AUTOMATIC RECOGNITION OF THREE DIMENSIONAL PLANAR OBJECTS BY HOUGH TRANSFORM TYPE OPERATIONS

by

ALBERT PURBENYAMIN
GHARACHLOU

A Thesis submitted for the degree of
Doctor of Philosophy

THE CITY UNIVERSITY

DEPARTMENT OF
ELECTRICAL, ELECTRONIC
AND INFORMATION
ENGINEERING

December 1992

This thesis is dedicated

to my sisters Katrin and Rita

for their kindness

and to my mother Zina Bit-Badal

for her love, warmth, tenderness

and never failing support

without which this thesis would have never been written.

TABLE OF CONTENTS

Chapter 4 Theoretical Concepts of the System

Chapter 5 Preprocessing Process

Chapter 6 Extraction of Edges and Matching Process

Chapter 7 Experimental Results

Declaration

I grant power of discretion to the University librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purpose, subject to normal conditions of acknowledgement.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Dr. John Hill for the supervision of the project and his encouragement during the work.

I am extremely grateful to Dr. Tim Ellis for his invaluable advice and guidance.

I am very thankful to Dr. Geoff West for his assistance throughout the project.

I particularly would like to thank Mr. Mike Mogford, for his help and technical support.

I like to thank Dr. John Ngwa-Ndifor and Arabella Bit-Varda for their hand in preparing the thesis.

ABSTRACT

This thesis describes an investigation into the recognition from range data of three dimensional objects with plane surfaces. In it a Hough transform type operation is used to identify objects. This is adapted for three dimensions and uses a voting scheme to identify objects.

First, all available edges of the object present in the scene are extracted. Then, two edges of the object and two lines of a model are taken at a time. These are pruned and potential matching lines are selected. Next, geometric transformations necessary to take them into a fixed position in space are calculated. Matrices resulting from successful matches are computed and stored. The presence of an object similar to a model results in the generation of the same matrices. Recognition is achieved by choosing the model with the highest occurring matrix.

In order to extract edges a vision system is designed and set up. In it a stripe of light generated from the projector together with a camera is employed. A procedure to calibrate the system and extract three dimensional information is devised. Then objects are scanned and from the images taken, coordinates of edge points are computed. Next, edge points are linked and edges of the object are extracted and a recognition algorithm is applied.

The system is tested on objects with varying complexity. Recognition is performed in two different categories. First objects are placed on a specific face. Then they are recognised in arbitrary position and orientation. For each object the results and implications of the recognition algorithm, are investigated. A modified version of the recognition algorithm with two and three connected lines is tested and compared with previous experiments.

CHAPTER ONE

INTRODUCTION

The subject of the present study is three dimensional object recognition. In this opening chapter, those fields and areas that are related to it are discussed. Then, the subject itself is described and some of its aspects are clarified. Next, the objectives and goals of the present work are specified and, finally, the content of the thesis is outlined.

1.1 Related Fields

In recent decades many great achievements and breakthroughs have occurred in science and technology. Advancements have been made that precipitated numerous new inventions and created many new theories. As a result there are many fields of studies which are newly created. Computer Vision is one of these subjects. It s function can be defined as the construction of explicit, meaningful descriptions of physical objects from images (Ballard and Brown 1982). Initially it has come about because of the following two factors.

1) The advent of modern computers and their continuing improvement and availability.
2) The ability to obtain digitised information by cameras.

In the first case, computers can be thought of as machines that can sort out great amounts of data and perform large amounts of calculations in a short space of time. Tasks that otherwise would be immensely difficult or practically impossible to handle by hand. In the second case, information is stored in digitised form in images. These are normally made of N by M (i.e. row and column) pixels, each with a certain grey level value. Since computers are capable of managing large amounts of data, they are employed to conduct various operations on images. This in turn is used to interpret their content. Normally in Computer Vision images are taken and some processing is done and interpretation is performed. Because in object recognition all these functions are performed it is considered to be part of Computer Vision.

The other area which is closely related to Computer Vision is Image Processing. The main concern in this field is image to image transformations. Interest in this subject can stem from improvement of pictorial information for human interpretation and enhancement, or highlighting particular features for specific purposes. But, this does not include recognition or manipulation or thinking about objects. However techniques in this subject are used to extract necessary information from images to interpret the content of images.

Object recognition can also be associated with Artificial Intelligence. This newly created subject is mainly concerned with making computers capable of human intelligence. This normally does not include the exact imitation of information processing by the mind. Since object recognition can be considered to be an intelligent function it can be classified under it. However, as a result of recent advancements in these fields each is developing its own distinct character and pursuing separate paths. Because of this, it is no longer possible to associate them as such.

The ability to recognise objects and determine their pose and orientation is applicable to manufacturing operations. Because of this object recognition is an important subject in Robotic Vision. This field as far as vision is concerned, can be considered within the general discipline of Computer Vision and its overlaps. But it is specifically concerned with applications related to the tasks such as automatic assembly and manipulation or inspection.

Recognition of objects may require analysing and interpreting images and extracting particular features from them. Information that may also be used in matching processes. As a result, subjects such as Pattern Recognition or Computer Graphics may also be involved. From what has been outlined it can be concluded that, object recognition is strongly an interdisciplinary subject which incorporates many fields of studies.

1.2 Object Recognition

Object recognition can be seen as the ability to identify and distinguish an object present in the scene from several others. This can also include locating, and determining its orientation relative to a known coordinate system once it is identified. In this definition, the word object is rather general and can imply many things. In this context, it is restricted to solid objects only. By solid it is meant that the distance between any two points on the object remains the same, as the object undergoes any legitimate transformation in the scene (Grimson 1990). Objects can be classified as two or three dimensional depending on their shape, or volume, or recognition requirements. The recognition process can involve more than one object together with some occlusion from the camera view.

The term recognition also indicates that objects to be identified are already known. Most industrial parts recognition systems are model based. That normally involves feature extraction, object modelling and matching. Features can be vertices, edges or patches of surfaces. Modelling includes using the shape of objects together with some geometrical constraint or measurement. Matching involves comparing features derived from the scene and features stored in libraries of previously defined models. Modelling can be view independent or view dependent depending on the vision system. Recognition is usually achieved by interpreting matches obtained between correspondent features of the object and models.

The difficulty of recognition depends on several factors such as the number and complexity of objects in the scene, the number of objects in the model database, and the amount of prior information about the scene. Also in applications related to manufacturing operations, qualities such as speed, accuracy and flexibility may also be included. The appropriate technique for object recognition depends on the specification of the task. A comprehensive description of various aspects of the subject can be found in Fisher(1989).

1.3 Objective

The goal of the present work is to design and build a vision system for the recognition of three dimensional objects. Building such a system would need to fulfil the following steps:

1) To establish a procedure for extracting depth information from the scene.
2) To perform some processing on the images in order to extract useful information related to the recognition of objects.
3) To construct required features of the objects.
4) To carry out a model based recognition algorithm for identifying and distinguishing an object in the scene from several others.
5) To investigate ways of improving the system if it is necessary and feasible.

A proper recognition system should be easily calibrated. One of the first steps in building a recognition system involves establishing required significant features and then deriving those features and organising them into the appropriate structure. Usually, in this process problems such as coherent representation of data or noise need to be dealt with. Also if necessary models of the objects should be automatically constructed. The correspondence between an object in the scene and a description of its model should be comparable and adequately realised. The shape and size of the objects and whether they are planar, curved or convex is determined by the recognition algorithm.

1.4 How the Work Developed

The present work, developed with the goal of exploring problems and obstacles involving building a recognition system is based on the direct extraction of three dimensional information from the scene. The practical work began explicitly with the aim of investigating the use of structured light, in 1986. For that a number of slides containing lines and grids of various sizes were prepared and their projected effects on the scene were examined. The questions asked at the time were, how to extract stripe line or grid lines from the images and how to effectively calibrate the camera. For that several ways of extracting and segmenting lines from the image were tested. Also to determine and solve for the parameters necessary to calibrate the camera the solution of simultaneous linear equations was attempted.

Meanwhile efforts were started to fit curves to the surfaces using the Nag routines. The aim was to recognise the objects through an interpretation tree and relations between edges and surfaces. Alternatively, the aim was to employ a voting scheme and transfer combined lines and surfaces into fixed positions in space.

Up to the this stage a Prime Computer was used and programmes were written in Fortran. Here, it becomes clear that if the system is to be automated then image acquisition hardware and a scanning table need to be used. For that programmes needed to be translated to Pascal and transferred to the appropriate PC computer. At this stage the problem of camera and calibration using stripe of light is solved and the option of using a grid set aside. In translating programmes from Fortran to Pascal many alterations especially with Nag routines were found unavoidable. Based on this and because of the suitability of the stripe to extract edges, it was decided that it would be most advantageous if recognition proceeded with polyhedral objects.

The method of recognition that is applied is similar to the Hough transform which is adapted for three dimensions. The rest of the research is conducted on this path. Objects that are used are planar with straight edges and roughly the same size. At the end images are taken, necessary processing conducted, matching performed and objects automatically recognised.

1.5 Outline of the Thesis

Chapter two starts by exploring previous related works in the field of three dimensional object recognition. In it, a short and concise description of relevant papers is given.

In chapter three, various methods in extracting three dimensional information from the scene are discussed. Then, equipment together with their set up is described and the system is calibrated.

Chapter four is concerned with the recognition algorithm itself. Here, first basic mathematical concepts are described. Then the theoretical aspects of the system in two dimensions and then in three dimensions are explained. Next, a few potential steps to improve the performance of the system are suggested. Then a number of modified algorithms are introduced.

Chapter five deals with extracting preliminary data from the scene in order to build the required features of the objects. This includes scanning and grabbing images, removing spurious data and locating and grouping edge points.

The actual details of edge extraction and classification are given in chapter six. Here, practical steps involved in the rotation and translation of edges together with the matching process are explained. Also the procedure by which some of the parameters were selected is described.

Chapter seven is mainly concerned with the results obtained from experiments. Nine different objects are introduced and their significant features are outlined. Next, each of those objects is recognised by placing them in various positions and in distinct circumstances in the scene. Then a number of problems encountered in practice are specified. Finally in chapter eight conclusions are given and suggestions for further study outlined.

CHAPTER TWO

STATE OF THE ART IN 3-D OBJECT RECOGNITION

In this chapter a selection of papers relating to three dimensional object recognition is presented. From these, it is intended to draw an overall picture of the subject and outline previous works in this field. For most of the papers a short description is given.

2.1 Organisation of the Survey

The body of literature related to recognition, representation and orientation of three dimensional objects is vast and widespread. It incorporates objects with different shapes and includes techniques and algorithms with diverse applications. One way to explore this subject is to arrange it according to the shape of the objects. That is, to classify it into topics such as block world, planar or curved objects etc.

In the same context R.T. Chin and C.R. Dyer (Chin 1986) grouped reported studies into three classes of 2-D, 2 1/2D, and 3-D representation. These are organised such that systems within each class usually make similar assumptions. In 2-D representation the viewpoint is fixed and only a small number of stable object positions are possible. Here, spatial descriptions are given in image space. In contrast, in 3-D spatial description, exact representations are given in object space using an object centred coordinate system. Here, representations are viewpoint independent. In 2 1/2D representation, features defined in surface space are used. Spatial descriptions are viewer-centred and depend on local surface properties of the object in each view (i.e such as depth or orientation).

Also it is possible to study the subject, according to the following two main areas:

1) Techniques which primarily rely on geometric constraints in the scene to recover shapes from two dimensional images or images.
2) Methods in which a depth map is obtained in order to construct the scene.

Beside these, in some cases it is convenient to study the subject according to specific techniques or methods that are employed.

The present survey is arranged, giving consideration to combinations of most of these factors. However, considerable emphasis and priority is given to reports in which:

1) Range data is employed to extract depth information from the scene.
2) Hough transform type operations are used and adapted to three dimensions.
3) Polyhedral objects are involved in the process.

These are included because many aspects of the present work are related to these topics. In addition, a number of the papers with historic significance or interesting content are also included. Some related topics such as bin-picking, and shape from motion are omitted.

2.2 Block Worlds L. G. Roberts (Roberts 1963) is one of the earliest researchers that initiated the field of computer vision related to three dimensional objects. His work is concerned with extracting a line drawing from an image of the scene, and then interpreting it. The condition under which objects are studied is known as Block World. In it white plane faced blocks are studied against a black background. Models of simple polyhedra are used in order to match line drawings. All objects are made of a combination of those primitives. Then a model is taken and its vertices are matched with the vertices of the line drawing. An evaluation function is defined to assess the result of the matchings. Then by partial matching, compound objects made of simple primitives are recognised.

A. Guzman (Guzman 1968) developed an analysis of line drawing known as "geometric theory of object identity". In it polyhedral are recognised without using explicit models of objects. First, a two dimensional scene, made of the set of regions generated from the various objects with plane surfaces is considered. Then by partitioning the set into appropriate subsets a three-dimensional body or object is identified. This is done by gathering enough information from the interconnection between regions, lines and vertices to suggest that two or more regions come from the same object. In this process particular attention is given to identifying and classifying various types of junctions obtained from incident lines. Then regions that get linked together by enough evidence are considered as "nuclei" of possible objects. However, in this process it is not easy to rectify mistakes since many heuristic rules have no theoretical foundation.

2.3 Line Labelling

The above work influenced later research by considering useful constraints on the type of lines at vertices. Huffman (1971) and Clowes (1971) independently produced similar work paying attention to the interpretation of line drawings. In Huffman's work depending on vertices, different types of lines are identified and a labelling scheme is proposed. Within constraints these are convex edge, concave edge, and occluding edge. Objects are assumed to be polyhedral and with exactly three surfaces coming together at each vertex. Constraint propagation is used to label lines. In it, a possible list of junction types are identified while labels remain unchanged throughout the process. Waltz (1975) interpreted more general scenes by including lines of shadow boundaries and cracks (i.e. intersection position of four or six surfaces). Here, in order to label the elaborate line drawings a sequential labelling method for junctions is proposed. In it, possible junctions are attributed one at a time. Rosenfeld et al. (1972) proposed constraint satisfaction as another systematic approach to the problem of line labelling . Here line labelling by a relaxation method is used. In it possible junction sets are updated during the process. Sugihara (1979) employed slit light projection to classify convex and concave lines directly. Also the junction dictionary containing possible configurations at a junction was introduced. Line drawings are compared with the list of possible junctions. If it is found that it is inconsistent with physical reality then the locations of the missing edges are predicted from the dictionary. In this process the result of partial processing at any level is examined and reflected in the later stages.

The main drawback with line labelling is its serious limitation. There are not many applications one can think of and also objects are limited to polygonal with three surfaces meeting at each vertex. Because of this it is hardly a satisfactory solution to the problem of object recognition.

2.4 Planar Objects

In this section, relations between planes as well as edges and vertices are utilised to recognise objects. Objects with plain surfaces are included only.

Shirai and Suwa (Shirai 1971) used a range finder to extract information from the scene. This is achieved by projecting a light beam through a vertical slit on the objects. Next, the objects are scanned and the three dimensional positions of points are calculated. Then, vertical and horizontal planes are extracted and from their shape and also the relation between planes, objects are identified.

Shirai and Saburo (Shirai 1972) used sequential lighting from several directions to extract line drawings from the objects. This helped to reduce the negative effects of shadows or poor edge contrast due to the small difference of illumination of two planes. The method is particularly advantageous for scenes with several overlapping polyhedra where one picture edge obstructs some of the other edges and does not produce the desired results.

Underwood and Coates (Underwood 1975) developed an algorithm in which a sequence of views from a solid planar object is shown a model of the object is automatically generated. The model consists of a description of the object's surface and how they are related to form the object. Then the view taken from the new object is used to match with the models. For multiple matches, the match with the smallest matching error is considered to be correct.

A. Kiessling and F. Rocker (Kiessling 1975) used a structured light technique to illuminate simple planar objects such as cubes and wedges. First parallel grid light is projected onto the objects. Then those points on the stripe which significantly changed direction are extracted. Next their three dimensional coordinates are computed and from that, edges and surfaces are extracted. Then objects are compared and classified with the models using features such as number and orientation of planes and number of edges in the plane and relative length of edges.

Wei and Gini (Wei 1983) used a vertical projector to produce an elliptical or circular bright area on the objects. Then the distance between the camera and centre of the projected beam is calculated. Also the direction of the slope, together with the angle between the light beam and the plane are calculated. This is achieved by means of trigonometrical computations. Then by moving objects under the camera various parts and features of the objects are extracted. K. C. Wong et al. (Wong 1991) grouped two line segments into open and closed polygons that correspond to feasible physical 3D structures. Both heuristic and physical rules have been utilised to control the search effort required in the polygon formation process.

2.5 <u>Curved objects</u>

In this section objects with smooth curved surfaces are considered.

Hall et al. (Hall 1982) used a single recorded image to extract surface coordinates of such objects. This is achieved by projecting a pattern of the grid onto the scene. First, a process for the calibration of the scene is described. This involves locating a few correspondent points from the image and scene and solving the consequent equations. Next a curved object is placed on the scene and a grid pattern is projected onto its surface. Vertices of the grid are extracted and their three dimensional values are calculated. Then a quadric polynomial is fitted to points that are located on the surface of the object. Quadric surfaces are classified according to the coefficients of their equation. These include surfaces such as cones, cylinders, elliptic paraboloids etc. Then according to the outcome of those coefficients surfaces are resolved.

B.C. Vemuri et al. (1986) implemented an approach where 3-D objects are represented by regions. These are derived from collections of patches which are homogeneous in curvature-based surface properties. First, range information is obtained for a single view of the scene by a laser ranging system. Then the range image is divided into overlapping windows and jump boundaries are detected. Then surface patches are fitted to windows of continuous range data. For each patch the principal curvatures are computed. This indicates how fast the surface is pulling away from its tangent plane at each point. Then patches are classified into groups such as parabolic, elliptic etc. Next, regions on the surface of an object are grown on the basis of this classification. Models are built from data obtained directly from the laser range device. Matching of an unknown view of a known object with its model is then performed.

Paul J. Besl and Ramesh C. Jain (Besl 1986) used classical differential geometry to obtain a complete local description of smooth surfaces. Here, mean curvature and Gaussian curvature are utilised as surface characteristics that possess desirable invariance properties. Then, based on the sign of these curvatures the surfaces are classified. These include eight basic viewpoint independent surface types such as flat, hyperbolic, valley etc.

2.6 Sculptured Objects

In this section the majority of objects possess no specific known geometrical shape. In most cases objects are studied by extracting regions and patches from their surface.

O.D. Faugeras et al. (Faugeras 1982) described a vision system for building a model of fairly complex industrial parts. Active triangulation in the form of laser range finder is utilised to measure the position of points on the parts. Then a three point seed algorithm is employed to represent the object. In it, the surface of the object is approximated by a set of planar faces. In this process a surface is fitted to three close points. Then the immediate near points are incorporated into the surface if the required conditions are satisfied. Non planar approximations and volume representation are also included. In the former, one normal to the surface at each point is calculated and represented as a point in the unit sphere. Recognition is performed by matching the face description of an unknown view with the stored model. In it planar faces are used as primitives. The descriptions of the surfaces are achieved by comparing elements, such the as number of points on the face and on the boundary, etc.

The continuation of the above work by O. D. Faugeras and M. Hebert (Faugeras 1986) considers tree search to guide the matching process. In it, at each level, translational and rotational matrices obtained between object and models are evaluated. Then paths are pruned if some error condition is not realised. This is performed by taking into account the number of matched features between the object and its model. The particular features that are used include significant points, curves and planes. No occlusion from other objects is introduced. Since the system relies on large numbers of surface patches it can be potentially sensitive to small changes in the viewing direction.

B.C. Vemuri and J.K. Aggarwal (Vemuri 1987) described a system for the representation and recognition of an object and identifying each of its features. Objects are tested on data obtained using a laser ranging system. In it objects and models are represented by regions which are derived from surface patches homogeneous in the sign of principal curvatures. The task of recognition is performed with an unknown view of the object, compared to its models obtained from different directions. This is performed by finding the one point correspondence between curves of the same constant principal curvature on the object and model surfaces. The model which registers the minimum error with the unknown view of an object is selected from a collection of candidate models. Principal vectors are employed to determine the

3-D rotation required to bring the model into the same orientation as that of the object.

R. Hoffman and A. K. Jain (Hoffman 1987a,1987b) attempted a complete recognition system by segmenting range images and classifying patches. In this process three types of surfaces, planar, convex and concave are defined. By this approach many of the complexities involved in representing objects are reduced. Also various edges such as creases, or jump edges are distinguished. In addition to that different curvatures such as minimum and maximum, or mean curvature are specified. Then an evidence based approach is adopted to perform the recognition process. In it object models are given a set of evidence features and an array of evidence weights. These indicate the degree to which each evidence feature supports or refutes the presence of the object. A decision on each model is made by observing which of these hypotheses are compatible with the object and therefore true. Then, a likelihood based on the strength of belief is derived. At the end, these likelihoods are compared to obtain a final decision. The system is reported to perform well but with some difficulty in handling unremarkable (i.e. insignificant features) objects.

Fridtjof Stein and Gerard Medioni (Stein 1991) used edges together with small surface patches as two different types of primitives for matching. The strategy is to use whatever information is available. For that, lines corresponding to depth or orientation discontinuity are extracted with whatever information current state of the art edge detectors can generate. Also for smooth free form surfaces differential properties called splashes are derived. These are then encoded by sets of super segments, consisting of connected line segments. Then they are entered in a hash table and provide the essential mechanism for fast retrieval and matching. Models are generated automatically and all features are recorded in a data base. Clusters of mutually consistent hypotheses represent instances of models.

2.7 Cylinder Type Objects

If a plane crosses a cylinder, then on its surface, depending on the angle of intersection, a circle or ellipse would emerge. This property is extended and used in the form of generalised cylinders in order to describe cylinder like objects. A general cylinder can be defined as the surface created by moving a cross section along an axis. Extensions may include axes which are curved or circles which differ in radius along the axis. Also cross sections which are not circular or perpendicular to the axis can be incorporated.

Agin and Binford (Agin 1976) used laser light stripe to scan the scene, 45 degrees from both sides of the vertical axis. Then straight lines or second order curves are fitted to the centre points of the stripe. Next, curves are grouped according to how near or parallel they are. Then a generalised cylinder is defined as a succession of points along a smooth curve and the 3D positions of points are used to establish it. In this approach, however, the relationship between surfaces and different parts of an object is not considered.

Yang and Tsai (Yang 1989) used a laser stripe to obtain horizontal cross sectional slice shapes to represent reference models of curved 3D objects. These are used to recognise objects by matching them to cross-sectional slice shapes obtained from the input image. Since matching is taking place in 2D, it can be assumed that 3D object recognition is achieved by 2D shape analysis techniques. Matching is accomplished in two steps by first registering and then evaluating the match. In the first step the centroid of either shape is translated to that of the other. Then one of the shapes is rotated until its principal axis is coincident with the other. Next the effectiveness of the match is measured by the Distance Weighted Correlation (DWC) which takes into account the geometrical distance between points. The decision tree by which matching is conducted efficiently is also introduced. As a result many models are recognised with just a single cross-sectional slice and a recognition rate of 95% for various positions of different objects is achieved.

2.8 Hough Technique and Recognition from 2-D Images

A generalised Hough transform relies on the accumulation of independent pieces of evidence for a match of model and object in the scene (or incidence of object in the image). First, local features with the same geometrical characteristics (i.e. such as edges or adjacent surfaces) are matched and resulting transformations are calculated. The presence of an object similar to a model results in the generation of a compatible translation. This produces a compact cluster in transformation space. Determining the orientation and location of surfaces or the recognition of objects is based on the detection of this cluster. In this section the recognition between an incidence of the object in the image and its three dimensional model is reviewed.

D.H. Ballard and D.Sabbah (Ballard 1983) considered the case where an object is represented by two frames. First body centred (viewer independent) and then viewer centred (image) frames are considered. Then it is proposed to use changes between these two frames in order to detect the object. The key problem is matching an internal description of the shape with the image data. If the object is present in the image then a transformation would exist which can be explained as the object. The principal idea is that correct matches between the image data and object data will define a common translation of the viewer centred frame with respect to the body centred frame. Conversely, incorrect matches will tend to define disparate translations. In other words each match can be thought of as a vote for a particular value of transformation. The transformation values receiving the most votes become the viewing transformation. Here, first it is indicated that orientation is easier to detect than translation. This is because it is functionally independent from translation, whereas the reverse is not true. Then a number of algorithms for detecting translation and orientation both in two and three dimension is proposed. Objects can be detected by first detecting the orientation of the body. Then it is only necessary to see if the oriented faces all have a consistent translation value. However, no practical example is presented.

Silberberg et al. (Silberberg 1984) described an iterative Hough procedure for recognising images of three dimensional objects. In it a model consisting of straight edges is matched against a set of line segments extracted from an image of that model. The image is taken from an unknown viewing position and consists of a single cube with letters T ,K on its surface to distinguish the faces. For each viewpoint a pair of line segments, one from the model and one from the image is taken. Then, the model line is projected onto the image plane and rotated so as to be positioned parallel to the image line. Next, it is translated so that it covers the image line. In this operation the maximum likelihood estimate of the viewing transformation is constructed. In it, first a few approximate candidate view point regions are estimated. Then, this is refined to provide a finer resolution estimate of the exact viewing position. In this process a parameter space clustering technique is used.

The extension of this work by Silberberg et al. considered a vertex from a 3D model and a junction from an image (Silberberg 1986). Each model vertex is matched to each image junction to compute a set of possible transformations. If the model vertex faces the viewer direction, then a cell in the accumulator array is incremented according to the transformation. When all the matches are made, non maximum suppression is applied to the smoothed neighbourhood of each of the cells in the list. The best estimate is given by that cell with the highest count. After a transformation is estimated a top down process is applied so that correspondence between model and image points can be determined. Then the least square algorithm is used to improve the transformation.

C. I. Connolly et al. (Connolly 1987) discussed a method for generating a 3-D model from range data. This model is used to recognise an instance of the model in an intensity scene. Matching is performed by first assigning projections of model edges and vertices to edges and vertices derived from image data. Then the transformation between the model coordinate system and image coordinate is obtained. The validity of the transformation is established with a voting or binning scheme. If the object is a rigid body then all valid assignments should result in close transformations. Usually, a key feature such as long edge or a set of parallel lines is used to establish an initial transformation. In this process, no initial key feature is provided and the validity of a transform is determined solely on the number of similar transform values. This provides robust criteria for selecting valid assignments.

Friedrich M. Wahl and Hans-Peter Biland (Wahl 1986) described a method to recognise polyhedral by transferring the captured image into the Hough space. This is achieved by single grey-level images of objects and by successive decomposition of their corresponding Hough-space representation. However, it is found that too short and fuzzy edges in noisy environments will make it difficult to identify all clusters in the accumulator array.

2.9 Hough Technique and 3-D Objects

In this section recognition is achieved by acquiring direct range data from the scene and matching it to the 3-D models. In each case several different objects are involved and differentiated from one another.

B. A. Boyer and J.K. Aggarwal (Boyter 1986) assumed that a scene is composed of rigid polyhedra objects, the description of which, exist as models in a database. If corresponding features between object and model are taken then the rotations and translations that cause them to coincide would result in the same matrix. For this, first the equations of planes are computed and edges and vertices are extracted. Then, the set of all the possible translation and rotation matrices between two edges of the object and models are calculated. The model which yields the best matching matrix is chosen as the final match. This can be thought of as an application of the Hough transform in the matrix space. This approach, like most Hough techniques, can be expensive in terms of computation. Also if objects with a large number of features are used or large libraries are employed it can lead to exponential explosion. In the experimental results that follow a voting scheme is used to identify the object. In this process scenes with only one object as well as a pile of several different objects are recognised.

Dhome and Kasvand (Dhome 1987) used a method, based on the principle of the Hough transform to recognise polyhedra in range data. The presence in the scene of an object similar to the model generates several compatible hypotheses. The recognition is based on the detection of the largest sets of agreeing hypotheses. For this the different objects are modelled by a set of local geometrical patterns (LGP). This is chosen to be a set of pairs of adjacent surfaces represented by local normals. Since the angle between the normals of two adjacent surfaces is invariant to rotation and translation this can be used for matching. The process of characterising the corresponding position of the model is conducted in the three level hierarchical partition. These include computation of model view axis, model orientation and model centre position. This is because the number of surfaces are not large enough to employ a voting scheme. Also, this minimises the computation, and simplifies clustering. Since at each level, some matches can be given up, as a result of belonging to a too small cluster. The experimental results obtained from this method correctly analysed artificial and real scenes with overlapping objects.

2.10 Hough Technique and Determination of the Pose

Determining orientation, location and size of the primitive surfaces relative to models can be useful intermediate tools in object recognition.

R. Krishnapuram and D. Casasent (Krishnapuram 1989) introduced a technique for determining the location and orientation of objects from a single range image view. First it is argued that for each point in the range image a position vector can be constructed. Then a plane is considered with position vectors stretching from the origin of the coordinate the system to the surface of the plane. Next it is shown that the dot product between any of these vectors and the unit vector perpendicular to the plane produces the same value. As a result all these points on the plane will vote for the same point in 3-D Hough space and will produce a peak of three. From this, the orientation and size of the planes can be calculated. To construct the discrete Hough space a finite number of unit vectors are required to cover evenly the surface of the unit sphere. However, even distribution is found difficult to achieve toward the poles. The technique uses range images directly with no preprocessing such as edge or gradient detection.

T. Nagata and H. B. Zha (Nagata 1988) described a method of determining the parameters that represent orientation, location and size of primitive surfaces. The input to this procedure is the images' needle map where the surface normals are computed for each picture cell or range data. Parameters are estimated by mapping and clustering such a needle map into the parameter space. Cylindrical and conical surfaces produce Gaussian spherical maps of very simple figures. This is used to map the needle map into the orientation parameter space independent of the others. Then a least square fitting algorithm is utilised to divide the orientation parameter space further into two sub-spaces in order to save computing time. Using the Hough transformation in the sub-spaces iteratively the orientation parameters are derived. The method is independent of viewer direction and is applicable to images with partially occluded parts.

2.11 Systems with 2 1/2D Surface Representation

In this section scene surface properties are derived from a single viewpoint to define features and construct models. If multiple views of an object are required then each is modelled independently.

Oshima and Shirai (Oshima 1975,1979,1983) used range data to recognise three dimensional objects with planar and smooth curved surfaces. In this process they constructed relational feature graphs in which nodes represent surfaces and arcs exemplify the relationship between them. In the preprocessing stage, first the three dimensional coordinates of the points are calculated. Then points are grouped into small surface elements and then merged together into regions classified as planar, curved and unknown. Recognition is achieved in two stages. The initial learning stage where the model of each object is obtained and stored in terms of the relations between its surfaces. Relations include type of intersection (convex,concave, mixed or no intersection), angle between the regions, and relative positions of the centroids. This is achieved by showing to the system all known objects one at a time. If one view is not enough to describe an object, several typical views are shown and multiple models are made. Then in the matching phase the description of the unknown scene is compared to the models so that the stacked objects are distinguished and recognised.

Ting-jun Fan et al. (Fan 1989) introduced a system which takes as input dense range data and automatically produces a symbolic description of the object in the scene in terms of their visible surface areas and their interrelationships. The representation is made of an attributed graph whose nodes represent the patches and whose links express geometric relationships between patches. A typical model includes a set of 4 to 6 multiple viewing angles. Since the surface descriptions are invariant for smaller changes in viewing angle local changes do not radically alter the representation. Matching between objects and models is performed in three stages. In the first step the most likely candidate view for each object is evaluated. By doing so searching time is significantly reduced. Here, factors such as the number of nodes and visible 3-D area of the largest node are examined. Then a graph matcher performs detailed comparison between the potential matching graphs. At the end an analyser evaluates the results and if necessary proposes alterations such as merging or splitting object graphs. Experiments are performed on composite scenes made of a number of objects.

Kumar S. Ray and D.Dutta Majumder (Ray 1991) made use of principle curvatures, mean curvature and Gaussian curvature to represent local invariant features of the surfaces. Views with partially occluded objects are considered and objects with planar or curved surfaces are included. Several typical views are shown to the vision system and multiple models of an object are made. Based on the matched descriptions of the scene a model hypotheses is generated and its validity is verified.

J. Mukherjee and B.N. Chatterji (Mukherjee 1990) constructed the wire frame structure of a 3-D object from its range data which is a kind of partial transformation from 2 1/2 D to 3-D. The approach is surface based which relies on the extraction and segmentation of nearly planar surface patches. Then by superimposition of vertices by fake edge removal wire frames of polyhedral as well as non-polyhedral objects are obtained.

L. Caponetti et al. (Caponetti 1992) introduced a 3D multi-view description for each object and integrated them in a tree structure data in order to simplify matching. In it the view-points related to similar surfaces are discarded which makes the data structure efficient especially when the objects are well discriminated.

Wen-Nung Lie et al. (Lie 1990) used intensity images to extract corners, edges and line drawings of the polyhedral. Then these are used to derive a set of control points to guide intensity range sensing. By use of both intensity and range information object description in 3-D space is constructed. Recognition is achieved through the interpretation tree where it is compared with a model graph.

2.12 Systems with 3-D Object Representation

Here, a single viewer independent model is used to describe objects. Multi-view feature representation is one type, where a set of descriptions are combined into a single composite model. Extended Gaussian Image (EGI) is an example of this type of modelling where an abstract representation of an object is constructed. For that, surface normals (needle map) are moved to the origin, ignoring positional information. This creates a distribution of points on a unit sphere (Gaussian sphere) where surface orientation can be established. Also, by associating a unit of mass with each point the distribution of mass can be obtained. By constructing a two dimensional table mass for each point, the surface area corresponding to the given viewpoint can be stored.

K. Ikeuchi (Ikeuchi 1981) modelled a 3-D object using a set of normalised EGIs, one for each possible viewing direction. Matching is performed by comparing an observed EGI with each model EGI and examining similarities in their mass distribution. The centres of two spheres are brought together and models are rotated and aligned about a candidate line of sight. The model that maximises mass distribution is chosen as the observed line of sight.

Another description in this class (i.e. 3-D representation) is known as exact representation. In it surface sweep and volume representation is used.

Rodney A. Brooks (1981) outlined a rule-based system known as ACRONYM to obtain 3-D models based interpretation of 2-D images. The initial bottom level extracts edges from the image and groups them into an intermediate representation of elongated structures known as ribbons and ellipses. These are used to predict the appearance of models. Independent Viewpoint, three dimensional models, are represented by the object graph. In it, the shape of the objects is modelled with a combination of the generalised cylinder. Generic models roughly describing the class of object and specialised models describing details of each object are created. It consists of frame hierarchies describing part verses, whole and specific verse general relations between generalised cylinders. The restriction graph describes the spatial constraints between model parts. Then matching is performed locally and globally. Observed features consistent with predicted features restrict interpretation. Then the registered model is validated by looking top-down for additional features. This system has been proved capable of locating aeroplanes in a few aerial images of airports.

2.13 3-D Interpretation of 2-D Images

This group is concerned with recognition with a single two dimensional view. Many monocular depth cues can be exploited to produce three dimensional interpretation of two dimensional images.

David G. Lowe (Lowe 1987) developed a system known as SCERPO. Here a perspective transformation from a model to an image is performed. This is done by searching for sets of model and image features that are consistent from a single view of a solid object.

First a process of perceptual organisation is used to form groupings and structures in the image that are likely to be invariant over a wide range of viewpoints. As an example, features such as nearly parallel lines are grouped together. Then an initial guess is made to map model features into the image by locating additional corresponding points. Finally a process of spatial correspondence brings the projections of three dimensional models into direct correspondence with the image.

R. Horaud (1987) introduced a model based interpretation of the 3-D world from viewing two dimensional images. For this, first, a simple low-level process produces a description of the image and performs line finding. Then lines are grouped into junctions and the angle between them. A group of features produces fewer interpretations than isolated ones and therefore the combinatorial explosion of the search space is reduced. Here, instead of predicting object appearance in the image matching is performed in 3-D space. For this image features are backprojected onto the 3-D space. In this process accumulation space consisting of two angles in a spherical coordinate system is used. The solution that receives the largest number of votes is selected. Recognition is performed for a class of industrial parts with straight edges and planar faces.

Daniel P. Huttenlocher and Shimon Ullman (Huttenlocher 1990) have shown that correspondence of three pairs of non-co-linear points is sufficient to determine a transformation from a three dimensional model coordinate frame to the two dimensional image coordinate frame. Matching is performed by computing possible alignment between local features extracted from the image and wire frame models of objects. Each alignment is verified using the complete model contour by transforming the model edge contours into image coordinates.

D.M. Gavrila and F.C.D. Groen (Gavrila 1992) used a Geometric Hashing technique to index transformation invariant features of the models. Three dimensional models are represented by their 2-D projection so that the problem of reducing the dimension of the image space compared with the model space can be solved. For that the set of viewpoints with equal distance to the centre of the model are taken and their 2D projections are obtained.

Comparable to above Kai Qian et al. (Qian 1992) considered the orthographic projection of 36 views of a model obtained through steps of 30 degree about all the three coordinate axis. Recognition is achieved by the matching of the unknown object photographed from an arbitrary angle and each of the given models. Experimental results showed satisfactory results for models which are not closely similar to one another.

Rong-Nan Chiou et al. (Chiou 1992) used both 3-D surface data and 2D image projection patterns for single polyhedron recognition. The recognition process first matches the partial 3D surface data reconstructed by the three-View system to obtain potential candidates. Then each candidate is backprojected onto the three image planes and verified by matching the outer boundaries.

K.C. Wong and J. Kittler (Wong 1992) described a model based polyhedral object recognition system for identifying the scene-model correspondences from a single perspective image. In the framework, trihedral vertices and their composite with V-junction are employed as key features for model invocation and hypothesis generation.

2.14 Vision System and CAD

The main goal of CAD (Computer Aided Design) packages is to design new shapes suitable for automatically manufacturing objects. Most of the CAD based systems rely on CSG (Constructive Solid Geometry) to represent objects. In it, objects are represented in terms of simple bodies which can be added and subtracted from one another. In this approach models are volumetric primitives. Because of this they cannot be easily reconstructed from the surface information. In vision systems, on the other hand, usually the range data is applied to obtain data from the surfaces. Because of this, there has been a dichotomy between these two areas.

Bir Bhanu and Chih-Cheng Ho (Bhanu 1987) proposed a CAD based approach for building multiple representations of objects. These included representations such as surface normal, surface curvatures, generalised sweep, etc. First object geometry is designed using a CAD modelling system. This is done by utilising spline-based boundary representation. Then various representations are developed from that CAD model. For instance the surface normal is a byproduct of the surface evaluation procedures in a spline-based system. These higher level descriptions of the object can be used to communicate with other vision modules. Also it can possibly be employed in applications such as 3D object recognition.

C. Hansen and T. Henderson (Hansen 1989) devised a method in which, by giving the CAD model of an object, a specific, tailor made system to recognise the object is synthesised. This includes automatic significant features selection. Here a complete original set of features of an object are filtered for qualities such as rarity, robustness, cost, etc. For instance, rare features are useful for quickly identifying the object and making a good root node in a search tree. Next a search strategy is conducted for the strongest features and how their presence in a scene constrains the remaining search. Then, based on this hypothesis recognition and matching on a single polyhedral object is performed.

Rongxing Li (Li 1992) generated various representations such as Boundary Representation (B-rep) , Constructive Solid Geometry (CSG) and octree for the CAD/CAM systems. By interfacing digital data and object representation, geometric models useful for different applications are obtained. This is applicable for the cases where geometric information such as the dimensions and shapes of the objects are not readily available.

2.15 Recognition by Superquadrics

Superquadrics are families of parametric shapes that are used as primitives for shape representation and recognition. They are an extension of basic quadric surfaces (Gardiner 1965). These prototypes can be deformed by linear stretching, bending etc. and then combined to form new complex shapes. These are used along with parameters representing position, orientation and size. For this the superquadrics are a super set and generalisations of the Constructive Solid Geometry (CSG) modelling primitives.

Alex P. Pentland (Pentland 1987) argues that most natural objects have a part structure that can be recovered from image data and used for general purpose recognition. For this superquadrics as a basic modelling element can be introduced. Then, the recognition process is reduced to one of search and optimisation. In this process, the goodness of fit function is employed to evaluate and establish the best matches. The optimisation problem is nonlinear, and iterative techniques are applied to converge on local minima. Experiments are performed using syntactic as well as range data. As a consequence of the concise nature of this shape language, even complex objects are modelled using a relatively small number of parameters.

F. Solina and R. Bajcsy (Solina 1987,1990) used superquadrics to model and represent single part objects in the scene. Here an energy or cost function is defined to evaluate performance of various parameters. This is known as the inside outside function and one of its parameters depends upon the proximity of points to the model's surface. A least square method is used to minimise the function. In the experiments that follow range data is employed to recover regular as well as irregular shape objects. Here it is found that, the adopted shape vocabulary is suitable for a rough description of the parts. Because of this, it can only be employed for recognition of basic categories of objects. As a conclusion it is stressed that superquadrics are just a handy mathematical representation of prototypical parts that can be made to fit the actual shapes.

2.16 Survey Papers

In this section a number of papers related to previous research work in this subject are mentioned.

Rafael C. Gonzalez and Reza Safabakhsh (Gonzalez 1982) examined those aspects of computer vision relating to industrial applications. Here topics such as segmentation, edge detection, thresholding, region growing together with recognition are discussed. Most of the contents are related to two dimensional industrial parts. However, topics such as illumination of the scene and three dimensional objects are also included.

Paul J. Besl and Ramesh C. Jain (Besl 1985) presented a comprehensive study of object recognition. Here the main problems involved in the subject are outlined and basic definitions and concepts associated with it are discussed. Then various methods of representing objects and surfaces are briefly mentioned. Next, topics such as intensity and range image processing and surface characterisation are reviewed. Finally, different algorithms related to object reconstruction and recognition are analysed.

Roland T. Chin and Chares R. Dyer (Chin 1986) explored model based object recognition algorithms for robot vision. It is arranged according to 2-D,2 1/2D and 3-D object representation. In 2-D, models can be automatically constructed from a set of prototype objects, one from each possible viewpoint. In 2 1/2D models use viewer centred description of the surfaces instead of boundaries. In 3-D representation the models consist of the most general and complete description of objects from an unconstrained viewpoint. Three central issues common to each category, namely, feature extraction, modelling and matching, are examined in detail.

J. P. Brady et al. (Brady 1989) investigated 3-D object recognition by range data. First various methods of acquisition of range data are mentioned. Next the processes involved in an adequate symbolic representation from a depth map are discussed. Then the various edge or region boundary types such as jump boundaries or curvature discontinuity employed on range data are specified. Next, problems involved in feature extraction and model building are argued. Then, some significant research approaches, together with their results are briefly discussed. The survey contains subjects such as the primal sketch approach, curvature-based representation and polyhedral models and matching.

2.17 Summary and Comment

In this chapter a brief description of various research work related to object recognition is presented. These are grouped according to the shape of the objects, technique of recognition or representation of the models. By doing so the important aspects of each approach are emphasised and classified with similar relevant works. This is done bearing in mind that some of the approaches can be classified in more than one group.

For many recognition methods extracting common features of the objects such as vertices, edges or surfaces is sufficient as in (Mukherjee 1990) , (Wong 1991) and (Chiou 1992). However, some approaches as in (Ikeuchi 1981) elaborate and abstract representation of the object (i.e. as in Extended Gaussian Image) are needed. Also, depending on the shape or recognition method obtaining some extra information may be necessary. As an example for curved objects fitting quadratic equations as in (Hall 1982) or obtaining surface curvatures as in (Besl 1986) and (Ray 1991) may be required. Equally for the objects with no regular geometrical shape as in (Faugeras 1986) and (Stein 1991) approximating surface by sets of planar or smooth curved faces may become necessary. Obtaining direct three dimensional coordinates as in (Oshima 1983) can be a powerful means to construct features but monocular images as in (Lowe 1987) can also reveal important clues and information about content of the scene.

To proceed with recognition the models of the objects need to be available as in (Horaud 1987) or constructed such as in (Fan 1989) or (Gavrila 1992). Otherwise recognition would not have any meaning if shapes or characteristics of the objects are not known. Some circumstances as in (Qian 1992) or (Caponetti 1992) may require several viewer centred models taken from different directions. This would need an assumption that the scene would remain the same if the camera view changes to some degree.

One way to proceed with the matching and find the correct model is through the interpretation tree as in (Lie 1990). Also it is possible to accumulate independent pieces of evidence until the right model emerges as in (Ballard 1983). In either case recognition depends on how fine essential features from the scene are extracted so that related matches can be found and established. In matching various features between objects and models large amounts of computation may be necessary. The challenge in this field is to avoid unrelated computation and truncate unnecessary processing as in (Dhome 1987).

Object recognition involves many diverse methods and approaches in analysing the scene and solving problems. This indicates that the subject is very fertile for research and open to further exploration.

CHAPTER THREE

DESIGN AND CALIBRATION OF THE VISION SYSTEM

Basically it is not possible to extract range data from a single image. But if two or more cameras are used then three dimensional information can be obtained. If one of the cameras is replaced by a specific pattern of controlled light then range data can be directly computed. Patterns that are normally employed are the dot, line and grid. Next, various factors concerning the design of a vision system are specified. Then, equipment requirements together with the calibration of the camera are discussed. At the end the calibration of the X,Y, and Z axis, together with their practical procedures, are described.

3.1 Range Data from a Monocular Image

Consider a camera with focal length f and lens centre F, and the image plane p as illustrated in Fig.(3.1). Here, (x,y,z) and (X,Y,Z) represent the image and world coordinate systems respectively. The two coordinate systems are positioned such that their axis are aligned together. Suppose point A with the world coordinates of (X,Y,Z) is captured at point a with the image coordinate (x,y). In order to obtain image coordinates (x,y) in terms of world coordinates (X,Y,Z) the following relations can be written,

(3-1) $$x/f = X/(Z-f)$$

(3-2) $$y/f = Y/(Z-f)$$

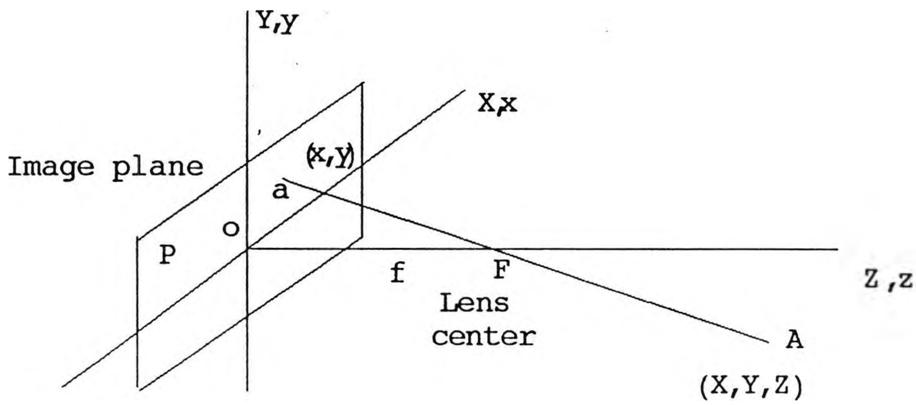


Fig. (3.1) Projection of point A onto the image plane. World and image coordinate are aligned.

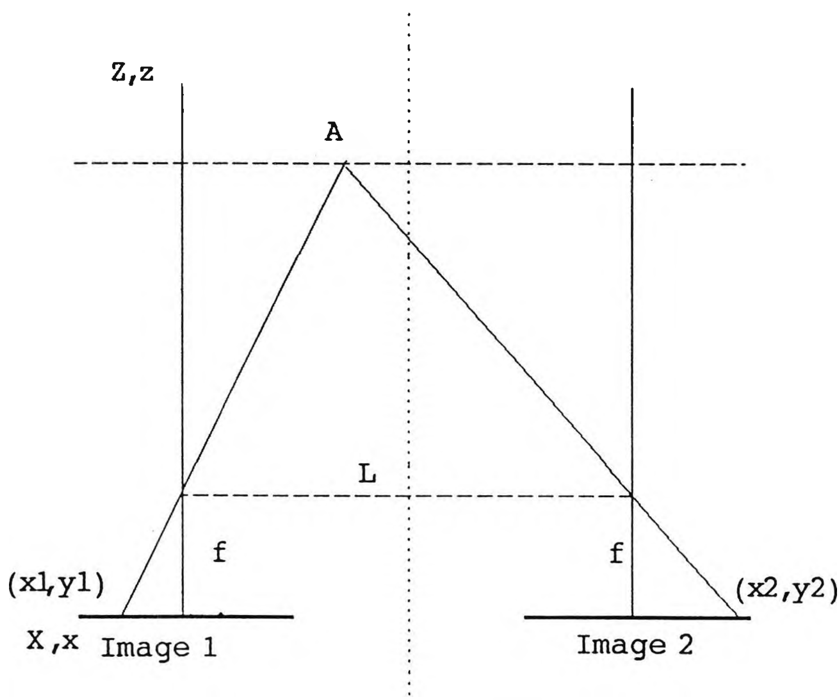


Fig.(3.2) Stereoscopic view of point A. World coordinates first aligned with the coordinates of image 1 and then with image 2.

where from these relations x and y can be easily derived. But, since these equations are divided by variable Z, they are nonlinear. This implies that all points along FA and with the value Z > f can make the same spot as point 'a' on the image plane. In other words mapping a 3D scene onto the image plane is 'a many to one' transformation. Therefore if some knowledge such as Z is not available beforehand, the position of an arbitrary point in the world coordinates cannot be established. If, however, some knowledge is available about the scene then surface orientation and the shape of the objects may be deduced from the monocular image. According to constraints on the scene and input information these are known as shape from shading, shape from shape, and shape from texture (Shirai 1987). R. K. Singh and R.S. Ramakrishna (Singh 1990) considered shadow geometry together with the texture to compute surface orientations for the scenes involving polyhedra.

3.2 Range Data from Binocular Images

Consider Fig (3.2) where two cameras with the same focal length f are illustrated. Here,the centres of the two lenses (i.e. the base line) are separated by L. Also, the coordinate systems of both cameras are considered to be aligned and differing only in the location of their origin. Suppose the image coordinate system of the first camera coincides with the world coordinate system as in Fig(3.1). Then the relation (3-1) can be rewritten as follows,

(3-3) $$X_1 = x_1(Z_1 - f)/f$$

The subscripts for X and Z are added in order to specify that the world coordinate is coincident with the first image coordinate. Then the same can be written if the world coordinate system is taken into the second image coordinate.

(3-4) $$X_2 = x_2(Z_2-f)/f$$

because,

(3-5) $$Z_2 = Z_1 = Z$$

(3-6) $$X_2 = X_1 + L$$

relations (3-3) and (3-4) can be rewritten as follows,

(3-7) $$X_1 = x_1(Z-f)/f$$

(3-8) $$X_1 + L = x_2(Z-f)/f$$

By subtracting these two relations, Z can be calculated as follows,

(3-9) $$Z = f - (fL)/(x_2 - x_1)$$

Suppose the baseline, and focal length and the difference between the positions of two corresponding elements in the two images is known. Then calculating Z or the third dimension would be a matter of substituting those values into relation (3-9). Once Z is established X and Y can be obtained from (3-1) and (3-2).

The calculation involved in extracting three dimensional information in binocular vision or stereo vision is known as passive triangulation. The difference between two correspondent elements is known as disparity. As is shown in equation (3-9), depth is inversely proportional to disparity. A point such as $a_1(x_1,y_1)$ on image 1 corresponds to a line of sight that passes through the optical centre and a_1 itself . The image of this line in image 2 (i.e. camera 2) is the epipolar line associated with a_1. For manufacturing applications where the object is in constant motion line-scan cameras in a stereoscopic arrangement can be used (Godber 1992). This has the advantage that two dimensional images can be generated according to the specific application and not locked to any particular standard as with television type cameras. The general discussion on stereo vision can be found in, (Gonzalez 1987),(Shirai 1987) and (Boyle 1988). For detailed analysis on binocular and trinocular vision systems (Ayache 1991) can be consulted.

3.3 Range Data and Structured Light

Consider the case where a source of light and a slit replace image 1 in Fig.(3.2). This is illustrated in Fig.(3.3) and Fig.(3.4). Here, the position of the light is determined by the rotating slit in front of the light source. For simplicity it is assumed that the location of y_1 (i.e. in the previous position of image 1 in section 3.2 above) and y_2 in image 2 remain the same. Then x_1 can be determined from the direction of the light stripe. Also the position of x_2 can be obtained from the observed stripe image. As a result the disparity required to find Z can be easily established.

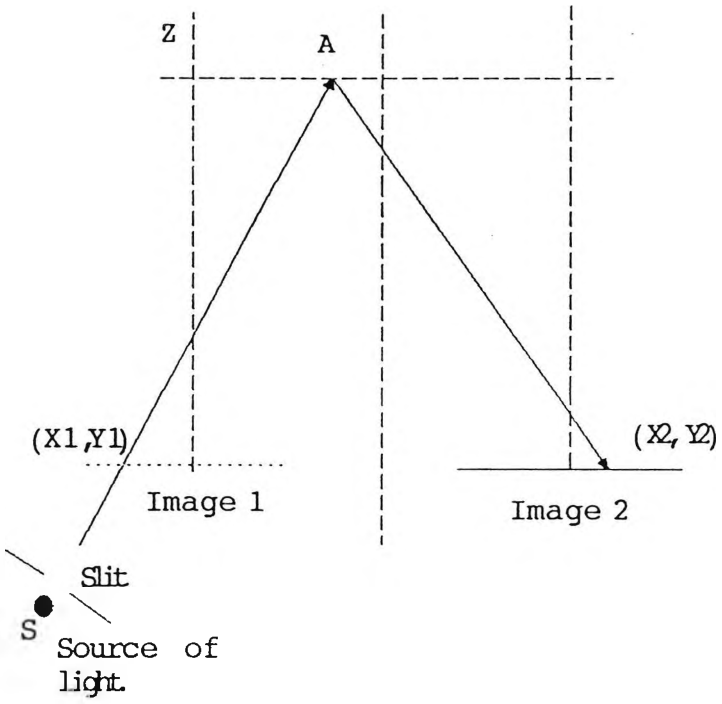


Fig.(3.3) Replacement of image 1 with light source and a slit.

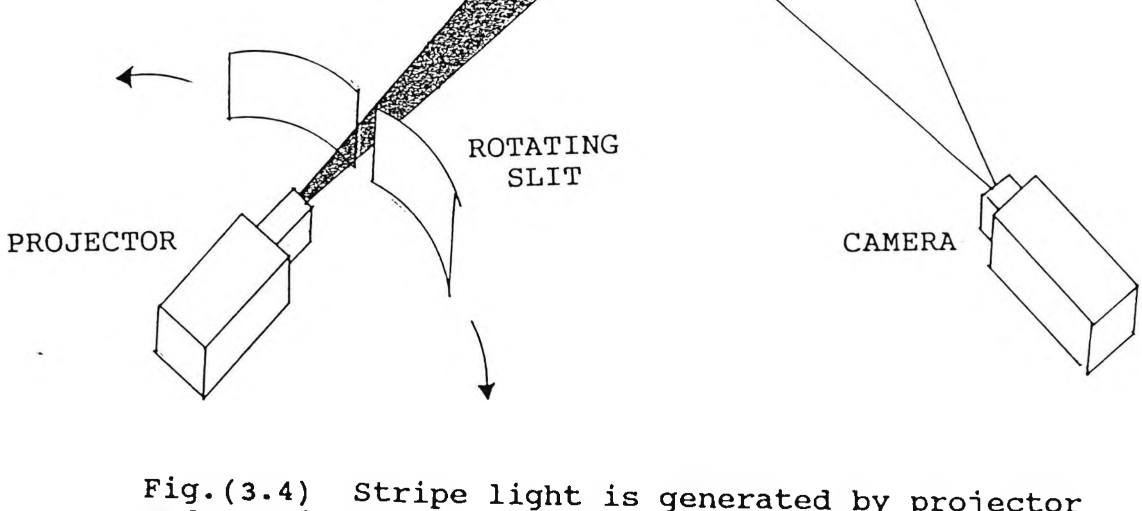


Fig.(3.4) Stripe light is generated by projector and a slit. Images are captured by camera. Object is scanned by rotating slit.

The principle of range data calculation is the same as stereo vision. The difference is that the camera is replaced by the projector which illuminates the scene with a specific shape of light. The calculation involved in extracting three dimensional information in a structured light technique is known as active triangulation. Suppose the light stripe is used to illuminate the scene as shown in Fig.(3.4). Also suppose i to be the light-stripe in its ith projection and j denotes the vertical position in the stripe image. Then the position of the stripe in the image corresponds to the position of the surface that reflects the light beam (calculations as in section 3.2). To reconstruct the scene the whole field of view needs to be scanned. The data structure obtained in this way is known as a range image.

3.4 Projected Patterns

In a structured light technique, in order to capture the range data of the scene, it is required to project a specially prepared pattern of light onto the objects. The patterns that are frequently used for the purpose are the dot, line and grid as shown in Fig(3.5).

One of the simplest patterns is a single dot. For this a spot beam is projected onto the object and its reflection is detected as illustrated in Fig(3.6). Then by triangulation methods its coordinates are calculated. One of the ways to scan the scene is to use a rotating mirror. This is employed by M. Ishii, and T. Nagata, (Ishii 1976). They also used an image dissector to scan and partition different parts of the image in order to detect the position of the spot at any one time. One other way to scan the scene is to use a rotating table which can move vertically up and down as illustrated in Fig(3.6). This is used by Faugeras O.D. et al.(Faugeras 1982). Objects are placed on the surface of the table and the reflected laser beam is detected by the detector.

Covering and scanning the whole surface of the objects by projecting a dot can take considerable time. Martin D. Altschuler et al. (Altschuler 1979 and 1981 and 1982) devised a mapping technique that in a short space of time can determine the

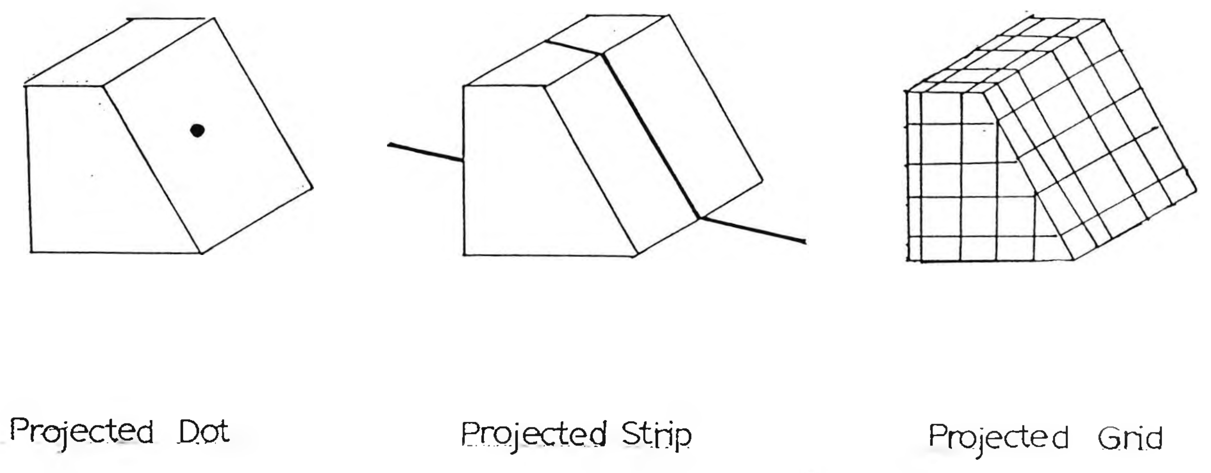


Fig. (3.5) Various projected patterns into the object.

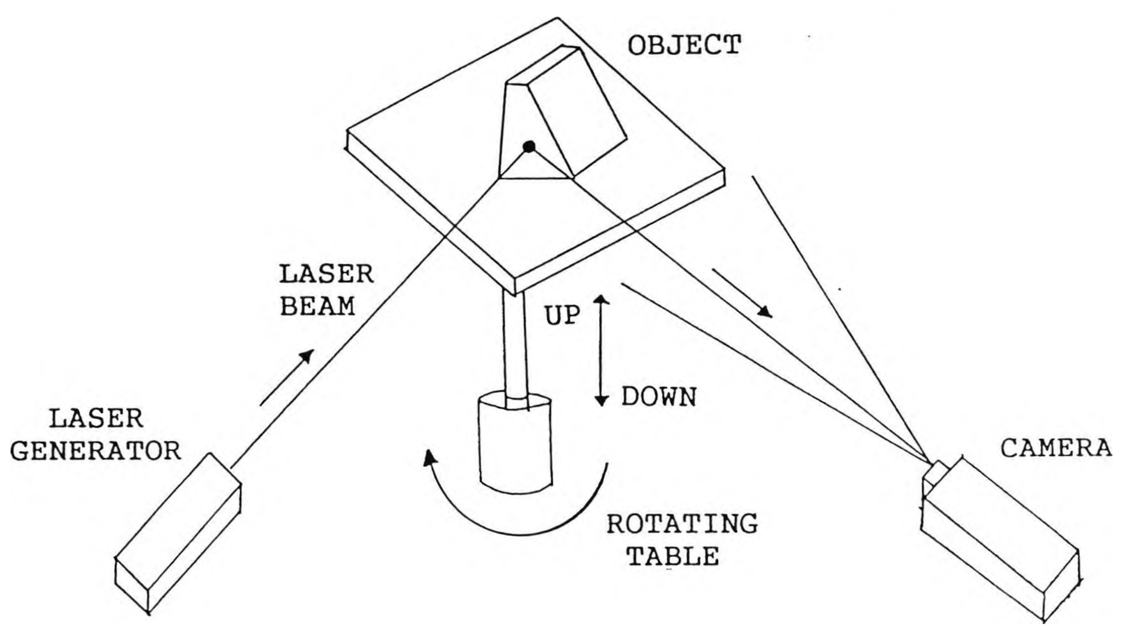
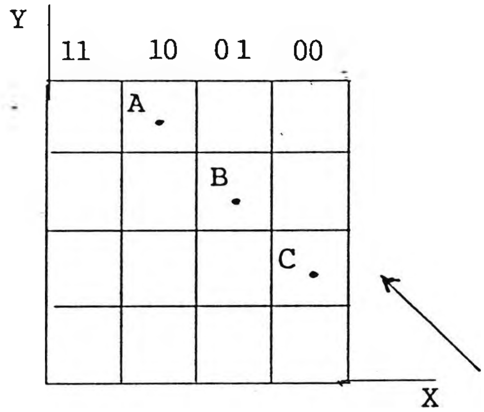


Fig. (3.6) A laser beam is projected onto the object and captured by a camera. Object is scanned by rotating table.

height of a surface. First they used a lens and shearing plate assembly to expand and partition the primary single laser beam into a two-dimensional array of N by N separate laser beams. Then a space coding method is used to identify each dot. This is illustrated in Fig(3.7) where a four by four laser mesh is demonstrated and three A, B, and C points are specifically labelled. Here, by taking only three images, all three points are identified. This is achieved by first taking an image with all four columns illuminated. Then the second image is taken while the last two columns are obstructed and then the third image with even numbered columns covered (i.e. second and fourth columns). Then by performing a few adding and subtracting operations each point is distinguished individually. Recently, solid state devices have been made for spot position detection. An example is a PIN diode position sensitive chip that directly outputs the x and y coordinates of the spot. Typical systems detect the spot position with the speed of 1000 points per second. D.D. Harrison and M.P. Weir (Harrison 1990) described improvements to the triangulation range-sensing technique that increased data collection rate using charge-injection-device image sensor.

The next pattern is a line. The advantage with the line is that a sharp discontinuity or jump boundary can suggest an edge in the object. Also a change in the height can result in the displacement of the line which in turn can be used to calculate coordinates of the objects (see section 3.10). Also if range data is required over a full field of sight,the light stripe is most effective. With the line it is possible to adjust the number of scans necessary to obtain an accurate picture of the scene. Shirai and Suwa (Shirai 1971) developed a system which employed a light stripe and used rotating mirrors to scan the scene. Scanning the scene with the stripe of light and then processing each image taken in each scan can be quite time consuming. K. L. Boyer and A. C. Kak (Boyer 1987) devised a system which could rapidly acquire a range map of the scene using colour encoded structured light. They used vertically striped patterns of colours from a slide projector to illuminate the scene. The image on the slide installed in the projector is just a set of vertical, coloured stripes, which are uniquely subpat-



Diagonal view of A,B,C

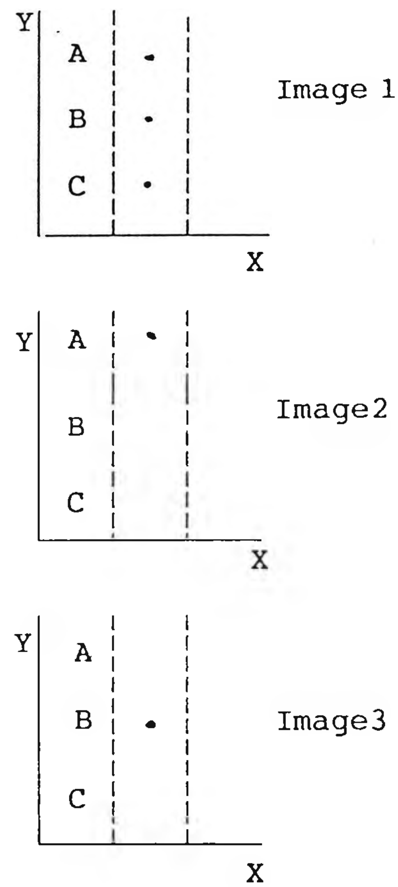


	Image1	Image 2	Image3
A	1	1	0
B	1	0	1
C	1	0	0

Fig.(3.7) Projected A,B,C laser points are identified by three images only. For image 1 all columns are illuminated. For image 2 column 11 and 10 are used. For image 3,column 11 and 01 are illuminated terned horizontally. Then by taking just one image, Red, Blue and Green camera outputs are processed to locate their peaks along each line. Also Paul M. Griffin et al. (Griffin 1992) used a matrix of coloured circles for generating an encoded coloured light pattern for use in obtaining range data from a scene. P. Vuylsteke and A. Oosterlinck (Vuylsteke 1990) used a coding scheme in order to identify individual lines. In it every sample point indicated by the light pattern, is made identifiable by means of a binary signature.

In the cases where a small number of points are adequate to construct the scene the grid pattern can be projected. Here, it is only required that one image be taken. However extracting and labelling the grid points and matching grid junctions under the stereoscopic principle are part of the process that needs to be carried out. Ernest L. Hall et al. (Hall 1982) used grid patterns to extract points from the curved surfaces and then fit quadratic equations on them. Y. F. Wang et al. (Wang 1987) used the direction of the projected stripe to infer local surface orientation. This does not require any correspondence relation between grid junctions in order to extract depth information. Also Y. F. Wang (Wang 1991) used a projected grid pattern to extract intrinsic surface properties. These are not affected by the choice of a coordinate system and the position of view and are ideal for representation and recognition.

3.5 Vision System and Design strategy

In the design of the present system the main objective is to recognise several different objects from one another. Objects are expected to have plane surfaces only. Some of the objects are expected to be more complicated than simple wedges or pyramids. Also algorithms to be implemented will require coordinates of high level features such as edges. Therefore obtaining range information is one of the main requirements that need to be accomplished. For that the following options are considered:

1) Monocular imaging.
2) Stereo vision.
3) Structured light techniques.

As is mentioned in section 3.1, surface orientation and the shape of the objects can be deduced from a monocular image. However, since complete range data cannot be extracted and not enough information is available from the scene this option is not considered.

In stereoscopic techniques, however, the required range data can be computed. But here, extracting features from the images relies on the abrupt change of intensity around edges of the objects. This can result in the omission or extraction of incomplete pieces of edges especially when more complicated objects than simple polyhedral objects are involved. Also certain edges or surfaces visible from one camera could be occluded from the other one. Furthermore a shadow present in one view may not be there from another direction. Also the same surface can have different image characteristics due to the lighting effect. In addition to that finding correspondent pixels and matching is one of the prominent problems associated with this technique. This may require considerable time and much computational processing to overcome. Also, the accuracy of disparity depends on the camera separation which in turn can result in having a limited overlapping field of view.

In structured light techniques, on the other hand, a break in the shape of the stripe can indicate the presence of an edge. Because of that, edges can be completely and reliably extracted. In addition to that, because the projected light can be controlled, the geometry of the scene can be accurately established. Also coordinates can be calculated using the distance between the camera and the source of light relative to the position of the scene. That is a more straightforward approach than finding correspondent pixels in a stereoscopic technique.

As a result of these comparisons it was decided to use projected light to extract information from the scene. From the patterns that are discussed, the dot and grid are particularly suitable for the cases where the surfaces of the objects need to be extracted. That is because the equation of a plane can be easily fitted to them. For extraction of edges, however, projected line techniques are most suitable. Since in the recognition algorithm to be used, coordinates of edges are required, the later option is selected.

In addition to the methods discussed above there is another range finding technique which is not considered. That is the time of flight technique which involves using light or ultrasonics as a medium to transmit signals. Here, the depth information is obtained from the time required to send a signal from the transmitter to the target and back. These techniques are mainly exploited in areas such as radar and obstacle detection. These can provide good quality data but require electronics with a very fine resolution. Because of that they are not particularly suitable for tasks such as object recognition. An overall review on various available options and different range finding techniques can be found in (Strand 1984), (Mcfarland 1984) and (Bastuscheck 1989).

3.6 Required Equipment

One of the central issues in using a line to illuminate the scene is to solve the problem of scanning. One way of doing it is to use a rotating slit as shown in Fig(3.4). This is used by Shirai and Suwa (Shirai 1971) with a computer controlled stepping motor. Here an image is taken for every direction of the light. In the present system the scanning table is used to move objects under the field of view. This is done while a projector is positioned overhead projecting a stripe of light perpendicular to the scene and the direction of motion of the table. The camera with its tripod is positioned on the side of the table taking images of the scene as shown in Fig.(3.8).

Scanning Table: The surface of this table is designed to move in two directions X and Y. This is done via instructions given from a computer and received via RS232 interface. It is capable of a precision of up to 1/1000 of a millimetre in each direction. In the scanning of the objects this very high level of precision is not required. For that, it is arranged to position it approximately 0.5 mm from where it should stop. As an example, suppose starting from position X = 0.0 it is moved forward and a scan is taken every one millimetre (in the direction of X). Then at the sixth scan, it would stand between 5.5 mm and 6.5 mm. That is achieved by adjusting its position on each scan until it complies with the 0.5 mm error range. The specifi-

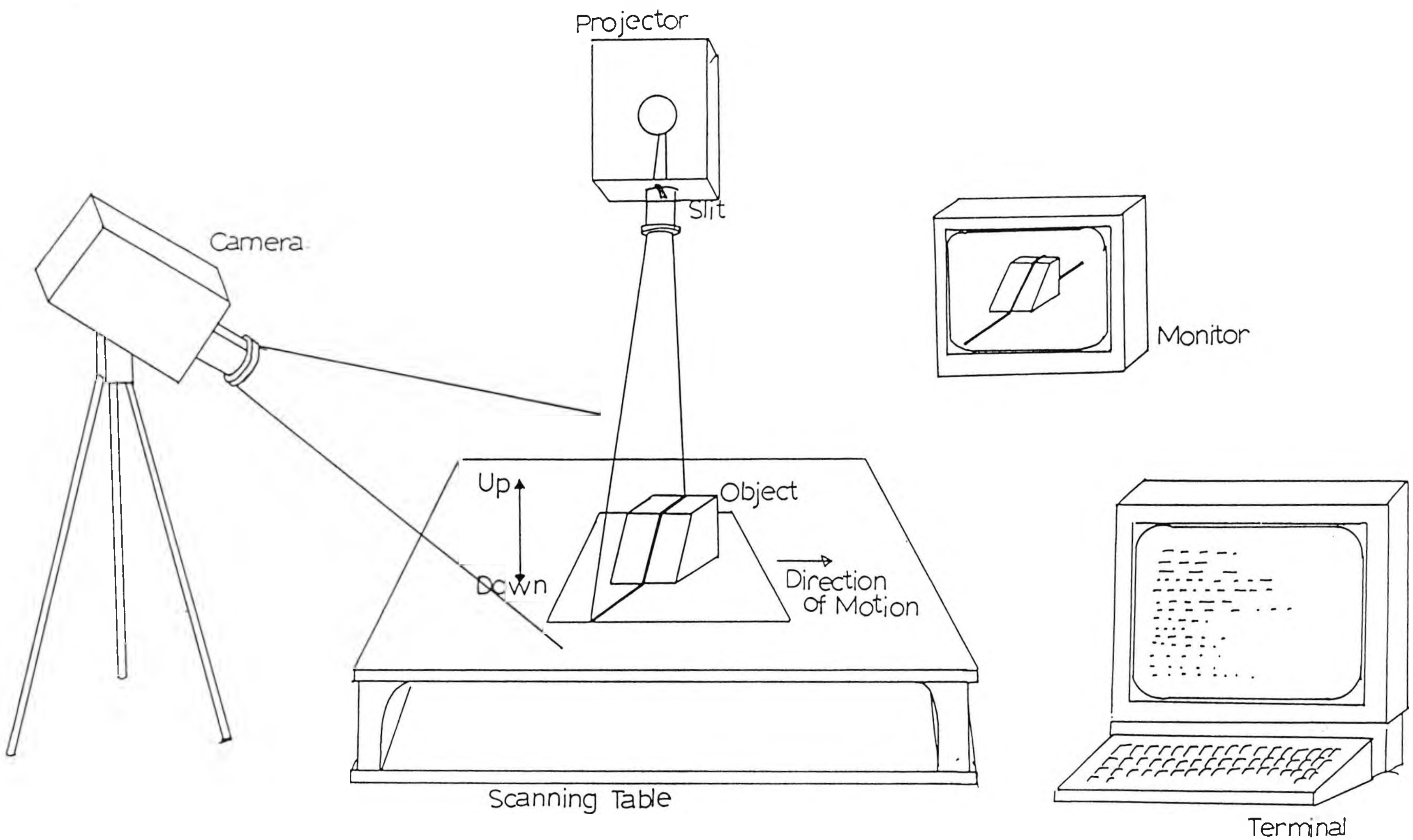


Fig.(3.8) Diagram of the system and all the equipment .

cation of the table can be found in appendix 1.

Camera: The camera that is used is a typical charge coupled device (CCD) solid state camera. These devices consist of arrays of photosensitive elements integrated in a chip. As each one of its elements come into contact with the light, a signal is generated. The strength of the signal is determined by the intensity of the light. The signal itself is divided into 256 levels representing various grey levels and the images are 512 by 512 pixels.

Projector: In order to illuminate the scene, laser light can be the most effective. This is primarily because a laser beam does not diverge with the distance. As a result it does not lose its initial shape and sharpness as it reaches the target. Also it is intense enough to be recorded by the camera. However in the absence of such a device, a slide projector is used as an alternative to the laser beam. In a slide projector ordinary light is used. Because of that it requires focusing, otherwise a defocussed pattern with reduced sharpness results. Since the height of the objects constitutes a change of the distance, it is hard to maintain a focused projected pattern all over the object. Because of that, special attention is required in order to adjust the light beam.

The slide: A sharp and preferably long line is what is required on the surface of the scene. For that, two aluminium pieces of metal are placed on the frame of the slide. They are positioned closely together so that a straight narrow gap between them is produced. This produces the desired sharp stripe of light. Using an aluminium sheet is particularly advantageous since it can be easily cut. Also it can effectively prevent the rest of the light from reaching the surface of the table. Because of that the contrast between the stripe and the scene can be increased and enhanced images of the stripe obtained.

Computer: A 386 PC computer with a speed of approximately 26.5 - 27 MHz is used to run programmes. It is attached to the framestore board in order to grab and process the images. Programmes are written in Turbo Pascal.

Monitor: In order to assess the state of the stripe and its position within the image it is necessary to directly display it.

For that a monitor is connected to the framestore so that the stripe and content of the scene can be directly followed.

3.7 Setting up the Camera ,Projector and the Table

Before the experiments start, the positions of the camera, projector and the surface of the table are fixed. This is done so that the best possible combination between them is achieved. For this, the following steps are taken:

1) The projector is fixed perpendicular to and above the table (i.e. the ray of light perpendicular to surface). Its distance from the surface of the table is adjusted so that a reasonably narrow stripe of light is obtained.

2) Then by moving the surface of the table up and down or by adjusting the position of the projector itself the narrowest possible stripe is produced. The stripe in its narrowest form can be sensitive enough to extract edge points, especially at corner points. For that it is used on top of the objects. To arrange that, an object is placed on the surface of the table. Then the surface is lowered down sufficiently enough so that the sharpest part of the stripe crosses the top of it. This position of the surface of the table is defined as zero for the Z coordinate.

3) The position of the projector is re-adjusted so that, the stripe is placed perpendicular to the direction of motion of the table or X axis as shown in Fig(3.9). That is achieved without altering the distance between the projector and the table which is already set. The line positioned along the stripe is defined as the Y axis.

4) Then the camera is positioned from the side as shown in the Fig.(3.8). The image of the stripe is arranged to be roughly in the middle part of the screen crossing it from top to the bottom of the monitor.

5) Suppose the surface of the table is moved up or down. Then the image of the stripe would shift to the left or to the right of its original position on the screen. This change of position due to the change of height, is exploited as a basis for calibration of the Z coordinates. Because of that, the position of the camera is readjusted so that these shifts occur as large and widely as possible. This way, two consecutive Z values can be

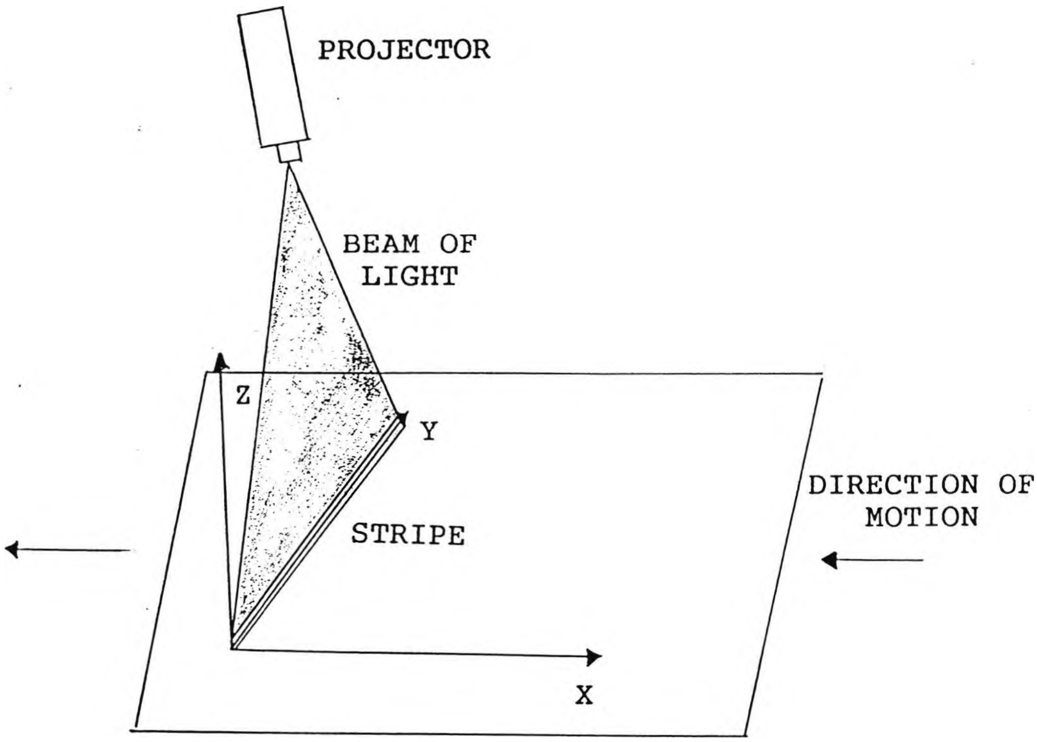


Fig.(3.9) The Y axis is positioned along the stripe and perpendicular to direction of motion of the table (or X axis).

better distinguished. This is achieved by changing the angle and distance of the camera from the table until a reasonable position is found. Too low a camera position with respect to the surface of the table makes it difficult to capture stripes. Also, placing it too near to the projector (in terms of angle) makes consecutive shifts for Z values too close together.

6) The camera lens is focused so that the image of the stripe emerges in its clearest possible shape. To extract the necessary information from the image it is necessary to threshold it. This converts it into the binary image where the stripe would be in white and the rest of the image in black. For this it is necessary to adjust the aperture so that a suitable thresholding level can be established. Letting too little light into the camera can result in having a faint image of the stripe which can make it difficult to threshold. Similarly letting in too much

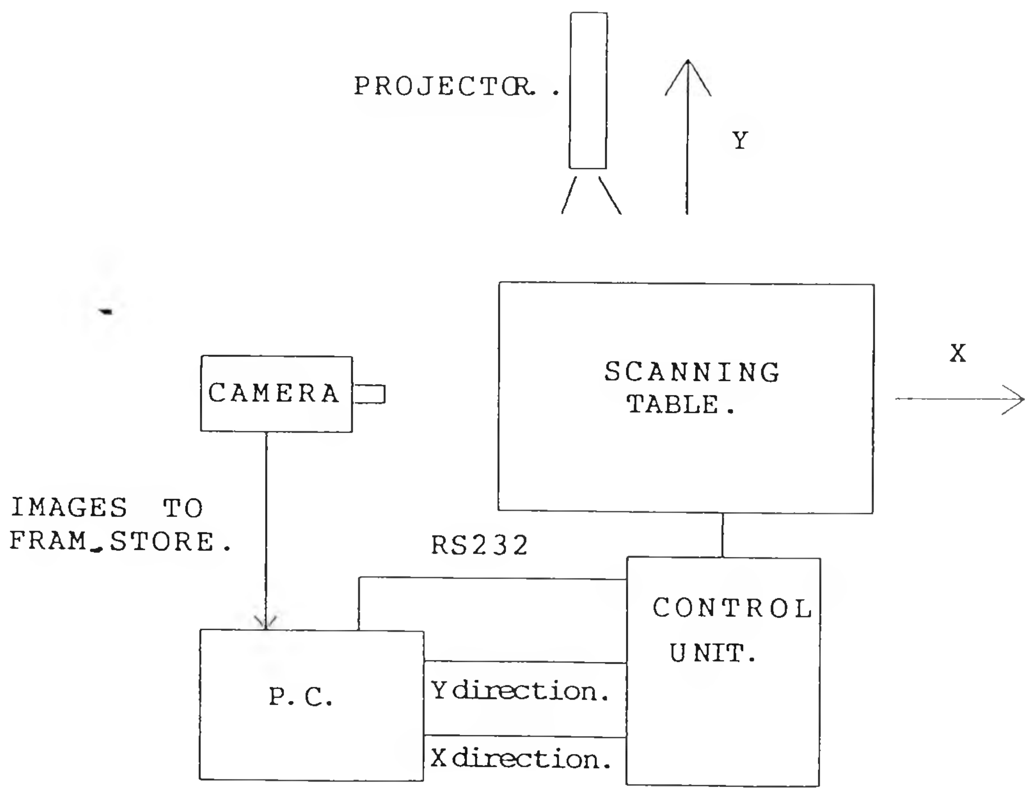


Fig.(3.10) The circuit of the system.

light can make it difficult, or impossible, to distinguish it from the background.

By this stage all items of equipment are placed in their respective positions and ready to use. The complete circuit of the system is shown in Fig.(3.10). The next immediate step is to calibrate the system.

3.8 Camera Calibration

By calibration it is meant to establish the relationship between two dimensional points in the image and their corresponding three dimensional values in the real world. A comprehensive survey of camera calibration can be found in (Tsai 1986). One of the methods which is used to calibrate cameras is based on a Pin hole model. In it, the transition between world coordinates and camera image coordinates is achieved through perspective transformation. Min-Hong Han and Sangyong Rhee (Han 1992) made use of a circle of arbitrary radius with one internal dot at the centre to indicate the centre position and the other inside the circle near the perimeter to provide directional information. Then depending on how the circle changes to the ellipse and how the position of the dots vary, different parameters of the camera are established.

One way to calibrate the cameras is to use a set of image points whose world coordinates are known (Hall 1982). As an example, suppose (X,Y,Z,1) to be the world coordinates and (x,y,h) to be its image coordinates in a homogeneous representation. In order to relate them the following relation can be written.

$$\text{(3-10)} \quad \begin{vmatrix} x \\ y \\ h \end{vmatrix} \begin{matrix} = \\ = \\ = \end{matrix} \begin{vmatrix} a & b & c & d \\ e & f & g & k \\ l & m & n & p \end{vmatrix} \begin{vmatrix} X \\ Y \\ Y \\ 1 \end{vmatrix}$$

By multiplying this matrix and substituting h, the following two equations can be obtained,

(3 -11) $\quad aX + bY + cZ - lX - mY - nZ - px + d = 0$

(3 -12) $\quad eX + fY + gZ - lX - mY - nZ - px + k = 0$

These contain 12 unknown coefficients. To solve them, at least 6 (non-coplanar) world points with their corresponding image coordinates need to be obtained. The optimal solution can be found through simultaneous linear equations (Noble 1969). As discussed in section 3.1, the relation between three dimensional world coordinates and the image plane is 'a many to one' map. But here actually a relationship between points in the image (or an area of the image) is constructed. Then through this relationship three dimensional coordinates can be obtained. Because of that, if unrelated coordinates are given in the above equations, their solution can lead to singular matrices.

In the present system, however, calibration is achieved through dividing images into different zones. This is done so that only specific ranges of the Z and Y coordinates can be found in each region. This explicit transition from two to three dimensions becomes possible, because of the use of controlled light. Once a point is found in one of these areas its coordinates can be calculated. The main advantage of this approach is that no knowledge of the distance between the camera and the source of light or surface of the table is required. The only direct measurement involved is in the various levels of the Z coordinate. Because of that the system can be calibrated fairly easily and in short time.

3.9 Calibration of the X Coordinates

In defining the X,Y and Z axes, the X axis is chosen to be in the direction of motion of the table. The table is set to move one millimetre on each scan. This is used to determine the X coordinates. The position of the table can fluctuate between half a millimetre above or below its actual scanning number. For this the error rate associated with each X can be considered to be in the same region of half millimetre. By choosing the X axis along the direction of motion of the table X remains independent of the coordinates of the image.

3.10 Calibration of the Z Coordinates

The image of the stripe is formed along an optical axis which is at an angle with the ray reflected from the surface of the table. This angle is known as the parallax angle and is shown in Fig.(3.11). More details of it can be found in the work of J.L. Mundy and G.B. Porter (Kanade 1987). Suppose the surface of the table is moved up or down from its original position (i.e. the Z coordinates is changed). Then the image of the stripe would shift to the right or to the left of its previous position (depending on the position of the camera). This shift of position due to the change of height or parallax angle is used as a basis for the calibration of the Z coordinate. This is demonstrated in Fig(3.12) where lines obtained from different positions of Z are illustrated. As an example suppose line 1 is taken while the surface of the table is at its lowest position. Then lines next to it (i.e. lines 2,3 4..) would be obtained from positions where the surface of the table is gradually taken to higher levels.

Based on these observations the following steps are taken in order to calibrate the Z coordinate:

1) To break up the image into the separate zones so that each part can only contain a specific range of values for the Z coordinates.
2) To determine the location of any arbitrary point with respect to zones and to establish in which area it is located.
3) Once the appropriate zone is located, to divide it into further parts so that the value of Z can be accurately estimated.

An example for the third step is shown in Fig.(3.13). Here each section can only hold a specific range of Z coordinates. For that, if a point is found within one of those parts then its value can be estimated. Now suppose point p is situated inside one of the zones and the objective is to determine in which area it is located. For that a horizontal line is drawn and its intersection with the other lines is found (i.e. a,b.c.. in Fig(3.12)).

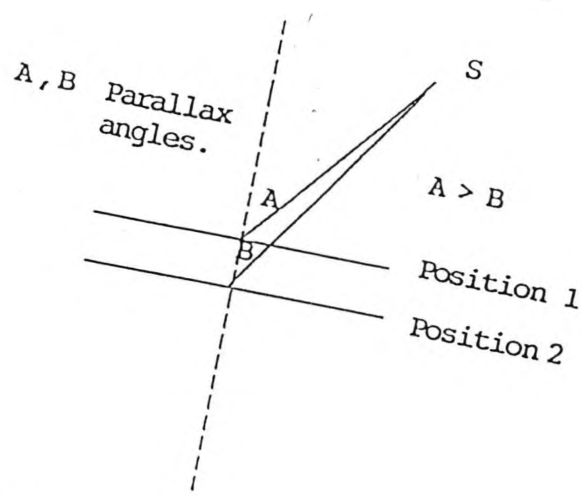


Fig.(3.11) Parallax angles obtained from two positions of the surface.

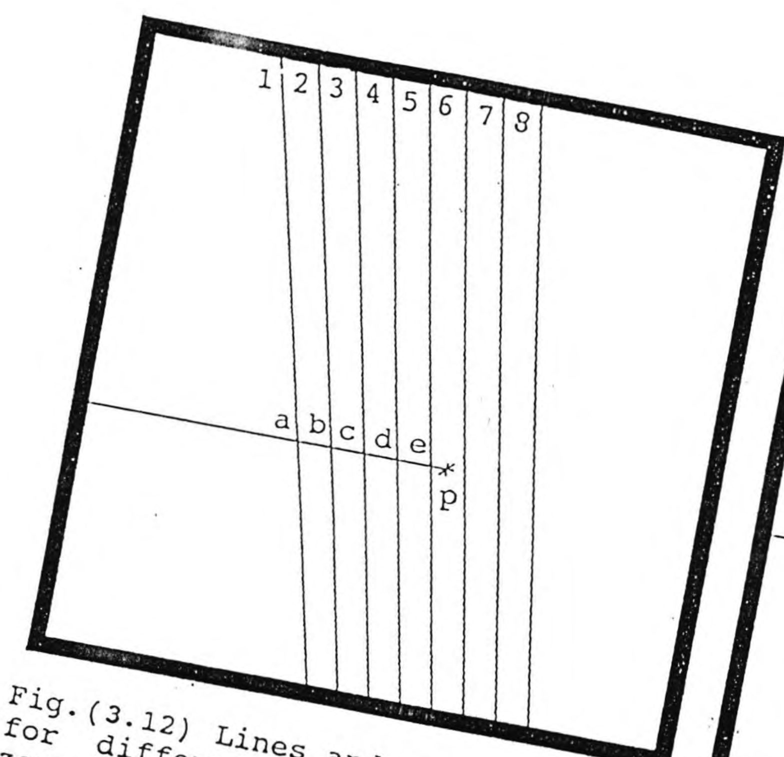


Fig.(3.12) Lines and zones obtained for different levels of Z. Each zone can contain specific values of Z coordinates only.

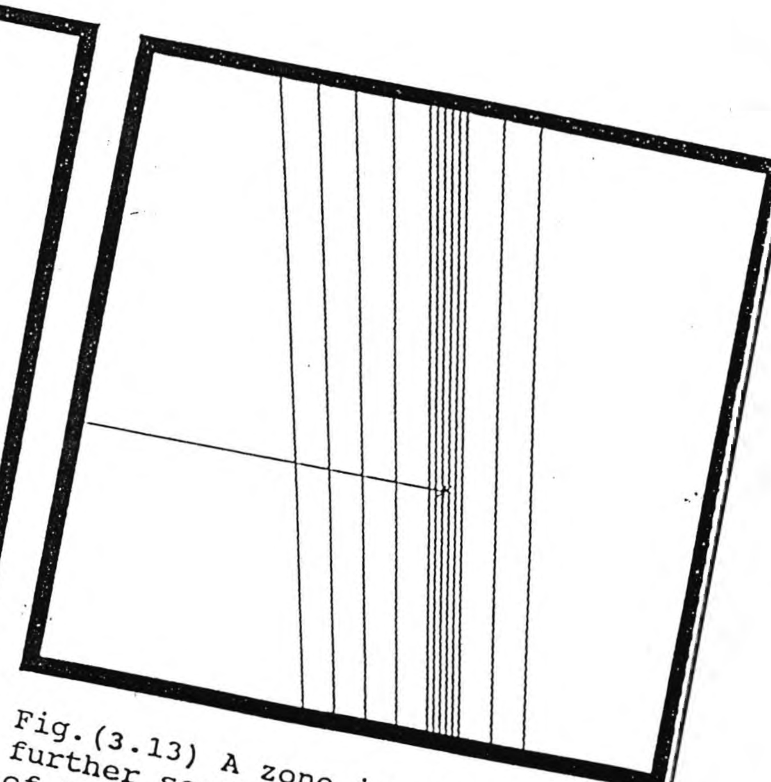


Fig.(3.13) A zone is divided into further sections and limits range of Z.

Then by subtracting P from every two consecutive points the following relations are obtained,

(3-13) $M = a-p$ and $N = b-p$

If M and N are found to be both negative or both positive then it is concluded, that they are located on one side of point P. Because of that point P cannot be inside the zone. If on the other hand it is found that the two successive points make opposite signs then it is concluded that point P is positioned between the two lines. To divide the zone into smaller sections four equidistant points from the top and bottom of the zone are taken and connected respectively. That breaks up the region into five parts as shown in Fig(3.13). Next, as in the previous case, the position of P with respect to the new lines is established. As an example suppose two consecutive lines in Fig.(3.12) represent a change of five millimetres in the Z coordinate. Then each further divided part in Fig.(3.13) would constitute a change of one millimetre in the real world.

3.11 Calibration of Y the Coordinates

Calibration of Y is the same as Z, in the sense that the screen is divided into separate sections as illustrated in the Fig(3.14). Here, line 1 is obtained from the middle part of the stripe while the surface of the table is positioned at Z=0.0 mm. Line 2 is obtained for Z = 35.00 which is the maximum required height. Then these lines are segmented into parts so that each part represents a specific and fixed length in real world. As an example AB,BC,..etc. are arranged to represent 10mm. Next, for an arbitrary point such as p, its location with respect to these lines is established. Then according to which section it is found in, its Y coordinate is calculated.

As illustrated in Fig.(3.9) in section 3.7 the stripe is positioned perpendicular to the X axis. Since line 1 is taken at Z=0.0 mm it also represents the Y axis. Now if point A is chosen to be the origin of the axes, then points B,C,..etc. would represent 10mm,20mm,..etc. on the Y axis. Since point A',B',...etc. are the same points as A,B,C..etc. on the surface of the table this would also apply to them. If a point is found to be between AA' and BB', it can be concluded that its Y coordinate would be

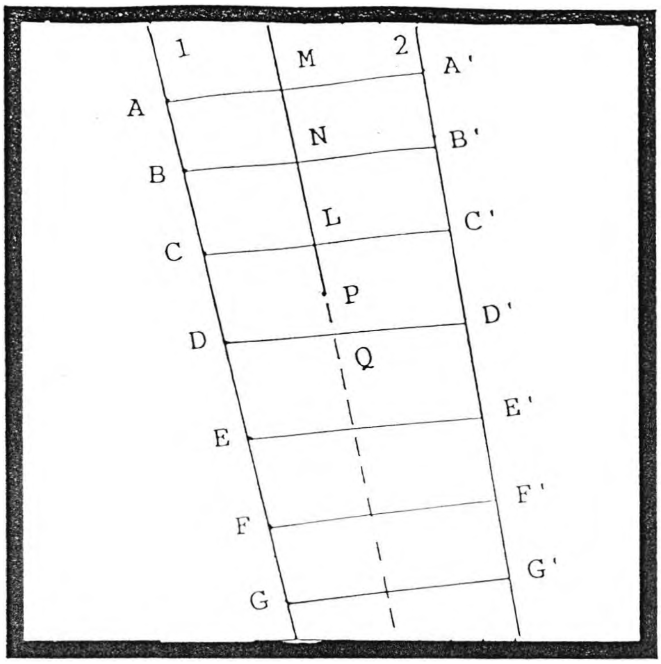


Fig.(3.14). Various sections obtained from calibration of Y coordinate. Each part can contain specific values of Y.

between zero and ten millimetres. Next for an arbitrary point such as p, it's location with respect to these lines is established. Then according to which section it is found in it s Y coordinate is estimated. Based on these measures, the following steps are taken,

1) To find between which two lines P is located.

2) To calculate the value of LP in the real world.

For that the line that passes nearby P and represents the Z coordinate is taken (i.e. obtained from calibration of Z). Then its intersection with lines AA', BB', CC'.. etc. is computed. These are denoted by N, M, L,.. etc. in Fig (3.14). Then the Y values of every two consecutive points (i.e. such as L and M or M and N) are subtracted from the Y value of p. If it is found that two consecutive points make opposite signs, then it is concluded that P is positioned between them. As an example suppose P is located between CC' and DD' as shown in Fig(3.14). Here (P_y-Q_y) would make an opposite sign to (P_y-L_y). This would indicate that P is between Q and L. Here CC' and DD' represent twenty and thirty millimetres respectively. Since P is between

CC' and DD', its Y would be greater than twenty and less than thirty millimetres.

The next step is to establish PL in terms of Y and then add twenty to it. For this the following can be written,

(3.14) U = (length of the CD)/10.0

(3.15) V = (length of the C'D')/10.0

Here U and V represent lengths of one millimetre on each side. Since CD is positioned on the line representing Z=0 mm and C'D' on Z=35 mm the following relation can be written,

(3.16) W = (U-V)/35.0

Here W represents the change in the length of one millimetre for each change in Z. From this

(3.17) T = V + W * (the value of Z)

where T represents the length of one millimetre on the PL line. Then,

(3.18) PL (in real world) = T * (length of LP)

(3.19) P_y = 20.0 + (value of LP in mm)

Since the calibration of Y depends on the line passing near p it has to be accomplished after Z is calibrated.

3.12 Calibration in Practice

In practice it is necessary to make arrangements so that the calibration can be performed relatively easily and in a short time. For that there are the following requirements:

1) To establish lines which divide the screen into the separate areas for calibration of Z, as in Fig.(3.12).
2) To find coordinates of A,B,C..etc. and A',B',C'..etc. in Fig.(3.14) where the stripe is fragmented for calibration of the Y coordinate.

For that a programme is written which first requires the surface of the table to be on the position of Z=0.0 mm. Then an image of the stripe is taken and points along the middle part are extracted. Next, a number of points along it are selected and a line is fitted to them and the coefficient of line is computed. Then, the surface of the table is raised by 5 mm and the calculation is repeated. This is continued until the surface of the table is raised to 35.0 mm (i.e. the highest possible Z coordinate required in experiments).

The next step is to fragment the stripe, so that the position of A, B ...etc. and A',B' ..etc. in Fig (3.14) can be established. During the experiments it was found that, in order to determine the positions of these points, no particular rule could be established. As an example, depending on the position of the camera, consecutive broken spaces (such as A-B,B-C, etc.) may first contract and then expand from one another. To overcome this problem, a transparent sheet of plastic paper is used. On its surface a column of thick black lines spaced 10.0 mm from one another is drawn. Once the sheet is placed on top of the stripe, it divides it into fragments as shown in Fig.(3.15) and Fig(3.17). Then, after thresholding (i.e. Fig.(3.16) and Fig.(3.18)) the centres of the divided areas are extracted. Here, Fig(3.15) is taken for Z=0.0 and Fig(3.17) is obtained for the position of Z = 35.0 mm (i.e. maximum Z). As it can be noted sizable shifts between the two positions can be found. W. Chen and B. Jiang (Chen 1991) used precision grid paper to calculate vanishing points and consequently calibrate the camera.

3.13 <u>Error Sources</u>

Here a number of error sources involved in the calibration procedure are listed.

The first error arises from positioning the surface of the table at different levels. This is in order to obtain coefficients of lines for the calibration of the Z coordinate. Here, the position of the table is directly measured by a ruler. For that an error of 0.5 mm. can be attributed.

The second error takes place in aligning the position of the stripe perpendicular to the direction of motion of the table and along the Y axis. This is illustrated schematically in Fig.(3.9). Also, if the surface of the table is moved up or down then the stripe can slightly swing from its position. Because of these an error in Y can take place (best possible guess 1 mm).

The next error occurs in fragmenting the stripe as shown in Fig.(3.16) and Fig.(3.18). Here first the broken areas need to be satisfactorily captured and then a proper level of thresholding applied. Because of these, the centre of the divided areas may be displaced(3 pixels approximately 1mm & best possible guess 1mm).

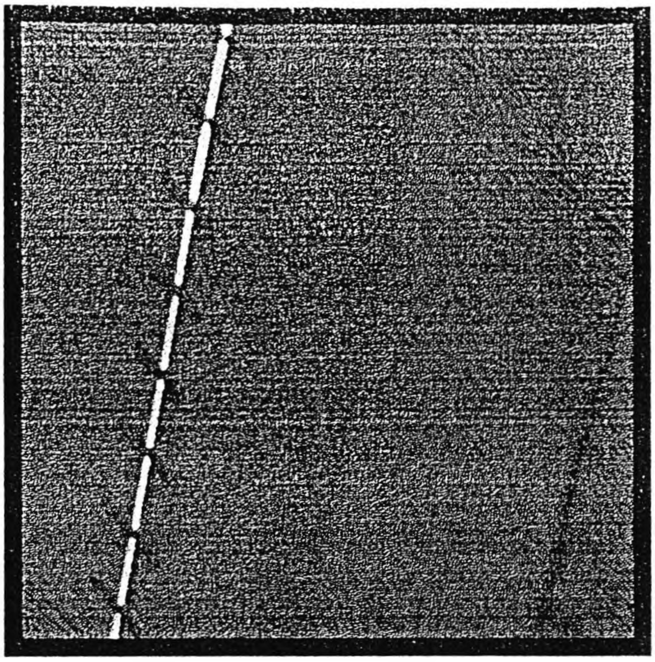


Fig.(3.15) Calibration plastic sheet with column of dashed lines is placed on top of the stripe.

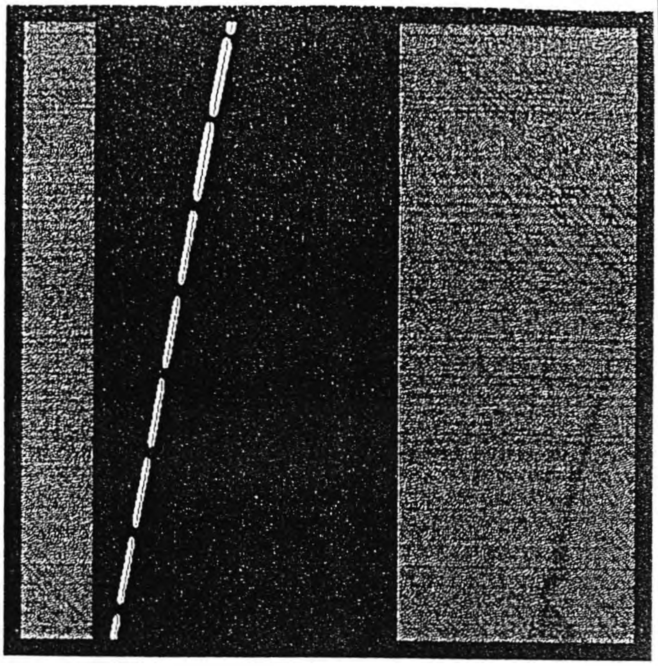


Fig.(3.16) Thresholded image of stripe and broken parts. Position is taken at Z = 0.0 mm.

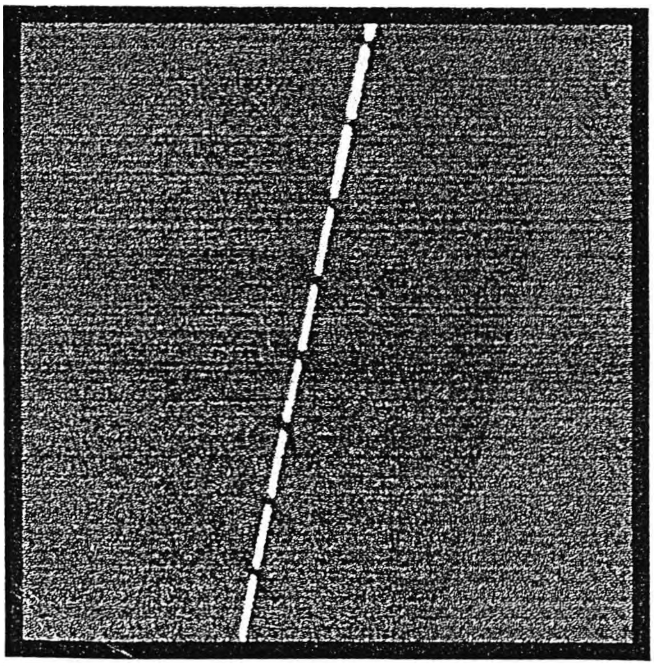


Fig.(3.17) Calibration column at the new position of Z. The sheet is not disturbed.

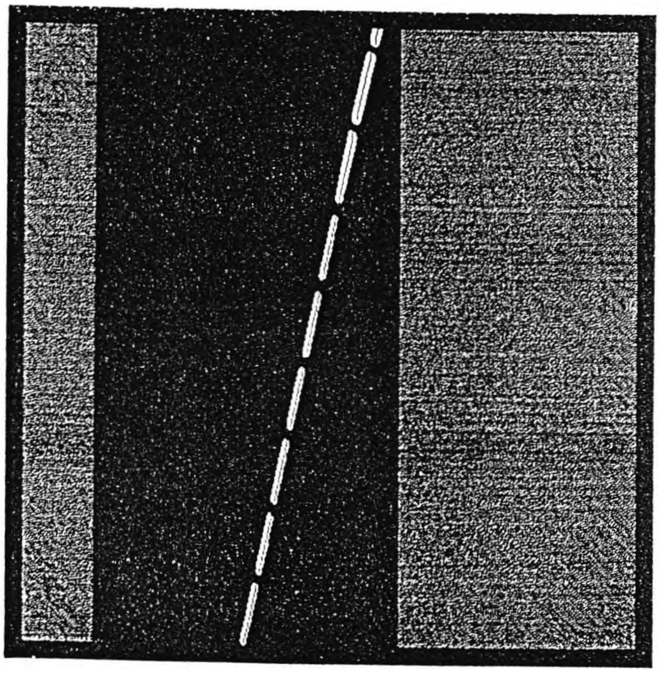


Fig.(3.18) Thresholded image of stripe and broken parts. Position is taken at Z= 35.0.

It is important to note that the precision of the light stripe is not fine enough to justify the use of many significant digits. Because of that, integer data is used.

To assess the overall effect of the above factors, a number of Z and Y points are directly measured (on the scene). Then after calibration their coordinates are computed. By comparing those two values, no more than 2 mm difference is found. Based on this comparison a 3 mm error for the worst possible cases is estimated. This included the image processing operations involved in the process.

3.14 Summary and Comment

Using passive triangulation or two cameras has the advantage that no special equipment is required. In comparison the active triangulation or the structured light technique, has the advantage that three dimensional information can be directly computed. In addition to that many problems associated with the passive triangulation can be avoided.

Based on this it is decided that for the task ahead using a line as a structured light technique can best serve the system. This is because with the line, coordinates of the edge points can be reliably identified and extracted. For that a complete vision system with camera, projector and scanning table is set up. In it the object instead of the source of light is moved along the camera view. Calibration of X is achieved independent of the image. But for the Z and Y coordinates , calibration is accomplished by dividing the screen into zones and sections.

CHAPTER FOUR

THEORETICAL CONCEPTS OF THE SYSTEM

Basically, the method of recognition, consists of the following steps:

1) Take two lines from the object and two lines from the model.
2) Match and compare their structure and examine their position within the object and model.
3) Attribute a score to the model with each successful match.
4) Repeat 1,2 and 3 for every possible pair of lines.
5) At the end, select the model with the highest score, as the one most similar to the object in the scene.

Before entering into the exact details of the operations, some transformations in space together with their matrices are reviewed. Then a few basic operations with lines and vectors are performed and the recognition process is described. This is conducted, first, with polygons in two dimensions, and then with objects in three dimensions in order to demonstrate clearly how the system works. Next the Hough transform is briefly described and then the recognition process is compared with some of its aspects. Finally some of the implementations of the process are discussed and their applicability and merits are evaluated.

4.1 Transformations in 3-D Space

In order to alter the positions of points in the space, there are three basic transformations that can be applied. These are change of scale, translation of the origin and rotation of an axis. Since, here, scale is not used it is not discussed.

In the case of translation, the coordinate axis of the old and new system remain in the same direction, and in the same scale. However, the position of axes can be changed so that a point such as (X,Y,Z) in the old system becomes the origin in the new system. A translation by (X,Y,Z) can be represented by the following four by four matrix in homogeneous coordinates even though the space is three dimensional (Roger 1976).

$$(4\text{-}1) \qquad T = \begin{vmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Applying homogeneous coordinates is necessary because translation of the origin requires a combination of X,Y,Z values together with a linear distance which is independent of these values (Angell 1981). For this a point in three dimensions is represented by a column vector such as $(x,y,z,1)^t$.

For the rotation, the origin and the scale of the coordinate system remain the same. But the new axes are the old ones rotated through an angle such as (Θ). In three dimensions the rotation can be regarded as successive rotations about three axes. The rotation by the angle Θ about the X axis, and rotation by the angle ϕ about the Y axis, and rotation by the angle μ about the Z axis. These are represented by following matrices.

Rotation about the X axis.

$$(4\text{-}2) \qquad R_x = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Theta) & -\sin(\Theta) & 0 \\ 0 & \sin(\Theta) & \cos(\Theta) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Rotation about the Y axis.

$$(4\text{-}3) \qquad R_y = \begin{vmatrix} \cos(\phi) & 0 & \sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Rotation about the Z axis.

$$(4\text{-}4) \qquad R_Z = \begin{vmatrix} \cos(\mu) & -\sin(\mu) & 0 & 0 \\ \sin(\mu) & \cos(\mu) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Here, the total rotational matrix can be written as,

$$(4\text{-}5) \qquad R = R_Z R_Y R_X$$

then the total transformation matrix is defined as,

$$(4\text{-}6) \qquad Q = RT$$

Here matrix Q contains six unknown parameters $X, Y, Z, \theta, \phi, \mu$.

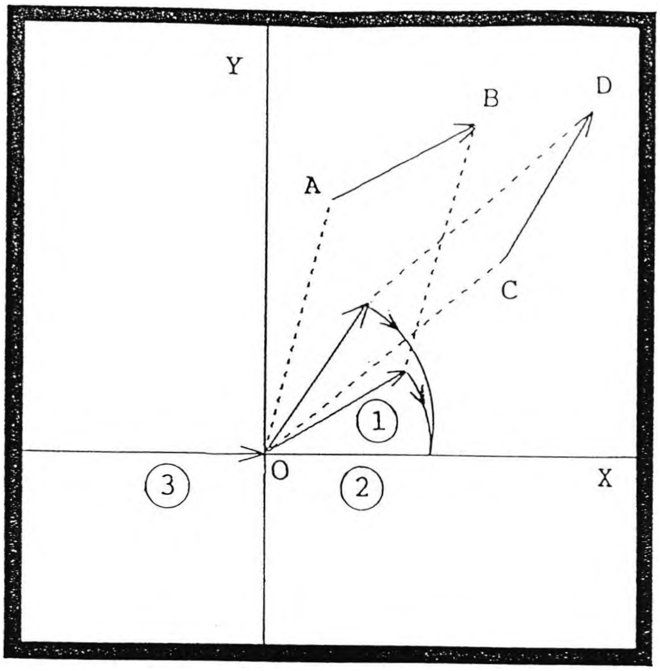
4.2 <u>Operations with Vectors</u>

Here case 1 and case 2 involving operations with vectors are discussed.

Consider vector AB in Fig. (4.1) with its origin at A and its direction from A to B and its magnitude AB itself. First consider case 1 where A is translated into the origin and then B is rotated on the positive side of the X-axis. If matrices of these two operations are successively multiplied, then a total transformation matrix would emerge. Its effect would be the same as the other two matrices combined. These two operations on their own are not adequate to distinguish a vector in two dimensions. This is because any other vector with the same origin and with the same direction but with a different magnitude can produce the same results. However by translating B into the origin by shifting the whole length of the AB along the X_axis the extra necessary operation can be obtained (note, the latest operation can be considered as a scaling as well).

Now consider case 2 with vector CD in Fig.(4.1) which is the same magnitude to AB but with the different origin and direction. By repeating the same operation on CD points C and D can be taken on the exact spots as points A and B in case 1 and the following relations can be written,

$$(4.7) \qquad P \begin{vmatrix} A_x \\ A_y \end{vmatrix} = K \begin{vmatrix} C_x \\ C_y \end{vmatrix} \qquad\qquad (4.8) \qquad P \begin{vmatrix} B_x \\ B_y \end{vmatrix} = K \begin{vmatrix} D_x \\ D_y \end{vmatrix}$$



Fig(4.1). (1) Translation of A into the origin; (2) Rotation of B on X axis; (3) Translation of AB along the X axis.

Here P and K are the total transformation matrices for AB and CD , and Ax,Ay.... etc. are the coordinates of the points A,B,C,D before the transformation take place. By multiplying these relations by the inverse of K the following can be written:

$$(4.9)\quad K^{-1}\,P \begin{vmatrix} A_x \\ A_y \end{vmatrix} = \begin{vmatrix} C_x \\ C_y \end{vmatrix} \qquad (4.10)\quad K^{-1}\,P \begin{vmatrix} B_x \\ B_y \end{vmatrix} = \begin{vmatrix} D_x \\ D_y \end{vmatrix}$$

In these relations matrix $K^{-1}P$ can be interpreted as the matrix through which, points A and B can be transferred and coincident with points C and D. This matrix would be called the Total Transformation Matrix and would be denoted by TTM.

If the magnitude of vector CD in case 2 is shorter or longer than AB in case 1 then the following consequences result:

1) Relation (4.7) would no longer remain true but CD in case 2 and AB in case 1 would remain on the same line along the X_axis.
2) At the origin points B and D would coincide and relation (4.8) would prevail.
3) The TTM matrix P in case 1 would not remain the same as the TTM matrix K in case 2. That is because the last translation would take place for the vector with a different magnitude.

4) Relation (4.10) would remain intact and points B and D would coincide as before.

5) Relation (4.9) would no longer remain valid.

If in case 1 and case 2 the magnitude of the two vectors remains the same, but the direction of one of them is reversed, then TTM would not remain the same. Instead two ends of AB would coincide with the two ends of CD in reverse order. In this case no matter which vector is reversed the TTM would remain the same. If the directions of both vectors are reversed, then the original TTM matrix would remain the same.

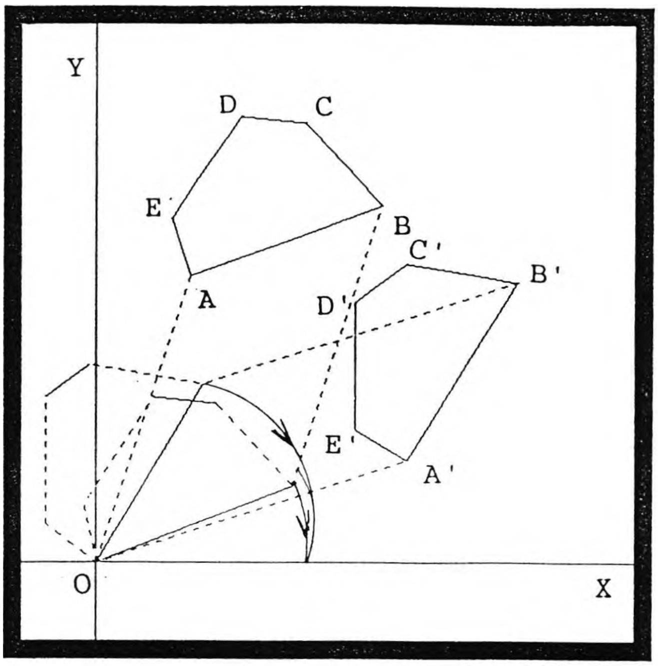
Now suppose the magnitude of the two vectors are the same. Then, the TTM matrix would not depend on the last translation along the X axis because it can be cancelled out from both sides of the relations (4-7) and (4-8).

4.3 Operations in Two Dimensions

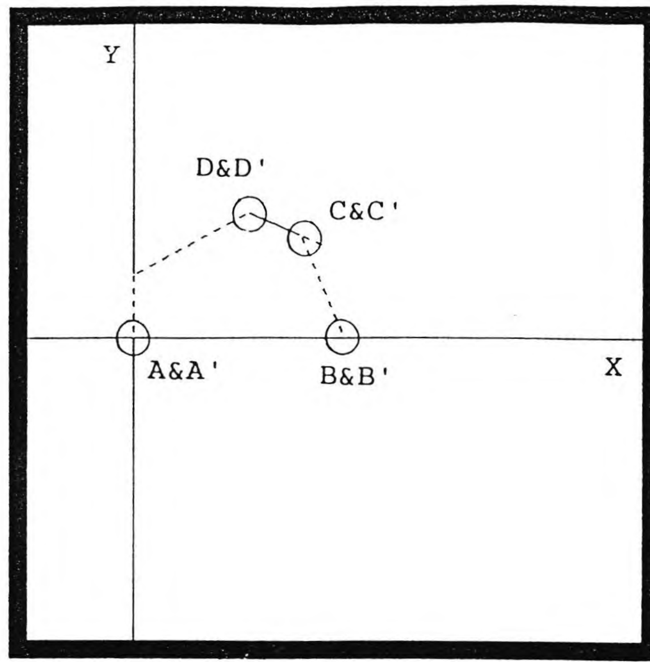
Consider the polygon ABCDE in the Fig.(4.2). Suppose point A is taken into the origin and then B is rotated on the positive side of the X axis. Here AB can be identified on its own if the whole length of it is translated along the X axis (i.e. similar to the case of the vector above). Now consider the case where the aim is to compare AB with another line such as A'B' in A'B'C'D'E'. For this, A' can be taken to the origin and then B' rotated on the X axis. Then the two ends of the two lines (i.e. B and B') can be compared and matched along the X axis. In this case no translation along the X axis is required. Another option is to calculate the TTM and compare the two ends of AB directly on A'B' itself. Here the two end points would coincide if the two lines are the same length.

Now consider the case where with each of those lines a second line is associated. As an example suppose AB to be associated with CD from ABCDE and A'B' to be associated with C'D' from A'B'C'D'E ' in Fig.(4.2). For simplicity, these are represented as (AB,CD) and (A'B',C'D'). Then, the following can be examined.

1) To assess how these sets of lines are structured with respect to one another by matching them while one of them is positioned along the X axis (i.e. one from each set) .



Fig(4.2). Translation of A and A' into the origin and rotation of B and B' on the X axis.



Fig(4.3). (1) Points from two polygons that are matched; (2) Circles indicate the acceptable margin of error.

2) To establish whether they originate from the same place within the polygons or not.

In this example these two sets of lines belong to two similar polygons. Also they are correspondent to one another. For this they would be similarly structured and would come from the same positions within the polygons. By similarly structured it is meant that:

1) AB is the same length as A'B'.
2) CD is the same length as C'D'.
3) The angle between the two lines is also the same (i.e. they can be placed on top of each other).

To establish this fact, suppose A to be transferred to the origin and B to be rotated on the positive side of the X axis as it is shown in Fig.(4.2). If these steps are repeated with (A'B', C'D') in the same order, then A would end up on top of A' and B on B' and so on. This would indicate that these two lines are matched and belong to similar structured lines as is shown in Fig(4.3). However, as explained in the next example, this is not sufficient to conclude that they come from the same place within the polygons.

Suppose two lines (AE,ED) are chosen from ABCDE and (C'D',D'E') are selected from A'B'C'D'E'. These two sets of lines are similarly structured (i.e. AE = C'D' , ED = D'E' and the angle between them is the same) but belong to different parts of the two polygons. Because they are similarly structured they would match. However, the TTM matrix obtained from this match would be different to one that is calculated from previous lines (i.e. those which originated from the same position within the polygons).

Note that the TTM matrix produced by (AB,CD) and (A'B',C'D') would remain the same if DC and D'C'(i.e. lines which are not along the X axis) are replaced by two other corresponding lines. As an example consider (AB,AE) and (A'B',A'E'). These two pairs of lines would make the same TTM matrix as before. This is because the rotational and translational matrices for AB and A'B' would remain the same.

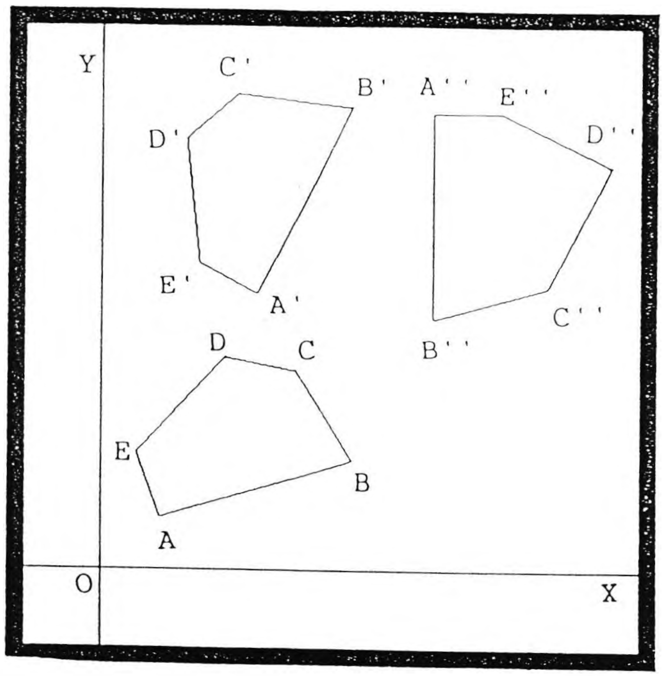
If instead of A and A', points B and B' are first translated into the origin, then the TTM matrix would remain the same (i.e. the same as in section 4.2 where two vectors changed direction). In this case the two other lines (i.e. say DC and D'C') would change position to below the X axis in Fig.(4.3). But they would match as before.

If instead of AB and A'B' (i.e. lines which are along the X axis) two other correspondent lines are taken along the X axis, the TTM matrix would remain the same. As an example consider two pairs of lines (AE,CD) and (A'E',C'D'). Suppose A and A' are translated into the origin and then E and E' are rotated on the positive side of the X axis (i.e. instead of B and B'). Here, an extra rotation of angle EAB and angle E'A'B' will take B and B' to the X axis (i.e. to the previous position of (AB,CD) and (A'B',C'D')). Since these additional rotations are equal they would cancel out. As a result the TTM would remain the same.

From this it can be concluded that, once two correspondent pairs of lines are matched, then for the other correspondent pairs the resultant TTM matrix would be the same.

4.4 Recognition of Polygons

The question that follows next is how this knowledge can be put into practice, in order to identify and distinguish a geometrical shape? To be more specific consider two geometrical shapes A'B'C'D'E' and A''B''C''D''E'' in Fig(4.4). Imagine these are models, and the purpose is to single out and recognise the one which is identical to ABCDE.



Fig(4.4).(1) E''A''B''C'', E'A'B'C', EABC, are the same in three polygons; (2) C''D''E'' is unequal to EDC and E'D'C';(2) AED structurally the same as CDE.

Here, the part consisting of E''A''B''C'' is structurally the same as EABC and E'A'B'C'. But the part consisting of E''D''C'' is not the same as EDC and E'D'C'. This is because the position of D'' is different to D and D'. This makes A''B''C''D''E'' an unequal shape to the other two polygons. Now as an example suppose the two lines (AB, BC) chosen are matched against (A'B', B'C ') and also against (A''B'', B''C''). Because these are similarly structured both of them would match and each match would make its own TTM matrix. If a score is to be given for each successful match, in this particular case both

A'B'C'D'E' and A''B''C''D''E'' would have one score. Next suppose lines (AE ,ED) chosen are matched against (A'E',E'D') and also against (A''E'',E''D''). Because(A'E', E'D') and (AE,ED) are similarly structured they would match. The TTM matrix produced by them would be the same to the one which was calculated earlier. For this, one more score would be added for A'B'C'D'E' . However lines (A''E'', E''D'') and (AB ,ED) are not similarly structured. Because of that they would not match. This would leave A'B'C'D'E' with one score ahead of the A''B''C''D''E''. Once all the lines are tested, then the polygon with the highest number of scores would be the one identical to the ABCDE. This can be extended to many more polygons each acting as a separate model provided one of them is identical to the unknown geometrical shape. In comparing a model and an unknown geometrical shape the following possibilities can occur:

1) The two polygons are identical.
2) The two polygons are identical and parts of their structure are repeated more than once along their shape.
3) The two polygons are not identical but part of their structure is the same.
4) The two polygons are not identical and possess no similar structure in their shape.

For the first case consider ABCDE and A'B'C'D'E'. Here for every two lines from the first one a correspondence can be found in the second one. These would be structurally the same and would come from the same position within the two polygons.

For the second case consider (AE,ED) versus (A'E',E'D') and also (D'C',E'D') in Fig.(4.2). Here (AE,ED) would match with the other two sets. This is because both have the same structure. But, the TTM matrix generated from them would be different. However the number of matrices which are generated from (A'E',E'D') would eventually outnumber the other matrix. At the end there would be several matrices associated with the model. However the matrix with the highest number of occurrences would be the one, selected for that model.

For the third case, consider polygons ABCDE and A''B''C''D''E'' in Fig.(4.3). Here because some parts of the two shapes are the same, similar matches would be generated. But their score would not be enough to top the score from identical model.

For the fourth case, obviously, no match would be found.

Once all the models are examined, each of them would end up with the number representing the highest scoring TTM matrix for that model. At the end the model with the highest number of TTM matrices would be chosen as the best match to the unknown geometrical shape.

Suppose a polygon has N sides. Each line along the X axis can couple with (N-1) remaining lines. If each line is taken once along the X axis, then there would be N(N-1) cases to be examined . In this way, each pair of lines would meet twice during the process. This would be reduced to N(N-1)/2 if each pair of lines is met only once. From this it can be concluded that to examine two identical polygons $N^2(N-1)^2/4$ cases need to be tested.

Each case itself consists of eight possible variations. As an example consider (AB,CD) and (A'B',C'D') while (AB,CD) is kept stationary with A at the origin and AB along the X. Then there are four variations where points A',B',C',D' can be taken into the origin one at a time. For each of these variations there is another one where the second line (i.e. the line that is not along the X axis) can be reflected against the X axis. The exact details of these operations are discussed in the matching process of Chapter six. Also, previous work in this area included Ballard (1981) and Davies (1986,1989).

4.5 Recognition with Connected Lines

Suppose (AE,AB) is chosen to be matched with (A'E',A'B') and (A''E'',A''B'') in Fig.(4.4). Because these set of lines are structurally the same they would match. This is despite the fact that (A''E'',A''B'') comes from unequal polygons to ABCDE. Now consider these lines in the position of matching and suppose ED, E'D' and E''D'' are also transferred with them respectively. Then (AE,AB,ED) and (A'E',A'B',E'D') would still remain structurally the same and would still match. But, this would not be true with (AE,AB,ED) and (A''E'',A''B'',E''D''). Since here the two ends of ED and E''D''(i.e D and D'') would not match. In this example by including an extra line, a match with an unequal polygon is prevented. This can be used to suppress the number of scores for unequal polygons. The major disadvantage is that it can result in a significant increase in the number of operations.

One way to reduce processing is to use connected lines only. This would not affect the overall recognition process. It can be applied for combinations of two lines as well as three or more. As an example consider the following three cases for ABCDE and A''B''C''D'''E'' in Fig.(4.4). Here, EABC is the same as E''A''B''C'' and any two corresponding lines from these shapes will match.

1) Consider a combination of any two connected and any two unconnected lines. Here, (A''E'',A''B''), (A''E'',B''C'') and (A''B'',B''C'') would match with their corresponding part in ABCDE.

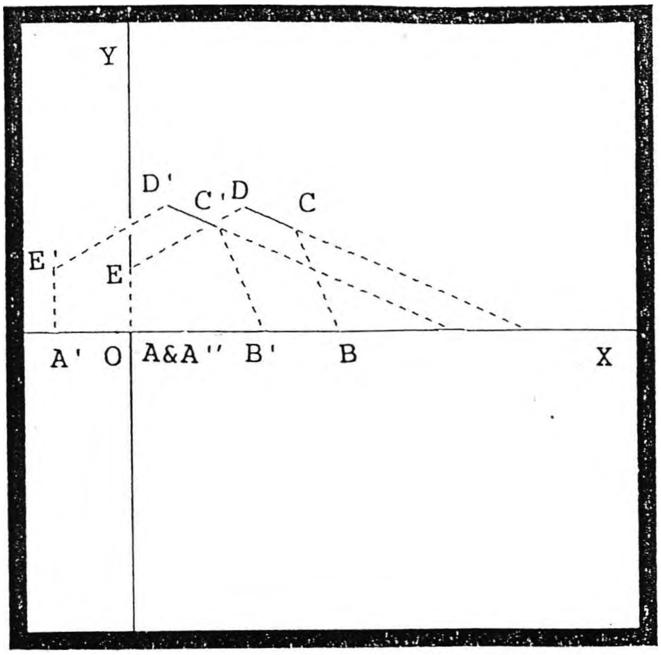
2) Consider a combination of two connected lines only. Here the number of matches would be reduced to (A''E'',A''B'') and (A''B'',B''C'').

3) Consider a combination of three connected lines only. For this case the number of matches would further reduce to a single case of (E''A'',A''B'',B''C''). This can be particularly useful if a large number of models are involved since, it is less likely to find a match with unequal polygons.

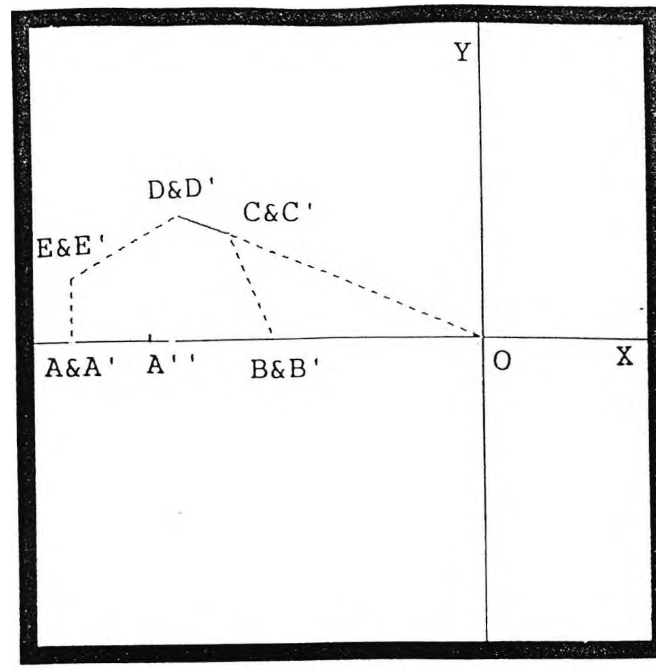
4.6 <u>Recognition with TTM Matrices</u>

As described above, recognition consist of two steps. First, the end points of the lines are matched. Then if it is found that they are coincident, the related TTM matrices are calculated. In this way only potentially useful matrices are stored.

Now imagine the case where some of the lines are only partially available. This can happen if some edges are occluded or only partially extracted. As an example, consider polygons ABCDE and A'B'C'D'E' in Fig.(4.5). Here instead of A'B' only A''B' is available. Consider the case where two sets of lines (A''B',D'C') and (AB,DC) are chosen to be matched. Here instead of point A', point A'' is translated into the origin and the whole position of the A'B'C'D'E' is dislocated. Because of that, matching can not be conducted. In this circumstance recognition can still be



Fig(4.5). A'' instead of A' is translated into the origin and consequently the two polygons can not be matched.



Fig(4.6). Intersections of DC and D'C' with X axis are translated into the origin and consequently TTM matrix can be calculated.

achieved, but only through calculating TTM matrices.

For this DC and D'C' can be extended to intersect the X axis. Then the two intersected points can be translated into the origin as shown in Fig.(4.6). This would place the two polygons in the same position as when the whole length of A'B' is available. However even in this position the end points cannot be coincident and matched (i.e. A and A'').

One way to achieve recognition, is to calculate all possible TTM matrices regardless of end points and positions of the lines. For this a large number of matrices would need to be calculated and stored. However, it is still possible to reduce the number of cases by examining the position of lines with respect to each other. For instance only cases need be considered where lines are positioned along each other (i.e. such as DC and D'C' in Fig.(4.6)), since this would be an indication of a potentially worthwhile position.

4.7 Recognition in Three Dimensions

Up to this stage the whole argument has been confined to two dimensions. Since the ultimate goal is to recognise objects the process is modified, in order to make it compatible with three dimensions. This includes transformation processes as well as the positions where matching can take place.

As an example, consider lines PQ and RS in the object of Fig (4.7.1). Here similarly to the case in two dimensions, the first objective remains to take one of the lines along the positive side of the X axis. To achieve that the following operations are required,

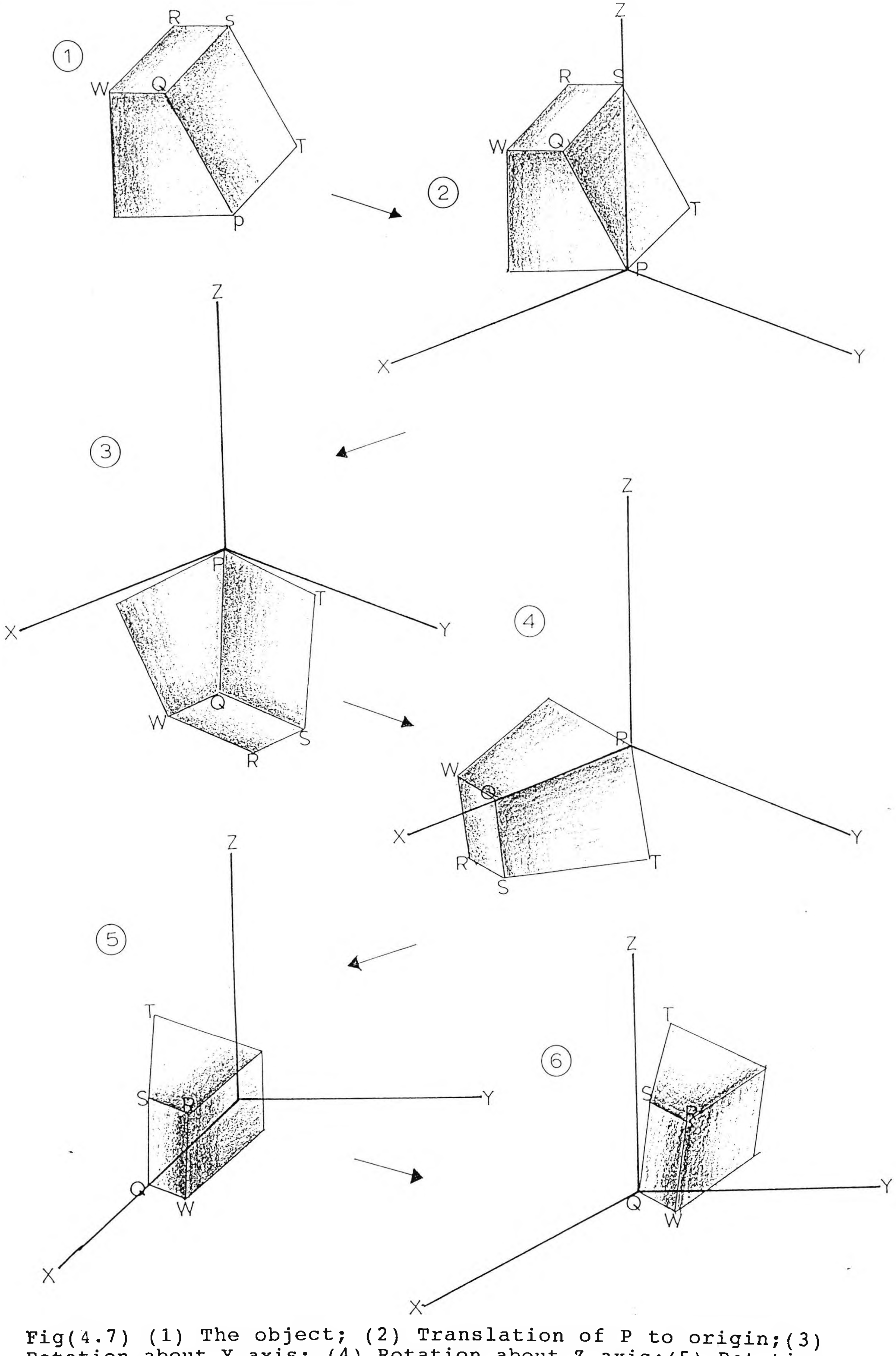
1) Translating point P into the origin as in Fig (4.7.2).
2) Rotating point Q together with R and S about the Y axis so that point Q is placed on the XY plane (as illustrated in Fig.(4.7.3)).
3) Rotating Q, together with R and S , about the Z axis, so that Q is placed on the positive side of the X axis (as shown in Fig (4.7.4)).

By this stage, while PQ is taken into the required position, RS can be located anywhere in the space. The next step is to take RS to a specific position where matching can take place. This has to be done without moving PQ from its position. Due to this the following operation is needed:

4) Rotating RS about the X axis, so that it is placed parallel and above the XY plane (as in Fig(4.7.5)).

Now consider the case where ST instead of RS is selected. Since ST is parallel to QP then no matter how it is rotated, it would always remain parallel to the XY plane. Because of that it is not possible to establish one specific location for it. For this, parallel lines are excluded from operations.

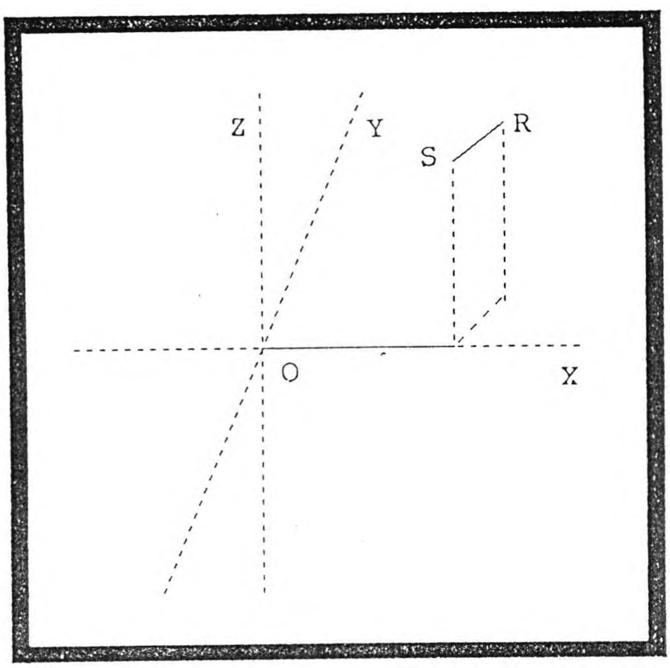
By the end of the fourth step, the two lines would be in the position of matching. Suppose two lines from the object are taken into their respective position and kept stationary. Then there are eight possible variations in which two lines from the model can be taken into the matching position. As an example consider RS and QP in Fig.(4.7.1). The first four variations include taking R,S,Q,P into the origin one at a time. For each of these variations there is an extra one, concerning reflection against



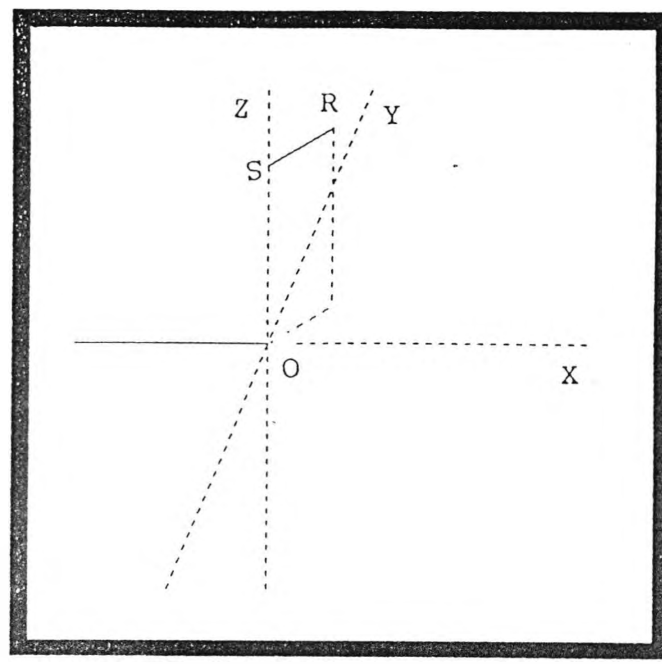
Fig(4.7) (1) The object; (2) Translation of P to origin;(3) Rotation about Y axis; (4) Rotation about Z axis;(5) Rotation about X axis; (6) Translation of Q to origin.

the XZ plane. More details of these are discussed in the matching process in Chapter six.

Now consider the case where some of the lines are only partially available. Here, similarly, to two dimensions recognition can still take place, but without matching (i.e. without testing the end points of the lines). For this, the line parallel to XY can be projected onto the XY plane as shown in Fig.(4.8). Then the point of intersection can be translated into the origin as shown in the Fig.(4.7.6) and Fig.(4.9).



Fig(4.8). Projection of SR into the XY plane and it's intersection with X axis.



Fig(4.9). Translation of point of intersection into the origin.

4.8 TTM Matrices in Three Dimension

In order to work out the TTM matrices in space, it is necessary to multiply all the matrices in the order in which they have taken place. This applies both to matrices obtained from models and objects. Suppose the first translational matrix is denoted by T_f and the three rotational matrices by R_x, R_y and R_z according to the axis around which they are rotated. Also suppose the reflection matrix is denoted by P (for variations in models). Then matrices F_o and F_m for object and model can be written as,

$$(4\text{-}11) \qquad F_o = [R_{ox}][R_{oz}][R_{oy}][T_{of}]$$

$$(4\text{-}12) \qquad F_m = [P][R_{mx}][R_{mz}][R_{my}][T_{mf}]$$

In these relations an extra translational matrix would also be required if the line along the X axis is shifted.

Suppose two related lines in the object and model are matched. If the two correspondent point are represented by $(X_o,Y_o,Z_o,1)^{-1}$, and $(X_m,Y_m,Z_m,1)^{-1}$ then the following can be written,

$$(4\text{-}13) \quad [F_o](X_o,Y_o,Z_o,1)^{-1} = [F_m](X_m,Y_m,Z_m,1)^{-1}$$

If this is multiplied by the inverse of F_m then the result would be,

$$(4\text{-}14) \qquad TTM = [F_m]^{-1}[F_o]$$

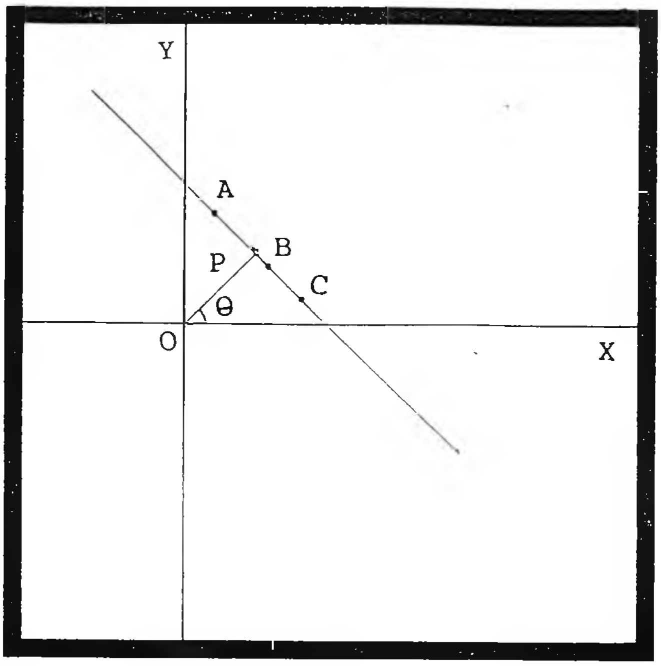
This can take a point from the object and transfer it into its corresponding point in the model.

The process of recognition is the same as in two dimensions. Several models would represent a set of different objects. One of them would be identical to the unknown object in the scene. First two lines from the object would be selected. Then each model would be taken one at a time. The two lines from the object would be matched against every two edges of each model. For each successful match the related TTM matrix would be calculated. This would be repeated until all the edges of the object are tested. At the end for each model the highest occurring TTM matrix would be selected. Eventually, the model with the top score would be chosen as the best match to the object in the scene.

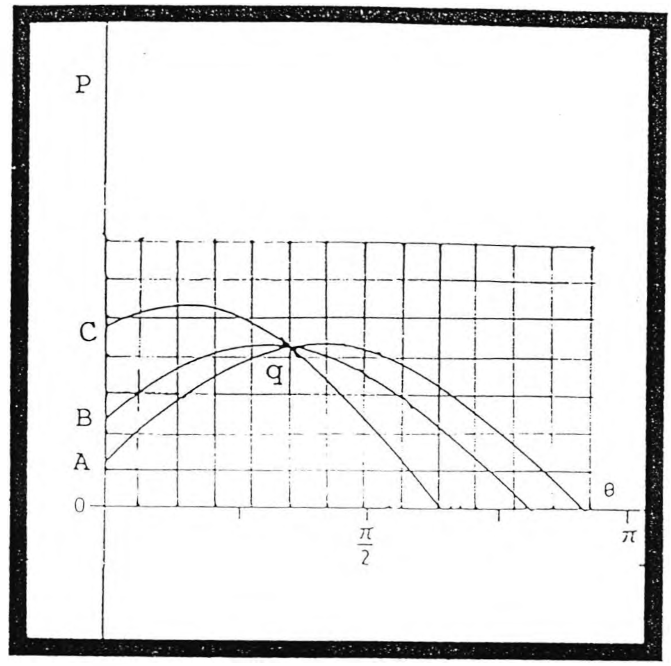
4.9 Hough Transformation

The recognition method that is described, has a number of common features with the Hough technique (Hough 1962). For that a brief description of it is given, before it is compared with the recognition process.

Originally the Hough technique was used to detect lines from edge points with unknown direction. As an example, consider the



Fig(4.10). A,B,C the three points along the same line. P the shortest distance to the line. θ the angle with X axis.



Fig(4.11). (1) curves obtained from lines passed through A,B,C; (2) Squares of accumulator cells; (3) q the point of intersection.

line in Fig(4.10) with points A, B and C along its path. The equation of the line can be represented by p and θ ($0<\theta<\pi$) as proposed by Duda and Hart (Duda 1972).

$$X \cos(\theta) + Y \sin(\theta) = p \qquad (4\text{-}15)$$

Here p is the shortest distance from the origin of the axis, and θ is an angle that the shortest line makes with the X-axis as in Fig.(4.10). Let the coordinates of points A,B,C be denoted by A_x, A_y,..etc. Then for a point such as A the above equation can be written as,

$$A_x \cos(\theta) + A_y \sin(\theta) = p \qquad (4\text{-}16)$$

Suppose this is plotted against angle θ and p (i.e. in parameter space with θ and p as its axis). Then the result would be a sinusoidal curve similar to Fig.(4.11). This curve represents all lines passing through the point A. If this is repeated with points B and C then each point would make a curve of its own. But all the curves would pass through a single point such as q. This is because A,B,C, are in a straight line as shown in Fig.(4.10). From this, it can be concluded that a line can be represented by a point in parameter space. In order to make use of this property, parameter space can be subdivided and digitised into squares as shown in Fig.(4.11). Then each square can be used as an accumulator cell initially set to zero. If a curve passes through a square then the counter related to it would be increased by a unit. After all the curves are plotted the square with the highest number of scores would be the one representing the two parameters of the line. As an example in Fig.(4.11) the square with the point q, would specify the values of parameters for the line in Fig(4.10). In practice, however, the number of curves would be much higher. For this, and also because of effects of digitisation and noise, selecting a single square is often not possible. As a result, peak detection would be required. This involves smoothing and then selecting values greater than a threshold Duda and Hart (Duda 1973).

This technique can be employed for the curves with the known equation. As an example consider the following equation for the circle. Here R is the radius and a and b are centre of the circle in Cartesian coordinates.

$$(4\text{-}17) \qquad (X-a)^2 + (Y-b)^2 = R^2$$

This equation can be processed in the same fashion as equations of a line, but in three dimensional parameter space (i.e. with R,a and b as parameters). Computation time for the Hough technique can be quite high. This can rise considerably if the number of parameters increases or the size of the accumulator cells decreases. The amount of calculation can be reduced if some information about the curves is available. As an example if the direction of the line is known then the dimension of parameter space can be reduced (Shirai 1987). Detailed study of Hough transform can be found in Davies (1990) and Grimson (1990).

4.10 Comparison of Hough Transform and Recognition Process

The Hough Transform is a technique that exploits a particular structural relation between pixels in an image. This is used in order to generate information about the presence of specific curves or features in the scene. The same can be said about the recognition process described for the objects. But here, instead specific structural relations between edges are exploited.

In the Hough technique, parameter space is used to transfer points of a line or curve into specific positions. Parameters involved are those which make equations of lines or curves. Suppose a particular curve or feature is present in the image and its points are transferred into parameter space. If its equation is compatible with parameter space then its points would transfer into the same location in that space. As a result, the equation of a line or curve can be verified and its parameters determined.

For the recognition process, the edges of an object and models are rotated and translated into the space. These operations are arranged so that a specific matrix is generated (compare with the specific point in parameter space). Parameters that are involved are those of rotational and translational matrices (compare with the parameters of lines or curves). Consider the situation where one of the models is identical to the object. Then all their corresponding edges would make the same TTM matrix at the end. As a result an object can be identified (compare with the equation of line or curve that can be verified). Also through the parameters of rotation and translation its position in space can be located (compare with the position of line or curve through its parameters).

In the Hough technique accumulator cells are used to count the number of points in an area. In the recognition process matrices are counted for this purpose. In the Hough transform not all the points would be transformed into the same accumulator cell (i.e. because of the effect of noise they would cluster). The same can be said about the recognition process where not all the matrices would be exactly the same . For this, special arrangements are needed to compare and group potentially similar matrices together (see Chapter six for grouping matrices).

In terms of computational time both the Hough transform and the recognition process can be considered to be expensive. In the Hough technique knowing extra information, such as the direction of the line can lessen the amount of computation needed. In the recognition process measures such as examining length, position and angle between the lines together with the matching process can reduce the amount of processing.

The Hough transform is not sensitive to local noise. Here noise is defined as data which does not belong to specific lines or curves. This is because in this technique global information of the image is used. As a result the overall outcome cannot be altered by unrelated data.

The same can be said about the recognition process. But here noise appears in the form of the undesired lines. Two types can be specified. First, lines which are present in the scene but do not belong to a particular object and second, lines which are only partially extracted . As shown earlier , only corresponding lines within the object and its model can produce the required TTM matrix. Because of that, lines which do not come from within the structure of the object cannot alter the overall results . For the second type (i.e. partially extracted lines), recognition can still be achieved as described in section 4.5.

From this it can be concluded that in many respects, the recognition process and Hough transform are similar techniques. Their implementation depends on the purpose for which they are used.

4.11 <u>Some aspects and implications of the Process</u>

This process is primarily employed by Boyter and Aggarwal (Boyter 1986) with the assumption that more than one object would be placed in the scene. For this, recognition of one object would have to be conducted in the presence of the other ones. This condition implies that edges from the other objects would interfere in the process and act as undesired lines (as defined in section 4.10). Also because of the occlusion of objects some of the edges would not be fully extracted.

Here, the problem is solved by taking advantage of the insensitivity of the process to undesired lines. Since some of the lines would be partially extracted matching cannot be performed. For that all possible TTM matrices are calculated and stored. This inevitably results in a large number of matrices to separate and process.

In the present system, however, it is assumed that objects would be placed one at a time in the scene. If several objects are used, they must be so placed so that edges would not be occluded. This condition implies that most of the lines would be adequately extracted so that edges can be compared and connected lines can be established and matching can be performed.

Here are three main objectives:

1) To make the process run efficiently by pruning the system and eliminating unnecessary calculations before operations are conducted.
2) To test algorithms on a number of objects with divers shapes and complexity and to explore its effects on various combinations of lines and in different positions.
3) To investigate some modified algorithms with which the process can be executed.

For the first point, the following measures are considered.

1) Comparing edges from the object and model before operations are conducted.
2) Matching edges, before TTM matrices are calculated and stored.

Here, step one is concerned with acquiring some information about edges and their position within the objects in order to avoid unwanted processing. Measures consist of:

1) Examining the angle between selected lines (i.e. from object and model).
2) Comparing lengths of the lines.
3) Examining whether lines are connected or not.
4) Calculating the shortest and longest distance between two end points of the lines.

Step two is concerned with avoiding those transformations and matrices that cannot be associated with the object and its model. By doing so, the number of potentially useful TTM matrices is kept low. This can help to separate and store matrices effectively. Also, similar matrices can be found and counted easily. In the same context, W. Eric L. Grimson and Tomas Lozano-Perez (Grimson 1984 and 1985) proposed using constraints on the search process in order to control the combinatorial explosion inherent in such a search process.

The second point is concerned with examining objects with diverse complexity and different shapes. These are to be recognised in various positions and orientations. Here the aim is to investigate how varying combinations of edges can interact with the system and affect the recognition process.

The third point is concerned with examining methods of executing the process. These consist of:

1) Two connected or unconnected lines (i.e. two from object and two from model).
2) Two connected lines only.
3) Three connected lines only.

Here, step one is to identify and recognise objects and to assess the performance of the system in general. Step two is concerned with reducing the number of operations and to examine how the recognition would be effected. Step three is concerned with introducing more complex patterns than two lines into the operation. This is to explore whether this would favour objects present in the scene or not since models which are not identical to the object would be less likely to possess the same pattern. In the same context T.M. Silberberg (1986) and R. Horaud (1987) used junctions with more than two lines to reduce interpretation of the scene.

4.12 <u>Summary and Comment</u>

The main purpose of this chapter has been to explain the principal concept of the recognition algorithm and operations involved with it. This includes cases where recognition can be achieved with connected lines or TTM matrices alone. Then the process is extended to three dimensions and an analogy with the Hough transform is drawn and their similarities are outlined.

In many vision systems in order to specify an object several models need to be made. These are required in order to demonstrate objects in different positions or specific views. However with the present method recognition can be achieved with only one model for each object with an independent view. Also objects can be studied in any position and view. This can be especially advantageous when a large number of objects are involved. But the process can be inefficient because each scene is processed in exactly the same manner as the others regardless of the scene. For that operations involving pruning, matching togeter with options with more than two lines are proposed.

CHAPTER FIVE

PREPROCESSING PROCESS

The preprocessing process refers to the section of the recognition process where objects are scanned and images are taken. Also, from each image, all the necessary information in its most compact form is extracted and stored. Functions of preprocessing can be summarised as follows:

1) Scanning and grabbing images from the scene.
2) Segmenting and distinguishing a stripe from the rest of the image.
3) Extracting the middle part of the stripe.
4) Noise reduction on the findings of the middle lines.
5) Separating connected lines and classifying and storing the end points of each line.

At the end, some problems related to extraction of information from stripes are discussed.

5.1 Scanning objects and grabbing images

The first step in the recognition process involves the scanning of objects . During each scan, first, an image of the scene is taken. Then it is processed and potential edge points are extracted. Once that is completed the surface of the table is moved forward and the whole process is repeated for the next scan.

For each scan the surface of the table is moved by a millimetre in the direction of the X axis (as is defined in the calibration). In order to cover the surface of the objects up to forty scans are required. But for a number of objects in some specific positions (depending on the shape of the objects), fewer scans are found to be adequate. The overall timing of scans was approximately varied between twelve and fourteen minutes. This includes all the necessary processing between two scans. The overall timing can be reduced by taking fewer scans and with wider spacings. But this results in construction of features with a smaller number of data points at the end.

Images are grabbed and stored in the frame store for the duration between two scans. Images are made of 512 by 512 pixels and consist of 256 grey levels. Once the necessary information from each image is extracted, it is overwritten by the incoming one. This is found to be necessary because each image can occupy a large amount of memory and it is not practical to store all of them. As an option to speed up the process, consideration was given to reduce the size of the images to 256 by 256 pixels. But it was found that by doing so, some information was lost. Because of this it was decided to proceed with the original 512 by 512 images. Instead it was arranged to process only that area of the image where stripes can be found. As an example, in Fig.(5.2) to Fig.(5.4) the wide black stripe crossing from the middle part of the image is the area where stripes can actually be found.

5.2 Thresholding images

The information that comes from the shape and patterns of the stripes, are the building blocks from which the real outside world can be reconstructed. Because of this it is necessary to segment and separate the shape of the stripe from the rest of the image as accurately as possible. For this, thresholding is used to segment the images. The first step in this approach is to assess a region in grey levels from which the pixels of the stripes would be located distinctly in it.

The importance of selection of a proper thresholding level is illustrated in Fig(5.1) to Fig(5.4) where an object is crossed by the stripe. In these images the same scene is thresholded with different level of thresholding. In Fig.(5.2) the level is chosen to be lower than what it has to be. The outcome

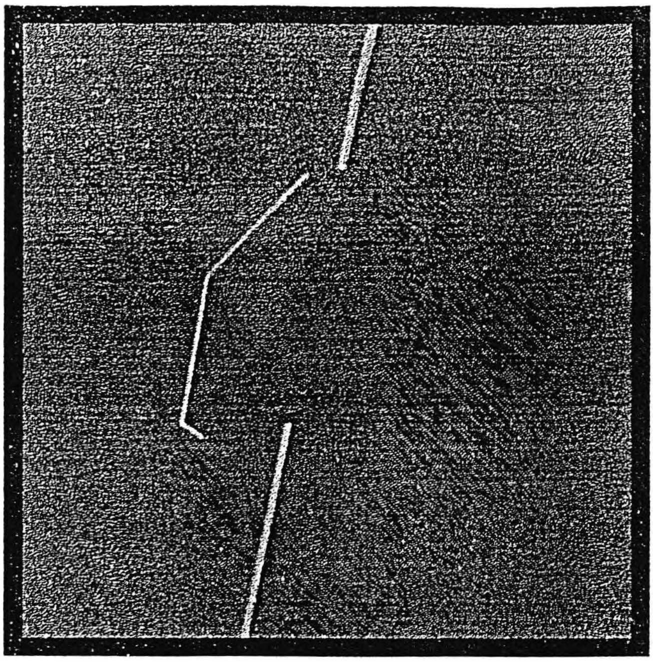


Fig. 5.1. The stripe and the object that is crossed by it.

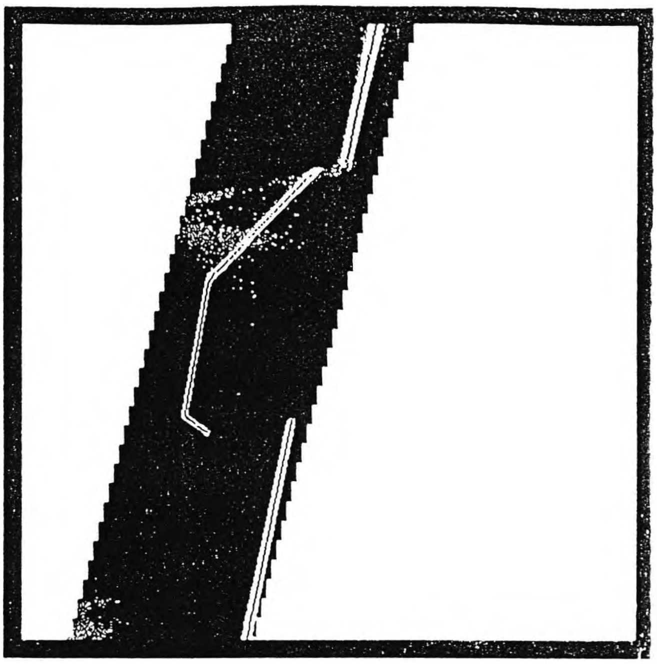


Fig. 5.2. The result of thresholdin by the lower level than is required

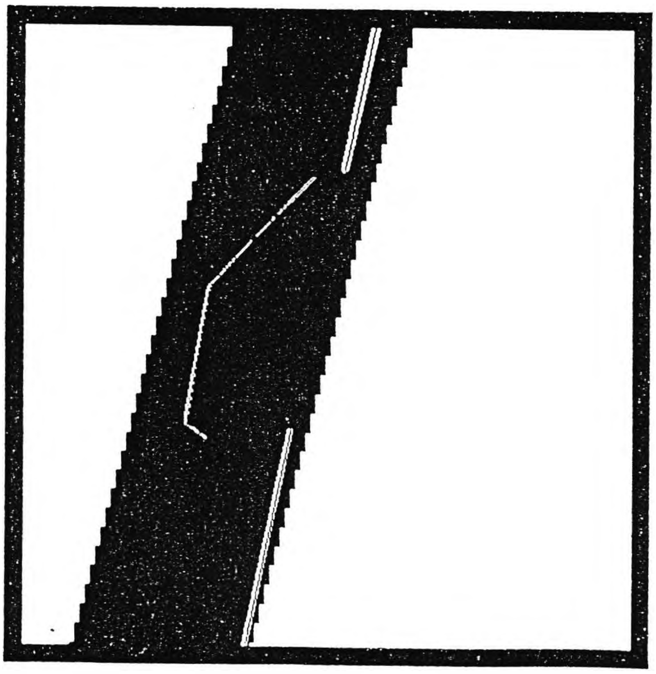


Fig. 5.3. The result of thresholding by the higher level than is required.

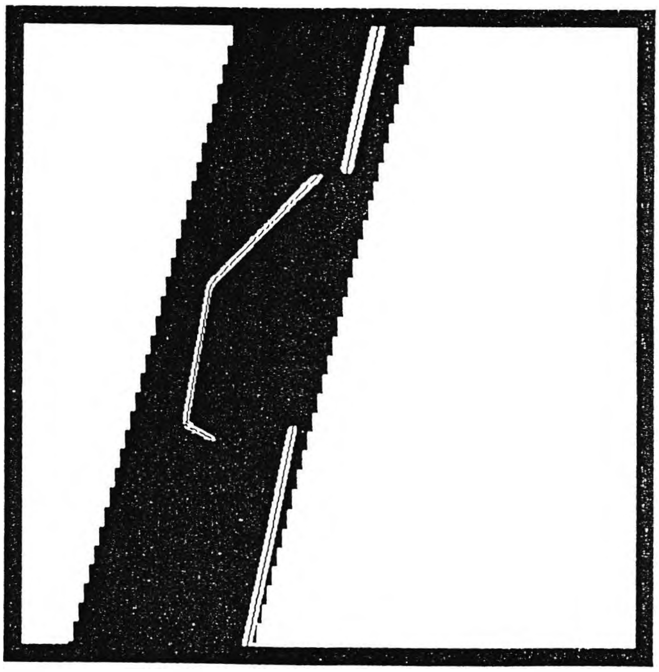


Fig. 5.4. The result of thresholding by just about right level.

is some white dots and points which are scattered around the image. This can affect the next stage of processing which is concerned with the extraction of the middle part of the stripe. In Fig(5.3), contrary to the previous image the level of thresholding is chosen to be higher than what it has to be. The result is gaps along the stripe which in turn can pass on unreliable data to the next stage of processing. But in the image of Fig(5.4), the level of thresholding is at just the appropriate level. Here the shape of the stripe is continuous and clear, and no sign of noise (i.e. white dots and spikes) can be traced in the image. As a result, the middle lines are extracted with no significant breakage along their path.

Before any attempt is made to grab and threshold an image the following considerations are taken into account:

1) The intensity and illumination component of the source light and reflectance factor of the objects and the surface of the table.
2) The ambient light.
3) The intensity of the light source itself.

Grey levels of an image are determined by two components. The amount of illumination that is received on the scene and the amount of light that is reflected from it. If the illumination component is denoted by $I(x,y)$ and reflectance by $R(x,y)$ then the light-intensity function $F(x,y)$ can be obtained by multiplying the reflectance and illumination factors together (Gonzales 1987),

$$F(x,y) = I(x,y)\, R(x,y) \quad (5-1)$$

where,

$$0 < I(x,y) < \text{Infinity} \quad (5-2)$$

$$0 < R(x,y) < 1 \quad (5-3)$$

The value of these component can be affected by factors such as material, colour or smoothness of the surfaces. As an example the surface of the table (made from metal) responds differently to the surface of the object (cut from wood) under the same lighting conditions. Even two surfaces of the same object can react differently in the same lighting condition due to the way that they are cut. Because of that the thresholding level that is found to be satisfactory on the object can produce poor results on the surface of the table or vice versa. To counter these effects objects and the surface of the table are sprayed with green fluorescent paint. This is found to produce desired effects after several other options, such as varnish or red fluorescent paint were tried. Once the paint is applied, selecting a single thresholding level for all parts of the image become possible.

These measures are adequate if the equipment is placed in surroundings so that ambient light does not change significantly. To assess the effect of the light in the environment consider the following average numerical figures which illustrate some typical ranges of I(x,y) (Gonzalez 1987). On a clear day the sun may produce in excess of 9000 foot-candles of illumination on the surface of the earth. This figure decreases to less than 1000 foot-candles on a cloudy day. On a clear evening, a full moon yields about 0.01 foot-candle of illumination. The typical illumination level in a commercial office is about 100 foot-candle. Similarly the following are some typical values of R(x,y): 0.01 for black velvet, 0.65 for stainless steel, 0.80 for flat-white wall paint, and 0.93 for snow. To counter some of these effects a hood is made and all the equipment is placed inside it so that, the surrounding light is stabilised and isolated.

The third case concerns the light stripe itself. At first glance it may seem that having an intense light stripe is an advantage in distinguishing the stripe from the background. In practice however, it is found that an intense light source causes dispersion around the vicinity of the stripe and actually makes thresholding more difficult. Because of this, the intensity of the light source is adjusted to produce the desired effect. Also some other factors such as minor variations in surface reflectance, can produce additional errors. But this is not considered because the equipment used is found to be not sensitive enough to respond to it.

Having stabilised the situation the next step is to assess a suitable level of thresholding. For this a histogram is used to obtain information about the content of images. Fig.(5.5) illustrates a typical histogram of an image with a stripe on it. Here values in the Y axis are divided by a constant to make them

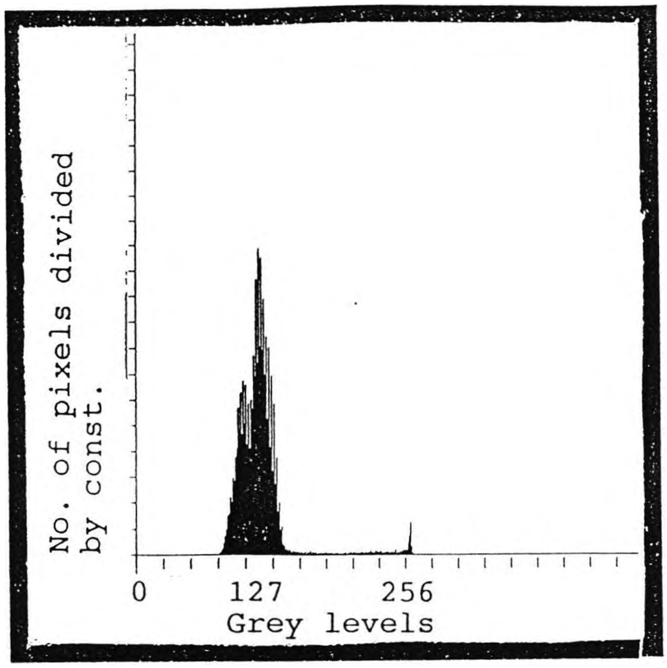


Fig. 5.5. The histogram of the central area of the Fig. 5.1.

compact and displayable. The main peak in the middle of the diagram covers those pixels in the background or the darker part of the image. The other peak which is much shorter is located at the far end of the histogram and related to those pixels which belong to the stripe itself. Between these two peaks a background of the small number of pixels can be traced, stretching all along the X axis. In practice it is found that the grey levels at the bottom right hand side of the main peak can produce favourable results. If grey levels further to the right of the main peak are used, many of the white pixels of the stripes would be absorbed in the background. If this is continued, at some stage gaps can occur along the stripes.

In segmenting stripes from the rest of the image paramount importance is given to obtaining data with little noise. In practice this is mainly achieved by the arrangements that are made to isolate the ambient light and stabilised illumination and reflection problems. By doing so segmentation is performed with a single thresholding operation and with no extra image processing operations and additional added time between the scans. This approach successfully delivered the required results, with the minimum amount of noise. However in a working environment it may be necessary to obtain data without such prior arrangements. Since image characteristics can change considerably with each of these factors, achieving a single thresholding level is not always feasible. One way to overcome this problem is to subdivide the image and find a different thresholding level for each area. This is known as adaptive thresholding and can be applied if the background varies from place to place, but remains almost constant locally. This technique was first used by Chow and Kaneko (Chow 1972) to outline some boundaries in an X-ray picture. Here, the original 256 by 256 image is subdivided into 7 by 7 regions with 50% overlap. This resulted in 49 regions each with 64 by 64 pixels. Then histograms of all regions are computed and those with two peaks (i.e. bimodal histograms which indicate the presence of a boundary) are selected and optimal thresholding carried out. Also in the same context Kittler and J. Illingworth (Kittler 1986) considered the problem of minimum error thresholding by directly averaging pixel classification error rate using either an exhaustive search or an iterative algorithm.

5.3 Extraction of the middle points from the stripe

The process of extracting the middle part of the stripe reads 512 horizontal lines of the image. For that the stripe is usually positioned such that for most of the cases a suitable, cross section can be achieved. Before reading takes place images are thresholded and converted into binary (i.e. the stripe in white and the rest of the image in black). Once the reading starts the search is to find pixels with the white value. On encountering such a point, pixels in its neighbourhood are also examined. If no more pixels of the same kind are found then it is concluded that the point is isolated noise and is ignored. If on the other hand several white pixels (i.e. more than one) are found consecutively in a row then the centre point is counted and stored.

In many situations, as in Fig.(5.6) some parts of a stripe can be obstructed by other stripes. Because of this once the

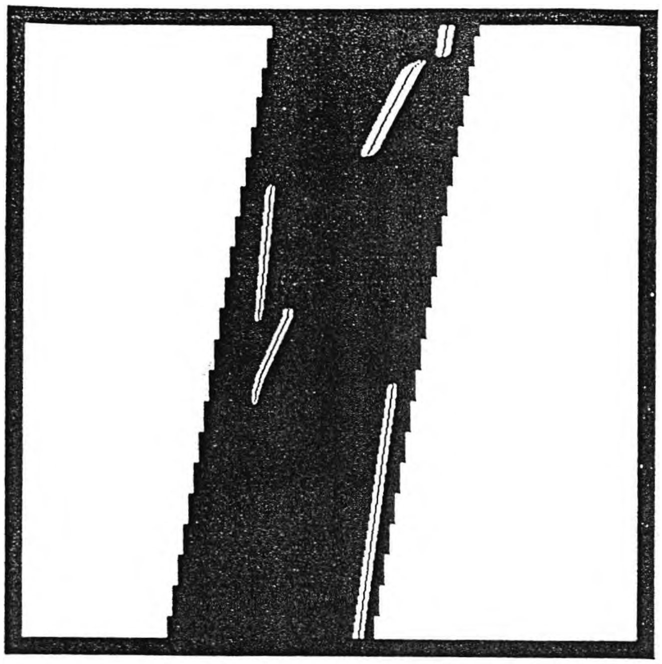


Fig. 5.6 If the image is read horizontally from right to left then second stripe from the bottom would be obstracted by the bottom stripe. Also it is obstructing the third stripe from the bottom.

first point on each line is extracted, there may still be one more point further on. In general for each object no more than two points can be detected on each line. For this, two arrays are used to store the coordinates of the middle points. Before the readings begin all the elements of the arrays are marked by a constant (i.e. such as zero) to indicate empty spaces.

Now consider Fig.(5.6) where the origins of the X and Y axis are chosen to be at the bottom left corner of the image. Suppose that the value of the Y coordinate corresponds to the number of lines across the image. Also suppose that readings have been taken from the right to left of the image. In this image the three stripes at the lower part of the image are referred to as first (i.e. from the bottom), second and third stripes respectively. As it can be seen the upper part of the second stripe occludes the lower part of the third stripe . At the same time the lower part of the second stripe is obstructed by the upper part of the first stripe . The typical data where these stripes come to contact and obstruct each other, can be written for the first and second stripes as follows:

First array

Y	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115
X	350	350	351	351	352	353	353	354	**250**	**250**	**251**	**251**	**252**	**253**	**253**	**254**

Second array

Y	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115
X	**245**	**246**	**247**	**247**	**248**	**249**	**249**	**250**	0	0	0	0	0	0	0	0

Here part of the lower part of the second stripe is registered in the second array. This is because the elements of the first array are occupied by the upper part of the first stripe beforehand. To bring these two separated parts together the elements of the first array are run through and examined. Once a sudden jump is encountered, then the same element in the second array is examined. If it is found that their contents are in close range, then elements are exchanged appropriately.
The same can be said for the second and the third stripes where the typical data can be as follows:

First array:

Y	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145
X	280	280	281	281	282	282	283	284	284	285	**272**	**273**	**273**	**274**	**274**	**275**

Second array:

Y	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145
X	0	0	**267**	**268**	**269**	**269**	**270**	**270**	**271**	**272**	0	0	0	0	0	0

Similarly to the first case above, here, the upper part of the second stripe occludes the lower part of the third stripe. These separated parts are brought together by swapping related data between the two arrays. In the programme that is used two end points with a difference of up to four pixels are interpreted as the same lines and therefore swapped.

It is important to note that the same procedure can also be applied without actually thresholding the images. Images are thresholded from the grey level from which, stripes are located above it. For that, the process can be simply conducted by searching for pixels greater than thresholding value. Thresholding can be useful if the aim is to monitor the result of each image during the scanning process. Otherwise an overall reduction on scanning time can be achieved (approximately three

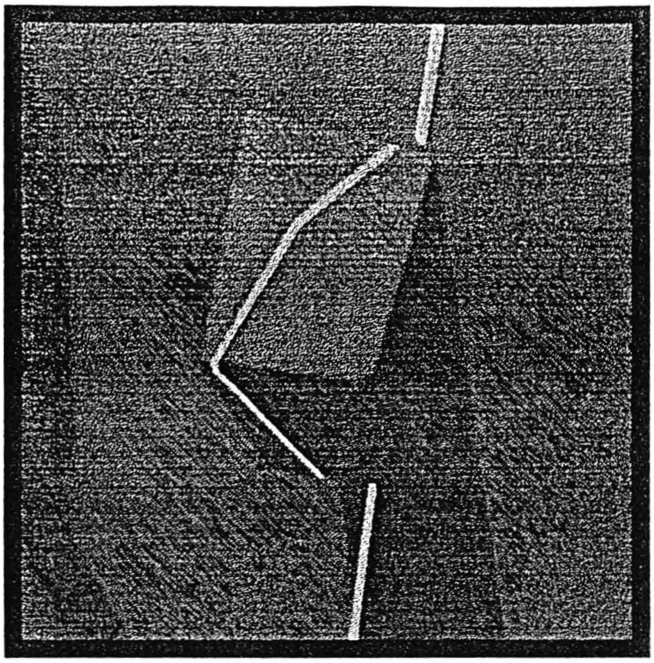


Fig. 5.7. The object and the stripe.

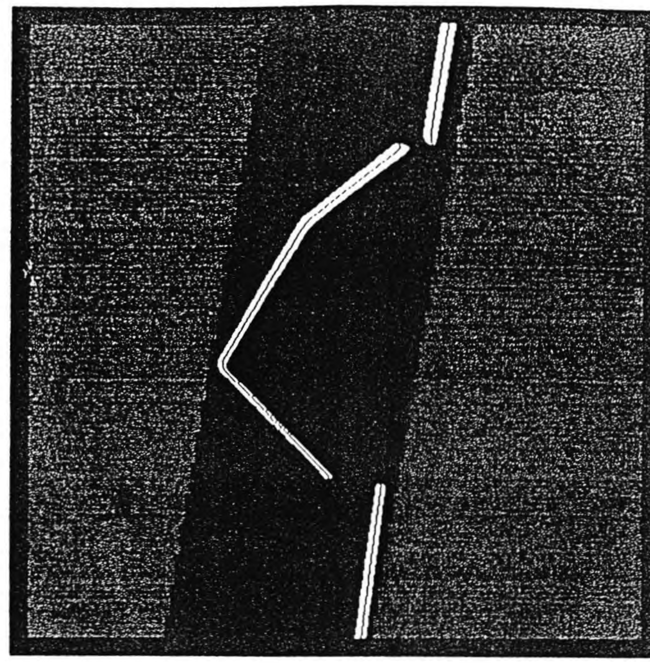


Fig. 5.8. Extraction of the middle part with thresholding the image.

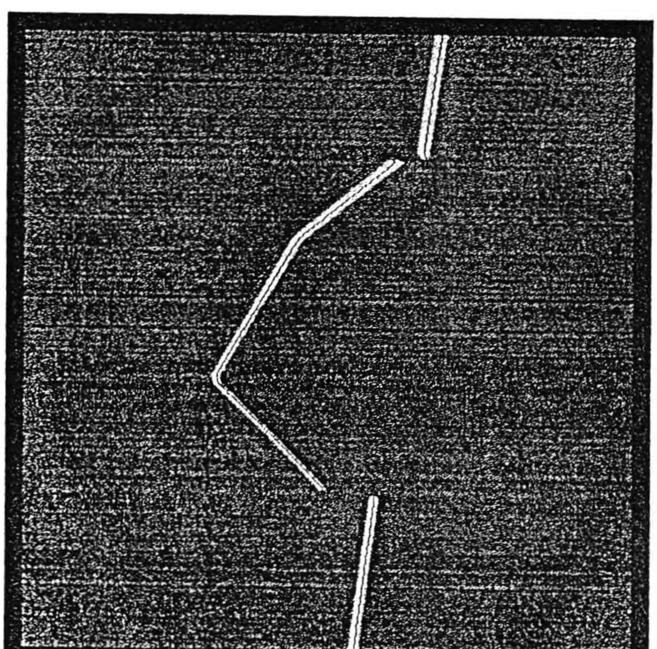


Fig. 5.9. Extraction of the middle part without thresholding the image.

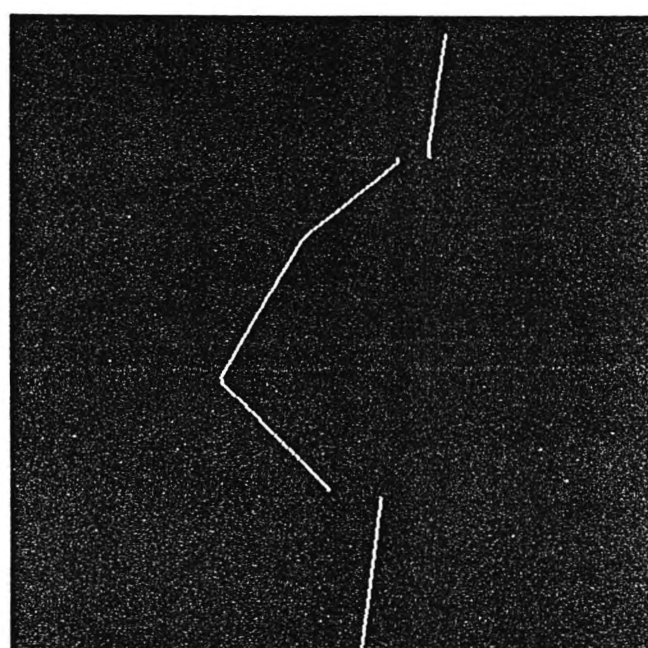


Fig. 5.10. The extracted middle part of the stripe.

minutes) if operations are performed without thresholding. An example of this is shown in Figs.(5.7 to 5.10) where extraction of the middle part is shown in both cases. D.K. Naidu and R.B.Fisher (Naidu 1992) provided a comparative analysis of five algorithms for determining the peak position of an observed laser stripe to sub-pixel accuracy. J. Mundy and B. Porter (Kanada 1987)discussed effects of reflectance on the location and symmetry of the centre of a Gaussian beam and measures such as normalising image intensity signals are suggested.

5.4 Corrections and noise reduction

In scanning objects, up to forty images may be required and proper care is taken to obtain them in satisfactory condition. Nevertheless it is essential to check for noise and undesired data during the course of the practice. In other words, the system has to be able to correct itself, since spurious data could lead into incorrect outcomes later on.

For that the extracted middle part of the stripes are examined for empty spaces or unwanted data. As demonstrated in Fig.(5.2) and Fig.(5.3) an unsuitable thresholding level can considerably alter the state of the images. Those examples show extreme cases, which with proper care are not likely to be experienced in practice. But there can be some occasions where a thresholding level slightly higher or lower than necessary is applied. This can result in some sporadic and remote dots and spikes (lower threshold) around the images which can interfere with processing and slip through as proper data. These can be detected and if necessary removed from the data. Equally minor gaps (high threshold) can be traced and amended if they are traced along the stripe.

In the case of spurious points, various filters and noise reduction algorithms with different sizes of windows can be employed across the image. Neighbourhood Averaging, Median Filtering and K-closest Averaging are some that can be mentioned. Their exact functions can be found in text books such as Gonzalez and Wintz (Gonzalez 1987), or in K. R. Castleman (Castleman 1979). It is a matter of skill and judgment to apply them in each particular application. However for the present application two main questions are asked before any algorithm is implemented. First, what is the basic character of the noise? And second, to what extent can the real data be mistaken for noise? For this it is assumed that, images are taken in the condition where no more than some sporadic noise in the shape of one or two pixels would occur. Also it is assumed that if more than two pixels are found, then the position is worth investigation. This is because in some cases the stripe itself can be as short or as narrow as a few pixels. Based on these assumptions, no specific noise reduction algorithm is found to be necessary. In other words spurious points are directly found and skipped, while line by line reading of images is taking place. For the second case (i.e. sorting out the good data from the noise), data is compared and corrected after the reading of the image is completed. In the following, some cases that are included in the programme are discussed.

The first case is the situation where a few points are significantly greater or smaller than their neighbouring points. As an example suppose the first and second arrays are occupied by the following data:

First array.

Y axis	100	101	102	103	104	105	106	107	108	109	110	111	112	113
X axis	250	250	251	252	252	253	**290**	**291**	**291**	255	256	256	256	257

Second array.

Y axis	100	101	102	103	104	105	106	107	108	109	110	111	112	113
X axis	0	0	0	0	0	0	253	254	254	0	0	0	0	0

Here it is assumed that reading is taking place horizontally line by line along the Y axis, and zeros represent empty elements. It can be seen that there are three elements in the first array which do not fit with the rest of the elements. At the same time the three elements in the second array, are in the same trend as the rest of the elements of the first array. This implies that, while the reading is taking place a spurious point is read first and its value is recorded in the first array. Then the reading is continued, and the proper data is recorded in the second array.

Also a similar situation can be found as follows:

First array.

Y axis	100	101	102	103	104	105	106	107	108	109	110	111	112	113
X axis	300	301	301	302	302	303	304	305	305	306	307	307	308	308

Second array.

Y axis	100	101	102	103	104	105	106	107	108	109	110	111	112	113
X axis	210	210	211	212	213	213	**280**	**281**	**281**	215	216	216	217	218

In this example the data in the first array is continuous and with no problem. In the second array ,however, a sudden fluctuation in the general trend of the data can be noticed. This can also be attributed to a spurious point encountered during the readings. For the first example the three unwanted data points are replaced with those in the second array. In the second example neighbouring elements are averaged and used as replacements.

The next case concerns the situation where the stripe is broken. This is illustrated in the following example:

First array.

Y axis	100	101	102	103	104	105	106	107	108	109	110	111	112	113
X axis	211	211	212	213	214	214	215	**0**	**0**	**0**	217	217	218	218

Second array.

Y axis	100	101	102	103	104	105	106	107	108	109	110	111	112	113
X axis	0	0	0	0	0	0	0	0	0	0	0	0	0	0

As can be noted there are three zeros in the area where otherwise continuous data can be expected. Here also neighbouring elements are averaged and used instead.

Then there is a case where one or two points are in the area where all other elements are empty. An example is as follows,

First array.

Y axis	100	101	102	103	104	105	106	107	108	109	110	111	112	113
X axis	0	0	0	0	0	0	0	221	222	0	0	0	0	0

Second array.

Y axis	100	101	102	103	104	105	106	107	108	109	110	111	112	113
X axis	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Here two isolated points can be detected with no other data around for comparison. They can equally be proper data or isolated noise. In the programme used this situation is interpreted as noise and the points are eliminated. This is because removing noise is found to be more important than two extra data points.

In practice, to detect these defects, first a one-dimensional mask with up to five elements is run along the arrays. Once a suspected area is located it is investigated according to the various parameters that are established during experiments. These include factors such as the acceptable difference between two neighbouring elements or the allowable number of empty elements. Then each element is eliminated, replaced or left intact accordingly. These corrections are carried out for no more than three adjacent points at a time. This is because involving greater number of points can rapidly create a new and sometimes complicated situation to investigate. Also parameters can increase considerably in number, and deciding on their value can become difficult. Because of that emphasis is given to obtaining satisfactory images in the first place since these arrangements can only deal with the relatively minor defects.

5.5 <u>Extraction of Edge Points</u>

Once extraction of the middle part of the stripe is completed the result is a set of single or a string of connected lines. Some of the end points of the lines and also the sharp turns along the connected lines are the indication of edge points.

To reconstruct the edges of the object it is necessary to extract every potential edge point from images. For this the splitting scheme introduced by Duda and Hart (Duda 1973) is employed. This is also known as the Divide and Conquer algorithm (Ballard 1982) which can segment a sequence of points. In this algorithm, first, two end points such as P and C in Fig.(5.12) are connected. Then the perpendicular distances between each point on the PABC and PC line are calculated. Next PABC is split from the point of maximum magnitude if it is found to be over a particular threshold.

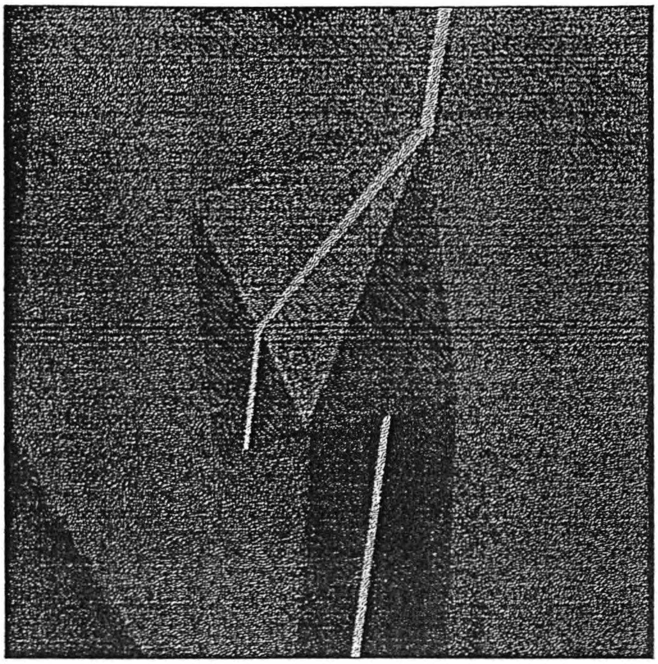


Fig. 5.11. The object and the stripe that is crossed it.

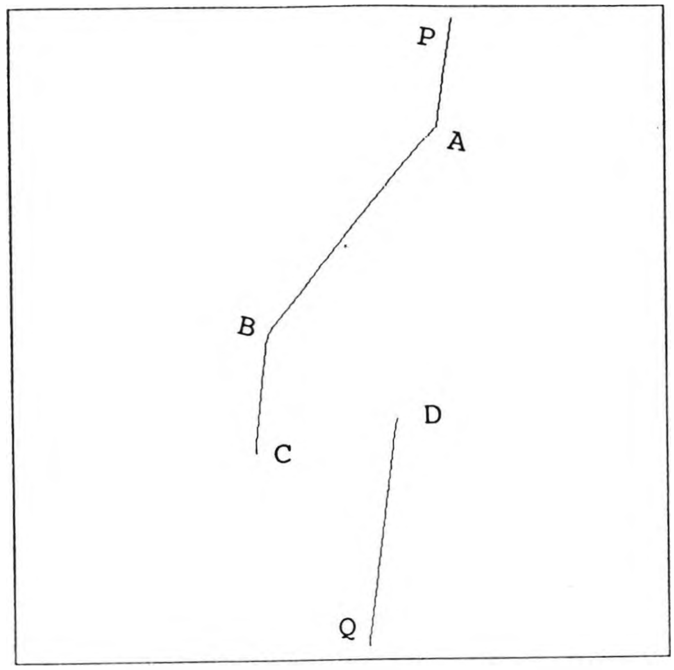


Fig. 5.12. The extracted middle part of the stripe.

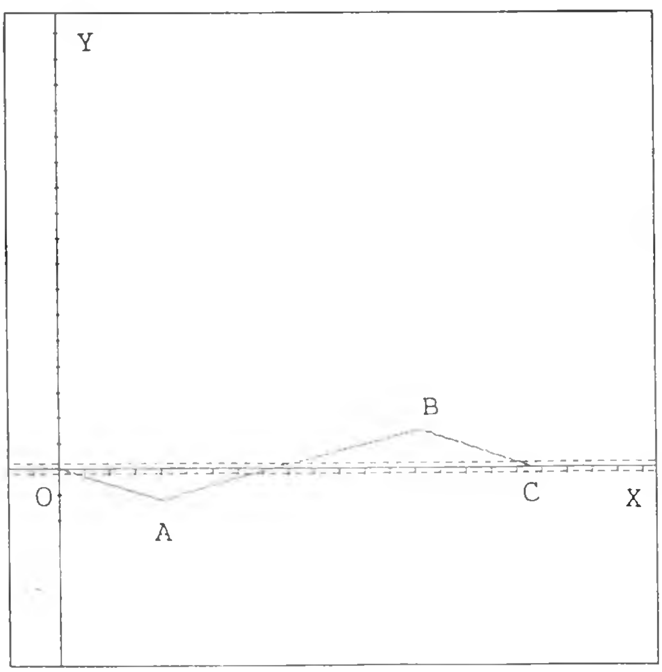


Fig. 5.13. Translating and rotating to X,Y axis and dividing the graph from it's peak point B.

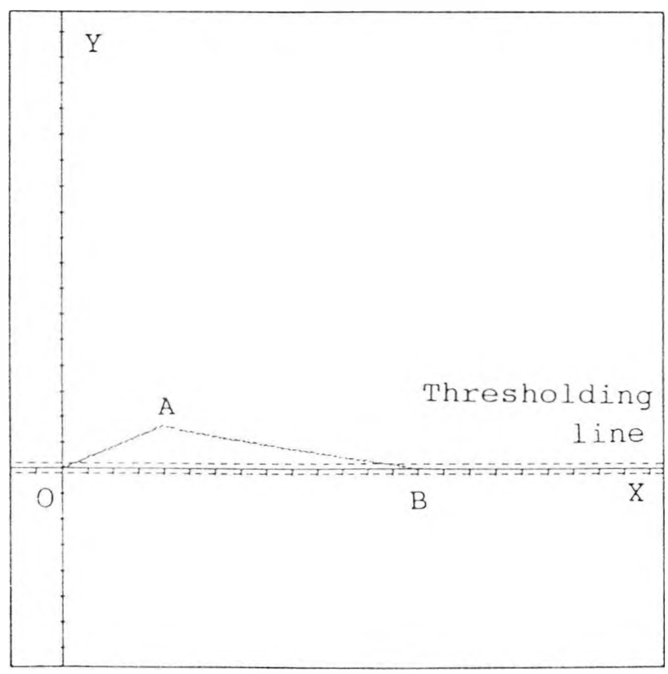


Fig. 5.14. The rest of the remaining graph after reflection against X axis.

Here, first the two end points of what could be a single line or string of connected lines are taken. Then the following steps are implemented.

1) Selecting an arbitrary X and Y axis and then translating one end to the origin.
2) Rotating the other end so that it ends up on the X axis.
3) Selecting an appropriate thresholding level parallel to the X axis.
4) If the graph (the connected lines) is found to be intersected by the thresholding line then it is divided from its maximum peak.
5) This is repeated recursively until all lines are found to be positioned between the thresholding line and the X axis.
6) The two end points of each line are then stored for the next stage of processing.

These procedures are sequentially demonstrated in Fig.(5.11) to Fig.(5.14). In Fig.5.11 an object is shown which is crossed by a stripe and the stripe is split into two parts. The upper part, crosses the object and consists of the three connected lines. The two end points of the line on the top of the object (i.e. B and C in Fig. 5.12) and the point at the bottom end of the middle line (i.e. A in the same figure) are points which are actually edge points of the object. But at this stage it is not possible to assess which points are edge points and which ones are on the surface of the table or are shadow points in the background (i.e. such as D in Fig. 5.12). For this all points regardless of their position, are collected.

In Fig.(5.11) and Fig.(5.12) the middle part of the stripe is extracted and the part with the three lines is translated and rotated to its position in Fig.(5.13). These lines in effect make a graph which passes below and above the X axis. Here a situation can occur where the whole graph is below the X axis and no intersection with the thresholding line above the X axis can be found. To deal with this situation a second thresholding line below the X axis is drawn (the dashed line parallel and below the X axis) . If no intersection from above is found, then the thresholding line below X axis is examined. If it is intersected, then the graph is reflected against the X-axis. In this way extraction of the information is confined to the thresholding line above the X axis.

If the graph intersects the thresholding line then this is the indication that it consists of more than one line .In this case the search is to find the peak point which is the highest point in the Y co-ordinate. Once this is detected, then the graph is divided into the two separate parts. Then each part is examined again, in case it is made of more lines. These procedures are illustrated in Fig.(5.13) and Fig.(5.14). Here, first the graph is divided from its peak point B above the X axis. The part on the right is made of a line and it cannot be intersected any more. For this there is no more division and its two end points can be stored for further use. But the part on the left still consists of two lines. Because of that, first it is reflected against the X axis and then it is divided once again from its peak point A.

There is no direct way to determine the value for the tolerance line. Two connected lines (depending on their length and angle) can exhibit quite considerable variation in the shape and the way that they are bonded together. To obtain an initial rough value, first various objects in different positions are examined. Then by trial and error, and by running the programme many times, an appropriate level is selected. Choosing too low a level of thresholding can lead to a gap in the line which in turn can result in incorrect information about the scene. Equally, choosing too high a level of thresholding can affect the ability to distinguish one line from the other. This can lead to having two separate lines remaining joined together.

In practice it is found that a line can fluctuate up to 0,1,2,3 pixels above and below the X axis. For this a slightly higher level of 4 is chosen to be the limit.

An example is given in Table (5.1) where typical data for the image such as Fig.(5.11), is as follows,

Scan N-1	Scan N	Scan N+1
....	point (A) Ax Ay N	....
....	point (B) Bx By N	....
....	point (C) Ax Ay N	....
....	point (D) Bx By N	....

EXTRACTED EDGE POINTS(pixels)			EXTRACTED EDGE POINTS (pixels)			EXTRACTED EDGE POINTS (pixels)			EXTRACTED EDGE POINTS (pixels)		
IMAGE	X	Y	IMAGE	X	Y	IMAGE	X	Y	IMAGE	X	Y
1	238	236	10	227	308	18	147	361	27	215	168
1	235	240	10	247	193	1000	1000	1000	27	157	233
1000	1000	1000	10	192	260	19	261	155	27	142	318
2	244	231	10	178	318	19	215	363	27	209	396
2	244	233	1000	1000	1000	19	254	153	1000	1000	1000
2	244	234	11	253	191	19	170	250	28	259	153
2	240	247	11	226	315	19	145	356	28	202	178
2	239	231	11	247	189	1000	1000	1000	28	158	227
2	224	250	11	191	256	20	262	150	28	143	312
1000	1000	1000	11	173	327	20	215	367	28	208	401
3	245	229	1000	1000	1000	20	256	147	1000	1000	1000
3	239	255	12	254	186	20	167	250	29	259	157
3	240	226	12	224	323	20	143	352	29	191	187
3	212	260	12	249	184	1000	1000	1000	29	159	222
1000	1000	1000	12	188	256	21	263	146	29	143	306
4	246	224	12	168	335	21	214	371	29	207	405
4	238	262	1000	1000	1000	21	256	143	1000	1000	1000
4	241	222	13	255	181	21	165	247	30	257	162
4	208	268	13	223	330	21	142	348	30	179	197
1000	1000	1000	13	249	180	1000	1000	1000	30	160	219
5	247	219	13	185	255	22	264	142	30	145	302
5	236	270	13	163	343	22	213	375	30	206	409
5	242	217	1000	1000	1000	22	257	139	1000	1000	1000
5	208	258	14	256	176	22	162	248	31	257	166
5	202	277	14	221	338	22	140	343	31	166	207
1000	1000	1000	14	250	175	1000	1000	1000	31	145	297
6	248	215	14	183	254	23	265	138	31	205	413
6	234	278	14	159	352	23	212	379	1000	1000	1000
6	243	212	1000	1000	1000	23	257	135	32	255	171
6	206	257	15	257	172	23	160	245	32	160	216
6	197	285	15	219	346	23	139	338	32	147	291
1000	1000	1000	15	251	171	1000	1000	1000	32	204	418
7	249	210	15	180	254	24	264	136	1000	1000	1000
7	233	285	15	154	360	24	255	138	33	254	176
7	244	208	1000	1000	1000	24	157	245	33	159	224
7	203	257	16	258	167	24	139	333	33	148	287
7	192	293	16	217	353	24	212	383	33	204	422
1000	1000	1000	16	250	169	1000	1000	1000	1000	1000	1000
8	250	205	16	178	252	25	263	140	34	253	181
8	230	293	16	150	367	25	241	148	34	157	232
8	245	203	1000	1000	1000	25	155	243	34	148	283
8	201	256	17	259	163	25	140	328	34	201	424
8	187	302	17	218	354	25	210	387	34	201	425
1000	1000	1000	17	254	161	1000	1000	1000	34	201	430
9	251	200	17	175	252	26	262	144	1000	1000	1000
9	229	300	17	148	365	26	226	159	35	252	185
9	246	198	1000	1000	1000	26	156	238	35	157	239
9	195	260	18	260	159	26	141	322	35	149	278
9	182	310	18	217	359	26	210	392	35	201	423
1000	1000	1000	18	254	157	1000	1000	1000	35	201	425
10	252	195	18	173	249	27	261	148	35	201	430
next column			next column			next column			next page		

Table (5.1) Edge points extracted from each of the 40 consecutive images (continue next page).

EXTRACTED EDGE POINTS(pixels)			EXTRACTED EDGE POINTS (pixels)			EXTRACTED EDGE POINTS (pixels)			EXTRACTED EDGE POINTS (pixels)		
IMAGE	X	Y	IMAGE	X	Y	IMAGE	X	Y	IMAGE	X	Y
1000	1000	1000	36	201	429	1000	1000	1000	39	248	206
36	251	191	1000	1000	1000	38	249	200	39	202	417
36	155	247	37	250	195	38	202	418	1000	1000	1000
36	150	273	37	153	255	38	202	421	40	247	210
36	201	421	37	151	268	38	202	425	40	202	416
36	201	424	37	202	420	1000	1000	1000	1000	1000	1000
next column			next column			next column					

Table (5.1) Edge points extracted from each of the 40 consecutive images.

Here the coordinates of each point come with the number of the scan. Also two end points at the top and bottom of the screen (i.e. points P and Q in Fig. 5.12) are not included because they are positioned such that they cannot be edge points.

Finally there are some special cases which need special attention. As an example if one of the lines is found to be parallel and above to the X axis then it is not possible to find a peak for it. In this case it is examined both from the right and left of the graph and then divided appropriately. Also if one of the parts is found to be no more than a few pixels then it is eliminated as noise.

5.6 <u>Some Problems in Practice</u>

In the calibration process the resulting off line errors due to measurement are outlined. However the main source of errors are due to the on line location of points in the images rather than the calibration itself. One of these cases is illustrated in Fig.(5.15) and Fig. (5.16). In the first image, at the junction between two lines a new line is about to emerge. But it is not clear enough to be detected yet. In the next scan (i.e. Fig.(5.16)) it emerges and moves quite a distance in the image. This relatively large move can constitute quite a space in the real world. It also represents the first data point of a new edge. Because of its sudden appearance, it can be quite a distance from where the edge actually starts in the object.

Now consider Fig.(5.6) and the second stripe from the top. Here, the middle line is extracted with the slight bend at the upper part of the stripe where the stripe becomes narrow. Similar effects can also occur because images are read horizontally or

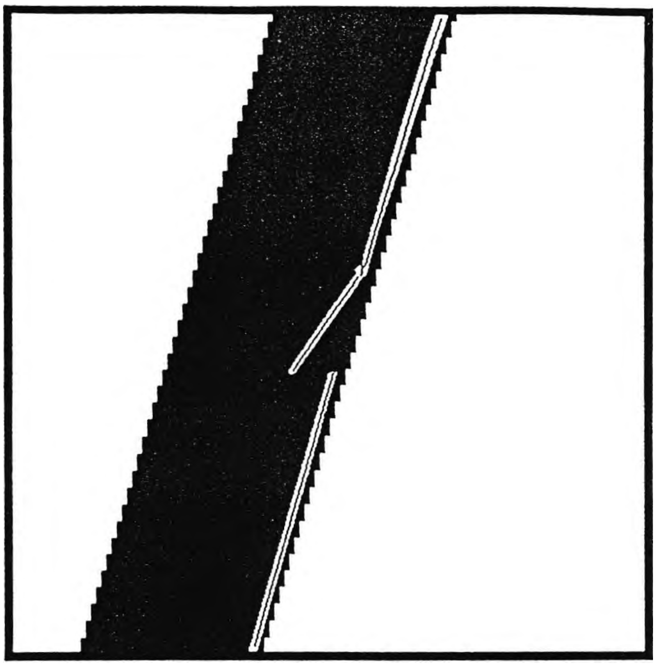


Fig. 5.15 A line is about to emerge from the junction of two stripes.

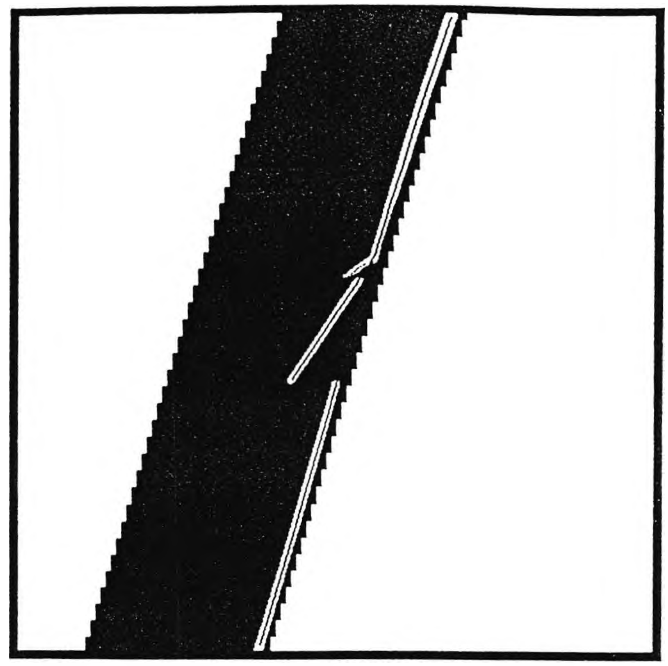


Fig. 5.16. A line is emerged in the next scan and is moved quite a distance in the image.

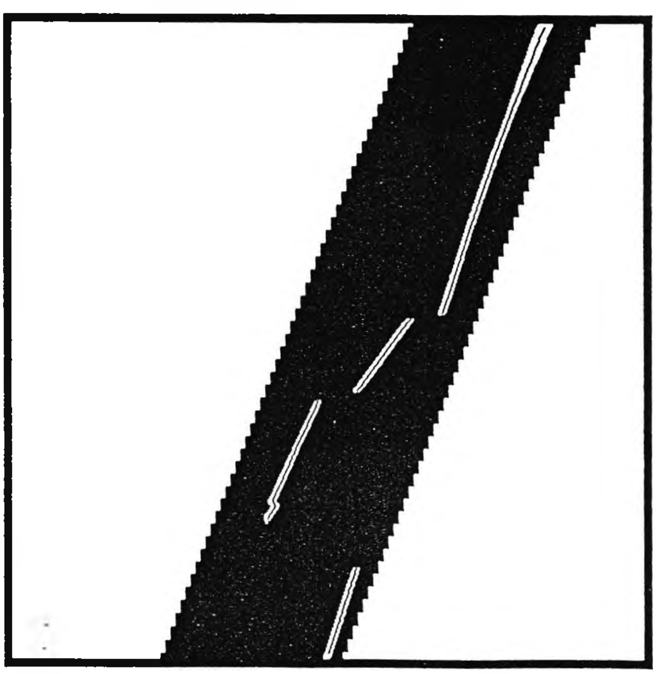


Fig 5.17. The second stripe from the bottom is made of the two separate stripes.

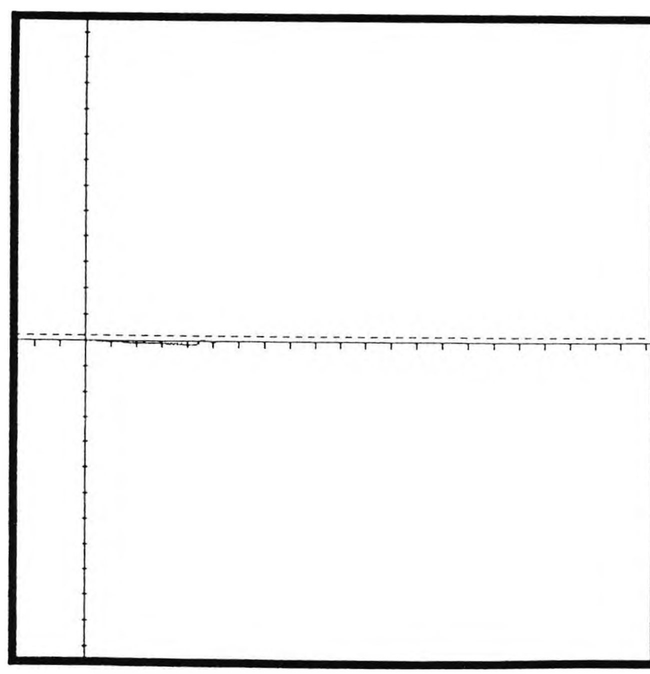


Fig 5.18. The two lines can not be separated by tolerance line.

the stripe is not clearly and sharply cut. In these cases bends can be short enough to remain undetected. Since approximately every three pixels constitute 1 mm in the real world, errors up to 2 mm can occur from them.

Another case concerns the situation where two stripes merge together at one stage of scanning. An example is given in Fig.(5.17) where the second stripe from the bottom is actually made of two stripes. Once it is transferred along the X axis (Fig.5.18.) they are too close to be distinguished. Four edge points can be extracted from these two lines. Since information is collected from consecutive images, missing two points can disrupt the sequence and result in incorrect data. A typical data set for the points between two stripes is as follows,

Y axis	100	101	102	103	104	105	106	107	108	109	110	111
X axis	200	200	201	202	202	203	**202**	**201**	**200**	201	202	202

As can be noted a short change of direction in the general trend of the numbers can indicate this situation. To counter this problem, a special algorithm is included so that it compares and examines neighbouring elements along the path and separates them whenever a similar situation occurs.

These cases are not significant on their own but occur in practice and to some extent can influence the results. These are dealt with individually by adding necessary algorithms for each case and consequently most of the inadequacies in this part are resolved.

5.7 Summary and Comment

In this part, the initial row materials (i.e. images with stripes) are processed and converted into compact and useful data. As in many vision systems, a considerable amount of computation and preprocessing is employed to collect the required data. The outline of operations is illustrated in Fig(5.19). Throughout these operations, emphasis is given to obtaining reliable data at the expense of running time. For this objects are intensely scanned and no change in the original size of images are made. In this process images are thresholded, stripes are separated and their middle parts are extracted. Then corrupted data is removed or corrected and potential edge points are extracted and stored.

The central objective of this section was to obtain necessary data so that edges can be constructed. For that, illumination and reflection factors, together with ambient light are stabilised. Consequently this permitted a straightforward thresholding operation to analyse the images. The necessity of having images, with relatively low level of noise resulted in this head on solution. This is particularly so when up to forty images in satisfactory condition are needed.

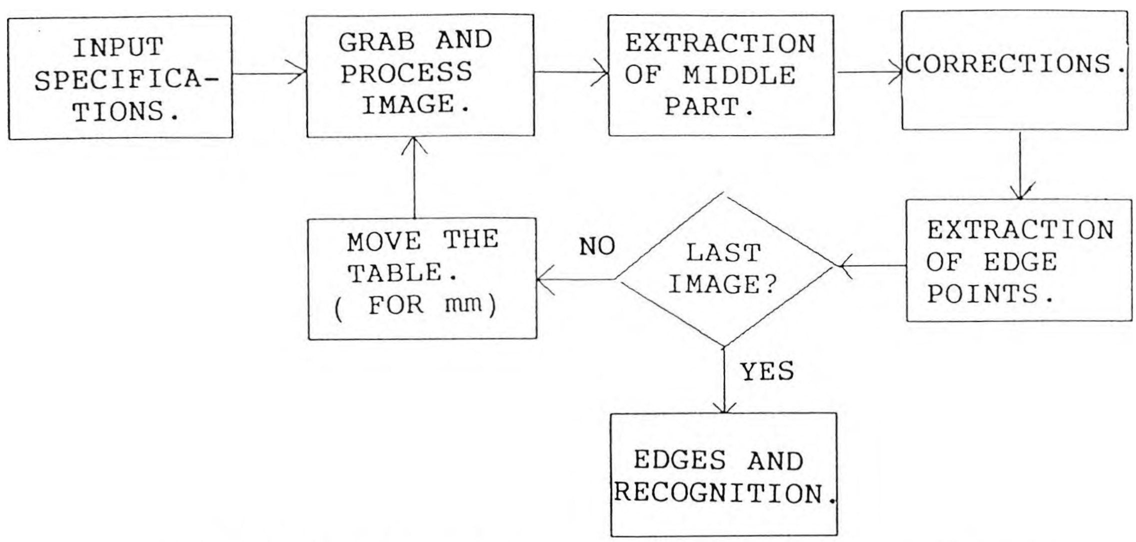


Fig.(5.19) Functions of preprocessing section.

In the condition where the experimental surroundings are isolated various edge detection operations may also be tried. However, edge detectors may create additional lines or fail at some points. Since in many cases distinguishing proper data from noise is difficult edge detectors may not necessarily offer a better solution to thresholding.

If the surfaces of the object contain patterns or consist of different reflective characteristics or colour then laser beams may be employed. By using laser beams many variations in the surface properties may be countered. In the cases where different areas of the surface can be segmented, then they may be analysed separately and according to the characteristic of each part.

CHAPTER SIX

EXTRACTION OF EDGES AND MATCHING PROCESS

In this chapter the edge points extracted in the preprocessing section are linked. At this stage the information is still kept in two dimensions. Next, points are converted into their corresponding three dimensional coordinates. Then data strings containing edge points are analysed and the edges within them are extracted. The edges are then compared and classified appropriately. At the next stage actual matching operations are performed. Here, each time round, edges from the object and model are selected. Then they are operated on and are matched. But before that, angles, lengths, and the connectivity of the lines are examined. If it is found that there is no prospect of matching them, then they are excluded from operations. For each successful match, the related TTM matrix is calculated and stored. At the end, the model with the highest number of similar matrices is selected to be the one identical to the object in the scene.

6.1 Linking Edge Points

The result of the preprocessing section is data which contains coordinates of the potential edge points obtained from images taken on each scan. The next step is to link related points and extract edges. A typical edge linking method is described by Y. Shirai (Shirai 1987) where linking is conducted in three dimensions. First, various edges such as jump edges, discontinuous edges, corner edges, or occluding edges are specified and the neighbouring elements are defined. Next depending on the direction of the edge points, candidates in neighbouring elements are examined and appropriate edge points are selected and linked. Another approach is taken by Boyter and Aggarwal (Boyter 1986) where a row segment and a column segment of edge points are randomly selected. Then the four end points of these segments are tested in order to find whether they are coplanar or not. For the points which are found to be coplanar the equation of the plane is calculated. This is repeated until all edge points are assigned to a plane. Next, in order to find straight lines the Hough transform is applied to the edge points associated with each plane.

In the present approach, however, edge points are linked by drawing circles around them. Starting from a small radius, circles around the edge points are grown. This is continued until an appropriate edge point from the next image falls inside appropriate circle. Then from the string of data obtained in this way edges are constructed. The immediate advantage of this approach is that points can be linked directly in two dimensions and no knowledge of the direction of lines is required. Also, measures such as fitting points into the equations of planes are not required.

As shown in table (5.1) points are grouped according to the number of the image (or scan) that they belong to. The task is to relate each point to its corresponding point in the previous image and in the next one. For this, consider two successive stripes which are drawn in Fig.(6.1a). Here for the sake of clarity the positions of the two lines are exaggerated. However, it is quite clear that points A,B,C,D,E are related to points A',B',C',D',E' respectively.

In order to link them, first, a circle is drawn around A,B,C..etc.. The radius of the circle is chosen to be quite small and not greater than two adjacent pixels. Then one of the points (i.e. such as A) is taken and its distance from the other points from the next stripe are calculated (i.e. from A',B',C'.etc.). If the length between A and one of these points is found to be less or equal to the radius of the circle, then it

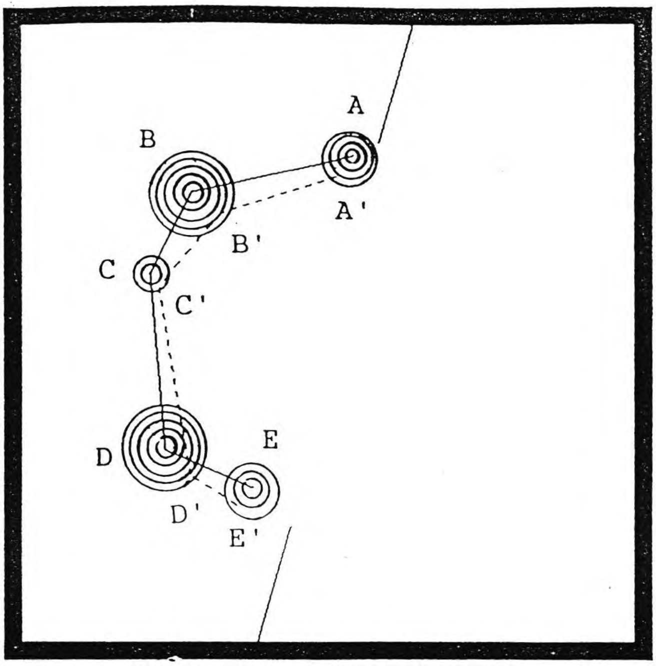


Fig.(6.1a). Selecting correspondent edge points from two consecutive stripe lines.

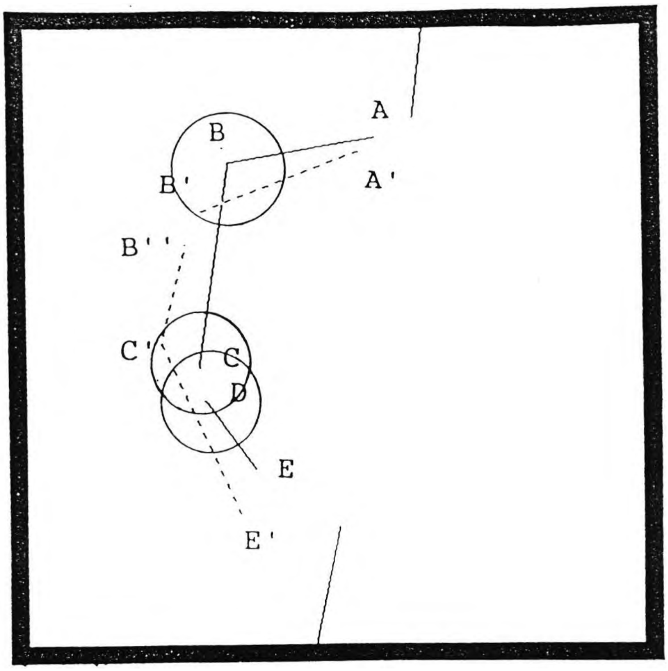


Fig.(6.1b). Edge points with the more than one corresponding.

is concluded that they correspond to each other. In this case they are stored in a data string and are excluded from the rest of the search. Then the same procedure is repeated for the rest of the remaining points (i.e. B,C,..etc.). At the end of the round, there may or may not be any two points found to be coupled. The next round starts with the radius of the circle slightly increased. Then the above procedure is repeated once more, this time with the slightly increased radius. As the radius is increased, more and more points fall inside the circles.

Now consider the first and the second images at the beginning of the process. Here for any two points that are coupled a new data string needs to be made. This is because for the first two images, no data string already exists. But once the procedure is under way the situation is different. As an example suppose two related points from the second and third images are coupled. In this situation the point from the second image is already positioned in the respective data string. Then the point from the third image needs to be directed to the data string where the point from the second image is present and no new data string is required. By repeating this process for the rest of the images all related edge points can be linked together.

In coupling points from two consecutive images the following possibilities can occur.

1) For each edge point in an image there is a candidate point in the next one.
2) A point from the first image has two or more candidates in its vicinity from the second one.
3) Two or more points from the first image have only one point in their immediate surrounding from the next image.

An example of the first case is illustrated in Fig(6.1a) where for each point a corresponding counterpoint exists. In this condition all points would eventually couple. An example for second cases, is provided in (Fig.6.1b) where B can couple with both points B' and B'' . As the circle around B increases, one of these two points would fall inside the circle first (i.e. such as B'). In this case B' would be coupled and stored along with B. But for B'' a new data string would need to be created and it would be the first element of it. In the Fig.(6.1b) also an example for the third case is given where both D and C can potentially couple with C'. As the radius of circles around C and D increases one of them would fall inside the circle first (i.e. such as C). In this case C' would be coupled and stored along with C but the data string associated with D would be terminated. The process of increasing the radius is continued until either no point is left or the radius reaches its limit. Setting a limit for the circles is necessary because, two consecutive edge points can only be displaced for a limited space and at some stage the process has to be stopped. Each radius is allowed to grow to its maximum limit of thirty pixels to cover all possibilities.

In the course of linking edge points two problems can occur. The first is when a spurious point originating from one of the images slips through and passes on as a proper edge point. In these cases there is a possibility that it can couple and enter into the data string before the actual edge point can do so. Each element of the data string determines the next element after it. Because of this, entering the wrong data may result in having a whole string of incorrect data and may result in having broken edges at the end. Furthermore if from one of the images a relevant point is not extracted, then the termination of an otherwise continuous chain of data can occur. This in turn can result in having a poor representation of an edge or some non existent edges. Countering this problem very much depends on obtaining relatively noise free data and the accuracy with which edge points are extracted. For that, in the preprocessing section considerable care and emphasis is given to extract reliable data.

The second problem is concerned with the situation where more than one point can enter into a circle simultaneously. In order to prevent this, the radius is increased by a smaller amount. In the programme that is used real numbers rather than integers are used and the rate is chosen to be less than the distance between two successive pixels.

Once these procedures for all the images are completed the result is several data strings. An example of this is shown in Table(6.1) where various data strings obtained from linking the edge points of table(5.1) is illustrated. Here, each element consists of the X and the Y coordinates in the images and the number of the scan they belong to. Here, data string (C) spans all forty images and contains one point from each scan. In contrast, data string (D) or (L) contain only one element, which most possibly comes from the spurious point within the image. There are also some data strings which span not all of the images such as (A),(B),(F) or contain only few points such as (H),(K).

6.2 Conversion to Three Dimensions

By this stage edge points that are linked are in two dimensions and their coordinates are positions in the images. To work out the coordinates of the edges as they appear in the real world it is necessary to convert them into their respective three dimensional values. Converting edge points at this stage is particularly suitable because:

1) The number of points involved is considerably less than when the middle part of the stripes is extracted.
2) Linking edge points in data strings can be conducted in two dimensions rather than in three dimensions.

data strings in 2D, (in pixels)			data strings in 2D,(in pixels)			data strings in 2D, (in pixels)			data strings in 2D,(in pixels)		
IMAGE	X	Y	IMAGE	X	Y	IMAGE	X	Y	IMAGE	X	Y
(A) 1	238	236	21	214	371	36	251	191	9	182	310
2	239	231	22	213	375	37	250	195	10	178	318
3	240	226	23	212	379	38	249	200	11	173	327
4	241	222	24	212	383	39	248	206	12	168	335
5	242	217	25	210	387	40	247	210	13	163	343
6	243	212	26	210	392	1001	1001	1001	14	159	352
7	244	208	27	209	396	(D) 2	244	233	15	154	360
8	245	203	28	208	401	1001	1001	1001	16	150	367
9	246	198	29	207	405	(E) 2	244	234	17	148	365
10	247	193	30	206	409	1001	1001	1001	18	147	361
11	247	189	31	205	413	(F) 2	224	250	19	145	356
12	249	184	32	204	418	3	212	260	20	143	352
13	249	180	33	204	422	4	208	268	21	142	348
14	250	175	34	201	424	5	208	258	22	140	343
15	251	171	35	201	423	6	206	257	23	139	338
16	250	169	36	201	424	7	203	257	24	139	333
17	254	161	1001	1001	1001	8	201	256	25	140	328
18	254	157	(C) 2	244	231	9	195	260	26	141	322
19	254	153	3	245	229	10	192	260	27	142	318
20	256	147	4	246	224	11	191	256	28	143	312
21	256	143	5	247	219	12	188	256	29	143	306
22	257	139	6	248	215	13	185	255	30	145	302
23	257	135	7	249	210	14	183	254	31	145	297
24	255	138	8	250	205	15	180	254	32	147	291
25	241	148	9	251	200	16	178	252	33	148	287
26	226	159	10	252	195	17	175	252	34	148	283
27	215	168	11	253	191	18	173	249	35	149	278
28	202	178	12	254	186	19	170	250	36	150	273
29	191	187	13	255	181	20	167	250	37	151	268
30	179	197	14	256	176	21	165	247	1001	1001	1001
1001	1001	1001	15	257	172	22	162	248	(H) 34	201	425
(B) 1	235	240	16	258	167	23	160	245	35	201	425
2	240	247	17	259	163	24	157	245	36	201	421
3	239	255	18	260	159	25	155	243	37	202	420
4	238	262	19	261	155	26	156	238	38	202	421
5	236	270	20	262	150	27	157	233	1001	1001	1001
6	234	278	21	263	146	28	158	227	(J) 34	201	430
7	233	285	22	264	142	29	159	222	35	201	430
8	230	293	23	265	138	30	160	219	36	201	429
9	229	300	24	264	136	31	166	207	1001	1001	1001
10	227	308	25	263	140	32	160	216	(K) 38	202	418
11	226	315	26	262	144	33	159	224	39	202	417
12	224	323	27	261	148	34	157	232	40	202	416
13	223	330	28	259	153	35	157	239	1001	1001	1001
14	221	338	29	259	157	36	155	247	(L) 38	202	425
15	219	346	30	257	162	37	153	255	1001	1001	1001
16	217	353	31	257	166	1001	1001	1001	0	0	0
17	218	354	32	255	171	(G) 5	202	277	0	0	0
18	217	359	33	254	176	6	197	285	0	0	0
19	215	363	34	253	181	7	192	293	0	0	0
20	215	367	35	252	185	8	187	302	0	0	0
next column			next column			next column					

Table (6.1) Data strings are connecting related points from consecutive images.

data strings in 2D, (in pixels)			data strings in 3D, (in mm)		
SCAN	X	Y	X	Y	Z
(A) 1	238	236	1	32	3
2	239	231	2	32	3
3	240	226	3	31	3
4	241	222	4	30	3
5	242	217	5	29	3
6	243	212	6	28	3
7	244	208	7	28	3
8	245	203	8	27	3
9	246	198	9	26	3
10	247	193	10	25	3
11	247	189	11	25	3
12	249	184	12	24	3
13	249	180	13	23	3
14	250	175	14	22	3
15	251	171	15	22	3
16	250	169	16	21	2
17	254	161	17	20	3
18	254	157	18	19	3
19	254	153	19	19	2
20	256	147	20	18	3
21	256	143	21	17	2
22	257	139	22	17	2
23	257	135	23	16	2
24	255	138	24	16	4
25	241	148	25	18	8
26	226	159	26	20	12
27	215	168	27	21	15
28	202	178	28	22	17
29	191	187	29	24	20
30	179	197	30	25	23
1001	1001	1001	1001	1001	1001
(B) 1	235	240	1	33	2
2	240	247	2	34	0
3	239	255	3	36	1
4	238	262	4	37	1
5	236	270	5	38	1
6	234	278	6	40	1
7	233	285	7	41	1
8	230	293	8	42	0
9	229	300	9	43	0
10	227	308	10	45	0
11	226	315	11	46	1
12	224	323	12	47	0
13	223	330	13	48	1
14	221	338	14	50	0
15	219	346	15	51	0
16	217	353	16	52	0
17	218	354	17	52	1
18	217	359	18	54	1
19	215	363	19	54	0
20	215	367	20	55	1

continue on the next two columns

data strings in 2D, (in pixels)			data strings in 3D, (in mm)		
SCAN	X	Y	X	Y	Z
21	214	371	21	56	0
22	213	375	22	56	0
23	212	379	23	57	0
24	212	383	24	58	1
25	210	387	25	59	0
26	210	392	26	59	1
27	209	396	27	60	1
28	208	401	28	61	1
29	207	405	29	62	1
30	206	409	30	63	0
31	205	413	31	63	0
32	204	418	32	64	0
33	204	422	33	65	1
34	201	424	34	65	2
35	201	423	35	65	2
36	201	424	36	65	2
1001	1001	1001	1001	1001	1001
(C) 2	244	231	2	32	1
3	245	229	3	31	1
4	246	224	4	30	1
5	247	219	5	30	1
6	248	215	6	29	1
7	249	210	7	28	1
8	250	205	8	27	1
9	251	200	9	27	1
10	252	195	10	26	1
11	253	191	11	25	1
12	254	186	12	24	1
13	255	181	13	23	1
14	256	176	14	23	1
15	257	172	15	22	1
16	258	167	16	21	1
17	259	163	17	20	1
18	260	159	18	20	1
19	261	155	19	19	1
20	262	150	20	18	1
21	263	146	21	18	1
22	264	142	22	17	1
23	265	138	23	16	1
24	264	136	24	16	1
25	263	140	25	17	0
26	262	144	26	17	0
27	261	148	27	18	0
28	259	153	28	19	0
29	259	157	29	19	0
30	257	162	30	20	0
31	257	166	31	21	0
32	255	171	32	22	2
33	254	176	33	23	0
34	253	181	34	23	0
35	252	185	35	24	2

continue on the next page

Table (6.2) Data strings and their converted values to three dimensions (continue next page).

data strings in 2D, (in pixels)			data strings in 3D, (in mm)			data strings in 2D, (in pixels)			data strings in 3D, (in mm)		
SCAN	X	Y	X	Y	Z	SCAN	X	Y	X	Y	Z
36	251	191	36	25	0	6	197	285	6	40	13
37	250	195	37	26	0	7	192	293	7	41	14
38	249	200	38	27	0	8	187	302	8	43	14
39	248	206	39	28	0	9	182	310	9	44	15
40	247	210	40	28	0	10	178	318	10	45	17
1001	1001	1001	1001	1001	1001	11	173	327	11	46	17
(D) 2	244	233	2	32	1	12	168	335	12	48	18
1001	1001	1001	1001	1001	1001	13	163	343	13	49	19
(E) 2	244	234	2	32	1	14	159	352	14	50	20
1001	1001	1001	1001	1001	1001	15	154	360	15	51	21
(F) 2	224	250	2	35	5	16	150	367	16	53	22
3	212	260	3	36	10	17	148	365	17	52	24
4	208	268	4	37	9	18	147	361	18	51	23
5	208	258	5	36	10	19	145	356	19	50	24
6	206	257	6	35	12	20	143	352	20	50	25
7	203	257	7	35	13	21	142	348	21	49	27
8	201	256	8	35	12	22	140	343	22	48	28
9	195	260	9	36	14	23	139	338	23	47	27
10	192	260	10	36	15	24	139	333	24	46	29
11	191	256	11	35	17	25	140	328	25	46	29
12	188	256	12	35	18	26	141	322	26	45	29
13	185	255	13	34	19	27	142	318	27	44	29
14	183	254	14	34	18	28	143	312	28	43	29
15	180	254	15	34	19	29	143	306	29	42	28
16	178	252	16	34	20	30	145	302	30	41	29
17	175	252	17	34	21	31	145	297	31	41	29
18	173	249	18	33	22	32	147	291	32	40	29
19	170	250	19	33	24	33	148	287	33	39	29
20	167	250	20	33	25	34	148	283	34	38	29
21	165	247	21	33	26	35	149	278	35	38	29
22	162	248	22	33	27	36	150	273	36	37	29
23	160	245	23	32	26	37	151	268	37	36	29
24	157	245	24	32	27	1001	1001	1001	1001	1001	1001
25	155	243	25	32	28	(H) 34	201	425	34	65	2
26	156	238	26	31	28	35	201	425	35	65	2
27	157	233	27	30	29	36	201	421	36	64	2
28	158	227	28	29	28	37	202	420	37	64	2
29	159	222	29	29	28	38	202	421	38	64	2
30	160	219	30	28	29	1001	1001	1001	1001	1001	1001
31	166	207	31	26	28	(J) 34	201	430	34	66	0
32	160	216	32	28	28	35	201	430	35	66	0
33	159	224	33	29	29	36	201	429	36	66	0
34	157	232	34	30	28	1001	1001	1001	1001	1001	1001
35	157	239	35	31	29	(K) 38	202	418	38	64	2
36	155	247	36	33	29	39	202	417	39	64	2
37	153	255	37	34	29	40	202	416	40	64	2
1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001
(G) 5	202	277	5	39	12	(L) 38	202	425	38	65	0

continue on the next two columns

Table (6.2) Data strings and their converted values to three dimensions.

As it is shown in the calibration process (Chapter three, sections 3.9) the number of the scan also represents the X coordinate. The Z and the Y coordinates are worked out as specified in sections 3.10 and 3.11. An example is given in Table (6.2) where points from Table (6.1) are converted into their respective three dimensional values. Here, data strings (B) (C), or (H) consisted of points which are on the surface of the table because the Z coordinate does not exceed 2 mm. In contrast, the Z coordinates for (F) and (G), start from 5mm and 12mm respectively, and stabilise around 28mm and 29mm. These are data strings that contain information about more than one edge. An example is shown in Fig(6.2) where the edge points obtained from AB, BC and CD can occupy an entire data string.

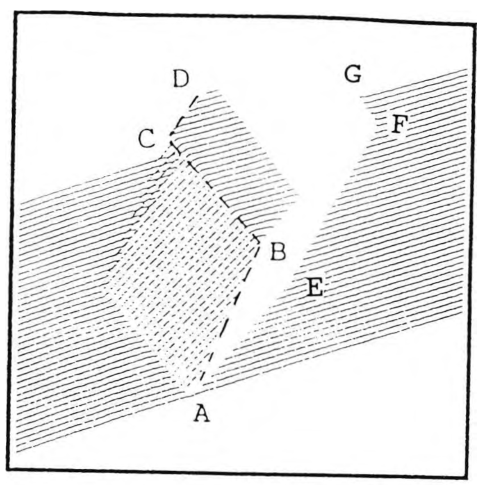


Fig.(6.2) A data string can contain information about several lines.

6.3 Extraction of Edges

The next step is to test and separate connected lines from data strings. This is performed in a similar fashion to separating lines in preprocessing (i.e. lines obtained from the middle part of the stripe in Chapter five, Section 4.5). This involves:

1) Transferring one end of the data string to the origin of an arbitrary X and Y coordinate and rotating the other end to one of the axis(i.e. such as positive side of X axis).
2) Setting appropriate thresholding lines parallel to the X axis, above and below it.

3) Dividing the graph from its highest Y point until all points are confined within the thresholding lines.

However, points that are extracted from the middle part of the stripe are much more uniformly distributed than points in data strings. So, identifying peak points in them does not pose any problem. In contrast points linked in the data strings, are gathered from different images and suffer from local fluctuation. Because of this neighbouring points are susceptible to uneven distribution. Due to this, there can be cases where more than one point can be found in the peak position. In these cases the decision making on the exact position to divide the graph (i.e. graphs made of points of data strings), is not straightforward. Furthermore since each point is obtained from a separate image, it can contain substantial information about the scene. For this, dividing a graph from an incorrect location can lead to misrepresentation and displacement of the edges. To counter these problems, initially, special cases were introduced into the programme. For example, depending on the number of points in the peak position and how they are distributed, a place is selected to divide the graph. But, in practice it is found that the number of special cases can increase considerably, without achieving the best results.

Because of that, it was decided to proceed with a controlled way of assessing and examining the content of each data string by taking the following steps:

1) To transfer one end of the data string to the origin and to rotate the other end to the positive side of the X axis.

2) To examine the graph for the points that are positioned outside the thresholding limits.

If it is found that there are some points outside the thresholding lines then,

3) To remove the point rotated on the X axis and to repeat the rotation once again, this time with the end point adjacent to the previous one.

4) Keep removing end points until all the remaining points are positioned inside the thresholding limits.

These are illustrated consecutively from Fig(6.3) to

Fig(6.8). In Fig(6.3) a set of points from a data string are taken to their respective positions (as described in the above steps). In the first picture many points are placed well above and outside the dashed thresholding lines. Next, the last point at the right hand side of the graph (the dash above the X axis) is removed and the operation repeated. This is repeated for the following points one at a time. After four steps in Fig(6.4) most of the points are positioned above the X axis. By this stage those four points were responsible for driving part of the graph below the X axis. In Fig(6.5) the number of points are much reduced but many of them are still outside the limits. This process is continued until eventually, in Fig(6.6), 10 points out of the original 34 are found to be confined within the limits. At this stage these points are separated from the rest of the graph and stored appropriately. Then in Fig.(6.7) the same operation is repeated all over again with the rest of the remaining points. In Fig(6.8) once more a group of points are found to be confined between the thresholding lines and the graph is split.

Separating points one at a time does not necessarily insure that the graph would be divided at the proper point. Nevertheless the process is controlled and the curves are divided when a set of points are actually found below the thresholding line. This helps to avoid some of the errors associated with splitting curves from the peak position which because of fluctuation may be not the best place to divide at.

Before it is concluded that a set of points represents an edge, the result is checked in each coordinate plane.

1) With the X and the Y coordinates .
2) With the X and the Z coordinates .
3) With the Z and the Y coordinates .

This is necessary because in some cases points in two coordinates alone (say X and Y) are positioned such that no indication of an edge can be found. On these occasions testing X and Y coordinates on their own is not enough and some of the edges would not be extracted. By repeating operations for the other coordinates (such as X and Z), those hidden edges become detectable. Once a set of points are found to be an edge, then the coordinates of the two ends are taken and stored.

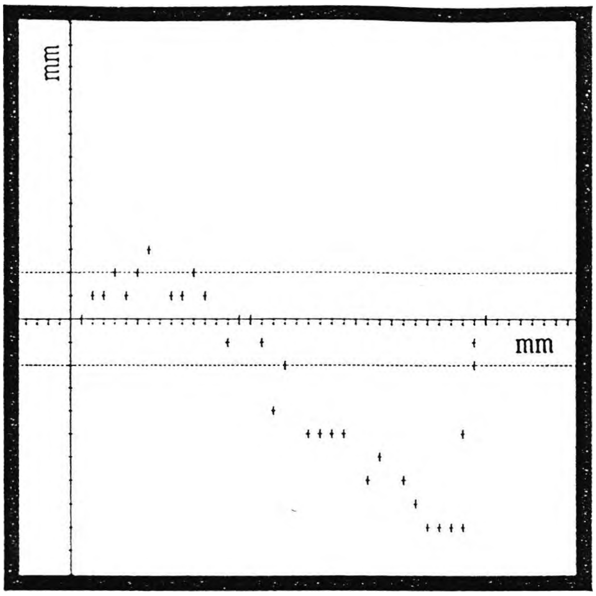


Figure 6.3. Content of a data string in it's original form

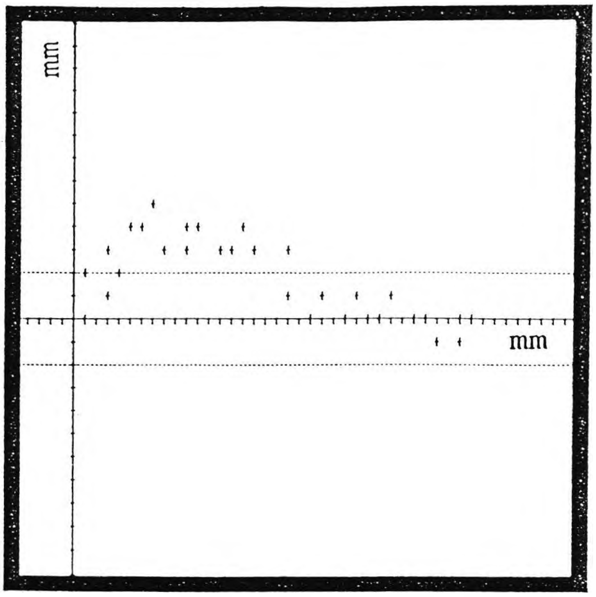


Figure 6.4. Data string after four points are removed

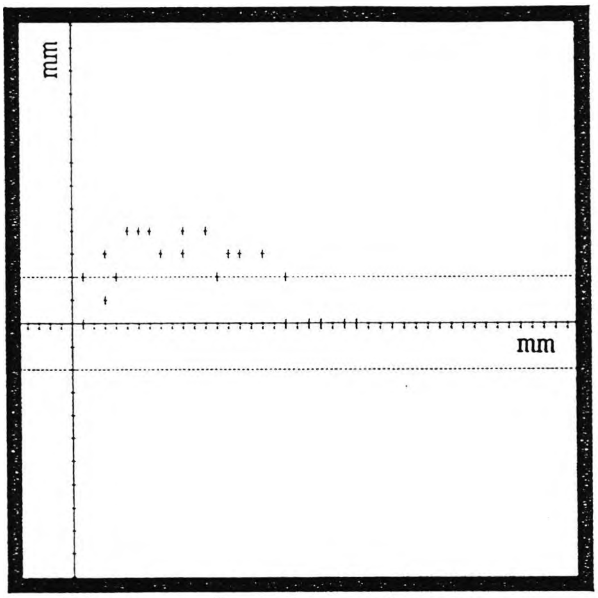


Figure 6.5. After thirteen points are removed

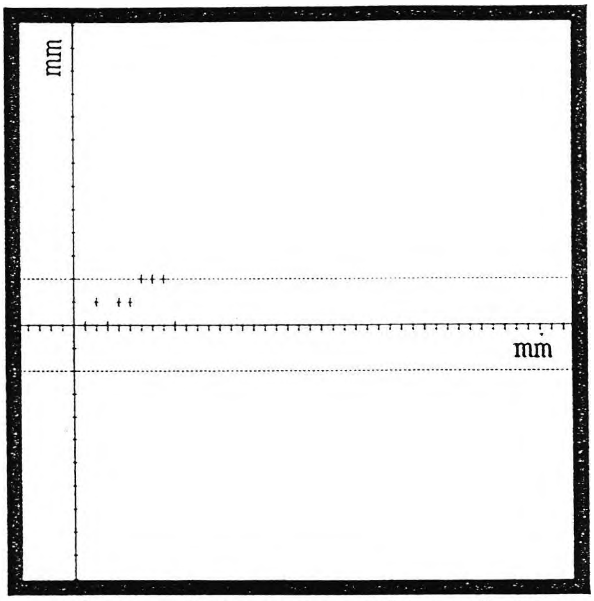


Figure 6.6. Extraction of an edge after twenty five points

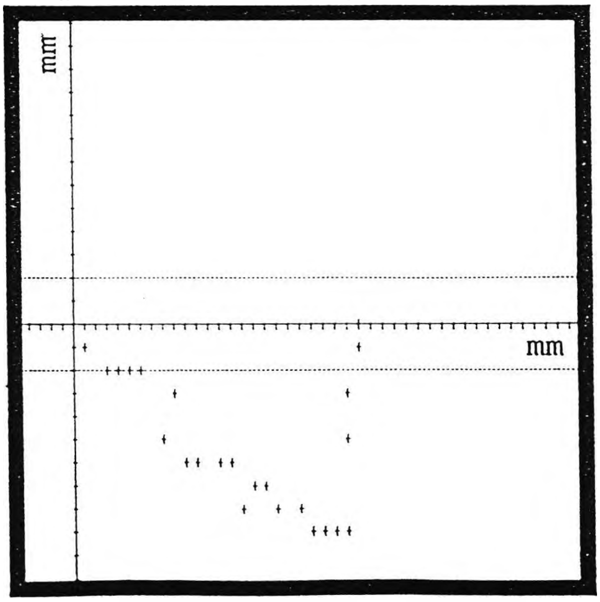


Figure 6.7. Rest of the remaining points

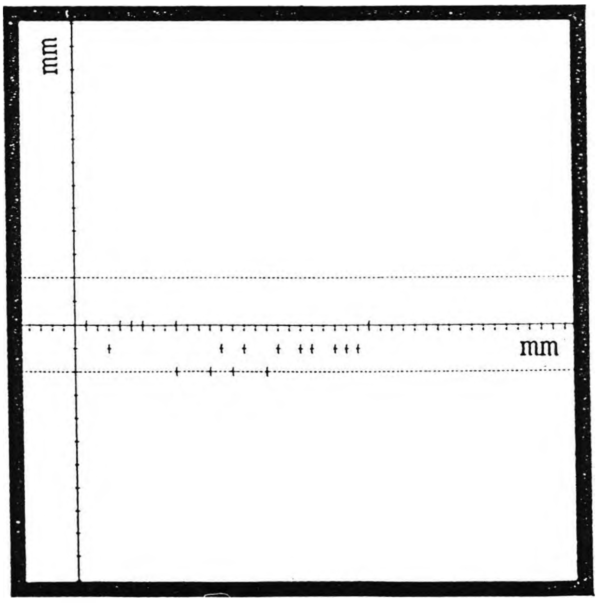


Figure 6.8. A new edge after four points are removed

An example is given in table (6.3) where edges extracted from the data strings of table (6.2) are shown. Here B,C and H consisted of lines on the surface of the table because their Z coordinate does not exceed 2.0 mm. Also data strings A,C,F and G contained several lines. For instance F consists of three lines with the first line beginning at Z = 10.0 mm, and ending at Z = 27.0 mm . The Z coordinates of the other two lines are situated at around 28.0 mm which indicates they belong to the surface on the top of the object (similar to BC and CD in Fig.(6.2)).

In these operations the thresholding line parallel to the X axis is set to be at two millimetres. This is decided after testing many different situations for various objects. But because no single thresholding line can cover every possible position, some of the edges are bound to be extracted in two fragments. In the following section an operation is added to deal with these particular cases.

The X & Y & Z co-ordinates of the two ends of the detected edges.							
Edges and data strings.		First end in mm			Second end in mm		
		X	Y	Z	X	Y	Z
1	A	1.00	32.00	3.00	21.00	17.00	2.00
2	A	22.00	17.00	2.00	30.00	25.00	23.00
3	B	1.00	33.00	2.00	36.00	65.00	2.00
4	C	2.00	32.00	1.00	23.00	16.00	1.00
5	C	24.00	16.00	1.00	40.00	28.00	0.00
6	F	3.00	36.00	10.00	24.00	32.00	27.00
7	F	25.00	32.00	28.00	30.00	28.00	29.00
8	F	31.00	26.00	28.00	37.00	34.00	29.00
9	G	5.00	39.00	12.00	15.00	51.00	21.00
10	G	16.00	53.00	22.00	24.00	46.00	29.00
11	G	25.00	46.00	29.00	37.00	36.00	29.00
12	H	34.00	65.00	2.00	38.00	64.00	2.00

Table(6.3) Extracted edges and their related data strings.

6.4 Classifying Edges

Prior to the matching process it is necessary to identify and mark those edges which are parallel (see Chapter four section 4.7). For this, the angles between the lines are calculated. Those lines which are found to be unconnected and make angles less than a certain limit are classified as parallel and marked appropriately.

To achieve that, the routine suggested by Ian O. Angell (Angell 1981) is used. It is modified to include lines in three dimensions and it can also be employed for unconnected lines in 3-D space.

In the programme that is used lines with an angle up to 22.5 degrees are classified as a parallel. This permits a considerable amount of freedom between the position of two lines in order to cover cases, encountered during the practice.

Edges are represented by X,Y and Z coordinates of their two ends and also marked so that parallel and nonparallel lines can be distinguished (i.e. the same numbers) as follows:

	X1	Y1	Z1	X2	Y2	Z2	Parallel
Line 1	..	..	..	..	..	..	1
Line 2	..	..	..	..	..	..	2
Line 3	..	..	..	..	..	..	1
Line 4	..	..	..	..	..	..	3
Line 5	..	..	..	..	..	..	2

Some of the lines extracted from the surface are in the background or created while the body of the object obstructed the light beam (i.e. EFG in Fig.(6.2)). These do not represent edges and distinguishing them from the real edges is not straightforward. Because of this all the lines extracted from the surface of the table are eliminated.

At this stage also lines broken earlier in the process are dealt with. This deals, with the cases where two lines make an angle less than 22.5 degrees and are also connected. Here the two lines are interpreted as one line. Then the two ends that are unconnected are used as two ends of a new line. In practice this adjustment is found to be quite useful but it is also open to misinterpretation. That is when the two lines make a wide obtuse angle and are also connected but they are actually two different edges.

As can be noted, here, two end points are taken and the rest of the points in between are not used. Normally in this situation a line can be fitted onto the points. But this immediately raises the question of where exactly lines come to an end. This in turn can bring in the whole problem of line interpretation and the special cases involved with it. In addition, while for some cases the situation may be improved this would not necessarily serve the requirements of the system. This is because, the initial aim is not to extract edges up to the corners but, the intention is to extract edges adequately so that matching can be accomplished. Also, introducing measures involving special cases can reduce the generality of the system.

6.5 Selecting Edges for Matching

The matching operation involves selecting two edges from the object and two edges from the model each time round. For this the following considerations must be taken into account:

1) How to select edges from the object?
2) How to select edges from the model and also how to move from one model to another?
3) How to exclude edges that cannot be matched (i.e. from object and model before operation)?

For the first and second cases the main requirement is to bring together every two edges once. For this, consider the selection of two edges from the object where its flowchart is given in Fig.(6.9). Here for the sake of explanation it is assumed that lines are selected from two identical sets, each containing all edges of the object. These are called A and B and the process of selection starts by taking the first top line from A and the second top line from B. If they are found to be nonparallel then they are sent to operations. If they are found to be parallel then the next line from B is selected. This is repeated until the bottom of the lines for B is reached. Then the next line from A is taken and the same is repeated with the rest of the lines below it (i.e. from B). By continuing this, all nonparallel lines are met once by the end of the process.

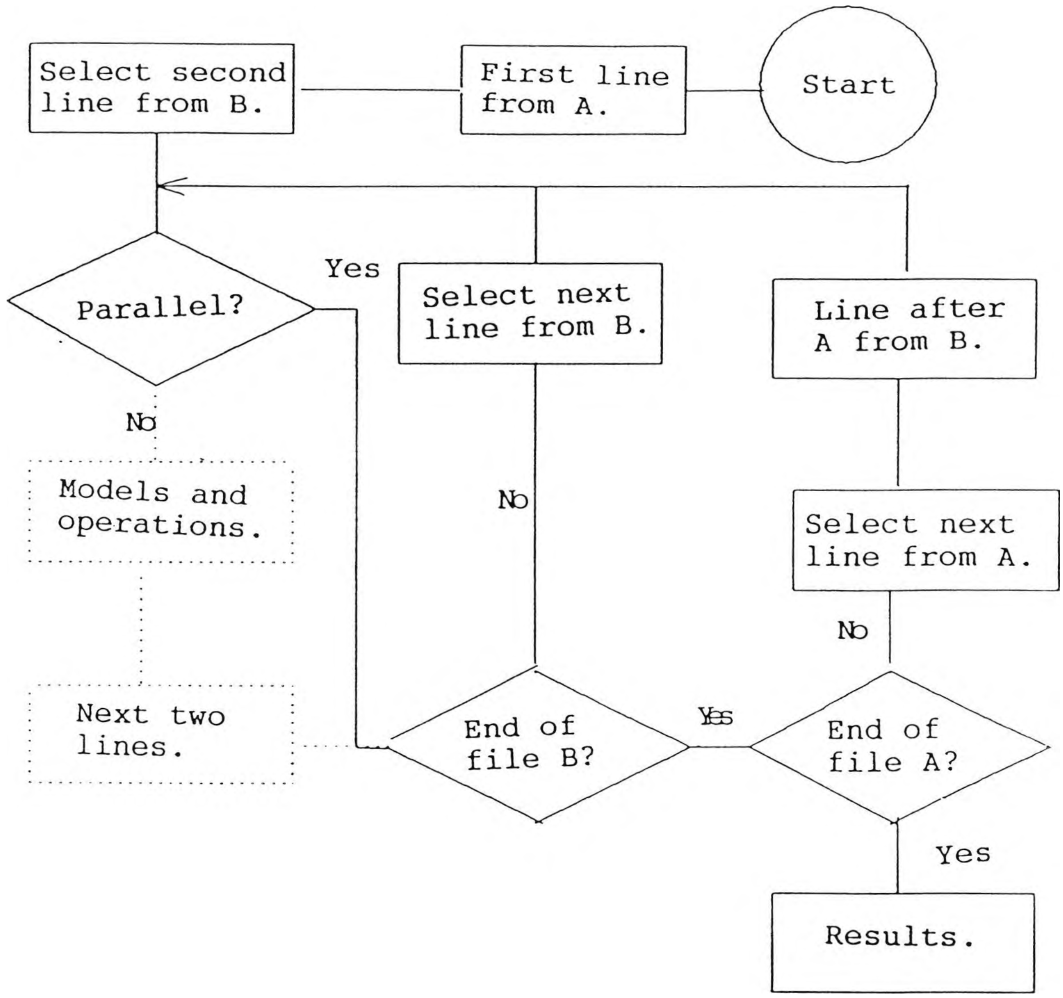


Fig.(6.9) Selection of two edges from the object. It is assumed that A and B contain all edges extracted from the scene.

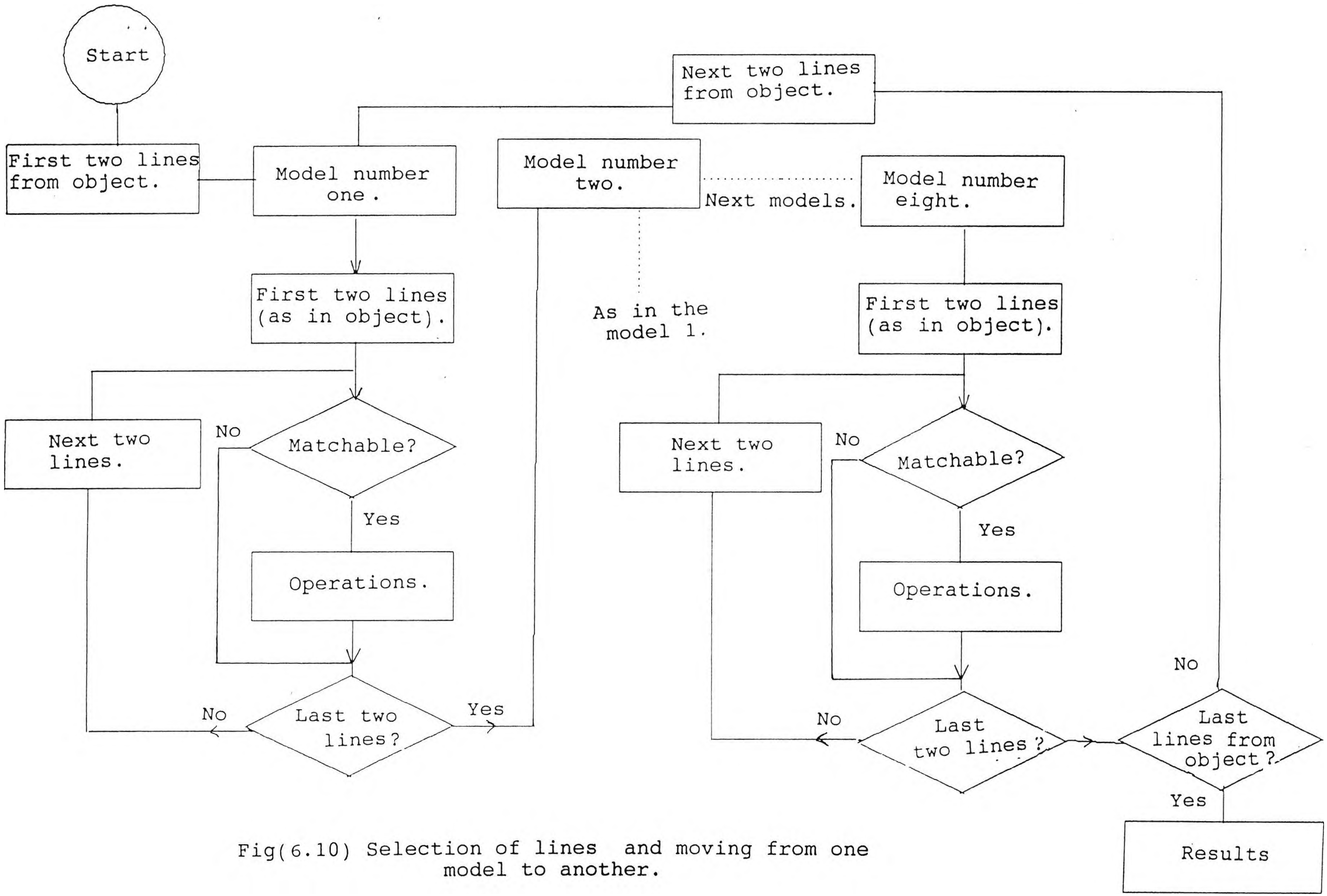
The same procedure is applied to the edges of the models as illustrated in Fig.(6.10). Here, at the start of the process the first model is examined and its lines are compared with the edges of the object. If they are found to be matchable, they are operated on, otherwise another pair of lines are selected. This is repeated until all the edges of the first model are tested and the same procedure is applied to the rest of the models one at a time. Then by going back to the edges of the object a new pair is selected and the process is repeated all over again. This is continued until all the edges of the object have been examined.

The process of selecting three connected lines is illustrated in the flowchart of Fig.(6.11). Here again for the purpose of explanation three identical sets of lines for the object and model are assumed. These are denoted by A,B,C for the object and A',B',C' for the model. First two connected lines from the object and two connected lines from the model are selected. The selection is exactly the same as described above for two lines. Then the third connected line is selected from C and C' as shown in the Fig.(6.11). The process of moving from one model to another is conducted in the same manner as in Fig(6.10).

Before the edges of the object and the model are operated on they are tested for the likelihood that they can be matched. This is necessary in order to avoid inefficient and wasteful processing since each matching operation can take a considerable amount of computing time. By eliminating those that do not have any chance of matching the whole recognition process can be speeded up. The three areas for comparison are as follows:

1) The angle between two lines.
2) Connected and unconnected lines.
3) The length of the lines .

As an example if it is found that lines in the object make a right angle and lines in the model make 135 degree then they obviously cannot be matched. Also if it is found that lines from the object are connected (i.e. the second case) and at the same time lines in the model are not connected then they also cannot be matched. Patrick J. Flynn and Anil K. Jain (Flynn 1991) described a recognition system via constrained search of the interpretation tree using pruning and heuristic termination.



Fig(6.10) Selection of lines and moving from one model to another.

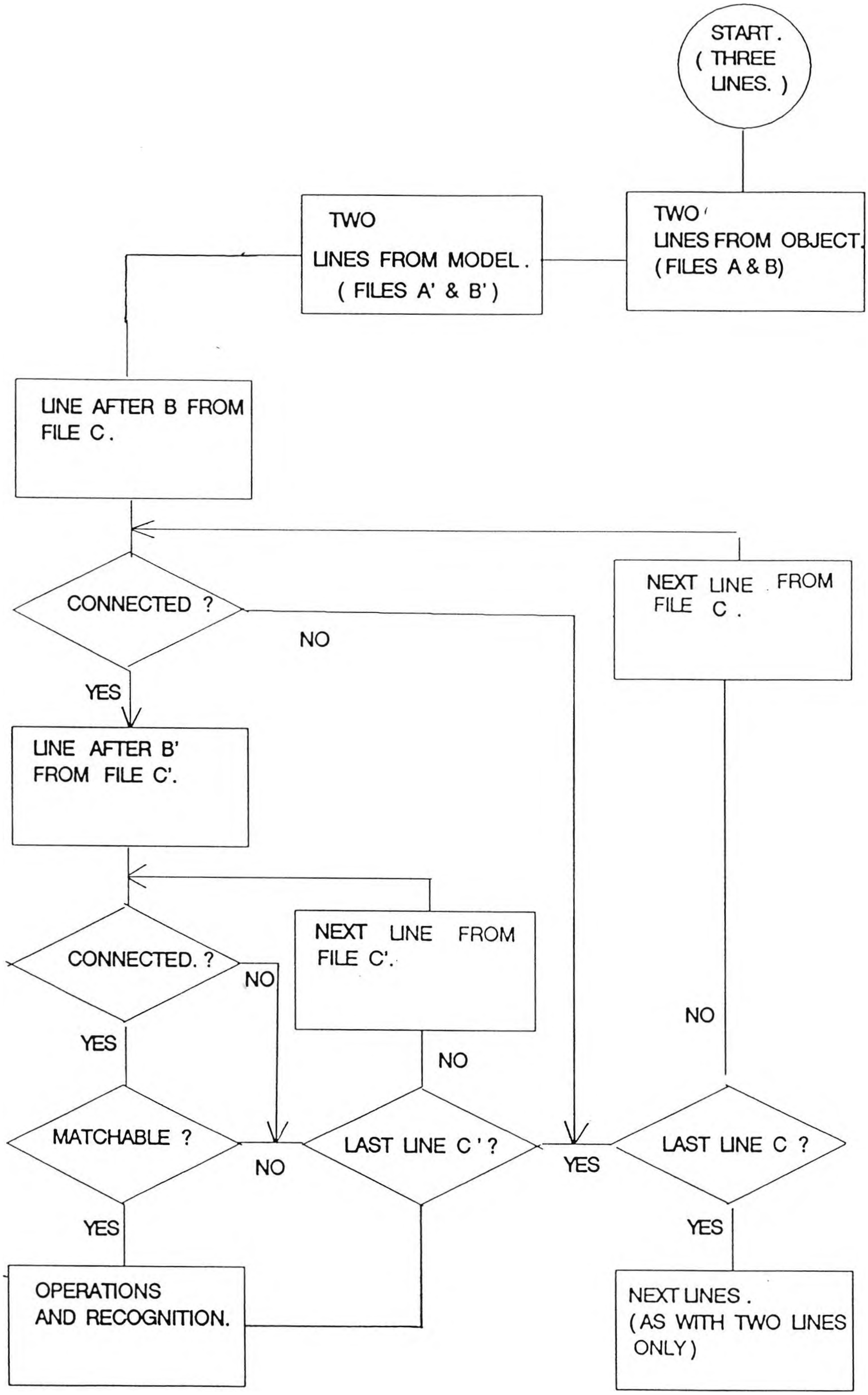


Fig.(6.11) Selection of three connected lines.
It is assumed that A,B,C together with A',B',C'
contain all edges from object and model.

6.6 Setting Parameters for the System

Throughout the building up of the system it is frequently required to set various parameters. The value of each parameter is decided individually and according to the circumstance that surrounded it. In the preprocessing section decisions on most parameters are reached merely by practical means and by trial and error. In this way not all possible circumstances can be predicted and some of the initial values are found to be inadequate in practice. As a result once the system is completed many readjustments are needed.

Once the final results are readily available the situation is much the same. Trial and error remains the only reliable method of setting various parameters. This is because with the limited number of objects it is not possible to evaluate the value of parameters by plotting the results. Since, in order to deduce any meaningful results from a plotted graph, a large number of cases need to be examined and error bars applied.

Nevertheless, using one or two sample objects can give useful indicators in showing how parameters affect the outcome of results. For that samples of objects number one (or A) and three (or C) are employed (see Fig(7.1 and 7.3)). Samples are taken while the objects are positioned on their base (see figures 7.1 and 7.3 for definition of the base). For the sample of object three (or C), a maximum twelve possible votes and for the sample of object number one a maximum fourteen votes can be obtained (An example for maximum votes is given in section 6.16). Then by gradually changing the value of each parameter, the number of votes are obtained and plotted. Next depending on whether the number of votes are higher or lower than expected the general trend of the results can be evaluated.

To run the programme there are three essential parameters which must be set. The first one is to establish when the lines from the object and model are matched. The other two are concerned with establishing when two matrices are the same. These three parameters are interconnected, and no results can be obtained without giving some preliminary values to all three of them. Because of this some parameters have arbitrary values while the others are decided. In the setting of the initial values it is possible that not the best values are chosen. For these, tests may need to be repeated before suitable initial values are selected. One practical and effective way in assessing the initial value is to take an object and its model and to work out which edges actually correspond to one another. Next to take two corresponding lines at a time and to examine their behaviour, while all the others are eliminated.

Other parameters are concerned with comparing angles between the lines, lengths of the lines, and the connectivity of the edges. These are added to prune the system and improve its performance. Their values can be easily established after the first three parameters are set.

6.7 Angle Between the Lines

This part decides on the maximum value that angles of the object and model can differ, and still remain matchable. This includes lines that are not connected in three dimensions. To establish this, various edges in different conditions are examined and 15 degrees is chosen as the worthwhile value to operate on. Here, by eliminating those lines that do not have any prospect of matching considerable processing time is saved.

The behaviour of this value is also tested on objects No. three and No. one, where Fig(6.12) and Fig.(6.13) are obtained. In these two curves the X axis represents the difference between two angles, and Y shows the number of scores. In both cases, curves are levelled off at the 12 degrees where the maximum possible votes are obtained (i.e. 12 matches for object three and 14 for object one). Here, the number of scores remain constant after the maximum value is achieved. This indicates that for these two particular samples if the parameter is set at 12, then the overall result will not be affected.

6.8 Connected Lines

Here, the objective is to establish the condition under which two lines can be considered to be connected. For this, suppose two arbitrary edges, AB and CD from the object are taken. Then the distance between point A and C and also A and D together with point B and C and also B and D are calculated.

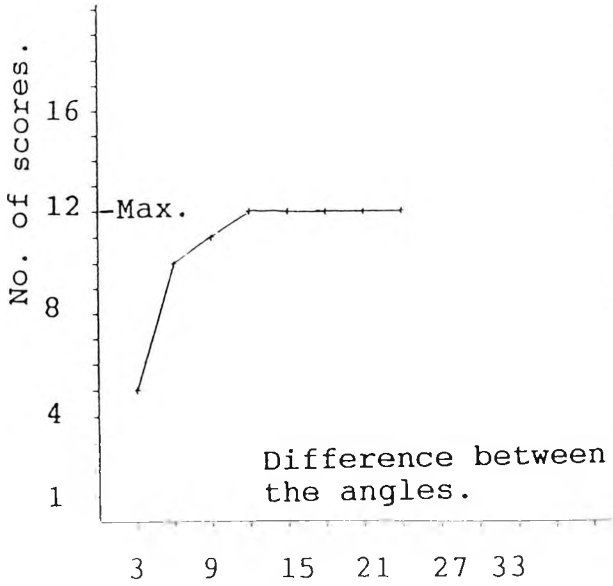


Fig.(6.12) Matches obtained for object number three. Max. is achieved at 12 degree.

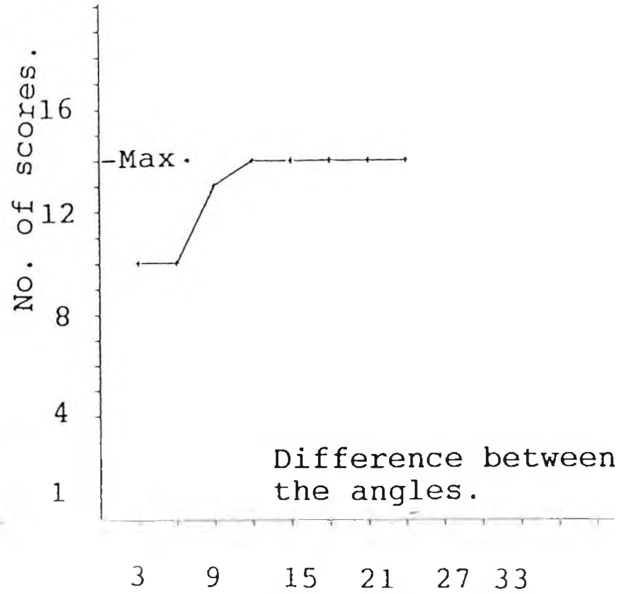


Fig.(6.13) Matches obtained for object number one. Max. is achieved at 12 degree.

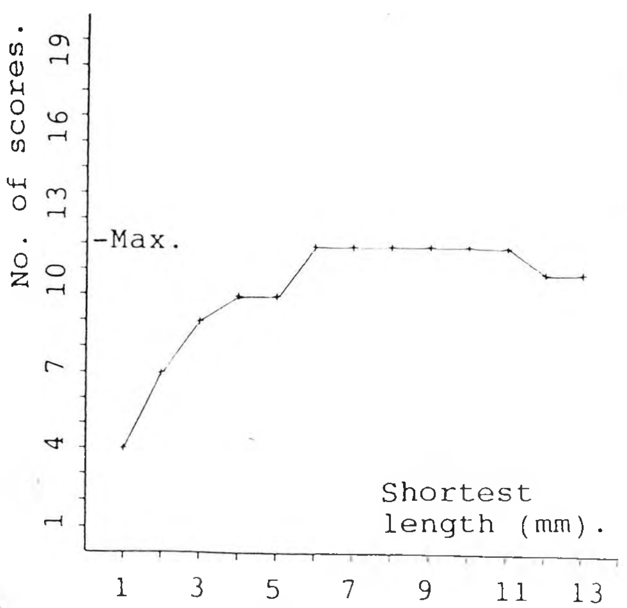


Fig.(6.14) Comparing connected and unconnected lines between object number three and its model.

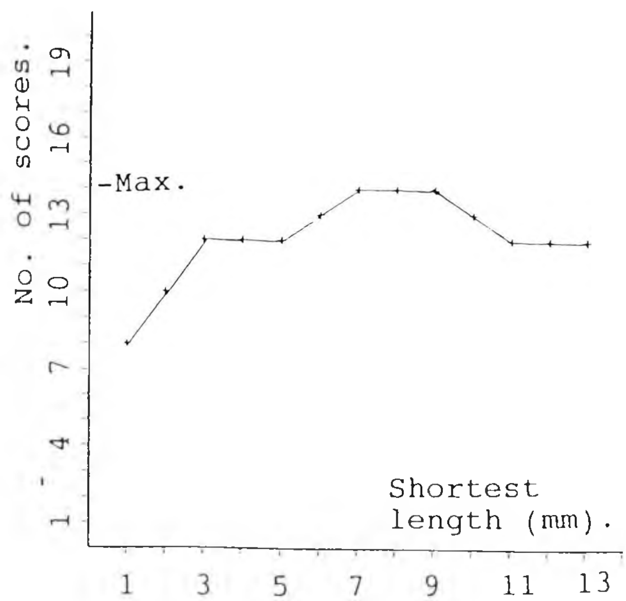


Fig.(6.15) Comparing connected and unconnected lines between object number one and its model.

Then, the shortest length is selected and if it is found to be less than a certain limit it is classified as connected. Otherwise it is considered as unconnected. These are compared with the lines from the model. Since in the models, perfect lines are defined, there is no difficulty in distinguishing connected or unconnected lines. If selected lines from the object and model are both found to be connected or both unconnected then they are operated on. In the programme that is used 7.0 mm is chosen as the limit at which two end points can reach and still be considered adjacent.

The behaviour of this parameter is plotted in Fig.(6.14) and Fig.(6.15). In these curves the X axis represents the shortest acceptable length as connected lines. For object No. three, the number of scores reached its maximum permitted level (i.e. 12 scores) between 6 and 11 mm. For values less than 6mm some of the connecting lines are classified as unconnected while their correspondent lines from the model are classified as connected. As a result they are excluded from operations and fewer matches are registered. Thus, it can be concluded that, choosing parameters in this region would have too small a value. For the values greater than 11mm. the reverse takes place. Here, some of the unconnected lines are classified as connected (because the margin is too large). Then in meeting their corresponding lines from the model (which are unconnected) they are not operated on. For object No. one in Fig.(6.15) the same trend occurs. But for this object the maximum is achieved between 6mm. and 8mm.

6.9 Length of the Lines

Suppose AB and CD are two arbitrary edges from the object and A'B' and C'D' are from the model. Here the length of AB is compared with A'B' and the length of CD with C'D'. If within a certain range they are found to be the same, then they are operated on. Otherwise AB and C'D' together with CD and A'B' are compared in reverse order. Here also, by trial and error various edges in different conditions are tested. Then 9.0 mm is selected as the margin at which the length of the two lines can differ and still be matchable.

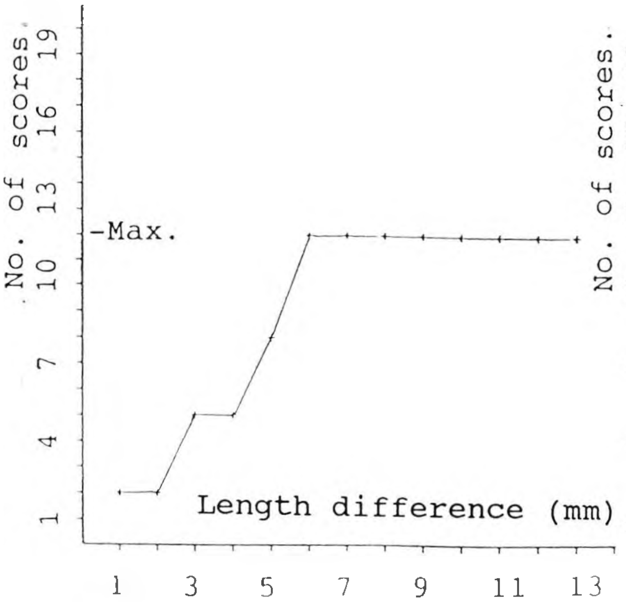


Fig.(6.16) Comparing length of the lines between object number three and its model.

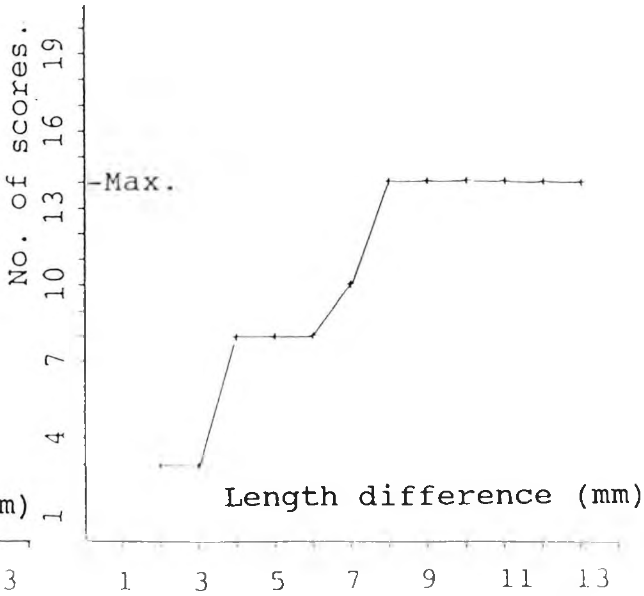


Fig.(6.17) Comparing length of the lines between object number one and its model.

The behaviour of this parameter is studied in Fig(6.16) and Fig.(6.17). For the sample of object No. three the maximum number of scores is reached at 6.0 mm. For object No. one this is obtained at 8.0 mm. For both cases the curves levelled off after the maximum value is achieved. This implies that lines with a difference greater than this can be excluded from the operations.

In these examples relatively large parameters for the connectivity of the lines and length are chosen. This can permit considerable variation between the position and length of the lines. The reason for this selection is, imperfect extraction of edges at two ends of the edges. In the section 7.16 of Chapter seven a number of reasons are given for it and combination of several factors can contribute to it. Two specific examples could be difficulty in scanning some of the edges and dislocation of the coordinate of edge points. These effects, if acting together, can further downgrade the lines. For that, a relatively large parameter is required to compensate for imperfect extraction of two end points and cover different situations.

6.10 Operations with Two Lines

Once potentially matchable pairs of lines from the model and object are selected they are operated on. Operations are the same as those which are described in chapter four and depicted in Fig.(4.7). These include taking one of the lines along the positive side of the X axis with one end at the origin and then positioning the other line parallel and above the XY plane. This requires one translation together with three rotations about the Y,Z and X axis. If no match is found then this is followed by a reflection against the XZ plane (or 180 degree rotation around the Z axis). This is illustrated in Fig.(6.18) and Figs.(6.20a & 6.21b). Here, AB and CD from the model and A'B' and C'D' from the object are chosen to be matched. First the two lines from the object are taken into their respective position. Then they are kept intact and the other two lines (i.e. from the model) are compared with them in different positions. Operations are conducted sequentially (as opposed to in parallel) and once a match is recorded, the TTM matrix is calculated.

Note that, if a match is found while point A or B is at the origin then the rest of the operations with point C or D at the origin are not truncated (see flowchart in Fig.(6.18)). This is because of a situation such as AB and BC in Fig(6.2). These two lines are approximately the same length and are at right angles. There are two possible ways of matching them with the lines of the same structure. First if AB is taken along the X axis and second if BC is taken along the X axis. Obviously, in each of these two cases different TTM matrices would result.

6.11 Operations with Three Lines

There are two central questions involved with three connected lines. These include how they are operated on and how they are matched. As an example consider a general case of AB, BC, and CD as shown in Fig.(6.19a). For a situation such as this, only AB and CD are operated on, and the middle line BC is ignored. With this assumption, operations and matching remained basically the same as the two lines above. This cannot be applied for all situations. As an example consider the three lines in Fig.(6.19b). In this example, AB and CD are parallel and cannot

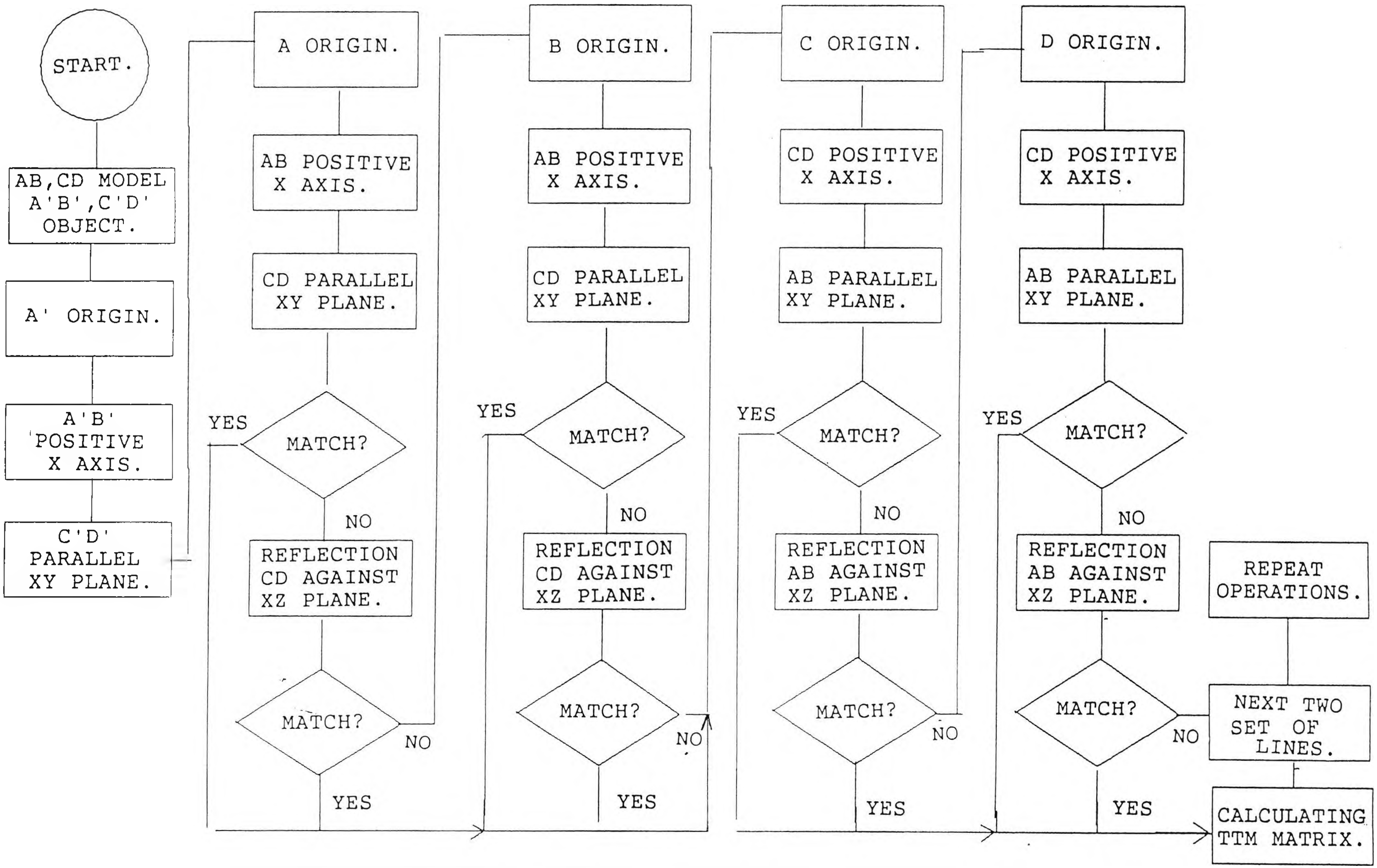


Fig.(6.18) Operations involves in matching and calculating TTM matrices. (1) operations are sequential. (2) operations are truncated once successful match is registered.

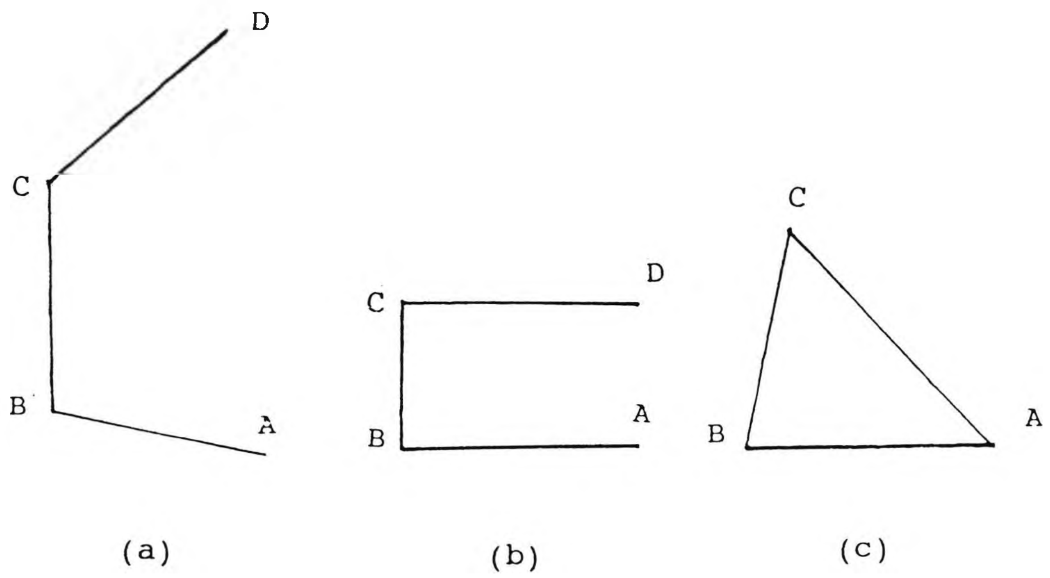


Fig.(6.19) (a) AB and CD are sent to operations and BC ignored. (b) AB and CD are parallel and cannot be used. First AB,BC then CD, BC sent to operations. (c) Shapes this kind require too many operations and excluded.

be operated on. For this, first (AB and BC) and then (CD and BC) are used and in each case the third line is excluded. Because conditions such as Fig.(6.19b) require two sets of operations it takes a longer time to conclude. Operations with three lines involve cases which need to be dealt with individually. As an example, consider the case such as Fig.(6.19c). These are excluded from operations because analysing them requires additional operations and inevitably extra time and complexity.

Pruning is conducted separately for combinations of every two lines such as (AB and BC), (AB and CD) and then (BC and CD) in Fig.(6.19a). For instance, suppose AB and BC are chosen from the object. Then these are examined against lines from the model and if the same structure is found then the rest of the lines are examined. If in all three cases the punning is successful then they are operated on, otherwise other sets of lines are tested.

6.12 Matching Procedure

The main purpose of the matching is to select only those positions from which potentially useful TTM matrices can be obtained. Otherwise, calculating and storing indiscriminately all the TTM matrices for all possible cases can add considerably to the running time. Furthermore, this can lead to a great number of matrices to be separated and stored. Because of this it is much more practical to consider those cases which are already in the position of matching.

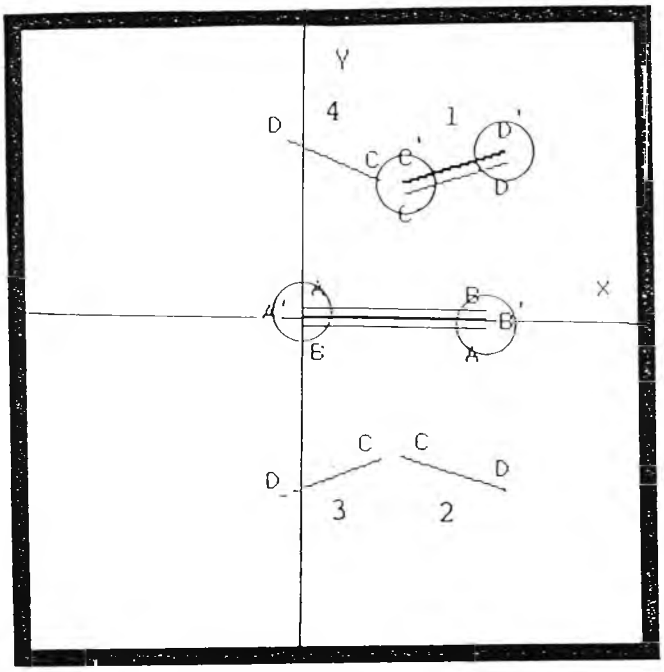


Fig.(6.20a) Matching A'B',C'D' and AB,CD from object and model. A at origin (positions 1,2).B at origin (positions 3,4).Match (position 1).

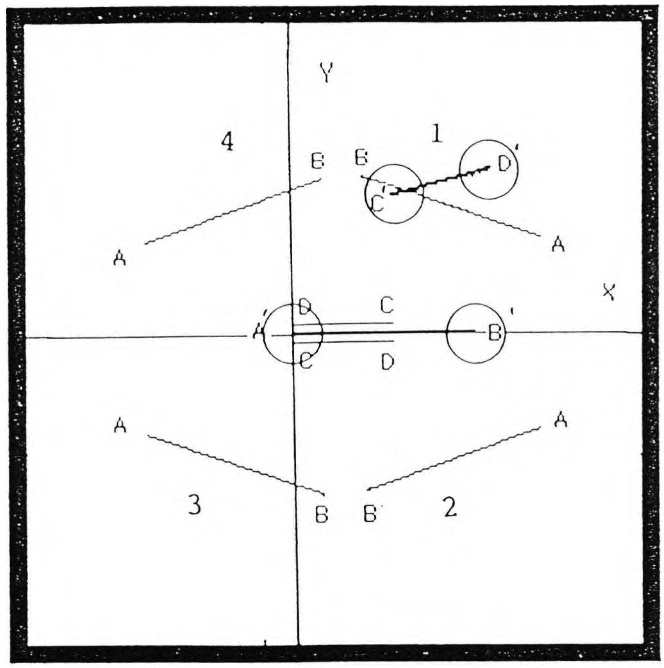


Fig.(6.20b) Matching A'B',C'D' and AB,CD from object and model. C at origin (positions 1,2).D at origin (positions 3,4).

An example of matching is illustrated in Fig.(6.20a) and Fig.(6.20b). Here the objective is to establish at which position the lines are actually matched. In Fig.(6.20a) AB is taken along the X axis with point A at the origin. Circles around the end points indicate the extent to which the end points can be displaced. In position one, end points of AB and CD from the model are coincident with the end points of A'B' and C'D' from the object. Therefore it can be concluded that in this position they are matched. In position two CD is reflected against the X axis. In position 3 together with its reflection in position 4, point B instead of A is taken to the origin. In Fig.(6.20b) the same has taken place but with CD along the X axis. Obviously none of these positions can be considered to be a match.

The two lines can be on the plane (i.e. as in Fig.(6.20a) and Fig.(6.20b) on the XY plane) or one line along the X axis and the other above the XY plane. This depends on whether two lines are coplanar or not. But the same would result if lines are positioned in three dimensions. In these cases reflections would be conducted against the XZ plane (or 180 degree rotation around the Z axis).

In the same context L. Du et al. (Du 1992) considered noise contamination had arisen from the perturbation of observed lines caused by imperfect extraction. Also, Grimson et al. (Grimson 1992) provided an analysis of affine point matching under uncertainty and the probability that false positive matching due to the sensor error can occur.

6.13 Selecting Parameters for Matching

To establish to what extent end points can be displaced, many different situations are examined. As a result the uncertainty of 9 mm is chosen from which two end points can differ and still be considered matchable. The reason for this relatively large selection is, to cover those situation where the edges are poorly extracted. As an example consider A'B' in Fig.(6.20a) and imagine the two ends of this line are poorly extracted and one end is taken into the origin. Then the other end would be left with the combined defects of both sides, when it is compared with lines from the model. It is possible to exclude all the poorly extracted edges by selecting a relatively low uncertainty. But obviously in this option some of the proper matches would not be recorded. In addition to that if false matches occur it should still be possible to distinguish them by their TTM matrices.

To assess the behaviour of the parameters for different values samples of object number three and one were examined and the curves of Fig.(6.21) and Fig.(6.22) obtained. Here, the space between the end points increased by a millimetre at a time. For object number three the number of matches increased until at 7 mm it reached its maximum possible 12 matches. Then it remained steady until at 10 mm where it exceeded its maximum value. The same trend emerged with object number one. Here the maximum value was obtained between 7mm and 11mm. Then at 12mm it also exceeded its maximum.

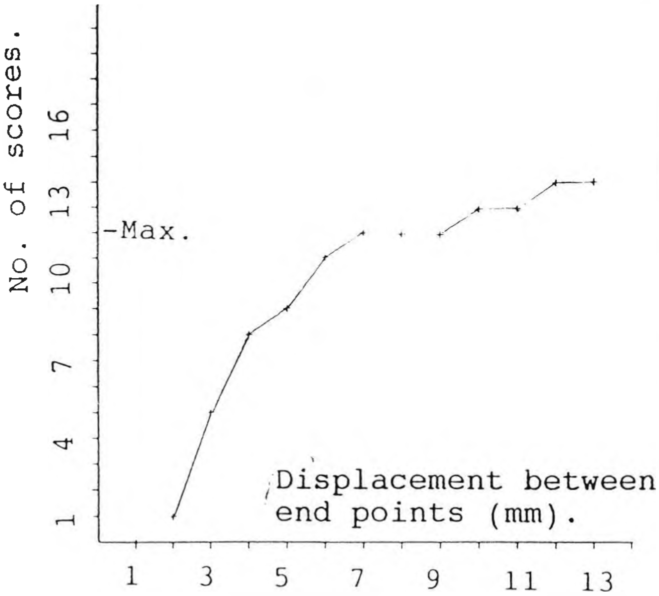


Fig. (6.21) Matches obtained for object number three. Maximum is achieved between 7mm and 9mm.

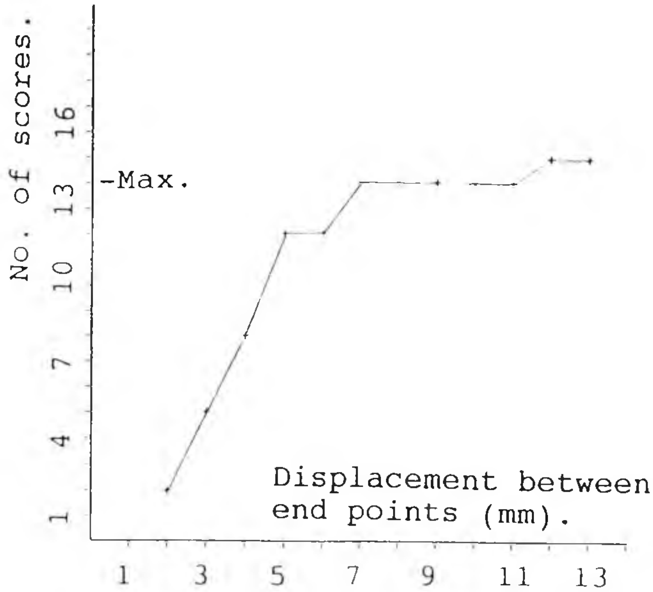


Fig. (6.22) Matches obtained for object number one. Maximum is achieved between 7mm and 11mm.

The number of votes cannot exceed their maximum possible matches. However, in these examples some positions are mistaken for a match as the uncertainty between two end points is considerably increased. This is because elements of some of the matrices, are very close to the TTM matrices of a real match (possibly from reflection matrices). Then because parameters chosen for separating matrices (next section) did not distinguish them, they were added to the matrices of the real match. This example shows how the parameter chosen for the matching is interconnected with those of the separating matrices.

6.14 Separating and Sorting TTM Matrices

Once a successful match is recorded the TTM matrix associated with it is calculated. As discussed in section 4.8 of Chapter four, calculating the TTM requires calculating the inverse of the matrix. For this the Gauss Jordan elimination method (Fox 1964) is employed. Next, its elements are compared with the elements of previous matrices obtained from the same model. If the same matrix is found then to that matrix a unit score is added. If the matrix is different to all others then it is stored separately. In this process there are two problems that need to be considered:

1) How to compare two matrices and determine whether they are the same or not.
2) How to increment similar matrices.

For the first case, the corresponding elements of the two matrices are subtracted. If it is found that all their elements are within a certain range then they are considered to be the same. In the following a typical TTM matrix is given,

0.35	0.89	-0.39	-24.58
-0.61	0.49	0.61	17.56
0.70	0.00	0.79	29.62
0.00	0.00	0.00	1.00

In this matrix the first three columns are associated with rotational elements and the fourth column with the translational part. During the operation it is frequently found that elements in the last column are much greater (or smaller) than elements in the other columns. This is because the first three columns are made of the sines and cosines which are limited to $-1 =< \sin(\theta)$ or $\cos(\theta) <= 1$. As a result the elements in the fourth column can have much greater values. Because of this two separate parameters are introduced. One to compare elements in the fourth column and the other, for the rest of the elements.

Each matrix is stored along a counter so that the number of its occurrence can be recorded. If a matrix is found for the first time the counter related to it is set to one. Then each time the same matrix is found a unit score is added to it. For each model sufficient memory is made available, so that a great number of matrices can be accommodated.

It is important to compare new matrices with every one of the matrices already stored since there could be some cases where roughly the same matrices can be produced. If the new matrix is only clustered with the first matrix in the memory, then the next similar matrix would not be counted.

At the end of the recognition process counters associated with each model are examined and the highest score is collected. Then it is compared with the largest votes from the other models and the model with the top score is selected to be the object in the scene.

6.15 Selecting Parameters for Matrices

To establish parameters for the matrices empirical data obtained from different situations is examined. From that, margins of 10 for translational and 0.30 for rotational elements are chosen so that two correspondent elements can differ and still be considered the same.

The margin of 10mm is chosen for translational elements and 0.30 radian for rotational elements of the matrix. These values indicate that if two lines from the object are transferred by the TTM matrix to their correspondent lines in the model to what extent they can be displaced and still considered matchable.

In extracting edges, various factors can contribute to error. Inadequacy in the edge extraction can be one of the main contributors. Here, to represent the line, two end points are taken and the rest are not considered. These can be significantly fluctuated and different from where the edges actually start (i.e. end points). In addition, in dividing the curve (i.e. in section 6.3) it is possible that the best place is not selected, which in turn can further displace the position of the edge. Also in taking one end of the line to the origin (i.e. as in section 6.12) the combined errors of both ends can transfer to the other side.

These examples demonstrate just some of the problems which can occur in practice. The need to cope with situations where an accumulation of various defects can occur simultaneously necessitates relatively large margins. The values selected above in effect exemplify such a situation and represent the worst possible case that can occur.

It is important to note that the main objective is to distinguish TTM matrices from one another and group them accordingly. For that, as long as these large parameter values (i.e. compared with overall size of the objects) do not interfere with the process they are justified.

To examine the behaviour of the parameters, samples of object number three and one are studied. In Fig.(6.23) and Fig.(6.24) results obtained for the elements of the fourth column are illustrated. Here a maximum uncertainty of 6 for object number three and 8 for object number one is recorded. For rotational elements Fig.(6.25) and Fig.(6.26) are obtained. For these curves a maximum uncertainty of 0.25 for object number three and 0.20 for object number one is resulted.

6.16 Example with Object No. Three or C

An example is given in Table (6.4) where matrices obtained from object number three are recorded. The object is shown in Fig(6.27) with its edges AB,BC,CD,DE,FC. As indicated earlier this sample can produce a maximum of 12 possible matches. The exact combinations of lines are as follows:

AB and BE	FC and CD	BE and DE
AB and BC	FC and CB	BE and BC
AB and CD	FC and BE	ED and CD
AB and DE	FC and DE	BC and CD

In this table the matrix with 12 scores is obtained from matching the above lines with their corresponding lines in the model. There are also three matrices with four scores which originate from the rectangle BCDE on top of the object. Here, depending from which corner matching starts four sets of matrices can be obtained. The fourth matrix is actually incorporated in 12 scores obtained for the object itself. Some matches can occur because of similarity in the structure of the lines be-

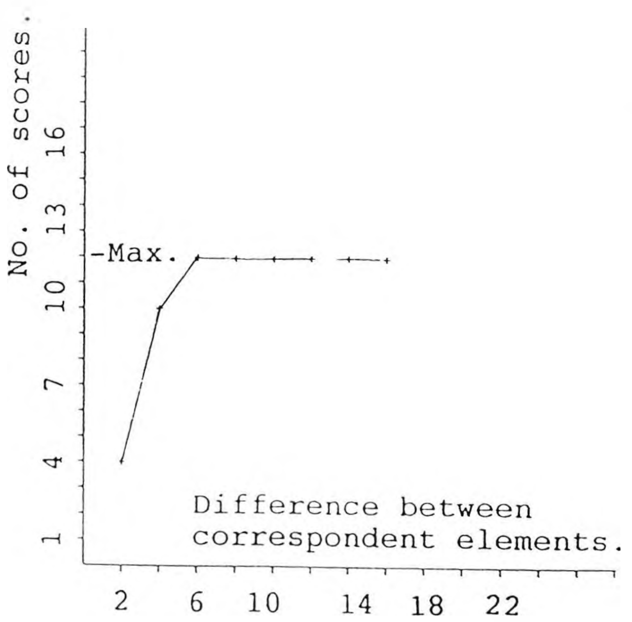


Fig. (6.23) Difference for translational elements of object number three (last column).

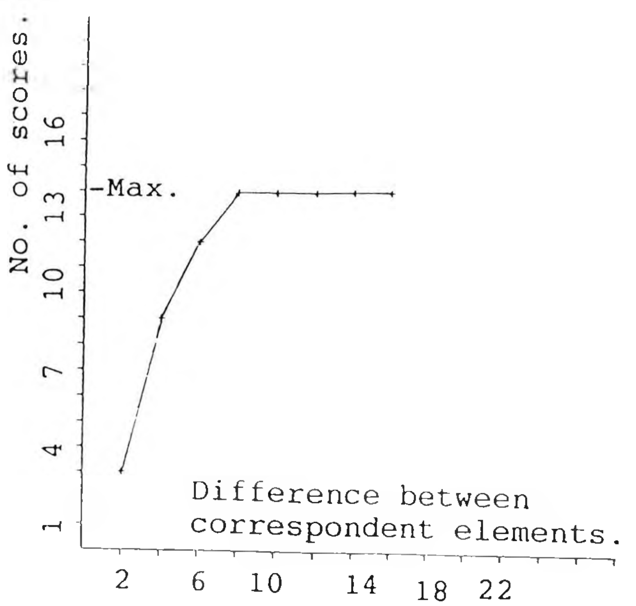


Fig. (6.24) Difference for translational elements of object number one (last column).

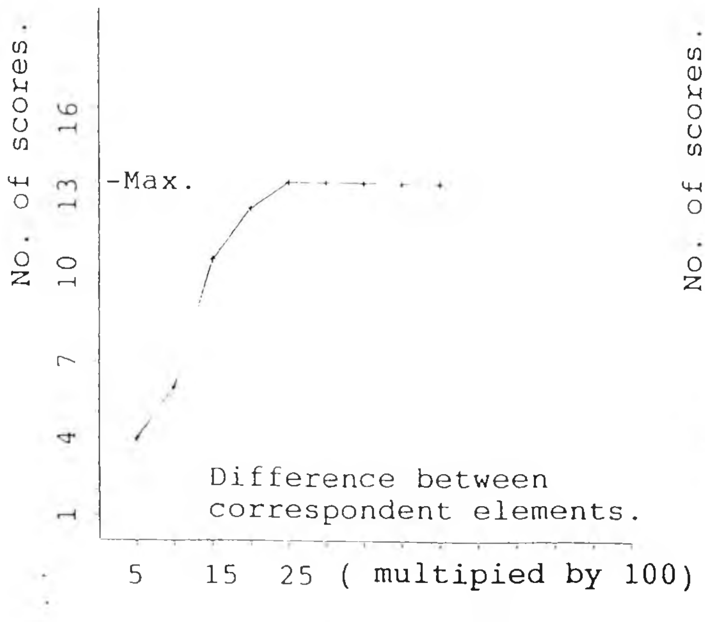


Fig. (6.25) Difference for rotational elements of object number three (first three columns).

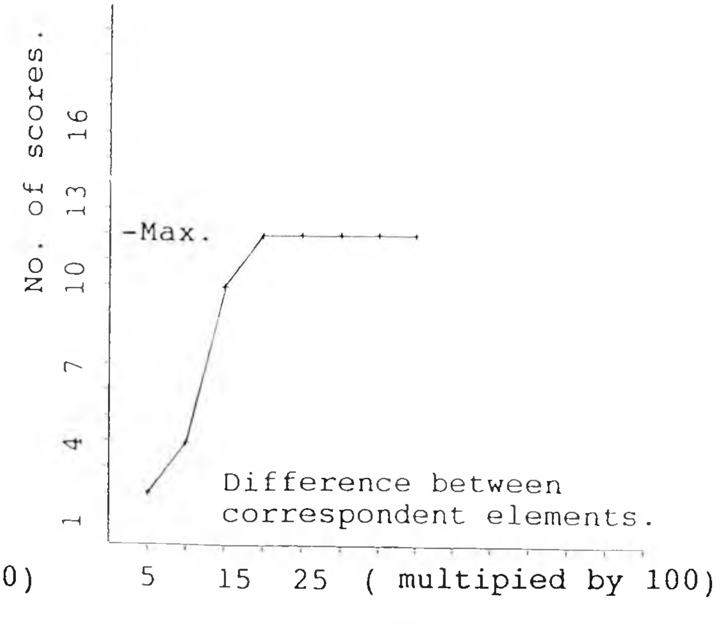


Fig. (6.26) Difference for rotational elements of object number one (first three columns).

-60	80	-1	-6	-60	80	-1	16	88	-24	-41	22
-80	-60	1	63	-80	-60	1	46	32	-34	88	15
0	1	100	2	0	1	100	2	-35	-91	-22	40
0	0	0	1	0	0	0	1	0	0	0	1
	Total	No.	1		Total	No.	12		Total	No.	2
-12	24	-96	31	43	-80	42	16	-60	-80	-3	39
-92	34	20	32	41	59	70	-2	-80	60	2	29
37	91	18	-6	-80	-13	58	23	0	4	-100	57
0	0	0	1	0	0	0	1	0	0	0	1
	Total	No.	1		Total	No.	2		Total	No.	4
60	-80	3	31	60	80	-3	11	-87	-30	39	30
80	60	-2	23	80	-60	2	38	-25	-41	-88	49
0	4	100	1	0	-4	-100	58	42	-86	28	19
0	0	0	1	0	0	0	1	0	0	0	1
	Total	No.	4		Total	No.	4		Total	No.	1
10	30	95	-3	-70	71	1	39				
88	41	-23	8	-70	-70	13	26				
-46	86	-22	17	10	9	99	2				
0	0	0	1	0	0	0	1				
	Total	No.	1		Total	No.	1				

Table (6.4) Matrices and their total number of occurrences for the sample of object number three. Note, rotational elements (the first three column) are multiplied by 100.

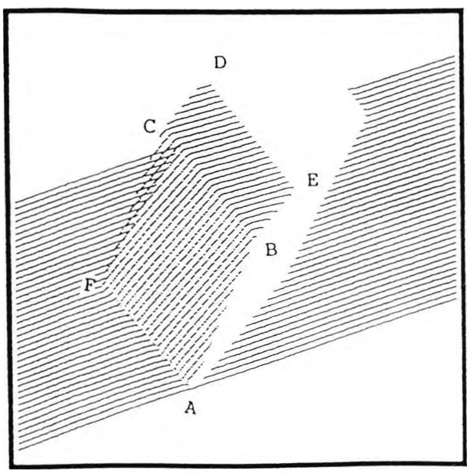


Fig.(6.27) Sample of object No. 3 with 12 possible matches.

tween the model and the object. As an example FCD from the object can match with ABE from the model. These belong to the two different sides and the matrix generated from them differs from those obtained from the same sides. Cases such as this generate single matrices of their own as shown in Table (6.4).

In this particular example the model is the same as the object and containsthe same edges. For that, for every couple of lines from the object 12 cases from the model need to be tested which makes a total of 144 cases. Each case requires a further 8 variations in the way that they are taken into the origin of the coordinates. This can make a total of 1152 in all. If these are processed indiscriminately a huge number of matrices would need to be stored and processed. But, by introducing pruning and matching, the number of matrices are reduced to those of table(6.4).

Highest scoring matrix for model one	3
Highest scoring matrix for model two	2
Highest scoring matrix for model three	**12**
Highest scoring matrix for model four	1
Highest scoring matrix for model five	8
Highest scoring matrix for model six	2
Highest scoring matrix for model seven	1
Highest scoring matrix for model eight	4

Table(6.5) Total number of scores obtained for various models.

In Table (6.5) the number of scores obtained from different models are compared. Here, model number three outnumbers other models and therefore, it is concluded that the object is most similar to it.

6.17 Summary and Comment

The result of linking edge points, is several data strings each containing information about the number of edges. Next data strings are divided and edges extracted from them are classified. Then various parameters are established and pruning and matching are introduced and relevant TTM matrices are computed and stored.

The performance of the system depends on the suitability of parameters chosen at different stages. Parameters are chosen by comparing empirical data together with two example objects. This is done because obtaining a large set of data with a limited number of objects is impractical.

In order to derive edges, nearest edge points from consecutive images are linked. This is achieved by increasing the radius of the circle around edge points. Compared with using the direction of edge points, this has the advantage of being less sensitive to local fluctuation of edge points. The performance of this method, depends on the reliability of data that is passed on from the preprocessing section and extraction of edge points in the sequence. Because of that considerable strain can be placed on preprocessing.

CHAPTER SEVEN

EXPERIMENTAL RESULTS

In this chapter the experimental results obtained with real data are presented. First the objectives of the experiments are outlined. Then nine different objects are introduced and some of their significance and characteristics are described. Subsequently, in two different categories, objects are recognised. For each object the results are examined and their implications discussed. Also recognition is performed with two objects at a time and with two or three connected lines only. Then, difficulties in extracting some of the edges are discussed and finally brief accounts of the results are given.

7.1 Objects and the Recognition Process

To examine and test the system nine different objects with different shapes and with varying degrees of complexity are used. These are depicted in Fig(7.1 to 7.9) with their sizes given by their side. They are cut from cubes approximately 28 mm in length. All objects have retained at least a few edges from the original cubes which makes them roughly the same size. Because of this it is difficult to distinguish them merely by their overall size. Also it is not easy to identify them by their overall shape. Among the objects there are fairly simple objects such as the cube in Fig.(7.8), object B in Fig(7.2) and C in Fig(7.3). There are also objects with moderately complex shapes and with edges which are short and difficult to extract. These are objects such as A in Fig(7.1) and D in Fig(7.4) and more complex objects such as F in Fig(7.6) and G in Fig(7.7).

These objects are put into experiments in order to assess how successfully the system can select the right model. Also, if the model is correctly chosen how reliable or robust is that selection? In addition to assess what conditions need to be fulfilled for the process and in which situations objects cannot be recognised. Here, the aim is to investigate characteristics or potential problems for each particular shape and object. This can range from difficulties in extracting some edges or problems with relatively complex objects. Also, this can include examining the effects of particular parameters or assessing how the number of edges can influence the processing time. In each of these cases the aim is to determine what possible measures can be taken to improve the performance and efficiency of the system.

7.2 A Brief Description of the Nine Objects

In this section objects that are used in the experiments are introduced.

Object A: In terms of complexity this object is ranked as average compared to the others. Here HG in Fig.(7.1) is comparatively short which could be left, potentially undetected. Otherwise, the size of most of the edges is long enough to allow the necessary number of scans to cross and identify them. This, of course, is very much dependent on how the object is positioned and how each edge is placed with respect to the stripe.

Object B: In this object if AFGE or AGCB or GEDC (i.e. as defined in Fig.(7.2)) is placed on the surface of the table (i.e. any face with the square) then it is not possible to distinguish its shape from the others. In terms of complexity this object is ranked as below average complexity compared to the others.

Object C: This object was originally used to test the system in its various stages of development. It is used because, it has fairly straightforward features which make it suitable for study in isolated conditions. As it is shown in Fig.(7.3) the length and the angle between the edges are such that they should be extracted properly. In terms of complexity this object is ranked as below average complexity compared to the others.

Object D: In this object point F in Fig. (7.4) is not located symmetrically at the middle. Instead it is positioned closer to AB rather than CE. For this, choosing AEDG or GDCB as a base would result in different edges. In terms of complexity this object is ranked as average. In most cases its edges should be extractable with relative ease.

Object E: In this object each rectangle has a different size to the others (see Fig.(7.5)). For this, choosing ARCB instead of ARJK as a base would result in different edges. Since the number of edges is relatively high, processing can take considerably longer than the others, particularly if models contain all the edges of the object. In terms of complexity this object is ranked as above average complexity compared to the others.

Object F: This object has a number of edges which are either short or in some positions difficult to extract (see Fig.(7.6)). Because of that, recognition may be conducted with not all edges found in the scene. This object is ranked as fairly complex compared to the others.

Object G: As it is shown in Fig.(7.7) this object can be ranked as complex compared to the others. This is because of its particular shape which can obstruct part of its body. As an example, CHGLD can obstruct HC or DG depending on how it is positioned in front of the camera.

Object H: In this object only four edges of the top square can be extracted (partly because of the position of the projector and camera relative to the surface of the table). Because of that its recognition can be conducted with one square only (see Fig.(7.8)).

Object J: This object (see Fig.(7.9)) sustains the general structure and features of object C. But in object J the positions of some of the edges are displaced and some of the edges are shorter (such as AB and GC) or longer (such as HB and CD) than their correspondent lines in object C. For that it can be used to assess to what extent two similar objects can be distinguished.

7.3 Two Categories of Experiments

For each object a base is defined and recognition is conducted in two different categories.

1) Objects are placed with their base on the floor and with their face or upper part randomly positioned in any direction. This class is defined as the base class.
2) Objects are placed randomly in any position or orientation with the condition that at least some edges cut from the original cubes are also exposed to extraction. (with the exception of the cube itself). This class is defined as the general class.

As an example AGFJ is chosen to be the base for object A in Fig.(7.1). By examining objects in two separate categories it is possible to study first, the behaviour of the recognition algorithm with a limited number of edges in the database (i.e. edges of the models). That is when only those edges are included that can be seen from the camera and can be extracted and are also not positioned on the floor. Next recognition can be applied more generally for the greater number of lines. That is when all edges without exception are included in the database. Obviously this requires a longer time to search and to process.

7.4 Objects and Their Models

The model of each object is constructed and stored separately in the database for recognition purposes. Models consist of X,Y and Z coordinates of the beginning and end points of each edge (i.e. coordinate of vertices). Edges are also indicated with a number showing which other edges they are parallel to. Proper care is taken to choose a system of coordinates compatible with the system of coordinates in the scene in terms of the right hand or left hand axis. This can be altered by a number of factors such as mounting the camera upside down, or the direction of motion of the table, or transitional matrices involved in the operations. Models are constructed in two groups and each group is used for one of the two categories of experiments. Models of the first group are only included with the edges that can be extracted from the scene and not situated on the surface of the table. These models are used whenever objects are placed on the base (as explained above). For the second category of tests, all the edges of each object are included. For the cube, however, only four edges of the top square are included. This is because with the setting of our equipment, only four edges at the top can be extracted.

7.5 <u>Experiments with Object A</u>

For object A experiments No. 1 and No. 2 are conducted (see Figs 7.10 to 7.13 and Tables 7.1 and 7.2). In experiment No. 1 the object is placed on the base and models include only edges of the first category of test (i.e. as shown in Fig.(7.1 to 7.9)). In experiment No 2 the object is positioned regardless of the base and models include all the edges of the objects. In the light-stripe images of Fig.(7.10 and 7.12) each line is taken from a scan and then consecutively placed beside previous lines so that the perspective of the object in the scene is constructed. Recognition results for these two experiments are given in tables (7. 1 and 7.2) where in both cases the object is recognised with the majority of votes (i.e. 14 and 11 votes).

In both cases the number of edges extracted from the scene is seven but the running time is substantially different. In experiment No. 1, the first model with 8 edges (in database) has taken 32 seconds to process while in experiment No.2 the same model with 14 edges (in database) has taken 1 min. 19 seconds to conclude. Here, while the number of edges are less than doubled the running time has increased considerably. In these examples edges before being operated on are heavily pruned. Because of this it is not possible to assess directly how the number of edges can affect the running time. Besides, the shape of the object and the models themselves can affect the processing time. Nevertheless, it can be concluded that an increase of only one

edge to a model with N edges, can produce N new combinations. If some of the new combinations get through the pruning stage, then this would add to the running time. This can escalate considerably, if the number of edges grows by a modest amount. This situation is best known as an exponential explosion. In addition, sorting out extra edges and additional pruning operations, can itself contribute to the running time.
In each of these experiments seven edges are extracted but, in none of the cases are the maximum number of votes achieved. This, implies that some of the possible matches have not taken place or been counted. Here imperfect extraction of the edges can be the major contributory factor to that.

6.6 Experiments with Object B

For object B, experiments No.3, No.4 and No.5 are carried out (see Figs 7.14 to 7.19 and Tables 7.3 to 7.5). In none of these cases is the object studied with FBD or ABF or DEF or BCD (as it is defined in Fig 7.2) placed on the surface of the table. This is because FB,BD,DF are the only edges which differ from the edges of the original cube. If any of those surfaces are placed on the table, then none of them would be exposed to the extraction process. This arrangement is necessary because if only edges shown to the camera are of the original cube then there would not be any clue to recognise the object. For that reason the object is studied while it is placed on one of its squares and only edges of the first category of tests are used for the models.

In each of these experiments five non parallel edges are extracted which can make a maximum ten possible votes in each case (i.e. combination of every two edges). In experiment No.3, the object is distinctly recognised with the expected ten maximum votes. However, in experiments No. 4 and No. 5 the object is distinguished with only 6 votes. In these experiments recognition is achieved while one of the edges is poorly extracted. Since object B has relatively few lines and comparatively long edges it is suitable for examining and investigating some of the circumstances in which votes are not accounted for.

In order to analyse the results each of the edges from the object is eliminated one at a time. For experiment No. 4 when AB or BD or AC or CD is removed, the number of votes are reduced further down from 6 to 3. But, when CB is taken out of the process, the results remain unaffected (i.e. the same 6 votes). This indicates that CB is singly responsible for the loss of four votes. As can be noted from Fig.(7.16) CB is situated roughly along the stripe lines. This makes it difficult to extract. Once edges are plotted on the XY plane, it becomes clear that CB is considerably displaced (i.e. as it is shown in Fig.(7.20)). This example emphasises the case, where, because of the position of an edge relative to stripe lines, it is inadequately extracted and as a result four votes are not counted .

In experiment 5, CB is found to be responsible for the loss of four votes. But, in this case a different reason can be attributed to it. As it can be noted from Fig.(7.18) stripe lines towards C become more straight lines rather than broken lines. This makes extraction of edge points from CB difficult. This is better illustrated in Fig.(7.21) where edges are plotted on the XY plane. In it, BC is to some extent displaced and is quite short towards C. In this example the difficulty in extracting edge points resulted in the loss of four votes.

Note that, other edges in Fig.(7.20) and Fig.(7.21) are not exactly terminated at the same point (i.e. such as A or D). But because, these are generally within the tolerance limits set for the system no loss of votes results from them.

Working out in detail all individual cases throughout the experiments is not practical. However, it has to be emphasised that the recognition algorithm consists of three rotations around the coordinate axis. The angle of these rotations is dependent on the position or length of the extracted edges. These are susceptible to considerable variation if, for any reason, some of the edges are extracted inadequately.

In all these three experiments, model D is positioned in the second place after B. This can be attributed to the comparative similarity of the two objects. For example the relative position of BDF and BFD in Fig.(7.2) is roughly the same as BFC and FCE in Fig.(7.4). Also the 52 seconds running time for object D is considerably higher than the other models. This is because pruning models which are similar to the object is more difficult than those with little resemblance.

7.7 Experiments with Object C

For object C experiments No. 6 to No.8 are conducted (see Figs 7.22 to 7.25 and Tables 7.6 and 7.7). In experiment No. 6 the object is placed on the base and in experiments No. 7 to No. 8 it is studied while it is placed on one of its sides. Object J is roughly the same as object C. Because of this it is included in all three experiments in order to assess how two similar objects would react in the process.

In experiment No. 6 object C, is correctly identified as the object present in the scene. This is achieved with 12 votes which is the maximum possible to obtain. Also object J, is rejected (i.e. with 7 votes) which demonstrates the feasibility of identifying similar geometrical shapes, provided the appropriate parameters are established. In these two experiments, object E, scores 7 which is rather high. The reason for this is discussed in detail in the section 7.9.

Note that there are some orientations in which object C cannot be recognised. As an example, imagine EB and DC in Fig.(7.22) placed perpendicular to the stripe lines. In this setting only two parallel edges (i.e. ED and DC) would be extracted which is obviously inadequate for recognition.

In experiments No.7 and No.8 only one surface is exposed to the scanning process. In experiment No.7 recognition is achieved with 5 votes which is the maximum achievable. That is closely followed by object A, with 4 votes. This closeness of votes is not a coincidence since DEFG in Fig.(7.1) is very similar to ABHF in Fig.(7.3). This example points to the fact that in order to recognise an object the overall structure of extracted lines must be unique.

The result of experiment No. 8 is a rectangle as shown in Fig(7.25). Here, AB and CD are of the length of the original cube, but AD and BC are longer because of the cut across the cube. However, when the recognition process is performed all objects scored 4 votes (see Table(7.7)). This is because if all edges are included in the database then all models would contain at least one square . Since rectangle ABCD in Fig (7.25) is roughly the same as the square, it is matched with all those squares. This further illustrates the necessity of having a unique combination of lines in order to distinguish an object.

7.8 Experiments with Object D

Object D is used in experiments No.9 to No. 11 (see Figs 7.26 to 7.29 and Tables 7.8 to 7.9).

In experiment No.9 the object is positioned on the base and recognised with 9 votes which is rather low considering 7 edges are extracted. This is mainly due to the poor extraction of point C which resulted in improper coordinate calculation and thus loss of votes.

In experiment No. 10 edges of the base and in experiment No. 11 all edges are included in the models. In both cases the object is recognised with a majority of votes.

Closer examination of the results also revealed that in experiment No. 11 the object was actually recognised with three different TTM matrices (each with the 5 votes) rather than with one. In this experiment recognition is performed with the edges extracted from the two adjacent surfaces as shown in Fig.(7.29). By comparing this with Fig.(7.4) it can be noted that there is more than one possibility of having the same combination of surfaces. As an example the combination of BCF and EFC is almost the same as BCF and ABF (or ABF and AFE). Each of these combinations can match with lines extracted in the scene and produce its own TTM matrix. To determine the pose of the object relative to the model the winning TTM matrix can be used. However in this situation if the faces of the object are marked then extra information would be required for identifying them (such as two cameras with different views).

7.9 Experiments with Object E

For object E experiments No.12 and No.13 are performed (see Figs 7.30 to 7.33 and Tables 7.10 and 7.11).

For experiment No. 12, twelve edges are extracted which can make a maximum 36 possible votes (for the combination of every two non parallel edges). However, results show that the object is recognised with 60 votes. Although the object is identified correctly, the number of votes considerably exceeded the maximum possible votes. This is because some of the votes were obtained from non correspondent lines which then clustered with proper votes. As an example suppose, the four edges at the third floor of the object (i.e. IJKL in Fig.(7.30)) are matched with those of the third floor of the correspondent model. With this combination a maximum of 4 votes can be achieved. Now, imagine the third floor of the object is matched against the third and also the second floor of the correspondent model. With this setting, the number of votes must remain the same since only two top floors must match. However, in practice it is found that the number of votes has risen to 8 votes. This shows that the TTM matrices obtained from the two top stairs are not distinguished, and therefore are clustered in one matrix. This is because for some non correspondence situation (such as KJ and KL versus GF and GH) rotational elements of the matrix remain the same and only translational elements of the matrix can distinguish them. Since translational parameters are too wide for this case, matrices mixup and create a large number of votes.

In selecting parameters empirical data together with two example objects are used to test and set appropriate values. While this is found to be adequate for most cases, it proved to be insufficient for some shapes such as object E (i.e. a lower parameter is required). However, in the present experiments the same values as those of the rest of the experiments are used. This is because the aim is to assess the behaviour of the system with parameters set in general rather than for each individual object.

In this experiment the running time is relatively high (i.e. 10 minutes 58 seconds). The reason for this is the comparatively large number of edges. In addition to that, all the matches involve two sides of the rectangle (longer and shorter sides). The shape and angle between them is roughly the same. In this condition, the pruning procedure is not very effective.

In experiment No.13, only one face of the object is exposed to scanning. Recognition is achieved with 8 votes. Here also, because all edges of the object are used running time is relatively high.

7.10 Experiments with Object F

For object F experiments No. 14 and No. 15 are conducted. (see Figs 7.34 to 7.37 and Tables 7.12 to 7.13). This object stretches the capability of the edge extraction to its limit and shows the shortcomings of edge detection in separating some of the lines. This is mainly due to the open obtuse angles and short edges which can result in having two or three lines derived as one and not distinguished (i.e. such as AB +BC in Fig.(7.35)).

In experiment No. 14 the object is recognised with only 4 votes which is rather low. In the programme that is used, rotational angles of TTM matrix are mostly determined by the length of the edges. Since in this object the edges are generally short, any imperfect extraction of edges can alter the TTM matrix significantly. Because of this the present system is not particularly suitable for objects with characteristics such as F. But, it has to be emphasised that other models also gained not more than one or two votes which allowed the object to be recognised with only four votes.

For experiment No. 15 the object is recognised with the maximum 6 possible votes from the four proper edges that are extracted. These examples demonstrate that recognition can be achieved with the few lines provided those that are extracted are specific to the object.

7.11 Experiments with Object G

For object G experiments No. 16 and No.17 are presented (see Figs 7.38 to 7.41 and Tables 7.14 to 7.15).

For experiment No. 16 line CG is partially obstructed by the body of the object (i.e. by MBDGAC in Fig.(7.38)) and QF is extracted in two segments instead of one (i.e. QF and GF). Results for this experiment are given in Table (7.14) where first it is assumed that QF is to be extracted as one line. Since, GF can be potentially extracted in segments rather than as a whole recognition is also performed with all possible segments included in the database. Here, the object is recognised with an improved 20 votes compared with the previous 12 votes. This shows the necessity of including more lines in the database in cases where the outcome is not certain.

In experiment No. 17 the two triangular surfaces obtained from the scene are depicted (Fig.(7.40). Recognition is conducted with six edges and the object is recognised with 15 votes which is the maximum that could be achieved.

7.12 <u>Experiments with Objects H and J</u>

In experiment No. 18 the object H (i.e. cube) is placed on the scene (see Figs 7.42 and 7.43 and Table 7.16). Here, only the top surface of the cube is exposed to the scanning procedure and all four edges of the square are extracted. In this experiment models of the first category of tests are used in which none of them contain the square (except cube) and recognition is achieved with 4 votes. If any of the models contained the square then similar to the case in experiment No.8 recognition would have been undecided. This demonstrates the fact that not only the combination of unique edges need to be extracted from the scene but models also must uniquely represent those features.

Since only four edges are involved in the process (i.e. from the object) running time is shortest of all experiments (i.e. 44 seconds).

In experiment No. 24 object J which is roughly the same as object C is placed on the scene and distinguished with a majority of 9 votes (see Figs 7.44 and 7.45 and Table 7.17). This illustrates the possibility of recognising objects with approximately the same shape when it is compared with experiment No.6 (i.e. for object C).

In order to determine the extent to which two objects can be distinguished, the model of the object can be gradually distorted. Then according to the recognition results an estimate of its limits can be assessed and established.

7.13 <u>Experiments with Two Objects</u>

Initially the system is meant to recognise objects one at a time and programmes are written to comply with this requirement. As an example, in section 5.3 in the preprocessing chapter, only two middle points are anticipated on each row. This may double if two objects are placed in the scene (i.e. middle points of stripes in the image). Because of this the system cannot generally cope with more than one object at a time. Also because of the matching process the system is not appropriate for occluded objects, since the matching process requires the full length of the lines to be extracted. Despite these, within limitations of the system three experiments (i.e. No.20 to No.22) with two non occluded objects are conducted (see Figs. 7.46 to 7.51 and Tables 7.18 to 7.20).

In experiment No. 20 objects C and D are placed on the base and edges of both objects are extracted. Then all the edges are participated in the process and the first two high scoring models are successfully distinguished.

In experiment No.21 objects B and C are placed on the base and object C is correctly recognised but B and E scored the same votes (i.e. 11 votes for C and 6 votes for both B and E). This is because the top rectangular shape of C is mixed with the rectangular steps of E and created excessive votes (i.e. as in experiment No.12 for object E). This example shows how in the case of two objects, one object can interfere with the recognition of the other one.

In experiment No. 22 objects B and E are placed arbitrarily in the scene and both objects are successfully recognised with nine votes. However, because of the comparatively large number of edges (i.e. from objects and models), recognition took a relatively long time to conclude (see tables related to experiments).

7.14 Experiments with Two and Three Connected Lines

In experiment No. 23 recognition results obtained from two and three connected edges are presented (see tables 7.21 to 7.23). Two connected edges involves less processing than the general case of two lines in any combination. But, for three connected edges this depends on how lines are selected and matched. However, since three connected lines have less interpretation than general two lines, fewer votes for models other than object can be expected.

In table (7.21) the sample of experiment No.1 (for A) is run for two and three connected lines again. In both cases the object is recognised with a reduced majority of 6 votes compared with the original 14.

For two connected lines some of the previous operations do not take place and therefore fewer votes than in the general case are achieved. Also for the models other than the object no significant changes occurred.

In the case of three connected lines, the reduction of votes can be attributed to those lines that previously produced votes and now combined with the third line. Then as a result of extra pruning, some of them are eliminated. Here, most of the models were heavily suppressed and some of them acquired no vote at all. This confirms the primary assumption that with three connected lines fewer interpretations can be found.

In table (7.22) the sample of experiment No.6 (for C) is run again. For two connected lines the object is recognised with a reduced majority of 8 votes which is the maximum expected. For three connected lines the object is recognised with 10 votes. Here, similar to the above most of the models scored fewer votes than the original result (with the exception of object E).

In table (7.23) the sample of experiment No.12 (for E) is run again. In both cases the object is recognised with almost the same number of votes (i.e. 21 and 20). With two connected lines no improved results in terms of suppression of models can be expected and it only results in fewer operations than the general case. However for the particular shape of object E using two connected lines could be as advantageous as three connected lines.

In these experiments the running time for two connected lines is normally less than for three connected lines. This is because in the case of three connected lines if two of the three lines are parallel then the number of operations actually doubles (for matching reasons, as explained in chapter 6 section 11).

7.15 Effects of Pruning and Matching

In experiment 24 effects of pruning and matching are examined (see Tables 7.24 and 7.25). In it samples of experiments No.1 and No 2 are processed again and in each case first, pruning is removed and then matching is taken out of the process (while pruning is restored).

Tables (7.24 and 7.25) show that when prunning is removed the number of votes stay virtually the same as the original results. This indicates that the parameters chosen for pruning are correctly selected (i.e. section 6.7 to 6.9 in chapter six). For the sample of experiment No 1 where the object is placed on the base the total running time is increased significantly to 53 minutes 5 seconds (compare with the 3 minutes 55 seconds with pruning). For the sample of experiment No. 2 (where all edges of the object are included), the total running time rises to a staggering 157 minutes 11 seconds (compared with 9 minutes and 16 seconds with pruning).

Tables (7.24 and 7.25) also show effects of removing the matching operation. In the programme that is written enough memory for each model is allocated so that 300 TTM matrices can be stored. In these tables those models that exceeded this amount are indicated by 999. Obviously, if pruning is also removed then much more memory space would be required. The total running time for the sample of experiment No.1 is 4 minutes and 31 seconds and for the sample of experiment No.2 is 11 minutes 43 seconds which compared with the original results is only a moderate increase.

These experiments demonstrate the importance of having the pruning operation in place since without it substantial increase in running time can be expected. However, with pruning in place and matching removed only a moderate increase in running time can be anticipated.

7.16 Objects and Extraction of the Edges

The following number of difficulties that were encountered during the experiments and related to edge extraction are specified:

1) Size of an edge.
2) Relative position of an edge with respect to the stripe.
3) Slope of an edge.
4) Angle between two connected edges.
5) Angle between two planes.
6) Blockage of an edge by part of the object.
7) Coordinates of two ends of the lines.

For the first case the length of an edge has to be long enough to allow a minimum number of scans cross it. Otherwise it cannot be identified and extracted. An example of this is line DE in Fig 7.10 in experiment No.1 . Because this edge is relatively short in some positions it may potentially be left undetected. If the edge is of adequate length but is positioned almost parallel or along the stripe then it cannot be scanned. In this situation even some of the long edges can remain undetected. Some of the examples are CH in experiment No.1 and BC in experiment No.4. Difficulties in scanning an edge can also arise when its slope is too steep with respect to the surface of the table . Examples of this, are lines KJ or HL in object F in Fig.(7.6). Here, if the object is placed on its AJHGM base then in some positions they are quite difficult to scan and potentially can be left undetected. Also in the present experiments (because of the position of the camera and projector) none of the lines that are perpendicular to the surface are extracted.

If the angle between two lines is obtuse and quite wide then in some positions distinguishing them can pose difficulty. An example of this is BC and CD in experiment 15 where they are extracted together as one line. This can also apply to the angle between two planes. If the angle between two planes is quite wide then the edge between those planes may be left undetected. The line KL in object F, is a potential example of this case. Also, two ends of the internal line AC in experiment 11 are extracted short from their corner. In some of these cases, such as short lines or connected lines, increasing the number of scans can improve the results.

For the case of blockage of a line by the body of the object experiment No.16 can be mentioned where GC is obstructed by MBDGAC and only partially visible from the camera.

In extracting edges coordinates of two end points are taken and the rest of the points in between are not used. If any of the end points fluctuate or are displaced then the line may become useless in matching. Because of that in many experiments a fewer number of votes than otherwise expected is obtained. To counter this, it is possible to fit a line along the edge points. But determining the exact termination of the lines can pose difficulty of its own.

7.17 Summary and Comment

In this chapter, analyses of the recognition results for various objects and shapes are given. Following that a brief account of the findings are outlined.

In order to recognise an object the overall structure of the lines extracted from the scene has to be unique (i.e. among the models). In most of the cases adequate edges in satisfactory condition are extracted to allow recognition to proceed. However, in some of the cases recognition is achieved with quite close votes. This is mainly due to the similarities in shape and structure that are involved. In some positions or orientations some of the objects cannot be recognised because not enough edges suitable for the processing can be extracted. Also if the overall structure of the lines is not unique then recognition would be undecided. Objects can be recognised with edges of only one surface, provided their structure is not repeated elsewhere.

Generally, parameters chosen for the system complied with requirements of the system. However, for some objects improved results can be expected if parameters are specifically adjusted for them (i.e. such as translational elements of the matrix in object E). This is also pointing to the fact that parameters selected for a group of objects may not be adequate with some other shapes and characteristics.

To determine the pose of the object relative to the model the TTM matrix can be used. But for the situation where several matrices each with the same number of votes occur extra information is necessary (i.e. such as two cameras in different positions).

In terms of edge extraction, some difficulties with short edges or edges which are positioned along the stripe lines, or edges between two obtuse surfaces are encountered. Also on a few occasions two connected edges are identified as one line. For some cases, such as short edges, improved results can be achieved if the number of scans increases. Also, some edges can be potentially extracted in segments rather than as a whole (such as in object G). By including possible segments as well as the whole line in the model, improved results are achieved.

As far as robustness is concerned, in most of the cases objects are identified correctly even though in some cases only a few edges are extracted (such as in the case of object F). This is when those few remaining edges adequately represented the object. However, in some cases inadequate extraction of one line can result in the loss of a whole group of votes. Votes may not be accounted for because of the limitation of equipment or edge detection or the specific position of the object relative to the stripe line rather than any inadequacy of the recognition method.

The feasibility of distinguishing objects with roughly the same shape is demonstrated. However in this case the model needs to be gradually distorted so that the exact limit of it can be established.

It is shown that more than one object can be recognised at a time. But, in some cases the presence of one object can interfere with the recognition of the other one. Also it is shown that with the three connected lines, models other than the object can be suppressed. However in terms of the efficiency of the process this is very much dependent on how the three lines are selected and matched. For some objects, (such as E) using two connected lines can be as advantageous as three lines with less processing to perform.

Generally processing time is proportional to the number of edges and in many cases a modest increase in the number of lines can considerably extend the total running time. The introduction of a pruning procedure very significantly lowered the running time and enhanced the performance of the system. In the situation where pruning has taken place, and a relatively small number of edges are involved, a matching procedure only moderately improved the efficiency. For the models which are the same or similar to the object pruning is difficult to achieve. This is also the case if a specific pattern is repeated in the model (or object). In this situation if the same pattern is also found in the object (or model) pruning cannot be easily performed. Inevitably this can result in longer processing times than those with no similarity.

OBJECT NUMBER ONE OR A

X,Y,Z coordinates (in mm.) of each corner provided AG to be X axis, AJ to be Y axis and AB to be Z axis.

A(0,0,0) B(0,0,22) C(0,9,28)
D(0,28,28) E(10,28,28) F(24,28,0)
G(24,0,0) H(20,0,7) J(0,28,0)
K(10,19,28)

Base: AGFJ

Edges included in the database for the first catagory of tests,

BC,CD,DE,EK,KC,KH,HG,EF

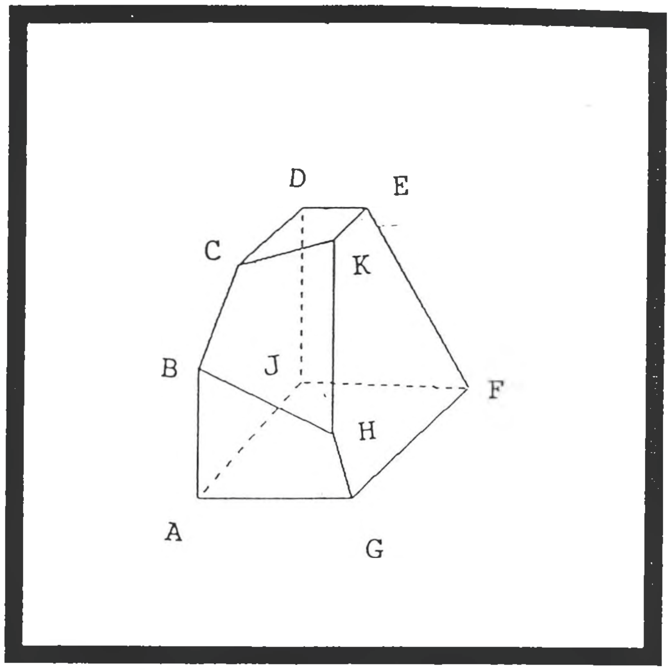


Figure 7.1

OBJECT NUMBER TWO OR B

X,Y,Z coordinates (in mm.) of each corner provided AF to be X axis, AG to be Y axis and AB to be Z axis.

A(0,0,0) B(0,0,28) C(0,28,28)
D(28,28,28) E(28,28,0) F(28,0,0)
G(0,28,0)

Base: AGEF

Edges included in the database for the first catagory of tests,

BC,CD,BD,FB,FD

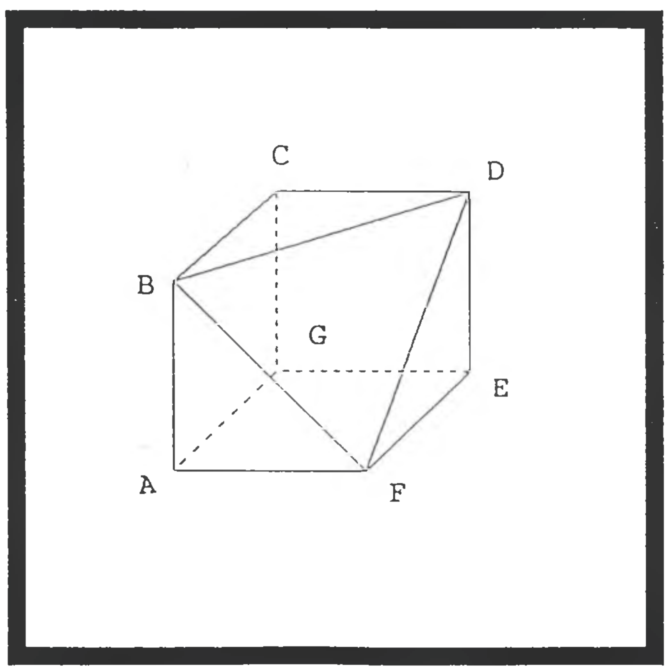


Figure 7.2

OBJECT NUMBER THREE OR C

X,Y,Z coordinates (in mm.) of each corner provided AF to be X axis, AG to be Y axis and AB to be Z axis.

A(0,0,0) B(0,0,28) C(0,28,28)
D(10,28,28) E(26,28,0) F(26,0,0)
G(0,28,0)

Base: AGEF

Edges included in the database for the first catagory of tests,

BC,CD,DH,HB,HF,DE

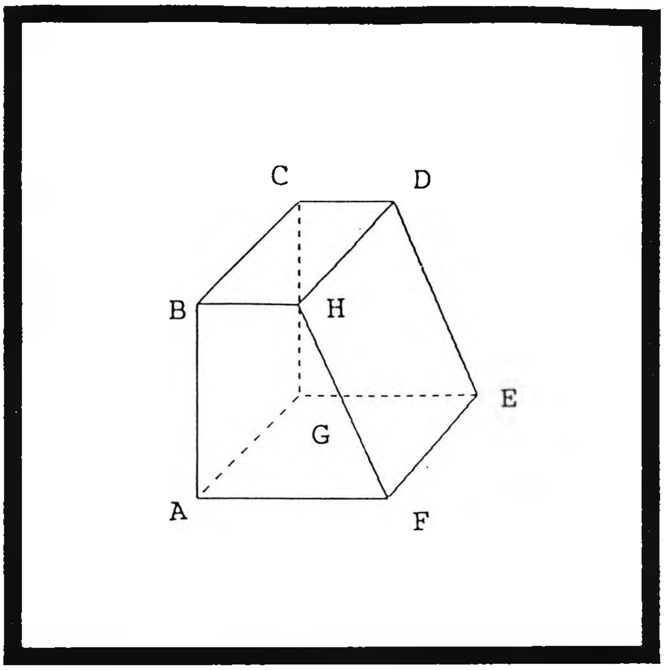


Figure 7.3

OBJECT NUMBER FOUR OR D

X,Y,Z coordinates (in mm.) of each corner provided GA to be X axis, GD to be Y axis and GB to be Z axis.

A(28,0,0) B(0,0,28) C(0,28,28)
D(0,28,0) E(28,28,0) F(28,9,28)
G(0,0,0)

Base: AGDE

Edges included in the database for the first catagory of tests,

AB,BC,CE,FE,FC,FB,FA

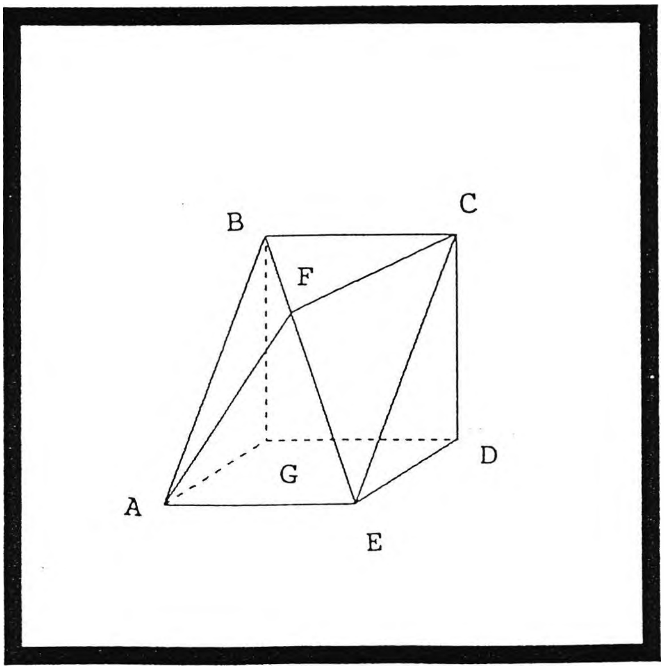


Figure 7.4

OBJECT NUMBER FIVE OR E

X,Y,Z coordinates (in mm.) of each corner provided AK to be X axis, AR to be Y axis and AB to be Z axis.

A(0,0,0)	B(0,0,28)	C(0,28,28)
D(10,28,28)	E(10,28,17)	F(20,28,17)
G(20,28,8)	H(28,28,8)	J(28,28,0)
K(28,0,0)	L(28,0,8)	M(20,0,8)
P(20,0,17)	Q(10,0,17)	T(10,0,28)
R(0,28,0)		

Base: ARJK

Edges included in the database for the first catagory of tests,

CB,CD,DT,TB,QP,PF,FE,EQ,ML,LH,HG,GM

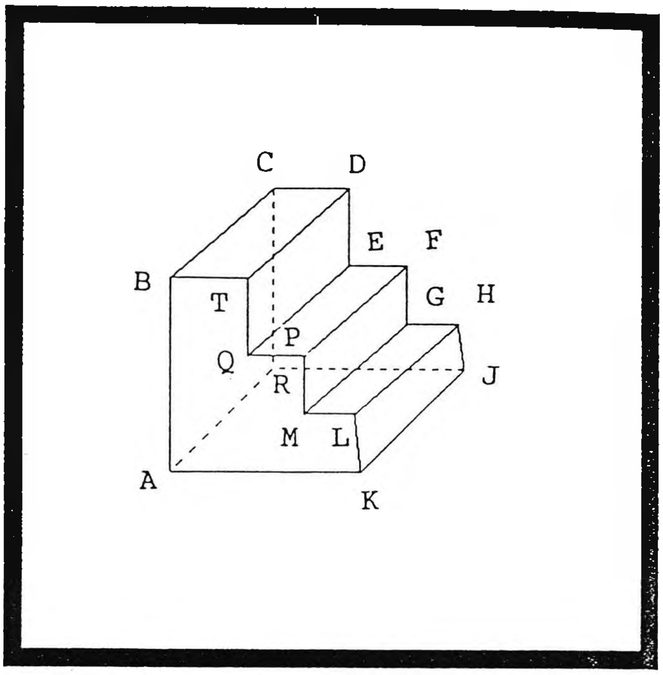


Figure 7.5

OBJECT NUMBER SIX OR F

X,Y,Z coordinates (in mm.) of each corner provided AJ to be X axis, AM to be Y axis and AB to be Z axis.

A(0,0,0)	B(0,0,23)	C(0,12,28)
D(0,28,28)	E(16,28,28)	F(28,28,23)
G(28,28,0)	H(28,9,0)	J(18,0,0)
K(7,0,20)	L(28,9,20)	M(0,28,0)

Base: AMGHJ

Edges included in the database for the first catagory of tests,

JK,KB,BC,CD,DE,EF,FL,LH,LK,CE

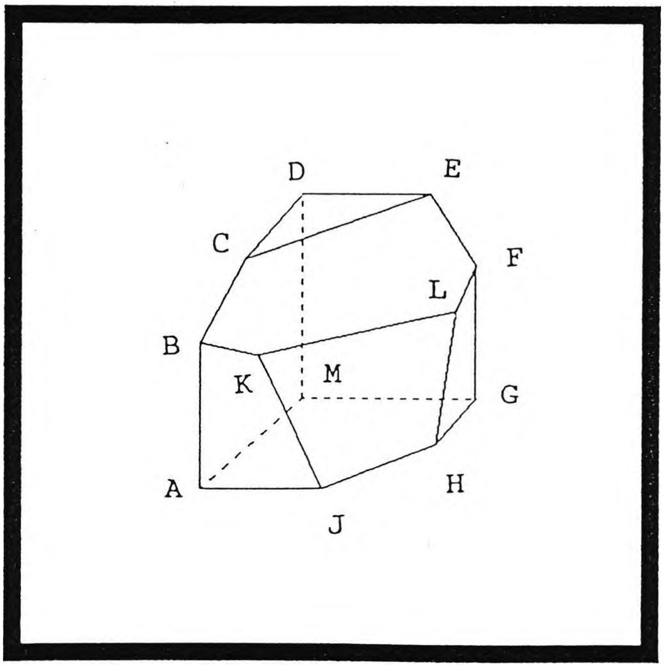


Figure 7.6

OBJECT NUMBER SEVEN OR G

X,Y,Z coordinates (in mm.) of each corner provided AJ to be X axis, AN to be Y axis and AB to be Z axis.

A(0,0,0)	B(0,0,28)	C(0,9,28)
D(0,19,28)	E(0,28,28)	F(28,28,0)
G(28,19,0)	H(28,9,0)	J(28,9,0)
K(18,9,28)	L(18,19,28)	N(0,28,0)

Base: ANFGHJ

Edges included in the database for the first catagory of tests,

BJ,CH,DG,EF,KL,KC,LD,BE

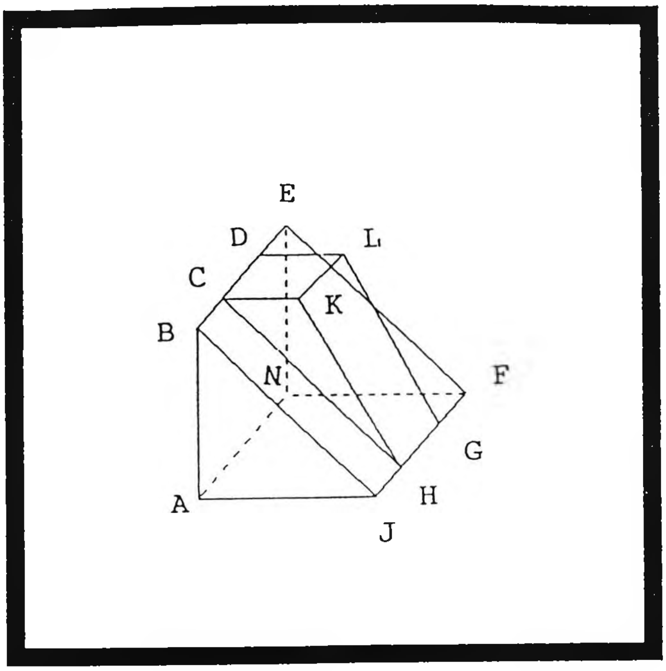


Figure 7.7

OBJECT NUMBER EIGHT OR H

X,Y,Z coordinates (in mm.) of each corner provided AF to be X axis, AG to be Y axis and AB to be Z axis.

A(0,0,0)	B(0,0,28)	C(0,28,28)
D(28,28,28)	E(28,28,0)	F(28,0,0)
G(0,28,0)		

Base: AGEF

Edges included in the database for the first and second catagory of tests,

BC,CD,DH,HB

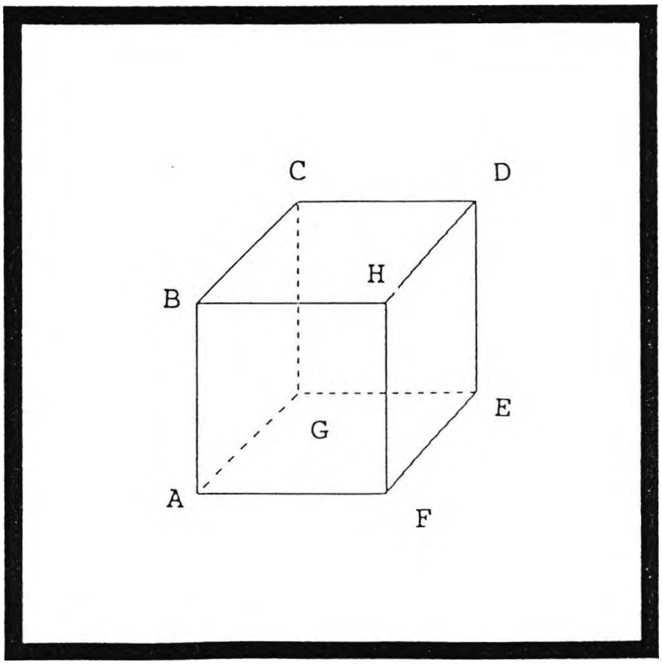


Figure 7.8

OBJECT NUMBER NINE OR J

X,Y,Z coordinates (in mm.) of each corner provided AF to be X axis, AG to be Y axis and AB to be Z axis.

A(0,0,0)	B(0,0,23)	C(0,28,23)
D(14,0,28)	E(28,28,0)	F(28,0,0)
G(0,28,0)	H(14,0,28)	

Base: AGEF

Edges included in the database for the first catagory of tests,

BH,HD,DC,CB,HF,DE

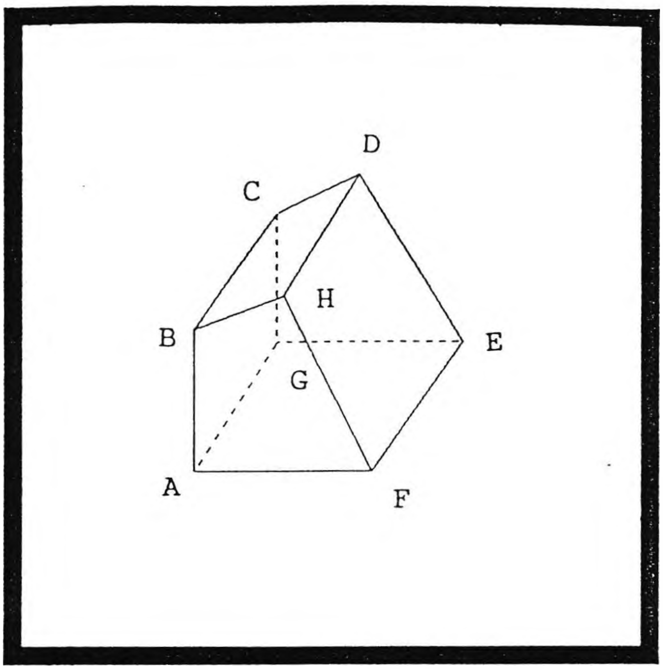


Figure 7.9

EXPERIMENT: NO. 1

OBJECT: A

OBJECT PLACED ON THE BASE

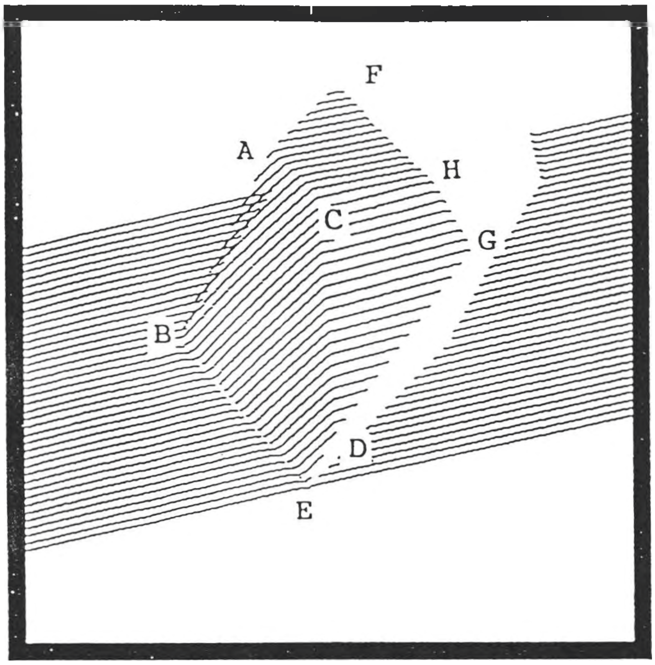


Fig.(7.10) Light-stripe image of the object

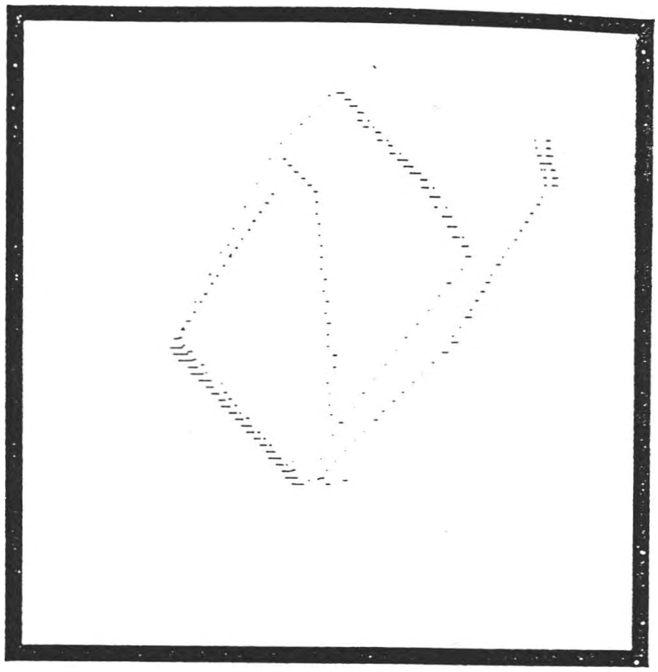


Fig.(7.11) Extracted edges (AB,AF,FH,AC,HG,CD,GD) EB

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES		0 MIN. 24 SEC.
NAMES & NO. OF EDGES IN MODELS	HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1 A 8 EDGES	14	0 MIN. 32 SEC.
2 B 5 EDGES	0	0 MIN. 10 SEC.
3 C 6 EDGES	1	0 MIN. 11 SEC.
4 D 7 EDGES	0	0 MIN. 14 SEC.
5 E 12 EDGES	1	0 MIN. 51 SEC.
6 F 11 EDGES	4	1 MIN. 13 SEC.
7 G 8 EDGES	1	0 MIN. 15 SEC.
8 H 4 EDGES	0	0 MIN. 2 SEC.
TOTAL TIME FOR THE MATCHING PROCESS		3 MIN. 28 SEC.
***** MODEL NUMBER 1		IS THE OBJECT *****

Table (7.1) Recognition results for experiment No.1
(Models include edges for first category of tests)

EXPERIMENT: NO.2

OBJECT: A

OBJECT PLACED ON A FACE OTHER THAN BASE

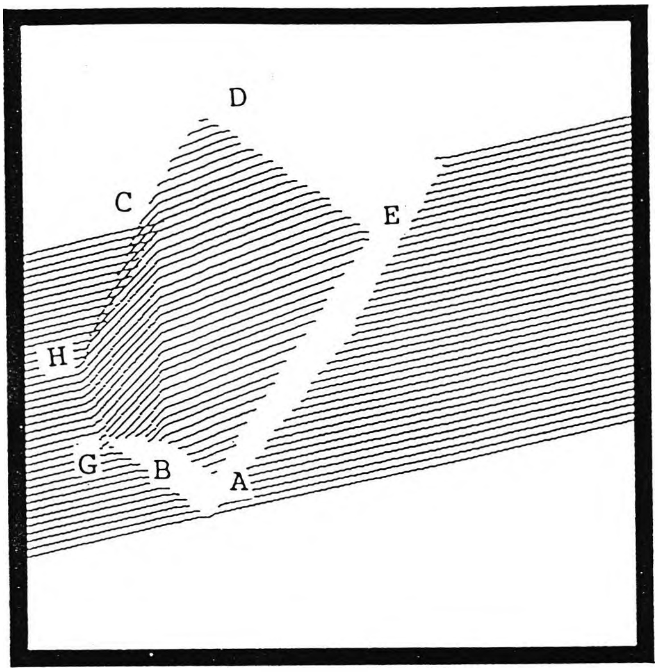


Fig.(7.12) Light-stripe image of the object. Placed on ABCDJ as defined in Fig.(7.1).

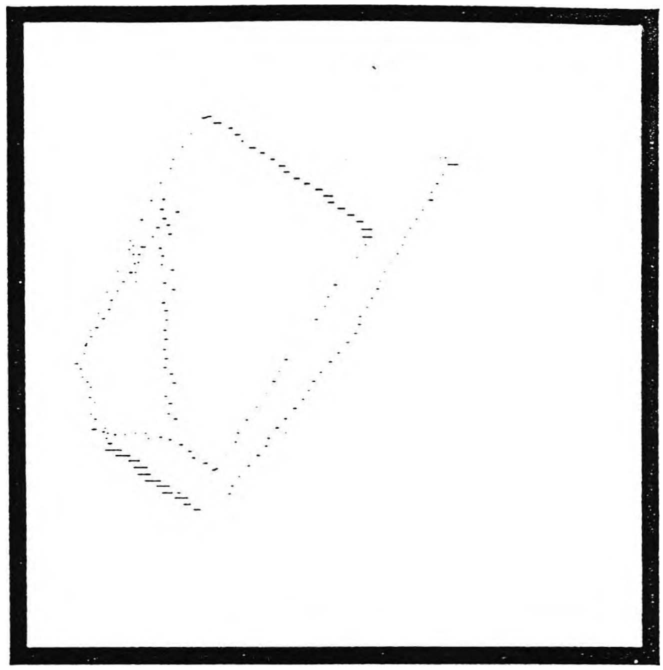


Fig.(7.13) Extracted edges (HC,CD,DE,EA,AB,BC,BG) GH

NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	14 EDGES	11	1 MIN. 19 SEC.
2	B	9 EDGES	1	0 MIN. 16 SEC.
3	C	12 EDGES	3	0 MIN. 43 SEC.
4	D	13 EDGES	1	1 MIN. 52 SEC.
5	E	24 EDGES	2	2 MIN. 14 SEC.
6	F	19 EDGES	4	2 MIN. 32 SEC.
7	G	16 EDGES	2	1 MIN. 7 SEC.
8	H	4 EDGES	1	0 MIN. 28 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				10 MIN. 31 SEC.
***** MODEL NUMBER 1 IS THE OBJECT *****				

Table (7.2) Recognition results for experiment No.2

EXPERIMENT: NO. 3

OBJECT: B

OBJECT PLACED ON THE BASE

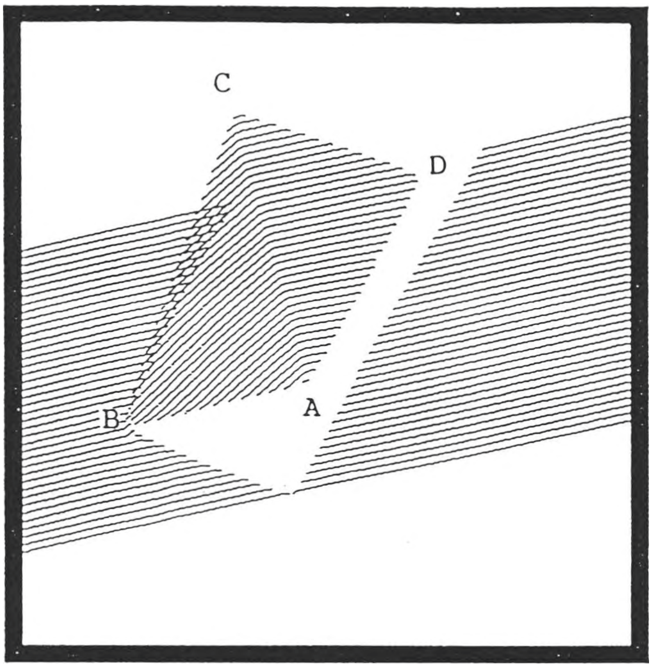


Fig.(7.14) Light-stripe image of the object

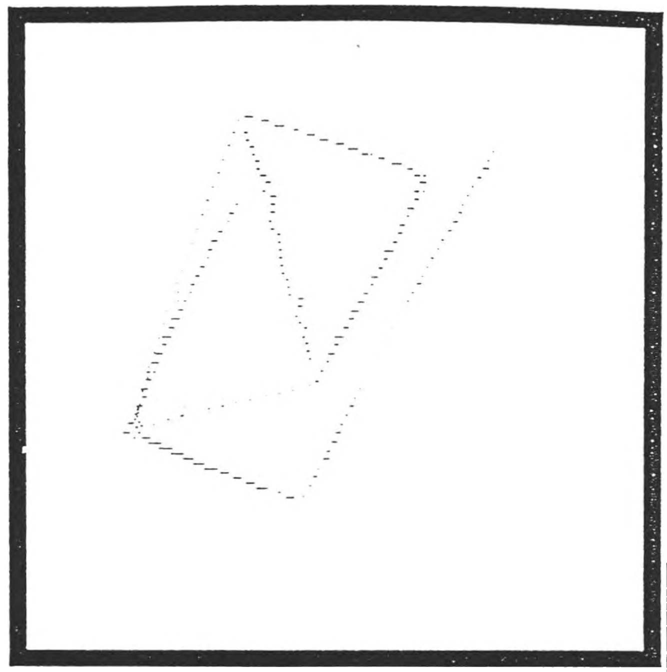


Fig.(7.15) Extracted edges (AB,AD,AC,BC,CD)

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES		0 MIN. 17 SEC.
NAMES & NO. OF EDGES IN MODELS	HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1 A 8 EDGES	0	0 MIN. 5 SEC.
2 B 5 EDGES	10	0 MIN. 22 SEC.
3 C 6 EDGES	1	0 MIN. 7 SEC.
4 D 7 EDGES	7	0 MIN. 52 SEC.
5 E 12 EDGES	0	0 MIN. 17 SEC.
6 F 11 EDGES	1	0 MIN. 12 SEC.
7 G 8 EDGES	2	0 MIN. 16 SEC.
8 H 4 EDGES	1	0 MIN. 10 SEC.
TOTAL TIME FOR THE MATCHING PROCESS		2 MIN. 21 SEC.
***** MODEL NUMBER	2	IS THE OBJECT *****

Table (7.3) Recognition results for experiment No. 3

(All models contain edges for first catagory of tests)

EXPERIMENT: NO.4

OBJECT: B

OBJECT PLACED ON THE BASE

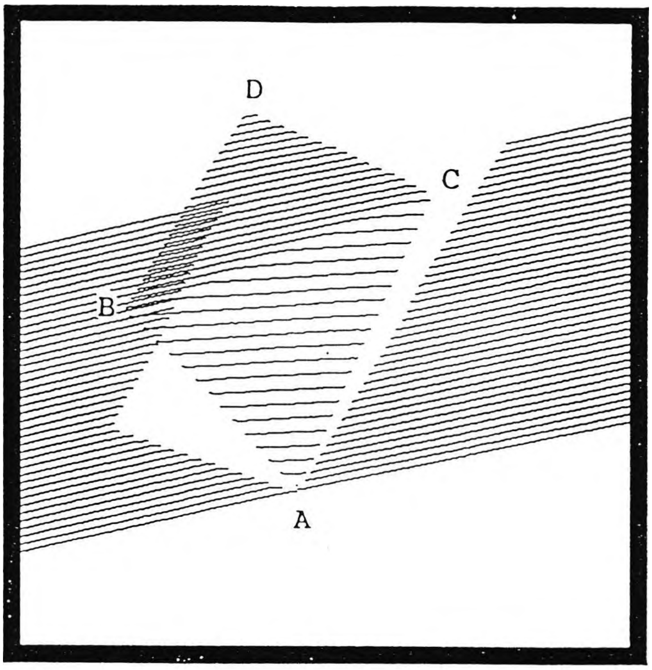


Fig.(7.16) Light-stripe image of the object

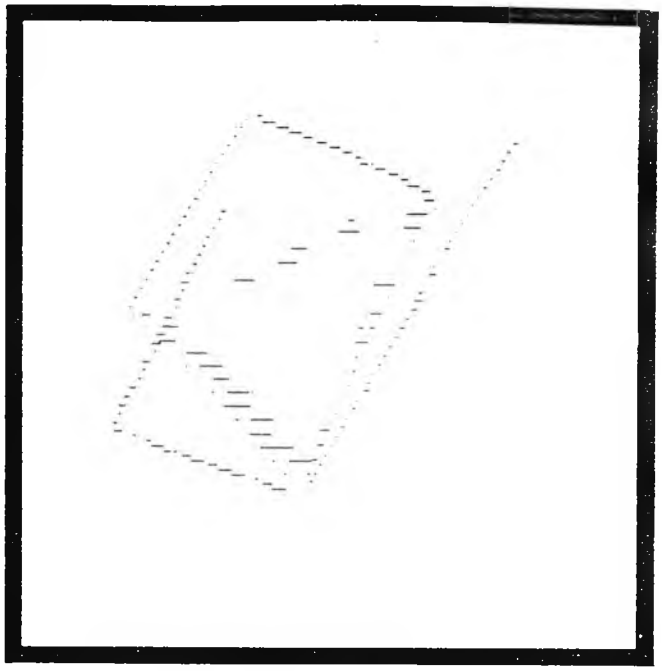


Fig.(7.17) Extracted edges (AB,AC,BC,CD,DB)

NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	8 EDGES	0	0 MIN. 5 SEC.
2	B	5 EDGES	6	0 MIN. 14 SEC.
3	C	6 EDGES	1	0 MIN. 7 SEC.
4	D	7 EDGES	5	0 MIN. 34 SEC.
5	E	12 EDGES	0	0 MIN. 17 SEC.
6	F	11 EDGES	1	0 MIN. 20 SEC.
7	G	8 EDGES	2	0 MIN. 11 SEC.
8	H	4 EDGES	1	0 MIN. 10 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				1 MIN. 58 SEC.
***** MODEL NUMBER			2	IS THE OBJECT *****

Table (7.4) Recognition results for experiment No.4
(All models contain edges for first catagory of tests)

EXPERIMENT: NO. 5

OBJECT: B

OBJECT PLACED ON THE BASE

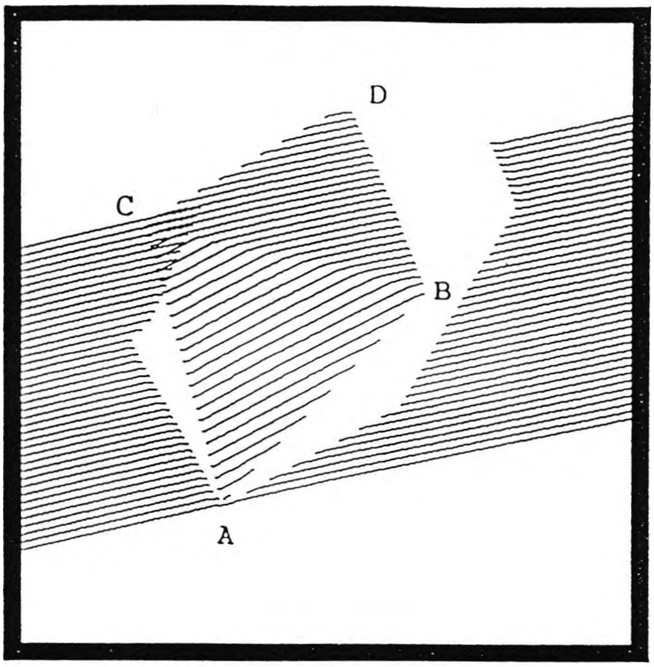


Fig.(7.18) Light-stripe image of the object

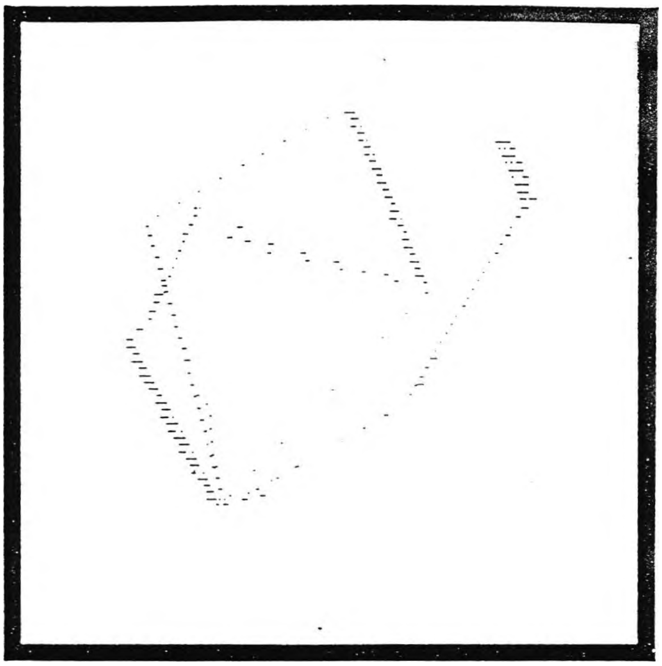


Fig.(7.19) Extracted edges (AC,AB,BC,BD,CD)

NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	8 EDGES	1	0 MIN. 9 SEC.
2	B	5 EDGES	6	0 MIN. 14 SEC.
3	C	6 EDGES	1	0 MIN. 7 SEC.
4	D	7 EDGES	4	0 MIN. 30 SEC.
5	E	12 EDGES	0	0 MIN. 16 SEC.
6	F	11 EDGES	1	0 MIN. 23 SEC.
7	G	8 EDGES	2	0 MIN. 16 SEC.
8	H	4 EDGES	1	0 MIN. 10 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				2 MIN. 5 SEC.
***** MODEL NUMBER			2	IS THE OBJECT *****

Table (7.5) Recognition results for experiment No.5 (All models contain edges for first category of tests)

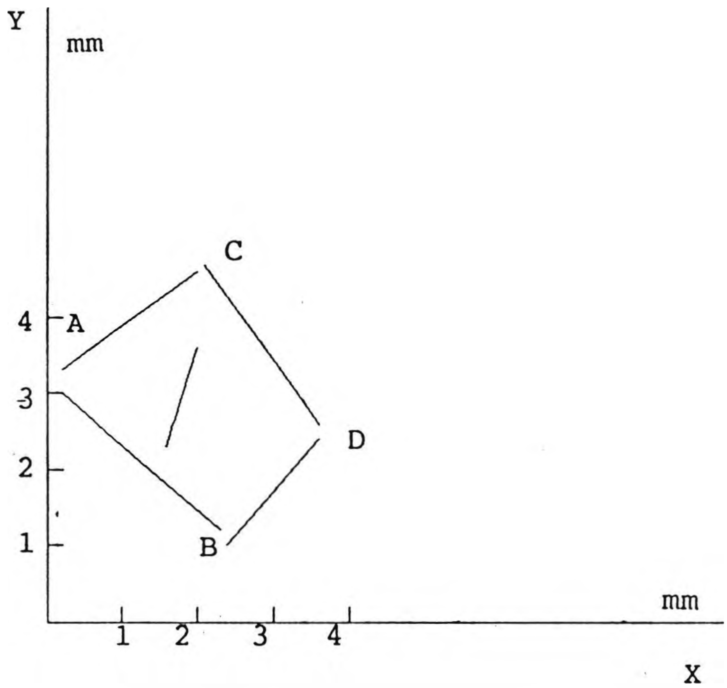


Fig.(7.20) Edges as they appear on XY plane for experiment No.5

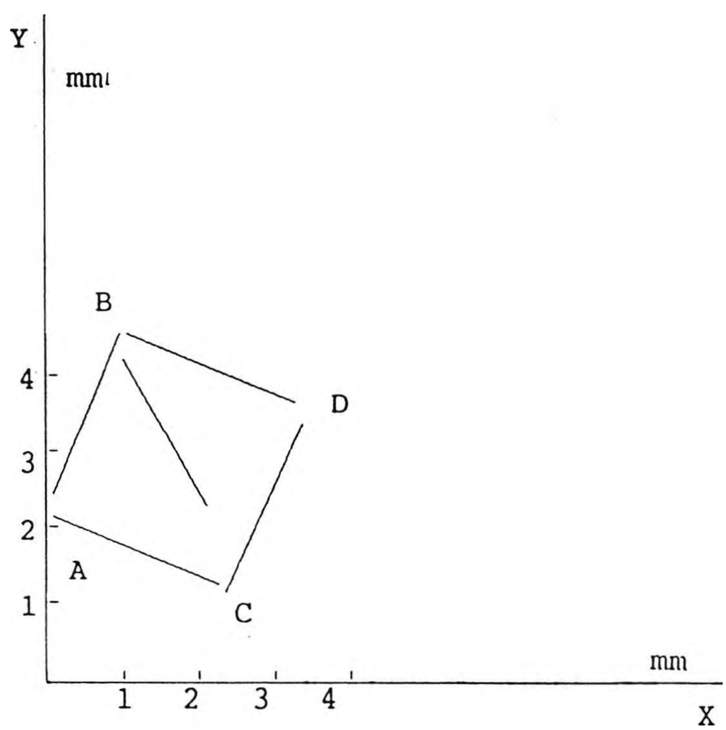


Fig.(7.21) Edges as they appear on XY plane for experiment No.6

EXPERIMENT: NO. 6

OBJECT: C

OBJECT PLACED ON THE BASE

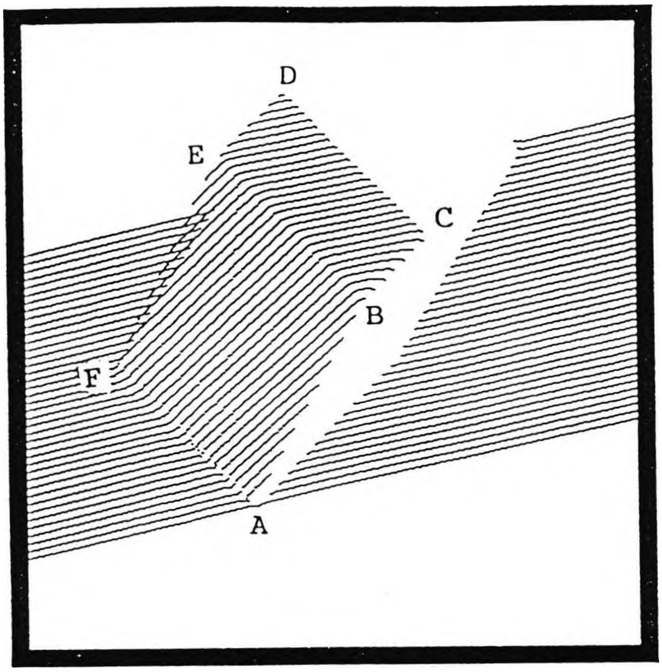


Fig.(7.22) Light-stripe image of the object

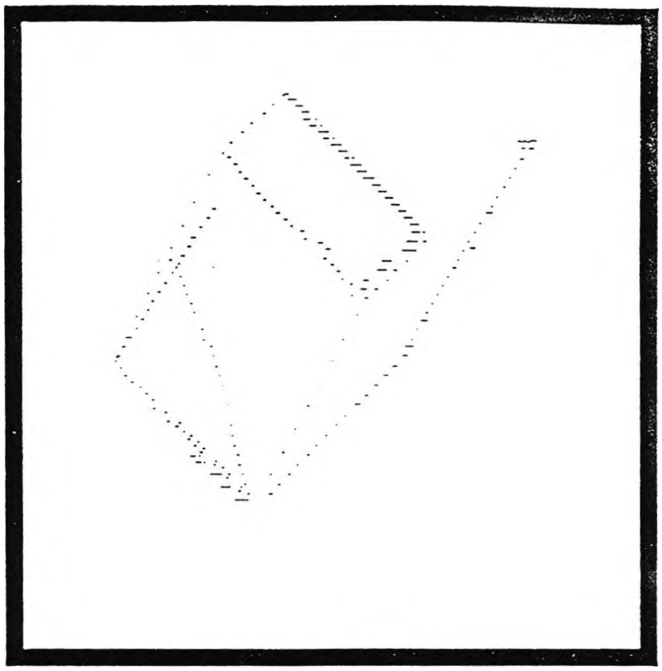


Fig.(7.23) Extracted edges (AB,BC,CD,DE,EB,FE)

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES			0 MIN. 21 SEC.
NAMES & NO. OF EDGES IN MODELS		HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A 8 EDGES	3	0 MIN. 20 SEC.
2	B 5 EDGES	1	0 MIN. 3 SEC.
3	C 6 EDGES	12	0 MIN. 27 SEC.
4	D 7 EDGES	1	0 MIN. 12 SEC.
5	E 12 EDGES	7	0 MIN. 55 SEC.
6	F 11 EDGES	2	0 MIN. 20 SEC.
7	H 4 EDGES	2	0 MIN. 7 SEC.
8	J 6 EDGES	7	0 MIN. 46 SEC.
TOTAL TIME FOR THE MATCHING PROCESS			3 MIN. 10 SEC.
***** MODEL NUMBER		3	IS THE OBJECT *****

Table (7.6) Recognition results for experiment No.6
Object J is introduced and Object G is not included.
(Models include edges for first category of tests)

EXPERIMENTS: NO. 7 AND NO. 8

OBJECT: C

OBJECT PLACED ON A FACE OTHER THAN BASE

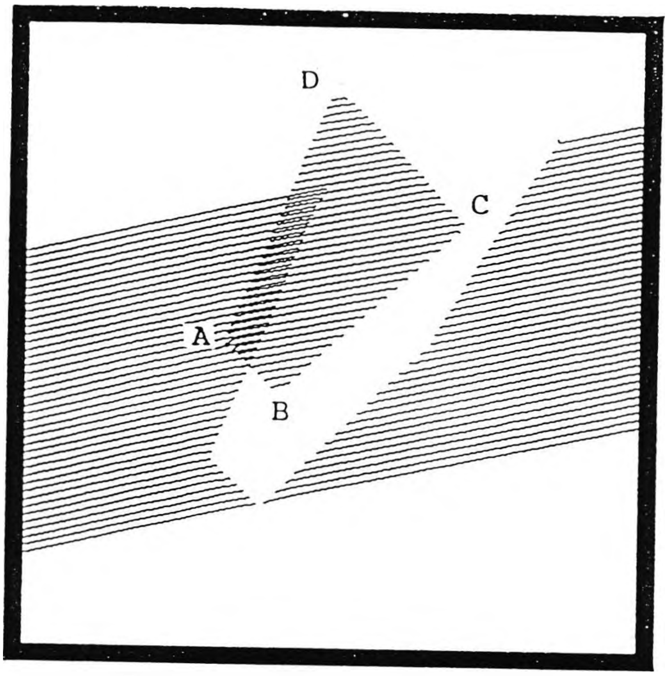


Fig.(7.24) Light-stripe image of the object for experiment No. 7 Placed on ABCG as in Fig.(7.3).
All four edges are extracted.

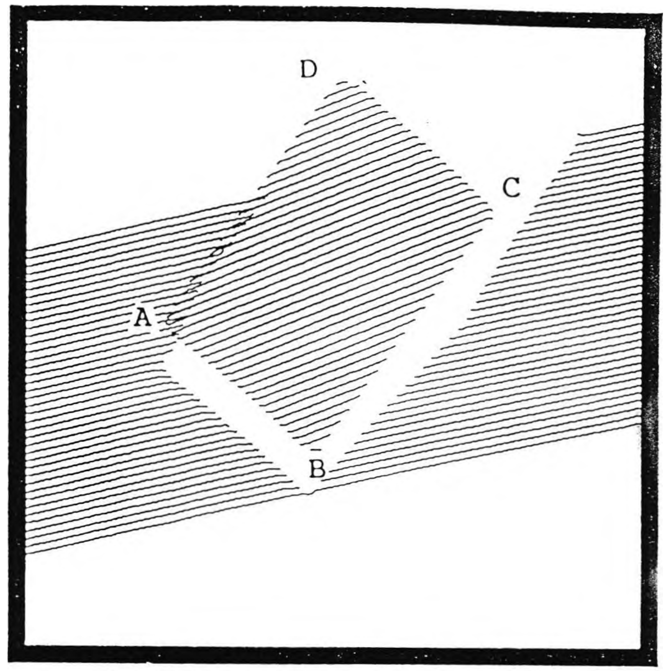


Fig.(7.25) Light-stripe image of the object for experiment No.8 Placed on ABHF as in Fig.(7.3).
All four edges are extracted.

NAME OF THE MODELS & NUMBER OF EDGES	HIGHEST SCORING MATRIX	
	EXPERIMENT NO. 7	EXPERIMENT NO. 8
1 A 14 EDGES	4	4
2 B 9 EDGES	1	4
3 C 12 EDGES	5	4
4 D 13 EDGES	1	4
5 E 24 EDGES	2	4
6 F 19 EDGES	2	4
7 G 16 EDGES	1	4
8 J 12 EDGES	1	4
CHOSEN MODEL	C	UNDECIDED

Table(7.7) Recognition results for experiments No.7 & No.8. In experiment No.7 one of the faces of A is roughly the same as Fig.(7.24) and its score is close to C. Experiment No.8 is undecided because all models contain square.(Models include all edges of the objects)

EXPERIMENT: NO.9

OBJECT: D

OBJECT PLACED ON THE BASE

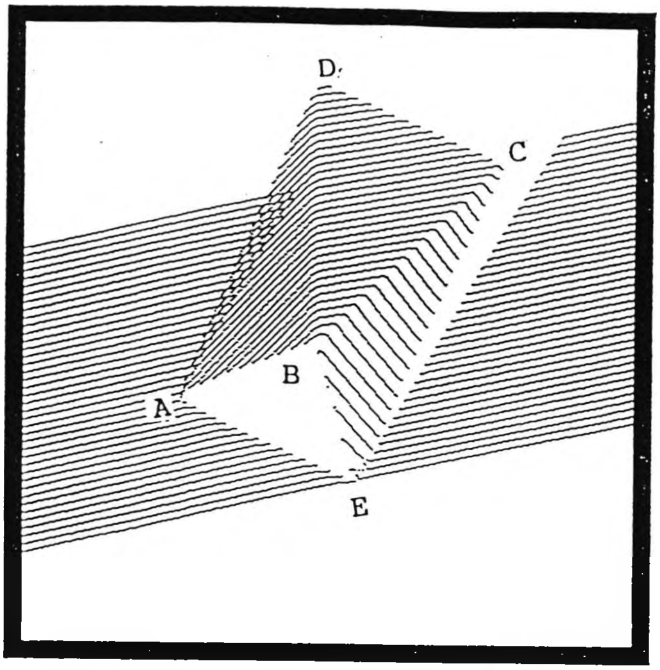


Fig.(7.26) Light-stripe image of the object Placed on base.

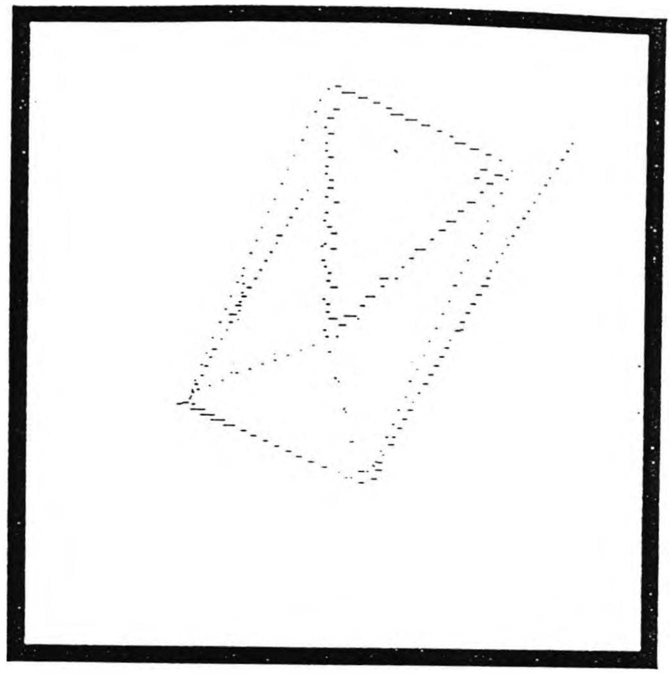


Fig.(7.27) Extracted edges (AB,BC,BD,DC,AD,BE,EC)

NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	8 EDGES	1	0 MIN. 9 SEC.
2	B	5 EDGES	2	0 MIN. 19 SEC.
3	C	6 EDGES	2	0 MIN. 8 SEC.
4	D	7 EDGES	9	0 MIN. 39 SEC.
5	E	12 EDGES	0	0 MIN. 22 SEC.
6	F	11 EDGES	1	0 MIN. 27 SEC.
7	J	8 EDGES	2	0 MIN. 10 SEC.
8	H	4 EDGES	1	0 MIN. 22 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				2 MIN. 36 SEC.
***** MODEL NUMBER			4	IS THE OBJECT *****

Table (7.8) Recognition results for experiment No.9
(Models include edges for first category of tests)

EXPERIMENTS: NO. 10 AND 11

OBJECT: D

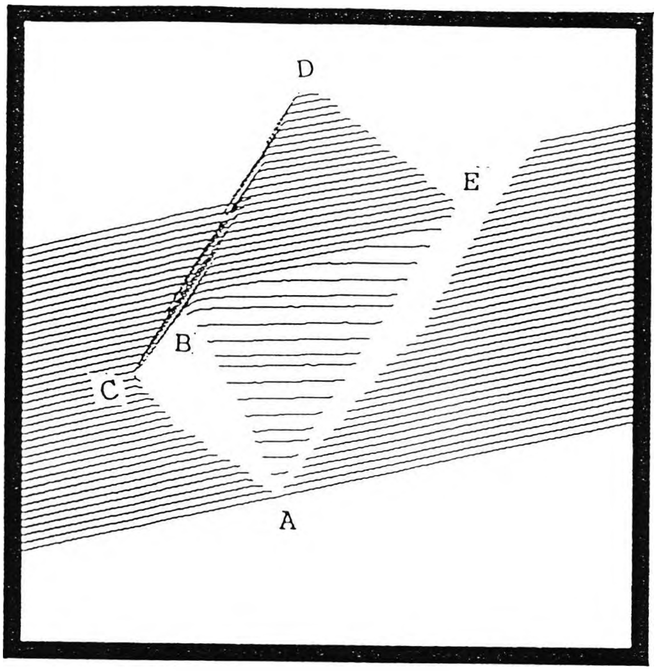


Fig.(7.28) Light-stripe image for experiment 9. Placed on GDCB as in Fig.(7.4). Edges of first category of tests are used in the models. (extracted edges AB,BC,AE,DE,CD,BD)

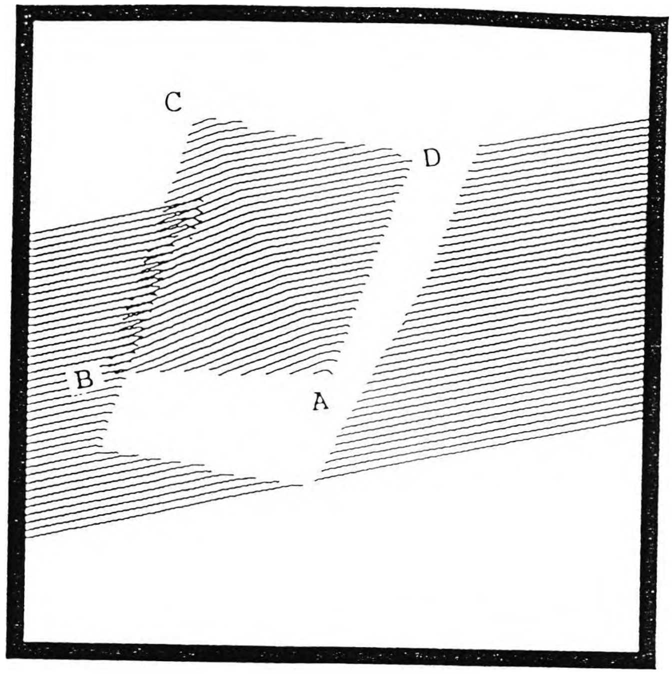


Fig.(7.29) Light-stripe image for experiment 9. Placed on AGB as in Fig.(7.4). All edges are included in the models.(extracted edges AB, BC,AD,CD,AC)(AC short at two ends).

MODELS & No.EDGES		EXPERIMENT 10 (BASE)	MODELS & No.EDGES		EXPERIMENT 11 (GENERAL)
		SCORE			SCORE
A	8	0	A	14	2
B	5	3	B	9	2
C	6	2	C	12	2
D	7	14	D	13	5
E	12	0	E	24	2
F	11	1	F	19	2
G	8	2	G	16	2
H	4	1	H	4	2
CHOSEN MODEL		4	CHOSEN MODEL		4

Table(7.9) Recognition results for experiments 10 and 11 shows object is recognised with majority of votes.

EXPERIMENT: No.12

OBJECT: E

OBJECT IS PLACED ON THE BASE

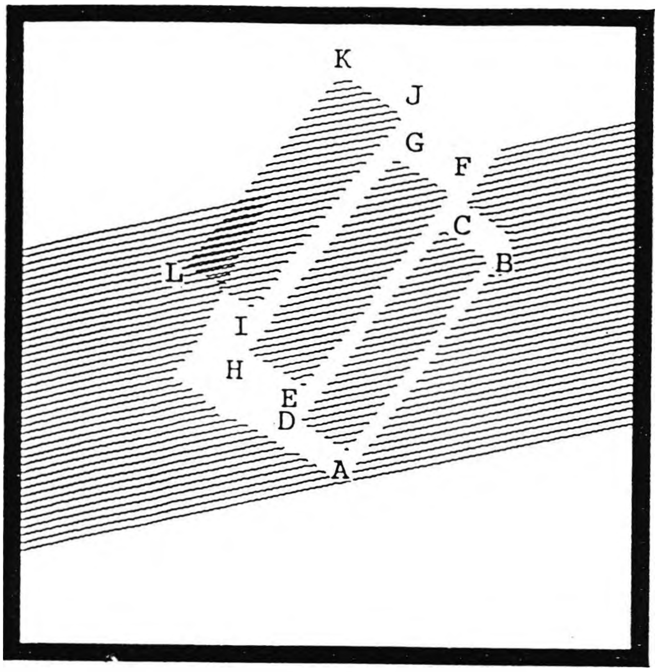


Fig. (7.30) Light-stripe image of the object Placed on base.

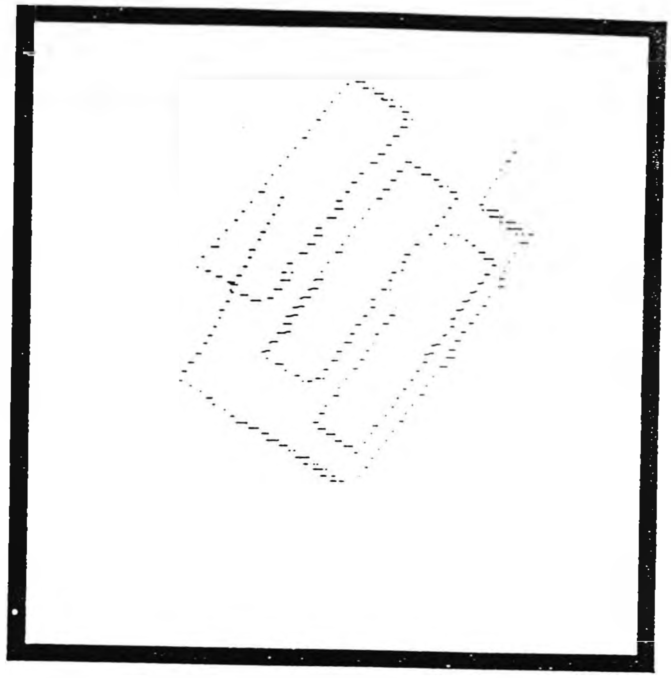


Fig. (7.31) Extracted edges

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES			0 MIN. 27 SEC.
NAMES & NO. OF EDGES IN MODELS		HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A 8 EDGES	2	0 MIN. 43 SEC.
2	B 5 EDGES	0	0 MIN. 6 SEC.
3	C 6 EDGES	6	0 MIN. 43 SEC.
4	D 7 EDGES	0	0 MIN. 13 SEC.
5	E 12 EDGES	60	7 MIN. 28 SEC.
6	F 11 EDGES	2	1 MIN. 17 SEC.
7	G 8 EDGES	1	0 MIN. 25 SEC.
8	H 4 EDGES	0	0 MIN. 3 SEC.
TOTAL TIME FOR THE MATCHING PROCESS			10 MIN. 58 SEC.
***** MODEL NUMBER		5	IS THE OBJECT *****

Table (7.10) Recognition results for experiment No.14 (Models include edges for first category of tests as defined in Fig.(7.5))

EXPERIMENT: NO.13

OBJECT: E

OBJECT PLACED ON A FACE OTHER THAN BASE

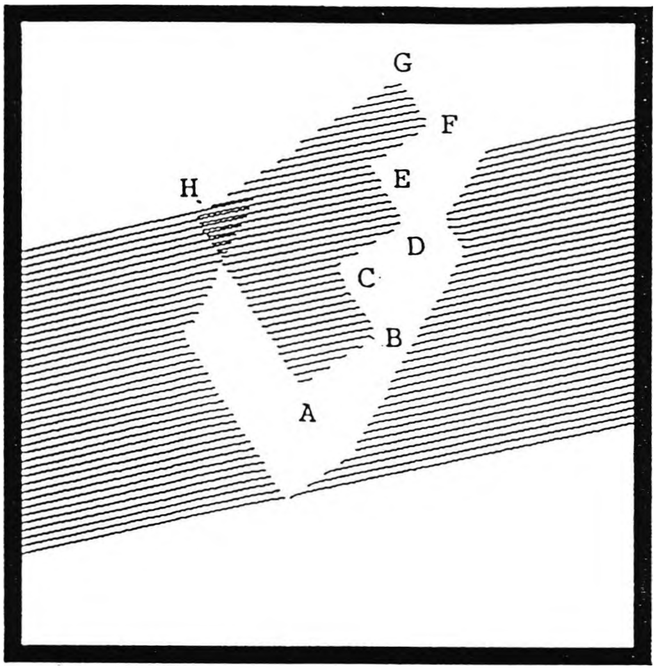


Fig. (7.32) Light-stripe image of the object. Placed on AKMQB as defined in Fig.(7.5).

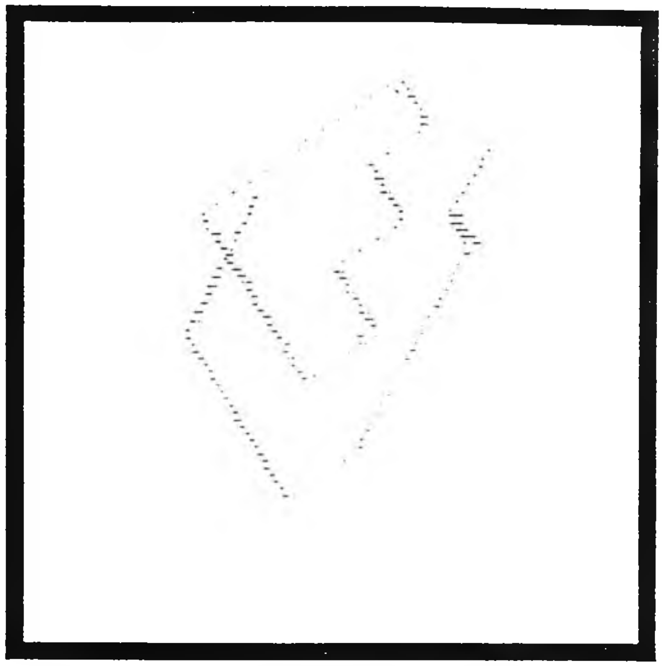


Fig. (7.33) Extracted edges

NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	14 EDGES	2	0 MIN. 59 SEC.
2	B	9 EDGES	1	0 MIN. 14 SEC.
3	C	12 EDGES	2	0 MIN. 39 SEC.
4	D	13 EDGES	1	0 MIN. 38 SEC.
5	E	24 EDGES	8	2 MIN. 32 SEC.
6	F	19 EDGES	2	1 MIN. 51 SEC.
7	G	16 EDGES	1	0 MIN. 55 SEC.
8	H	4 EDGES	1	0 MIN. 28 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				8 MIN. 16 SEC.
***** MODEL NUMBER			5	IS THE OBJECT *****

Table (7.11) Recognition results for experiment No.16 (Models include all edges)

EXPERIMENT: NO. 14

OBJECT: F

OBJECT PLACED ON BASE

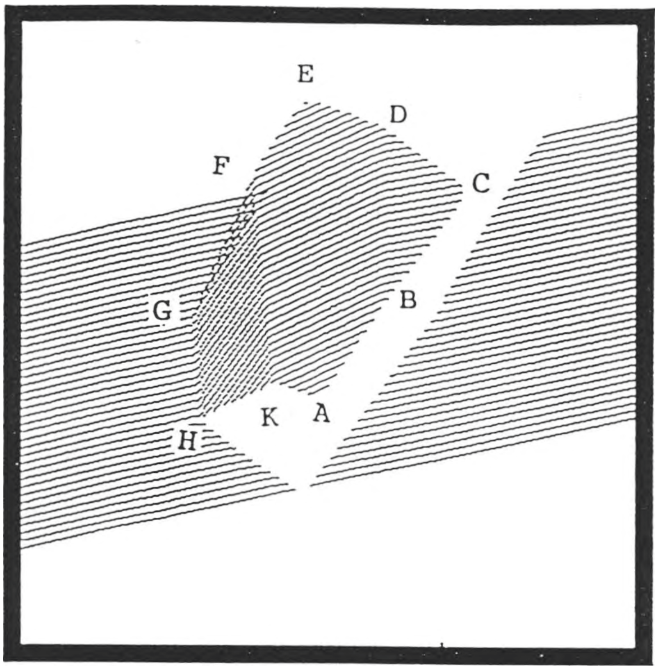


Fig. (7.34) Light-stripe image of the object Placed on base.

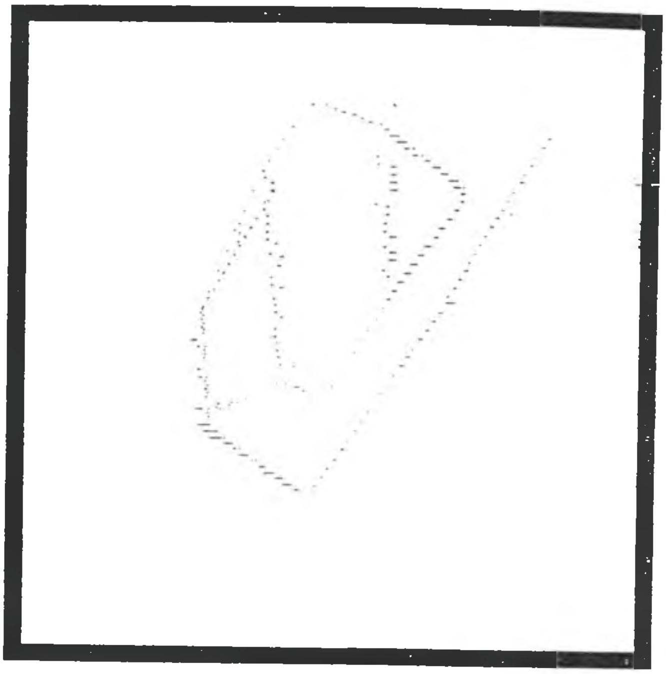


Fig. (7.35) Extracted edges (AB+BC), (AK+KH), CD,ED,FK FE,BD,FG

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES				0 MIN. 27 SEC.
NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	8 EDGES	2	0 MIN. 22 SEC.
2	B	5 EDGES	1	0 MIN. 6 SEC.
3	C	6 EDGES	1	0 MIN. 15 SEC.
4	D	7 EDGES	1	0 MIN. 9 SEC.
5	E	12 EDGES	2	0 MIN. 30 SEC.
6	F	11 EDGES	4	1 MIN. 3 SEC.
7	G	8 EDGES	1	0 MIN. 14 SEC.
8	H	4 EDGES	0	0 MIN. 2 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				2 MIN. 41 SEC.
***** MODEL NUMBER			6	IS THE OBJECT *****

Table (7.12) Recognition results for experiment No.14 (Models include edges for first category of tests as defined in Fig.(7.6))

EXPERIMENT: NO. 15

OBJECT: F

OBJECT PLACED ON BASE

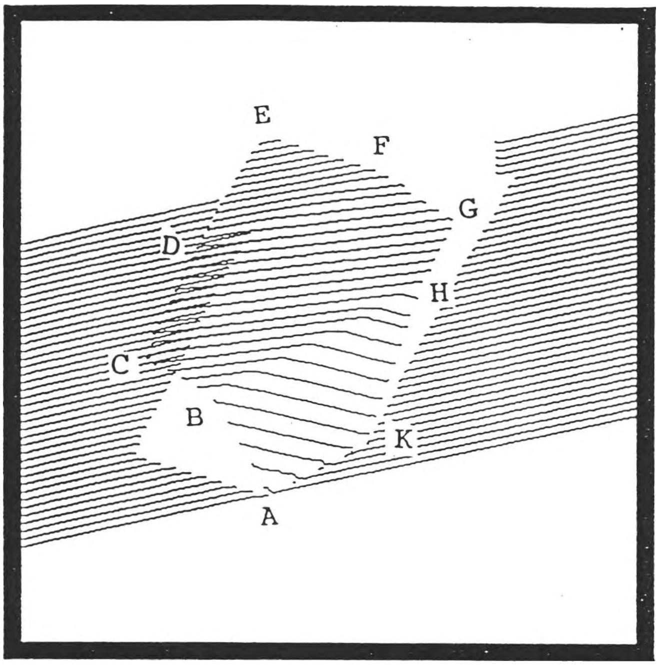


Fig. (7.36) Light-stripe image of the object Placed on base.

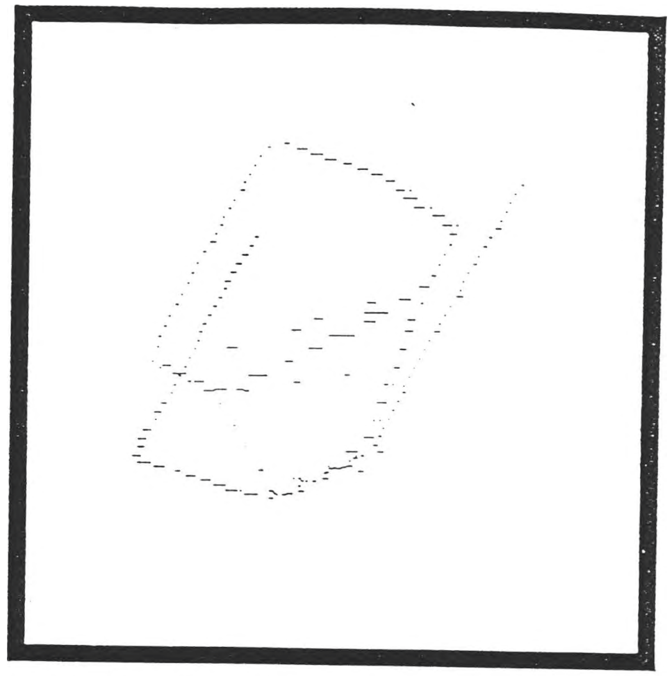


Fig. (7.37) Extracted edges AB, (BC+CD), DE, HB, KH (HG+GF+FE)

NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	8 EDGES	1	0 MIN. 30 SEC.
2	B	5 EDGES	1	0 MIN. 5 SEC.
3	C	6 EDGES	2	0 MIN. 13 SEC.
4	D	7 EDGES	1	0 MIN. 16 SEC.
5	E	12 EDGES	2	0 MIN. 39 SEC.
6	F	11 EDGES	6	1 MIN. 39 SEC.
7	G	8 EDGES	0	0 MIN. 17 SEC.
8	H	4 EDGES	1	0 MIN. 5 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				3 MIN. 44 SEC.
***** MODEL NUMBER			6	IS THE OBJECT *****

Table (7.13) Recognition results for experiment No. 15 (Models include edges for first category of tests

EXPERIMENT: NO.16

OBJECT: G

OBJECT PLACED ON THE BASE

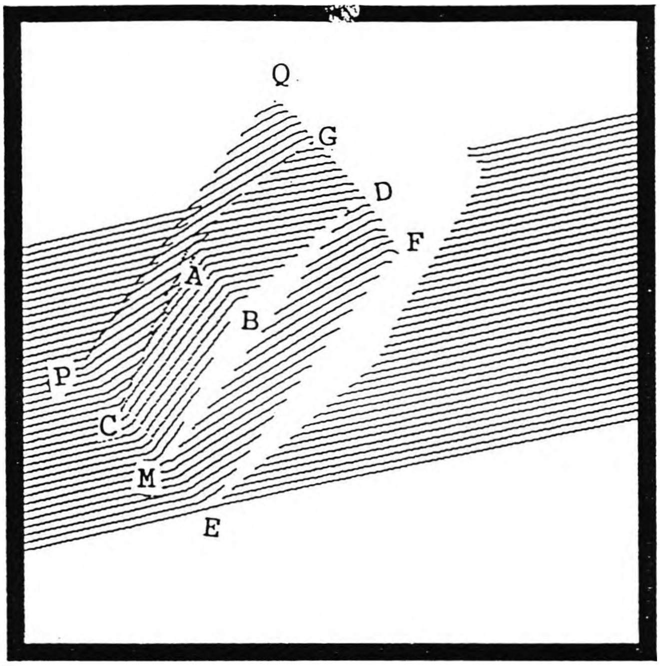


Fig. (7.38) Light-stripe image of the object. Placed on the base. (AJFN as defined in Fig.(7.7)).

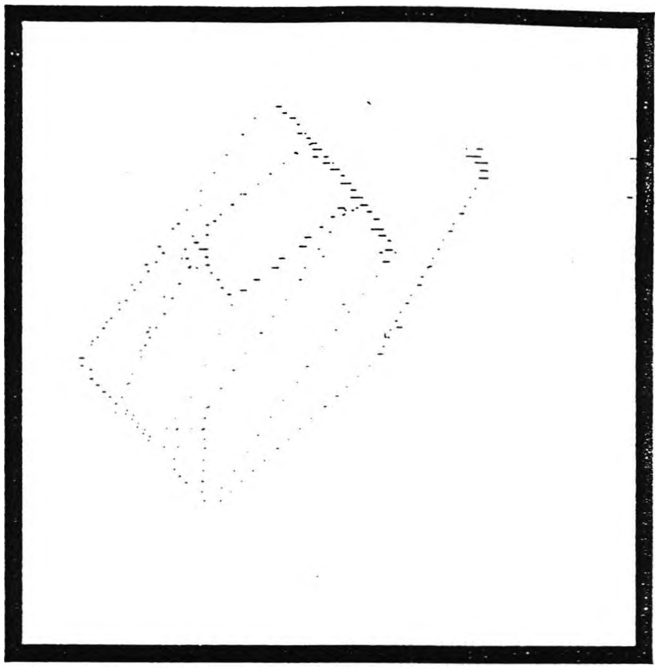


Fig. (7.39) Extracted edges BM,AB,BD,EF,FG,MD,AG,CG, GQ,PQ

MODELS & No.EDGES		QF on its own only as in Fig.(6.38)	MODELS & No.EDGES		QF together with QG, GD,DF,QD,GF included
		SCORE			SCORE
A	8	4	A	8	4
B	5	0	B	5	0
C	6	1	C	6	1
D	7	0	D	7	0
E	12	3	E	12	3
F	11	2	F	11	2
G	8	12	G	8	20
H	4	0	H	4	0
CHOSEN MODEL		7	CHOSEN MODEL		7

Table(7.14) Results for experiment No. 16. First, only QF (as above) included in the model and recognition achieved with 12 votes. Then all possible shorter lines are also added to the model and improved results with 20 votes obtained.

EXPERIMENT: NO. 17

OBJECT: G

OBJECT PLACED ON A FACE OTHER THAN BASE

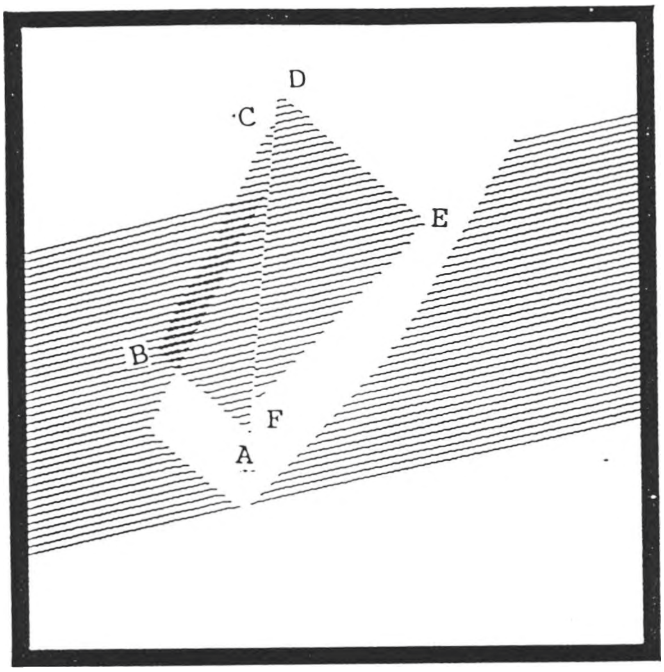


Fig.(7.40) Light-stripe image of the object. Placed on ABJ as defined in Fig.(7.7)).

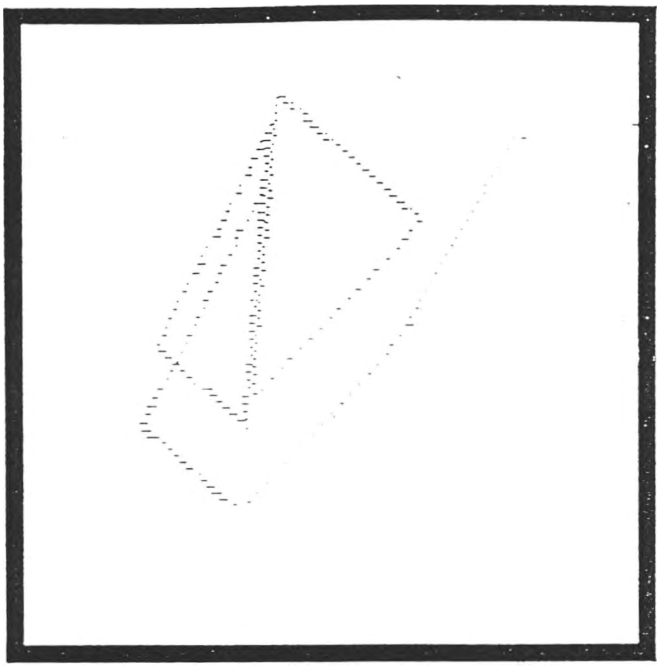


Fig.(7.41) Extracted edges AC,AB,BC,FD,FE,ED

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES		0 MIN. 25 SEC.
NAMES & NO. OF EDGES IN MODELS	HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1 A 14 EDGES	2	0 MIN. 38 SEC.
2 B 9 EDGES	3	0 MIN. 16 SEC.
3 C 12 EDGES	1	0 MIN. 32 SEC.
4 D 13 EDGES	3	0 MIN. 45 SEC.
5 E 24 EDGES	2	1 MIN. 1 SEC.
6 F 19 EDGES	2	0 MIN. 32 SEC.
7 G 16 EDGES	15	1 MIN. 18 SEC.
8 H 4 EDGES	1	0 MIN. 19 SEC.
TOTAL TIME FOR THE MATCHING PROCESS		5 MIN. 21 SEC.
***** MODEL NUMBER	7	IS THE OBJECT *****

Table (7.15) Recognition results for experiment No.17 (Model includes all edges)

EXPERIMENT: NO. 18

OBJECT: H

TOP SQURE USED IN DATABASE

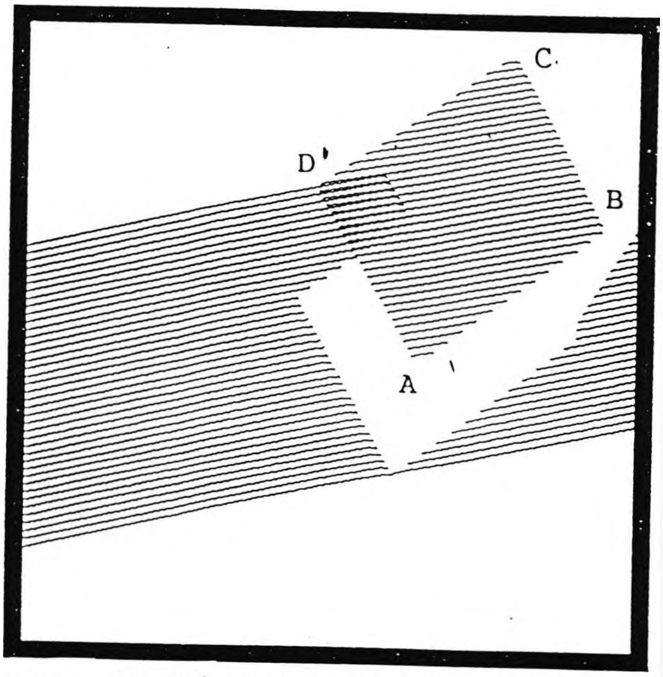


Fig. (7.42) Light-stripe image of the cube.

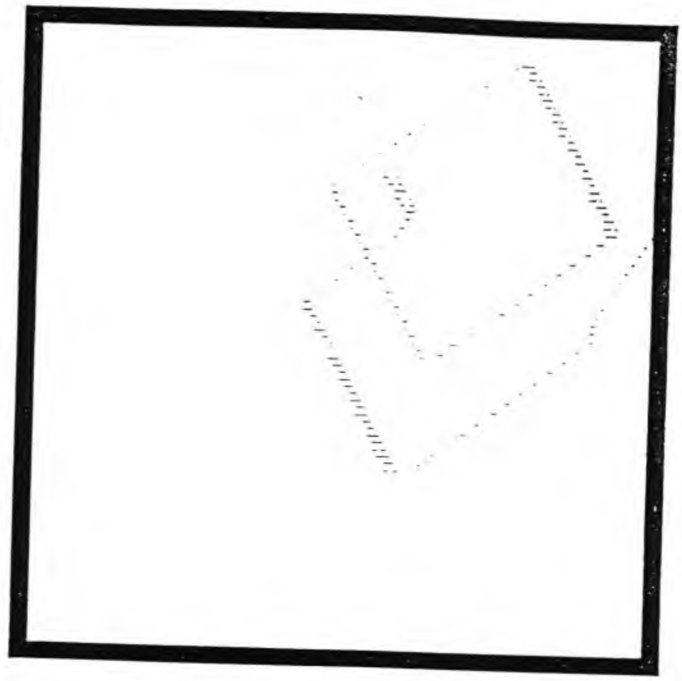


Fig. (7.43) Extracted edges AD,DC,AB,BC

NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	8 EDGES	0	0 MIN. 3 SEC.
2	B	5 EDGES	1	0 MIN. 4 SEC.
3	C	6 EDGES	2	0 MIN. 7 SEC.
4	D	7 EDGES	1	0 MIN. 9 SEC.
5	E	12 EDGES	0	0 MIN. 3 SEC.
6	F	11 EDGES	0	0 MIN. 4 SEC.
7	G	8 EDGES	0	0 MIN. 2 SEC.
8	H	4 EDGES	4	0 MIN. 12 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				0 MIN. 44 SEC.
***** MODEL NUMBER			8	IS THE OBJECT *****

Table (7.16) Recognition results for experiment No. 18. Model includes four edges of the square.

EXPERIMENT: NO. 19

OBJECT: J

OBJECT PLACED ON THE BASE

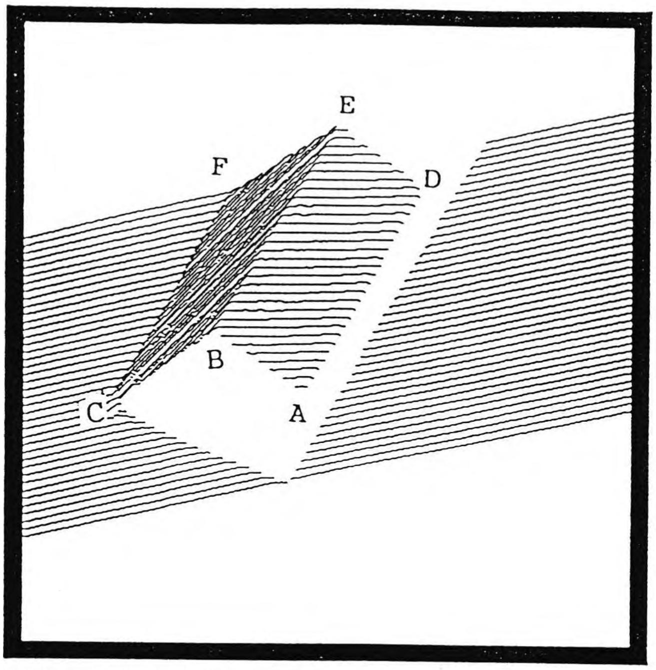


Fig. (7.44) Light-stripe image of the object Placed on base.

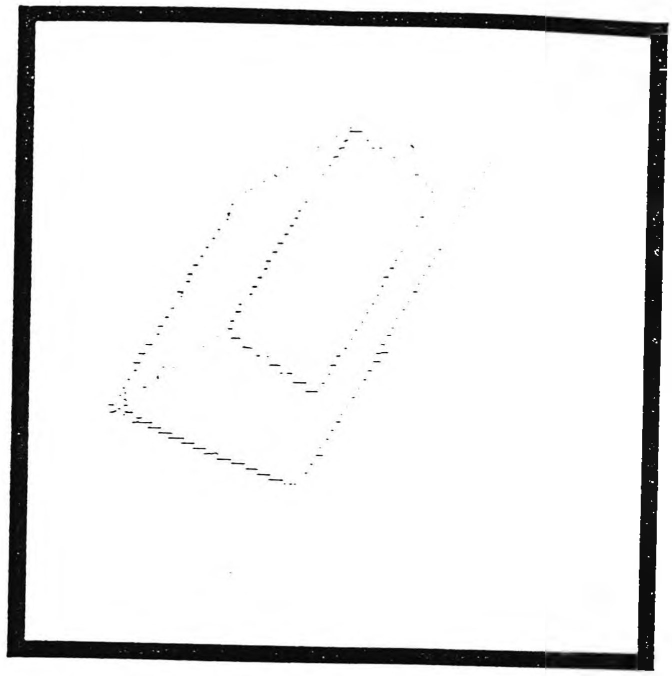


Fig. (7.45) Extracted edges AB,BC,AD,DE,BE,EF

NAMES & NO. OF EDGES IN MODELS			HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A	8 EDGES	1	0 MIN. 20 SEC.
2	B	5 EDGES	1	0 MIN. 8 SEC.
3	C	6 EDGES	4	0 MIN. 25 SEC.
4	D	7 EDGES	1	0 MIN. 15 SEC.
5	E	12 EDGES	5	1 MIN. 4 SEC.
6	F	11 EDGES	3	0 MIN. 35 SEC.
7	G	8 EDGES	2	0 MIN. 20 SEC.
8	J	6 EDGES	9	1 MIN. 10 SEC.
TOTAL TIME FOR THE MATCHING PROCESS				4 MIN. 17 SEC.
***** MODEL NUMBER			8	IS THE OBJECT *****

Table (7.17) Recognition results for experiment No. 19. Model includes edges of the base as in Fig.(7.9).

EXPERIMENT: NO.20

TWO OBJECTS PLACED ON THE BASE

OBJECTS C AND D

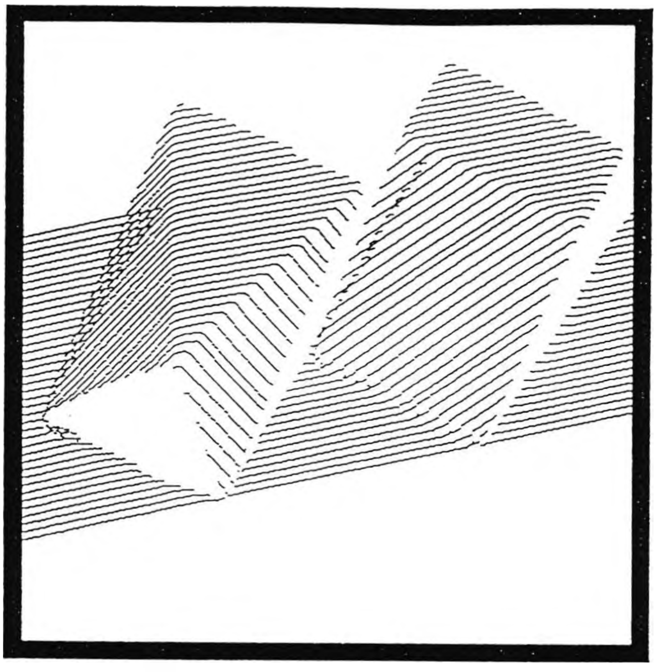


Fig. (7.46) Light-stripe image of the objects C & D placed on base.

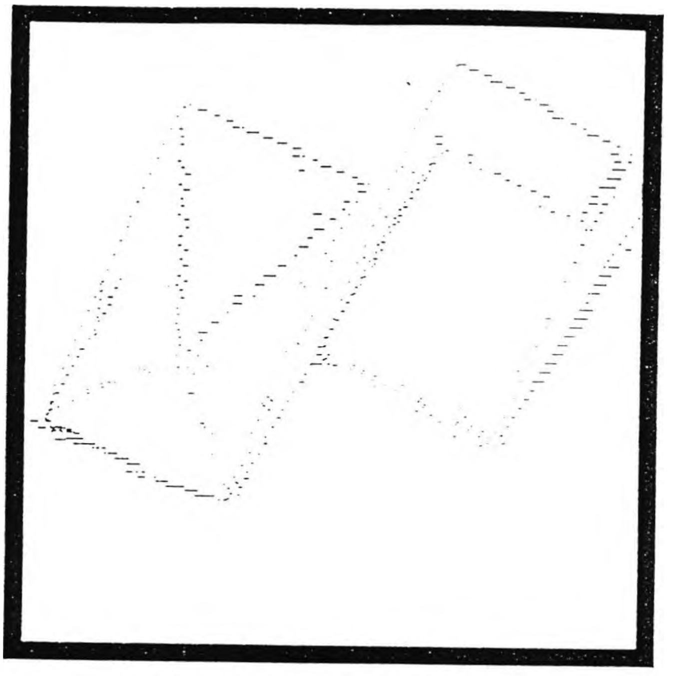


Fig. (7.47) Extracted edges from two objects.

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES		2 MIN. 4 SEC.	
NAMES & NO. OF EDGES IN MODELS	HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL	
1 A 8 EDGES	1	0 MIN. 41 SEC.	
2 B 5 EDGES	2	0 MIN. 32 SEC.	
3 C 6 EDGES	8	0 MIN. 42 SEC.	
4 D 7 EDGES	12	1 MIN. 36 SEC.	
5 E 12 EDGES	3	1 MIN. 27 SEC.	
6 F 11 EDGES	1	1 MIN. 43 SEC.	
7 G 8 EDGES	5	1 MIN. 53 SEC.	
8 H 4 EDGES	2	0 MIN. 27 SEC.	
TOTAL TIME FOR THE MATCHING PROCESS		9 MIN. 1 SEC.	
***** MODEL NUMBER	3 & 4	IS THE OBJECT *****	

Table (7.18) Recognition results for experiments No. 20 Model includes edges of the base. Objects C & D are correctly recognised.

EXPERIMENT: NO. 21

TWO OBJECTS PLACED ON THE BASE

OBJECTS: B AND C

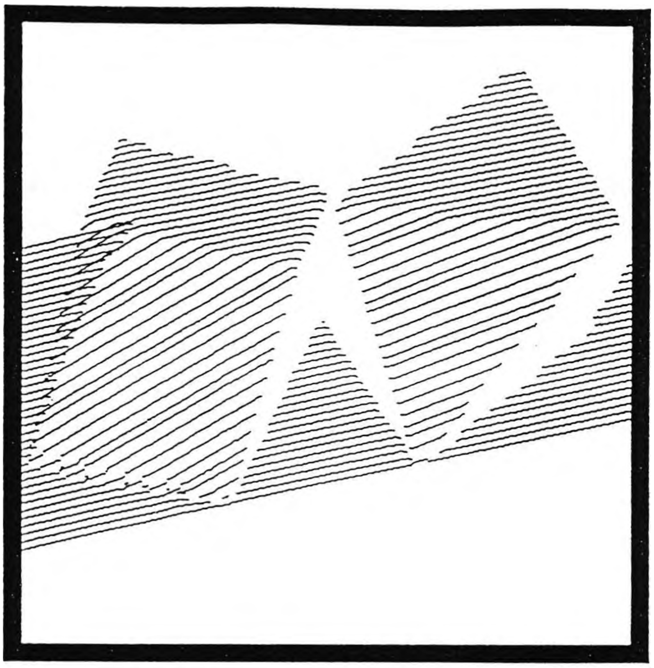


Fig. (7.48) Light-stripe image of the objects B & D Placed on base.

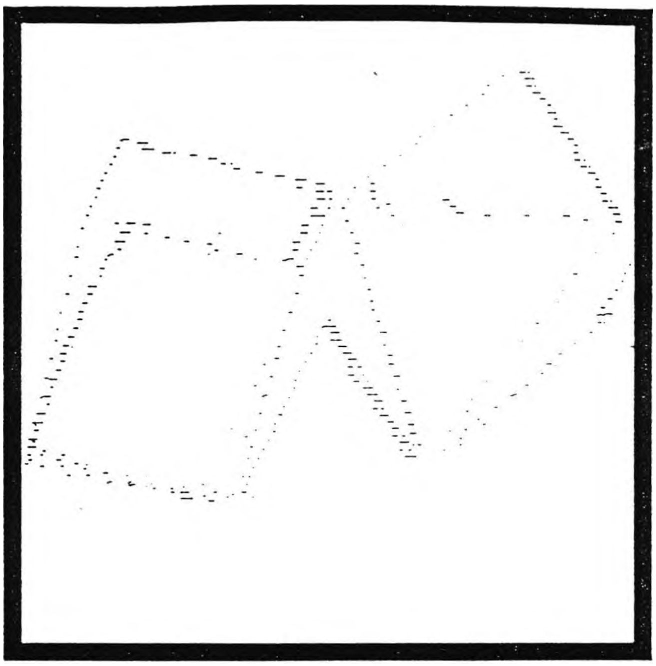


Fig. (7.49) Extracted edges from two objects.

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES			0 MIN. 49 SEC.
NAMES & NO. OF EDGES IN MODELS		HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1	A 8 EDGES	3	0 MIN. 38 SEC.
2	B 5 EDGES	6	0 MIN. 20 SEC.
3	C 6 EDGES	11	0 MIN. 44 SEC.
4	D 7 EDGES	5	0 MIN. 42 SEC.
5	E 12 EDGES	6	1 MIN. 23 SEC.
6	F 11 EDGES	2	1 MIN. 27 SEC.
7	G 8 EDGES	3	1 MIN. 22 SEC.
8	H 4 EDGES	2	0 MIN. 18 SEC.
TOTAL TIME FOR THE MATCHING PROCESS			6 MIN. 54 SEC.
***** MODEL NUMBER		3 & ?	IS THE OBJECT *****

Table (7.19) Recognition results for experiments No. 21 Model includes edges of the base. Objects C is recognised but object B is mixed by E.

EXPERIMENT: NO. 22

TWO OBJECTS PLACED ARBITRARILY

OBJECTS: B AND E

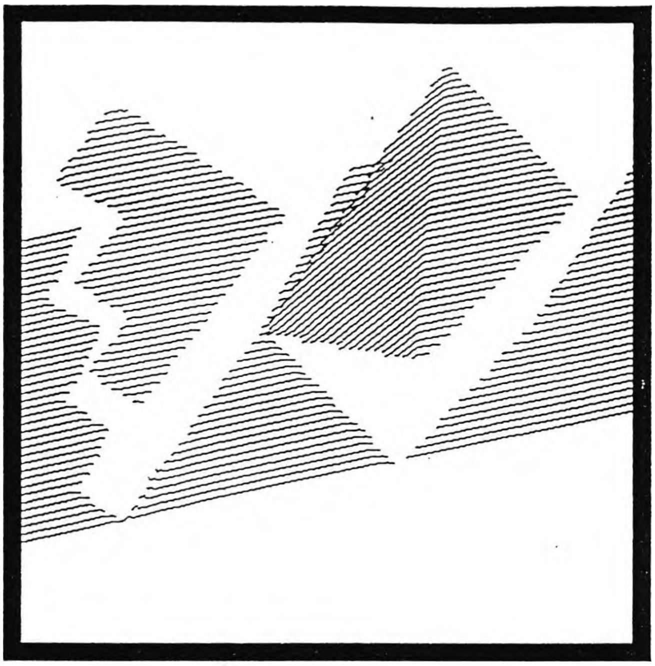


Fig.(7.50) Light-stripe image of the objects B & E placed on the arbitrary position.

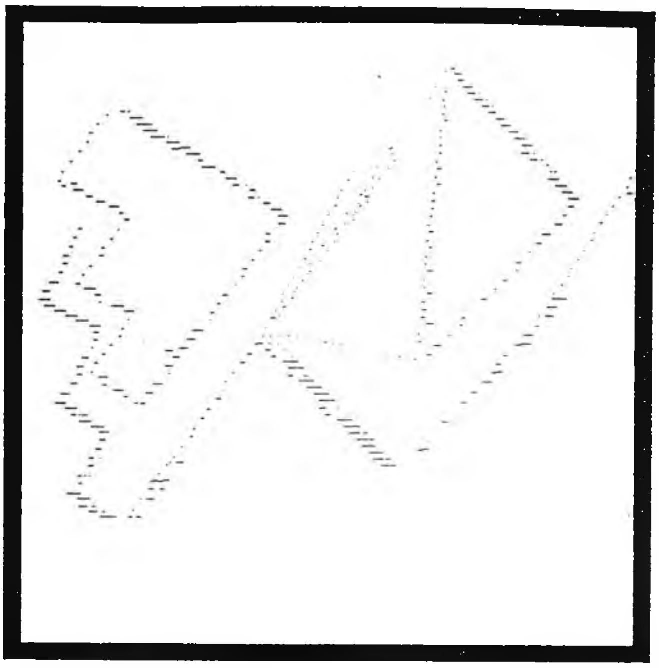


Fig.(7.51) Extracted edges from two objects.

TIME TO SET UP DATA STRINGS COMPUTE X,Y,Z EXTRACT AND REARRANGE EDGES		1 MIN. 14 SEC.
NAMES & NO. OF EDGES IN MODELS	HIGHEST SCORING MATRIX	TIME TO PROCESS EACH MODEL
1 A 14 EDGES	2	2 MIN. 39 SEC.
2 B 9 EDGES	9	1 MIN. 53 SEC.
3 C 12 EDGES	2	2 MIN. 19 SEC.
4 D 13 EDGES	8	4 MIN. 8 SEC.
5 E 24 EDGES	9	6 MIN. 15 SEC.
6 F 19 EDGES	2	4 MIN. 58 SEC.
7 G 16 EDGES	4	4 MIN. 7 SEC.
8 H 4 EDGES	1	1 MIN. 41 SEC.
TOTAL TIME FOR THE MATCHING PROCESS		28 MIN. 0 SEC.
***** MODEL NUMBER 2 & 5		IS THE OBJECT *****

Table (7.20) Recognition results for experiments No. 22 Model includes all edges. Both objects B & E are correctly recognised.

EXPERIMENT: NO.23

TWO AND THREE CONNECTED LINES

NAME OF THE MODELS & NUMBER OF EDGES	HIGHEST SCORING MATRIX		
	ORIGINAL RESULTS	TWO CONNECTED LINES	THREE CONNECTED LINES
1 A 8 EDGES	14	6	6
2 B 5 EDGES	0	0	0
3 C 6 EDGES	1	1	0
4 D 7 EDGES	0	0	0
5 E 12 EDGES	1	1	0
6 F 11 EDGES	4	3	2
7 G 8 EDGES	3	3	0
8 H 4 EDGES	0	0	0
CHOSEN MODEL	A	A	A

Table (7.21) Original results for the experiment No.1 and two and three connected lines. Results shows models other than object are heavily suppressed.

NAME OF THE MODELS & NUMBER OF EDGES	HIGHEST SCORING MATRIX		
	ORIGINAL RESULTS	TWO CONNECTED LINES	THREE CONNECTED LINES
1 A 8 EDGES	3	2	1
2 B 5 EDGES	1	1	0
3 C 6 EDGES	12	8	10
4 D 7 EDGES	1	1	0
5 E 12 EDGES	7	7	8
6 F 11 EDGES	2	1	2
7 G 8 EDGES	3	3	1
8 H 4 EDGES	2	2	2
CHOSEN MODEL	C	C	C

Table (7.22) Original results for experiment No. 6 and two & three lines. Three lines doing slightly better than two.

NAME OF THE MODELS & NUMBER OF EDGES	HIGHEST SCORING MATRIX		
	ORIGINAL RESULTS	TWO CONNECTED LINES ONLY	THREE CONNECTED LINES ONLY
1 A 8 EDGES	2	2	0
2 B 5 EDGES	0	0	0
3 C 6 EDGES	6	6	8
4 D 7 EDGES	0	0	0
5 E 12 EDGES	60	21	20
6 F 11 EDGES	2	2	2
7 G 8 EDGES	1	1	1
8 H 4 EDGES	0	0	0
CHOSEN MODEL	E	E	E

Table (7.23) Original results for experiment No.12 and two & three lines. Two lines doing almost the same as three lines.

EXPERIMENT: NO. 24

PRUNING OR MATCHING REMOVED

MODELS &No. OF EDGES		HIGHEST SCORING MATRIX AND THE TIME								
		ORIGINAL RESULT			PRUNING REMOVED			MATCHING REMOVED		
		SCORE	MIN	SEC	SCORE	MIN	SEC	SCORE	MIN	SEC
A	8	14	0	32	14	6	54	14	0	38
B	5	0	0	4	0	7	42	0	0	4
C	6	1	0	42	1	3	6	3	0	42
D	7	0	0	7	1	5	19	0	0	7
E	12	1	0	35	2	14	7	6	0	37
F	11	4	1	16	4	14	7	999	1	43
G	8	3	0	37	1	5	24	6	0	38
H	4	0	0	2	0	1	1	0	0	2
CHOSEN MODEL		1			1			--		

Table (7.24) Original results for experiment No.1 and effects of removing pruning and matching. For the pruning, running time considerably higher but number of votes the same. For the matching, running time slightly higher and object F exceeded 300 TTM matrices.

MODELS &No. OF EDGES		HIGHEST SCORING MATRIX AND THE TIME								
		ORIGINAL RESULT			PRUNING REMOVED			MATCHING REMOVED		
		SCORE	MIN	SEC	SCORE	MIN	SEC	SCORE	MIN	SEC
A	14	11	1	18	12	19	56	999	1	45
B	9	1	0	17	1	7	18	1	0	17
C	12	3	0	43	3	12	38	3	0	50
D	13	1	0	38	2	18	24	1	0	41
E	24	2	2	13	5	31	10	999	2	56
F	19	4	2	33	5	40	57	999	3	31
G	16	1	1	7	2	25	28	3	1	15
H	4	1	0	27	1	1	20	1	0	28
CHOSEN MODEL		1			1			--		

Table (7.25) Original results for experiment No.2 (all edges included) and effects of removing pruning and matching. For the pruning, running time considerably higher but number of votes the same. For the matching, running time slightly higher and objects A, E and F exceeded 300 TTM matrices.

CHAPTER EIGHT

CONCLUSIONS

In this section a brief overview of the work is given and some of the research aspects are considered. Then general conclusions are drawn and the sufficiency of the algorithms is discussed. Finally, some comparisons are made with related work and a number of plausible measures for improvement are proposed.

8.1 Summary

The initial purpose of the research described in this thesis was to concentrate on the recognition of three dimensional planar objects. For that, acquiring direct three dimensional information from the scene was found to be the best way to serve the purpose. This can be achieved by a stereoscopic method using two cameras viewing the same scene with an angle between them. Alternatively one camera, together with a specific pattern of light can be employed. Using two cameras involves many problems such as finding corresponding pixels between the two images. For this reason it was decided to use a line in the form of a stripe of light as a means to extract 3-D information from the scene. Using a line has the advantage that any break or sharp discontinuity can indicate the presence of an edge in the object.

To evaluate this method a computer vision system has been designed and implemented. In it a scanning table together with a controlled stripe of light generated from a projector is used. The stripe is arranged to be projected perpendicular to the surface of the table. The camera is positioned at a suitable angle to the surface so that images of the stripe crossing the object can be taken. First, an object is placed on the table and its surface is moved forward and scanning is performed. On each scan an image of the scene is taken and necessary processing is conducted.

A procedure has been devised so that 3-D coordinates can be computed. In it the X coordinate is directly determined from the position of the table. The Z coordinate is computed from the shift of the stripe on the image as the height of the surface changes. To establish the Z and the Y coordinates, images are divided into separate regions. Then according to the location of the points their coordinates are calculated.

Image acquisition proves to be an important part of the early stages of processing. Any inadequacy of the initial images can create great problems with image analysis and reliable data extraction. The first serious difficulty is found in choosing the right thresholding level so that the stripe can be clearly differentiated from the rest of the image. This is tackled by creating a constant lighting environment and uniform illumination. Next, from each image the middle parts of the stripes are extracted and related parts are found and joined together. This is accomplished with necessary noise removal so that reliable data can be acquired. Then from each image potential edge points are extracted and grouped.

Next, the edge points obtained from consecutive images are linked in data strings and their values are converted to three dimensions. Then the edges within each of the data strings are extracted and coordinates of two ends of the lines are found. During the experiments difficulties are found in extracting edges which lie along the stripe, edges which are short in size or edges which are otherwise difficult to scan. Setting parameters for various parts of the system proved to be an important part of the procedure. At the earlier stages of the development trial and error was used to set parameters. But, when recognition results became readily available, empirical data together with sample objects was used to assess parameter values. Before edges are processed further their size, angle and their position within the object and models are examined. This is found to be very effective in reducing unnecessary processing.

The recognition algorithm is based on the translation and rotation of edges to specific locations in 3-D space. Matrices computed and accumulated from these operations are used to recognise objects. The recognition algorithm is applied to objects with different shapes and varying complexity. First edges selected for operations are taken to their respective positions and matched. In matching, a relatively large margin of error is allowed so that the shortcomings of edge extraction or the imperfections of the equipment can be accommodated. Successful matches can reduce processing and modestly improve running time.

Matrices obtained from successful matches are calculated and stored. In order to compare matrices two separate parameters are used. Each matrix is compared with all the previous matrices and clustered with similar ones. Each matrix is considered as one vote and at the end, the model with the most votes is selected as the most similar to the object in the scene. Recognition is also performed with two and three connected lines considered together and also with two objects at a time.

8.2 <u>Some Aspects of the Research</u>

One of the by products of the present research work consists of the direct extraction of 3-D information without obtaining camera parameters or calculating perspective transformation matrices. This procedure offers a straightforward approach to the problem of calibration. In the case of the construction of the edges, nearest points from consecutive images are used to link related edge points. This is an improvement on using the direction or angle between points, which are sensitive to local fluctuation.

The recognition process has a number of similar features to The Hough transform as described in section 4.10 (Chapter four). Among them are insensitivity to noise, and normally expensive running time using accumulator cells (in case of Hough transform) or counting matrices as a voting scheme (in the recognition process). The importance of the Hough transform originates from the fact that it is capable of acquiring global information regardless of the local problems. As an example points belonging to a line can be collected, bypassing the noise. As a result lines in the images can be detected. But it has its own limitations when it comes to deciding the size of accumulator cells or distinguishing close lines in the image.

The same applies to the recognition process where, by transferring edges to a specific position in space the necessary information for recognition of an object can be extracted. In this context one main contribution of the present work consists of a critical evaluation of various aspects of the process. These consist of:

1) Investigating situations and circumstances surrounding the selection of various parameters and their interrelations, so that the process can be performed effectively.
2) Examining the behaviour of the algorithm on objects with varying shapes and complexities, and in different positions and orientations, and determining the necessary conditions to accomplish recognition.
3) Exploring possible means to run the process more efficiently.

8.3 General Conclusions and Findings

Parameters chosen for the system must be carefully and appropriately selected, so that matrices and matches can be distinguished. If parameters are chosen to be too large then the TTM matrices can be confused and create excessive votes. If parameters are selected to be too small then some of the TTM matrices can be left unaccounted for. These effects can interfere with the recognition process and result in selecting an incorrect model. A similar process can affect the matching procedure and alter the results. The value of a parameter initially set for a matrix can affect the subsequent selection of matching parameters and vice versa. For a given set of parameters not every object can be recognised. In other words only objects whose shapes and sizes comply with the parameters set for the system can be effectively dealt with.

In order to achieve recognition the following condition must be fulfilled.

> The overall pattern of edges extracted from the scene must be unique and only one model must contain the same pattern.

This implies that if more than one model contains the same pattern then the recognition would be undecided. However, in this case, the incorrect model would not be selected either.

In order to reduce processing time the angle, the length, the shortest and the longest distance between lines are compared so that unnecessary operations can be prevented. In addition by introducing matching, the number of necessary matrices is considerably reduced, modestly improving processing time. Generally, the following factors contribute most to the overall processing time:

1) The number of edges involved in the process (i.e. The greater the number of edges the longer the time).
2) The effectiveness of pruning and preventing unnecessary operations.

For the first case it is found that a modest increase in the number of edges can considerably affect processing time. For the second case difficulties can arise due to similarity in the overall pattern of edges present in the object and in the model (such as objects B and D). Difficulty can also arise due to specific patterns of edges which are hard to prune and are also repeated in the object (i.e. such as object E). After pruning has taken place, matching can modestly improve running time.

Using three connected lines can better suppress incorrect models because undesirable processing can be better eliminated and also fewer interpretations found.

8.4 Capabilities and Limitations

The importance of the method lies in the fact that:

1) It is noise independent and edges other than those belonging to the object cannot interfere in the process.
2) It can recognise objects despite similarities of size.
3) Any number of models can be involved.
4) Models are independent of view and can be easily

replaced if necessary.

The first point implies that lines which do not belong to the object but are present in the scene cannot influence the outcome of the results. The second point is particularly applicable when utilising relations between features (i.e. such as edges) is the best possible option to discriminate objects. The third point indicates that the number of models does not need to be restricted. For that, any number of models can be placed in the database and participate in the process. Models are made of a set of edges with each edge represented by coordinates of two ends of the line. In the fourth case this implies that no specific arrangements are required to construct the models. Also because models are view independent recognition of objects can be performed from any perspective. In addition it is shown that, if necessary conditions are fulfilled, then objects with roughly the same shapes can also be recognised.

However, in practice there are certain constraints that restrict these qualities. As an example, in discriminating objects of similar shape appropriate parameters must be set for the system. Also the camera view can only capture a limited number of edges at a time. If extracted edges are found to be the same in more than one model then recognition will be undecided. In addition, objects with a large number of edges can considerably extend the running time. This can affect the process if a great number of models with many edges are involved. Limitations can also arise from difficulties in extracting edges accurately. This can result in the loss of votes and consequently hamper the recognition process.

Having considered these situations the method offers a powerful option to identify objects with plane surfaces.

8.5 Some Comparisons with Other Works

In the recognition process, 3-D edges of the models are directly matched with that of the 3-D object and no transition to 2-D is required. This is an advantage to (Ballard 1983),(Silberberg 1984) or (Connolly 1987) where the 3-D features are matched with that of the 2-D images. In terms of the number of models the process is also advantageous to (Cavrila 1992) where projected 2-

D models are matched with that of the image. So recognition is achieved at the expense of having many models for different views of the projections.

For most cases, edges on their own can produce enough independent pieces of evidence to achieve recognition. While surfaces can also yield an important source of information, they may not be as effective as edges in terms of delivering votes. This is because normally the number of extracted surfaces are considerably less than the edges. Nevertheless, surfaces are indirectly utilised by (Dhome 1987) in the form of the angle between the surfaces or by (Nagata 1988) and (Krishnapuram 1989) in the shape of a needle map and extended Gaussian image (EGI). These indicate that while edges are advantageous, surface information can also be used especially when a curve or a stack of objects is involved.

In terms of generality, the process offers a complete recognition as long as a unique combination of lines is extracted and parameters properly selected. However when it comes to scenes with mixed objects or sculptured objects several view dependent models are required and recognition is not necessarily complete. An example of this is (Stein 1991) where all possible information which can be extracted from the scene is used (such as whatever edge extraction algorithms are available in the state of the art). Also (Ray 1991) can be mentioned where relations between surfaces and edges are constructed for recognition purposes. This demonstrates that while the process described in this thesis is capable of using view independent models and avoids many complex processing operations and generally is complete, it is restricted to polyhedral objects.

In comparison to methods where direct image information is utilised, examples such as (Lowe 1987), or (Caponetti 1992) can be considered . For these, camera calibration is not required and compared to the system described in thesis, the procedure is simplified. But these generally suffer from sensitivity to sensor noise, or large searching space. Nevertheless they are particularly advantageous when 3-D information is hard to obtain or specific clues such as parallel lines or relations such as affine transformation can be utilised.

8.6 Suggestions for Improvements and Further Work

There are many potential applications for vision driven automated systems. One important area concerns identifying 3-D objects. For example, it may be necessary to mark particular objects or determine their position and orientation relative to a known co-ordinate system. However, it is essential to have real time processing in a working environment.

At the early stages of processing emphasis is given to the extraction of reliable data in order to construct edges of the objects. In it, significant time is spent to scan objects, move the table, grab and process images. Here, special hardware may be employed to process images and significantly reduce processing time. Alternatively, a series of colour coded or binary coded stripes of lines can be projected on the scene (Chapter 3 section 5). Then by taking just one image each of the projected lines can be identified and processed.

In the method that is used several models are compared with the object in the scene. The processing related to each model is completely independent from the others. In addition, operations involved with each pair of lines, are themselves divided into four independent parts (section 6.10, Chapter 6). These can be executed with algorithmic parallelism where the computational task for different parts is split and then implemented by separate processors. The potential for improvement also exists to divide the process into stages and in each stage analyse the results. If at each stage evidence shows that some of the models cannot be selected then processing related to those models can be truncated.

Since the recognition algorithm is based on straight edges, it is not possible to directly apply it on curved edges or objects with curved surfaces. Nagata and Zha (Nagata 1988) described a method where surface normals computed for each picture cell are mapped and clustered into the parameter space. If a suitable polygon is fitted to the curves then it should be possible to extend the recognition algorithm by using its lines instead. In the case of curved surfaces it should be possible to project a grid pattern and use its lines in the process.

REFERENCES

Agin 1976.
Gerald J. Agin and Thomas O. Binford, "Computer Description of Curved Objects," IEEE Transactions on Computers, April 1976.

Altschule 1979.
Martin D. Altschule, Bruce R.Altschule and J.Taboada, "Measuring Surface Space-coded By a Laser-projected dot matrix," SPIE, vol. 182. pp. 187-191, 1979.

Altschule 1981.
Martin D. Altschule, Bruce R.Altschule and J.Taboada, "Laser Electro-optic System for Rapid Three-dimensional Topographic Mapping of Surfaces," Optical Engineering, vol.20, no.6, pp. 953-961, Nov/Dec 1981.

Altschule 1982.
J.L. Posdamer and M. D. Altschule, "Surface Measurement by Space-encoded Projected Beam Systems," Computer Graphics and Image Processing, no. 18, pp. 1-17, 1982.

Angell 1981.
Ian O. Angell,(Ed), "A Practical Introduction to Computer Graphics," Macmillan, 1981.

Ayache 1991.
Nicholas Ayache, (Ed), "Artificial Vision for Mobile Robots," MIT Press, 1991.

Ballard 1981.
D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," Pattern Recogn., vol. 13, pp. 111-122.

Ballard 1982.
Dana H. Ballard and Christopher M. Brown, "Computer Vision," Prentice-Hall, New Jersey, 1982.

Ballard 1983.
Dana H. Ballard and Daniel Sabbah, "Viewer Independent Shape Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, no. 6, Nov. 1983.

Bastuscheck 1989.
C. Marc Bastuschech, "Techniques for Real-time Generation of Range images," Proc. IEEE Computer Society Conference on computer Vison and Pattern Recognition, San Diego, CA, USA, pp. 262-268, June, 1989.

Besl 1985.
Paul J. Besl and Ramesh C. Jain, "Three Dimensional Object Recognition," Computer Surveys, vol. 17, no. 1, March 1985.

Besl 1986.
P. J. Besl and R.C. Jain, "Invariant Surface Characteristics for 3D Object Recognition in Range Images," Comput. Vision, Graphics & Image processing, vol. 33, no. 1, pp. 33-79, 1986.

Bhanu 1987.
Bir Bhanu and Chih-Cheng Ho, "CAD Based 3D Object Representation for Robat Vision," pp. 19-35, Computer, August, 1987.

Boyer 1987.
K. L. Boyer and A. C. Kak, "Color-Encoded Structured Light for Rapid Active Ranging," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 1, Jan. 1987.

Boyle 1988.
R. D. Boyle, R. C. Thomas, (Ed), "Computer Vision," Blackwell, 1988.

Boyter 1986.
B. A. Boyter and J.K. Aggarwal, "Recognition of Polyhedra from Range Data," IEEE Expert, PP. 47-59, Spring, 1986.

Brady 1989.
J. P. Brady, N. Nandhakumar and J.K. Aggarwal, "Recent Progress in Object Recognition from Range Data," Image and Vision Computing, vol.7, Nov. 1989.

Brooks 1983.
Rodney A. Brooks, "Model Based three dimensional Interpretations of Two Dimensional images," IEEE transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, no. 2, March 1983.

Caponetti 1992.
L. Caponetti et al., "3D Object Recognition Based on a View-point-analysis," Image and Vision Computing, vol. 10, no. 8, Oct. 1992.

Castleman 1979.
Kenneth R. Castleman, (Ed), "Digital Image Processing," Prentice Hall, 1979.

Chen 1991.
William Chen and Bernard C. Jiang, "3-D Camera Calibration Using Vanishing Point concept," Pattern Recognition, vol.24, no.1, pp. 57-67, 1991.

Chin 1986.
R.T. Chin and C. R. Dyer, "Model Based Recognition in Robot Vision," Computing Surveys, vol. 18, no. 1, March 1986.

Chiou 1992.
Rong_Nan Chiou, "Polyhedron Recognition Using Three-View Analysis," Pattern Recognition, vol. 25, no. 1, pp. 1-16, 1992

Chow 1972.
Chow,C.K. and Kaneko,T, "Automatic Boundary Detection of the Left Ventricle from Cineangiograms," Comp. and Biomed. Res., vol. 5, pp. 388-410, 1972.

Clowes 1971.
M. B. Clowes, " On Seeing Things," Artificial Intelligence 2(1), pp. 79 - 116, 1971.

Connolly 1987.
C. I. Connolly et al., "Matching From 3-D Range Models into 2-D Intensity Scenes," Proc. ICCV, London, U.K., 1987.

Davies 1986.
E.R. Davies, "Reduced Paarameter Space for Polygon Detection Using the Generalised Hough Transform," Proc. 8th Int. Conf. on Pattern Recog., Paris (27-31 Oct.), PP. 495-497, 1986.

Davies 1989.
E.R. Davies, "Minimising the Search Speace for Polygon Detection Using the Generalised Hough Transform," Pattern Recogn. Lett. 9, pp. 181-192, 1989.

Davies 1990.
E.R. Davies, "Machine Vision: Theory, Algorithms, Practicalities," Academic Press, 1990.

Dhome 1987.
M. Dhome and T. Kasvand, "Polyhedra Recognition by Hypothesis Accumulation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 3, May 1987.

Du 1992.
L. Du, G.D.Sullivan and K.D.Baker, "Modelling Data Complexity for Model-based Vision," British Machine Vision Conference (BMVC), PP. 570-579, Leeds, Sep. 1992.

Duda 1972.
Duda, R.O., Hart, P.E., " Use of Hough Transformation to Detect Lines and Curves in Pictures," Commm. Association for Computing Machinery, 15, no. 1, pp. 11-15, 1972.

Duda 1973.
Duda,R.O., Hart, P.E. (Ed), "Pattern Classification and Scene Analysis," Wiley, New York, 1973.

Fan (1989)
Ting-Jun Fan et al. "Recognizing 3-D Object Using Surface Descriptions," IEEE Transactions on Patter Analysis and Machine Intelligence, vol. 11, no. 11, Nov. 1989.

Faugeras 1982.
Faugeras,O.D. et al. "Toward a Flexible Vision System," 12th International Symposium on Industrial Robots, 9-11 June, Paris, France, 1982.

Faugeras 1986.
O. D. Faugeras and M. Hebert, "The Representation , recognition, and Locating 3-D Objects," Int. J. Robotics Res., 5(3).: 27-52, 1986.

Fisher 1989.
Robert B. Fisher, " From Surfaces to Objects," JohnWiley and sons, 1989.

Flynn 1991.
Patrick J. Flynn and Anil K. Jain "BONSAI: 3-D Object Recognition Using Constrained search," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 10, Oct. 1991.

Fox 1964.
L. Fox, (Ed), "An Introduction to Numerical Linear Algebra," Clarendon press, 1964.

Gardiner 1965.
M. Gardiner, "The Superellipse: A Curve that Lies between the Ellipse and the Rectangle," Sci. Amer., vol. 213, pp. 222-234, 1965.

Gavrila 1992.
D.M. Gavrila and F.C.A. Groen, "3D Object Recognition from 2D images Using Geometric Hashing," Pattern Recognition Letters 13, pp. 263-278, 1992.

Guzman 1968.
A. Guzman, "Decomposition of a Visual Scene into 3-D Bodies," AFIPS, Fall Joint Computer Conference, 33, (1), pp. 291-304,1968.

Godber 1992.
S.X. Godber, M.Robinson, P.Evans, "Steroscopic Vision Using Line-Scan Sensors," XVII International Congress for Photogrammetry and Remote Sensing, Washington D.C., pp. 618-627, 1992.

Gonzalez 1982.
Rafael C. Gonzalez and Reza Safabakhsh, "Computer Vision Techniques for Industrial Applications and Robot Control," IEEE Computer, Dec. 1982.

Gonzalez 1987.
Rafael C. Gonzalez and Paul Wintz, (Ed) "Digital Image Processing," second edition, Addison Wesly , 1987.

Griffine 1992.
Paul M. Griffine et al., "Generation of Uniquely Encoded Light Patterns for Range Data Acquisition," Pattern Recognition, vol.25, no. 6, pp. 609-616, 1992.

Grimson 1984.
W. E. L. Grimson and Lozano-Perez, "Model Based Recognition and localization from Sparse Range or Tactile Data," International Journal Robotics Research, vol. 3, Part 3, pp. 3-35, 1984.

Grimson 1985.
W. Eric L. Grimson and Tomas Lozano-Perez, "Search and Sensing Strategies for Recognition and Localization of Two and Three Dimensional Objects," Proc. 3rd, Int Symp on Robotics, Res(1985)

Grimson 1990.
W. E. Grimson, "Object Recognition by Computer," MIT press, 1990.

Grimson 1992.
W. E. Grimson, Daniel P. Huttenlocher and David W. Jacobs. "A Study of Affine Matching with Bounded Sensor Error," Second European Conference on Computer vision (ECCV), Santa Margherita, Ligure, Itally, pp. 291-306, May 1992.

Hall 1982.
Ernest L. Hall,James B.K.Tio,Charles A. McPherson,C.S.Draper and Firooz A.Sadjadi, "Measuring Curved Surfaces for Robot Vision," IEEE Computer, pp. 42-54, Dec. 1982.

Han 1992.
Min-Hong Han and Sangyong Rhee, "Camera Calibration for Three Dimensional Measurment," Pattern Recognition, vol.25 no.2, pp. 155-164, 1992.

Hansen 1989.
C. Hansen and T. C. Henderson, "CAGD-Based Computer Vision," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 11, Nov. 1989.

Harrison 1990.
D.D. Harrison and M.P. Weir, "High-Speed Triangulation_based 3-D Imaging with Orthonormal Data Projections and Error Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no.4, April 1990.

Hoffman 1987a.
R.Hoffman and Anil K. Jain, "Segmentation and Classification of Range images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 5, Sep. 1987.

Hoffman 1987b.
R.Hoffman and Anil K. Jain, "An Evidence Based 3D Vision System for Range Images," Proc. ICCV, London, U.K. pp. 521-525, 1987.

Horaud 1987.
Radu Horaud, "New Methods for Matching 3-D Objects with Single Perspective Views," IEEE Transactions on Pattern Analysis and Machine Inteligence, vol. PAMI-9, no. 3, May 1987.

Hough 1962.
P. V. C. Hough, "Methods and Means for Recognizing Complex Patterns," US, Patent 3069654, (1962).

Huffman 1971.
Huffman, D.A. "Impossible Objects as Nonsense Sentences," Machine Intelligence vol. 6, pp. 295-323. Meltzer, B., Michie, D. (eds). Edinburgh University Press, Edinburgh, 1971.

Huttenlocher 1990.
Daniel P. Huttenlocher and Shimon Ullman, "Recognizing Solid Objects by alignment with an Image," Intl. J. Comp. Vision, vol. 5, pp. 195-212, 1990

Ikeuchi (1981)
K. Ikeuchi, "Recognition of 3-D Objects Using the Extended Gaussian Image," Proceedings of the International joint Conference on Artificial Intelligence, (Vancouver, Canada, Aug.) Kaufmann, Los aitos, Calif., pp. 595-600, 1981.

Ishii 1976.
Ishii,M. and Nagata,T., "Feature Extraction of Three Dimensional Object and Visual Processing in a Hand-Eye System Using Laser Tracker," Pattern Recognition, vol.8, pp. 229-237, 1976.

Kanade 1987.
Takeo Kanade, "Three-Dimensional Machine Vision, "Kluwer Academic Publisher, 1987.

Kiessling 1975.
A. Kiessling and F.Rocker, "Methods for Analyzing Three Dimensional Scenes," International Conference on Artificial Intelligence, Procs., pp. 669-673, 1975.

Kittler 1986.
J.Kitter and J.Illingworth, "Minimum Error Thresholding," Pattern Recognition, vol. 19, no. 1, pp. 41-47, 1986.

Krishnapuram 1989.
Raghu Krishnapuram and David Casasent, "Determination of Three Dimensional Object Location and Orientation from Range Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, No. 11, Nov. 1989.

Li 1992.
Rongxing Li, "Generation of Object Representations of 3-D Objects in CAD/CAM by Digital Photogrammetry," XVII International Congress for Photogrammetry and Remote Sensing, Washington D.C., PP. 714-721, 1992.

Lie 1990.
Wen-Nung Lie et al., "Model-Based Recognition and Positioning of Polyhedra Using Intensity-Guided Range Sensing and Interpretation in 3-D space," Pattern Recognition, vol.23, no.9, pp. 983_997, 1990.

Lowe 1987.
David G. Lowe, "Three Dimensional Object Recognition from Single Two Dimensional images," Artificial Intelligence, pp. 355-395, 1987.

McFarland 1984.
William D. McFarland and Robert W. Mclaren, "Problems in Three Dimensional Imaging," Society of Photo-optical Instrumentation Procs., vol. 449, pp. 148-157, 1984.

Mukherjee 1990.
J. Mukherjee and B.N. Chatterji, "From Range to Frame: Extraction of 3-D Information from 2 1/2 D Data," Pattern Recognition Letters 11, pp. 493-498, 1990.

Nagata 1988.
T.Nagata and H.B. Zha, "Determining Orientation, Location and Size of Primitive Surfaces by Modified Hough Transformation Technique," Pattern Recognition, vol.21, no.5, pp.481-491, 1988.

Naidu 1992.
D.K. Naidu and R.B. Fisher, "A Comparative Analysis of Algorithms for Determining the Peak Position of a Stripe to sub-pixel Accuracy," BMVC, Leeds, pp. 217-225, Sep. 1992.

Noble 1969.
B. Noble, "Applied linear Algebra," Prentice-Hall, Englewood Cliffs, N.J., 1969.

Oshima 1975.
M.Oshima and Y.Shirai,"Representation of Curved Object Using Three-Dimensional Information," USA_Japan Computer Conference, Tokyo, Japan, 2nd, Procs. 1975.

Oshima 1979.
M.Oshima and Y.Shirai, "A Scene Description Method Using Three-Dimensional Information," Pattern Recognition, vol. 11, pp. 9-17, 1979.

Oshima 1983.
Masaki Oshima and Yoshiaki Shirai, "Object Recognition Using Three-Dimensional Information," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, no.4, July 1983.

Pentland 1987.
Alex P. Pentland, "Recogniton by Parts," Proc. ICCV, London, U.K., 1987.

Qian 1992.
Kai Qian et al., "A 3D Object Recognition System Using Voxel Representation," Pattern Recognition Letters 13, pp. 725-733, 1992.

Ray 1991
Kumar S.Ray and Dutta Majumder, "Recognition and Positioning of Partially Occluded 3-D Objects," Pattern Recognition Letters ,12, pp. 93-108, 1991.

Roberts 1963.
L. G. Roberts, "Machine Perception of Three-Dimensional Solids," Tippet JT et al. (EDS), Optical and Electro Optical Information Processing, MIT Press, Cambridge, 1965.

Rogers 1976.
David F. Rogers and J. Alen Adams, (Ed), "Mathematical Elements for Computer Graphics," McGrow-Hill, 1976.

Rosenfeld 1976.
A. Rosenfeld, R. A. Hummel,S. W. Zucker, "Scene Labeling by Relaxation Operations," IEEE Trans. SMC-6, no.6, pp. 420- 432, 1976.

Shirai 1971.
Yoshiaki Shirai and Motoi Suwa, "Recognition of Polyhedrons with a Range Finder," International Joint Conferance on Artificial Intelligence, 2nd, London, 1971.

Shirai 1972.
Yoshiaki Shirai and Saburo Tsuji, "Extraction of the Line Drowing of 3-Dimensional Objects by Sequential Illumination from Several Directions," Pattern Recognition, vol.4, pp. 343-351, 1972.

Shirai 1987.
Yashiaki Shirai, "Three Dimensional Computer Vision," Spring-Verlag FRG, 1987.

Silberberg 1984.
Teresa M. Silberberg, Larry Davis and David Harwood, "An Interactive Hough Procedure for Three Dimensional Object Recognition," Pattern Recognition, vol. 17, no. 6, pp. 621-629, 1984.

Silberberg 1986.
Teresa M. Silberberg, Larry Davis and David Harwood, "Object Recognition Using Orientated Model Points," Computer Vision Graphics and Image Processing, 35, pp. 47-71, (1986).

Singh 1990.
R.K. Singh and R.S. Ramakrishna, "Shadows and Texture in Computer Vision," Pattern Recognition Letters, 11, pp. 133-141, Feb. 1990.

Solina 1987.
F. Solina and R. Bajcsy, "Three Dimensional Object Representation Revisited," Proc. ICCV London, UK, pp. 231-240, 1987.

Solina 1990.
F. Solina and R. Bajcsy, "Recovery of Parametric Models from Range Images: The Case for Superquadric with Global Deformations," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 2, Feb. 1990.

Stein 1991.
Fridtjof Stein and Gerard Medioni, "Structural Hashing: Efficient Three Dimensional Object Recognition," IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Hawaii, June, 1991.

Strand 1984.
T. C. Strand, "Optics for Machine Vision," SPIE, vol. 456, Optical Computing, 1984.

Sugihara 1979.
K. Sugihara, "Range data Analysis Guided by a Junction Dictionary," Artificial Intelligence, 12, no. 1, pp. 41-69, 1979.

Tsai 1986.
R.Y. Tasi, "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," Proc. IEEE Computer Soc., Conf. on Computer Vision and Pattern Recognition, Miami, FL, pp. 364-376, 1986.

Underwood 1975.
Stephen A. Underwood and Clarence L.Coates, JR., "Visual Learning from Multiple Views," IEEE Transactions on Computers, vol. C-24, no.6, June 1975.

Vemuri 1986.
B. C. Vemuri, A. Mitiche, J.K. Aggarwal, "Curvature- base representation of objects from range data," Image Vision Cmoput., vol 4, no. 2, pp. 107-114, May 1986.

Vemuri 1987.
B. C. Vemuri and J.K. Aggarwal, "Representation and Recognition of Objects from Dense Range Maps," IEEE Transactions on Circuits and Systems, vol. CAS-34, no.11, Nov. 1987.

Vuylsteke 1990.
P. Vuylsteke and A. Oosterlinck, "Range Image Acquisition with a Single Binary-Encoded Light Pattern," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12., no. 2, Feb. 1990.

Wahl 1986.
F.M. Wahl and H.P. Biland, "Decomposition of Polyhedral scenes in Hough Space," Proceedings of International Conference on Pattern Recognition, Paris, France, pp. 78-84, Oct. 1986,

Waltz 1975.
D.L. Waltz, "Understanding Line Drawings of Scenes with Shadows," The Psychology of Computer Vision, pp. 19-91, Winston, P.H. (ed.), McGraw-Hill, New York, 1975.

Wang 1987.
Y. F. Wang, A. Mitiche, J.K. Aggarwal, "Computation of Surface Orientation and Structure of Objects Using Grid Coding," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 4, April 1989.

Wang 1991.
Y. F. Wang, "Characterizing Three-Dimensional Surface Structures from Visual Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 1., Jan. 1990.

Wei 1983.
Dai wei and M. Gini, "The Use of Taper Light Beam for Object Recognition," Pugh A. (Ed.), Robot Vision, pp. 143-153, 1983.

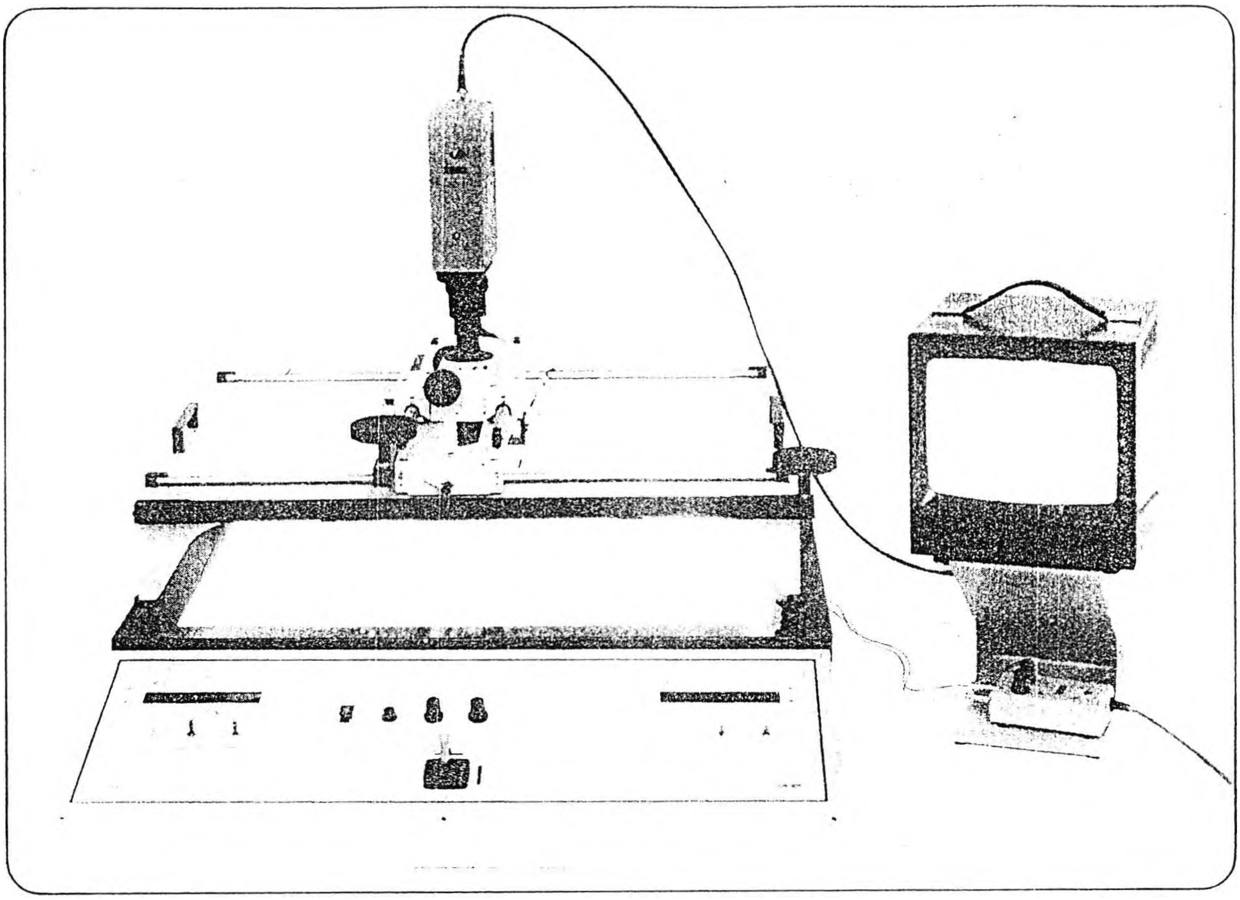
Wong 1991.
K.C. Wong et at., "Heuristically Guided Polygon Finder," British Machine Vision Conference, pp. 400-406, Glasgow, 1991.

Wong 1992.
K.C. Wong and J. Kittler, "Recognising Polyhedral Object from a Single Perspective View," BMVC, pp. 590-599, leeds, Sep. 1992.

Yang 1989.
Meng-Chien Yang and Wen-Hsiang Tsai, "Recognition of Single 3D Curved Objects Using 2D Cross Section Slice Shapes," Image and Vision Computing, vol.7, no.3, August, 1989.

Appendix A

Specificatin of the scanning table



A group of instruments and accessories for use in a wide range of application in industry, research education, photography and medicine.
Some applications are fulfilled by standard instruments, others by special adaptations of the lighting and viewing systems or by automation. MAXTASCAN can provide the answer to many problems of measurement or inspection.

FEATURES

Precise smooth ball race bearings.	Digital readout of position.
Stabilised lightweight castings	Wide range of accessories
Hand or motor drive.	

ACCURACY

Measurement can be made with the VMS range of instruments as follows:-
In any one axis to $\pm$ 15 microns.
Overall accuracy of readings when measuring diagonals would normally be within 50 microns.
Repeatability of readings $\pm$ 10 microns.

Figures quoted assume the instrument is in an inspection environment at normal room temperature.

GRATICULES LTD
MORLEY ROAD, TONBRIDGE,
ENGLAND, TN9 1RN

Telephone 0732 359061
Telex 95311 Gratic G