "stick" of a certain width and length over the curve, and the other studied the curve from different "views". A selection of break-points was made according to the maximum number of votes obtained from these views. Both methods are based on segmenting the curve over different scales, and on perceptual organisation.

In recent years, two methods has been proposed for line segmentation. The split-and-merge method is based on the offset from a straight line. In the split phase, the input data is segmented to assure that each segment fulfils a certain condition. In the merge phase, redundant breakpoints that have been produced during the split phase are eliminated. The other method is ψ-s algorithm. The ψ -s domain in 2-D consists of a plot of the orientation (ψ) versus length (s) of the original spatial curve. The slope of the line corresponds to the curvature of the original curve. A straight line in the spatial domain appears as a horizontal line in the ψ -s domain, and a circular arc appears as an arbitrary straight line. The ψ-s curve for a spatial curve is constructed by computing the directions between points. These methods have been developed in planar curves by Grimson (1986) and Wuescher and Boyer (1991). Krupnik and Schenk (1992) extended these algorithms to segment curves in 3D object space.

The algorithms for global segmentation of geometrical lines allows detection of straight lines and circular arcs. The split-and-merge method can detect straight lines, but it is not easy to merge a group of short straight lines into a regular curve, especially when an edge is composed of two or more curves. The ψ-s method is very sensitive to noise, and it only provides a rough segmentation. Furthermore, a circle in object space is often projected as an ellipse in image space, the description of circular arcs is not adequate for presenting the shape of curves correctly. However, the concept of ψ-s domain, based on the local orientation on a curve, motivated the method used in this thesis.

Hough Transform is a well known technique for shape and motion analysis in images containing noisy, missing, and extraneous data. It converts a difficult global detection problem in images into a more easily solved local peak detection problem in a parameter space. Hough Transform can be applied to any function, but is easy to explain for a straight line. For detecting straight lines in an edge image, any single point in the image space has a line passing through it. the equation of the line is $d = x \cos\theta + y \sin\theta$, where d is the length of the normal to the line from the original of image coordinate system, and θ is its orientation. Each point in the image space can be presented as a curve in the parameter space. Points which lie on the same straight line will result in curves that intersect in a point in the parameter space, whose co-ordinates are the parameters of the line. An intersection point with a large number of curves passing through it indicates a line in the image space. A survey of Hough Transform technique can be referred to Illingworth and Kittler (1988).

3.1.3 A New Method for Line Segmentation

In this chapter, the suitability of the algorithm developed in this thesis for line segmentation is demonstrated, based on the analysis of the local directions of edges. The edge detection method discussed in chapter 2 provides edge local direction for each edge point. The local direction of an edge point is a 1-D value which clearly reflects the trend of the edge at the edge point with respect to the next point. If a group of edge points contain the same trend in edge direction, it means that these edge points are of similar geometry. The method attempts not to find corners, but directly to find the basic components of straight lines or regular curves by the assessment of the local direction at the edge points along a complete edge. The approach of the method is similar to Hough Transform for detecting local peak in parameter space, but the parameter of direction for each point is directly obtained from edge detection process, rather than transforming from image coordinates, so that the problem of line segmentation is reduced to one dimension. Another characteristic of the method is that edge points have already been connected. Therefore, the process of line segmentation can be performed on each edge, without processing all edge points in the whole image at one time as the normal Hough Transform methods have done. The principles of the line segmentation in the method are: all edge points on a straight line should indicate approximately the same edge direction; and all edge points on a regular curve should indicate the same sign of the difference in direction.

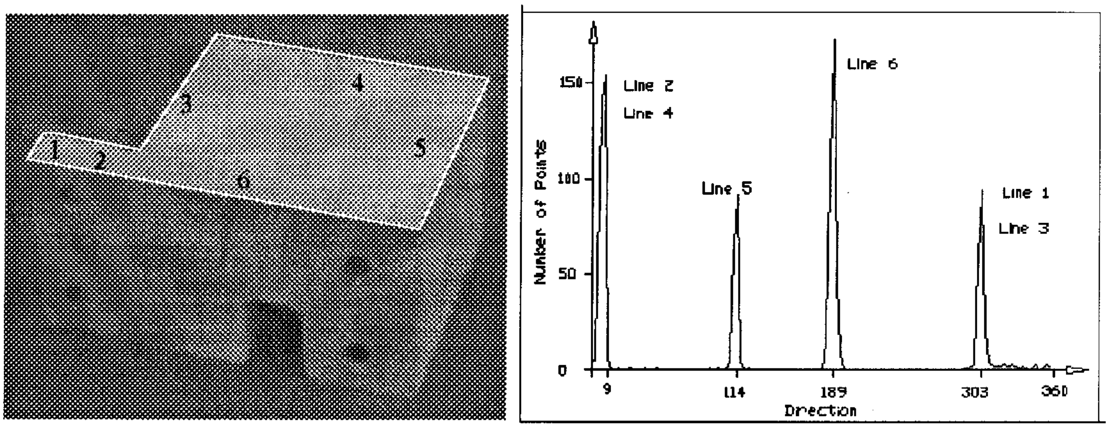
In section 3.2 and 3.3, the process of line segmentation is first performed on a single edge. Straight lines and regular curves are detected respectively, based on the above principles of line segmentation. In section 3.4, the basic elements of geometric shape determined by the above steps are extended in both directions to form end points. These geometrical lines extracted from different edges with the similar properties are then merged. In section 3.5, straight lines and open regular curves are linked at their terminals, so that the boundaries of an object can be closed. In section 3.6, surface patches are formed in terms of certain constrains. The results of experiments on these procedures will be given in section 3.7.

3.2 Detection of Straight Line

The procedure for line segmentation starts with the classification of the local directions assigned to each of the edge points by Eq[2.14]. Figure 3.1(b) shows the number of edge points in the range of directions for one edge displayed in figure 3.1(a). The edge points are separated into a small number of groups which correspond to the straight lines. By detecting the local maxima of edge directions, the general direction of a straight line can be determined. However, there exist two factors which influence the detection of straight lines. An edge may be formed by a mixture of straight lines and curves, so that a simple threshold which tests local maxima will not suit all situations. Secondly, errors existing in edge directions will create many local maxima which are confused with the true direction of straight lines. Figure 3.2 (b) displays the distribution of edge directions on the edge which surrounds the cylinder as shown in figure 3.2 (a). The local maxima of the distribution outside the range of directions of straight lines vary significantly, depending on the type of curves and their smoothness.

In order to solve this ambiguity problem, constraints must be used to select the correct peak from the many local maxima in the distribution of edge directions which applies to the straight lines. By analysing the distribution of edge points on straight lines and other areas, several significant phenomena are evident (figure 3.1(b) and 3.2(b)). The distribution of the edge directions of a straight line is often characterised by a sharp change, with the peak relatively higher than its neighbouring maxima, and its width very narrow. Therefore, for the detection of straight lines, it is necessary to assess the width and the relative height of the distribution. The width can be assessed by fitting the distribution in the region of local maxima to a quadratic model:

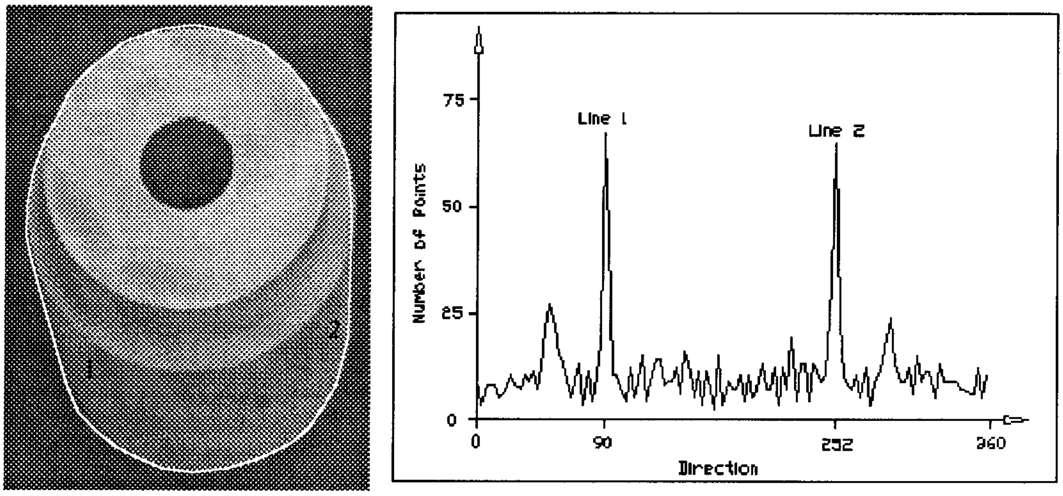
$$n = ax^2 + bx + c \tag{3.1}$$



(a) (b)

Figure 3.1: A machine part: (a) the extracted edges;

(b) the distribution of edge points in the range of directions



(a) (b)

Figure 3.2: A cylinder: (a) the extracted edges;

(b) the distribution of edge points in the range of directions

Where n is the number of edge points in a certain range of directions (3 degrees). x is the coordinate in the direction axis. a, b and c are the parameters of the model. The parameter a reflects the shape of the distribution as shown in figure 3.3. The greater the absolute value a, the narrower the width of the extension. If 5 numbers are chosen from equally spaced points along the extension, with the local maximum in the central position, we obtain the expression for the residual of the quadratic function for a region of the distribution :

$$V_i = a x_i^2 + b x_i + c - n_i \qquad (3.2)$$

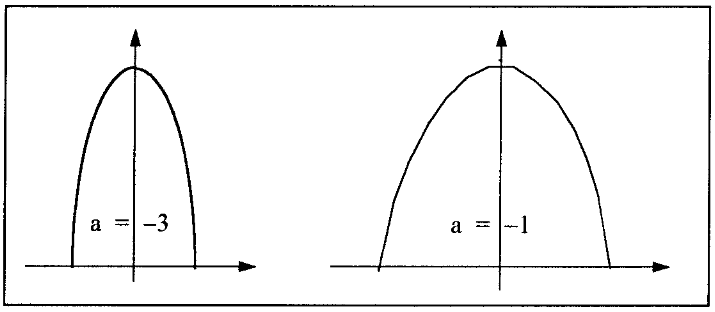


Figure 3.3: The shape of the distribution in the different value a.

The subscript of x_i presents the position in the description with respect to the local maximum, and n_i is the number of edge points in the corresponding range. i = –2,–1, 0, 1, 2. Using a least squares adjustment, the parameter a can be presented as the following:

$$a = -\frac{1}{14}(2(n_0 - n_{-2} - n_2) + n_{-1} + n_1) \qquad (3.3)$$

From Eq[3.3], an expression for the assessment of the width of the distribution is written as:

$$T_W = 2(n_0 - n_{-2} - n_2) + n_{-1} + n_1 \qquad (3.4)$$

If an edge lies on a polygon, there exist just a few edge points whose directions are between two maximums of the distribution as shown in figure 3.1(b). However, more edge points will exist between two maximums of the distribution, if an edge lies on the combination of curves and straight lines as shown in figure 3.2(b). For the detection of a straight line linking with curves, the relative height of a maximum in the distribution, must be considered. The synthetic measure of a local maxima can be given by

$$T = T_W - n_{-3} - n_3 - n_{-4} - n_4 \qquad (3.5)$$

The general direction of a straight line can be determined by detecting T above a threshold. Figure 3.4 (a) and (b) show examples of the distribution of edge points on a straight line and a curve respectively. The numbers of edge points n_{-3}, n_3, n_{-4} and n_4 on a straight line in figure 3.4 (a) tend to be zero, while those numbers on a curve in figure 3.4 (b) are of significant effect on the assessment. Even though the numbers of edge points in the peak are the same, a straight line in figure 3.4 (a) is detected, while the assumption of a straight line in figure 3.4 (b) is rejected.

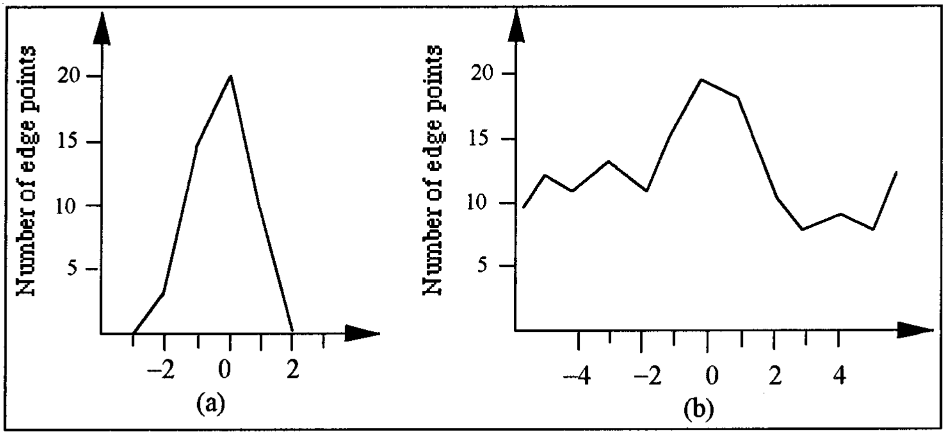


Figure 3.4: Distribution of edge points: (a) on a straight line; (b) on a curve.

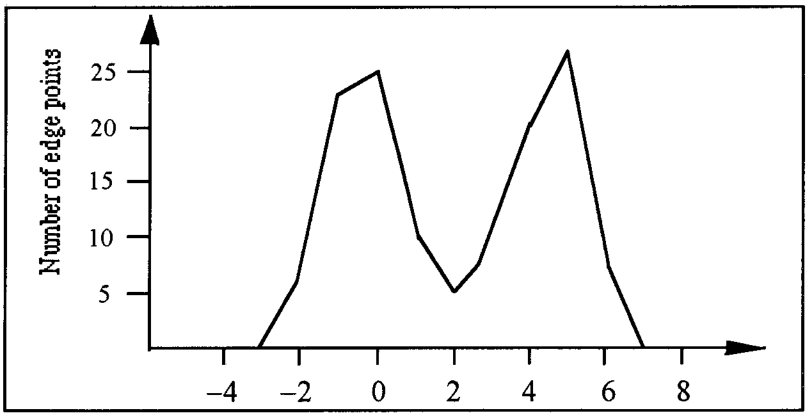


Figure 3.5: Distribution of edge points on two straight lines

Since the order of the straight lines in figure 3.1 (a) is not the same as that in figure 3.1 (b), which ranks the lines in terms of direction, parallel boundaries in object space may not be parallel in the image, due to the distortion of image projection. Therefore, the two peaks of distribution may overlap as shown in figure 3.5. If two peaks are close to one another, the straight lines detected directly by assessment of T will fail due to the large values of n_3, n_4 in Eq[3.5]. In order to avoid such a situation, the peak search includes two steps. The first step is to detect local maximums by checking the value T_W using a low threshold. The second step is then to check whether a peak exists near each peak being searched, and if so, to determine the side on which it lies on. If one peak exists near the current peak, the assessment of T can be written as follows:

$$T_l = T_W - n_{-3} - n_{-4} - n_{-5} - n_{-6} \qquad (3.6)$$

or

$$T_2 = T_W - n_3 - n_4 - n_5 - n_6 \qquad (3.7)$$

The numbers of edge points in negative sign indicate the numbers in the left side of a local peak. T_l is computed when a peak exists close to the right side of the current peak, while T_2 is derived when a peak exists close to the left side. Threshold setting is important for checking whether these edge points are composed of a straight line. Some methods of threshold selection have been proposed by Reddi et al. (1984), Wang and Haralick (1984), and Papamarkos and Gatos (1994). However, most methods are used for image segmentation, where the number of thresholds are known. These method try to find the optimum valleys between peaks. Since the number of peaks corresponding to straight lines in an edge of the above example is unknown, the threshold for line segmentation is difficult to determine. In the thesis, an ad hoc threshold is given to the line segmentation, which may not be fitted to all situations. The problem of setting an optimal threshold is needed to be further studied. All detected peaks are listed in the order in terms of their magnitude. A range of direction, such as 8 degrees, is given to each peak. Straight lines are detected by searching linked edge points, whose directions lie within the range of the corresponding peaks. Edge points, which number above a certain limit (eg. 15 points), are then fitted by a straight line function using a least squares algorithm. The error equation for each edge point along a straight line can be written as:

$$V_i = \sin\alpha\ X_i - \cos\alpha\ Y_i + \rho \qquad (3.8)$$

Where X_i, Y_i are the coordinates of an edge point; a is the direction of the straight line; ρ is the distance from the centre of the image coordinate system to the line. Since the assessment parameters α and ρ in observation equation (3.8) are non-linear. it is difficult to estimate the parameters α and ρ directly by the equation. The alternative parameters α and ρ can be estimated firstly in the following equation:

$$V_i = \lambda_1 X_i + \lambda_2 Y_i + 1 \qquad (3.9)$$

The parameters of a straight line are then determined by λ_1 and λ_2. We have:

$$\begin{aligned} \sin\alpha &= \frac{\lambda_1}{\sqrt{\lambda_1^2 + \lambda_2^2}} \\ \cos\alpha &= -\frac{\lambda_2}{\sqrt{\lambda_1^2 + \lambda_2^2}} \qquad (3.10) \\ \rho &= \frac{1}{\sqrt{\lambda_1^2 + \lambda_2^2}} \end{aligned}$$

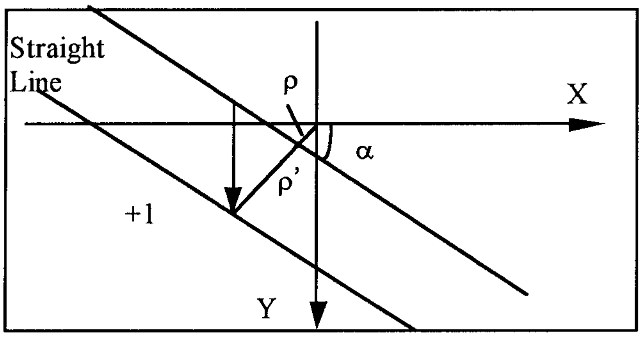


Figure 3.6: Edge points on the straight line move in Y direction

It should be noted that if a straight line passes through the origin of the image coordinate system, the parameter λ_1 or λ_2 tends to approach infinity, because ρ is equal to zero. In this case, a derived by the Eq[3.10] may be subject to calculation error. In order to avoid this situation, all edge points on the straight line are moved a certain known distance from the coordinate centre as shown in figure 3.6, so that the parameter a will be constant. The movement of the straight line is depending on the parameters λ_1 and λ_2. If $|\lambda_1|$ is larger than $|\lambda_2|$, all edge points move one pixel in Y direction and the error equation for each edge point can be written as:

$$V_i = \lambda'_1 X_i + \lambda'_1 (Y_i + 1) + 1 \qquad (3.9')$$

The parameters α' and ρ' can be calculated by Eq[3.10] by substitution of λ'_1 and λ'_2. for λ_1 and λ_2. The correct parameter α is the same as α', while the parameter ρ is equal to ρ' – cos(α). If all edge points are moved in X direction, in the case of $|\lambda_2|$ being larger than $|\lambda_1|$, the parameter ρ is equal to ρ' – sin(α).

3.3 Detection of Regular Curves

Excluding points detected as the elements on straight lines, other edge points are regarded as the candidates to describe regular curves. A regular curve is a curve which could be presented as a simple mathematical function. Currently, second order polynomials are used to define regular curves. Geometric groups of edge points are separated by testing the sign of the difference of direction of the edge at each edge point. Because of noise in these directions, the difference is calculated from points a certain distance apart, rather than adjacent elements. The linked edge points with the same sign of difference in direction are fitted by a curve functions follows:

$$F(X,Y) = aX^2 + bXY + cY^2 + dX + eY + 1 \qquad (3.11)$$

where a, b, c, d and e are the parameters of the polynomial function. Unfortunately, the error equation for each edge point cannot be derived directly from the above equation, because the determination of the normal distance from any point (X_e,Y_e) to a regular curve is not simple. A possible method is to determine an approximation of the distance, assuming that the distance is small. Let the equation of a curve be F(x,y) = 0. The distance from a point (X_i,Y_i) to the point (x_i,y_i) on the curve is :

$$V_i^2 = \Delta X_i^2 + \Delta Y_i^2 \qquad (3.12)$$

where $\Delta X_i = X_i - x_i$, and $\Delta Y_i = Y_i - y_i$. Assuming that ΔX_i and ΔY_i are small, F(x,y) is expanded as follows:

$$F(x_i,y_i) = F(X_i,Y_i) - F'_{xi}\,\Delta X_i - F'_{yi}\,\Delta Y_i = 0 \qquad (3.13)$$

Where F'_x and F'_y denote the first derivatives of the equation. The distance V_i can be obtained by minimising V_i^2 under the constraint of Eq[3.13]. The evaluation function is assumed as:

$$\Phi = \Delta X_i^2 + \Delta Y_i^2 + 2\,k\,(F(X_i,Y_i) - F'\Delta X_i - F'\Delta Y_i) \qquad (3.14)$$

Where k is a parameter. By differentiating Eq[3.14] with respect to ΔX and ΔY, and setting the results to be zero, the following equations are derived:

$$\frac{\partial \Phi}{\partial X_i} = 2\,\Delta X_i - 2\,k\,F'_{Xi} = 0 \qquad\qquad \frac{\partial \Phi}{\partial Y_i} = 2\,\Delta Y_i - 2\,k\,F'_{Yi} = 0 \tag{3.15}$$

The solutions to Eq[3.13] and Eq[3.15] are

$$\Delta X_i = \frac{F(X_i, Y_i)\,F'_{Xi}}{F'^2_{Xi} + F'^2_{Yi}} \qquad\qquad \Delta Y_i = \frac{F(X_i, Y_i)\,F'_{Yi}}{F'^2_{Xi} + F'^2_{Yi}} \tag{3.16}$$

Therefore, the distance V_i from the point (X_i, Y_i) to the curve can be calculated by:

$$V_i = \frac{F(X_i, Y_i)}{\sqrt{F'^2_{Xi} + F'^2_{Yi}}} \tag{3.17}$$

Using the definition of a regular curve as a function of second order polynomials in Eq[3.11], the distance V_i from an edge point to the regular curve can be written as:

$$V_i = \frac{aX_i^2 + bX_iY_i + cY_i^2 + dX_i + eY_i + 1}{\sqrt{F'^2_{Xi} + F'^2_{Yi}}} \tag{3.17'}$$

where

$$F'_{Xi} = 2aX_i + bY_i + d \qquad\qquad F'_{Yi} = bX_i + 2cY_i + e \tag{3.18}$$

The coefficients are obtained by minimising the distance V_i in Eq[3.16'] using a least squares adjustment. If there exist outliers in data, a robust approach, such as least-median square (Rousseeum and Leroy, 1987), may be more effective, but this has not proved necessary in this study. As the function of the residual V_i is non-linear, the calculation must be iterative. The RMS error derived from the least squares adjustment is used to assess whether the edge points define a regular curve. If the value is above a certain threshold, the group of edge points is regarded as a natural curve and rejected. The parameters a, b, c, d and e provide basic parameters of the polynomial function, which can be transformed to the parameters of an ellipse, hyperbola or parabola. In the case of an ellipse, the parameters are the direction of the major axis, the central coordinates and the magnitude of the major and minor axes.

3.4 Line Extension and Merging

The basic elements of geometric shape determined by the above steps should be extended in both directions to form end points, while these geometrical lines extracted from different edges with the similar properties are required to be merged. If we have priori knowledge about the scene object, the model-based approach (El-Hakim, 1992) can be used to solve the line extension and merging problem. When the environment is unknown, however, the problem should be solved based on already extracted geometric information.

In the above section, the local directions of an edge are used as a primitive to segment edge points into geometrical groups. These geometrical lines, however, are often split due to errors in local directions of an edge. In this section, a distance from an edge point to an assessed line is used as a primitive to extend that line in both directions to form end points. In referring to the measurement of how closely an edge point is related to an assessed line, a primitive "direction", tests the correspondence of the local direction of that edge point to the direction of a straight line, while the primitive "distance" describes the displacement length of an edge point to the straight line. Since assessed lines have been formed by the above process, the distance can be used as a geometrical primitive for further line segmentation. The process of line extension and merging in the following includes: line extension on a single edge, extraction of geometrical points, and line merging on different edges.

3.4.1 Line Extension on a Single Edge

Figure 3.7 shows an assessed straight line detected with two ends A and B. The direction of an edge point adjacent to A is outside the range of direction of the straight line, but its location is close to that of the assessed line. Such a point can potentially be added to the elements which have been fitted to a straight line function, to extend the geometrical line, provided it is within a certain distance, referred to as the threshold, of the assessed line. The resulting extended line with new parameters is again extended by measuring the closeness of additional edge points. In this way, a line is gradually extended and its location becomes closer to the complete line. Currently, the offset threshold is set at 0.5 pixel. If the distance from an edge point to the geometric line is above the threshold, the process of extension stops. The end points are determined by projecting the edge points at two ends to the geometrical line.

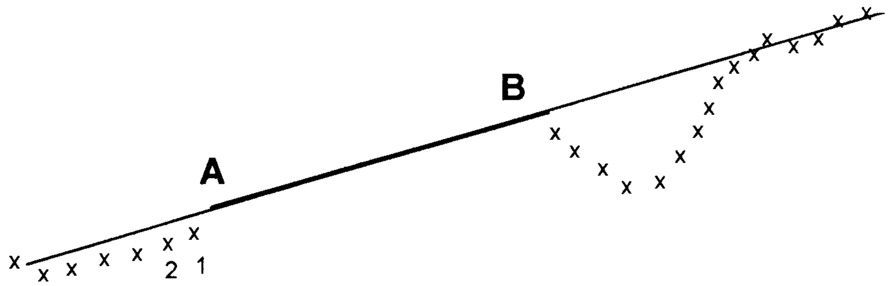


Figure 3.7: Extension of a straight line

However, this process cannot be performed on the edge points beyond point B, which display a different situation. Due to image noise or blemishes on the boundaries, the edge points beyond B depart from the straight line path for a certain distance and then again follow the straight line. This indicates that line extension should be able to bridge the gap and extend the edge points to the actual ends of the line. In the research, gaps in the edge points are limited to 15 points. In this way, the process of line extension is more tolerant to noise.

Regular curves are treated in the same way as straight lines, so that the method of line extension discussed above can also be applied to regular curves. However, a regular curve contains more degrees of freedom than a straight line, since the number of the coefficients in a regular curve function is 5, whereas it is 2 for a straight line. Figure 3.8 shows that a regular curve is detected at ends A and B. E1 is a curve determined by edge points between end A and end B. The points 1, 2 and 3 satisfy a certain offset from the curve E1, while the point 4 does not. After the first 3 points are added to the basic edge points to create a new curve E2, point 4 now satisfies the offset from the curve E2 and could be used to define a new curve function. Gradually, as the ends are extended, the assessed curve becomes closer to its correct shape.

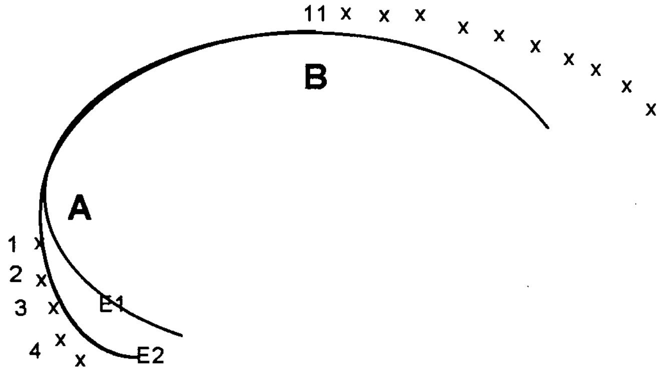


Figure 3.8: Extension of a regular curve

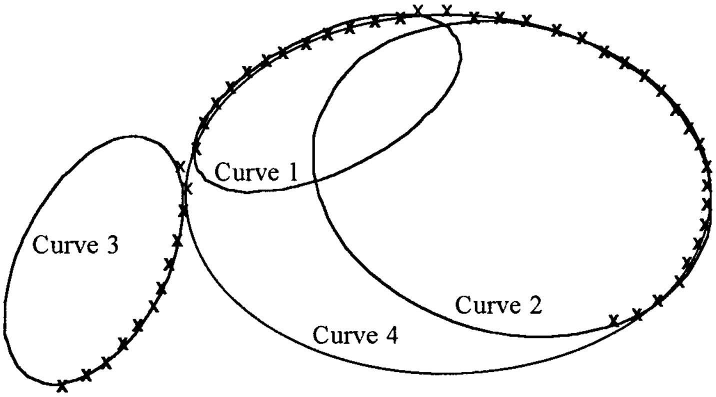


Figure 3.9: One edge separated into several short curves

The shape of an assessed curve is sensitive to the noise, and edge points near ends of sections of a curve have a greater effect on changing the shape of the curve than the points which form the curve in the middle of the basic elements. This shape change is clearly reflected in the extended sections. In figure 3.8, the distance from edge points to the curve from point 11 to the curve end gradually become large. If the point 11 at point B does not satisfy the offset criterion, no further edge points will be close to the assessed curve, and the regular curve cannot be extended. The shape change of the extended assessed curve is influenced by not only the errors in the edge points, but also the number of edge points. In figure 3.9, one edge is separated into several short curves (1, 2 and 3). In fact, the edge points on curve 1 could be merged with the edge points on curve 2 to form curve 4, while their geometrical properties are different from those of the edge points on the curve 3. In order to segment lines correctly, edge points on two neighbouring curves may have to be fitted to a new curve. The estimation of whether the two groups of edge points are combined together is dependent on testing the RMS error, which is derived from a least square adjustment to determine the assessed curve. If the

RMS is below a threshold, the properties of the two groups of edge points are regarded as the same, and they are merged to form a new curve. Otherwise, the two curves remain unchanged.

3.4.2 Extraction of Nodes

A node is the junction of two arcs. Since line extension is based on the offset criterion of the distance of an edge point to an assessed geometrical line, edge points at the connection of two neighbouring lines may be included into either group of the elements which form two geometrical lines. This situation often occurs where a straight line links with a curve, or two curves are connected. In figure 3.10, the edge points 1, 2, 3 and 4 can be included in either the edge elements of the straight line, or the regular curve. As discussed above, the edge points near ends will have more influence on the shape formation of assessed lines than the edge points away from ends. The straight line in the figure will be assessed in the direction L2, because the elements fitted to the straight line include points 3 and 4. In the same way, a regular curve is formed in the shape C2, since it is influenced by edge points 1 and 2. In order to create correct geometrical lines with correct ends, the edge points at the connection of two neighbouring lines must be separated or only one edge point should be assessed as a common point.

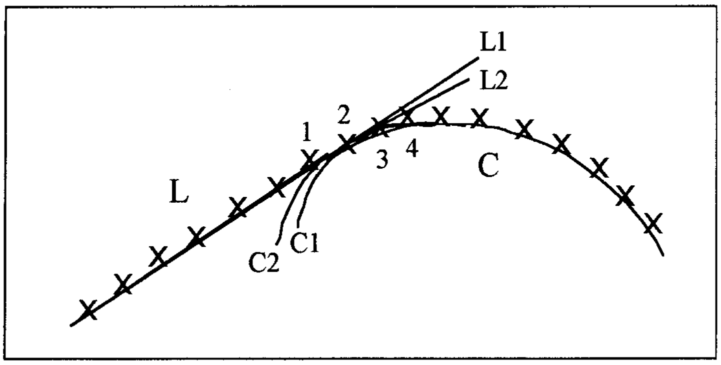


Figure 3.10: Edge points on connection of two geometrical lines

In figure 3.10, the straight line is a tangent to the regular curve at the geometrical point. The calculation of the geometrical point will be discussed in next section. The overlapping edge points are separated by testing the distances from the geometrical point to these edge points. The edge point with the smallest distance is considered as a reference point to separate the edge points 1, 2, 3 and 4 into the groups of elements for the two neighbouring lines. The correct parameters of the two geometrical lines can be calculated based on the newly classified edge points.

3.4.3 Line Merging from Different Edges

Line extension and merging is performed on the straight lines and regular curves derived not only from one edge, but also from different edges. Since edges are chained along the pixels with high gradient magnitude, a geometric line may be divided into several edges. Lines with the same geometric properties should be merged into a single line. The criterion for line merging is that the ends of the separate lines should be close to each other, and the elements of one line should satisfy the offset criteria, that they should be within a certain distance of the other line. Since the parameters of long lines are more accurate than those of short lines in the merging process, long lines are assumed as basic lines against which to check the offset distances of the elements of short lines. If the check is successful, the edge points corresponding to merged lines are then fitted to a geometric function. A straight line is simply presented by its two ends. A regular curve is described by parameters of a polynomial function and the coordinates of its end points. The numbering hierarchy of end points corresponds to the edge direction, ie. a line formed from end1 to end2 separates the intensity of an image into two areas, with the intensity on the right higher than that on the left. If end1 of one line is linked to end2 of another, the property of the combined line remains the same.

3.5 Lines Linking at Terminals

A geometrical point is defined as one where two neighbouring lines are connected. Straight lines and open regular curves should be linked at their terminal points, so that the boundaries of objects can be closed. A terminal can be created by intersecting combinations of straight lines and regular curves, which includes: intersection of two straight lines, intersection of two regular curves, intersection of a straight line and a regular curve, and a tangent point connecting a straight line and a regular curve.

3.5.1 Intersection of two Straight Lines

The intersection of two straight lines requires the solution of two linear equations in two unknowns x and y. The equations have the form :

$$\begin{aligned} \sin\alpha_1 X - \cos\alpha_1 Y &= -\rho_1 \\ \sin\alpha_2 X - \cos\alpha_2 Y &= -\rho_2 \end{aligned} \qquad (3.19)$$

The solutions of this form can be written as following:

$$\begin{aligned} X &= \frac{\rho_2 \cos\alpha_1 - \rho_1 \cos\alpha_2}{\cos\alpha_1 \sin\alpha_2 + \cos\alpha_2 \sin\alpha_1} \\ Y &= \frac{\rho_2 \sin\alpha_1 - \rho_1 \sin\alpha_2}{\cos\alpha_1 \sin\alpha_2 + \cos\alpha_2 \sin\alpha_1} \end{aligned} \qquad (3.20)$$

The intersection is placed as the end point for line 1 and line 2, instead of P_1 and P_2 as shown in figure 3.11.

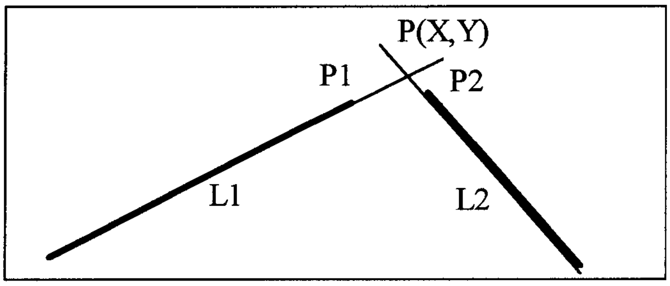


Figure 3.11: Intersection of two straight lines

3.5.2 Intersection of a straight line and a regular curve

The equations of a straight line and a regular curve are as follows:

$$\begin{aligned} \sin\alpha X - \cos\alpha Y &= -\rho \\ F &= aX^2 + bXY + cY^2 + dX + eY + 1 \end{aligned} \qquad (3.21)$$

After removing one unknown Y from above equations, we have the equation of one unknown in the second order degree:

$$a' X^2 + b' X + c' = 0 \qquad (3.22)$$

Where

$$\begin{aligned} a' &= a + b\,tg\alpha + c\,tg^2\alpha \\ b' &= d + e\,tg\alpha + \frac{\rho}{\cos\alpha}(b + 2\,c\,tg\alpha) \\ c' &= c\left(\frac{\rho}{\cos\alpha}\right)^2 + \frac{e\,\rho}{\cos\alpha} + 1 \end{aligned} \qquad (3.23)$$

The solution to Eq[3.21] and Eq[3.22] is

$$\begin{aligned} X &= \frac{-b \pm \sqrt{b'^2 - 4\,a'\,c'}}{2\,a'} \\ Y &= tg\alpha X_{12} + \frac{\rho}{\cos\alpha} \end{aligned} \qquad (3.24)$$

There are two solutions to the equations (3.21). The correct intersecting point P should be near the ends P_1 on the straight line and P_2 on the regular curve as shown in figure 3.12.

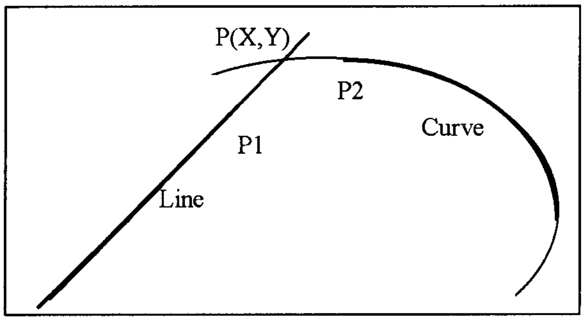


Figure 3.12: Intersection of a straight line and a regular curve

3.5.3 A Tangent Point Connecting a Straight Line and a Regular Curve

Another type of a terminal which connects a straight line and a regular curve is a tangent point, where the straight line is tangent to the regular curve, as shown in figure 3.13. In order to make the calculation simple, it is assumed that the shape of the regular curve does not change and the terminal P moves along the curve until the direction of the straight line is identical to that of the tangent of the curve. The final result should satisfy the equation as follows:

$$\Phi = F'_x\,(X_0 - X) + F'_y\,(Y_0 - Y) \qquad (3.25)$$

Where F'_x and F'_y is described in Eq[3.18]; (X_0,Y_0) are the coordinates of one end of the straight line; (X, Y) are the coordinates of the tangent point. For one point on a curve, the magnitude of the value F indicates how close the point is to the tangent point, while the sign of the value F illustrates the side of the tangent line on which the point is placed. If there exist two points with different signs of F in Eq[3.25], the tangent point must exist between the two points. Using simple dichotomising search, the tangent point can be approached by an iterative calculation.

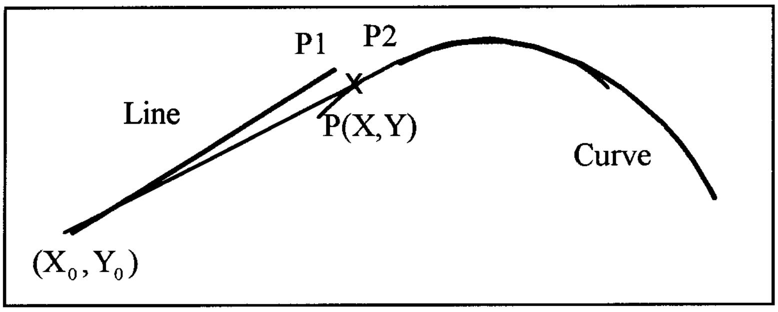


Figure 3.13: A tangent point connects a straight line and a regular curve

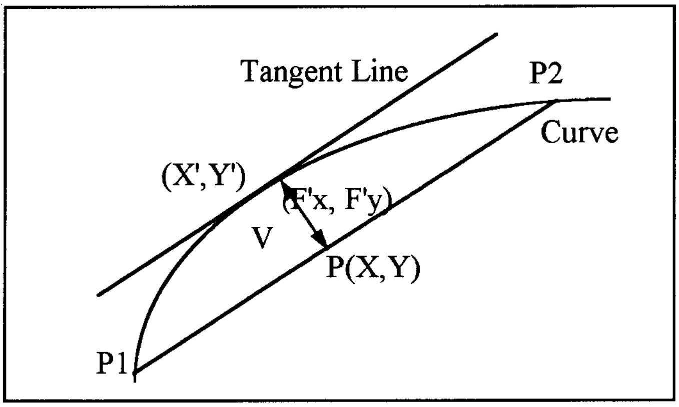


Figure 3.14: One point projected on a curve in the direction (F'_x, F'_y)

The search of a tangent point in the use of Eq[3.25] requires that the searched point P(X,Y) should be along a curve. As a middle point P(X,Y) derived by dichotomising between two points P1 and P2 on a curve, must not be on the curve, as shown in figure 3.14, this point should be projected onto the curve. The distance from a point to the curve is described by a residual V in Eq[3.17']. The correction to position of the point can be made towards (F'_x, F'_y) the corresponding point (X',Y') on the curve by :

$$
\begin{aligned}
X' &= X - \frac{V F'_x}{\sqrt{F'^2_x + F^2_y}} = X - \frac{F(X,Y)\, F'_x}{F'^2_x + F^2_y} \\
Y' &= Y - \frac{V F'_y}{\sqrt{F'^2_x + F^2_y}} = X - \frac{F(X,Y)\, F'_y}{F'^2_x + F^2_y}
\end{aligned}
\qquad (3.26)
$$

3.5.4: Intersection of two Regular Curves

The determination of the intersection of two regular curves is viewed as a solution to two functions as follows:

$$\begin{aligned} F_1(X,Y) &= a_1X^2 + b_1XY + c_1Y^2 + d_1X + e_1Y + 1 = 0 \\ F_2(X,Y) &= a_2X^2 + b_2XY + c_2Y^2 + d_2X + e_2Y + 1 = 0 \end{aligned} \qquad (3.27)$$

However, it is difficulty to calculate the intersecting point by directly solving these two equations. Figure 3.15 displays an appropriate iteration instead of determining the intersecting point, by repeatedly projecting a point onto the two curves using Eq[3.26]. In the first instance, the point P_1 with the distance V_1 is projected on the curve 2 to create the point P_2, which is then projected back onto the curve 1 to create the point P_3. Since the point P_3 is closer to the intersecting point than the point P_1, the intersecting point can be approached by iterative calculation.

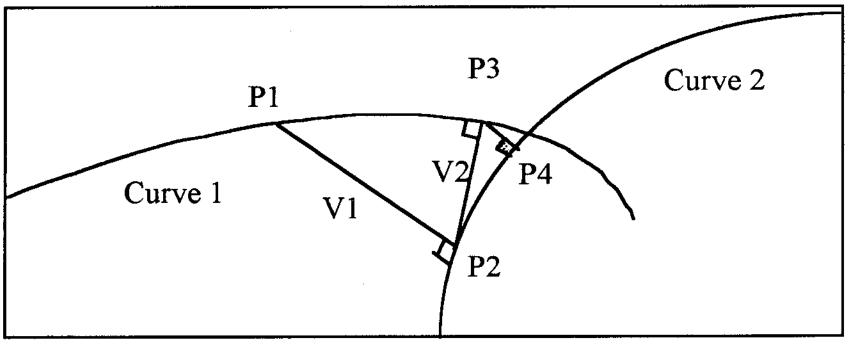


Figure 3.15: Intersection of two curves is approached by iterative calculation

3.6 Generation of Surface Patches

A surface patch is bounded by a sequence of geometrical lines. Each surface patch in image space presents one surface of an object in object space. Since straight lines and open regular curves have been linked at their terminals, it is necessary to chain straight lines and regular curves in a proper order to form surface patches. Figure 3.16 displays an object, which has 7 surface patches in image space. However, there are two ways to make a closed chain at the terminal B. For example, a chain, shown by the black arrows in the figure, bounds two surface patches. In order to chain a sequence of geometrical lines correctly and automatically, some rules should be applied to the process.

Each straight line separates two surface patches, except the area of the background of the object. If a chain at terminals is always kept on the left side of the chain (or right side), the chain will give a correct result, and only one surface patch is bound. This concept can be performed by comparing angles between possible chained directions, and the direction with the smallest angle is chosen as a chaining direction. As shown in figure 3.16 at terminal B, the angle between the direction shown by the white arrow to the previous chained line is smaller than the angle shown by the black arrow, so that the chain at terminal B should follow the white arrow.

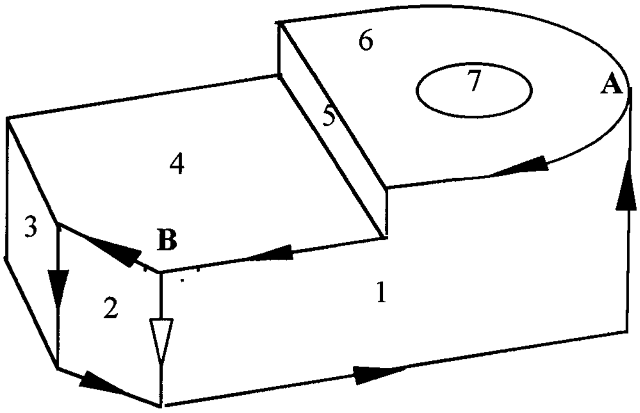


Figure 3.16: Surface patches generated by chaining the geometrical lines

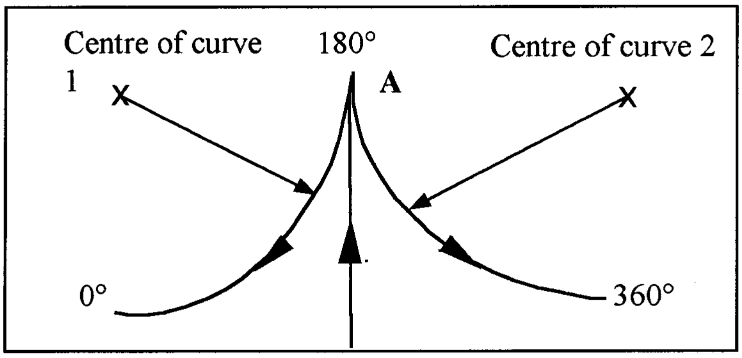


Figure 3.17: Tangent direction of curves at terminal A

The chain method can also be applied to regular curves. The direction of a curve at junction A in figure 3.16 is a tangent direction, which is in the same as the direction of the straight line linked to the curve. Generally, the angle between a tangent direction of a curve and a direction of a straight line could be 0°, 180° or 360° as shown in figure 3.17. However, the difference between 0° and 360° can be distinguished by checking the position of the centre of the curve. If the centre of a curve is on the left side, the tangent direction of the curve on terminal A is 0°, otherwise, it is 360°. The steps for the generation of surface patches include:

1) Divide a straight line or a regular curve into two parts, if there is a junction in the middle of it.
2) Find junction points which link three or more geometrical lines.
3) Generate a surface patch from a starting geometrical line.
4) Chain the geometrical line from one of its ends to another line with the same end.
5) Calculate the directions of each line, if they are linked at a junction.
6) Determine the chained line by comparing the magnitude of their angles.
7) Go back to step 4 until the chained line is the same as the starting line.
8) Compare the properties of the surface patch with those of the previously generated surface patches, and delete the current one if it is the same as a previous determined patch.
9) Go back to step 3 and chain the starting line in another direction or chain other lines until all lines have been chained.
10) Generate surface patches with closed regular curves.
11) Delete the patch with geometric lines which have one side as the background.

A surface patch can be presented as a complete regular curve, such as an ellipse, or a list of terminal numbers with their properties, such as a straight line and an open regular curve. The process of the generation of surface patches is performed on 2D data. Due to poor illumination on some surfaces, the boundaries between these surfaces may not be detected, resulting in 2D surface patches including two or more real surfaces in 3D object space.

3.7 Experiments

The proposed strategy has been tested for line segmentation on real images. Figure 3.18(a) displays the image of a cylinder, which is processed by edge detection as shown in figure 3.18(b). An edge along the contour of the object is segmented into two straight lines and four sections of the ellipses, some of which are merged with the others in terms of geometric similarity. These open ellipses are then linked at terminals to form four surface patches. The linking includes the connections between a straight line and an ellipse, and the intersection between two ellipses. Figure 3.18(d) displays the results of the line segmentation.

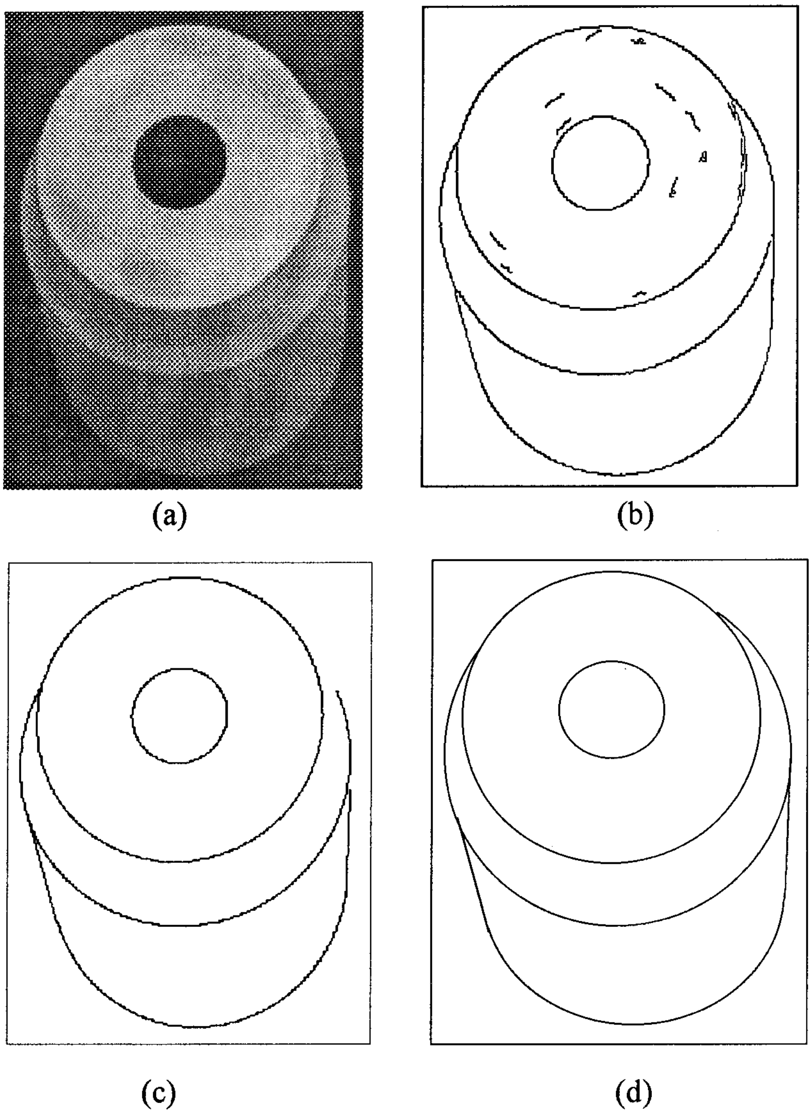
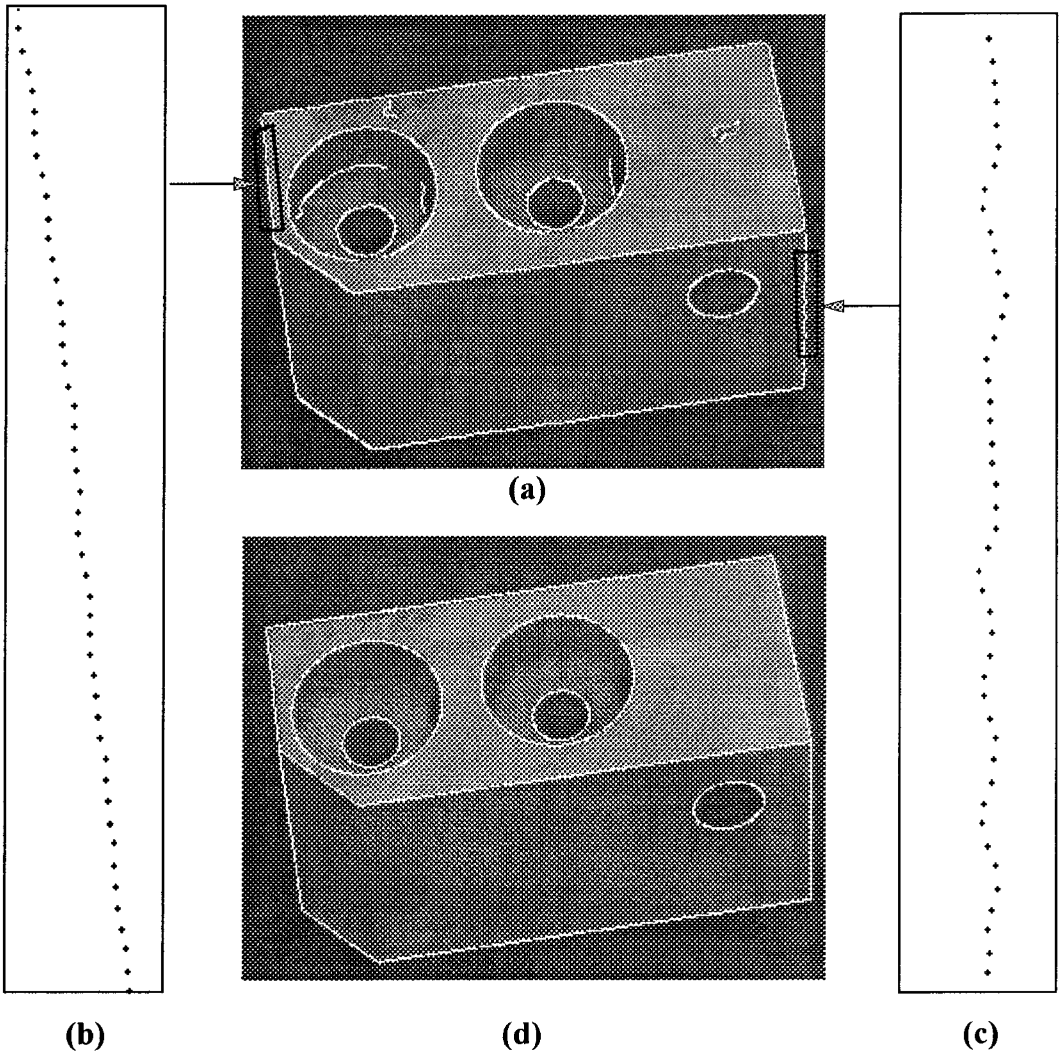

(a) (b) (c) (d)

Figure 3.18 : A Cylinder : (a) An original image; (b) Edges detected from the image; (c) Straight lines and regular curves; (d) Surface patches.

Figure 3.19(a) displays the image of a block. The influence of noise on edge location can be seen in enlarged edge maps, where the contrast of the edge in figure 3.19(b) is higher than that in figure 3.19(c). The proposed method can segment these edge points corrupted by noise into straight lines as shown in figure 3.19(d).


(a)

(b) (d) (c)

Figure 3.19 : A block: (a) Edges detected from the original image; (b) and (c) Edge points in subpixel location; (d) The result of line segmentation.

3.8 Summary

The strategy, developed for line segmentation in the thesis, mainly deals with industrial components. The success of the procedure relies heavily on the detection of edges. The precise edge detection method discussed in chapter 2 enables the description of objects in image space by simple geometric functions, such as straight lines and regular curves. In the method, the local direction of an edge is regarded as an initial primitive to be analysed for the segmentation of edge points. Straight lines are detected by grouping edge points with approximately the same local direction, while regular curves are detected by checking the sign of the difference of edge local directions. The detected geometrical lines are extended in both directions, depending on an offset criterion. The straight lines and regular curves with the same properties are then merged. In order to close boundaries of objects in image space, straight lines and open regular curves are linked at their terminal points. Finally, surface patches are generated by constrained chaining the geometrical lines. The results of linc segmentation require less data to clearly represent important features of imaged objects.

Chapter 4 Camera Orientation

4.1 Introduction

Surface patches of an object created by the above process are presented in 2D image space. If the object is recorded from two or more different view points, 3D position on the object can be measured. Camera orientation is an important part of digital photogrammetry. It establishes the relations between the 3D object coordinate system and the 2D coordinate system in each image. These relationships are given as a function of the camera parameters which are solved by bundle adjustment using 3D control points and their corresponding coordinates on each image. In digital photogrammetry, artificial targets are usually designed as control points to facilitate automatic detection and identification. The general process of camera orientation consists of the following steps:

- Image acquisition of the calibration frame;
- Digital search for the targets in the images;
- Precise location of the targets in the images;
- Determination of the correspondence between control points and their image targets;
- Bundle adjustment to obtain camera parameters.

Each of these stages is a precursor to its next stage, and is essential for the completion of the process of camera orientation.

4.1.1 Formulation

The transformation between image and object coordinate system is based on the precise determination of the camera parameters, which are described by the collinearity equations. The collinearity equations form a perspective transform which mathematically describes the relation that the object point, the perspective centre of the camera and the measured image point ideally lie on a straight line. The transform includes parameters for interior and exterior orientation and additional parameters which model lens distortion and image shear. The collinearity equations (Kraus, 1993) are defined by :

$$\begin{aligned} x_j - x_0 + \Delta x_j &= -S_x f \frac{m_{11}(X_j - X_C) + m_{12}(Y_j - Y_C) + m_{13}(Z_j - Z_C)}{m_{31}(X_j - X_C) + m_{32}(Y_j - Y_C) + m_{33}(Z_j - Z_C)} \\ y_j - y_0 + \Delta y_j &= -S_y f \frac{m_{21}(X_j - X_C) + m_{22}(Y_j - Y_C) + m_{23}(Z_j - Z_C)}{m_{31}(X_j - X_C) + m_{32}(Y_j - Y_C) + m_{33}(Z_j - Z_C)} \end{aligned} \qquad (4.1)$$

and

$$\Delta x_j = (x - x_0)(r^2 d_{r3} + r^4 d_{r5}) + Sh_X y \qquad (4.2)$$

$$\Delta y_j = (y - y_0)(r^2 d_{r3} + r^4 d_{r5}) \qquad (4.3)$$

$$r^2 = (x - x_0)^2 + (y - y_0)^2 \qquad (4.4)$$

where

x_j, y_j = the image coordinates of target j;

X_j, Y_j, Z_j = the object coordinates of ball j;

x_0, y_0 = the coordinates of the principal point in the image coordinate system;

X_C, Y_C, Z_C = the coordinates of the perspective centre in the object coordinate system;

f = the principal distance of the camera, which is constant and not adjusted;

S_x, S_y = the scale factor in x and y directions;

d_{r3}, d_{r5} = the lens distortion parameters for radial distance r;

Sh_X = the shearing factor in the image geometry;

m_{11}, ..., m_{33} = the elements of a 3 x 3 orthogonal rotation matrix M which is a function of 3 rotations of the camera coordinate system, commonly termed ω, φ and κ, about the 3 axes X, Y and Z respectively.

The matrix M is defined as follows :

$$
\begin{aligned}
M &= \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \\
&= \begin{pmatrix} \cos\varphi\cos\kappa & \cos\omega\sin\kappa + \sin\omega\sin\varphi\cos\kappa & \sin\omega\sin\kappa - \cos\omega\sin\varphi\cos\kappa \\ -\cos\varphi\sin\kappa & \cos\omega\cos\kappa - \sin\omega\sin\varphi\sin\kappa & \sin\omega\cos\kappa + \cos\omega\sin\varphi\sin\kappa \\ \sin\varphi & -\sin\omega\cos\varphi & \cos\omega\cos\varphi \end{pmatrix}
\end{aligned}
\tag{4.5}
$$

4.1.2 Camera Configuration

Camera configuration involves the camera arrangement in relation to the sensed objects. A 3D point on an object can be determined from ray intersection of two or more corresponding points in overlapping images. The accuracy of the coordinates of the point is influenced not only by the error of camera orientation and the image coordinates of conjugate points, but also the configuration of cameras. A configuration defect will cause an unstable solution to the object location. Figure 4.1 displays two cases of configuration defects. If the base/depth ratio is too small, a small error in image coordinates will cause a large error $\Delta\delta_1$ in the direction of the depth (a shift from point P to point P'). If the base/depth ratio is too large, however, a similar error of image coordinates will be magnified into a large error $\Delta\delta_2$ in the directions orthogonal to the depth direction. In order to avoid configuration defects and obtain more accurate object coordinates, the errors in the coordinates of 3D points created by improper set-up of cameras should be analysed and limited to a reasonable range.

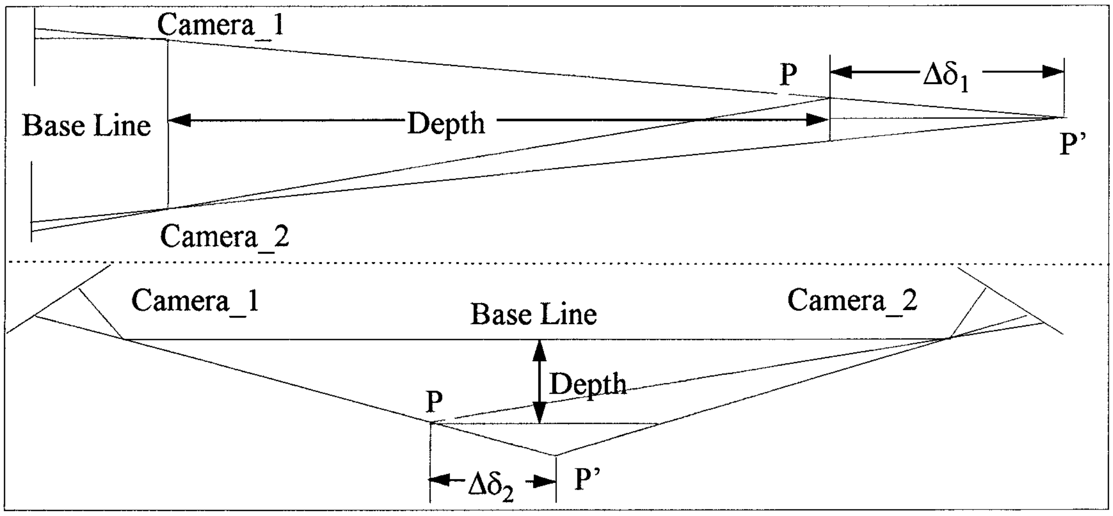


Figure 4.1 : Two cases of configuration defects

In "normal" stereo photography, if the parameters of camera orientation are assumed to be error free, the accuracy can be estimated by applying the well-known formulas (Kraus, 1993).

$$\sigma_x = \sqrt{\left(\frac{x}{b}\frac{d^2}{f^2}\sigma_p\right)^2 + \left(\frac{d}{f}\sigma_x\right)^2}$$

$$\sigma_y = \sqrt{\left(\frac{y}{b}\frac{d^2}{f^2}\sigma_p\right)^2 + \left(\frac{d}{f}\sigma_y\right)^2} \qquad (4.6)$$

$$\sigma_d = \sigma_p \frac{d^2}{fb}$$

with: σ_x, σ_y ... mean accuracy of the restitution of X and Y orthogonal to depth direction;

σ_d ... mean accuracy of the restitution of the depth;

σ_p ... mean accuracy of the image parallaxes (= accuracy of matching);

d ... maximum distance (= depth) between camera and object;

f ... calibrated focal length or principal distance of cameras;

b ... photogrammetric base line;

x, y ... image coordinates with the origin at the principal point.

In close range photogrammetry, sensed objects are required to be maximally covered and captured in all images. Therefore, the optical axis of multiple cameras are generally not parallel, but converge a suitable angle for the application. The convergence angle, which is fundamental to evaluating the imaging geometry, is defined as the angle subtended by the camera optical axes. If the cameras are pointed towards the sensed object centre as shown in figure 4.2, the magnitude of the angle can be related to base/depth ratio in the photogrammetric normal case, as follows:

$$\mathrm{tg}\,\frac{\alpha}{2} = \frac{b}{2d} \qquad (4.7)$$

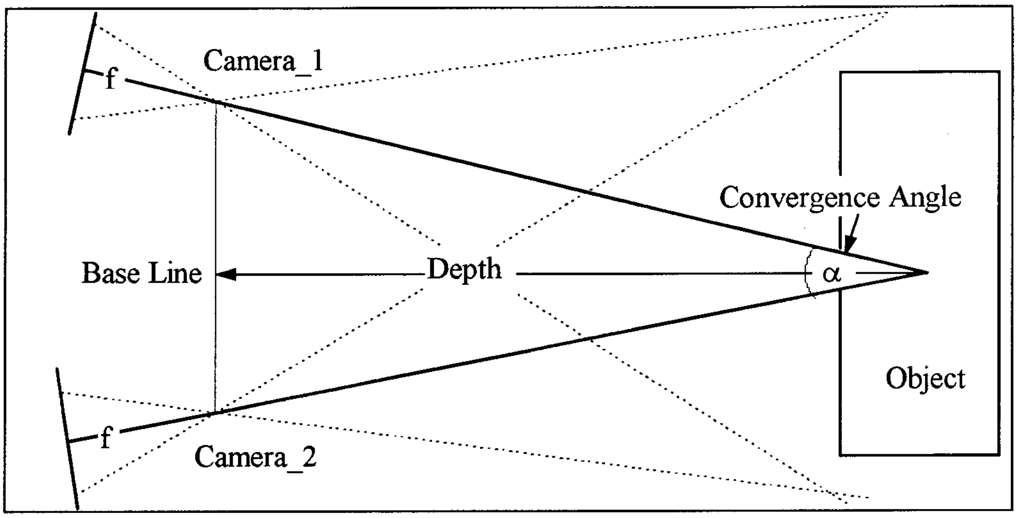


Figure 4.2 : Configuration of two cameras

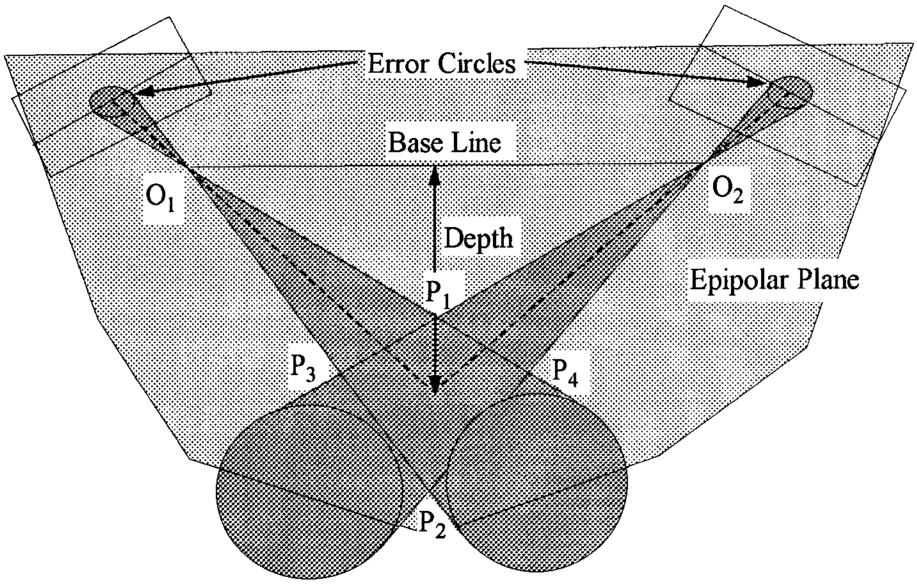


Figure 4.3 : Intersection of two conic solids created by two error circles

The geometry of the camera configuration problem can be viewed in figure 4.3. If the error of image coordinates is the same in X and Y directions, it can be described as a circle around a given image point. An error circle forms a conic solid with its apex at the projection centre. The possible locations of 3D points are the intersection of two conic solids. An epipolar plane can be determined from two projector centres O_1, O_2 and the intersection point P. The error of the point can be divided into two orthogonal parts: one error in the epipolar plane; the other in the normal direction of the plane. If any two bundles of rays within the two cones do not meet together, the ray intersection will be computed by the least squares adjustment. The coordinate of the point in the normal direction of the epipolar plane will be located within the range of $\overline{P_5P_6}$ as shown in figure 4.4. This error is not related to the base/depth ratio. It is only depended on the distance between the point and the camera centres. The errors on the epipolar plane can be described in terms of $\overline{P_1P_2}$ in depth and $\overline{P_3P_4}$ orthogonal to the depth in figure 4.3. The length $\overline{P_1P_2}$ will decrease with the increase of the base/depth ratio, while the length $\overline{P_3P_4}$ will increase in this case. Evaluation of the convergent configuration shows that a convergence angle of 90 is the "best" configuration, because it yields a near isotropic form, ie. precision estimates $\overline{P_1P_2}$ and $\overline{P_3P_4}$ are approximately equal.

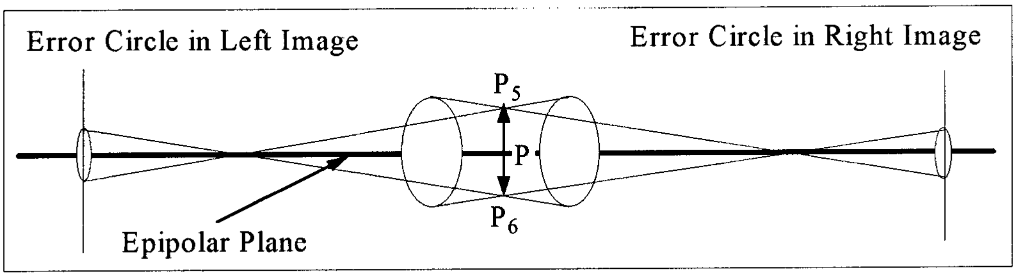


Figure 4.4 : Error of a 3D point in the normal direction of the epipolar plane

Image scale, principal-distance/object-distance ratio, is another factor, which influences object accuracy. It is obvious that the size of a conic section increases with the increase of the distance from the apex to the section. A decrease in image scale number will result in a proportional increase in object point precision (that is, a reduction in the standard error of object coordinates), irrespective of the base/depth ratio. Generally, cameras are expected to be set up as close as possible to the sensed objects with the condition that these objects are to be fully covered by all image formats, so that the highest accuracy can be achieved.

Object point coordinate accuracy is also influenced by camera orientation. If the locations of projector centres and the point P are assumed to be stable, the camera configuration can be changed from the parallel axes (normal case) to the convergent axes, as shown in figure 4.5. In this case, the image will shift from the location in image L_1 and R_1 to L_2 and R_2. As discussed above, an error circle, which surrounds an image point, can form a conic solid in object space. An angle subtended by the conic axis and the camera optical axis is dependent on the coordinates of the image point and the

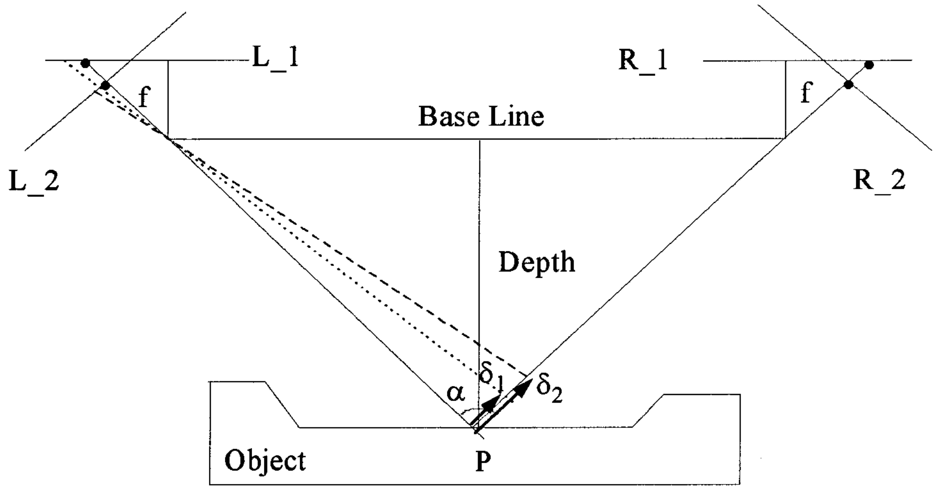


Figure 4.5 : The comparison between normal and convergent configurations

size of an error circle. For circles in an image with the same size but in different locations, the apex angles for those conic solids created by these circles are different, resulting in the different conic shapes and sizes. Secondly, the change of camera orientation will result in the variation of image scale for a particular point. In figure 4.5, the same error circles of the left image in normal and convergent cases will create the error in object space shifting from δ_1 to δ_2. Therefore, object point coordinate precisions for a particular point in normal and convergent configurations are different. The convergent configuration will also result in different shapes of the object in stereo images as shown in figure 4.6, where a rectangle is imaged as a trapezoid.

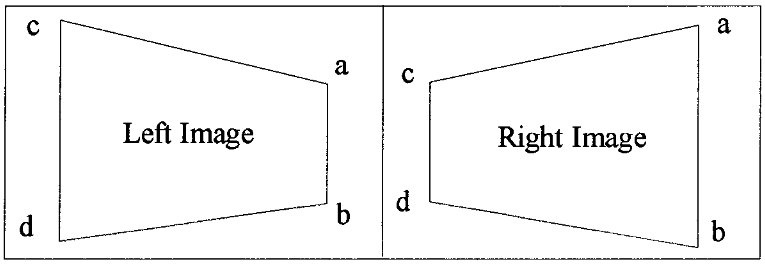
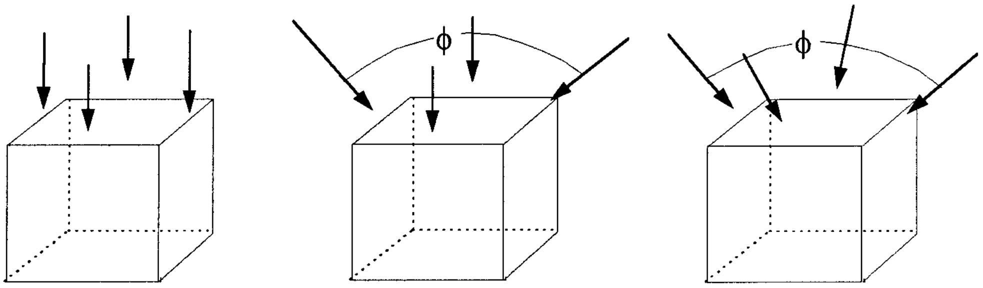


Figure 4.6 : Perspective shape of a rectangle in stereo images

The number of cameras influences the accuracy of the solution and configuration. In general, the accuracy increases with the increase of camera number, because it improves network reliability (Fraser, 1984). Normally, the accuracy of object point coordinates needs to be estimated over the whole object, to determine average accuracy of the 3D object coordinates and the lowest accuracy. If the camera configuration is composed of more than two camera stations, the estimation of the accuracy or precision of object coordinates becomes more difficult. The preferred concept of Limiting Error Propagation based on simulation studies was developed by Brown (1980) and Fraser (1987). The concept is based on the assumption that the camera orientations are regarded to be error free. The only errors considered as influencing the object point coordinate accuracy estimates will be errors in the image coordinates. the accuracy of object point coordinates in simulation studies can be estimated by limiting error propagation for a particular camera configuration. Marshall (1989), for example, applied this technique to evaluate the factors influencing the network configuration problem. His tests were performed on simulation data with target points regularly spaced on a cube. Four cameras are set up in five configurations as shown in figure 4.7. The simulation results are given in table 4.1.



Configuration 1
Base/Distance ratio = 0.7

Configuration 2
Base/Distance ratio = 0.7
$\phi = 120°$

Configuration : 3 = 80°
4 = 120°
5 = 160°

Figure 4.7 : Configurations of four cameras

Configuration	$\bar{\sigma}_p$ mm	$\bar{\sigma}_{xy}$ mm	$\bar{\sigma}_z$ mm	$D/\bar{\sigma}_{xy}$
1	0.18	0.13	0.25	1 in 28,000
2	0.15	0.12	0.20	1 in 33,000
3	0.14	0.10	0.20	1 in 36,000
4	0.11	0.10	0.12	1 in 46,000
5	0.11	0.12	0.09	1 in 46,000

Table 4.1 : The influence of camera configuration on object point precision

The notation used in table 4.1 is as follows:

$\bar{\sigma}_p$ = mean standard error, of all object points, in X, Y and Z;

$\bar{\sigma}_{XY}$ = mean standard error, of all object points, in X and Y;

$\bar{\sigma}_Z$ = mean standard error, of all object points, in Z;

D = the object distance.

Marshall's conclusion (1989) shows that the combination of "normal" and convergent geometry for multiple camera configuration does not influence the planimetric precision to any large degree. Evaluation of the convergent configurations showed that a convergence of 120° was the best configuration of four cameras, since it yields a near isotropic form.

Mason (1994) proposed a conceptual model for the task of configuring a strong imaging geometry in photogrammetric network design. This model is based on the notion of generic networks. A generic network constitutes compiled expertise, describing an ideal configuration of sensor stations that can be employed to provide a strong imaging geometry for a class of network design problem. In addition to improve the understanding of network design, this model performs an important role in the development of an expert system for automated network design for dimensional inspection task.

In practice, camera configuration may also be influenced by space limitations and the restrictions imposed to achieve successful image processing algorithms, such as those presented in this thesis. Secondly, an object may only be partly imaged from some view points. Camera configurations with a large convergent angle may lead to some parts of the area in the images not overlapping, so that objects cannot be completely constructed. Different image scales in the overlapping images is another problem that may be introduced by camera configuration. Parts of an object may have different image scales in different images, specially for large convergent angles. Finally, the intensity of object surfaces in images may vary significantly due to camera set-up for large convergent angles. The magnitude of the intensity on an object surface in an image is dependent on the lighting source, the normal direction of the surface, the camera location and the materials of the object, if all surfaces of an object contain the same colour. The significant shift of cameras will lead to the significant differences of the intensity in images. It is possible that the intensity on one surface is larger than that on a second surface in one image, but smaller than or equal to it in the other image. This may have a significant influence on the success and accuracy of image processing algorithm that have been used in this thesis. In the edge detection process, if one boundary has low contrast of the intensity in an image, its edge cannot be detected, or may be detected with low accuracy.

Therefore, the determination of the camera configuration in this thesis has been controlled by a compromise between the desire to obtain high accuracy of object coordinates and the requirements of the image processing procedure. The set-up of cameras should ensure high contrast of object boundaries in all images, which is also related to lighting source. Secondly, the corresponding components of an object, which are intended to be measured, must appear in at least two images, so that they can be matched and intersected into object space. Cameras should be set as close as possible to the sensed object, with the convergent angles as large as possible but not over 90° in two camera case. For the tests in this thesis, only two cameras were available. Hence, in order to satisfy the above conclusion, a scene object is always captured and located in the centre of an image. The convergent angle is limited in a range between 15° and 90° as shown in the experiments of chapter 5 and 7.

4.1.3 Target Design

In close range photogrammetry, targeting is commonly done with circular retro-reflection bright disks which will be imaged with high contrast, thus facilitating automatic recognition and precise location of the targets. Gruen and Stallmann (1992) created a calibration field, which consists of a black plate and small towers with circular targets as shown in figure 4.8. Heuvel and Kroon (1992) designed a white circle with a concentric ring on a black base. Around this there is another concentric ring in which a ten bit target identification is coded as shown in figure 4.9. They also designed a reseau cross as a target shown in figure 4.10. These targets are all plane targets, and hence due to image projection, the shape of targets in images will be distorted.

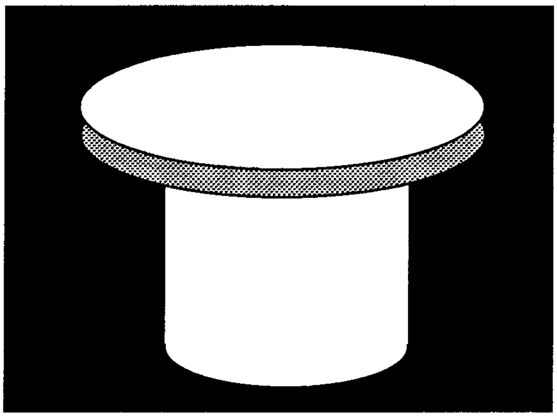


Figure 4.8 A circular retro-reflective target

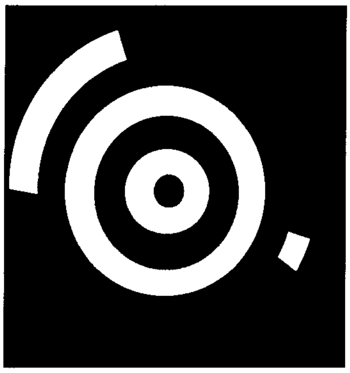


Figure 4.9 Concentric rings

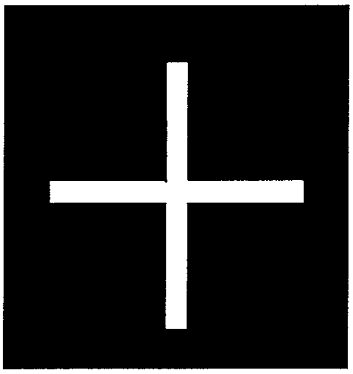


Figure 4.10 A reseau cross

Jansa et. al (1993) used a control field of small balls for camera calibration and orientation. The spherical shape enables robust detection of the target in the image, as the image of a ball must always be recorded as a circle, since it is invariant with different view points. A calibration frame mounted with these targets, the coordinates of whose centres are known, serves as control for this task. For the purpose of camera orientation, however, the size of a calibration frame should be commensurate with the size of imaged objects, while the size of targets in the images should be appropriate for the camera location process. In the thesis, control frames used for camera orientation are similar to that as shown in figure 4.11.

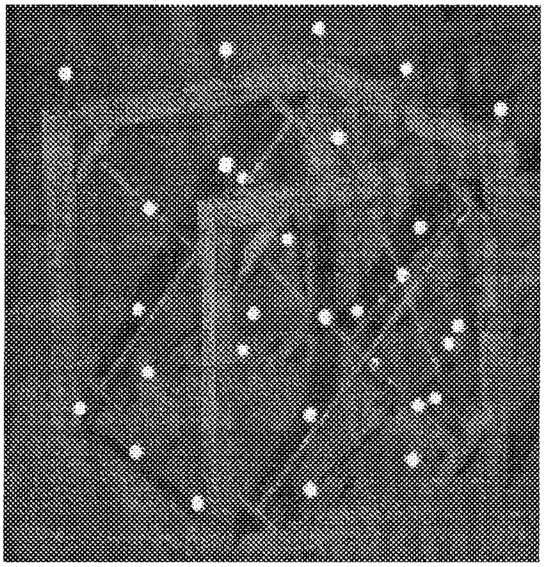


Figure 4.11 An origin image of a calibration frame

4.1.4 Methods for Target Location

Since the accuracy of orientation parameters is fundamentally a function of the directional measurements of target centres in object and image space, precise target location is a precondition for precise 3D restitution. Normally, 3D coordinates of control points can be located very accurately by theodolite (for example, 0.03^{mm}, 0.03^{mm}, 0.04^{mm} in X, Y, Z coordinates), so that the errors in the camera parameters are mainly determined by the accuracy of target measurement of their 2D image coordinates. In recent years, extensive studies have been made on the target location problem in digital photogrammetry. The main algorithms used include: location of the centroid, least squares template matching, and edge extraction and feature fitting (Trinder et. al 1995).

The centroid location approach is based on the calculation of the centre of gravity of the target pixels. The target pixels are determined by thresholding the window, which separates the background from the target. The unweighted mean value of the positions of all pixels of a target yields the target centre. A variation of that method is the weighted centroid centring (Trinder, 1989). The basic principle is the same, but the pixel positions are weighted with the corresponding grey value of the pixels. This algorithm is not so sensitive to incorrect thresholding as the small weight of the darker and uncertain edge pixels do not significantly influence the result, but it is significantly affected by noise or uneven illumination.

Least squares template matching is based on a process of minimisation of the disparities between the grey values of corresponding pixels in a reference image, containing the target itself, and a search image, which is an artificially created window, containing a template of the target (Beyer, 1992). The unknowns derived from this adjustment are parameters of the (usually affine) transformation, which takes into consideration differences in position, scale or rotation between the reference image and search windows. The computation will yield the estimated position of the template of the target on the image, and thus the position of the real target in the image. The advantage of this method is that it is not very sensitive to noise, but will be affected by uneven illumination. The perspective distortions in the target image are compensated for, as part of the computation, and an accuracy which describes the quality of the match can be calculated. One of the disadvantages is that one must know an approximate position within 3 to 5 pixels, to ensure convergence of the solution.

Edge extraction and feature fitting method (Huang and Trinder, 1993) exploits the edge detection operator discussed in chapter 2, to extract the circular or elliptical edges (if the image scales in X and Y direction are different) with subpixel accuracy. The parameters of the ellipse can be obtained from the adjusted parameters of the appropriate second order function. The main advantage of this method is that it is possible to check the internal precision of the location by evaluating the RMS error of the ellipse or circle fitting procedure.

The precision and accuracy of methods of digital target location have been investigated by Jansa et.al (1993) and Trinder et.al (1995). The test indicated that the precision of target location derived from simulated targets based on the methods centroid determination, template matching and edge location, varied from less than 0.01 to 0.04 pixel for 5 or more bit quantisation. Errors occurred in the target location, if illumination on the object is uneven. The exterior orientation procedure of the overlapping images may not detect those errors, leading to shifts and distortions of the restituted 3D objects. Target location based on edge extraction is more or less able to detect the real edge of the target provided that the background of the target is fairly homogeneous and does not disturb the edge extraction process.

In this chapter, an automatic procedure of camera orientation is described, focusing on the location and identification of circular or elliptical targets in the images. To locate the targets in the image, the edge detection method described in chapter 2 is exploited to extract the elliptical target edges with subpixel accuracy. The ellipses are located by fitting an elliptical function to the extracted edge points, from which the centre of the targets can be determined. The process of target location will be presented in section 4.2. The list of the targets in the image must be arranged in their correct order in object space. In section 4.3, a fast search to find the correspondence between 3D balls and their image targets is described, which is based on exploiting the available information in the image, in order to limit the number of tests of possible alternative numbering systems of the targets. The process results in the correct number being attached to each targets. Beside the automatic target identification, a simple manual identification is also discussed in this section. In section 4.4, bundle adjustment is discussed based on the initial approximations of camera parameters being obtained from a DLT computation. Experiment results will be given in section 4.5.

4.2 Location of Targets

The white balls on the black background in the calibration frame shown in figure 4.11 enable high contrast targets in an image to be acquired. The edge detection procedure discussed in chapter 2 is a suitable tool for extracting the target edges which correspond to the physical contours of these targets. The process of target location requires the digital search of the targets in images by their shape, and the determination of their central position automatically from the surrounding edge points. Before giving the details of the processing, the geometric relations between the balls and their images will be discussed in the following subsection.

4.2.1 Projection of Balls on an Image Plane

Though a ball is round in 3D object space, its shape in the image is usually elliptical. One reason for this is that the image scale in X direction is different from that in Y direction in the video image, due to the different pixel sizes along the two axes. This factor, however, does not change the location of a target centre, which is described by the elliptical centre at any scale. Since the target centre in an image is not normally at the principal point of the image, there will be a bias in the elliptical centre away from the principal point. Figure 4.12 displays the geometrical relation of a ball with its elliptical position in the image plane. The major axis of the ellipse is directed towards the principal point in the image. The distance between the true image position of a target centre and an elliptical centre can be defined by :

$$\Delta = \frac{1}{2}(S_1 - S_2) \qquad (4.8)$$

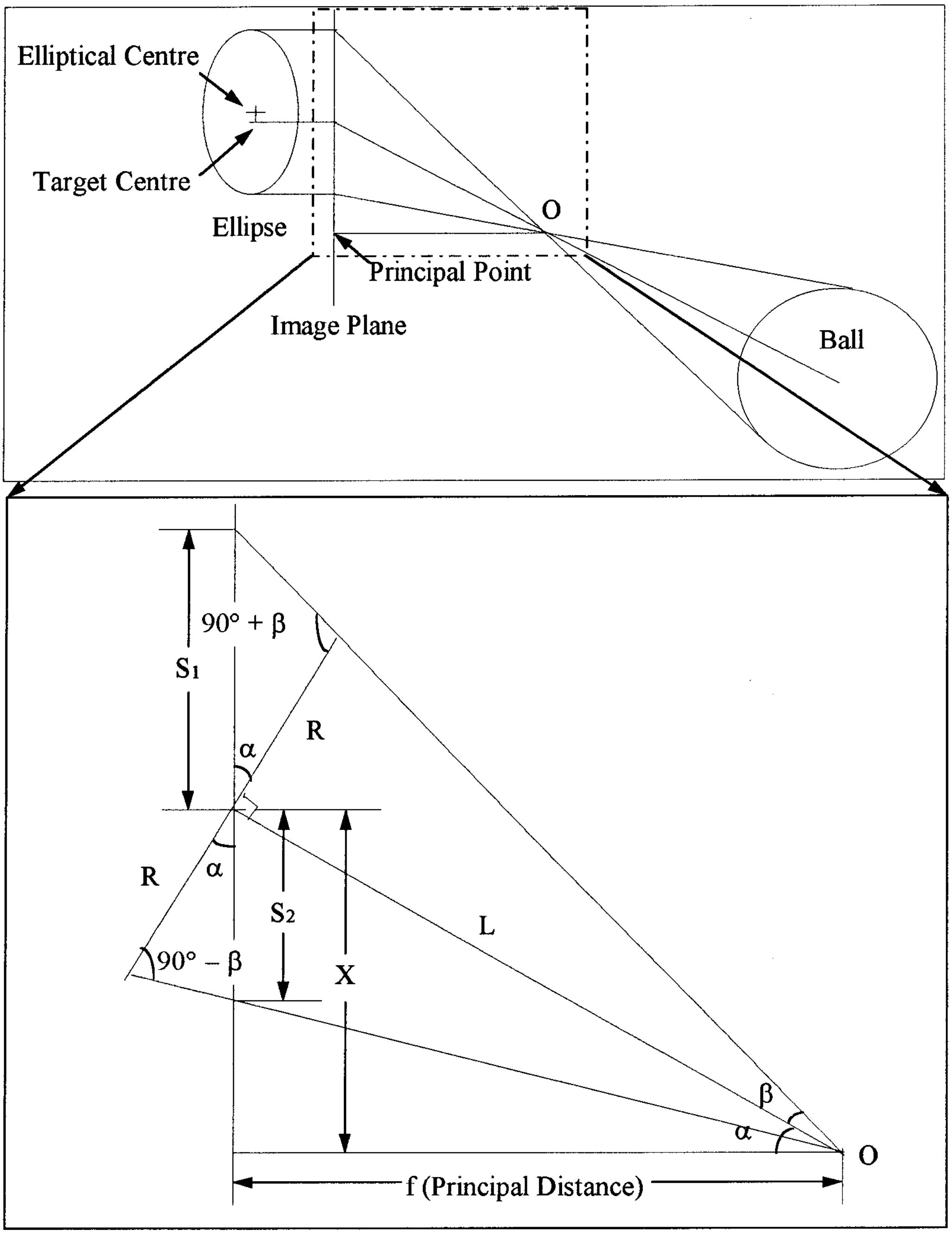


Figure 4.12 : Projection of a ball on an image plane

where S_1 and S_2 are the projected lengths of a ball radius along the major axis of the ellipse in the image, with an assumption that the image scales in X and Y directions are the same. Using a triangle function, the two elements are computed by :

$$S_1 = \frac{R\cos\beta}{\cos(\alpha+\beta)}$$

$$S_2 = \frac{R\cos\beta}{\cos(\alpha-\beta)} \qquad (4.9)$$

Where R is the semi minor axis of the ellipse, or the radius of a circle when the principal point is at the target centre. α is the angle measured at perspective centre between the direction to the principal point and the direction to the target centre. β is the angle measured at the perspective centre to the extreme side of the target. Substituting Eq[4.9] into Eq[4.8], we obtain

$$\Delta = \frac{2\,R \sin\alpha \,\cos\beta \,\sin\beta}{\cos^2\alpha + \cos^2\beta - \sin^2\alpha - \sin^2\beta} \qquad (4.8')$$

The triangle functions can be transformed into image coordinate X and principal distance f, using the geometrical relation in figure 4.12 :

$$\begin{aligned} L &= \sqrt{f^2 + X^2} \\ \cos\alpha &= \frac{f}{L} \\ \sin\alpha &= \frac{X}{L} \qquad\qquad (4.10) \\ \cos\beta &= \frac{L}{\sqrt{R^2 + L^2}} \\ \sin\beta &= \frac{R}{\sqrt{R^2 + L^2}} \end{aligned}$$

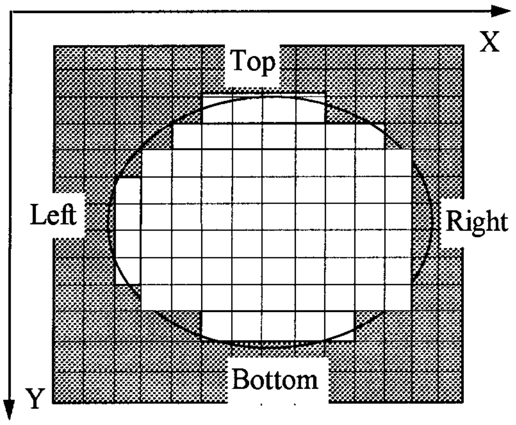
Substitute Eq[4.10] into Eq[4.8'], we obtain

$$\Delta = \frac{R^2\, X (X^2 + f^2)}{f^2\,(f^2 + X^2) - R^2\, X^2} \qquad (4.11)$$

For each camera, the principal distance f is constant (for example, 16 mm). The magnitude of the projected error Δ is therefore dependent on the distance X from target centre to principal centre and the minor axis R of the ellipse in the image. For example, an image is assumed to be 720 by 586 pixels with the principal point in its centre. The image scale is about 130 (pixel / mm), so that the principal distance f is about 2000 pixels. The semi-minor axis of an ellipse R is assumed as 7 pixels. Substitute these data into Eq[4.11], the maximum projected error of the target centre in a 720 x 586 pixel image is Δ = 0.006 pixel. This error is so small, that it can be ignored. If targets are imaged by high resolution cameras, and have significantly larger R (in pixels) than the one in above example, however, this correction must be considered.

4.2.2 Digital Search of Targets and Their Centre Location

To extract the edges of targets, the image of the calibration frame in figure 4.11 is processed by using the procedure of edge detection described in chapter 2. However, this process is relatively time consuming comparing with the subsequent processes. Since the targets in the image only occupy a small percentage of the area, in order to speed up the process, it is necessary to do a quick search for regions occupied by the targets first. The edges are then detected from the section of the images occupied by the target.



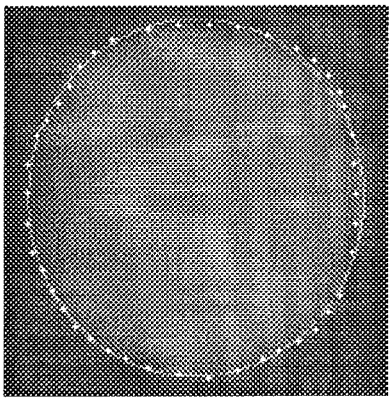


Figure 4.13 : An ellipse in binary image **Figure 4.14 : Edge points fit to an ellipse**

A window is selected on each target by analysing the targets in a binary image, based on a threshold of the average grey value in the image. The target will display a regular circular shape as show in figure 4.13. Since the difference of intensity between the targets and the background is significant, the variation of threshold setting, which may change the target size in binary image, will not influence the final results. For the fast search, the threshold set as the average grey value is effective. The circular shape can be recognised by connectivity analysis from a binary image. The centre of each row of white pixels centre will be displaced a maximum of one or two pixels in X-axis, comparing with neighbour rows. The number of white pixels in each row varies according to their location on the target. These characteristics will allow the shape of targets to be distinguish from other irregular shapes. The target recognition results in the coordinates of the left and right extremities of the target in each row in X-axis, and top and bottom extremities in Y-axis. A small image window is then determined ten or more pixels from the extremities of the target. Using the edge detection method discussed in chapter 2, edge points on a target can be detected as shown in figure 4.14. An elliptical function is then fitted to the shape determined by these edge points.

As discussed above, there are two reasons why the shapes of the image targets are elliptical. The first is due to the different pixel sizes in X and Y directions. This fact will not change central position of targets, and the major and minor axes of the ellipses are parallel to X and Y axes. The second fact is due to the lack of coincidence between the principal point and the target centre which forces the centre away from the principal point. The major axis will be along the direction from the principal point to the target centre. Since the influence is very weak, only variations in image geometry of the target, due to the different pixel sizes, are considered. If the axes of an ellipse is parallel to X and Y image axes, four parameters can determine the structure of the ellipse as following:

$$aX^2 + bY^2 + cX + dY + 1 = 0 \qquad (4.12)$$

The error equation for each edge point (X_i, Y_i) can be written as:

$$V_i = \frac{aX_i^2 + bY_i^2 + cX_i + dY_i + 1}{\sqrt{F'^2_{xi} + F'^2_{yi}}} \qquad (4.13)$$

where

$$\begin{aligned} F'_{xi} &= 2aX_i + c \\ F'_{yi} &= 2bY_i + d \end{aligned} \qquad (4.14)$$

a, b, c and d are the transform parameters, which can be calculated by using a least squares adjustment method. These parameters can be transformed into an equation for an ellipse, which is denoted by :

$$\frac{(X - X_C)^2}{R_x^2} + \frac{(Y - Y_C)^2}{R_y^2} = 1 \qquad (4.15)$$

Where X_C and Y_C are the coordinates of an elliptical centre; R_X and R_Y are the major and minor axis of the ellipse respectively. Let the Eq[4.15] be presented in the form of Eq[4.12] as follows:

$$\frac{1}{F\,R_x^2}X^2 + \frac{1}{F\,R_y^2}Y^2 - \frac{2X_C}{F\,R_x^2}X - \frac{2Y_C}{F\,R_y^2}Y + 1 = 0 \qquad (4.15')$$

where

$$F = \frac{X_c^2}{R_x^2} + \frac{Y_c^2}{R_y^2} - 1 \qquad (4.16)$$

Comparing Eq[4.8'] with Eq[4.5], we obtain :

$$a = \frac{1}{F\,R_x^2};\; b = \frac{1}{F\,R_y^2};\; c = \frac{2X_c}{F\,R_x^2};\; d = \frac{2Y_c}{F\,R_x^2}; \qquad (4.17)$$

The elliptical parameters are presented by the following equations :

$$\begin{aligned} X_c &= -\frac{c}{2a} \\ Y_c &= -\frac{d}{2a} \\ R_x &= \sqrt{\frac{1}{a}(aX_c^2 + bY_c^2 - 1)} \\ R_y &= \sqrt{\frac{1}{b}(aX_c^2 + bY_c^2 - 1)} \end{aligned} \qquad (4.18)$$

The RMS error derived from the adjustment is used to check the interior quality of ellipse fitting. If the value is smaller than an accepted threshold, the shape of the feature is regarded as an ellipse and the result is adopted.

4.2.3 Improvement of Accuracy of Target Location

Balls serving as targets are expected to be imaged as ideal round bright disks on a black background. However, the balls were fixed on the frame by a steel string or a bar, which may occlude part of the ball, resulting in the distortion of part of the edge as shown in figure 4.15. These distorted edge points have a detrimental effect on the assessment of the target location. In order to locate targets precisely, the erroneous sections should be segmented and deleted. The residual in Eq[4.13] represents the distance from an edge point to the assessed ellipse. If the number of distorted edge points are only a small percent of the total number of edge points describing the target, the magnitude of their residuals can reflect the quality of the extracted edge points for the target location process. The larger the residual, the poorer the quality. This differentiation between distorted and correctly located edge points can be determined from their relationship to the RMS error derived from a least squares adjustment of Eq[4.13]. If a residual is more than 3 times the RMS error, the edge point is regarded as erroneous and rejected. The deletion of erroneous edge points can also be done manually. Targets which do not suffer from erroneous edge points can clearly be located more accurately.

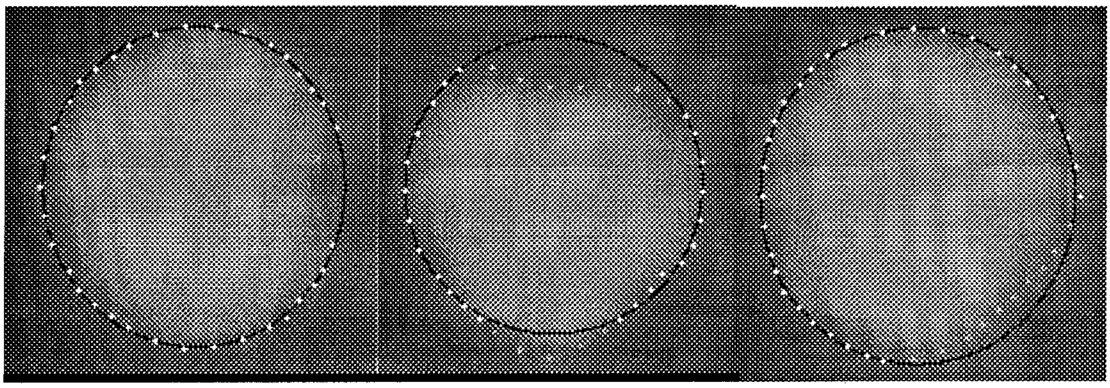


Figure 4.15 : Some erroneous edge points on targets

Table 4.2 displays a comparison between target locations before and after deleting erroneous edge points. For target (a), even though only 3 edge points are deleted, the displacement of target centre is 0.14 pixel, and the precision is less than a quarter of its original value. The improvements are greater for the two targets (b) and (c). The number of deleted edge points must be limited to a certain threshold number (for example, a quarter of the total number of edge points) to achieve satisfactory results. If the number of deleted point is above the threshold, the target is regarded as unreliable and ignored. The deletion of erroneous edge points can also be done manually, since human being can adjust the results more flexibly and reliably.

Targets in Figure 4.15		a	b	c
Total Number of Edge Points		38	35	41
Number of Deleted Points		3	12	10
Displacement (Pixel)	dx	0.14	0.07	0.15
	dy	0.00	0.42	0.18
Precision (Pixel)	Before	0.264	0.384	0.216
	After	0.061	0.101	0.088

Table 4.2 : A comparison between the results before and after deleting erroneous edge points

4.3 Identification of Ellipses

Once the targets have be located, the list of the ellipses in the image must be arranged in the order of the balls in 3D space. A possible solution to this problem is to search all combinations of the order of the balls and determine the correct order by testing its error, based on using the Direct Linear Transformation (DLT). The search speed is largely dependent on how quickly the correct order of listing, amongst the possible alternatives, is found. However, the total number of possible combinations of the target list is often very large, especially when a large number of control points occurs, making the exhaustive search extremely lengthy. In this application, there are 25 control points. If 15 ellipses exist in the image and the DLT is assumed to take 10^{-5} second for each calculation, in the worst situation, it will take 1.3×10^6 years to finish the search.

Some strategies in artificial intelligence, such as hill climbing and least-cost partial path (Winston, 1984), take the form of incomplete rules which tend to indicate the right choice, but without giving a guarantee. The problem addressed here requires finding the correct order, not a better choice, because there are no alternative "correct" choices. Therefore, a reduction in the combination complexity by exploiting the geometrical information in the image is the basic approach adopted for the solution.

4.3.1 Direct Linear Transformation (DLT)

Direct Linear Transformation is the prime method used to determine the correct order of the target numbers. It is a method which establishes a direct relation between image space and object space. Image coordinates are related to object coordinates by means of eleven transform parameters as follows :

$$\begin{aligned} x &= \frac{L_1X + L_2Y + L_3Z + L_4}{L_9X + L_{10}Y + L_{11}Z + 1} \\ y &= \frac{L_5X + L_6Y + L_7Z + L_8}{L_9X + L_{10}Y + L_{11}Z + 1} \end{aligned} \qquad (4.19)$$

Where $L_1, \ldots\ldots, L_{11}$ are the transform parameters. If the control point coordinates are assumed to be without error, the observation equations can be presented by :

$$\begin{aligned} V_x &= \frac{1}{A}(L_1X+L_2Y+L_3Z+L_4-L_9xX-L_{10}xY-L_{11}xZ-x) \\ V_y &= \frac{1}{A}(L_5X+L_6Y+L_7Z+L_8-L_9xX-L_{10}xY-L_{11}xZ-x) \end{aligned} \qquad (4.20)$$

where

$$A = L_9X + L_{10}Y + L_{11}Z + 1 \qquad (4.21)$$

Eq[4.20] is linear but A is not constant. The transform parameters can be computed iteratively using a least squares adjustment. Since there are 11 parameters, the minimum number of targets required for the formation of 12 observation equations is 6. The advantage in the use of the DLT transform is that it does not need provisional values for the adjustment. In this application, the RMS error obtained in the DLT is used to estimate the internal precision of the adjustment. If one or more targets in the image are not correctly assigned to target numbers, it will be clearly reflected in the magnitude of the RMS error.

4.3.2 Number of Targets

Since the total number of possible combinations of target lists will be affected by the total number of targets, for 25 control points, the numbers of possible combinations for the different numbers of targets are shown as:

1 target :		25
2 targets :	$25 \times 24 =$	600
3 targets :	$25 \times 24 \times 23$	$= 13800$
... ...		
7 targets :	$\frac{25\,!}{(25-7)!} =$	2.4×10^{9}
... ...		
25 targets :	$25\,! =$	1.5×10^{25}

The combination complexity rises with an increase in the number of targets. Motivated by this fact, one way of reducing the number of possible choices is to correctly identify a small number of the targets in one image. As the DLT method, which is used to check the correct order, requires a minimum number of six targets, for the purpose of fast calculation and reliability, seven targets are the appropriate number for use in the search process. If the list of the seven targets is in the correct order, other targets can be identified by the camera parameters derived from the seven points. To achieve a good geometric solution, the seven chosen targets should be scattered throughout the image, including the four corners and the image centre.

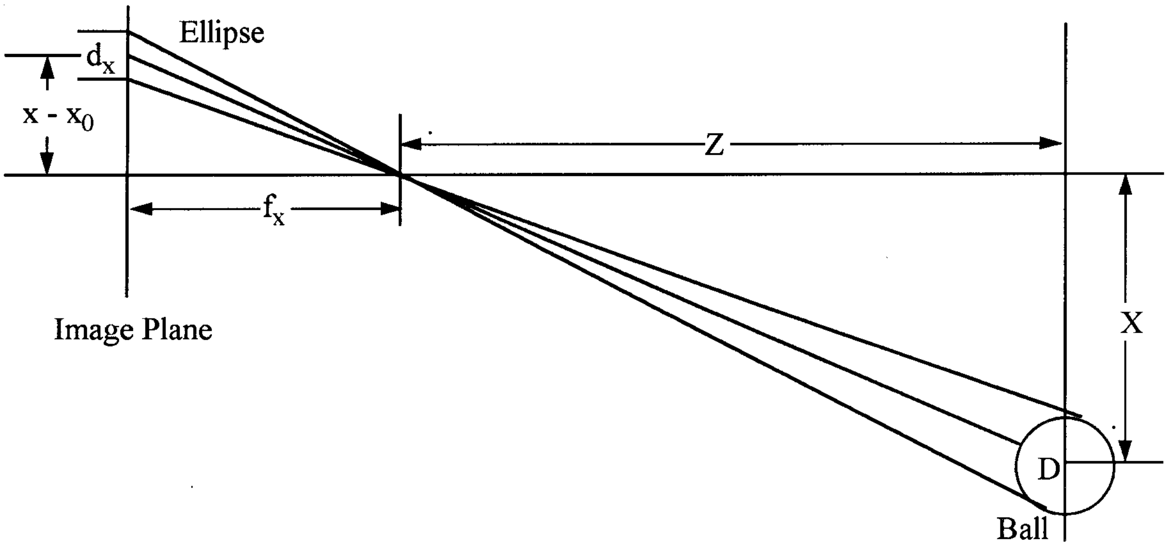


Figure 4.16 : The relation between an ellipse and a ball

4.3.3 Target Size in an Image

Even though the combination complexity is greatly reduced by choosing seven targets for identification, the possible numbers of combinations are still huge. Exploiting the information of the target size in the image, which is a function of the ball's location in object space, is another way of reducing the combination complexity. The centre of a ball in the 3D coordinate system can be approximately located with respect to the perspective centre from the position of the centre of the target in the 2D image coordinate system. This can be derived from its 2D coordinates and the ratio of major axis or minor axis of the ellipse and the ball diameter. Figure 4.16 displays the relation between an ellipse in an image and a ball in object space in the X–Z plane. The approximate coordinates of the centre of a ball can be derived from:

$$X = \frac{D}{d_x}(x - x_0)$$
$$Y = \frac{D}{d_y}(y - y_0) \qquad (4.22)$$
$$Z = \frac{f_x}{d_x}D = \frac{f_y}{d_y}D$$

Where

D = the diameter of a ball;

f_x, f_y = the principal distance of the camera in X and Y direction;

dx, dy = the major axis and minor axis of an ellipse;

x_0, y_0 = the coordinates of the principal point in the image;

x, y = the central coordinates of an ellipse.

The units in the equations are millimetres or pixels, which are transformed from one to the other using the image scale. From the above, an additional set of 3D coordinates of the balls is created, as well as the group of the "known" coordinates of the control points. Since there is no knowledge of translation and rotation between the two coordinate systems, it is impossible to identify the targets directly by comparing the coordinates in the different systems. However, the distance between balls should be invariant in the two coordinate systems. This condition can be used as a distance check in the search process of determining the correct order of the targets. The distances between the 3D coordinates of seven chosen points derived from the assumed set of coordinates can be checked against the distances between the set of known control points in their assumed order. If one difference is above a threshold distance, it will indicate that the assumed point numbering system is incorrect. The threshold limits the difference in distance within which pairs of targets are assessed as equal. Most wrong pairs are filtered out, if the difference is above the threshold. The determination of the threshold is crucial for the speed and reliability of the search process. The accuracy of the 3D coordinates of control points derived from Eq[4.22] is depended on not only the precision of the principal distance and the location of the principal point, but also the accuracy of the central coordinates of the ellipses and their major and minor axis. The threshold must be determined in relation to the estimated errors in distances. If a threshold is set too small, the correct order may be missed, while if the threshold is set too large, the distance checks will require less effort, but erroneous point numbering system will be accepted for the DLT computation. This will result in more lengthy calculations. In order to set the threshold correctly, it is necessary to analysis the error in the coordinates derived from Eq[4.22].

4.3.4 Analysis of Coordinate Accuracy

The error in the coordinates can be derived in terms of the errors in the elements in the function. The differentials of Eq[4.22] in terms of X, Y and Z are defined by :

$$\begin{aligned}
\Delta X &= X(\frac{\Delta D}{D} - \frac{\Delta d_X}{d_X} + \frac{\Delta x}{x - x_{0'}} - \frac{\Delta x_0}{x - x_0}) \\
\Delta Y &= Y(\frac{\Delta D}{D} - \frac{\Delta d_y}{d_y} + \frac{\Delta y}{y - y_{0'}} - \frac{\Delta y_0}{y - y_0}) \qquad (4.23) \\
\Delta Z &= Z(\frac{\Delta D}{D} - \frac{\Delta d_X}{d_X} + \frac{\Delta f_X}{f_X})
\end{aligned}$$

The magnifications of X, Y and Z are written by :

$$\begin{aligned}
M_X &= \sqrt{(\frac{X}{D}M_D)^2 + (\frac{X}{d_X}M_{dX})^2 + (\frac{D}{d_X}M_X)^2 + (\frac{D}{d_X}M_{X0})^2} \\
M_Y &= \sqrt{(\frac{Y}{D}M_D)^2 + (\frac{Y}{d_y}M_{dy})^2 + (\frac{D}{d_y}M_y)^2 + (\frac{D}{d_x}M_{y0})^2} \qquad (4.24) \\
M_Z &= \sqrt{(\frac{Z}{D}M_D)^2 + (\frac{Z}{d_X}M_{dX})^2 + (\frac{f_X}{d_X}M_{FX})^2}
\end{aligned}$$

Since the diameter of a ball can be measured by a calliper accurately, the relative error $\frac{M_D}{D}$ can be as small as 1: 50000 and the influence of this error can be ignored. The accuracy of a target centre can normally be less than 0.1 pixel. If D is approximately 12.8 mm in this case and d_X or d_y is assumed as 10 pixels, the error in $\frac{D}{d_x}M_X$ or $\frac{D}{d_y}M_y$ is approximately 0.128^{mm} which is very small and can be ignored. The magnitudes of the errors M_{fx}, M_{x0} and M_{y0} are determined by the provisional estimation of f_X, x_0 and

y_0 for the cameras. If the error of these estimates is assumed to 2 pixels, $\frac{f_x}{d_x}M_{fx}$ is equal to 3.2^{mm} ($f_x = 16^{mm}$), and $\frac{D}{d_x}M_{x0}$ or $\frac{D}{d_y}M_{y0}$ is about 2.56^{mm}. The errors M_{dx} and M_{dy} are the errors in the elliptical major and minor axis created by the edge detection and the elliptical fitting. If their errors are assumed to be 0.2 pixel and the 3D coordinates of a target are (0.2^m, 0.2^m, 2.5^m), ($\frac{X}{d_x}M_{dx}$, $\frac{Y}{d_y}M_{dy}$, $\frac{Z}{d_x}M_{dx}$) can be of the order of (4^{mm}, 4^{mm}, 50^{mm}). This result indicates that the location of targets derived by Eq[4.22] is very sensitive to the error in the elliptical major and minor axis, which is the largest contribution to the errors. Since the coordinate Z is normally larger than X and Y, for the approximate estimation of a threshold, only the error $\frac{Z}{d_x}M_{dx}$ is considered. The distance between two points is written by :

$$\text{dist} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2} \qquad (4.25)$$

The approximate RMS error of the distance is therefore represented by :

$$M_{dist} = \sqrt{2}\,\frac{Z_{ave}}{d_x}\,M_{dx} \qquad (4.26)$$

Where Z_{ave} is the average Z coordinate of control points. Using the data assumed above, the RMS error is about 7^{cm}, which is a reference value for setting the threshold.

4.3.5 Process of Target Identification

The seven targets are arranged in alphabetical order as a, b, c, d, e, f and g. The numbers of 25 control points are attached to the targets in the numeric order. As a distance is derived from two points, a distance check can be carried out when the numbers of more than one candidate are assumed. The first action in the search process assumes the target a to be the first control point, and b to be the second. The distance S_{ab} between the targets a and b is then compared with the distance S_{12} between the control point 1 and the control point 2. If the difference is above a threshold, the assumption is wrong and the target b is assumed to be the third control point. If the difference is smaller than the threshold, the following target c is determined accordingly. The distance check will be performed between the three targets. In this way, the search process is carried out until seven targets are checked as shown in figure 4.17. The number of checked distances rises with the increases of the number of involved targets. If all the differences of the distances are within the threshold, the set of seven points selected will then be checked by the DLT. The DLT check is a reliable step, which determines whether the set of points is in the right order. If the RMS error derived from the DLT is above an acceptable value, the process will return to check in the distance between the targets.

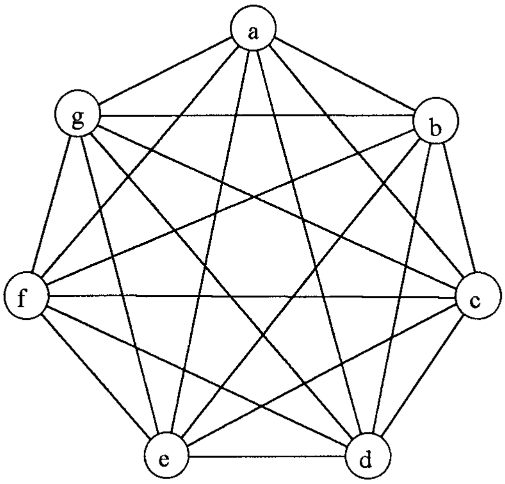


Figure 4.17 : Distances among seven targets

The threshold for distance check is an important value which determines whether the search process selects a new candidate or assumes the current candidate is incorrect. Since the RMS error derived from Eq[4.26] roughly gives the error of an average distance, the true error for each distance is unknown, so that the determination of a reasonable threshold is not easy. However, it is interested to note that if the threshold is set at different levels, such as 10^{cm}, 15^{cm}, ..., 40^{cm}, the number of times the DLT computations are carried out are 58, 2314, ..., 3.3×10^5 respectively. The search process for the first threshold level takes a very small amount of time compared to that required for the next larger threshold level. This makes it possible to set multi-level thresholds. The search process is firstly performed on the threshold which is set to be 10^{cm}. If no correct order of targets can be found, the search process goes to the next level of the threshold which is set to be 15^{cm}. Since the time used on the first and second level of the threshold is approximately the same as the time used on only the second level, the search process using multi-level threshold does not waste much time. If the error in the difference of distances among seven targets is small, however, the search can be processed quickly with a small value of the threshold. In this way, the search can be achieved efficiently.

4.3.6 Manual Identification of Targets

The balls as control points are expected to be round and of an identical size. The errors in roundness, resulting in the different shapes of the balls, are caused by the different view points, while the error in size directly influences the 3D position of targets in the use of Eq[4.22]. These errors hinder the automatic process of target identification. As a supplement, manual identification of targets is performed on images by giving identification numbers directly to the targets. Several targets are identified first, while other targets are identified by the camera parameters derived from these points. The identification of targets results in the numbers being attached to the corresponding targets as showed in figure 4.18.

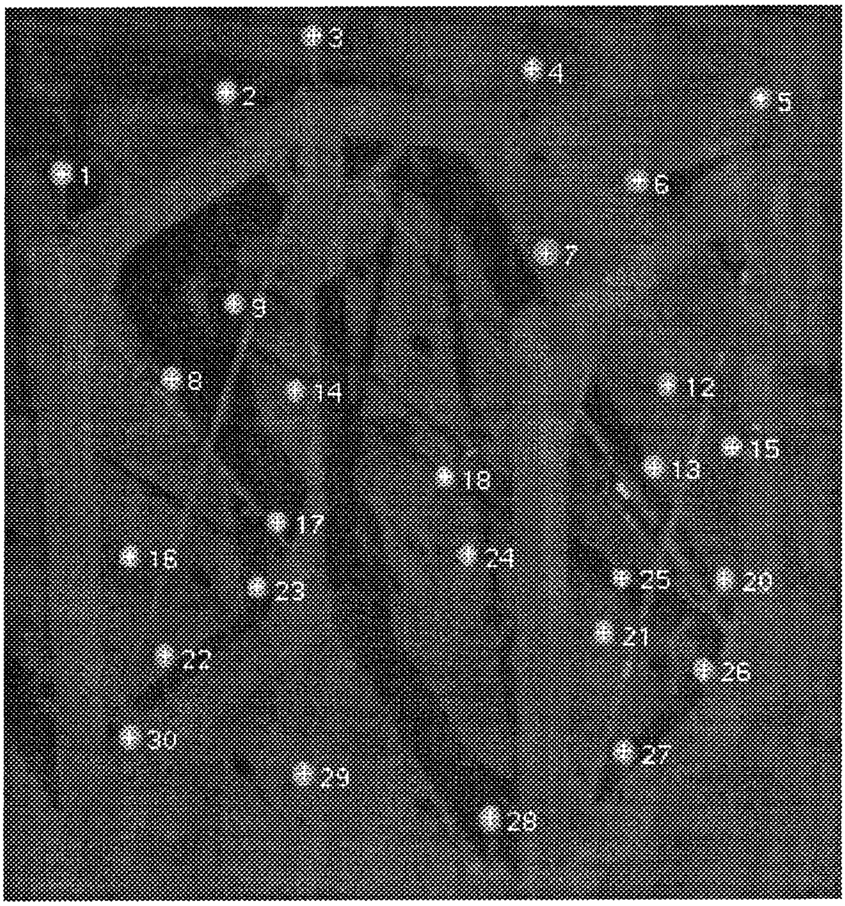


Figure 4.18 : Identification of the targets

4.4 Bundle Adjustment

Once the coordinates of target centres in image space are related to corresponding coordinates of ball centres in object space, the precise parameters of camera orientation can be obtained by bundle adjustment. Bundle adjustment is based on the collinearity equations, where the perspective geometry is modelled in a straight forward way. The observation equations can be obtained by differentiating the unknown elements in Eq[4.1], while normal equations can be formed through conventional methods. The solution of the normal equations provides the parameters of camera orientation.

Before using least squares methods, however, the initial approximate values are needed for the estimation of inner and outer orientation elements. It is not easy to estimate these approximate values in the different situations, such as when the viewpoint or the orientation of the calibration frame is changed. Furthermore, an automatic procedure requires initial approximations to be given automatically. The DLT method provides adjustment values of the parameters without the requirement for provisional values. Since these results are approximately related to the results of bundle adjustment, it is possible to obtain the initial approximations for the bundle adjustment from the DLT results.

The basic idea (Karara, 1989) of the solution of initial approximations is to transform the collinearity equations Eq[4.1] into the form of DLT transform equations. The initial approximations are then calculated from the DLT transform parameters. Since lens distortion and shearing factor in bundle adjustment are very small and need no provisional values, they are ignored in relating the collinearity equation to the DLT as follows :

$$\begin{aligned} x &= x_0 - f_x \frac{m_{11}X + m_{12}Y + m_{13}Z + \gamma_1}{m_{31}X + m_{32}Y + m_{33}Z + \gamma_3} \\ y &= y_0 - f_y \frac{m_{21}X + m_{22}Y + m_{23}Z + \gamma_2}{m_{31}X + m_{32}Y + m_{33}Z + \gamma_3} \end{aligned} \qquad (4.27)$$

where

$$\begin{aligned} \gamma_1 &= -(m_{11}X_C + m_{12}Y_C + m_{13}Z_C) \\ \gamma_2 &= -(m_{21}X_C + m_{22}Y_C + m_{23}Z_C) \\ \gamma_3 &= -(m_{31}X_C + m_{32}Y_C + m_{33}Z_C) \end{aligned} \qquad (4.28)$$

and

$$\begin{aligned} f_X &= S_X f \\ f_y &= S_y f \end{aligned} \qquad (4.29)$$

The above equations are transformed into the form of DLT equations :

$$\begin{aligned} x &= \frac{(m_{31}x_0 - m_{11}f_X)X + (m_{32}x_0 - m_{12}f_x)Y + (m_{33}x_0 - m_{13}f_X)Z + (\gamma_3 x_0 - \gamma_1 f_X)}{m_{31}X + m_{32}Y + m_{33}Z + \gamma_3} \\ y &= \frac{(m_{31}y_0 - m_{21}f_y)X + (m_{32}y_0 - m_{22}f_y)Y + (m_{33}y_0 - m_{23}f_y)Z + (\gamma_3 y_0 - \gamma_2 f_X)}{m_{31}X + m_{32}Y + m_{33}Z + \gamma_3} \end{aligned} \qquad (4.27')$$

Comparing the elements of Eq[4.27'] with the elements in DLT equations, we obtain :

$$\begin{pmatrix} L_1 & L_5 & L_9 \\ L_2 & L_6 & L_{10} \\ L_3 & L_7 & L_{11} \\ L_4 & L_8 & 1 \end{pmatrix} = \frac{1}{\gamma_3} \begin{pmatrix} m_{31}x_0 - m_{11}f_X & m_{31}y_0 - m_{21}f_y & m_{31} \\ m_{32}x_0 - m_{12}f_X & m_{32}y_0 - m_{22}f_y & m_{32} \\ m_{33}x_0 - m_{13}f_X & m_{33}y_0 - m_{23}f_y & m_{33} \\ \gamma_3 x_0 - \gamma_1 f_X & \gamma_3 y_0 - \gamma_1 f_y & \gamma_3 \end{pmatrix} \qquad (4.30)$$

The initial approximates of interior orientation can then be calculated, referring to Karara (1989), as following:

$$\begin{aligned} x_0 &= \gamma_3^2 (L_1 L_9 + L_2 L_{10} + L_3 L_{11}) \\ y_0 &= \gamma_3^2 (L_5 L_9 + L_6 L_{10} + L_7 L_{11}) \\ f_X^2 &= \gamma_3^2 (L_1^2 + L_2^2 + L_3^2) - x_0^2 \\ f_y^2 &= \gamma_3^2 (L_5^2 + L_6^2 + L_7^2) - y_0^2 \end{aligned} \qquad (4.31)$$

where

$$\gamma_3^2 = \frac{1}{L_9^2 + L_{10}^2 + L_{11}^2} \qquad (4.32)$$

The elements of exterior orientation can be written as:

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} = \begin{pmatrix} \dfrac{L_9 x_0 - L_1}{f_X} & \dfrac{L_{10} x_0 - L_2}{f_X} & \dfrac{L_{11} x_0 - L_3}{f_X} \\ \dfrac{L_9 y_0 - L_5}{f_y} & \dfrac{L_{10} y_0 - L_6}{f_y} & \dfrac{L_{11} y_0 - L_6}{f_y} \\ L_9 & L_{10} & L_{11} \end{pmatrix} \qquad (4.33)$$

and

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C \end{pmatrix} = -\begin{pmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{pmatrix}^{-1} \begin{pmatrix} L_4 \\ L_8 \\ 1 \end{pmatrix} \tag{4.34}$$

4.5 Experiments

The procedure of camera orientation has been tested by placing a camera in different viewpoints. The number of images is unlimited. Two or more cameras can image the control frame at the same time, while their orientations can be determined simultaneously. Tests have been performed on the image in figure 4.18. Table 4.3 shows the RMS error of the adjusted image coordinates. One result is adjusted from 27 target centres derived by all surrounded edge points, while the other is adjusted from the target centres calculated without erroneous edge points. The improvement of the precision can be seen in the table, where the RMS error goes down from 0.147 pixel to 0.053 pixel of X coordinates, and from 0.119 to 0.073 pixel of Y coordinates. By comparing the two adjusted results, the RMS error of the coordinate shift could be as large as 0.086 pixel in X axis and 0.047 pixel in Y axis. It should be noted that the error of target location is created not only by the edge detection process, but also lighting condition, when ball shaped targets are used. If the light source is positioned at some distance from the camera, the brightest spot in the image is shifted out of the target centre. This results in the measured target centre biasing towards the brightest spot (Jansa et.al 1993). In order to generate a symmetrical intensity distribution of targets, lighting source should be mounted as close as possible to the camera.

	All Edge Points		Without Erroneous Points		Difference Adjustment Results	
	x	y	x	y	x	y
RMS	0.147	0.119	0.053	0.073	0.086	0.047

Table 4.3 : Accuracy of bundle adjustment

The results of bundle adjustment create a data file, which includes the correct parameters of interior and exterior orientation and their precision as shown in table 4.4. The first column gives the name of orientation parameters, whose values are displayed in the second column. The third column gives the precision of the estimated parameters. This file can be input into other softwares written for object measurement, to establish the relation between 2D image space and 3D object space.

* _BEGIN_* Project :	Camera_Orientation	
*_ORIENTATION_DATA_*		
FOCAL_LENGTH (cm)	16	(not adjusted)
X_SCALE (Pixel/cm)	1196.194213	1.94716
Y_SCALE (Pixel/cm)	1312.304632	2.12818
X_PRINCIPAL_POINT (Pixel)	361.956245	3.33347
Y_PRINCIPAL_POINT (Pixel)	259.888035	4.29740
X_SHEARING	-0.000469	0.00030
R3_LENS_DISTRORTION	0.000391	0.00038
R5_LENS_DISTRORTION	0.000035	0.00004
X_PROJECTION_CENTRE (cm)	-26.150962	0.09870
Y_PROJECTION_CENTRE (cm)	-221.466939	0.43850
Z_PROJECTION_CENTRE (cm)	88.080683	0.10467
OMEHA_ROTATION	77.083632	0.11777
PHI_ROTATION	-103.082169	0.10150
KAPPA_ROTATION	-2.970492	0.02054

Table 4.4 : Result of bundle adjustment

4.6 Summary

An automatic and semi-automatic procedure of camera orientation has been developed for a digital close range photogrammetric system. In this application, small bright balls mounted on a calibration frame serve as control points, since their shape in an image is invariant to the camera position. To recognise the targets, white spots with regular circular shapes are searched throughout the image. The edges are then detected using the algorithm described in chapter 2 from the small image which just covers the area of a target. The central location of targets and their size can be determined from these edge points. The list of the targets in the image must be arranged in the correct order of the balls in the 3D world. A fast search is described which is based on exploiting the available metric information about the targets in order to limit the number of possible alternative orders of the target numbers. In this way, the search can be achieved efficiently. The identification of targets results in the correct numbers being attached to the corresponding targets. The precise camera parameters are calculated by a bundle adjustment, while the initial approximations of the parameters are transformed from DLT computation.

Chapter 5 Reconstruction of 3D Objects

5.1 Introduction

Reconstruction of 3D objects from two or more images is an important step for object recognition and inspection. Digital photogrammetry enables the determination of 3D geometric data of objects, based on multiple overlapping images. In a digital photogrammetry system, corresponding features in overlapping images can be found automatically by a matching procedure. Given known transformation parameters between image and object space, locations of these image features can be calculated in object space. Generally, image matching technique may be classified into area-based matching and feature-based matching.

Area-based matching uses the radiometric values of individual pixels in image windows centred on the conjugate point in different images. From a point in one image, its conjugate in the other is found on the bases of a certain similarity measure, defined by the gray values within the image window. This method is widely applied in practice and achieves good results in cases where the surface relief is relatively flat and the images of the surface has a rich image texture (Wong, 1986). Conversely, the reconstruction of discontinuous surfaces and surfaces with few texture images by means of this method is problematic.

Feature-based match is a process of finding corresponding features on the different images. Features in an image may be interest points, edges and geometric items, such as straight lines, regular curves or surface patches. Image features are usually only a small percentage of the whole content of the image, but they convey the essential information about surface discontinuities. Generally, matching based on geometric features is more reliable than that based on interest points and edges, because there exist some geometric constraints on the matching process. Surface patches created by the line segmentation process described in chapter 3 clearly present geometric features. The elements, such as corner points, in the surface patches are related each other by virtue of their relative position. These relations provides a clue to help finding the correspondences of features.

In addition, the condition of epipolar geometry, based on the camera orientation, is also used to limit the search for matched candidates on an image from two dimensions to one dimension. However, some boundaries of an object may not be extracted in the edge detection process due to poor illumination, so that one detected surface patch may contain two surfaces of the object. Therefore, direct matching on surface patches in stereo images will fail in some cases. Straight lines and regular curves are primary elements of surface patches, as well as containing geometric information. Using such elements as matching features is more appropriate. Once the correspondences of image features are determined, the three dimensional positions of those features are easily computed from the pairs of features in both images.

In this chapter, an automatic procedure of 3D object construction is presented in terms of epipolar geometry and feature geometry. In section 5.2, epipolar geometry is discussed, which gives the relation between image coordinates in overlapping stereo images. In section 5.3, straight lines and regular curves are matched in terms of the epipolar constraint and feature geometry. Section 5.4 describes in detail the intersection of the geometric features. Section 5.5 shows several experimental results, while a summary is given in section 5.6.

5.2 Epipolar Geometry

The results of bundle adjustment in chapter 4 give the relation between image and object space. This relation can be transformed into the relation between the coordinates of points in a pair of images which lie on an epipolar line, called the epipolar geometry, which is used to limit the search for conjugated points in both images to the line. The functional epipolar geometry can be found in Wong (1986) and Tan et. al. (1994). However, since the epipolar geometry is an important constraint for stereo matching, the computation equations of epipolar lines developed for this thesis will be described in the following subsection. The epipolar geometry is presented by a 3×3 matrix, which directly relates the coordinates of points in images on an epipolar line.

In order to show the mathematical relationship between the image points in stereo images, it is necessary to analyse the parameters of the camera orientation, which comprise the elements of interior orientation, additional parameters, such as lens distortion and image shear, together with the external orientation. The process of object reconstruction in this research is mainly dependent on the extracted edges. Since the locations of edge points along the object boundaries have been corrected for errors caused by lens distortion and image shear after edge chaining, the effects of lens distortion and image shear will not be considered in this chapter.

5.2.1 Epipolar Line in Stereo Images

The epipolar geometry relates homologous points in a pair of images along the corresponding epipolar line. The epipolar constraint can be derived from the parameters of absolute orientation as discussed in chapter 4. If two images are named as left and right, the collinearity equation (4.1) in the left image can be written in a matrix form as :

$$\begin{pmatrix} \dfrac{x_L - x_{L0}}{f_{Lx}} \\ \dfrac{y_L - y_{L0}}{f_{Ly}} \\ -1 \end{pmatrix} = \frac{1}{\lambda_L} M_L \begin{pmatrix} X - X_{Lc} \\ Y - Y_{Lc} \\ Z - Z_{Lc} \end{pmatrix} \qquad (5.1)$$

The subscript L refers to the coordinates or parameters of the left image. The parameter λ_L is a scale factor. The vector α from the projection centre of the left camera to one ground point (X, Y, Z) can be transformed into the following formula :

$$\alpha = \begin{pmatrix} X - X_{Lc} \\ Y - Y_{Lc} \\ Z - Z_{Lc} \end{pmatrix} = \lambda_L M_L^T H_L \begin{pmatrix} x_L \\ y_L \\ 1 \end{pmatrix} \qquad (5.2)$$

where

$$H_L = \begin{pmatrix} \frac{1}{f_{Lx}} & 0 & -\frac{x_{L0}}{f_{Lx}} \\ 0 & \frac{1}{f_{Ly}} & -\frac{y_{L0}}{f_{Ly}} \\ 0 & 0 & -1 \end{pmatrix} \tag{5.3}$$

We can also obtain the vector β from the projection centre of right camera to the ground point (X, Y, Z).

$$\beta = \begin{pmatrix} X - X_{Rc} \\ Y - Y_{Rc} \\ Z - Z_{Rc} \end{pmatrix} = \lambda_R M_R^T H_R \begin{pmatrix} x_R \\ y_R \\ 1 \end{pmatrix} \tag{5.4}$$

where

$$H_R = \begin{pmatrix} \frac{1}{f_{Rx}} & 0 & -\frac{x_{R0}}{f_{Rx}} \\ 0 & \frac{1}{f_{Ry}} & -\frac{y_{R0}}{f_{Ry}} \\ 0 & 0 & -1 \end{pmatrix} \tag{5.5}$$

If we assume γ to be a vector of the baseline from projection centre of left camera to that of right camera, we have :

$$\gamma = \begin{pmatrix} \Delta X_C \\ \Delta Y_C \\ \Delta Z_C \end{pmatrix} = \begin{pmatrix} X_{Rc} - X_{Lc} \\ Y_{Rc} - Y_{Lc} \\ Z_{Rc} - Z_{Lc} \end{pmatrix} \tag{5.6}$$

According to the coplanarity condition, three vectors should satisfy that $\alpha \bullet \gamma \times \beta = 0$. Therefore the coplanarity equation can be written as

$$\begin{pmatrix} X - X_{Lc} \\ Y - Y_{Lc} \\ Z - Z_{Lc} \end{pmatrix} \bullet \begin{pmatrix} \Delta X_C \\ \Delta Y_C \\ \Delta Z_C \end{pmatrix} \times \begin{pmatrix} X - X_{Rc} \\ Y - Y_{Rc} \\ Z - Z_{Rc} \end{pmatrix} = 0 \tag{5.7}$$

In order to write these equations for the use of matrix operations, the cross product is expressed as follows :

$$\begin{pmatrix}\Delta X_C\\ \Delta Y_C\\ \Delta Z_C\end{pmatrix} \times \begin{pmatrix}X-X_{Rc}\\ Y-Y_{Rc}\\ Z-Z_{Rc}\end{pmatrix} = \begin{pmatrix}0 & -\Delta Z_C & \Delta Y_C\\ \Delta Z_C & 0 & -\Delta X_C\\ -\Delta Y_C & \Delta X_C & 0\end{pmatrix}\begin{pmatrix}X-X_{Rc}\\ Y-Y_{Rc}\\ Z-Z_{Rc}\end{pmatrix} = B\begin{pmatrix}X-X_{Rc}\\ Y-Y_{Rc}\\ Z-Z_{Rc}\end{pmatrix} \qquad (5.8)$$

where B is an asymmetric or skewsymmetric matrix. Hence, the equation of coplanarity becomes:

$$\begin{pmatrix}x_L\\ y_L\\ 1\end{pmatrix}^T H_L^T M_L B M_R^T H_R \begin{pmatrix}x_R\\ y_R\\ 1\end{pmatrix} = \begin{pmatrix}x_L\\ y_L\\ 1\end{pmatrix}^T E \begin{pmatrix}x_R\\ y_R\\ 1\end{pmatrix} = 0 \qquad (5.9)$$

where

$$E = H_L^T M_L B M_R^T H_R = \begin{pmatrix}e_{11} & e_{12} & e_{13}\\ e_{21} & e_{22} & e_{23}\\ e_{31} & e_{32} & e_{33}\end{pmatrix} \qquad (5.10)$$

The equation (5.9) relates the coordinates of the left image with the coordinates of the right image on an epipolar line. Given a point in one image, an epipolar line function can be formed in the other image. This relation is invariant to the transformation and rotation of coordinate system determined by control points. If the relative positions of two cameras are fixed, the calculation of the matrix E with the different placement of a control point frame differs only by a ratio factor. Dividing all elements in the matrix E by the last element e_{33}, the standardised matrix E_d will be invariant to the coordinate system, which can be written as follows :

$$E_d = \begin{pmatrix}e'_{11} & e'_{12} & e'_{13}\\ e'_{21} & e'_{22} & e'_{23}\\ e'_{31} & e'_{32} & 1\end{pmatrix} \qquad (5.11)$$

It is useful to check the stability of camera position during the working process. Since the orientation of the control frame may not be placed exactly in the same position each time, the camera parameters may not be constant. However, the standardised matrix E_d and the distance between the projection centres of two cameras will be invariant. If those values are significantly different, it indicates that the relative positions of the two cameras have been changed between two sequential checks.

5.2.2 Epipolar Line in One Image

The equation (5.9) determines an epipolar line function in one image from a point in the other image. The epipolar line function can also be determined directly by a point in the same image. Any two points on the same epipolar line in one image can determine the corresponding epipolar line function in the other image as shown in figure 5.1. Using the condition that the same epipolar lines have the same orientation, the coordinates of the two points can be directly related. The equation of an epipolar line in the right image, determined by point 1 and point 2 in left image, can be written as follows :

$$\begin{pmatrix} e_{11}x_{L1} + e_{21}y_{L1} + e_{31} \\ e_{12}x_{L1} + e_{22}y_{L1} + e_{32} \\ e_{13}x_{L1} + e_{23}y_{L1} + e_{33} \end{pmatrix} \begin{pmatrix} x_R \\ y_R \\ 1 \end{pmatrix} = 0 \qquad (5.12)$$

$$\begin{pmatrix} e_{11}x_{L2} + e_{21}y_{L2} + e_{31} \\ e_{12}x_{L2} + e_{22}y_{L2} + e_{32} \\ e_{13}x_{L2} + e_{23}y_{L2} + e_{33} \end{pmatrix} \begin{pmatrix} x_R \\ y_R \\ 1 \end{pmatrix} = 0 \qquad (5.13)$$

Since the two points are assumed to be on the same epipolar line, the orientation of the epipolar line on the other image, determined by the two points respectively, must be the same. Hence, we have :

$$tg\theta = -\frac{b}{a} = -\frac{e_{12}x_{L1} + e_{22}y_{L1} + e_{32}}{e_{11}x_{L1} + e_{21}y_{L1} + e_{31}} = -\frac{e_{12}x_{L2} + e_{22}y_{L2} + e_{32}}{e_{11}x_{L2} + e_{21}y_{L2} + e_{31}} \qquad (5.14)$$

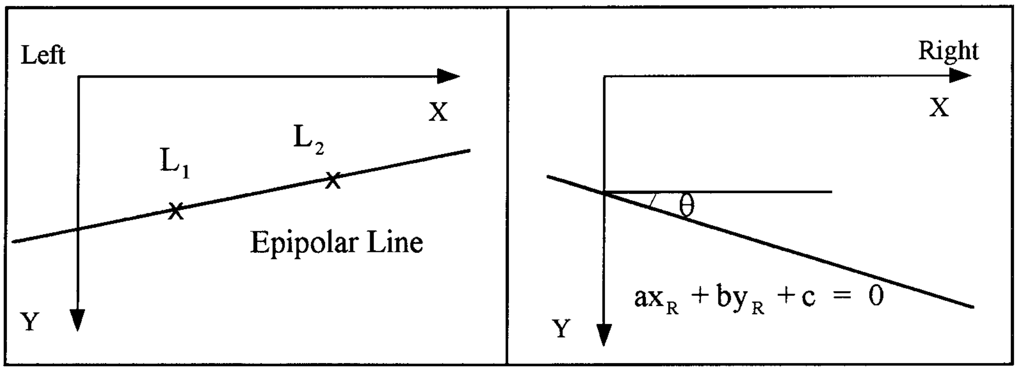


Figure 5.1 : Two points on the epipolar line in the left image determine the same epipolar line in the right image

Where a and b are the parameters of a straight line. If (x_L,y_L) is defined for all points (x_{L2},y_{L2}) on the epipolar line, transforming this equation into the form of Eq[5.9], we obtain:

$$\begin{pmatrix} x_{L1} \\ y_{L1} \\ 1 \end{pmatrix}^T E_L \begin{pmatrix} x_L \\ y_L \\ 1 \end{pmatrix} = \begin{pmatrix} x_{L1} \\ y_{L1} \\ 1 \end{pmatrix}^T \begin{pmatrix} 0 & -e_{L3} & e_{L2} \\ e_{L3} & 0 & -e_{L1} \\ -e_{L2} & e_{L1} & 0 \end{pmatrix} \begin{pmatrix} x_L \\ y_L \\ 1 \end{pmatrix} = 0 \qquad (5.15)$$

where

$$\begin{aligned} e_{L1} &= e_{21}e_{32} - e_{22}e_{31} \\ e_{L2} &= e_{12}e_{31} - e_{11}e_{32} \\ e_{L3} &= e_{11}e_{22} - e_{21}e_{12} \end{aligned} \qquad (5.16)$$

Similarly, we can obtain epipolar line equation in the right image :

$$\begin{pmatrix} x_{R1} \\ y_{R1} \\ 1 \end{pmatrix}^T E_R \begin{pmatrix} x_R \\ y_R \\ 1 \end{pmatrix} = \begin{pmatrix} x_{R1} \\ y_{R1} \\ 1 \end{pmatrix}^T \begin{pmatrix} 0 & -e_{R3} & e_{R2} \\ e_{R3} & 0 & -e_{R1} \\ -e_{R2} & e_{R1} & 0 \end{pmatrix} \begin{pmatrix} x_R \\ y_R \\ 1 \end{pmatrix} = 0 \qquad (5.17)$$

where

$$\begin{aligned} e_{R1} &= e_{12}e_{23} - e_{22}e_{13} \\ e_{R2} &= e_{21}e_{13} - e_{11}e_{23} \\ e_{R3} &= e_{11}e_{22} - e_{12}e_{21} \end{aligned} \qquad (5.18)$$

Similar to the equation (5.9), the relation of image coordinates on an epipolar line in equations (5.15) and (5.17) is invariant to the transformation and rotation of the coordinate system. Another characteristic is that the product of the vector (e_{L1}, e_{L2}, e_{L3}) with the matrixes E_L is zero, which is useful for matching geometric features in the following process. The equations (5.9), (5.15) and (5.17) provide the epipolar geometry on a pair of images. Given a point in an image, the function of an epipolar line can easily be determined by the 3 × 3 matrix in one of these equations.

5.3 Stereo Matching

Stereo matching is a method of matching geometric information in stereo images. Identical features in different images can be found automatically by a matching procedure. There are two main features derived from the line segmentation in chapter 3: straight lines and regular curves. Stereo matching of these features is based on the geometry of these lines and the epipolar geometry described above.

5.3.1 Matching of Straight Lines

The correspondence of a pair of straight lines in two images is determined in terms of their terminal points, which should satisfy the epipolar constraint. Matching of straight lines is more reliable than matching of single points, because they include the line geometry. Since errors exist in the parameters of straight lines, two corresponding points in the pair of straight lines may not lie exactly on the same epipolar line. To determine the correspondence, the distance from a candidate point to an epipolar line must be checked. If two points (x_{L1},y_{L1}) and (x_{R1},y_{R1}) are the ends of two straight lines in stereo images respectively, the epipolar line created by point (x_{L1},y_{L1}) can be presented as follows :

$$a_R x_R + b_R y_R + c_R = 0 \tag{5.19}$$

where

$$\begin{aligned} a_R &= e_{11}x_{L1} + e_{21}y_{L1} + e_{31} \\ b_R &= e_{12}x_{L1} + e_{22}y_{L1} + e_{32} \\ c_R &= e_{13}x_{L1} + e_{23}y_{L1} + e_{33} \end{aligned} \tag{5.20}$$

The distance from the point (x_{R1}, y_{R1}) to the epipolar line can be calculated by Eq[3.16] in the following equation :

$$\text{dist}_{R1} = \frac{a_R x_{R1} + b_R y_{R1} + c_R}{\sqrt{a_R^2 + b_R^2}} \tag{5.21}$$

Similarly, the distance from the point (x_{L1}, y_{L1}) to the epipolar line created by the point (x_{R2},y_{R2}) can be as following :

$$\text{dist}_{L1} = \frac{a_L x_{L1} + b_L y_{L1} + c_L}{\sqrt{a_L^2 + b_L^2}} \tag{5.22}$$

where

$$\begin{aligned} a_L &= e_{11}x_{R1} + e_{12}y_{R1} + e_{13} \\ b_L &= e_{21}x_{R1} + e_{22}y_{R1} + e_{23} \\ c_L &= e_{31}x_{R1} + e_{32}y_{R1} + e_{33} \end{aligned} \tag{5.23}$$

The distances dist_{L1} and dist_{R1} will most likely not be the same, but they should be very similar. If both distances are smaller than a threshold, such as 0.5 pixel, those two points are regarded as on the same epipolar line. The problem of ambiguous elements occurring in repetitive patterns should be solved by reference to the point order in a polyline which surrounds the surface patch, or by checking the elevation of the ends of the straight line in object space.

However, there are two kinds of straight lines for which correspondence cannot be found. Since a pair of images is taken from different view points, some straight lines occur in one image, but not in the other due to occlusion. Secondly, extracted straight lines are not always presented as the boundaries of objects, such as the contour of a cylinder as shown in figure 5.2. The straight lines 1 and 2 corresponding to the object contours have different locations on an object on different view points, so that these straight lines cannot be matched.

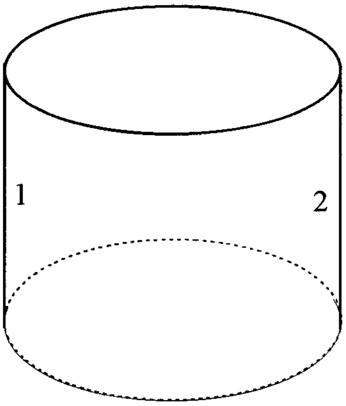


Figure 5.2 : A cylinder

5.3.2 Matching of Regular Curves

To find the correspondence of regular curves, it is necessary to consider their size and location. Since an open curve in one image may not correspond to that in the other, the elements being matched should be closed curves, which are normally ellipses, expressed as second order polynomials. It should be noted that the centres of conjugated ellipses in stereo images may not lie along the same epipolar line, because an elliptical centre does not correspond to the centre of its object in 3D space due to projectivity as shown in figure 5.3. If a line, passed by P_1, P_2 and P_c, is not parallel to the image plane, the image scales at their corresponding points P_1', P_2' and P_c' are different. Therefore, the point P_c located in the middle of P_1 and P_2 may not indicate its corresponding P_c' located exactly in the middle of P_1' and P_2'. In addition, the size and shape of an ellipse in the image varies for different view points, due to different image scales. However, one important fact is that two epipolar lines which are tangential to a given ellipse in one image, are also tangential to a matched ellipse in the other. This condition includes the basic requirement for the size and location of matched ellipses.

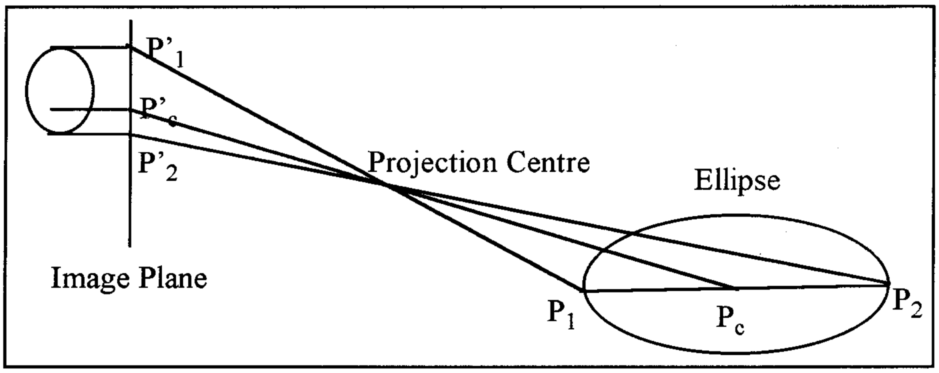


Figure 5.3 : Projection of an ellipse on image plane

Let us consider an ellipse, whose equation in an image is written as a second order polynomial:

$$ax^2 + bxy + cy^2 + ex + dy + 1 = 0 \qquad (5.24)$$

Its matrix form can be presented as:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}^T \begin{pmatrix} 2a & b & d \\ b & 2c & e \\ d & e & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \qquad (5.25)$$

Where a, b, c, d and e are the parameters of a second order polynomial. The tangent line at a point (x_t, y_t) can be presented as

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}^T \begin{pmatrix} 2a & b & d \\ b & 2c & e \\ d & e & 2 \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} = 0 \qquad (5.25')$$

Assuming the ellipse is in the left image, the equation (5.15) of the epipolar line passing the point (x_{Lt}, y_{Lt}), can be written as :

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}^T \begin{pmatrix} 0 & -e_{L3} & e_{L2} \\ e_{L3} & 0 & -e_{L1} \\ -e_{L2} & e_{L1} & 0 \end{pmatrix} \begin{pmatrix} x_{Lt} \\ y_{Lt} \\ 1 \end{pmatrix} = 0 \qquad (5.15')$$

If the tangent line is assumed to be the same as the epipolar line, we get :

$$\begin{pmatrix} 2a & b & d \\ b & 2c & e \\ d & e & 2 \end{pmatrix} \begin{pmatrix} x_{Lt} \\ y_{Lt} \\ 1 \end{pmatrix} = k \begin{pmatrix} 0 & -e_{L3} & e_{L2} \\ e_{L3} & 0 & -e_{L1} \\ -e_{L2} & e_{L1} & 0 \end{pmatrix} \begin{pmatrix} x_{Lt} \\ y_{Lt} \\ 1 \end{pmatrix} \qquad (5.26)$$

Where k is a scale factor. Multiplying the vector $E_l = (e_{L1}, e_{L2}, e_{L3})$ with the matrix E_L, we have :

$$\begin{pmatrix} e_{L1} \\ e_{L2} \\ e_{L3} \end{pmatrix}^T \begin{pmatrix} 0 & -e_{L3} & e_{L2} \\ e_{L3} & 0 & -e_{L1} \\ -e_{L2} & e_{L1} & 0 \end{pmatrix} = 0 \qquad (5.27)$$

Therefore, we obtain the following function of a line which joins two tangent points on the ellipse as shown in figure 5.4.

$$\begin{pmatrix} e_{L1} \\ e_{L2} \\ e_{L3} \end{pmatrix}^T \begin{pmatrix} 2a & b & d \\ b & 2c & e \\ d & e & 2 \end{pmatrix} \begin{pmatrix} x_{Lt} \\ y_{Lt} \\ 1 \end{pmatrix} = 0 \qquad (5.28)$$

The equations (5.25) and (5.28) determine two tangent points on the ellipse. The matched elements of an ellipses can be presented as a straight line with two tangent points as its ends. In a similar manner to matching straight lines, the ellipses being matched should satisfy the requirement that their corresponding tangent points lie on the same epipolar line.

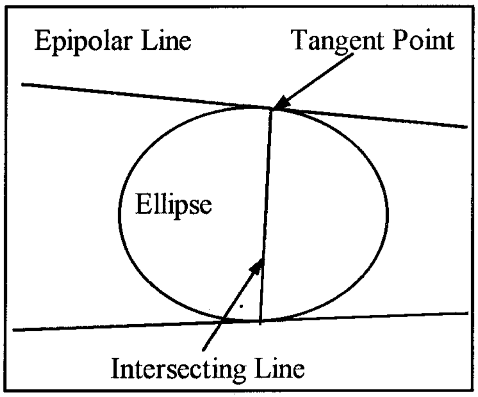


Figure 5.4 : Epipolar lines are tangent to an ellipse.

5.4 Intersection of Geometric Features

The matched image features, such as straight lines and ellipses, have to be transformed into object space by photogrammetric methods. The location of 3D straight lines are determined by ray intersection of their end points using camera orientation. A 3D straight line is simply presented by the 3D coordinates of its two terminals.

The determination of 3D ellipses, however, is a non-trivial task. If homologous edge points on stereo images are known, the 3D points along an ellipse border can be calculated by ray intersection. The 3D ellipses can then be formed by fitting these points to the function of an ellipse. Since edges detected in different images are different, edge points in one image usually have no corresponding edge points in the other image. Furthermore, it is not accurate to intersect the edge points within an area where the tangent direction of the ellipse is nearly parallel to an epipolar line. In order to recover a 3D ellipse precisely, it is necessary to develop other approaches to the problem.

Andresen and Yu (1994) developed a method using a parametric form of an element in space optimising the parameters with respect to a suitable residual. A residual for any ray, starting on the ellipse boundaries in the images, to the ellipse in space can be derived, if suitable initial values of the parameters are determined. Minimising the residuals by a least squares method yields the parameters. In the method, deriving suitable initial values for the parameters of the spatial ellipse is important for the convergence of the iteration. The rough approximation of the spatial ellipse is determined by intersecting almost homologous edge points. These data then are used to derive an approximate plane of the ellipse. Intersecting the rays of each edge point with this approximate plane yields a new set of data, which determines approximate parameters of an ellipse.

In this section, a more simple method will be described. The main principle of the method is that a spatial ellipse is regarded as an intersection of an object plane with a conic surface, whose apex is at the projection centre, as shown in figure 5.5. An ellipse in an image can also be referred to as the intersection of an image plane with the conic surface. The parameters of an ellipse on two different planes are transformed, if the relations of the two planes are known. In fact, a matched pair of ellipses in stereo images involves two conic surfaces, one for each image. A spatial ellipse can be determined by the intersection of these two surfaces. However, the calculation of the intersection of two conic surfaces is complicated, specially when errors exist in the parameters of two conic surfaces. Therefore, the determination of an object plane is preferred. A method for the calculation of an ellipse in space is developed, which contains two aspects: one is to determine an object plane; the other is to establish the relation between image and object planes. The derivation is based on the coordinate system of the AutoCAD graphic package.

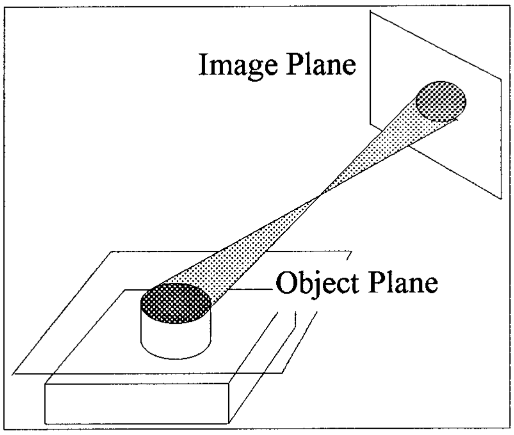


Figure 5.5 : Intersection of a plane with a conic surface

5.4.1 Determination of Object Plane

In order to determine an object plane, which is intersected by both conic surfaces, two epipolar lines are created at the same distance on each side of the centre of the ellipse, which intersect the ellipse in 4 points in the stereo images as show in figure 5.6. These points are then transformed into object space to determine a plane. The equation of a plane can be presented as :

$$\gamma_1 X + \gamma_2 Y + \gamma_3 Z + \rho = 0 \qquad (5.29)$$

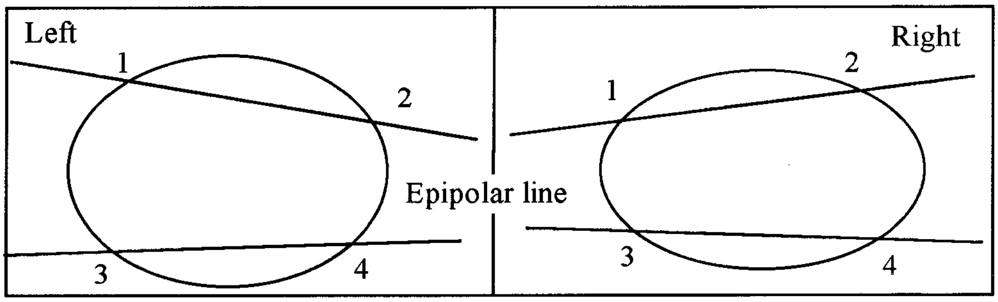


Figure 5.6 : Epipolar lines intersect an ellipse in stereo images

and

$$\gamma_1^2 + \gamma_2^2 + \gamma_3^2 = 1 \qquad (5.30)$$

The vector $\gamma = (\gamma_1, \gamma_2, \gamma_3)$ is along the normal direction of the plane. In order to generate the arbitrary but consistent local coordinate system on the plane, an arbitrary axis algorithm is used, which is referred to the AutoCAD® system. Given a unit-lengthvector to be used as the Z axis of a coordinate system, the arbitrary axis algorithm generate a corresponding X axis for the coordinate system. The Y axis follows by application of the right-hand rule. The method is to examine the given Z axis and see if it is close to the positive or negative World Z axis. If it is, cross the World Y axis with the given Z axis to arrive at the arbitrary X axis. If not, cross the World Z axis with the given Z axis to arrive at the arbitrary X axis. The boundary at which the decision is made was chosen to be both inexpensive to calculate and completely portable across machines. This is achieved by having a sort of "square" polar cap, the bounds of which is 1/64, which is precisely specifiable in 6 decimal fraction digits and in 6 binary fraction bits. The origin (X_0, Y_0, Z_0) of the system can be determined from the average coordinates of four points, and the vector γ is defined as the Z axis. The vector α in X direction is defined as $(\gamma_3, 0, -\gamma_1)$. if the absolute values of $\gamma_1$and γ_2 are smaller than 1/64. Otherwise, the vector α is defined as $(-\gamma_2, \gamma_1, 0)$. The vector β in the Y direction is determined by the vectors γ and α, where $\beta = \gamma \times \alpha$. The cross product can be expressed in matrix form :

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix} \times \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} 0 & -\gamma_3 & \gamma_2 \\ \gamma_3 & 0 & -\gamma_1 \\ -\gamma_2 & \gamma_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} \qquad (5.31)$$

By standardising the vectors α and β with lengths equal to one, an orthogonal rotation matrix (α, β, γ) is obtained.

5.4.2 Relation between Image and Object Planes

Any point on the plane can be presented as (x_p, y_p) with zero elevation in the normal direction (Z axis) of the plane. This point can be transformed into object space by using the orthogonal rotation matrix and the origin (X_0, Y_0, Z_0) on the plane.

$$\begin{pmatrix} X_p \\ Y_p \\ Z_p \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} + \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \beta_1 & X_0 \\ \alpha_2 & \beta_2 & Y_0 \\ \alpha_3 & \beta_3 & Z_0 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} \tag{5.32}$$

Combining the equation (5.32) with the collinearity equation (5.2), we get the equation which relates the coordinates in image plane with the coordinates in the object plane as follows :

$$\begin{pmatrix} x_L \\ y_L \\ 1 \end{pmatrix} = \frac{1}{\lambda_L} H_L^{-1} M_L \begin{pmatrix} X-X_{Lc} \\ Y-Y_{Lc} \\ Z-Z_{Lc} \end{pmatrix} = \frac{1}{\lambda_L} H_L^{-1} M_L \begin{pmatrix} \alpha_1 & \beta_1 & X_0-X_{Lc} \\ \alpha_2 & \beta_2 & Y_0-Y_{Lc} \\ \alpha_3 & \beta_3 & Z_0-Z_{Lc} \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}$$

$$= \frac{1}{\lambda_L} H_L^{-1} M_L G_L \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} \tag{5.33}$$

Using this relationship, a regular curve on the image plane can be transformed on to the object plane :

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}^T \begin{pmatrix} 2a' & b' & d' \\ b' & 2c' & e' \\ d' & e' & 2f' \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}^T Q_L \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = 0 \tag{5.34}$$

where

$$Q_L = G_L^T M_L^T (H_L^{-1})^T \begin{pmatrix} 2a & b & d \\ b & 2c & e \\ d & e & 2 \end{pmatrix} H_L^{-1} M_L G_L \tag{5.35}$$

In a similar manner to the computation of a regular curve derived from left image plane, an equation of the regular curve on the object plane can also be transformed from the right image plane. The two regular curves are conic sections derived by the intersection of the conic surfaces, whose apexes are at projection centres, and an object plane. A 3D regular curve can be determined by the parameters of two curves on the plane and the location of the plane. The accuracy of the recovery of 3D regular curves is affected by the error of the feature location in the image, camera orientation, an intersected plane, any existing outliers in the data of edge points and the size of the curved feature in image space. Figure 5.7 displays the intersection of planes with conic surfaces formed from the left and right images. There is one correct curve on the plane_1 in the object, while two curves occur on the plane_2 which is not determined correctly.

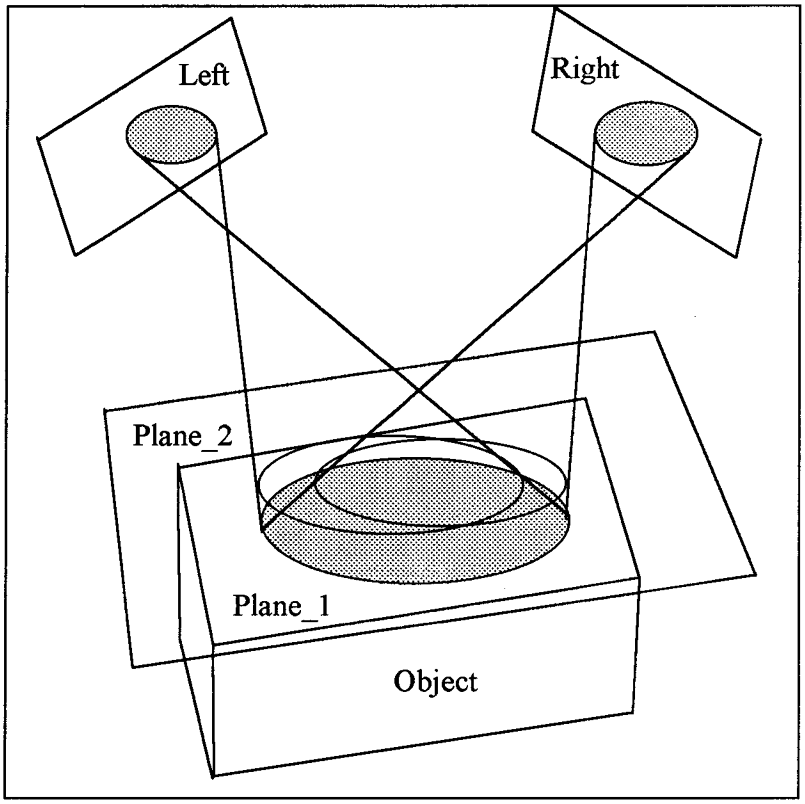


Figure 5.7 : Intersection of two conic surfaces with plane_1 and plane_2

5.5 Experiments

Experiments have been conducted on an industrial component by following the procedure discussed above. Two cameras were set up horizontally and positioned about 0.5 m in front of the object range of interest with a convergent angle about 50°. A control frame with white balls fixed on it was first imaged, in order to obtain the parameters of camera orientation as described in chapter 4. The control frame was then replaced by an object, with dimension of about $150 \times 150 \times 70$ mm^{3}, comprising 400 by 400 pixels in the image. Figure 5.8 illustrates a pair of stereo images which are processed by edge detection and line segmentation (figure 5.9). The two main polylines are matched and transformed into object space, thus determining two surfaces (figure 5.10).

As discussed in chapter 2, the distribution of gradient magnitude on edges in the images is asymmetric, and a smoothing filter will cause shifts of the edge location. In table 5.1, the results were calculated for three sigma values $\sigma = 0.0, 0.5, 1.5$ for image smoothing, and compared with the measurements of the object using a vernier calliper. It was obvious that the smoothing factor influences three dimension of objects, with the best result when $\sigma = 0.5$ or below. The experiment also provides the distances between the circle centres and line 4, 6, 10 and 11, which are difficult to measure by callipers. The test indicates that the dimensions of the object can be determined to an accuracy expressed as an RMS of the difference between photogrammetric and direct measurement of 0.13 mm or 0.37 pixel. The diameters of the circles in the object were determined to an accuracy of 0.07 mm or 0.1 pixel.

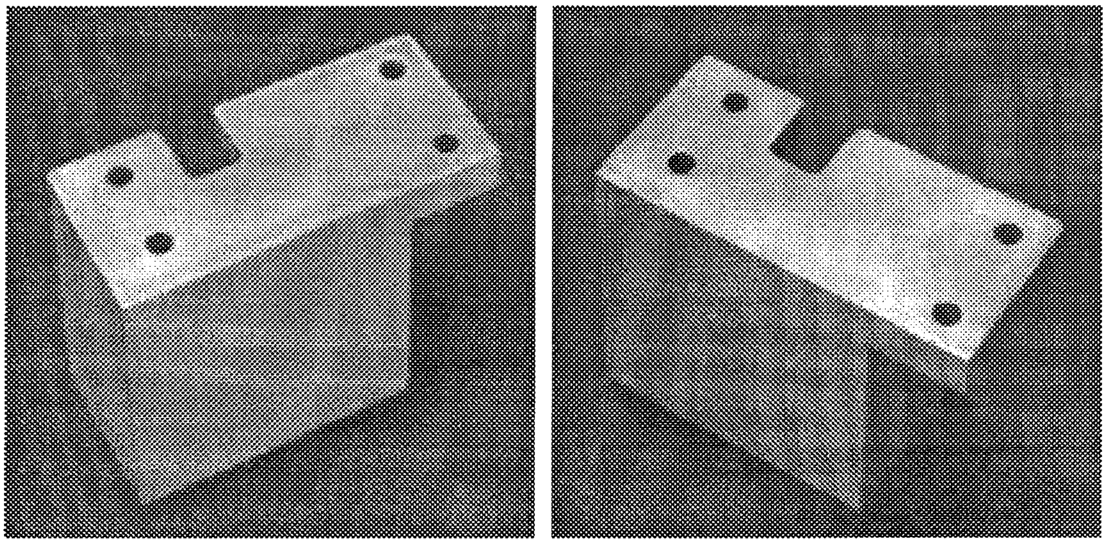


Figure 5.8 : Stereo images of an object

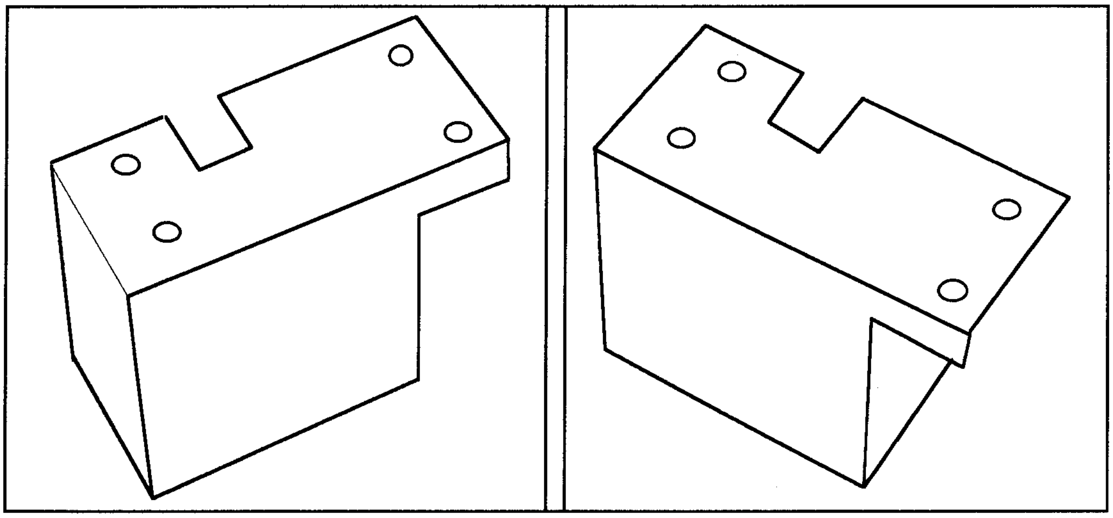
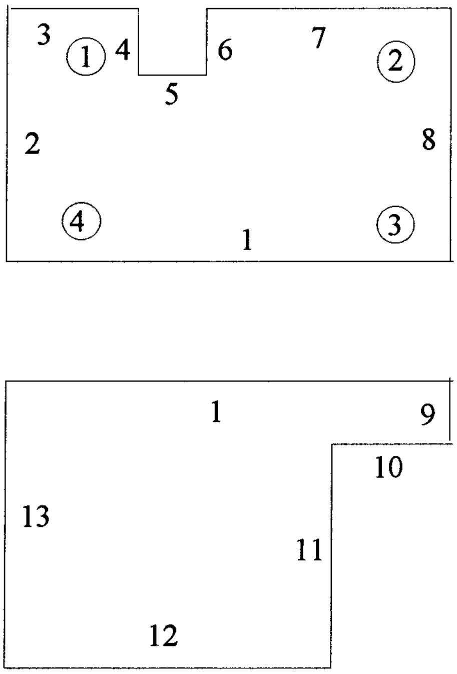


Figure 5.9 : Line segmentation of edges



		Measure (mm)	Difference	(mm)	
			0.0	0.5	1.5
Lines	1	103.10	–0.13	–0.08	–0.09
	2	47.90	0.04	–0.13	–0.14
	3	29.98	0.05	0.06	0.14
	5	14.84	0.13	–0.12	–0.08
	7	57.00	0.04	0.05	0.11
	8	47.42	0.20	–0.20	–0.21
	9	10.82	0.02	0.02	0.16
	12	77.22	0.20	–0.20	–0.17
	13	66.58	–0.22	–0.17	–0.30
	RMS		0.137	0.130	0.168
Circles	1	6.90	–0.02	–0.02	–0.02
	2	6.90	–0.12	–0.12	–0.13
	3	6.90	–0.07	–0.08	–0.15
	4	6.90	–0.04	–0.03	–0.04
	RMS		0.073	0.074	0.102

Figure 5.10: Top and front views Table 5.1: Results of dimension measurements

Figure 5.11 displays the stereo images of a block, using normal lighting sources: one fluorescent lamp from ceiling; the other from horizontal direction. The dimension of the object is about 60 × 30 × 25 mm^3, comprising about 250 by 200 pixels in the images. The edges are detected along the boundaries of the object as shown in figure 5.12, followed by line segmentation (figure 5.13). The results of the test is summarised in table 5.2. From the stereo images, some reflections can be seen on the edges of circle 3 and 4, resulting in the circles being a little bit smaller than that expected.

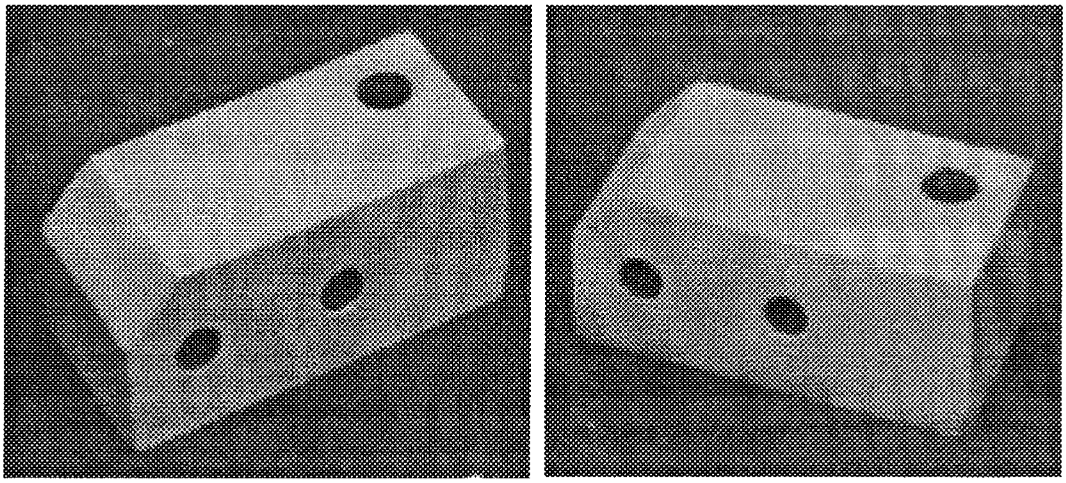


Figure 5.11 : Stereo images of a block

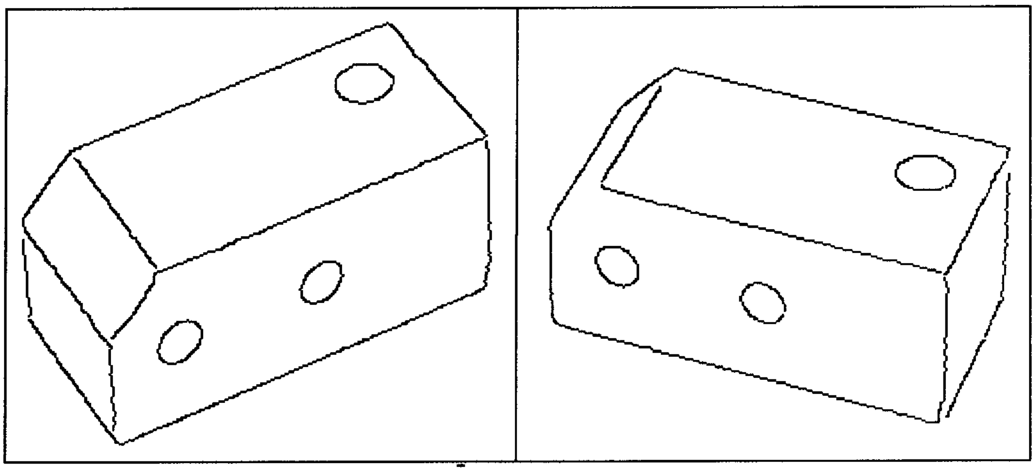


Figure 5.12 : Edges on the block

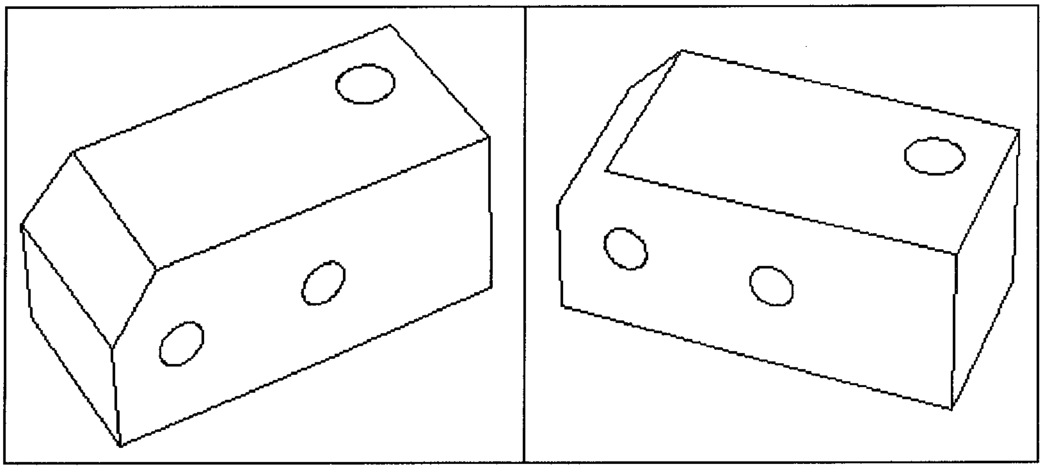


Figure 5.13 : Results of line segmentation

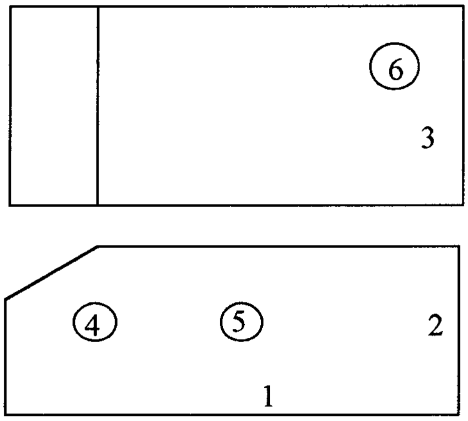


Figure 5.14 : Two main views

		Measure (mm)	Difference (mm)
Line	1	60.47	0.07
	2	25.41	0.11
	3	28.53	0.05
Circle	4	6.56	−0.12
	5	6.57	−0.11
	6	8.29	0.05

Table 5.2 : Measurement results

5.5 Summary

This chapter describes an automatic procedure to construct 3D objects by matching geometric features on stereo images. The epipolar constraint presented by a 3×3 matrix can be derived from the parameters of camera orientation. This matrix relates the coordinates in one image with their corresponding coordinates in the other image along an epipolar line. The presentation of the epipolar constraint is invariant to the location and rotation of a control frame, if the relative positions of two cameras are constant. Stereo matching of geometric features is based on the line geometry and epipolar geometry. The correspondence of a pair of straight lines in two images is determined in terms of their terminals, which should satisfy the epipolar constraint. To find the correspondence of regular curves, it is necessary to consider their size and location.

An epipolar line, which is tangential to a given ellipse in one image, must be tangential to a matched ellipse in the other. This condition includes the basic requirement for the size and location of matched ellipses. The matched elements of an ellipse can be presented as a straight line with two tangent points as its ends. In a similar manner to matching straight lines, the ellipses being matched should satisfy the condition that their corresponding tangent points lay on the same epipolar line.

A 3D straight line is simply presented by 3D coordinates of its two terminals, while a 3D regular curve can be described by a plane in object space and the parameters of a 2D curve on the plane. The plane can be determined either by the points on a given curve, or the corner points of a polygon which surrounds the curve and contains the same plane. Geometric functions of straight lines and regular curves provide more reliable information and are easy to calculate. However, there exist certain types of surfaces whose boundaries cannot be detected or viewed, resulting in the description of an object not being complete.

Chapter 6 Recognition of 3D Objects

6.1 Introduction

3D object recognition is a difficult and yet an important problem in computer vision. It is a necessary step in many industrial applications, such as the identification of industrial parts, the automation of the manufacturing process, and it is essential for intelligent robots equipped with a powerful visual feedback systems. Most computer vision systems which are designed to recognise three dimensional objects compare a scene object against entities in a model database containing a description of each object the system is required to recognise. The development of such model-based recognition techniques has occupied the attention of many researchers in the computer vision community for years.

Object recognition consists of identifying an object in a scene with a model in database, to determine its correspondences, and the 3D information of the object, such as the orientation and position. The recognition methods may be classified into global and local methods (Grimson, 1990). Global methods try to match the model by either using a template or by computing global parameters such as area, perimeter, moments, and Fourier descriptors (Bamieh and De Figueiredo, 1986; Reeves et al., 1988). Most methods assume that the image has been converted into binary format before the computation of parameters. Global methods are not well suited for recognition of most objects since they do not find correspondence between measured local features on the object and those on the model. However, they are well suited for recognising plannar targets of various shapes.

Local methods are designed to match individual features of the object to their corresponding features in the object model (Besl and Jain, 1985; Chin and Dyer, 1986; Brady et al., 1988; Fan, 1990; Flynn and Jain, 1991). The performance and competence of a recognition system is conditioned by two major factors. One is the method used to describe the objects and models, the other is the method used to establish the correspondences between objects and models. The term "object" refers to an actual object in the scene, while "model" refers to a designed model in the database. In order to match an object with a model, the method of description of matching elements in the object and model should be the same. In an industrial environment, CAD systems are usually used to design objects for the manufacturing task. They are therefore suitable for exploring for representing the model database. The term CAD-based computer vision has been used to describe research employing CAD models for various visual tasks (Brady et al. 1988; Ikeuchi and Kanade, 1988; Hansen and Henderson, 1989, deFigueriedo, 1987; Lu and Wong, 1988; Kak et al. 1988).

In general, CAD models (Requicha, 1980) may be used for the geometric description of objects based on: spatial occupancy enumeration or cell based methods, such as octrees; constructive solid geometry (CSG), comprising families of solid models together with Boolean operations on these objects to construct new objects; sweep representations such as translations and rotations to form solid objects; and boundary representations based on the descriptions of the boundaries of the objects in terms of surfaces, curves of intersection of the surfaces and relationships between them. There are advantages and disadvantages in the use of each of these structures, while the transformation between structures may be a non-trivial task. However, CAD systems have been developed to the stage where the particular representation strategy used in model-building can be "hidden" from the end user of modelling software. Most commercial solid modellers allow the user to build objects using CSG techniques, while the internal representations of the solid so constructed need not be a CSG representation. Indeed, some solid modellers maintain multiple representations of objects under construction, in order to offer the user flexibility of design, speed of display, and representational power (Flynn and Jain, 1991).

CAD models serve as a basic description of object geometry. Inference procedures of various sorts are applied to the CAD models to produce features which can be useful in object recognition procedures. Flynn and Jain (1991) produced a relational graph representation from CAD models which are stored in a database for object recognition. Goad (1986) developed recognition codes for a single object based on edge matching. Hansen (1989) employed feature filters to select useful features of objects designed on a CAD system. Burns and Kitchen (1988) proposed a hierarchical representation to store object databases with a large number(hundreds) of objects. Horn (1986) developed the extended Gaussian image (EGI) as a method to represent objects. The EGI is a collection of points, or impulses, located at positions defined by the orientation of the surface patches, on the unit sphere. A surface with a large change in its curvature is represented by points spread over a large area on the GI, while a surface with very small change in curvature will be represented by a cluster of points (a single point in case of a plane). It is obvious that the ratio of the surface patch area on the GI to the actual area on the object is a representation of the surface curvature. The advantage of the EGI representation is that it rotates with the object, and thus any object orientation can be easily transferred to another.

The approach to object recognition presented in the research is based on a digital photogrammetric technique which extracts object features in 2.5 or 3-D geometry as described in previous chapters. Therefore, the presentation chosen from CAD models should correspond to those obtained from the results of digital photogrammetry. In this chapter, a graphic representation of models transformed from CAD models using DXF file as input, is described. The graphic models are represented by planar surfaces which are bounded by straight lines or regular curves. These surfaces are grouped in terms of their normal directions and are stored together with their areas and perimeters as matching elements. The topological relations between surfaces are constructed in terms of the centre of each surface and the common edge of two surfaces. The number of models in the database could be as many as a few hundred depending on the application.

Object recognition from the digital images must be determined by matching the extracted elements of the objects with the models in database. The objects in the scene are described by the same type of relational graph presentation as the models in the database. Since the objects in the images are only partly visible in this study, although they could be completely measured if the imagery system were designed appropriately, their description will not be complete. Therefore, the number of detected surfaces in an object will always be less than the number of surfaces in its corresponding model, if there are no other object elements falsely grouped with those belonging to the object. One important characteristic of the system is that the dimensions of objects are measured accurately. As a first stage of model matching, most models can be discarded as not matching the scene object by checking their size against the size of the object. In the second stage, all visible graphic elements of an object should exist and satisfy those in the corresponding model.

Their topological relations should also be identical with the relations between the graphic elements in the model. One condition not being satisfied will indicate that the model does not correspond. The object recognition process results in the identification of the sensed object, its position and orientation.

In the next section, a drawing interchange file (DXF) will be introduced as an output file of the CAD models. Section 6.3 provides an inference engine to transform CAD models to graphic presentation. Section 6.4 presents the details of model matching process. Section 6.5 shows experiment results of object recognition on real images. Finally, section 6.6 summarises the chapter.

6.2 CAD Output: The DXF Format

A CAD system is a general purpose Computer Aided Design program for preparing two-dimensional drawings and three-dimensional models. It is widely used in manufacturing, architecture and other areas. In order to allow designs created on one CAD system to be used by others, and to assist in interchanging drawings between CAD and other programs, a drawing interchange file format (DXF) has been defined. DXF files are standard ASCII text files, which can easily be submitted to other programs for specialised analysis. Since a DXF file is a complete representation of the drawing database, for the presentation of matching features, it is not necessary to use all information in the file. Attention is concentrated on that portion of the DXF standard devoted to the description of 3D geometry.

A DXF file is subdivided into four editable sections, plus the END OF FILE marker. The HEADER section contains settings of variables associated with the drawing. The TABLES section contains several tables, each of which contains a variable number of table entries. The BLOCKS section contains the entities that make up the blocks used in the drawing, including anonymous blocks generated by associative dimensioning. All basic geometric elements in the sequence design stage of a model are stored in this section. The ENTITIES section contains entity items, which can also appear in the block sections. The appearance of entities in the two sections is identical, but this section provides the final drawing of the CAD design.

A DXF file is composed of many groups, each of which occupies two lines in the DXF file. The first line of a group is a group code, used to indicate both the type of the value of the group and the general use of the group. The second line of the group is the group value, in a format that depends on the type of group specified by the group code. For example, a line is presented by two points. The codes for the coordinates of a start point are (10, 20, 30), and for the coordinates of a end point are (11, 21, 31). Each code is followed by a coordinate value. A program can easily read the value following a group code without knowing the particular use of this group in an item in the file. A DXF file can often specify object geometry in terms of group entities such as: lines, circles, arcs and polylines. These basic details of the geometry of models can be used to construct graphic presentations for object matching.

6.3 CAD Models to Graphic Presentation

An ideal 3D representation is unique and unambiguous, and has a rich set of representable objects. The graphic presentation of models used in this research is constructed by deriving a subset of the basic geometric entities from DXF files. The computational burden of graphic presentation is not incurred at object recognition time, since the transformation of CAD models to graphic presentations need only be applied when a new model definition is created and the corresponding vision object is needed. Each model is handled separately, so, the addition of a model to the database does not change the representations of existing models.

The geometric inferencing performed first searches the entity section to find the location of the subset which models the geometric elements of a model, in blocks section. Then, all elements of the object boundaries are processed, and topological relations produced as described below. Finally, the graphic representations are stored into a model database.

6.3.1 Attributes of Geometric Primitives

The basic geometric elements of object boundaries are stored explicitly in the analytic format in the DXF file, in terms of lines, circles and arcs.

1) Line Segments: In the DXF specification, a line segment is characterised by a starting point and an ending point. The coordinates of points are stored in the list of vertices. A line is presented as two numbers of vertex and length.

2) Circular arc: In the DXF file, a circular arc is specified in its own (arc-centred) coordinate system (x_a, y_a, z_a), in which the plane of the arc is parallel to the $x_a y_a$-plane, and displaced from it along the z_a axis. The direction of the z_a axis is given and related to the world coordinate system. The primitives of an arc contain its central coordinates, z axis direction, radius, start angle and end angle. A circle is presented similarly to an arc without the start angle and end angle, while an ellipse is presented by 12 arcs which link smoothly at their ends. This system computes major axis and minor axis of an ellipse from these symmetry arcs. The attributes for an ellipse are radius (an average of major axis and minor axis), and ratio (major axis divided by minor axis). These attributes are identical with those of circles, where the ratio is 1.0.

6.3.2 Planar Surfaces and Their Topology

The inference system does not attempt to present objects in a complete way, but rather dominant features are used for model matching. Planar surfaces are chosen as the main features which are related to each other. In the inference system, planar surfaces are generated from basic geometric elements of object boundaries, which are classified into two kinds: regular curves (circles and ellipses) and polygons. Each planar surface is presented by the normal direction of the plane (α, β, γ), its central coordinates and bounded edges. A 3D regular curve for a planar patch is presented by 2D parameters projected on the plane, while a polygon is simply a group of straight lines. Additional primitives of a planar surface are radius and ratio for an ellipse or a circle, and perimeters and area for a polygon. To establish the topological relation among planar surfaces, planar surfaces are grouped in terms of their normal directions. The planar surfaces are also related by their common edges and the distances between their central coordinates.

6.3.3 Graphical Presentation of a Model

The graphical presentation of an object derived from the DXF file can be created as shown in table 6.1, which corresponds to its model in figure 6.1. The presentation of each object includes: object name, orientation section, ellipse section, polygon section, line section and vertical point section. The name MODEL 1 stands for the first model in

```
MODEL 1
 113.52      9.9      3.80        3.11
DIRECTION 0                 DIRECTION 1               DIRECTION 2
 0  1  0                     1  0  0                   0  0  1
ELLIPSE 8                   POLYGON 5                 POLYGON 4
 0  1  2  3  4  5  6  7      4  6  8  9  11            0  2  3  10
POLYGON 3                   ENDDIR 1                  ENDDIR 2
 1  5  7                     0                         0
ENDDIR 0
 0
ELEMENT_ELLIPSE 8
 0    -86.80     33.80    0.0       3.45       1
 ................................
 7    -10.15      9.75   11.0       3.45       1
ELEMENT_POLYGON 12
 0       58.22    15.80    0.0   2028.25      194.0         4
         0   3    7   26
 .................................
 11      72.70    33.25    9.8   1303.40      172.2         4
         0   3    7   26
ELEMENT_LINE 30
 0     0      1      0      1    30.5
 ..............................
 29    18     19     10     11   66.5
ELEMENT_VETP 28
 0     72.7   0.0    0.0
 ..............................
 27    10.15  11.0   9.75
ENDMOD 1
 0
```

Table 6.1 : Graphic representation and inferred quantities of bounded lines

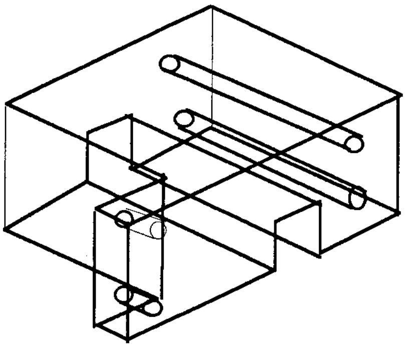


Figure 6.1 : A model displayed in CAD system

a database, following the four values which are maximum and minimum of line lengths and maximum and minimum of circle or ellipse radius. Since objects are constructed tohigh accuracy as discussed in chapter 5, by comparing the dimension between sensed objects and models, most models whose dimensions are beyond the range of the object dimensions can be ignored. The DIRECTION section contains main orientations of planar surfaces, each of which includes the list of ellipses and polygons. The ELEMENT_ELLIPSE section list all ellipses whose elements are central coordinates, radius and ratio. The direction of each ellipse is derived from direction section. The ELEMENT_POLYGON section contains all polygons whose elements are central coordinates, perimeter, area and the list of bounded lines. The ELEMENT_LINE section list all lines where the first two numbers are vertices, and next numbers are polygons between which the line lies. The final element is the length of a line. The ELEMENT_VETP section list the coordinates of all vertices. The marker ENDMOD 1 indicates the end of the first model presentation.

6.4 Model Matching

Matching between the objects in a scene and the database of the models is performed by a detailed comparison between their graphic presentations. A sensed object is presented in the same way as the models in the database. Since the object in the images is only partly visible, its description will not be complete. Therefore, the number of detected surfaces in an object will always be less than the number of surfaces in its corresponding model. The matching process contains two steps: the screener, in which most models which are unmatched to the given object are excluded, and the graph matcher, which performs a detailed comparison between the potential matching graphs and computes the 3D transformation between them.

6.4.1 Screener

In principle, the number of models in the database may be large, and evaluating each pair to find possible correspondence would be prohibitively expensive. Instead, a simple comparison between the dimensions of a scene object and models is used to ignore most models whose size is different from the size of the object. There are two elements used for dimension comparison: length of lines and average radius of ellipses. Each model has its maximum and minimum lengths of straight lines and size of ellipse if such features exist on the model. The size range of a sensed object should be within the range of a matched model, since the number of geometric features of an object is less than that of a corresponding model. Considering the errors in image processing, the possible range of model size is allowed to increase by 10% for the test. This process limits the candidates of the possible matching models to a very small number, which are then performed in the next process.

6.4.2 Graph Matcher

The graph matching procedure consists of finding the pairs (in the model surface and object surface) forming all possible sets of matching elements consistent with a single rigid 3D transform. Grimson (1990) provided a comprehensive list of possible constraints for 3D edges, cylindrical features and surface patches, some of which are used in the object recognition procedure. The process begins by finding all the possible pairs <m, o> where m and o are the model and object surfaces, respectively. The geometric comparison of surfaces is dependent on the perimeter, the area and the number of lines which bound a polygon, or radius and ratio when the surface is an ellipse. In measuring the similarity between m of the model and o of the scene object, the normalised measure of the difference is computed for each of the following properties:

- $d_{m,o}(1) = d(A_m, A_o)$, where A_m and A_o represent the surface area of a polygon in a model and an object, respectively.
- $d_{m,o}(2) = d(P_m, P_o)$, where P represents the perimeter of a polygon.
- $d_{m,o}(3) = d(R_m, R_o)$ where R represents the average radius of an ellipse.
- $d_{m,o}(3) = d(Rt_m, Rt_o)$ where Rt represents the ratio of major axis and minor axis of an ellipse.

A normalised measure between a and b is defined as follows:

$$d(a, b) = \frac{|a - b|}{\max(a, b)} \qquad (6.1)$$

Thresholds are set for each of the differences to determine whether the match is accepted or rejected. One surface of an object may correspond to more than one surface of a model, as shown in the example in figure 6.1, where the size of all circles are the same, so that one circle in an object may be found to correspond to eight circles in the model. If one surface of an object does not match with any surface in a model, however, it will indicate that the model does not match with the object and is rejected. The process results in each surface of the sensed object having multiple possible corresponding candidates in a model. It is obvious that only one matching candidate is possible, if the object corresponds with the model. Therefore multiple candidates must be reduced to a single candidate for each surface of an object. This process involves a compatibility constraint using topologic relations.

Topologic relations exist among planar surfaces of a model or an object. If two pairs $< m_i, o_i >$ and $< m_j, o_j >$ satisfy similarity measures respectively, the relation between m_i and m_j should be the same as that between o_i and o_j. Everytime a pair of nodes $< m_j, o_j >$ is selected, it is compared to all the already matched pairs $< m_i, o_i >$ using a compatibility constraint. If this constraint is not satisfied, the chosen pair $< m_j, o_j >$ is discarded. The constraint contains the following relation checks.

- Orientation Relation ($\xi 1$): Planar surfaces of a model or an object are grouped in terms of their normal directions. The angle between the orientations of two surfaces reflects their orientation relation. Let θ_m and θ_o denote the angles between the orientation of $< m_i, m_j >$ and $< o_i, o_j >$, and let $\Delta\theta = |\theta_m - \theta_o|$, then the pairs $< m_i, o_i >$ and $< m_j, o_j >$ are said to be $\xi 1$_compatible if and if $\Delta\theta$ is less than a certain threshold.

- Proximity Relation ($\xi 2$): Proximity relations summarise the distance between surface centres. Let L_m and L_o denote the distance between centroids of the surfaces m_i and m_j, and o_i and o_j, respectively, and let $\Delta L = |L_m - L_o|$, then the pairs $< m_i, o_i >$ and $< m_j, o_j >$ are said to be$\xi 2$_compatible if and only if ΔL is less than a certain threshold. If two surfaces are polygons and adjacent (ie. they share a common edge), an adjacent relation is checked between the two surfaces.

After all surface nodes of an object match with the model nodes and satisfy compatible constraints, a geometric transform is calculated between the coordinate systems of a model and object. Computing the geometric transform between matched objects not only indicates how to bring the matched objects into correspondence, but also helps to verify the matching process. The estimate of actual transform between the model and object coordinate systems can be given by a set of vertices of the polygons. If all elements of the object after transformation match with the elements of a model, the object is considered to correspond to the model, and its position and orientation are determined.

6.5 Experiments

The system has been tested on several industrial components. The CAD models were constructed from physical prototypes whose dimensions were measured by hand. AutoCAD system designs of each model in terms of the data dimension were generated, and output into a DXF file. The geometric inferencing is then performed on models to create graphic representations which are stored into a model database. Figure 6.2 displays the models listed in the database.

A sensed object is captured using two CCD cameras as shown in figure 6.3. The image processes as discussed in the previous chapters are then applied on the images to construct 3D objects (figure 6.4). The representation of the object is described in the same way as models. The maximum and minimum of the line lengths in the object are 148.89 mm and 70.08 mm, and the maximum and minimum of the circle radius are 21.04 mm and 5.06 mm. The four values are used to screen the models in the database and delete those whose ranges of dimension cannot cover the range of the object dimensions. In the database, only models 4 and 7 cover the dimension of the object, while the sizes of the other models are either too small or too large and are therefore ignored.

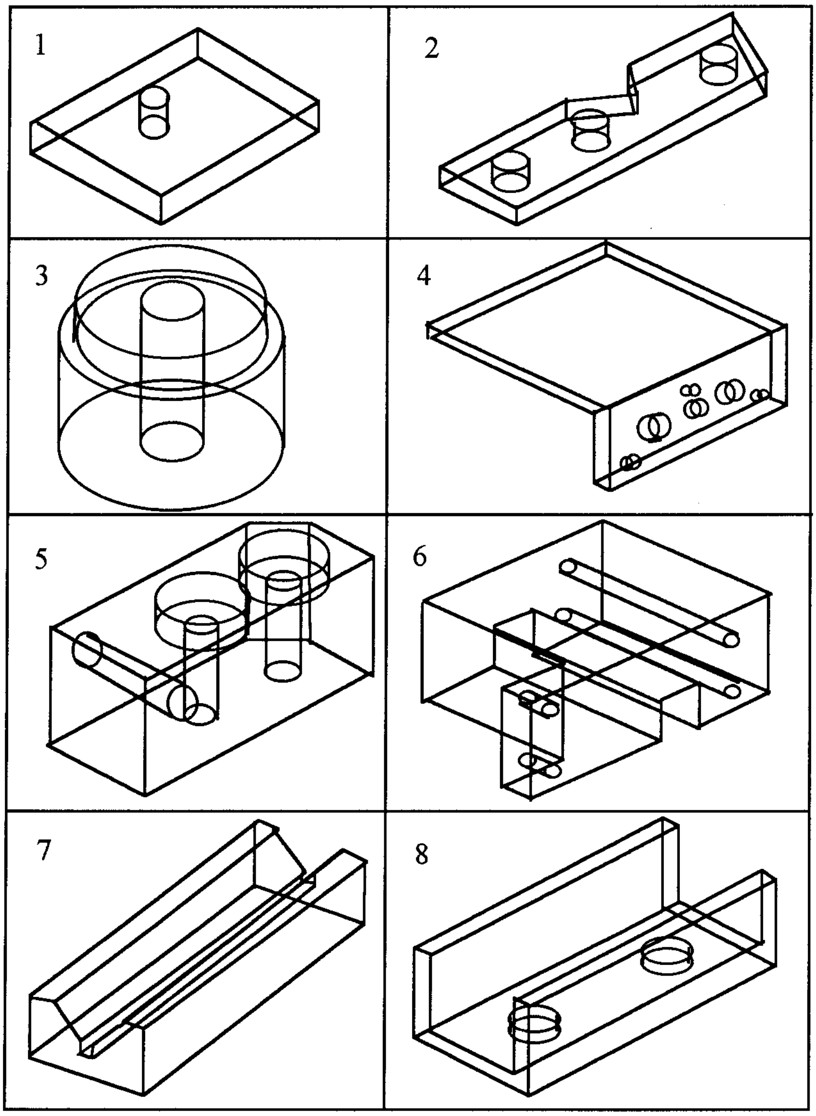


Figure 6.2 : CAD models in the database

The graph searching process is then used to find the correspondence between the nodes in the object and the nodes in the model. Graph matching starts with one planar surface.

Once one match is established, more matches can be added if the resulting match meets the constraints of the node similarity and topologic relations. After all elements of the object are matched with those of a model, the object is transformed into the coordinate system of the model. Finally, not only the object is recognised as model 4, but also its position and orientation are determined. In the example, the system on a Sun workstation takes about four minutes to complete the 3D object construction, and one second to search the correspondent model in the database. The computation time in object recognition is depended on the complex of scene objects and matching parameters.

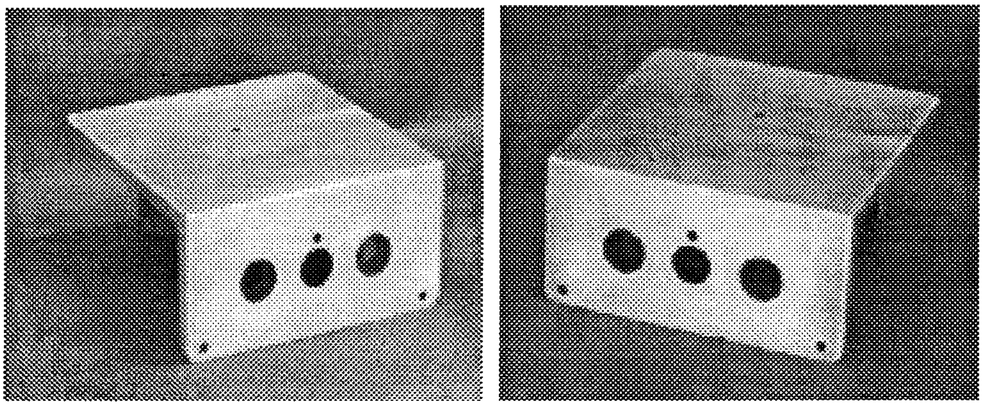


Figure 6.3 : Stereo images of an object

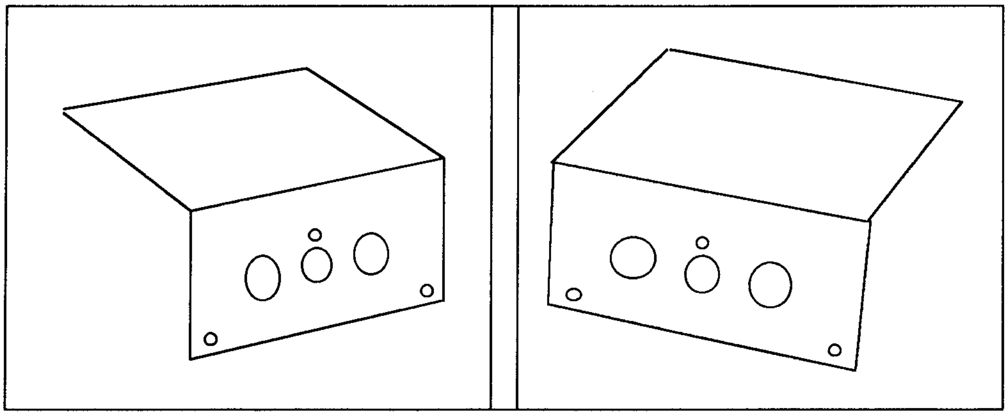


Figure 6.4 : Line segmentation of the object

Another example is the object in the example of chapter 5 as shown in figure 5.10. There are two polygons and four circles in the constructed object. However, the corresponding model has twelve polygons and eight circles as shown in model 6 of figure 6.2. Since all circle size are the same, the constraint on their topologic relations becomes very important in determining their geometric position.

6.6 Summary

This chapter provides a complete method to describe and recognise 3D objects, using boundary information of these objects. A CAD system provides designed models which are output in DXF files. An inference system is then applied on DXF files to create graphic presentation. A planar surface is presented by an ellipse or a polygon. All planar surfaces are grouped in terms of directions. Topologic properties are used to relate them to one another. Matching between the objects in a scene and a database of the models is performed by a detailed comparison between their graphic presentations. A sensed object is presented in the same way as the models in database. The matching process contains two steps: in the screener, most models which are obviously unmatched to the given object are excluded; the graph matcher performs a detailed comparison between the potential matching graphs and computes the 3D transformation between them. The object recognition process results in the identification of the sensed object, its position and orientation.

Chapter 7 Conclusion

7.1 Summary of the Thesis

The work described in this thesis is aimed at studying some methods used in digital photogrammetry and machine vision, and investigating their applications in digital photogrammetry for industrial monitoring. The main task is to develop a complete system which automatically describes 3D regular objects, using the same descriptions to represent model and scene objects, and recognise each object in the scene by identifying it with one of the models in a database. The research pays attention to not only the measurement of industrial components with high resolution, but also the interpretation of objects in the scene.

The system starts by detecting significant features of a sensed object in images. Images of industrial components are generally characterised by sharp discontinuities which represent features on the object. The accurate extraction of these discontinuities, which are characterised by the peak of the intensity change in the image, is an essential part of the digital photogrammetry system. An edge detection algorithm has been developed in chapter 2, based on a linear model which locates an edge point in the operator window with sub-pixel accuracy. The procedure commences with an analysis within 3 × 3 windows of an image to determine gradient magnitudes. These gradient magnitudes above a certain threshold will reveal that the edge may exist and must be located by the succeeding step. Location and orientation of edges are derived by a least squares fit in a 5 × 5 window. A linear model for every set of pixel corners contributes to the peak of intensity changes by using the gradient as weight, to define the position of the edge. Thinning is done to delete false edge pixels and improve the accuracy of edge location by a weighting function from both sides of the edge. Finally, the scattered edge points are chained along edges in terms of local distances between edge points.

Edge detection is referred to as a low-level image process. A simple chained edge has no geometric meaning, since in industrial environments, regular shapes such as ellipses and straight lines, often occur as elements of object boundaries. In order to efficiently interpret such objects in the scene, it is generally more relevant to present the boundaries of objects revealed in the images in geometric form. The strategy, developed for line segmentation in chapter 3, is motivated by the fact that all edge points on a straight line should indicate approximately the same edge direction, and all edge points on a regular curve should indicate the same sign of the difference in direction. The method starts with finding basic components of straight lines or regular curves by the assessment on the edge direction at the edge points along a whole edge. Straight lines or regular curves are then extended to their end points using geometric parameters determined by their components. Those lines with the same properties extracted from different edge section can be merged in terms of their geometric similarity. In order to close boundaries of objects in image space, straight lines and open regular curves are linked at their terminal points. Finally, surface patches are generated by constrained chaining the geometrical lines.

Camera orientation is an important part of digital photogrammetry, since it establishes the relationship between the 3D object coordinate system and the 2D image coordinates. The parameters are solved by bundle adjustment, using 3D control points and their corresponding image coordinates. Small bright balls are ideal targets as control points, since their shape in an image is invariant to the camera position and they are always imaged as circles. A calibration frame mounted with these balls, the coordinates of whose centres are known, serves as control. In the research, an automatic or semi-automatic procedure of camera orientation has been developed for the digital photogrammetric system. To recognise the targets, white spots with regular circular shapes are searched throughout the image. The edges are then detected using the algorithm described in chapter 2 from the small image which just covers the area of a target. The central location of targets and their size can be determined from these edge points. Using a bundle adjustment, the precise camera parameters are obtained, while the initial approximations of the parameters are transformed from DLT computation.

Using two or more cameras, 3D objects can be reconstructed by matching the same properties in stereo images with the addition of the epipolar line constraint. In chapter 5, epipolar geometry is discussed, which gives the relation between image coordinates along epipolar lines in overlapping stereo images. The epipolar constraint presented by 3×3 matrix can be derived from the parameters of camera orientation. The presentation of the epipolar constraint is invariant to the location and rotation of a control frame, if the relative positions of two cameras are constant. Stereo matching of geometric features is based on the line geometry and epipolar geometry. The correspondence of a pair of straight lines in two images is determined in terms of their terminals, which should satisfy the epipolar constraint. The locations of 3D straight lines are determined by ray intersection of their end points using the camera orientation. A 3D straight line is simply presented by the 3D coordinates of its two terminals. The correspondence of a pair of ellipses in two images is determined in terms of two epipolar lines which are tangential to a given ellipse in one image, and also tangential to a matched ellipse in the other. The calculation of 3D ellipses is based on the assumption that a special ellipse is an intersection of an object plane with a conic surface, whose apex is at the projection centre. An object plane is determined by a few intersection points on the corresponding ellipses in the stereo images. The relation between image and object planes is established in terms of the camera orientation. The ellipse on the image plane is then transformed onto the object plane. A 3D ellipse can be described by a plane in object space and the 2D curve parameters on the plane.

3D object recognition consists of identifying an object in a scene with a model in a database, to determine their correspondences and the 3D information of the object, such as the orientation and position. Models in the database are created by a CAD system and transformed into a graphic representation, by planar surfaces which are bounded by straight lines or regular curves. These surfaces are grouped in terms of their normal directions and are stored together with their areas and perimeters as matching elements. The topological relations between surfaces are constructed in terms of the centre of each surface and the common edge of two surfaces. Matching between an object in the scene and models in the database is performed by two modules: the screener, in which most models are deleted due to their size being beyond the range of the scene object; and the graph matcher, which performs a detailed comparison between the potential matching graphs, and then computes the 3D transformation between them.

7.2 Contribution

This research describes a bottom-up procedure to reconstruct 3D objects from raw image data for object measurement and object recognition. The key contributions of this research are summarised as follows:

- An edge detection method was developed to locate edges with subpixel accuracy. The precisely extracted edges in low-level image processing enable the description of objects by simple geometric properties. The edge detection method is not limited to the features with regular shapes, but any step edges. Tests have indicated that the edges can be located with a precision about 0.1 pixel, depending on the contrast of intensity.

- Line segmentation, approached from a global view, segments the extracted edges into straight lines and regular curves. This step further reduces the processed data to a few parameters of geometric functions, which are more meaningful and reliable.

- A method of recovering 3D ellipses by an intersection of an object plane with a conic surface, provides more accurate results than that by matching individual edge points along a curve. The size of an ellipse can be determined with the accuracy about 0.13 mm on the railway sleeper, and 0.074 mm within an object dimension of 103 × 50 × 67 mm^3.

- A graphic presentation of objects and models was established. It is a straight forward approach, which is efficient for object recognition.

- The digital photogrammetry system for object recognition is data-driven in that no a priori scene knowledge is required. The descriptions of the objects are computed without any knowledge about existing models, which is important when the environment is unknown.

- An automatic and semi-automatic procedure of camera orientation was developed for a digital close range photogrammetric system. It is characterised by high accuracy, high speed and reliable results.

7.3 Suggestions

The research presented here is an attempt to solve the problem of industrial monitoring using digital photogrammetry and machine vision techniques. As always in cases of a new approach, the elements of the proposed methods have to be refined and improved.

One problem which occurs in line segmentation is that the degree of noise in the location of the edge points varies in terms of lighting source, object materials, surface colour and the normal direction of object surfaces. Some straight lines may be short, while some curves may be long. A simple threshold to determine line segmentation will not fit different situations. In order to segment geometric features correctly, an adaptive threshold is required.

The system for the application of object recognition is limited to the industrial components with simple regular shapes. It is not efficient for complicated objects or objects with occlusion. One reason is that the processes of edge detection and line segmentation cannot extract small detail features correctly, since there is not enough edge information. A region growing approach may help to solve this problem.

References

Agnard J. P. and Gagnon P. A., "Consideration of Geometry Aspects in the Practice of Softcopy Photogrammetry," ARIDA News Letter, No. 7, pp. 5-10, 1995.

Andresen K. and Yu Q., "Calculation of the Geometric Parameters of an Ellipse in Space by its Edges in the Images," ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 49, No. 2, pp. 33-37, 1994.

Asada H. and Brady M., "The Curvature Primal Sketch," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 1, pp. 2-14, 1986.

Bamieh B. and De Figueiredo, R.J.P., "A General Moment-Invariants/Attributed-Graph Method for Three-Dimensional Object Recognition from a Single Image," IEEE Journal of Robotics and Automation, Vol. 2, No. 1, pp. 31-41, 1986.

Barlaud M., Gaidon T., Mathieu P. and Feauveau J. C., "Edge Detection Using Recursive Biothogonal Wavelet Transform," IEEE cat n 91CH2977-7, pp. 2553-2556, 1991.

Besl P. J. and Jain R. C., "Three-Dimensional Object Recognition," ACM Comput. Surveys, Vol. 17, No. 1, pp. 75-145, 1985.

Beyer H. A., "Geometric and Radiometric Analysis of a CCD-Camera Based Photogrammetric Close-Range System," Ph.D Thesis, Institute of Geodesy anf Photogrammetry, Swiss Federal Institute of Technology (ETH), Zurich, 1992.

Boulanger P., "Multiscale edge detection based on a new geometrically intrinsic filter," SPIE, Vol. 2350, Videometrics III, pp. 264-278, 1994.

Brady J., Nandhakumar N. and Aggarwal J., "Recent Progress in the Recognition of Objects from Range Data," in Proc. 9th Int. Conf. Pattern Recognition, pp. 85-92, 1988.

Brown D. C., "Application of Close Range Photogrammetry to Measurements of Structures in Orbit," Vol. 1, GSI Technical Report NO. 80-012, Geodetic Services Inc., Melbourne, Florida, USA, 1980.

Burns J. B. and Kitchen L. J., "Rapid Object Recognition from a Large Model Base Using Prediction Hierarchies," in Proc. 1988 DARPA Image Understanding Workshop, pp. 711-719, 1988.

Caelli T., Johnston M. and Robison T., "3D Object Recognition: Inspirations and Lessons from Biological Vision," Three-Dimensional Object Recognition Systems, Elsevier Science Publishers B. V., pp. 1-16, 1993.

Canny J. F., "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679-698, 1986.

Cheng F. and Venetsanopoulos A. N., "An Adaptive Morphological Filter for Image Processing," IEEE Transaction on Image Processing, Vol. 1, No. 4, pp. 533-539, 1992.

Chin R. T. and Dyer C. R., "Model-Based Recognition in Robot Vision," ACM Comput. Surveys, Vol. 18, No. 1, pp. 67-108, 1986.

Cristi R., "Edge Detection by 2D Recursive Least Squares and Markov Random Fields," Sixth Multidimens Signal Process Workshop, pp. 27-34, 1989.

deFigueriedo R., "A Framework for Automation of 3D Machine Vision," TI Tech. J., pp.62-72, 1987.

El-Hakim S. F., "Application and Performance Evaluation of a Vision-Based Automated Measurement System," In Videometrics I, Sabry F. El-Hakin, Editor, Proc. SPIE 1820, pp. 181-195, 1992.

Fan T. J., "Describing and Recognizing 3-D Objects Using Surface Properties," Springer-Verlag New York Inc. 1990.

Fischler M. A. and Bolles R. C., "Perceptual Organization and Curve Partitioning," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 1, pp. 100-105, 1986.

Flynn P.J. and Jain A. K., "CAD-Based Computer Vision: From CAD Models to Relational Graphs," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 2, pp. 114-132, 1991.

Förstner W., "A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features," ISPRS Intercommision Workshop on "Fast Processing of Photogrammetric Data," Interlaken, pp. 281-305, 1987.

Fraser C. S., "Network Design Considerations for Non-Topographic Photogrammetry," Photogrammetric Engineering and Remote Sensing, Vol. 50, pp. 1115-1126, 1984.

Fraser C. S., "Limiting Error Propagation in Network Design," Photogrammetric Engineering and Remote Sensing, Vol. 53, pp. 487-493, 1984.

Fua P. and Leclerc Y. G., "Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading," In Proc. DARPA Image Understanding Workshop, 1993.

Goad C., "Fast 3D Model-Based Vision," in From Pixels to Predicates, Pentland, A. P. Ed. Norwood, NJ: Ablex, pp. 371-391, 1986.

Grimson W. E. L., "Object recognition by computer: the role of geometric constraints," MIT Press, Cambridge, 1990.

Grimson W. E. L., "On the Recognition of Curved Objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 6, pp. 632-642, 1986.

Grimson W. E. L. and Pavidis T., "Discontinuity Detection for Visual Surface Reconstruction," Computer Vision, Graphics and Image Processing, Vol. 30, pp.316-330, 1985.

Gruen A. and Agouris P., "Linear Feature Extraction by Least Squares Template Matching Constrained ByInternal Shape Forces," Proceeding of ISPRS Commission III Symposium on Spatial Information from Digital Photogrammetry and Computer Vision, Vol. 30, pp. 316-323, 1994.

Gruen A. and Stallmann D., "High Accuracy Dimensional Measurement Using Non-Targeted Object Features," International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Commission V, pp. 694-700, 1992.

Haggren H., " 3-D Video Digitizing," From Analytical to Digital - A Time of Change, pp. 1-9, 1993.

Hansen C. D. and Henderson T., "CAGD-Based Computer Vision," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 11, pp. 1181-1193, 1989.

Heuvel F.A. van den and Kroon R.J.G.A., "Digital Close Range Photogrammetry Using Artificial Targets," International Archives of Photogrammetry and Remote Sensing, Vol. XXIX, Commission V, pp. 222-229, 1992.

Horn B.K.P. and Brooks M. J., "Shape from Shading," The M.I.T. Press, Cambridge, 1989.

Huang Y. and Trinder J. C., "A procedure for Fully Automatic Orientation of Camera in Digital Close Range Photogrammetry," Proc. of Optical 3D Measurement Techniques II Conference, pp. 339-346, 1993.

Huang Y. and Trinder J. C., "Reconstruction of Discontinuous Surfaces by Edge-Based Feature Extraction," International Archives of Photogrammetry and Remote Sensing, Vol. 30-5, pp. 198-205, 1993.

Huang Y. and Trinder J. C., "A Feature-Based Approach to Reconstruction of 3D Objects from Digital Images," International Archives of Photogrammetry and Remote Sensing, Vol. 30-3, pp. 391-398, 1994.

Huertas A. and Medioni G., "Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 5, 1986.

Ikeuchi K. and Kanade T., “Automatic Generation or Object Recognition Programs,” Proc. IEEE, Vol. 76, No. 8, pp. 1016-1035, 1988.

Illingworth J. and Kittler J., “A Survey of the Hough Transform,” Computer Vision, Graphics, and Image Processing, Vol. 44, No. 1, pp. 87-116, 1988.

Inoue H., Tachikawa T. and Inaba M., “Robot Vision with a Correlation Chip for Real-Time Tracking,” Optical Flow and Depth Map Generation, Proc. IEEE International Conference on Robotics and Automation, pp. 1621-1626, 1992.

Jain, R., Kasturi R. and Schunck B. G., “Machine Vision,” McGraw-Hill, Inc., 1995.

Jansa J. and Trinder J. C., “(Concept of) A Knowledge Based System for Close Range Photogrammetry,” International Archives of Photogrammetry and Remote Sencing, Vol. XXIX, Part B5, Commission V, pp 461-463, 1992.

Jansa J., Huang Y. and Trinder J. C., “Problems of Precise Target Location and Camera Orientation in Digital Close Range Photogrammetry,” SPIE, Vol. 2067, Videometrics II, pp.164-174, 1993.

Jarvis R., “Range Sensing for Computer Vision,” Three-Dimensional Object Recognition Systems, Elsevier Science Publishers B. V., pp. 17-56, 1993.

Jeong H. and Kim C.I., “Adaptive Determination of Filter Scales for Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 5, pp. 579-585, 1992.

Kak A., Vayda A., Cromwell R., Kim W. and Chen C., “Knowledge-Based Robotics,” Int. J. Production Res., Vol. 26, No. 5, pp. 707-734, 1988.

Kisworo M., Venkatesh S. and West G., "Modeling Edges at Subpixel Accuracy Using the Local Energy Approach," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 4, pp. 405-410, 1994.

Kraus K., “Photogrammetry,” Vol.1 and Vol.2, 4th Edition, Dummer, Bonn, 1993.

Krupnik A. and Schenk T., “Segmentation of Edges in 3D Object Space,” Research Activities in Digital Photogrammetry at The Ohio State University, pp. 89-98, 1992.

Liao F. Y., Middler M. and Lin W. C., “Neural Networks for Step Edge Detection,” In Proc. of IJCNN-International Joint Conference on Neural Networks, 1991.

Littlehales C. and Rioux M., “White Light Magic,” Iris Universe (Magazine), No. 18, pp. 24-27, 1992.

Lu S. and Wong A. K. C., “Analysis of 3-D Scene with Partially Occluded Objects for Robot Vision,” in Proc. 9th Int. Conf. Pattern Recognition, pp.303-308, 1988.

Lu Y. and Jain R. C., “Reasoning about Edges in Scale Space,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 5, pp. 450-468, 1992.

Lu Y. and Jain R. C., "Behaviour of Edges in Scale Space," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 4, pp. 337-356, 1989.

Lyver E., Mitchell O., Akey M. and Reeves A., "Subpixel Measurement Using a Moment-Based Edge Operator," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 12, 1989.

MacVicar-Whelan P. and Binford T., "Line Finding with Subpixel Precision," in Proc. SPIE, Vol. 281, 1981.

Mallat S., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, 1989.

Mallat S. and Zhong S., "Signal Characterization from Multiscale Edges," (IEEE cat n 90CH02898-5) pp.1596-1601, 1990.

Marr R. and Hildreth H., "Theory of Edge Detection," Proc. of the Royal Society of London, B 07, pp. 187-217, 1980.

Marshall A. R., "Network Design and Optimization in Close Range Photogrammetry," Master Thesis, School of Geomatic Engineering, University of New South Wales, 1989.

Morrone M. and Owens R., "Feature Detection from Local Energy," Pattern Recognition Lett., Vol 6, 1987.

Nalwa V. S. and Binford T. O., "On Detecting Edges," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 699-714, 1986.

Otsu, N., "A threshold selection method from gray level histograms," IEEE Transctions on Systems, Man, and Cybernetics, Vol. 9, No. 1, pp. 62-66, 1979.

Paik J. K. and Katsaggelos A. K., "Edge Detection Using a Neural Network," IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 4, pp. 2145-2148, 1990.

Papamarkos N. and Gatos B., "Anew Approach for Multilevel Threshold Selection," Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing, Vol. 56, No. 5, pp. 357-370, 1994.

Parvin B. and Medioni G., "Adaptive Multiscale feature Extraction from Range Data," In proceedings of IEEE Workshop on Computer Vision, pp. 23-28, 1987.

Pavlidis T. and Liow Y., "Integrating Region Growing and Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 3, pp. 225-233, 1990.

Philippe S. M., Chen J. S. and Gerard M., "Adaptive Smoothing: A General Tool for Early Vision," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 6, pp. 514-529, 1991.

Ramer U., "An Iterative Procedure for the Polygonal Approximation of Plane Curves," Computer Graphics and Image Processing, pp. 244-256, 1972.

Rao K. and Nevatia R., "Generalized Cone Descriptions from Sparse 3D Data," In Proc. of the IEEE Computer Vision and Pattern Recognition, pp. 256-263, 1986.

Rarvis R., "Range Sensing for Computer Vision," Three Dimensional Object Recognition Systems, Elsevier Science Publishers B.V., pp. 17-56, 1993.

Reddi S.S. and Rudin S.F. and Keshavan H.R., " An Optimal Multiple Threshold Scheme for Image Segmentation," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 14, No. 4, pp. 661-665. 1984.

Reeves, A. P., Prokop R. J., Andrews, S. E. and Kuhl F. P., "Three-Dimensional Shape Analysis Using Moments and Fourier Descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, pp. 937-943, 1988.

Requicha A. A., "Representations for Rigid Solid: Theory, Methods, and System," ACM Comput. Survey, Vol. 12, No. 4, pp. 437-464, 1980.

Roberts L. G., "Machine Perception of three Dimensional Solids," Optical and Electro-Optical Information Processing, MIT press, pp. 159-197, 1965.

Rousseeum P. J. and Leroy A. M., "Robust Regression and Outliers Detection," J. Wiley & Sons, 1987.

Samadani R. and Veseckly J. F., “Finding Curvilinear Features in Speckled Images,” IEEE Transactions on Geoscience and Remote Sensing, Vol. 28, No. 4, pp. 669-673, 1990.

Sarker S. and Boyer K. L., “Optimal Infinite Impulse Response Zero Crossing Based Edge Detectors,” CVGIP Image Understanding, Vol. 54, No. 2, pp.224-243, 1991.

Tabatabai A. and Mitchell O., "Edge Locaton to Subpixel Values in Digital Imagery," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 6, No. 2, 1984.

Tan Z., Brandstatter G. and Xu X., “Method for Solving the Inverse Problem of Photogrammetry,” International Archives of Photogrammetry and Remote Sensing, Vol. 30-3, pp. 391-398, 1994.

Trinder J. C., “Precision of Digital Target Location,” Photogrammetric Engineering and Remote Sensing, Vol. 55, No. 6, pp. 883-886, 1989.

Trinder J. C. and Huang Y., “Edge Detection with Sub-Pixel Accuracy for a Flexible Manufacturing System,” SPIE, Vol. 2067, Videometrics II, pp. 151-161, 1993.

Trinder J. C. Jansa J. and Huang Y., “An Assessment of the Precision and Accuracy of Methods of Digital Target Location,” Photogrammetry & Remote Sensing, Vol. 50, No. 2, pp. 12-20, 1995.

Wang S. and Haralick R.M., “Automatic Multithreshold Selection,” Computer Vision, Graphics, and Image Processing, Vol. 25, No. 1, pp. 46-67, 1984.

Williams D. J. and Shah M., “Edge Contours Using Multiple Scales,” Computer Vision, Graphics and Image Processing, Vol. 51, No. 3, pp. 256-274, 1990.

Witkin A. P., “Scale-Space Filtering,” In Proceedings of International Joint Conference on Artificial Intelligence, pp. 1019-1022, 1983.

Wong K.W., "Close-Range Mapping with a Solid State Camera," Photogrammetric Engineering and Remote Sensing, Vol.52, No. 1, pp. 67-74, 1986.

Wuescher D. M. and Boyer K. L., "Robust Contour Decomposition Using a Constant Curvature Criterion," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 1, pp. 41-51, 1991.

Xue K. and Breznik C. W., "A Neural-net Computing Algorithm for Detecting Edges in a Gray Scale Image," In Proc. of the IEEE Conference on Decision and Control, Vol. 4, pp. 2368-2373, 1990.

Yu X. and Yla-Jaaski J., "A New Algorithm for Image Segmentation Based on Region Growing and Edge Detection," In Proc. of IEEE International Symposium on Circuits and Systems, Vol. 1, pp. 516-519, 1991.

Zhang D., Liu J. and Wan F., "Hierarchical Relaxation Method for Edge Detection," Proc. of 1988 Int Conf Syst Man Cybern Vol. 2, pp. 906-908, 1988.

Ziou D., "Line Detection Using an Optimal IIR Filter," Pattern Recognition Letters Vol. 24, No. 6, pp.465-478, 1991.

Publications from

THE SCHOOL OF GEOMATIC ENGINEERING

(Formerly School of Surveying)

THE UNIVERSITY OF NEW SOUTH WALES

All prices include postage by surface mail. Air mail rates on application. (Effective August 1997)

To order, write to Publications Officer, School of Geomatic Engineering
The University of New South Wales, Sydney 2052, AUSTRALIA

NOTE: ALL ORDERS MUST BE PREPAID

UNISURV REPORTS - S SERIES

S8 - S20	Price (including postage) :		$10.00
S29 onwards	Price (including postage) :	Individuals	$25.00
		Institutions	$30.00

S8 A. Stolz, "Three-D Cartesian co-ordinates of part of the Australian geodetic network by the use of local astronomic vector systems", Unisurv Rep. S8, 182 pp, 1972.

S10 A.J. Robinson, "Study of zero error & ground swing of the model MRA101 tellurometer", Unisurv Rep. S10, 200 pp, 1973.

S12. G.J.F. Holden, "An evaluation of orthophotography in an integrated mapping system", Unisurv Rep. S12, 232 pp, 1974.

S14. Edward G. Anderson, "The Effect of Topography on Solutions of Stokes` Problem", Unisurv Rep. S14, 252 pp, 1976.

S16. K. Bretreger, "Earth Tide Effects on Geodetic Observations", Unisurv S16, 173 pp, 1978.

S17. C. Rizos, "The role of the gravity field in sea surface topography studies", Unisurv S17, 299 pp, 1980.

S18. B.C. Forster, "Some measures of urban residential quality from LANDSAT multi-spectral data", Unisurv S18, 223 pp, 1981.

S19. Richard Coleman, "A Geodetic Basis for recovering Ocean Dynamic Information from Satellite Altimetry", Unisurv S19,332 pp, 1981.

S20. Douglas R. Larden, "Monitoring the Earth's Rotation by Lunar Laser Ranging", Unisurv Report S20, 280 pp, 1982.

S29 Gary S Chisholm, "Integration of GPS into hydrographic survey operations", Unisurv S29, 190 pp, 1987.

S30. Gary Alan Jeffress, "An investigation of Doppler satellite positioning multi-station software", Unisurv S30, 118 pp, 1987.

S31. Jahja Soetandi, "A model for a cadastral land information system for Indonesia", Unisurv S31, 168 pp, 1988.

S33. R. D. Holloway, "The integration of GPS heights into the Australian Height Datum", Unisurv S33, 151 pp.,1988.

S34. Robin C. Mullin, "Data update in a Land Information Network", Unisurv S34, 168 pp. 1988.

S35. Bertrand Merminod, "The use of Kalman filters in GPS Navigation", Unisurv S35, 203 pp., 1989.

S36. Andrew R. Marshall, "Network design and optimisation in close range Photogrammetry", Unisurv S36, 249 pp., 1989.

S37. Wattana Jaroondhampinij, "A model of Computerised parcel-based Land Information System for the Department of Lands, Thailand," Unisurv S37, 281 pp., 1989.

S38. C. Rizos (Ed.), D.B. Grant, A. Stolz, B. Merminod, C.C. Mazur "Contributions to GPS Studies", Unisurv S38, 204 pp., 1990.

S39. C. Bosloper, "Multipath and GPS short periodic components of the time variation of the differential dispersive delay", Unisurv S39, 214 pp., 1990.

S40. John Michael Nolan, "Development of a Navigational System utilizing the Global Positioning System in a real time, differential mode", Unisurv S40, 163 pp., 1990.

S41. Roderick T. Macleod, "The resolution of Mean Sea Level anomalies along the NSW coastline using the Global Positioning System", 278 pp., 1990.

S42. Douglas A. Kinlyside, "Densification Surveys in New South Wales - coping with distortions", 209 pp., 1992.

S43. A. H. W. Kearsley (ed.), Z. Ahmad, B. R. Harvey and A. Kasenda, "Contributions to Geoid Evaluations and GPS Heighting", 209 pp., 1993.

S44. Paul Tregoning, "GPS Measurements in the Australian and Indonesian Regions (1989-1993)", 134 + xiii pp, 1996.

S45. Wan-Xuan Fu, "A study of GPS and other navigation systems for high precision navigation and attitude determinations", 332pp, 1996.

S46. Peter Morgan et al, "A zero order GPS network for the Australia region", 187 + xii pp, 1996.

S47. Yongru Huang, "A digital photogrammetry system for industrial monitoring", 145 + xiv pp, 1997.

S48. Kim Mobbs, "Tectonic interpretation of the Papua New Guine Region from repeat satellite measurements", 256 + xc pp, 1997.

S49. Shaowei Han, "Carrier phase-based long-range GPS kinematic positioning", 185 + xi pp, 1997.

S50. Mustafa D Subari, "Low-cost GPS systems for intermediate surveying and mapping accuracy applications", 179 + xiii pp, 1997.

MONOGRAPHS

Prices include postage by surface mail

		Price
M1.	R.S. Mather, "The theory and geodetic use of some common projections", (2nd edition), 125 pp., 1978.	$15.00
M2.	R.S. Mather, "The analysis of the earth's gravity field", 172 pp., 1971.	$8.00
M3.	G.G. Bennett, "Tables for prediction of daylight stars", 24 pp.,1974.	$5.00
M4.	G.G. Bennett, J.G. Freislich & M. Maughan, "Star prediction tables for the fixing of position", 200 pp., 1974.	$8.00
M8.	A.H.W. Kearsley, "Geodetic Surveying", 96 pp, (revised) 1988.	$12.00
M11.	W.F. Caspary, "Concepts of Network and Deformation Analysis",183 pp., 1988.	$25.00
M12.	F.K. Brunner, "Atmospheric Effects on Geodetic Space Measurements", 110 pp., 1988.	$16.00
M13.	Bruce R. Harvey, "Practical Least Squares and Statistics for Surveyors", (2nd edition), 319 pp., 1994.	$30.00
M14.	Ewan G. Masters & John R. Pollard (Ed.), "Land Information Management", 269 pp., 1991. (Proceedings LIM Conference, July 1991).	$20.00
M15/1	Ewan G. Masters & John R. Pollard (Ed.), "Land Information Management - Geographic Information Systems - Advance Remote Sensing Vol 1" 295 pp., 1993 (Proceedings of LIM & GIS Conference, July 1993).	$30.00
M15/2	Ewan G. Masters & John R. Pollard (Ed.), "Land Information Management - Geographic Information Systems - Advance Remote Sensing Vol 2" 376 pp., 1993 (Proceedings of Advanced Remote Sensing Conference, July 1993).	$30.00
M16.	A. Stolz, "An Introduction to Geodesy", 112 pp., 1994.	$20.00
M17	Chris Rizos, "Principles and Practice of GPS Surveying", 565 pp., 1997.	$50.00