Geometric Feature Distributions for Shape Representation and Recognition

by

Alun C. Evans

Artificial Intelligence Vision Research Unit
University of Sheffield

Thesis submitted to the University of Sheffield
in partial fulfillment of the degree of
Doctor of Philosophy

January 1994

Geometric Feature Distributions for Shape Representation and Recognition

Summary

One of the fundamental problems in computer vision is the identification of objects from their shape. The research reported in this thesis is directed toward the development of a scheme for representing the shape of an object which allows it to be recognised both quickly and robustly across a wide range of viewing conditions.

Given a shape described by a set of primitive elements, eg. straight line segments, the proposed scheme involves using a histogram to record the distribution of geometric features, eg. angle and distance, measured between pairs of primitives. This form of shape representation has a number advantages over previously proposed schemes. Foremost among these is the fact that it is able to produce local representations of shape, based on individual line segments. Recognition based on such representation is robust to the problems arising in cluttered scenes. Representations produced by the scheme are also invariant to certain object transformations, they degrade gracefully as the shape is fragmented and are strong enough to support discrimination between dissimilar objects.

By treating the histogram recording a geometric feature distribution as a feature vector it is possible to match shapes using techniques from statistical pattern classification. This has the advantage that optimal matching accuracy can be achieved using processing which is both simple and uniform. The approach is therefore ideally suited to implementation in dedicated hardware.

A detailed analysis is undertaken of the effect on recognition of changes in the description of a shape caused by fragmentation noise, scene clutter and sensor error. It is found that the properties of both the representation and matching components of the system combine to ensure that recognition is, in theory, unaffected by fragmentation noise, while it is maintained to very high levels of scene clutter. The factors which determine the effect of sensor error on the performance of the recognition system are fully analysed.

The ability of the representational scheme to support object recognition is demonstrated in a number of different domains. The recognition of both 2D and 3D objects from a fixed viewpoint is demonstrated in conditions of severe fragmentation noise, occlusion and clutter. The scheme is then shown to extend straightforwardly to the representation of 3D shape. This is exploited to perform recognition and localisation of 3D objects from an arbitrary viewpoint, based on the matching of 3D scene and model shape descriptions. Finally, the use of the scheme within a multiple view-based approach to 3D object recognition is demonstrated.

Acknowledgements

Grateful thanks are due to my supervisors; Professor John E.W. Mayhew, for his useful support and advice, and Dr. Neil Thacker, for providing the inspiration for this work and for his patience and guidance while I strove to match his level of understanding. Thanks also to Professor John Frisby for his support and encouragement.

I would like to thank the following:

The designers of the TINA vision system for providing such a wonderful environment in which to develop ideas.

Past and present members of AIVRU for helping to create a friendly and stimulating environment in which to conduct research. Thanks especially to John Porrill, Li-Dong Cai and Stephen Hippisley-Cox for their kind help on matters mathematical.

Patrick Courtney, Gareth Ellwood and John Oxley for sharing my enjoyment of tea drinking.

David Buckley, for letting me live in his house.

Gareth Palmer, for sharing with me daily, via email, his incisive wit and wise council.

Derek Jones, Stuart Cornell and Julian Briggs for maintaining the computing environment in which this work was carried out.

Grace Crookes, for secretarial assistance.

Andy Wright and Mick Brown at British Aerospace for their kind support and encouragement in time of need, and for financial support during the three years.

Finally, I would like to say a special thank you to Anne for her unconditional support and understanding during the writing of this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1

MOTIVATION and BACKGROUND

1.1 Motivation for Study

The capacity of a robotic agent to interact intelligently with its environment depends crucially on its ability to interpret its sensory input. In particular, the ability to recognise which objects are present in the environment, and to determine their position and orientation, is central in supporting a wide range of intelligent behaviours, with numerous industrial applications. In manufacturing for instance, visual object recognition is useful in supporting the automated inspection of manufactured objects. The purpose of this inspection might be to identify objects in order to monitor the throughput of some process, or it might be applied as part of a quality control system, helping to detect manufacturing faults that are outside acceptable limits. In both cases it might also be useful to be able to locate objects so that they can be picked up by a robot and manipulated or sorted. Object recognition can also play an important part in making robots more mobile. The ability to identity and locate known objects in the world provides useful information to an autonomous vehicle engaged in some navigation and/or positioning task. Thus, vision can be seen as helping to realise the potential of automation in manufacturing by improving the versatility of robots and thereby increasing the range of tasks to which they can be applied.

This thesis addresses the problem of developing a computer vision system that is able to identify and locate arbitrary rigid objects from 2D, grey-level or monochrome images of the scene. If this system is to be of use in practical applications then it must must be able to deal robustly with the kinds of problems that are likely to occur in any real, unconstrained, environment. These may arise from factors such as the partial occlusion of one object by another, transformations in the position and/or orientation of the object relative to the viewing camera, the presence of noisy or background features in the image or from poor lighting of the scene. The effect of such problems is to cause a change in the appearance of an object, thereby increasing the difficulty of performing recognition. If the proposed system is to be robust to these factors

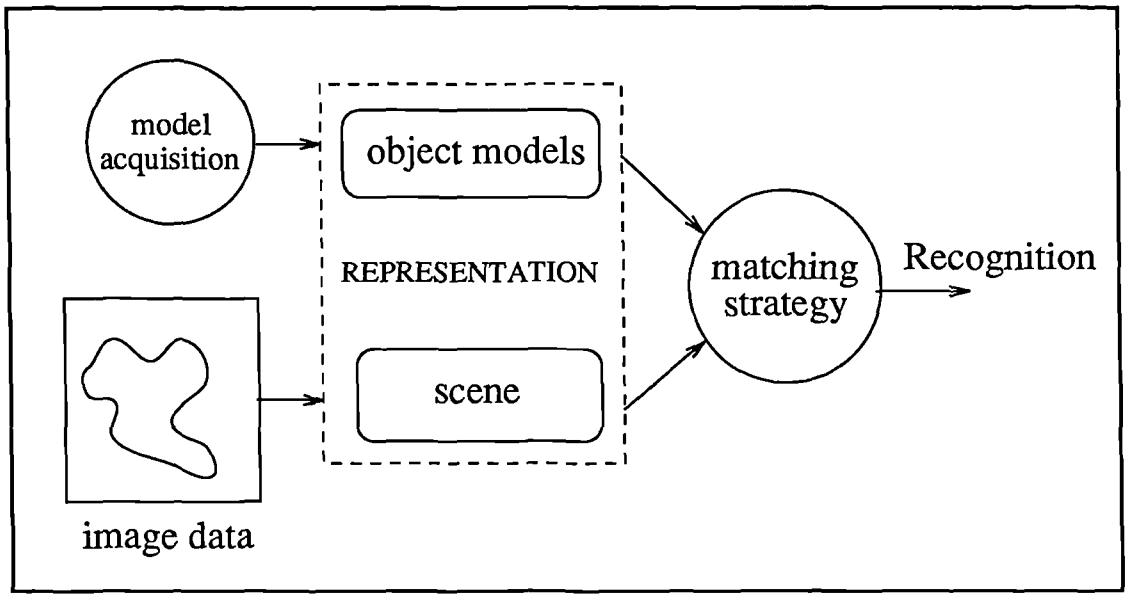

Figure 1–1: A general recognition system.

then it must be capable of associating each of the possible images of an object with a single, unique entity. There are also a number of practical considerations that must be taken into account when assessing the applicability of a system in an industrial setting. Firstly, the system must be both fast and accurate, as determined by requirements of the particular task. Secondly it should be flexible, both in the types of objects that can be recognised and in the environments in which they can be encountered. Finally, it should be capable of dealing with large numbers of objects.

The ability to make sense of ones surroundings implies some form of prior knowledge, in the form of a stored, internal representation of the objects to be recognised. This statement is not intended to constrain the nature of this representation. For example, if one were interested in asking questions such as, "Is there an object in the scene that can serve as a chair?", then the representation would be in terms of an object's functionality. However, the work presented in this thesis falls within a *model-based* approach to recognition. This proposes that representation is based upon some fixed property or characteristic of an object, or of its appearance in an image, that can be exploited in order to identify the object in a wide range of changed circumstances. Given a set of object models, the goal of recognition then becomes one of exploiting the characteristics upon which the models are based in order to interpret elements of the world in terms of elements of the set of object models. This is achieved through some form of matching algorithm. A recognition scheme therefore has two components; a representational scheme for modeling objects and a matching algorithm. This situation is summarised in figure 1–1. The practicality of any proposed recognition system depends to a large extent on the success with which the properties of both its representational and matching components combine to meet the challenges listed above. Consequently, research in the field of visual object recognition can be seen as an

investigation into the nature of object representation, the mechanisms by which they might be acquired and the processes by which they support recognition.

The work presented in this thesis is based upon the assumption that representation can be based upon the *shape* of an object. In particular, a form of representation is proposed which is based on recording the distribution of geometric features, eg. angle and distance, measured between primitive elements describing the shape of an object. This could either be an explicit description of the 3D structure of the object or it could be the 2D projected shape, as found in a single image. This form of representation has many advantages in terms of its ability to support distinctions between objects and in its robustness to changes in shape description caused by image noise or occlusion. Such representations can also be matched using techniques from statistical pattern classification. The system based on the matching of these representation is shown to be capable of recognising and locating both 2D and 3D objects in conditions of considerable image noise and clutter.

1.2 Background

The purpose of the remainder of this chapter is to provide background to the work presented in this thesis by reviewing previous approaches to shape representation and object recognition. This serves both as background for the remainder of the thesis and as a detailed exposition of the motivation behind the study of geometric feature distributions, (GFD's), as representations of shape. In order to organise the review into manageable proportions the first few sections restrict consideration to the issues involved in, and possible approaches to, 2D object recognition. The final section then extends the review to the additional problems posed by 3D object recognition. In particular, the review is divided into the following sections:

1. **2D Object Recognition**
   This section briefly reviews some of the concepts and issues involved in performing 2D object recognition.

2. **Statistical Pattern Classification**
   The statistical pattern classification approach to object recognition is presented and the importance of the role of shape representation is discussed. Several previously proposed representational schemes are reviewed. It is argued that while the statistical pattern classification approach offers many potential advantages in terms of simplicity and speed of recognition, the failure of existing shape representations to overcome the various problems involved in recognition means that this potential has not been realised. Through this discussion the desirable properties of an ideal shape encoding are established.

3. **The Role of Geometric Relationships in Recognition**
   The concept of basing recognition on measurements of the geometric relationship between local shape primitives is introduced and shown to have many advantages. Several previous approaches to recognition based upon their use are reviewed. It is claimed that, while these systems often perform impressively, the fact that recognition is based upon individual geometric relationships necessitates the use of complex procedures for implementing global constraints on matching. This is necessary in order to overcome the the inherent weakness of these local features. Previous attempts at representing and matching shape on the basis of the distribution of *multiple* geometric relationships within a shape are reviewed, and shown to possess many of the desirable properties established in the previous section.

4. **3D Object Recognition**
   Possible approaches to the problems involved in 3D object recognition are reviewed. These include the *3D approach*, (in which 3D shape descriptions of model and scene are matched directly), the *alignment approach*, (in which 3D object models are matched to 2D image data through a search of transformation space), and the *multiple view-based approach*, (in which 2D image data is matched to 2D, appearance-based object models).

1.2.1 2D Object Recognition

This section presents a brief review of the issues involved in 2D object recognition.

Problem Description

The term "*2D object recognition*" is taken here to describe the situation in which the relationship between an object and the viewing camera is fixed, such that variations in the appearance of an object arising from changes in viewpoint need not be considered. However, objects are free to undergo transformations in their position and orientation *within* the image plane. Additionally, changes in the apparent size of an object may occur if the object to camera distance is not fixed. The objects themselves may be either 2D or 3D, but are assumed to be rigid and from the same mould, ie. there is no variation in structure between objects of the same class. The aim of recognition is, minimally, to identify which known objects are present in the scene. In addition the recognition system may also be required to provide estimates of the position and orientation, or *pose*, of each object. Given certain assumptions, together with a suitably calibrated system, this information can be used to determine the world coordinates of each object.

Visual Cues

As with most previous approaches to recognition, consideration is restricted to the recognition of static objects from their projected shape, as contained in a grey-level image of the scene. This obviously ignores a number of information sources that could be usefully exploited in recognition. These include an object's colour, its texture or any characteristic motions that it may exhibit. Each of these visual cues could be expected to have considerable disambiguating power. This narrowing of focus is justified on two grounds. Firstly, the more visual cues that are considered the more complex the processes involved in recognition become. Secondly, a number of researchers have argued that for certain objects, crude encodings of their projected boundaries are sufficient for recognition, [2,8].

Shape Description

Given a grey-level image of a scene, the first stage in recognition is therefore to extract a description of the projected shape of the objects present in the scene. In cases where the image describes a complex, natural scene there are numerous advantages to producing a region-based description of shape, [106]. This involves thresholding an image intensity histogram to detect spatially continuous areas of the image that share a common intensity value. This process may be applied locally if the image histogram is not bimodal, and produces a description of shape in the form of a region map. However, the proposed representational scheme is to be based upon recording the distribution of geometric features measured between shape primitives that are both local and uniform. For this reason we limit discussion to boundary based shape descriptions. The extraction of such descriptions is based upon the fact that object contours tend to be projected in the image as intensity gradients. By detecting points in the image at which such gradients occur, and noting the location and orientation of the gradient, it is possible to obtain a description of the projected shape of an object in terms of a set of edge elements, or *edgels*. These low-level primitives may be grouped, using some form of approximation process, to form higher level primitives, such as extended line segments or conics. Marr termed this level of representation a 'primal sketch', [65].

Variation in Shape Description

The difficulty of the recognition task is determined to a large extent by the degree of variation that occurs in the description of an object's shape between model acquisition and recognition. This in turn depends on the context in which objects are to be encountered. If, as in certain industrial applications, the environment can be engineered such that objects are viewed in isolation, at a fixed position and under constant, favourable lighting then the amount of shape variation will be minimal. Under such conditions

the problem of recognition becomes quite straightforward, and can be performed using some form of template matching. However, in general such conditions cannot be assumed. The relaxation of the above constraints means that the shape extracted from an image may vary quite drastically from that used in model acquisition. The major sources of this variation are now briefly examined.

The fact that edgels are based upon relative, rather than absolute, image intensity values suggests that they should be relatively stable to changes in the lighting of a scene. However, the need to place a threshold on the size of an intensity gradient that is to be regarded as a valid edgel means that the process of edge detection becomes unstable in poor lighting conditions, leading to fragmentation of the edgel-based shape description. The presence of multiple, possibly unknown, objects in a scene can cause two types of problem. Firstly, the possibility that one object may partially obscure another means that elements of the shape description may be lost through occlusion. Secondly, the presence of additional objects in the scene means that any shape description obtained from low-level, bottom-up processing must contain elements due to more than one object. The shape description may even contain elements due to composite objects, created by several overlapping objects. A number of researchers have proposed bottom-up strategies for segmenting elements of the shape description that are most likely to belong to a meaningful object, as opposed to background clutter, eg. Ullman [103]. However, this approach does not promise a general solution. The description of an objects' projected shape extracted from an image may therefore be fragmented, perturbed and augmented by spurious elements.

1.2.2 Statistical Pattern Classification

One of the earliest approaches to object recognition was to regard it as a problem that could be tackled using techniques from statistical pattern classification. This involves describing each object using a set of features that are adjoined to form a feature vector, such that each object is represented as a point in a multidimensional *feature space*. Where recognition is based on an object's shape, possible features range from simple measures, such as area or perimeter, to the more complex shape characteristics described below. One of the strengths of the pattern classification approach is that information from different visual cues, eg. shape, colour or texture, can be easily integrated, either by combining features to form a single feature vector prior to classification, or by performing some form of data fusion.

Given an encoding of shape in the form of a feature vector, recognition can be achieved by applying some form of statistical classification rule, eg. nearest-neighbour, once the value of a distance metric between the feature vectors representing scene and model have been computed. Interest in this approach has recently been revived through work on the use of artificial neural networks, although the techniques used to perform recog-

nition are directly related to those of standard pattern classification, (see Lippmann [61] for a review).

The classification approach is based upon the assumption that similarity between shape representations can be interpreted as similarity between the objects themselves. This fact, together with the simplicity of the recognition process, means that its success is largely dependent upon the characteristics of the chosen representational scheme, and the shape features upon which it is based. Any properties that are required of the recognition system must be included as part of the representational scheme.

Desirable Properties

It will be useful at this stage to list a number of properties that should ideally be possessed by any proposed representational scheme.

i) Uniqueness

An encoding is *unique* if each distinct shape is mapped to a different point in feature space. Such an encoding is capable, theoretically at least, of supporting discrimination between all dissimilar shapes. The uniqueness of an encoding scheme is determined by the *strength* of the features upon which it is based. For example, the area or perimeter of a shape, while possibly sufficient for distinguishing between certain objects, are not strong, since it is possible for different objects to produce the same set of feature values.

ii) Robustness

The representation should be as robust as possible to changes in a shape description caused by fragmentation noise, occlusion or scene clutter. Obviously, no useful representation can be completely unaffected by such changes. However, it is reasonable to require that the change in the shape encoding be proportional to the degree of shape variation. This *graceful degradation*, [65], of the encoding should enable the system to provide a best estimate of recognition, given the uncertainty caused by the changed viewing conditions.

iii) Invariance

The representation should be invariant to changes in shape caused by certain object transformations. This both cuts down on the number of patterns that must be stored and ensures that the recognition system has the ability to generalise across a wide range of viewing situations. Ideally, a recognition system would be able to automatically identify invariant features from a set of patterns representing transformed instances of an object. However, the problem of extracting non-trivial commonalities between such patterns has proved very difficult. For this reason it is desirable that any required invariances are built into the representational scheme. Although much recent interest has centred on discovering shape features that are invariant to general projective transformations, we restrict consideration here to the Euclidean transform, ie. changes in the 2D position and orientation of an object.

These are the main, theoretical, criteria upon which any proposed shape encoding must be assessed. Further, practical, considerations include the scheme's accessibility - how easy is it to compute representations from the available image data, and its versatility - are there any restrictions on the range of objects that can be represented?

Previous Approaches

A number of shape encodings that have been widely used in the recognition literature are now reviewed, (for a complete review of such techniques see Marshall [66].) These encodings are intrinsically *global* in nature, in that they exploit characteristics which are sensitive to the presence of the complete shape. The matching of such global shape representations is commonly termed *non-correspondence* recognition.

Moment Invariants

A shape characteristic that has commonly been used for the recognition of unoccluded, rigid objects is the set of moment invariants, Hu [47]. An excellent review of moment-based techniques is presented in Prokop & Reeves [81]. Moments are typically computed from a region-based description of shape, although they can also be applied to boundary-based descriptions. Given a segmented description of the image, described by the characteristic function $g(x, y)$, where

$$g(x,y) = \begin{cases} 1, & \text{if point (x,y) is part of the object} \\ 0, & \text{otherwise} \end{cases}$$

the general moment M_{pq}, said to be of order $p+q$, is then given by

$$M_{pq} = \sum_x \sum_y x^p y^q g(x,y)$$

Thus, the zeroth order moment defines the area of a shape,

$$M_{00} = \sum_x \sum_y g(x,y)$$

while the normalised 1st order moments give its centroid,

$$C_x = \frac{M_{1,0}}{M_{0,0}} \qquad\qquad C_y = \frac{M_{0,1}}{M_{0,0}}$$

Higher order moments correspond to such characteristics as the principle axes, radii of gyration and skewness of a shape, [81]. The values of individual moments, or combinations of moments, are not sufficiently strong to support useful recognition and so are adjoined to form a feature vector. Hu demonstrated that by combining the values of individual moments, shape encodings with the desired invariance properties could

be produced. The main advantages of moment-based shape encodings are their invariance, their accessibility and their versatility, (they can be applied to arbitrary shapes, even those containing internal structure.) Their main disadvantage is their weakness; unless a large number of moments are considered, representations are unlikely to be unique. The fact that higher order moments become increasingly less intuitive is also considered a disadvantage, [81].

Fourier Transform

Given a shape described by a set of boundary points it is possible to characterise the shape using the coefficients of a Fourier transform, [111,74,60,39]. The 2-D boundary curve is first transformed into a 1-D boundary profile, (an *arc length-turning angle* graph can be used), which is then normalised to the interval $[0, 2\pi]$ to produce a rotation, translation and scale invariant representation of shape. This is then used as the basis for a Fourier expansion, given by:

$$X_n = \frac{1}{N} \sum_{s}^{N} \psi(s) e^{-j2\pi ns}$$

The coefficients of the Fourier series are combined to form a feature vector. The main advantage of Fourier methods is in their strength; in the limit, the infinite series of Fourier coefficients provide a unique representation. In practice the minimum number of terms are used to characterise the shapes to the required level of discrimination.

Log Polar Mapping

A log-polar mapping can be used to transform points in image space into points in log-polar parameter space. If the mapping is centred at some point in the image then each image point $z = x + yj$ is transformed to a point w in log-polar parameter space, such that

$$w = \ln(z) = \ln(|z|) + j\theta_z$$

The act of positioning the centre of the mapping at some fixed point on the object effectively provides invariance to object translation. The property of this mapping which makes it useful for recognition is that transformations in the orientation and scale of an object in image space are converted into translations along orthogonal axes in log-polar parameter space. Various techniques can then be used to provide invariance to this translation, eg. the Fourier transform, [109,91,82,84]. The difficulty in this approach is in ensuring that the mapping is centred on a consistent point within each object. A point commonly chosen is the centroid of the object, which obviously makes the approach sensitive to scene clutter and occlusion.

<u>Assessment</u>

The main advantage of treating object recognition as a pattern classification problem is in the simplicity of model acquisition and matching. The training phase, in which the

system acquires models of the objects to be recognised, involves simply computing the values of the chosen shape characteristic from an image of the object. Optimal matching accuracy can then be achieved by simply applying a classification rule, once the value of a distance metric has been computed between model and image feature vectors. This process can be performed either in parallel or sequentially using a decision tree, and is obviously well suited to implementation in parallel hardware. Systems based on pattern classification therefore have the potential to deliver fast recognition.

However, to date the application of the pattern classification approach has been limited, primarily by its reliance on the global shape representations described above, to the recognition of unoccluded objects. The Fourier coefficients or moment invariants computed for an occluded or composite shape bear an arbitrary relation to those computed for the original shape. Given the stated difficulty in performing bottom-up segmentation prior to representation, it seems clear that recognition based on global shape characteristics faces severe problems in cases where multiple objects may be encountered.

The Need to Make Local Shape Measurements

The above discussion implies that recognition in cluttered scenes should be based upon the matching of *local* shape elements. These could be individual shape primitives, such as edgels or extended line segments, or they could be spatially restricted regions of the shape, commonly termed "parts" Hoffman & Richards, [46]. The obvious advantage of such recognition is that the effect of losing some region of a shape, either through fragmentation noise or occlusion, does not prove fatal, since correspondence should be preserved in the remaining areas.

For this reason a number of researchers have attempted to adapt the global representational schemes mentioned above to deal with the representation of local shape. Gorman et al. [38] have proposed restricting the computation of Fourier coefficients to partial shapes formed by triples of consecutive line segments, a feature vector being associated with the middle line segment. By considering all such triples a shape representation is formed which is composed of n feature vectors, where the shape is composed of n line segments. Taubin [100] has investigated the application of moment invariants to local regions of a shape, each region being defined by a circle. In each case, correspondences between local shape elements are established using standard nearest-neighbour classification techniques.

These approaches are noteworthy in that they attempt to retain the advantages of treating recognition as a pattern classification problem while acknowledging that matching must take place between local shape elements. However, both systems suffer due to the fact that the chosen representational schemes, Fourier coefficients or moment invariants, are sensitive to the loss and/or addition of shape elements. This sensitivity can only be overcome by restricting the region of shape to which representation is limited

to such a degree that local feature vectors become weak, since they are based upon insufficient shape information. Matching based upon such representations therefore tends to be ambiguous.

Conclusion

The conclusions drawn from this section are, firstly, that recognition based on matching local shape elements is essential if the problems arising from scene clutter are to be dealt with successfully. Secondly, performing local shape matching using standard pattern classification techniques has advantages in terms of simplicity, speed, and ease of model acquisition. It has been argued that attempts to adapt what are essentially global shape representations to support such matching do not provide an acceptable solution. The primary motivation behind the present study is therefore to develop a form of local shape representation that can be matched using pattern classification techniques and which is robust to the kinds of shape variation that typically occur in real imaging situations.

The proposed solution is based upon recording the distribution of geometric features, eg. angle and distance, computed between pairs of shape primitives. The next section briefly introduces the concept of pairwise geometric relationships before reviewing previous approaches based upon their use.

1.2.3 The Role of Geometric Relationships in Recognition

Any worthwhile measure of shape should be based, at some level, on its geometric structure. Given a shape described as a collection of primitive elements, the local geometry of the shape can be made explicit by computing the value of geometric features, eg. angle and distance, between pairs of primitives, figure 1–2. The concept of a geometric feature is described in more detail in Chapter 2, for the moment it is sufficient to state that the purpose of a geometric feature is to capture some aspect of the geometric relationship between the two shape primitives. The obvious advantage of such measures is that they depend only upon the presence of the pair of shape primitives between which they are defined. This suggests that recognition systems based on matching the values of pairwise geometric features have the potential to deal robustly with the loss of shape information. Whether this potential is realised depends on the mechanism used to perform matching.

In addition to being local, geometric features have a number of further advantages:

i) Invariance

Geometric features, being relative measures, are invariant to certain rigid transformations of an object. In the case of 2D shape this covers rotation and translation of an

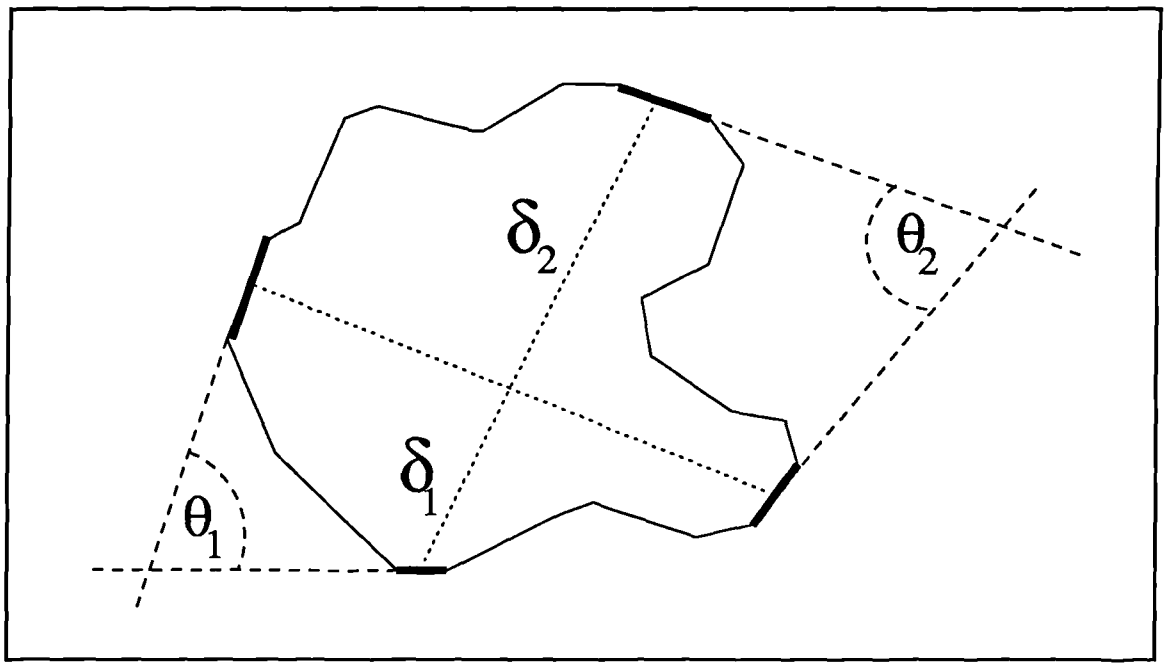

Figure 1–2: Pairwise geometric relationships between line segments.

object within the image plane. The effect of scale changes is less clear and depends on the particular geometric features used.

ii) Robustness

Errors, due both to the sensor and to the feature extraction process, can result in variations in the exact measured position and orientation of a shape primitive from one viewing instance to another. The likely effect of such variations on the expected value of geometric features can be predicted using an appropriate model and accounted for in the representation and matching process, eg. by simply storing bounds on the range of geometric feature values that are accepted as a match, [40].

iii) Flexibility

Although geometric relationships have been introduced using pairs of extended line segments, the concept is a general one. Geometric relationships can, in general, be defined between any number or type of shape primitive. This allows shapes to be described using the most appropriate set of primitives, without affecting the applicability of the characteristic upon which representation and matching is based. This has favourable consequences for the versatility of systems which exploit geometric features for recognition.

Pairwise geometric features therefore provide a robust, invariant measure of local shape that are versatile in their application. Not surprisingly, therefore, many previous approaches to object recognition have been based on the geometric relationships between shape primitives. These schemes are now reviewed.

Geometric Model-Matching Schemes

Previous recognition schemes based on geometric relationships can be viewed as giving increased weight to the role of matching in recognition. Shape representations are often simply records of the pairwise geometric relationships within a shape. Recognition then typically requires a relatively complex mechanism to establish correspondences, eg. tree-search, graph-analysis, or clustering. This is in obvious contrast to the pattern classification approach discussed in Section 1.2.2.

This increased complexity in matching is necessitated by the fact that recognition in these systems is typically based on the values of *individual* pairwise geometric relationships. While these measures have the advantage of being local, they are, consequently, quite weak, since a particular geometric relationship may be common to many pairs of shape primitives, both within and between objects. The matching of shape on the basis of such measures is therefore likely to produce a *set* of candidate matches. Considerable effort must then be expended in order to apply a *global* constraint on matching, so that this ambiguity can be resolved. The mechanism used to apply this global constraint differs between approaches.

Interpretation-Tree Search

If each model primitive is considered as a match for every image primitive then the problem of finding a consistent set of matches can be phrased in terms of a search of the resulting *interpretation tree*. Each leaf node in this tree represents a possible interpretation of the image data that must be evaluated for global consistency. The obvious problem with this simple approach is that the number of possible interpretations grows exponentially with the number of scene and model primitives.[1] For all but the simplest of problems this proves intractable.

A possible solution is to build the interpretation tree dynamically, checking at each branch point whether the matches hypothesised to that point correspond to a consistent transformation of the object in the image. If not then the tree can be pruned at that point. While this drastically reduces the the size of the interpretation tree, the overhead of checking for global consistency at each branch point proves prohibitive.

The solution proposed by Grimson & Lozano-Perez [40,42,41] is to attempt to prune the interpretation tree by exploiting local geometric constraints. This involves checking at each point in the interpretation tree that the hypothesised match observes pairwise geometric consistency with all previous matches. If not then the hypothesised match cannot be part of a consistent transformation and the interpretation tree can be pruned at this point. The force of this constraint increases with the depth of the search, which means that branches of the interpretation tree in which incorrect associations

[1] For S scene primitives and M model primitives there are M^S possible interpretations.

are established, due either to the ambiguity of geometric features or to the presence of spurious shape elements, are quickly pruned. This guarantees that interpretations of the image data derived through tree search are consistent *within* local geometric constraints, [15]. A *model test* must be performed to confirm globally consistency. A degree of robustness to shape variation can be provided by placing bounds on the values of geometric features considered as being consistent. The advantage of this approach is that the pruning constraint can be computed locally. Furthermore, the values of pairwise geometric features between model primitives can be pre-computed and stored in a look-up table to further speed the search process. Pruning on the basis of local geometric constraints proves sufficiently quick to make search tractable.

The efficiency of this basic scheme can be improved by attempting to direct search to promising areas of correspondence space. This is achieved by using various heuristics to order shape primitives, such that those likely to provide more information are considered first. Ayache & Faugeras [3] propose several heuristics. The *proximity* heuristic dictates that the current interpretation should be extended by giving priority to nearby primitives, on the assumption that these are more likely to belong to the same object. The concept of *saliency* is exploited by initially restricting consideration to the 10 longest line segments within a shape. McAndrew & Wallace [67] have extended the notion of saliency by ranking primitives on the basis of the uniqueness of geometric relationships measured between them, both between and within objects. A further strategy for speeding up recognition is to terminate search once a "good enough" fit to the image data has been found, [40,3]. This requires that a certain percentage of the shape boundary is matched. The directed nature of the search means that a termination point is typically reached well before full correspondences are established, with a resulting increase in the speed of recognition.

The major weakness of the search approach is in its sensitivity to missing data; the loss of even a single shape primitive means that valid branches of the interpretation tree cannot be extended. While this can be overcome by allowing model primitives to be matched to the *null character*, it does so at the cost of a considerable increase in computational load.

Graph Analysis

The problem of establishing correspondences between model and image shape primitives can be framed in terms of graph analysis. This involves constructing a graph in which each node represents an hypothesised match between a model and image primitive and an arc between two nodes denotes that the hypothesised matches have pairwise geometric consistency. Given such a graph, recognition can be achieved by performing *Maximal Clique Analysis* [1,12]. A clique denotes a subgraph which is completely connected. Thus, within a clique, each hypothesised match possesses pairwise geometric consistency with all other matches. A clique is maximal if it cannot be extended. Therefore, the largest maximal clique within a graph represents the most

likely interpretation of the image data. A model test must again carried out to check for global consistency. If this fails then further maximal cliques are evaluated.

The limiting factor of this approach is obviously in the size of the graph. Bolles & Cain [12,13] have demonstrated, through the *focus-feature* method, that by exploiting a notion of saliency the number of possible matches, and therefore the size of the graph, can be reduced to manageable proportions. A subset of the model primitives are automatically selected to serve as focus features. Recognition involves selecting, for each focus feature, possible matching image primitives. This process is based on an evaluation of unary constraints, both on the type of primitive, eg. circle or line, and on specific parameters. Matched focus features are then used to predict further matches with image primitives. In the *local-feature-focus* method, consideration of possible matching image primitives is restricted to a circular region, centred on the focus feature, whose radius is determined by the maximum distance between two model primitives. This procedure enables the set of potential matches within certain objects, ie. those containing distinct features such as holes and corners, to be reduced to practical levels. However, its ability to deal with objects described using homogeneous sets of shape primitives, eg. line segments, is less clear.

Relaxation Labelling

An alternative method for extracting the maximal set of consistent feature matches from a graph is *Relaxation Labelling*, Davis [24], Bhanu & Faugeras [10]. This involves constructing a graph similar to that used in maximal clique analysis. The major difference is that connections between nodes are now *weighted* with a measure related to the degree of pairwise geometric consistency between the hypothesised matches. Connections that are below a certain threshold are discarded. An iterative process is then applied which updates the strength of each node according to the degree of support it receives from all other connected nodes. Nodes are removed from the graph if their support falls below a certain threshold. Eventually, this process converges to produce a stable sub-graph that represents the most likely interpretation of the image data.

These three approaches, *interpretation-tree search*, *graph analysis* and *relaxation labelling* all rely on essentially the same mechanism for overcoming the weakness of pairwise geometric features. The strategy employed is the formation of multiple symbolic relationships between matches. This ensures that the set of matches produced are consistent within local geometric constraints. While this does not guarantee global consistency it is often sufficient to rule out all incorrect data associations. The three approaches differ primarily in the mechanism used to establish these relationships. The following approach offers an alternative to the formation of symbolic links which is based on correspondence clustering.

Correspondence Clustering

Bray [15] has proposed using local geometric constraints to support the clustering of

evidence for matches between pairs of model and image primitives. The approach involves the use of several structures; a *Binary constraint array* is used to store the values of geometric features computed between pairs of shape primitives, while a *correspondence array* is used to accumulate evidence for matches between individual model and image primitives. Recognition is achieved by checking for pairwise geometric consistency between pairs of image and model primitives, using the information stored in *binary constraint arrays.* Those associations that are found to be consistent cause a vote to be cast in the correspondence array. Once all pairings have been considered the value in element (u, v) of the correspondence array provides an indication of the support, based on accumulated evidence, for the hypothesis that image primitive i_u matches model primitive m_v. A competition between model primitives is performed to guarantee that an image primitive matches only a single model primitive. Bray notes that matches established in this way are not guaranteed to be consistent within local geometric constraints; the maximal clique algorithm is applied to ensure this. Finally, a model test is performed to validate the global consistency of the matches.

Bray's system overcomes the ambiguity of pairwise geometric features through clustering; valid associations vote consistently for the same element in the correspondence array, whereas votes resulting from false associations, caused by the weakness of geometric features or the presence of spurious shape elements, will tend to be distributed evenly throughout the array. The loss of shape information is also handled naturally within this approach. The computational requirements of this approach are obviously quite large. However, the processing is both local and uniform and so is ideally suited to hardware implementation.

The primary aim in the above systems is to establish valid matches between model and image primitives by carrying out search in the space of possible correspondences. Our discussion now turns to a class of approaches in which matches are hypothesised merely in order to enable a search of the space of possible object transformations.

The Generalised Hough Transform

The Hough transform is a method for extracting parameterised, low-level shape features, such as lines and circles, from, noisy, cluttered image data, (see Leavers [59] for an in depth discussion). Ballard has extended this method, by means of the generalised Hough transform, (GHT), to deal with the problem of locating arbitrary shapes through pose clustering, [4]. Shape is represented in the GHT by means of an "R-Table"; a structure in which the position and orientation of each shape feature, relative to some fixed origin, is stored. Recognition is performed by considering each image primitive as a potential match for every model primitive. For each hypothesised match a set of transformation parameters are computed, using the information stored in the "R-table", that would bring the model into register with the image. These parameters are then used to cast a vote in a quantised representation of transformation space. The mechanism exploited in this scheme is that each valid association between an image and model feature should vote for the same transformation, while votes resulting from

spurious associations will tend to be evenly distributed throughout the space. By detecting peaks in the Hough space, probable object transformations can be obtained. These are then evaluated using a model test.

Hashing Techniques

One of the drawbacks of the standard GHT approach is that each model primitive must be considered as a potential match for every image primitive. This involves a large amount of computation and typically generates a cluttered transformation space, which makes peak detection difficult, especially from noisy scenes. One way of cutting down on the number of matches that need be considered is to apply unary geometric constraints, eg. on line length. Another possibility is to extend matching to multiple shape primitives based on the value of local geometric features computed between them. That is, for a particular set of image primitives, only make votes for transformations computed from model primitives that have the same geometric relationship to one another. A favoured method for indexing sets of model primitives that have a particular geometric relationship is hashing. Systems based on this approach include *Geometric Hashing*, Lambdan & Wolfson [57,58,110], *Structural Indexing*, Stein & Medioni [96,97] and *Multidimensional Indexing*, Califano & Mohan [21]. In these approaches the values of geometric features computed between sets of model primitives are used to compute the value of a hash key. This indexes a position in a hash table in which information concerning the identity and relative position of the model primitives is stored. During recognition, the values of geometric features computed between sets of image primitives are used to index this information, from which possible transformation parameters can be computed. The above approaches differ primarily in the number and type of shape primitives considered and in the geometric features chosen to define the relationship between them.

It should be pointed out that hashing techniques do not rely exclusively on explicit measurements of a shape's geometry. For example, Kalvin et al. [52] propose a method of indexing based on the idea of a *footprint*. Each footprint is a representation, (given by the Fourier coefficients of an *arc length-turning angle* graph), of a segment of the boundary of an object. Segments are obtained by breaking the shape description at concavities. The values of footprints are then used as an index to a hash table.

The Alignment Approach

In the above systems the model test serves to validate hypothesised object transformations obtained by performing a search of correspondence or transformation space. In either case, extensive computation is required in order to derive possible object transformations. Huttenlocher & Ullman [50] have proposed an alternative form of recognition which forgos the need for extensive prior computation by placing increased emphasis on the role of the model test. Likely matches between small numbers of model and image primitives are formed on the basis of unary and binary geometric constraints. These are then used to derive possible object transformations that can be evaluated directly using a model test. Competing hypotheses are ranked on the

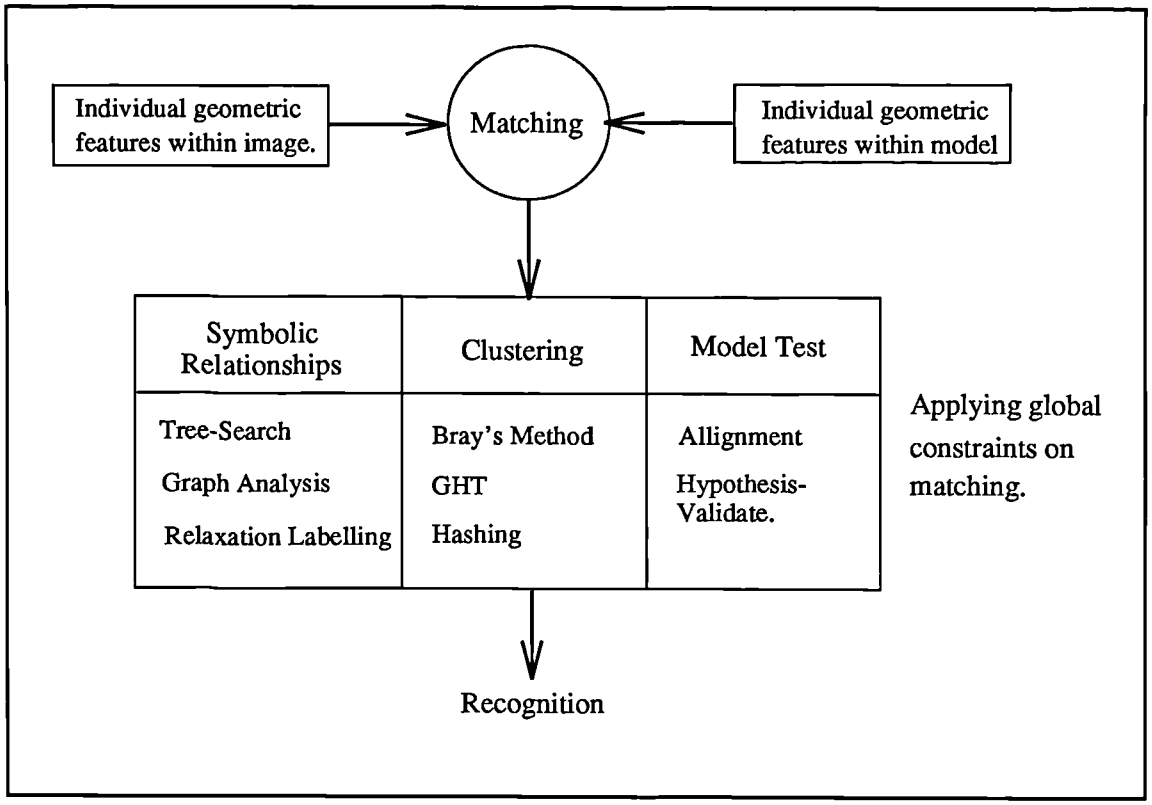

Figure 1–3: Summary of recognition schemes based on individual pairwise geometric relationships

basis of the amount of supporting evidence. Highly ranked hypotheses can either be accepted as valid interpretations, as in [50], or used to direct further search in a hypothesis-validation cycle, as in [3].

The alignment approach overcomes the weakness of geometric features by direct application of the model test. False associations arising from the ambiguity of features are unlikely to generate hypotheses that receive significant image support. Given the computational expense of performing model tests, the practicality of this approach depends on the success with which correct associations can be formed.

Summary

This section has reviewed a number of previous approaches to recognition which base matching on the values of individual geometric features computed between small numbers of shape primitives. The ambiguity arising from the weakness of these local shape measurements is overcome in these approaches by applying some form of global constraint on matching. The mechanisms employed to apply this global constraint include the formation of multiple symbolic links between pairs of matched primitives, established through tree search, graph analysis or relaxation labelling, clustering, performed either in correspondence or transformation space, and alignment. This situation is summarised in figure 1–3.

The conclusion drawn from this review is that if matching is to be based upon the values of individual geometric features then considerable processing is required to overcome

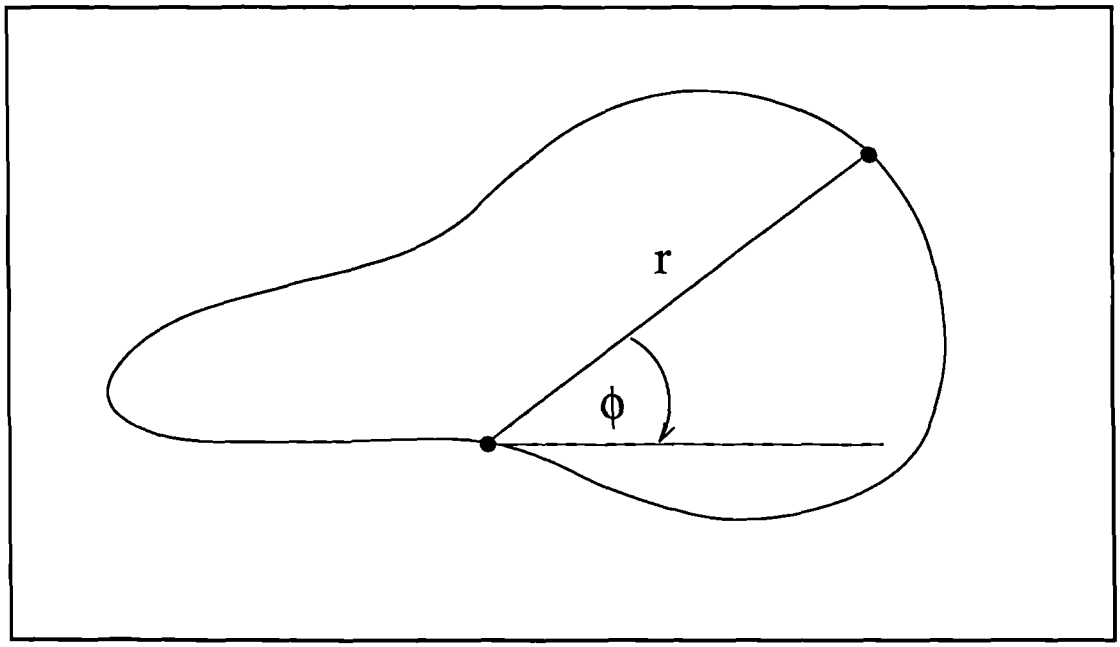

Figure 1–4: A chord defined between two boundary points.

their inherent weakness. The next section discusses an alternative use of local geometric measurements in which matching is based upon multiple, rather than individual, geometric features.

Recording Geometric Feature Distributions

The idea of representing shapes by recording the distribution of geometric features computed between their primitive elements has received relatively little attention in the recognition literature. However, Minsky & Papert [68] suggested as early as 1969 that a shape described by a collection of points could be represented by recording the frequency with which points at particular distances occurred, a form of representation they termed a *distance spectra.* The first serious application of this idea to the problem of object recognition was presented by Moore & Parker [69]. In addressing the problem of extracting features of a pattern suitable for classification, they suggested that a suitable characteristic would be the

> "... non-random distribution of points defining the pattern."

Indeed, they went on to claim that it was the non-random, statistical distribution of features within a pattern which determined its *structure.* Moore & Parker considered shapes represented as a collection of boundary points and invoked the notion of a *chord*, essentially a vector joining any two points on this boundary, figure 1–4. They claimed that the distribution of simple geometric features, such as length and angle, computed for all chords within a shape could be utilised as an encoding of the shape. Given a binary contour image $b(x, y)$, where

$$b(x,y) = \begin{cases} 1 & \text{if (x,y) is a point on the boundary} \\ 0 & \text{otherwise} \end{cases}$$

the chord distribution $h(r,\phi)$, in polar coordinates, is given by

$$h(r,\phi) = \int_Y \int_X b(x,y)b(x + r.\cos\phi, y + r.\sin\phi) \; dxdy$$

The use of the chord to define the pairwise geometric relationship between two points means that this representation is invariant only to translation of a shape within the image. Representations with invariance to either rotation or scale can be derived by integrating the distribution over ϕ or r respectively, although these added invariances are bought at the cost of a decrease in the uniqueness of the representations.

$$h(r) = \int_0^{\pi} h(r,\phi) \; d\phi$$

$$h(\phi) = \int_0^{R} h(r,\phi) \; dr$$

Distributions are normalised to provide invariance to the exact number of contour points sampled. The matching of chord distributions is achieved by computing scalar measures that describe the structure of both angular and radial chord distributions. The measure used in [69] depends simply on the position of peaks within each distribution. The scalar measures for each distribution are combined to form a two dimensional feature vector upon which classification is based. The performance of the scheme was demonstrated on the classification of handwritten numerals.

Smith & Jain, [95] also make use of chord distributions in deriving a test for the "circularity" of a shape, based on comparing the chord distribution of the target shape with that of a circle. In this case distributions are compared directly, the "goodness-of-fit" between two distributions being given by a Kolmogorov-Smirnov test. The use of this scheme for the classification of arbitrary shapes is also investigated.

Burgess et al. [17,18] have recently applied radial chord distributions to the problem of representing 3D shape descriptions obtained through stereo matching. This approach is interesting in that it treats the histogram recording the radial chord distribution for a shape as a feature vector that can be classified using an artificial neural network.

The chord distributions used in these approaches are essentially *global* representations of shape, similar to Fourier coefficients or moment invariants described in Section 1.2.2, and therefore support non-correspondence matching. There is, however, an important difference; representations in the form of chord distributions are constructed from multiple *local* shape measurements. They can therefore be expected to degrade gracefully

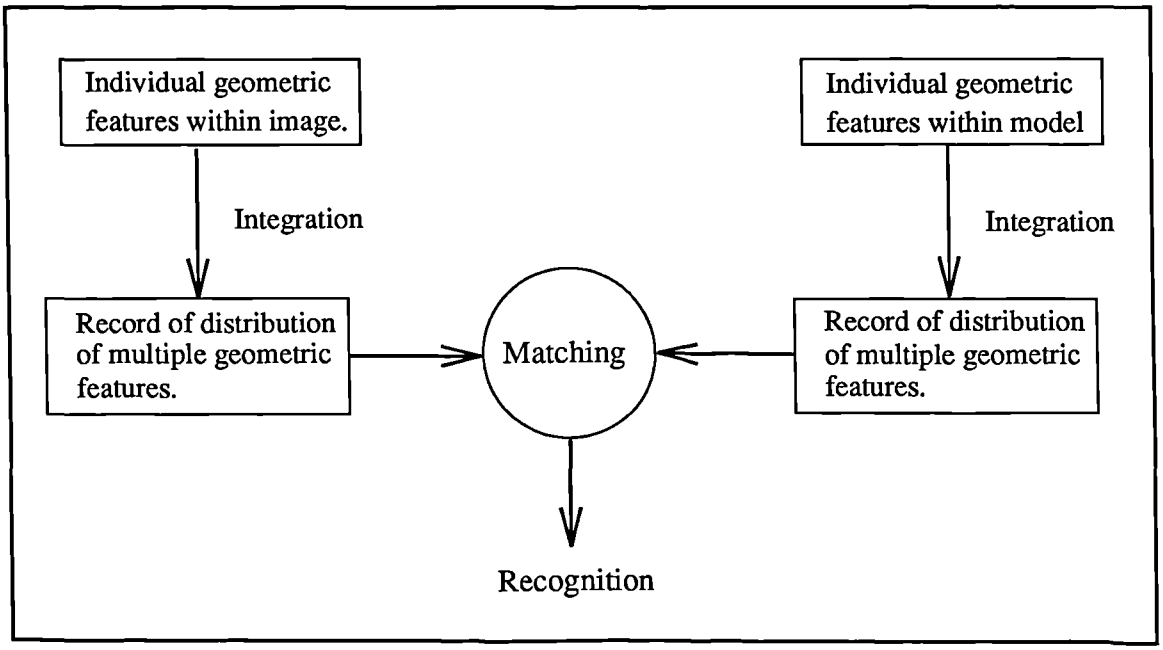

Figure 1–5: Recognition based on geometric feature distributions.

as individual shape primitives are lost through image noise or occlusion. However, they cannot be expected to deal with the presence of spurious shape elements due to scene clutter. As described in Section 1.2.2, this requires that matching be based upon local shape information.

Noll et al. [71] have recently generalised the chord distribution scheme to the problem of representing local elements of shape. This involves recording the distribution of chords defined between a particular shape primitive S_i, in this case a corner feature, and all other primitives within the shape. The histogram recording this distribution is then associated with the shape element S_i. Local chord distributions are computed for all model and image primitives which are then matched using a *context similarity measure*, where this is based on the proportion of intersecting non-zero bins in each distribution. This *context similarity measure* is combined with a unary similarity measure to weight votes in Hough space. This is shown to have significant advantages over the use of the unary similarity measure alone.

Discussion

Representing shape by recording the distribution of geometric features computed between its primitive elements has a number of advantages. Firstly, the integration of multiple local shape measurements provides a strong form of representation. Consequently, matching based on such representations should be much less ambiguous than that based on individual pairwise geometric relationships. Furthermore, the matching of geometric feature distributions can be performed using simple pattern classification techniques, as opposed to the complex procedures described in Section 1.2.3. Geometric feature distributions can be seen as meeting many of the requirements of an ideal shape representation; they are invariant, robust, strong, can be computed easily from

image data and are versatile in their application. Schemes based on matching geometric feature distributions can be seen as integrating information from local shape measurements *prior* to matching, figure 1–5, whereas the schemes described in Section 1.2.3 perform integration *after* matching.

The present study can be seen as an extension of previous approaches based on recording chord distributions, and builds upon ideas originally presented in Thacker & Mayhew, [102]. Improvements are made in the following areas:

i) A more sophisticated definition of the geometric relationship between a pair of shape primitives than that represented by the *chord* is proposed. This involves using geometric features previously proposed in search-based techniques where their values are used as a direct constraints on matching. This has benefits both in terms of the range of object transformations over which representations are invariant and in the strength of the representations.

ii) The matching of geometric feature distributions is performed using techniques from statistical pattern classification. This is more robust than previously proposed methods, eg. [71].

iii) Particular attention has been paid to ensuring that the method by which geometric feature distributions are recorded and matched is robust to variations in shape description caused by fragmentation noise, occlusion and scene clutter.

1.2.4 3D Object Recognition

The problem in 3D object recognition is to associate the wide range of possible views of an object with a single entity, an ability termed *object constancy*, [49]. The variation in the appearance of an object comes about through changes in the spatial relationship between the object and the viewing camera. This variation is in addition to that caused by image noise, occlusion and scene clutter. Approaches to 3D object recognition differ both in the dimensionality of the models used to represent objects and in the dimensionality of the scene descriptions to which they are matched. Three approaches will be considered: systems within the *3D Approach* are concerned with matching 3D models to 3D scene descriptions, those within the *Alignment Approach* relax the need for 3D scene data but still make use of 3D object models, *Multiple View-based Approaches* propose strategies for recognition based on 2D image data and 2D, appearance-based models. These three approaches are now reviewed.

The 3D Approach

The difficulty in performing 3D object recognition comes from the fact that objects exist in a 3D world, while the image descriptions to which they must be matched are 2D, and so *view-dependent*. Therefore, in performing 3D object recognition, some method

must be found for overcoming this loss of dimensionality caused by the projection of a 3D world onto a 2D image plane. A possible solution to this problem, and one which dominated early approaches to 3D object recognition, is to attempt to exploit some form of depth cue present in an image, or images, in order construct a *depth map*, a 3D, view-*independent* description of the contours or surfaces in the scene. Of course, such a description is view-independent only in the sense that it describes the true 3D structure between the set of *visible* object features; the features contained in this set are still determined by viewpoint. However, the availability of a 3D scene description, together with a similarly described set of object models, means that the 3D recognition problem becomes a relatively straightforward extension of the 2D matching problem.

Various "*Shape from X*" algorithms have been developed for the purpose of constructing 3D scene descriptions, each exploiting a different aspect of the depth information available in an image, or series of images, of the scene. These include *Shape from Stereo, Shape from Shading, Shape from Texture, Shape from Contour* and *Shape from Motion.* Alternatively, depth information can be obtained through laser range finding or tactile methods. The problem of matching 3D scene descriptions to explicit 3D models has been addressed using many of the schemes previously reviewed in the context of 2D shape matching. Indeed, many of these schemes were first proposed for this purpose. Representative examples include interpretation tree-search, [41,34,70], graph analysis, [14,77], relaxation labelling, [9], the generalised Hough transform, [6], and hashing techniques, [97]. These approaches are all, at some level, based on the matching of local shape primitives, and so are robust to the loss of shape information caused by self-occlusion. Systems based on these approaches are often able to perform impressive recognition and localisation in cluttered scenes. In certain cases the the accuracy of localisation is sufficient to guide a robotic arm to perform pick and place tasks, eg. [78].

To date, very few attempts have been made to apply pattern classification techniques to the problem of 3D shape matching. This is due, primarily, to the fact that previously proposed global representational schemes, eg Fourier coefficients or moment invariants, are not capable of dealing with the loss of shape information arising from self-occlusion. The application of geometric feature distributions to the task of representing and matching descriptions of local 3D shape is presented in Chapter 5.

The Alignment Approach

The difficulty in obtaining 3D scene descriptions, together with certain results from psychophysical studies which seemed to indicate that humans do not require such input, led a number of researchers to investigate strategies for performing 3D object recognition based on the 2D shape information present in a single image of the scene. Explicit 3D models were, however, retained. So called *alignment* or *hypothesise and test* methods rely on searching both correspondence and transformation space, and are

typified by the work of Lowe [63,62] and Huttenlocher & Ullman [50]. Initial hypotheses regarding the pose of an object are derived from possible matches between model and image primitives, established using some measure of saliency. For example, Lowe proposed using *perceptual groupings*, sets of object primitives whose relationship to one another is preserved over projection, eg. lines that are parallel or which meet at at point. Such qualitatively invariant object features can be used to derive a set of possible matches. The availability of the 3D model allows pose hypotheses obtained from these matches to be evaluated by comparing the appearance of the object from the hypothesised pose with the shape found in the image. Hypotheses which receive significant support are retained and used to direct the search for further matches. In this way hypothesised poses are refined until they satisfy some pre-determined threshold.

As with the 3D approach, alignment methods often perform impressively at recognising and locating objects in cluttered scenes. However, the fact that they require detailed descriptions of the 3D structure of an object means that their use is often restricted. Many classes of *natural* objects are not composed of planar surfaces or straight contours and are therefore difficult to describe using present modeling techniques. Even in the ideal case where objects are polyhedral, the process of obtaining models, often through hand coding, is time consuming.

This leads us to a discussion of multiple view-based approaches, which forgo the need for both 3D scene data *and* explicit 3D models.

The Multiple View-based Approach

Multiple view-based systems are characterised by the use of object models that are view-dependent , in that they are composed of a relatively small number of examples of the 2D appearance of an object from differing viewpoints. Recognition is achieved in such systems by proposing some form of generalisation mechanism for extending recognition from these example views to all possible views of an object. One of the advantages of this approach is its potential for easy model acquisition, since there is no longer the need to specify explicitly the 3D structure of an object. If realised then this would both increase the range of objects that could be represented and introduce the possibility that object models could be learned through the normal operation of the recognition system. Before reviewing previous view-based systems it will be useful to introduce a number of important concepts, together with supporting evidence from psychological and neurophysiological studies.

Characteristic Views & the View-Sphere

One of the major questions in the view-based approach is the basis upon which the possible views of an object are to be grouped. The idea of grouping or clustering views of an object was first given a mathematical foundation by Koenderink & van Doorn [55]. They noted that as an object undergoes various transformations, eg. rotation in depth, the set of visible object features is often quantitatively unchanging.

Of course, at certain points in the transformation changes in the number of visible features will occur. Such points are commonly termed *catastrophic* or *visual* events, and are characterised by the emergence or occlusion of one or more object features. However, between such points the set of visible object features is stable, giving rise to the notion of a "characteristic view". Koenderink & van Doorn proposed that this fact could be used as a basis upon which to cluster views of an object. Objects could then be represented by a relatively small number of characteristic views.

If one considers a sphere whose centre lies at the origin of an object then each point on the surface of this sphere, termed a *view sphere*, can be thought of as corresponding to a particular view of the object. The surface of this sphere can then be divided into regions, commonly termed *aspects*, which represent sets of stable views. Any point within an aspect can be therefore be regarded as a "characteristic" view for that aspect, although points towards the "centre" of the aspect will obviously be more characteristic. Points on the surface of the view sphere where two or more aspects meet are often termed "degenerate" [53,54], since they may well represent a minima in the number of visible features. As such they often pose problems for 3D recognition systems.

The uniqueness of the view-based representation can be improved by encoding temporal links between aspects. This involves constructing an *aspect graph*, in which nodes represent aspects and an arc between two nodes represents the fact that views within each aspect are temporally adjacent. This structure allows the allowable transitions between successive views of an object to be encoded and exploited in resolving ambiguity in recognition.

Support from Psychology

Biological vision systems, and the human visual system in particular, provide a valuable existence proof of a reliable 3D object recognition system. Their study should therefore provide useful insights for the designers of artificial vision systems. There is a growing body of psychophysical evidence which suggests that the human visual system makes use, at least in certain instances, of view-dependent object representations. A number of studies have found that the recognition performance of human subjects, as measured by error rate and/or response time, is better for views of an object that are familiar than it is for previously unseen, or novel, views, [86,99,48,11]. This is not the behaviour one would expect if recognition were based on 3D models, since performance should then be uniform across all views.

Support from Neurobiology

Further evidence for the use of view-dependent object representations in biological vision systems is provided by a series of neurophysiological studies on the visual cortex of monkeys, [73,72]. These appear to show evidence for the presence of cortical cells which respond selectively to particular views of a head. Furthermore, it is claimed that evidence has been found for cells that are responsive to transitions between these views.

Whether these results generalise to less familiar classes of object is not clear. If true then such studies provide considerable support for systems that propose a view-based approach, since the notion of a "grandmother" cell often plays a central role in such systems.

Issues in Multiple View-Based Recognition

Before examining previous multiple view-based systems it will be useful to list a number of issues that can be used in assessing their performance.

i) Model Acquisition Is the potential for automatic model acquisition realised, ie. can object models be constructed from the information available in example 2D images of the object?

ii) View Representation How is the collection of 2D shape features produced by the projection of an object from a particular viewpoint represented?

iii) Grouping Strategy Upon what basis are the views of an object grouped? How does this conform to Koenderink & van Doorn's notion of a characteristic view?

iv) Mechanism of Generalisation What is the proposed mechanism for generalising recognition from the set of stored views to all possible views of an object?

v) Number of Views How many views need to be stored for each object? This is obviously related to the power of the proposed generalisation mechanism and is important in that it determines the practicality of the system.

vi) View Transitions Are the allowable transitions between the views of an object exploited for recognition?

vii) Recognition Information Does the recognition system simply provide identity information or does it also attempt to provide an estimate of object pose?

Previous view-based approaches to 3D object recognition are now reviewed.

Projective Invariants

One approach to view-based recognition that has received considerable interest in recent years is that based on projective invariants. This involves discovering some property of the relationship between the projected features of an object that is invariant over projection. As Burns [19,20] points out, to be of use in recognition the invariant property must be *non-trivial*, in the sense that it generates the same value for all views of an object, or those within an aspect, while generating distinct values for different objects. For example, Rothwell et al. [87] identify a number of relationships between the line segments of planar objects that are invariant to full perspective transformations. Alternatively, a set of object features can be used to define a *canonical frame*, in which all views of the object are mapped to a common curve. Shape measurements computed

in this frame are then invariant to object transformation, eg. Zisserman et al. [112] and Lambdan & Wolfson [57].

This approach has considerable advantages in terms of ease of model acquisition and matching. Model acquisition is achieved by simply computing the value of projective invariants from a single view of the object. Objects can then be identified from the full range of potential viewpoints by simple, direct matching. Furthermore, only one view per aspect need be stored, since the generalisation mechanism is very powerful. Systems based on invariants are also able to provide pose information and can operate in limited amounts of scene clutter.

The major difficulty with the approach is in discovering projective invariants. Previous successes have been limited to finding invariants for 2D, planar objects. Indeed, Burns et al. [19,20], have recently proved that for arbitrary 3D objects, general projective invariants do not exist. If certain assumptions are made about the structure of the objects, eg. if they are polyhedral or symmetric, then invariants can be found. However, the potential of this approach to provide a *general* solution to the problem of 3D object recognition seems limited. For this reason, the approach adopted in this thesis is to base recognition on properties of projected shape that do vary with viewpoint, and to propose mechanisms for dealing with this variation.

Recognition from Hypersurfaces

If the appearance of an object from a particular viewpoint is encoded as a feature vector, which exists in some multi-dimensional space, then the mapping of all possible views of an object into this space produces a collection of points that can be thought of as lying on a hypersurface. The behaviour of such hypersurfaces is determined by many factors, including the characteristics of the representational scheme, the type of projection and the properties of the object itself, eg. whether or not it contains symmetries. The characteristics of the hypersurfaces generated by a particular set of circumstances are crucial in determining the ease with which view-based recognition can be performed. The issue of hypersurface generation is discussed in detail in Chapter 5.

A number of mechanisms have been proposed for achieving recognition based on such hypersurfaces. These often make certain assumptions about the nature of the hypersurfaces, eg. smoothness, which places restrictions on the type of objects that can be recognised and the range of conditions under which they can be viewed. The following sections examine four such approaches; nearest-neighbour classification, learning vector quantisation, linear combinations and view interpolation.

i) Nearest-Neighbour Classification

One of the simplest methods of performing view-based recognition is to extend the nearest-neighbour classification technique previously described in Section 1.2.2. However, recognition now involves determining the identity of the object whose *hypersurface* is closest to the current input feature vector. This involves storing, for each object, a collection of views, each corresponding to a point on the hypersurface of the object.

Recognition is then achieved by performing standard nearest-neighbour classification between these stored views and the current input. A rough estimate as to the view direction can also be obtained in this way. Representative examples of this approach include Richard & Hemani [85], Wallace & Wintz [108], both of whom represent views using Fourier coefficients, and Dudani et al. [25], Reeves [83] who both use moment invariants. Their reliance on global shape encodings means that these systems are not able to perform recognition in cluttered scenes, although Wallace et al. [107] have attempted to extend the approach to perform matching of *local* sections of shape represented as chain codes.

The generalisation mechanism used in this approach is obviously very simple, and assumes that a stored view representing the correct object will be closer to the current view than a view of any other object. If the hypersurfaces of each object are well separated then this assumption should be valid. However, if the hypersurfaces of two or more objects approach one another, or even intersect, then misclassifications can result. The possibility of this occurring is obviously lessened by increasing the number of views that are stored for each object. Consequently, systems which adopt this approach typically have to store as many as 1200 views of each object before satisfactory performance is achieved, [25,85].

The set of views stored for an object are obtained from a fixed tessellation of the view sphere. This is usually achieved using sets of equally spaced points, although Fekete & Davis [33] later proposed a more appropriate tessellation using points from an icosahedron, the views being stored in a structure termed a *property sphere.* However, storing views from a fixed tessellation is naive, since it does not take into account the differential rate at which the appearance of an object varies. Areas of the view sphere in which an object's appearance changes rapidly are represented at the same resolution as are regions in which its appearance is relatively stable. The resolution at which views from an area of the view sphere are stored should be determined by the rate at which the appearance of an object is changing, as evidenced by the behaviour of the hypersurface. The desire to cut down on the number of views that must be stored by taking into account the differential nature of shape variation is one of the motivations behind the following approach.

ii) Learning Vector Quantisation

Edelman & Weinshall [28] have proposed using a self-organising artificial neural network to cluster views of an object. The processing of the network is based upon a a winner-takes-all competition between nodes, each of which represents a particular view of an object. The networks is trained by presenting a series of example views of each object. During training the node closest to the current input, ie. the nearest view, is updated by moving it closer to the input. The result of training is that nodes distribute themselves throughout the feature space so as to optimally represent the hypersurfaces, as sampled by the training views. The main advantage of this approach over simply storing views from a fixed tessellation of the view sphere is that views are

stored at a resolution determined by their distribution in feature space, rather than in pose space. Recognition is essentially the same however, again being based on nearest neighbour classification. Views are represented in [28] as simple coarse codings of the image, and so possess no invariance to object transformations.

Seibert & Waxman [92,91,90] have also proposed using a self-organising ART network to cluster views of an object. Views are represented by a coarse coding of a centred log-polar mapping, and so have full invariance to 2D transformations. Rak & Kolodzy. [82] make use of a similar form of shape encoding. The novelty of Seibert & Waxman's approach comes through the use of a node generation scheme. The value of a *vigilance* parameter controls the maximum allowable variation in shape between the current view and any stored view before a new node is generated. By varying the value of this parameter the number of aspects into which the views of an object are grouped can be varied. However, while views are grouped on the basis of similarity, the criteria upon which grouping takes place is based not on the notion of a characteristic view, but rather on some distance threshold in feature space.

Seibert & Waxman's system is also of interest in that it places a great deal of emphasis on the role of temporal information in performing recognition. Objects are represented using a structure called an *aspect network*. This, it is claimed, constitutes a realisation, using a neural network, of the concept of an aspect graph, and so enables the characteristic transitions between views of an object to be encoded. In the situation where the current, static input view is common to more than one object the system uses differential equations to exploit this temporal information in order to resolve the ambiguity. This use of temporal information means that the system is able to overcome the inherent weakness of its coarse shape encoding.

iii) Linear Combinations

Basri & Ullman, [104,7] have demonstrated that, given certain restrictions, the hypersurface generated by an object exists in a linear subspace. Any view of an object can therefore be expressed as a linear combination of a small number of other views. This led Basri & Ullman to claim that having enough 2D views of an object is equivalent to having its 3D structure specified. Given this finding, an object specific linear operator can be derived which maps any view of the object into a standard or characteristic view. Recognition can then be performed by applying the operator of each object to the unknown view and comparing the transformed view with the set of characteristic views for each object. Obviously, an operator will be specific to views within an aspect, and so for a solid object a number of operators have to be used. There are, however, a number of important limitations to this approach which rule it out for use in a practical recognition system. Firstly, the subspace occupied by a hypersurface is only strictly linear if objects are viewed under orthographic, or parallel, projection. Secondly, and more importantly, linearity relies on the use of a shape encoding that requires full feature correspondences to be known. Such encodings cannot be computed from the information available in an image.

iv) View Interpolation

Edelman & Poggio [75,26] have attempted to generalise the linear combinations approach by characterising the problem of view-based recognition as one of approximating the multivariate function mapping the appearance of an object to its identity, where the value of the function if known only at a small set of points corresponding to the example views. Edelman & Poggio showed that such approximations can be synthesised using a radial-basis-function neural network. Recognition is then achieved by applying the function approximation represented by each network to the unknown view and comparing the result with the characteristic views of each object. While this approach is able to deal with full perspective projection, its success has only been seriously demonstrated as applied to the impractical shape encoding previously proposed by Basri & Ullman. It is unclear whether its performance generalises to more practical shape encodings.

View Clustering

A number of researchers have attempted to provide accounts of view-based recognition that include mechanisms for clustering views into sets that are in some sense *characteristic.* This involves discovering properties of the projected shape of an object that are stable within an aspect. Underwood, [105], proposed a mechanism for automatically grouping views by exploiting the stability of the relationship between the number of bounding edgels describing the projection of connected planar surfaces of an object. Chakravarty & Freeman [23,35] clustered views on the basis of the number of visible junctions of a particular type, such that a range of views could be represented by a single junction-type histogram. This form of shape encoding is potentially quite ambiguous, and so Chakravarty & Freeman exploit the allowable transformations between views of an objects in order to provide resolution of ambiguous views.

Both of these approaches are restricted to relatively simple, polyhedral objects, composed of planar surfaces, whose intersections project in the image as classifiable junctions. Intrator et.al [51] have recently attempted to automatically extract, using an unsupervised neural network, general object features that are stable across many views. This approach is promising in that it provides the potential for clustering views of arbitrary objects.

Geometric Methods

The above methods are, in an important sense, global, since recognition is based on computing the distance between a set of features representing the current view with those representing views of each object. Another approach is to attempt to extend the geometry-based strategies, previously introduced for 2D matching, to perform matching of local shape elements. Again systems can be divided into those that attempt to search correspondence space and those that merely hypothesise matches in order to perform a search of transformation space.

Correspondence Search

Goad [36,37] has extended the interpretation tree search approach, originally proposed for 2D and 3D shape matching, to the problem of recognising 3D objects from 2D image data using view-based object models. This involves storing views from 218 points on the view sphere. The positions of the points are obtained by radially projecting a regular grid on the sides of a cube. Each view is represented by a look-up table recording the geometric relationships between all pairs of object features that are visible from that view direction. This information must be computed from a 3D model of the object, since feature correspondences across views are required to derive the visibility constraints. Recognition is performed by carrying out standard tree search using geometric constraints, together with constraints derived from visibility conditions, to prune invalid interpretations. Bray [15] uses a similar approach although matching is performed using correspondence clustering.

The major difficulty in basing recognition on the values of geometric features computed between projected object features is that such measures are view-dependent. The solution adopted in the above systems is to increase the bounds on the geometric constraints. While this obviously increases the level of ambiguity, the mechanisms for applying global constraints on matches, ie. the formation of symbolic relationships or clustering, effectively overcome this problem.

Transformation Search

Silberberg et al. [93,94] have proposed using the GHT to perform iterative, view-based recognition by means of a *coarse-to-fine* strategy. Initially the view sphere is tessellated quite coarsely using 80 points obtained from an icosahedron. Hypothesised matches between model and image line segments are then used to compute transformation parameters in the standard way. The Hough space used to accumulate votes is initially binned quite coarsely, such that promising areas of parameter space can be identified. These areas are further investigated by repeating the process using a finer resolution of the view sphere, generated by recursively subdividing the icosahedron. The system eventually converges to a solution of the required accuracy. Lambdan & Wolfson [57] have extended the Geometric Hashing scheme to deal with view based recognition.

1.2.5 Summary

This section has presented a review of previous approaches to shape representation and object recognition. The conclusion drawn from this review are now summarised and used to motivate the specific aims of the research carried out in this thesis.

- It was argued that there are certain advantages in treating recognition as a pattern classification problem, but that previously proposed representation schemes have not been able to overcome the problems posed by recognition in cluttered scenes.

Thus, one of the main aims of this thesis is to develop a representational scheme which is robust, invariant, discriminant, which can be used to represent local elements of shape and which can be matched using techniques from statistical pattern classification.

- Previous approaches to recognition that base matching on individual geometric relationships between small numbers of shape primitives were reviewed. It was argued that while these approaches often perform impressively, the local nature of the shape measurements means that relatively complex processing must be performed in order to apply global constraints on matching.

- An alternative use of geometric relationships was reviewed in which shape is represented by recording the distribution of *multiple* geometric features. It was argued that this form of shape encoding has the potential to meet many of the requirements of an ideal representational scheme to be used within a pattern classification approach to recognition.

  Thus, the aim of developing a robust representational scheme is to be achieved by investigating the use of geometric feature distributions. Once developed, the performance of the scheme is to be tested by applying it to both 2D and 3D object recognition.

1.3 Organisation of Thesis

The remainder of this thesis is organised in the following way:

Chapter 2 provides a detailed description of a scheme for representing shape by recording the distribution of geometric feature values measured between its primitive elements. This includes discussions on the appropriate form of the shape primitives, constraints on the geometric features that may be used, the structure of the histogram used to record their distribution and the levels of representation that are possible within the scheme. Finally an assessment of the potential uniqueness of GFD representations is presented.

Chapter 3 describes the use of geometric feature distributions in support of 2D object recognition. The method by which GFD representations may be matched using simple statistical classification techniques to provide correspondences between image and model shape features is presented. This chapter also contains a detailed assessment of the performance of the GFD scheme under various forms of shape variation, including the loss of shape primitives through fragmentation, the addition of primitives resulting from scene clutter and the perturbation in the position and orientation of primitives arising from sensor error. The use of the generalised Hough transform, (GHT), to

determine, on the basis of established correspondences, the position and orientation of an object is discussed.

Chapter 4 provides a demonstration of the ability of the combined GFD and GHT schemes to recognise and locate multiple objects in cluttered scenes under conditions of severe image noise and occlusion.

Chapter 5 presents the application of the GFD scheme to the problem of 3D object recognition. Two approaches are considered. The first involves extending the GFD scheme to the problem of representing 3D shape. The problem of 3D object recognition then becomes a a relatively straightforward extension of the 2D matching problem. The second approach investigates the use of GFD representations of 2D shape within a probabilistic, multiple view-based strategy.

Chapter 6 summarises the work presented in the thesis and discusses its contribution to the field of visual object recognition. Possible directions for further study are identified.

Chapter 2

RECORDING GEOMETRIC FEATURE DISTRIBUTIONS

2.1 Introduction

This chapter describes a method of representing shape which is based upon recording the distribution of geometric features computed between local shape primitives. Much of this chapter is based upon work presented in Evans et al. [30,32,29]. The chapter is organised into the following sections:

1. **Shape Description**
   The process of extracting a line-based description of shape from an image is described and the advantages of basing representation on such descriptions are discussed.

2. **Geometric Features**
   The idea of using geometric features to define the relationship between shape primitives is introduced. A formal definition of a geometric feature is provided along with a list of desirable properties. The chosen geometric features, defined between pairs of line segments, are introduced. These are shown to provide useful measures of shape while possessing the required invariance properties.

3. **Recording Geometric Feature Distributions**
   A histogram is used to record the distribution of geometric features within a shape. The structure of the histogram is presented and the advantages of this approach are discussed. The method of recording the relationship between pairs of line segments is explained and shown to approximate the entry due to an edgel-based description, thereby ensuring that line-based representations degrade gracefully under fragmentation. A flexible method for encoding allowable shape variation is presented.

4. **Levels of Representation**
   The flexibility of the histogramming scheme is demonstrated by considering the different levels at which shape information can be represented in order to support different forms of recognition. Possible extensions of the basic scheme are considered, including the definition of a local region of shape and the use of a measure of saliency.

5. **Uniqueness**
   The factors affecting the uniqueness of the proposed representational scheme are discussed. It is argued that the scale at which shape is represented can be determined simply by varying the parameters of the histogram used to record geometric feature distributions.

6. **Discussion and Summary**
   The properties of the proposed representational scheme are discussed and the content of the chapter is summarised.

2.2 Shape Description

The proposed representational scheme is based upon recording the distribution of geometric relationships between local elements of a shape. The first stage in constructing such representations is therefore to process the image(s) of the scene containing the object in order to extract a description of its shape in terms of a set of primitives. Depending on the nature of the recognition task these primitives may be 2D or 3D. For ease of explanation the proposed representational scheme is introduced using 2D shape primitives; the extension of the scheme to deal with 3D shape representation is presented Chapter 5.

2.2.1 The Need for High-Level Primitives

Various algorithms have been developed for performing *edge detection.* The current system is based upon the use of the Canny edge detector, Canny [22], applied at a single spatial resolution. Edgels are grouped on the basis of eight-connectivity to form edgel strings. A typical edge map of a scene is composed of many such edgel strings, figure 2–1.

The set of edgel strings produced by the Canny operator provide valuable information on the sub-pixel position and orientation of projected object contours. However, the relative simplicity of edgels means that they provide a less than parsimonious description of shape; even quite simple scenes require large numbers of edgels. If one considers that constructing shape representations in the proposed scheme involves making n^2 compu-

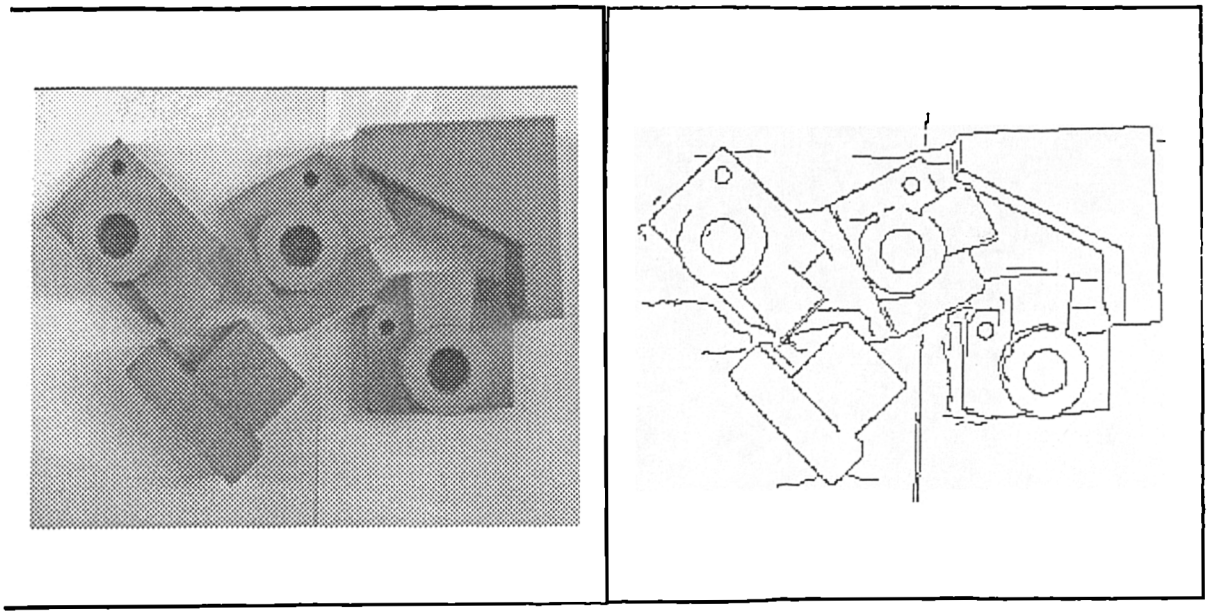

Figure 2–1: An image (a) and (b) the edgels extracted using Canny.

tations, where a shape is described using n primitives, then it is obviously desirable to produce a more compact shape description. This can be achieved by replacing groups of edgels by single, *high-level* primitives that provide an equally good description of underlying shape but which have the advantage of being more compact. For example, linear segments of an edgel string can be described, without loss of any significant information, by a single extended edge or line segment. The following distinctions can be drawn between low and high level shape primitives:

Low-Level Primitives

These are obtained directly from image data, are typically of low complexity and can usually be described using a small number of parameters. Examples are the edgels described above and corner points.

High-Level Primitives

These are obtained by some form of grouping or approximation process applied to sets of low-level primitives. They are typically more complex than low-level features and require more descriptive parameters. Examples are the line segments mentioned above and conics [80].

There are a number of advantages to describing shape using sets of high-level primitives. Foremost is the reduction in the number of shape primitives that need to be considered, with corresponding reductions in memory requirements and increase in the speed of recognition. The increased complexity of high-level primitives means that each geometric relationship has the potential to provide greater shape information.

The added complexity of high-level primitives also means that they are more amenable to some form of saliency measure that can be used to rank primitives. This enables attention to be focussed on a reduced set of primitives, thus providing a further increase in the speed of recognition. Finally, while low-level primitives are, in general, quite stable, their proximity to image data means that certain aspects of their description may have relatively large error tolerances. If the grouping or approximation process is based on the more stable aspects of the low-level primitives then the measurement error of the resulting high-level primitives is reduced. For example, while the detected position of an edgel is relatively stable across images, its orientation is quite sensitive to measurement error. Basing straight line approximation solely on the position of edgels generates a more robust shape descriptor.

2.2.2 Linear Approximation

Having established the advantages of describing shape using high-level primitives we now address the question of which type of shape primitive to use, eg. lines, circular arcs, elliptical arcs, or whether it is preferable to use a combinations of these? If the latter approach is adopted then the set of edgel strings must be segmented into sections that are best described by each class of primitive. While this approach has the advantage of enabling shapes to be described using the most appropriate set of primitives, it is discounted here for two reasons. Firstly, the need to make decisions as to where and how to segment the edgel strings introduces a source of instability into the process of obtaining shape descriptions. Secondly, there are certain advantages, in terms of computational complexity, in being able to compute representations from a uniform class of shape primitives. Shapes are therefore described using a single class of high-level primitive.

As with many approaches based on computing geometric relationships, we choose to use line segments, obtained by performing a linear approximation of the low-level edgel strings. There are a number of algorithms that can be used to perform this approximation. The present system makes use of the recursive-split algorithm described in Ballard & Brown [5]. If shapes are polygonal then a description in terms of line segments is entirely appropriate. Shapes containing curved sections can be described to an arbitrary degree of accuracy by splitting lines until their deviation from the curve is below some pre-determined threshold, figure 2–2. Examples of the kind of line description produced by applying the algorithm at differing levels of accuracy are shown in figure 2–3.

Line segments are a good choice of primitive for a number of reasons. Firstly, provided the level of approximation accuracy matches the degree of curvature in a shape then line segments provide an adequate description of the underlying shape of an object. Secondly, the process of performing linear approximation of the edgel strings is quite straightforward and produces a relatively stable set of primitives. Thirdly the likely

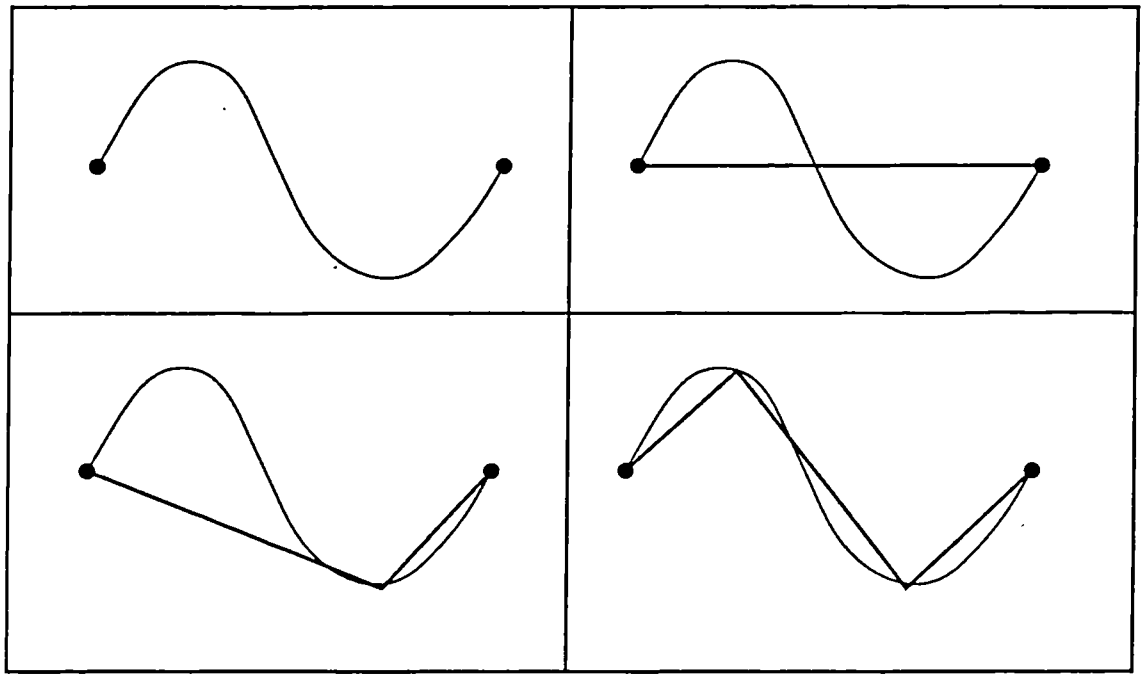

Figure 2–2: The recursive-split approximation algorithm

effect of fragmentation noise and measurement error on the line description can be modelled and accounted for in the construction of the shape representation. Also, despite the fact that line segments are produced by grouping many edgels, they remain quite local. In addition, line segments have the advantage of being suitable for both $2D$ and $3D$ shape description. These factors combine to ensure that measurements based on line segments are quite robust, as evidenced by their use in a number of successful object recognition and stereo systems.

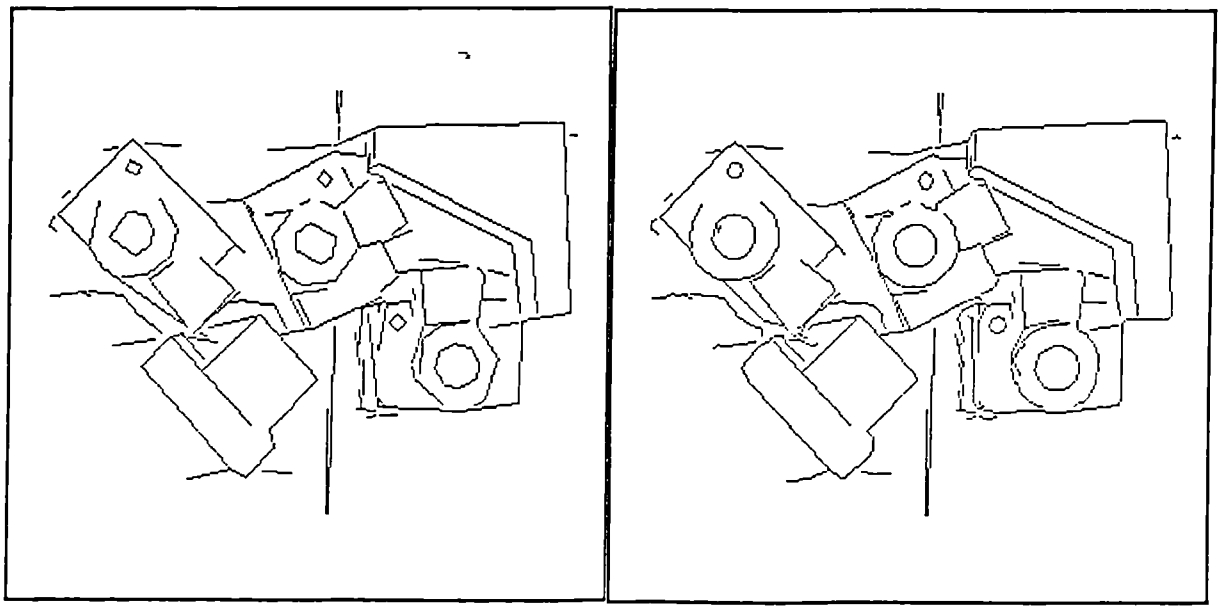

Figure 2–3: Line maps obtained at different levels of approximation accuracy.

2.3 Geometric Features

This section formalises what is meant by a geometric relationship and describes the way in which it can be measured.

2.3.1 Geometric Features

The geometric relationship between a set of shape primitives can be defined to an arbitrary degree of uniqueness by the use of *geometric features*. The purpose of each feature is to capture some aspect of the geometry between the primitives. Thus the angle and distance between a pair of line segments are both examples of a geometric feature. In general, geometric features may be defined between any number of primitives. However, in practice there are two conflicting factors regarding the number that should be considered. If geometric features are defined between too few primitives then there is a risk that they will be too local, in the sense that they do not provide sufficient shape information to be discriminatory. However, features defined between too many primitives have a higher chance of being affected by loss of shape primitives through noise or occlusion. Obviously the complexity of the primitives being used should be taken into account when resolving this decision. The simpler the primitive the greater the number needed to obtain a useful measure of shape. With points, for example, three or four may need to be considered before useful shape information is provided. In the case of line segments an acceptable balance is struck by restricting consideration to binary features defined between pairs of line segments.

Following Bray [16], a binary geometric feature is defined as a function which maps a pair of shape primitives to the set of real numbers. If S is a set of line segments, $S \equiv \{\vec{s_1}, \vec{s_2} ... \vec{s_n}\}$, then the set S^2 is given by

$$S^2 = \{\vec{s^2} : \vec{s^2} = (\vec{s_p}, \vec{s_q}) \text{ where } \vec{s_p}, \vec{s_q} \in S, p \neq q\}$$

We can now define a binary function such that:

$$\textit{Binary Feature} \qquad g_2 : \vec{s^2} \mapsto R^m$$

Where m denotes the number of values returned by the feature.

2.3.2 Feature Properties

It will be useful to list a number of desirable properties upon which the proposed geometric features can be assessed.

Robustness

One of the most important properties of a geometric feature is its robustness to changes in a shape description caused by image noise. The behaviour required of a feature depends to a large extent on the role which its values play in recognition. Consequently, discussion of this property is delayed until details of the representational scheme have been introduced.

Invariance

One of the stated advantages of basing shape representation on the geometric relationships between shape primitives is their invariance to certain object transformations. The range and complexity of the type of transformation over which representations are invariant depends on the way in which geometric relationships are defined, as determined by the chosen geometric features. For the present the required invariance is restricted to the effects of a similarity transforms, ie. changes in the position, orientation and scale of the shape.

Strength

The *strength* of a feature determines the degree to which it characterises the geometric relationship between the shape primitives over which it is defined. A *strong* feature is therefore one whose values, together with details of one shape primitive, uniquely determines the parameters of the second. However, the vast majority of geometric features are *weak*, since they leave certain aspects of the geometric relationship undefined. This can be overcome by combining several weak features to form a feature set. One can then talk of the *completeness* of a feature set, ie. the degree to which the values of the features within a set combine to uniquely define a geometric relationship.

Independence

A set of features is *independent* if the value of any one feature gives no information on the expected value of any other feature in the set. The independence of a feature set guarantees that maximum shape information is obtained from each set of measurements.

2.3.3 The Geometric Feature Set

This section introduces the set of geometric features used in the present scheme.

Relative Angle

The relative angle feature is defined simply as the angle, θ, between the direction vectors of the two line segments, figure 2–4.

$$g_\theta : \vec{s^2} \mapsto R^1$$

$$\theta_{pq} = Angle(d_p, d_q)$$

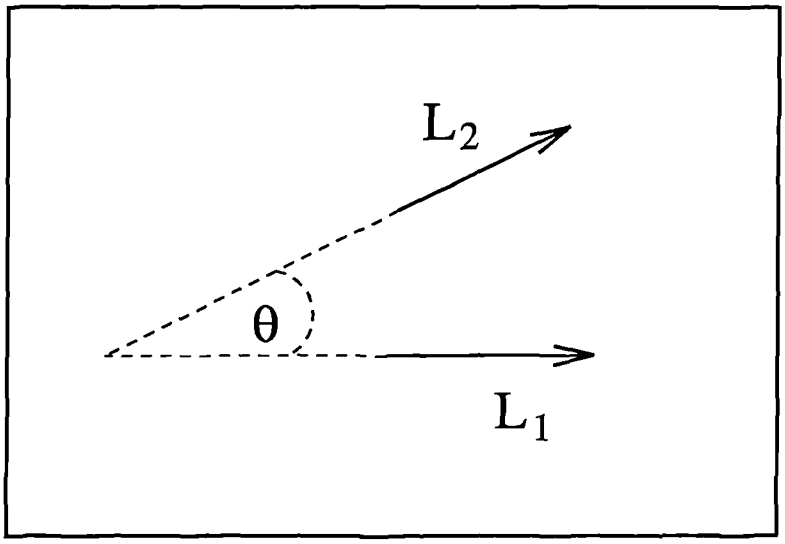

Figure 2–4: The relative angle feature

This definition assumes that the sign of a line's direction vector can be consistently recovered from image data. However, while the process of recovering a line's orientation is quite stable, assigning it the correct direction vector is less so. Two solutions to this problem are proposed. If the shape to be represented is due to a 2D, planar object, then the direction of the intensity gradient upon which the detection of the line segment is based can be exploited. However, if the shape is the result of the 2D projection of a 3D object then changes in lighting may result in local changes in the direction of an intensity gradient. In this case the sign of the direction vectors can be ignored by reducing the range of the angle feature from $[0 \rightarrow 2\pi]$ to $[0 \rightarrow \pi]$. The value of the relative angle feature, θ, is then related to the measured angle θ' by the rule

$$\theta = \begin{cases} \theta' - \pi & \text{if } \theta' > \pi \\ \theta' & \text{otherwise} \end{cases}$$

This obviously reduces the discriminability of the feature, but has the advantage that it can be robustly computed from image data. The remainder of this chapter assumes that line segments are directed.

The relative angle between two line segments is a very intuitive geometric feature, and one which possesses the required invariance properties. It therefore provides a useful feature upon which to base representation. It is, however, quite a weak feature, which suggests that representations constructed using its values are liable to be ambiguous. This is demonstrated by the two shapes shown in figure 2–5. Although these shapes are perceptually quite different, the set of angles within the shapes are identical. Therefore, any representation based on recording these angles could not support discrimination. While this is obviously a severe example, (such symmetries are unlikely to occur in non-geometric shapes), the possibility of ambiguity arising out of the weakness of the relative angle feature remains a problem. For this reason an additional feature is used, based on the distance between a pair of line segments.

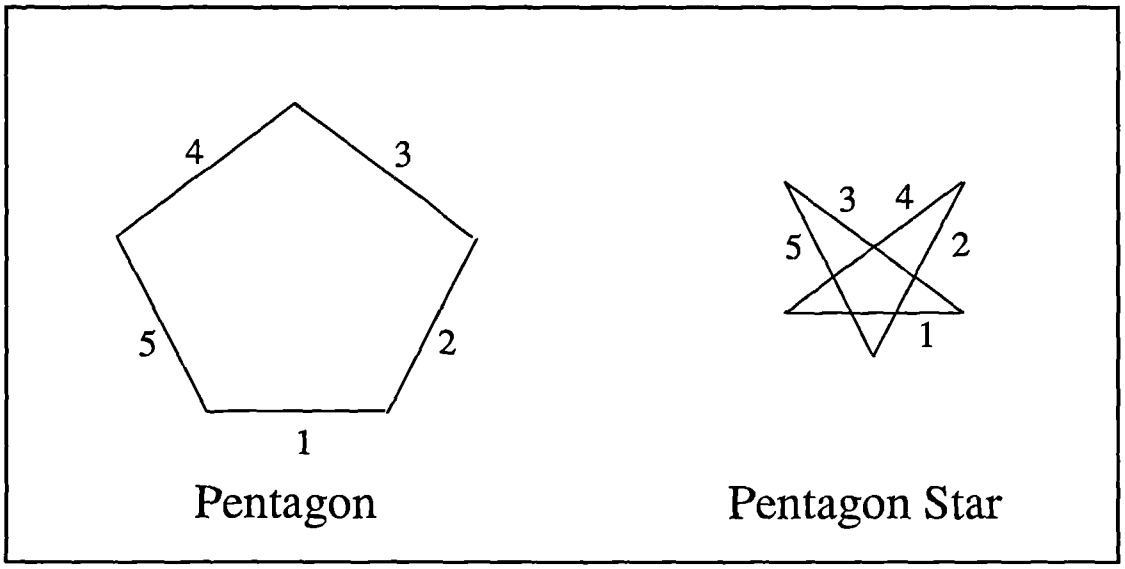

Figure 2–5: The pentagon star ambiguity

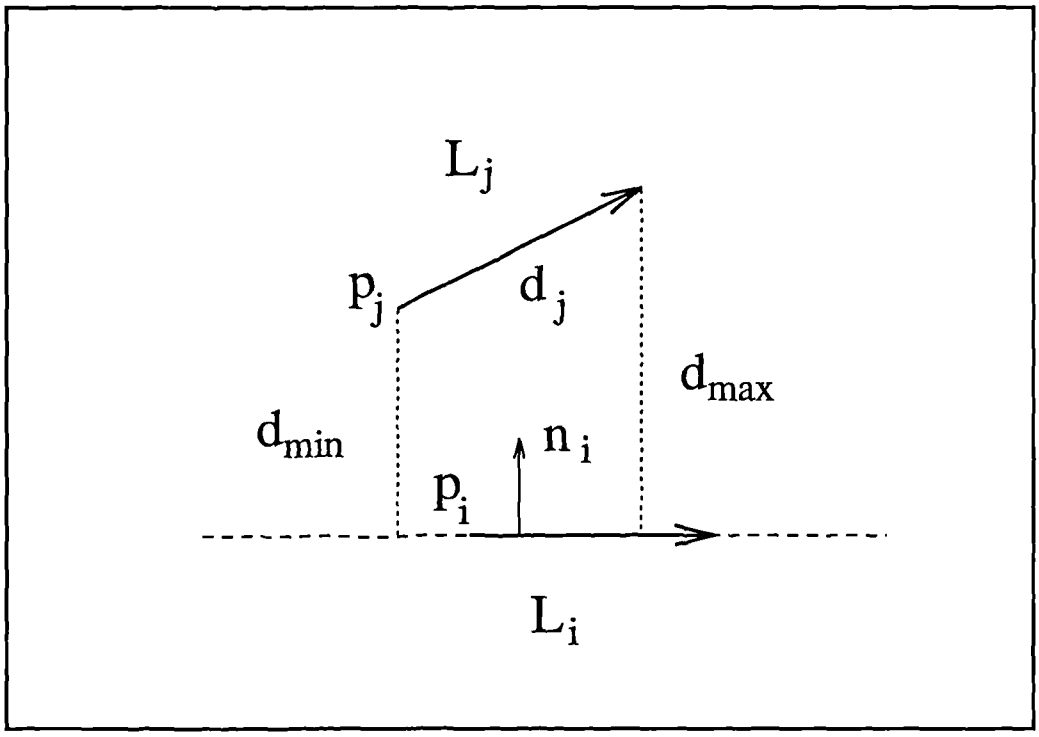

Figure 2–6: The perpendicular distance feature

Perpendicular Distance

Most approaches based on geometric relationships make use of a geometric feature based on some notion of the distance between two line segments. The chosen feature is based on the range of perpendicular distances between the two line segments. Given the two line segments ℓ_i and ℓ_j shown in figure 2–6, the perpendicular distance feature is defined as the range of components of a vector from ℓ_i to ℓ_j in the direction of ℓ_i's normal.

Following Grimson [40], this is expressed algebraically as

$$g_d : \vec{s^2} \mapsto R^2$$

$$g_d = \langle p_j - p_i, \vec{\mathbf{n}}_i \rangle + \alpha_j \langle \vec{\mathbf{d}}_j, \vec{\mathbf{n}}_i \rangle \quad \alpha_j \in [0, \ell]$$

where p_i and p_j are endpoints on lines ℓ_i and ℓ_j respectively, $\vec{\mathbf{d}}_j$ is the unit direction vector of ℓ_j and $\vec{\mathbf{n}}_i$ is the normal to ℓ_i. We are interested in the extrema of this expression, where $\alpha_j = 0, |\ell_j|$. In practical terms this denotes the perpendicular distances from the endpoints of ℓ_i to the extension of ℓ_j, represented by d_{min} and d_{max}. This feature can obviously be applied in either direction. If the direction vectors of the line segments are preserved, eg. by the method explained above, then distances may be measured as being $+ve$ or $-ve$. If signs are ignored then all distances are treated as being $+ve$.

While this feature retains invariance to rotations and translation, the fact that it returns absolute distances means that it is sensitive to scale. A popular solution to this problem has been to normalise distance measurements using the lengths of the two line segments. For example, Bray [16] defines the distance between the mid-points of two line segments as

$$g_d = \frac{d}{|\ell_i| \times |\ell_j|}$$

This approach is discounted on the grounds that it makes the feature value overly sensitive to line fragmentation. However, if a reliable measure of scale is available, eg. from crude stereo or range data, then this can be used to normalise the distances returned by the feature.

2.4 Recording Geometric Feature Distributions

This section describes the details of recording geometric feature distributions.

2.4.1 The Histogram

A histogram is used to record the distribution of geometric feature values within a shape. This is a sensible approach since it enables measurements of local shape to be recorded in an orderless manner. This means that the representation can be readily computed from the available image data. It also enables local shape information to be combined in an additive fashion, with favourable consequences for both the strength and robustness of the resulting shape representation.

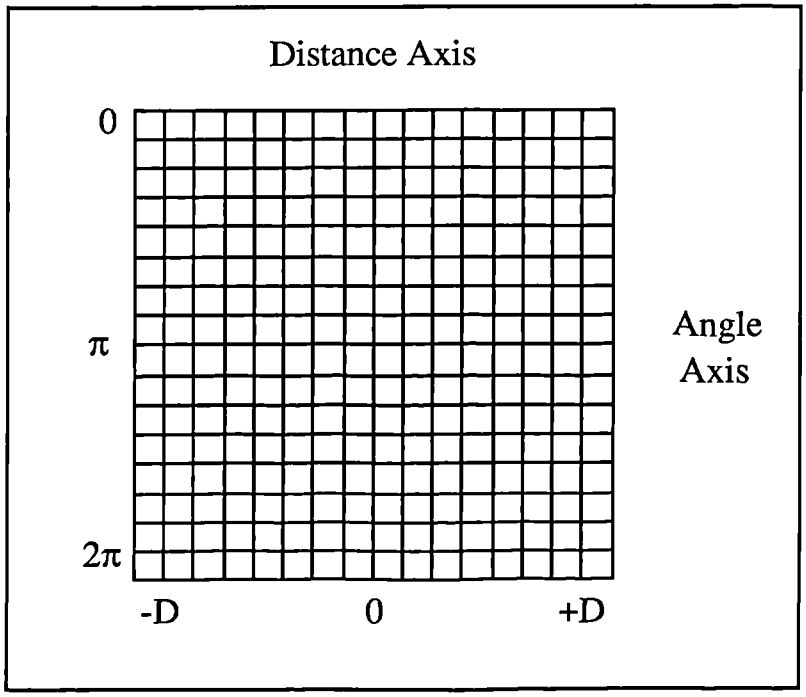

Figure 2–7: The histogram used to record feature distributions

The chosen geometric features have two identifiable parameters: angles, measured in degrees, and distances, measured in pixels.[1]. The histogram therefore has two axes: an angle axis that ranges from $[0 \rightarrow 2\pi]$, and a distance axis that ranges from negative to positive D, where D is the maximum possible distance between two line segments, figure 2–7. This is nominally set to the length of the diagonal distance within the image, although it can be reduced to produce a *local* form of shape representation, (see Section 2.5.1). The angle and distance axes are divided into n_θ and n_d bins respectively. In making the quantisation of the axes uniform it is assumed that geometric features are not biased towards any particular range of values. If this is not the case, for example due to some characteristic of the class of shapes being represented, then the distribution of bins can be varied accordingly.

2.4.2 Ensuring the Validity of Line Approximation

The basic operation in recording geometric feature distributions is to make an entry in the histogram at a position determined by the values of the chosen geometric features computed between a pair of line segments. Constructing the full shape representation involves making many such entries. It was argued in Section 2.2.1 that there a significant advantages to describing shape using high-level primitives, such as line segments, rather than low-level edgels. However, there are important differences to recording

[1]Or physical distances if the shape is 3D

the distribution of geometric features computed between high and low level primitives which can potentially cause problems.

Representing shapes described using low-level primitives, such as edgels, involves making many entries in the histogram. The importance of each edgel in determining the overall representation is therefore small. If an individual edgel is lost through image noise or occlusion then the change in the representation is proportional to the amount of lost data. This ensures that representations constructed from edgel-based shape descriptions should degrade gracefully in noisy conditions. Constructing representations from high-level primitives on the other hand typically involves making far fewer entries in the histogram. Each primitive therefore has a much greater effect on the overall shape representation. This in itself does not cause a problem; the difficulty comes from the effect that shape fragmentation has, both directly on line segments, and indirectly on the value of geometric features.

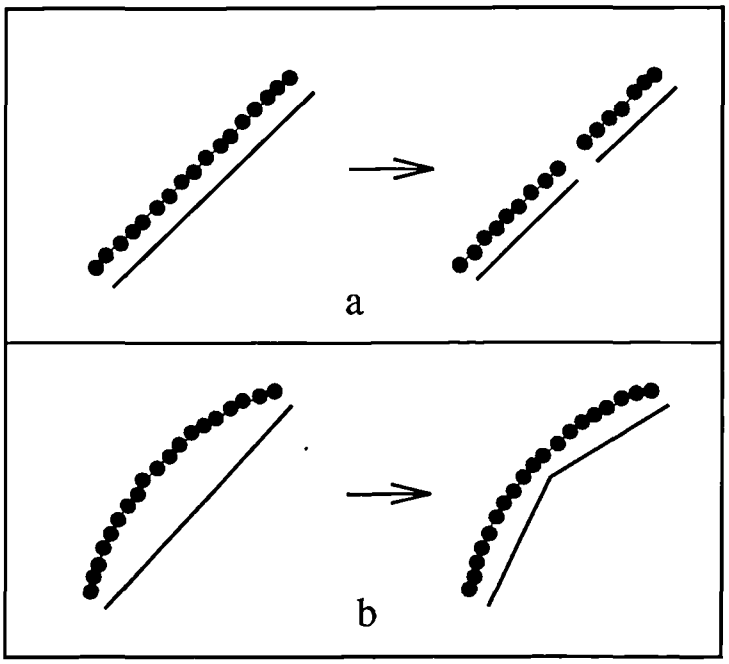

Figure 2–8: (a) The effect of losing an individual edgel, and (b) the effect of an increase in linear approximation accuracy.

The fact that line segments are obtained by grouping many low-level edgels means that the chances of them being affected by image noise or occlusion are increased. The effect of noise or occlusion is, typically, to corrupt the description of the line segment. For example, the loss of a single edgel can cause the fragmentation of a line segment, figure 2–8(a). A similar change can occur if the accuracy of linear approximation is increased, figure 2–8(b). If the chosen geometric features are adversely affected by such changes then this has serious consequences for the stability of the shape representation based upon the distribution of their values. The reason for this can be summarised as follows; the values of geometric features determine the position at which entries are made in the histogram, each entry has a relatively large effect on the overall shape representation, therefore, a small change in the shape description, eg. the loss of a

single edgel, results in a large change in the shape representation. The conclusion from this is that if care is not taken in recording the distribution of high-level primitives then the resulting shape representation will not degrade gracefully.

One solution to this problem is to ensure that the entry made in the histogram recording the relationship between a pair of line segments approximates as closely as possible the entry that would have resulted from considering individual edgels. The next section examines the net effect of recording the distribution of the proposed geometric features computed between two edgel strings. This will used to determine the appropriate form of the entry to be made for the relationship between a pair of line segments.

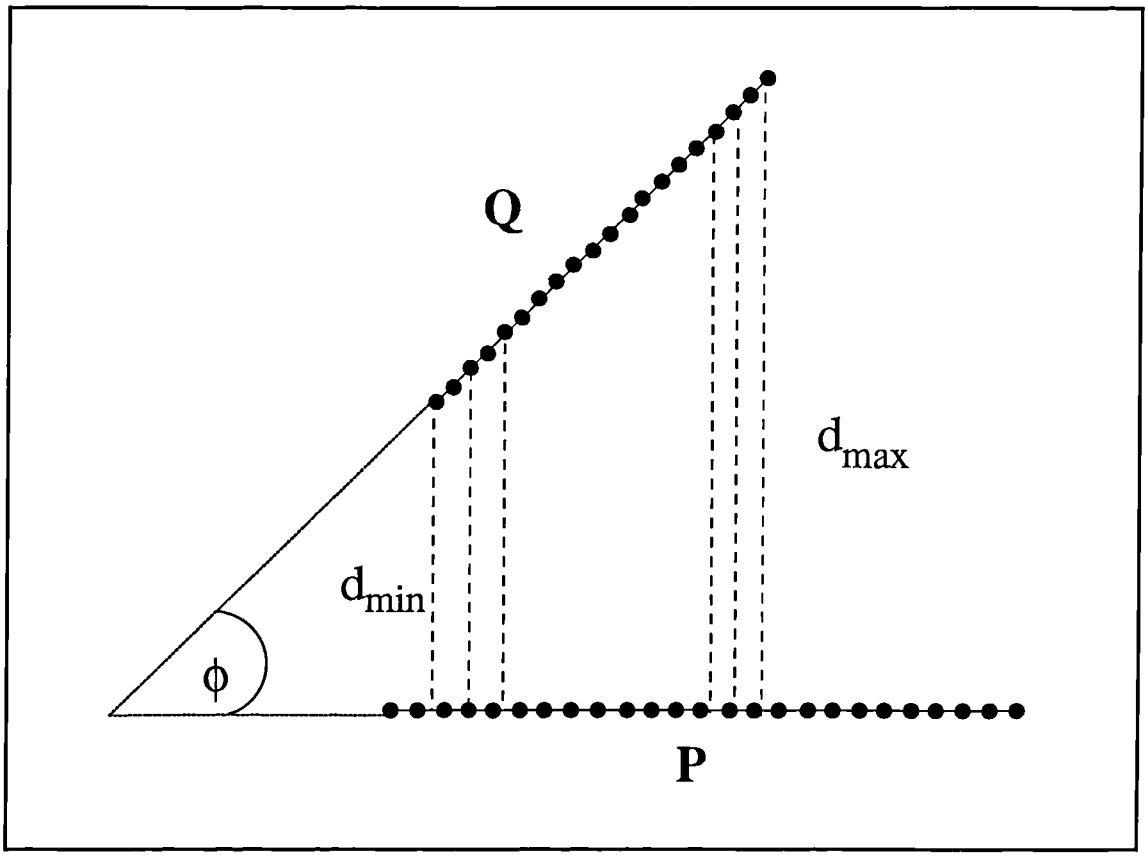

Figure 2–9: Considering edgels.

2.4.3 The Net Effect of Considering Edgels

Consider the two linear edgel strings represented by the sets **P** and **Q** respectively, figure 2–9. We now define the set of geometric features, **G**, computed between individual edgels in **P** and **Q**,

$$\mathbf{G} \equiv \{g_{ij} : g_{ij} = g(p_i, q_j) \quad \forall p_i \in \mathbf{P}, \forall q_j \in \mathbf{Q}\}$$

where g is some geometric feature. We now consider the distribution of values in the sets $\mathbf{G}_\theta$ and $\mathbf{G}_d$, representing the relative angle and perpendicular distance features respectively.

Relative Angle

If we make the simplifying assumption that all edgels within a string have the same direction vector, equal to the mean tangent of the edgel string, [2] then all elements of $\mathbf{G}_\theta$ have the same value, ϕ.

Perpendicular Distance

We now consider the perpendicular distance feature, g_d, applied in the direction from $\mathbf{Q}$ to $\mathbf{P}$. Since g_d depends upon the direction, but not the position, of the first primitive, it must return the same value for all edgels in $\mathbf{P}$. Since an edgel is essentially a point element, g_d computed between two edgels returns a single value. It should therefore be clear that the values in $\mathbf{G}$ must be evenly ditributed between d_{min} and d_{max}, the perpendicular distances from an edgel in $\mathbf{P}$ to to the extreme edgels in $\mathbf{Q}$.

If we now consider recording the distribution of the values in the set $\mathbf{G}_{\theta\mathbf{d}}$ then all entries are made at the same position, ϕ, on the angle axis, and are evenly distributed between d_{min} and d_{max} on the distance axis. The total number of entries made in the histogram is equal to $i \times j$, where $\mathbf{P}$ and $\mathbf{Q}$ contain i and j edgels respectively.

2.4.4 Recording the Relationship Between Line Segments

We now consider how this explanation affects the way in which entries should be made for the relationship between the two line segments ℓ_p and ℓ_q that approximate $\mathbf{P}$ and $\mathbf{Q}$ respectively. If the linear approximation is performed to a sufficient degree of accuracy then the direction vectors of these lines should be roughly equal to the mean tangent of each edgel string. The value of g_θ between the two line segments, ϕ, will therefore be same as that for the individual edgels. Again, if the deviation of the approximating line segments from the edgel strings is sufficiently small then the endpoints of each line should be roughly equal to the position of the extremal edgels in each string. In this case the values returned by g_d for the two line segments will be d_{min} and d_{max}. Thus, if entries are made in the histogram at ϕ on the angle axis and from d_{min} to d_{max} on the distance axis then as far as the position of the entry is concerned the goal of approximating the net effect of entries due to individual edgels has been achieved, figure 2–10. As regards the size of the entry, this is handled by distributing entries in the bins such that the total size of the entry is equal to $|\ell_p| \times |\ell_q|$. This is justified on the grounds that, if approximation accuracy is high enough, then the length of each line segment is approximately equal to the number of edgels it replaces, ie. $|\ell_p| \approx i$. In practical terms, this final step can also be thought of as ensuring that the importance of each line segment in defining the shape is taken into account in the representation.

---

[2] In practice this should be approximately true.

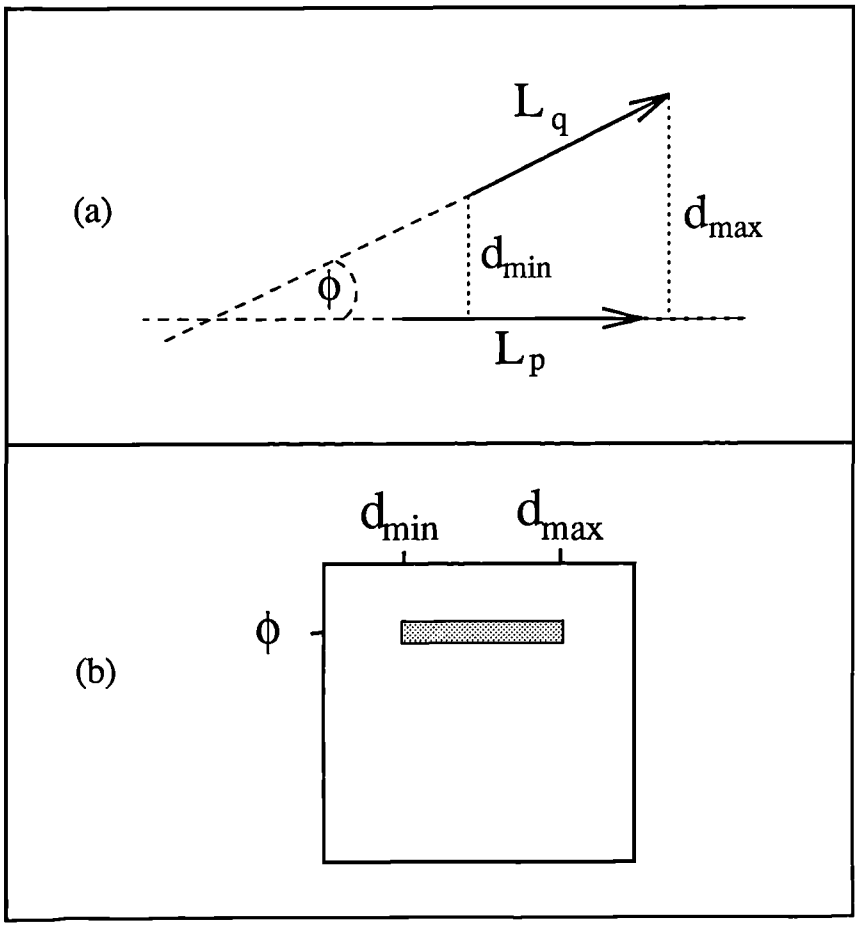

Figure 2–10: (a) the pair of line segments and (b) the position of the entry in the histogram.

This section has demonstrated that, by recording the distribution of appropriate geometric features computed between high-level primitives in the correct way, it is possible to approximate the net effect of considering the distribution between multiple low-level edgels. This enables the representational scheme to exploit the advantages of working with high-level primitives while retaining the robustness of representations based on low-level primitives. In particular, it should ensure that representations based on high-level primitives will degrade gracefully as they become corrupted by fragmentation noise, or vary with the accuracy of linear approximation. These issues are investigated empirically in Chapter 3.

2.4.5 Encoding Allowable Shape Variation

The purpose of any representational scheme is to enable a set of models to be compared with the shape of an object extracted from an image. An overly restrictive form of matching would require that for an object to be accepted as a match for a particular model then their shapes must match exactly. However, in certain circumstances it is desirable to weaken this strict form of matching, eg. where changes in the shape description occur through image noise or object transformation. In either case, both

the position and orientation of line segments may vary, causing a change in the values of the geometric features computed between them. If the representational scheme is to be robust to such changes then some way must be found of encoding these allowable variations in shape.

In the present scheme this involves representing the range of geometric feature values that are to be expected. Many schemes achieve this by simply storing bounds along with the value of the geometric features, eg. [40,16], while others attempt to provide a degree of tolerance by quantising feature values prior to matching, [57]. This latter strategy is obviously central to the present scheme, since the binning of feature values provides a degree of invariance to small variations. However, simply binning or quantising values does not provide a principled solution. The value of geometric features computed between elements of a smoothly varying shape will themselves change smoothly. However, the quantising effect of the binning process means that this smooth change is not translated into the shape representation. This can be overcome by blurring entries in the histogram over several bins. This has the effect of encoding both the sub-bin position of the entry and its accepted variation.

Blurring

In blurring the entry recording the relationship between a pair of line segments it is again important that the effect of the blur approximates the entries that would have occurred had multiple edgels been considered. The problems of blurring entries along both axes are now considered.

Relative Angle

The nature of the distribution of the values of the relative angle feature is not obvious, and will probably depend on the source of the shape variation. For the moment it is assumed to be Gaussian, of width σ_ϕ, figure 2–11. Again considering the entries made for multiple edgels, it should be clear that, since all entries are centred on the same value ϕ, and since Gaussians add in quadrature, the net effect of entries is itself a Gaussian. Thus, blurring the entry for a pair of line segments using a Gaussian of width σ_ϕ, centred on ϕ reproduces the effect of blurring entries for multiple edgels.

A potential problem arises if, as a result of blurring, entries extend beyond the range of the histogram. This can be handled in a principled way by "wrap around"; blurred entries which extend beyond the limits of the angle axis are moved to the opposite end of the axis. The angle used to determine the position in the histogram, ϕ', is then related to the blurred angle by the rule

$$\phi' = \begin{cases} 2\pi + \phi & \text{if } \phi < 0 \\ \phi - 2\pi & \text{if } \phi > 2\pi \\ \phi & \text{otherwise} \end{cases}$$

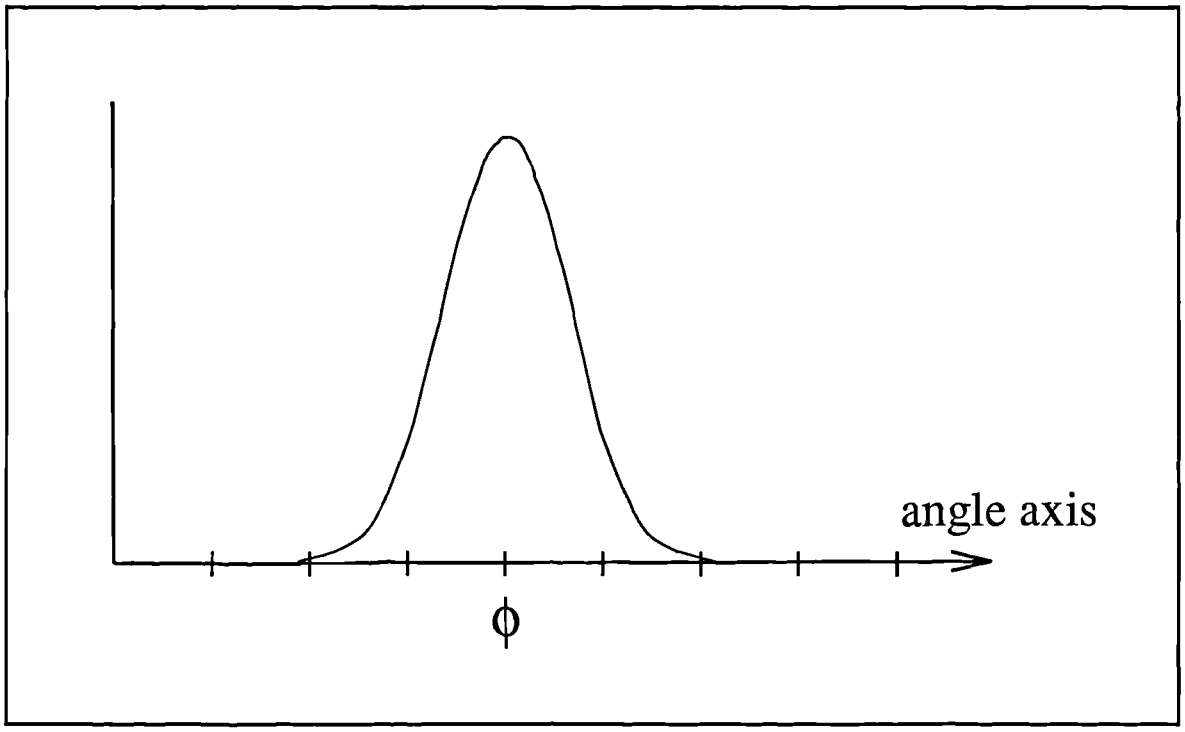

Figure 2–11: The Gaussian blurring function used on the angle axis.

Perpendicular Distance

Determining the distribution of perpendicular distance values is again problematic. For convenience, a rectangular blurring function of width σ_d, is used, figure 2–12.

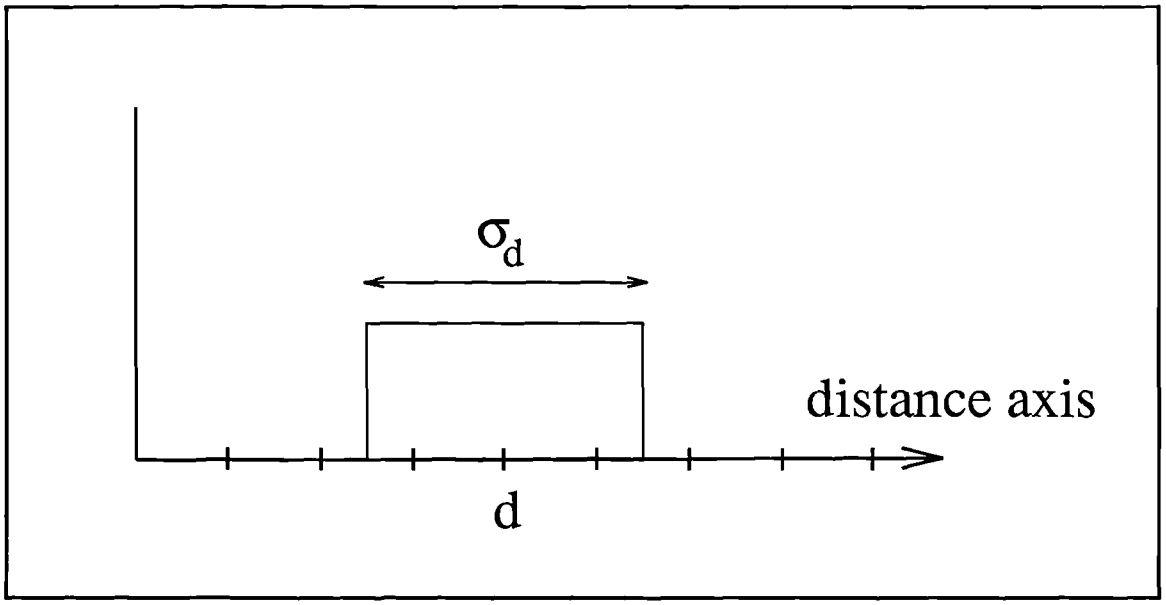

Figure 2–12: The rectangular blurring function used on the distance axis.

The net effect of making multiple blurred entries for pairs of edgels is shown in figure 2–13. In general the entry has the form of a rectangle, of width $d_{max} - d_{min} - 2\sigma_d$, with a linear ramp at either end of width σ_d, although for parallel lines, where $d_{max} \equiv d_{min}$, the entry is simply a rectangle. As with the blurring on the angle axis, this effect can be reproduced when making an entry for the pair of approximating line segments.

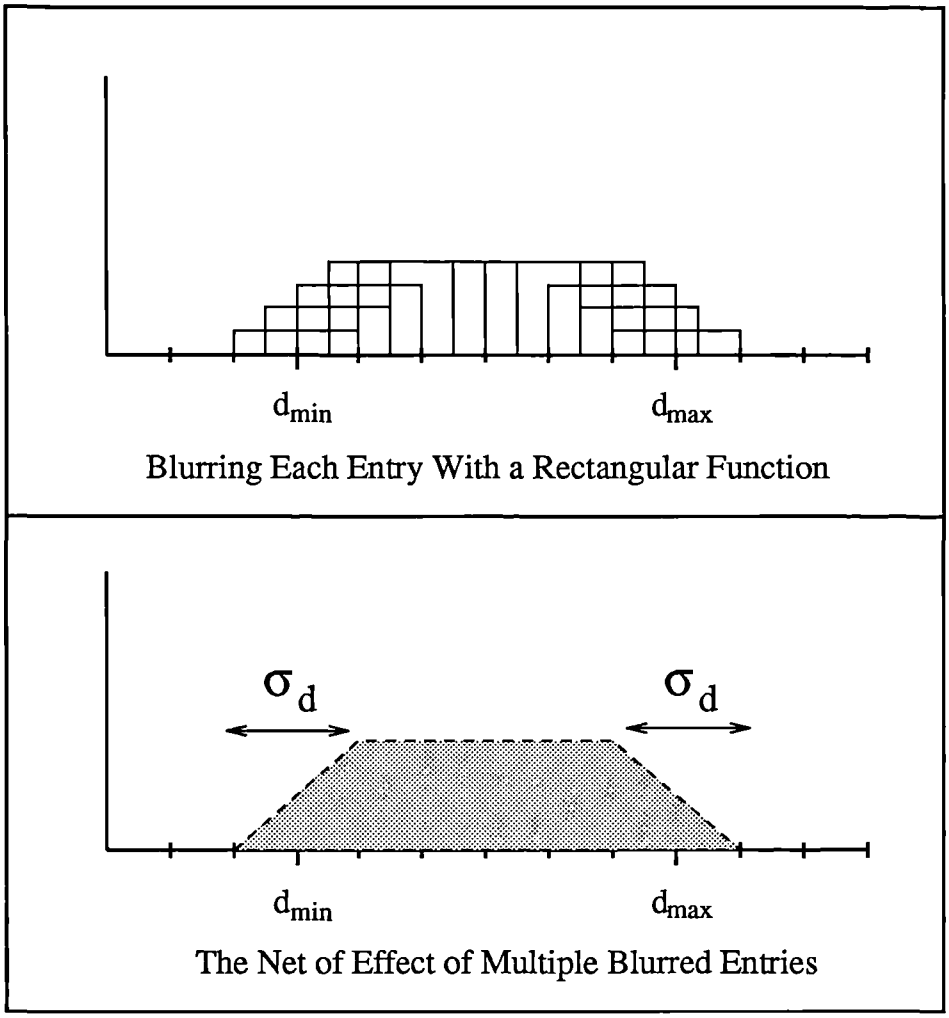

Figure 2–13: The net effect of blurring multiple entries on the distance axis

Again, cases where entries are blurred outside of the range of the distance axis must be considered. Such entries are simply accumulated in the final bin in the histogram, such that the value used to make entries in the histogram, d' is related to the blurred distance d by the rule,

$$d' = \begin{cases} +D & \text{if } d > D \\ -D & \text{if } d < -D \\ d & \text{otherwise} \end{cases}$$

where D is the maximum possible perpendicular distance between two line segments.

Number of Bins

The resolution of the histogram, as determined by the number of bins used on each axis, should obviously be related to the degree of expected shape variation. This effectively places an upper bound on the number of bins that should be used. For example, the maximum number of bins that should be used on the angle axis is given by

$$N_\theta \leq \frac{2\pi}{\alpha}$$

where α is some measure of the expected variation in the relative angle feature. Clearly, the width of blur, σ and resolution, r, used in the histogram should be related to one another; if σ, is much greater than r, then entries are blurred over too many bins, while in the reverse situation blurring becomes redundant. However, the optimal relationship between the values of σ and r is not immediately obvious, although $\sigma = r$ provides a good starting point. This relationship can be varied depending on the circumstances of a particular application.

2.5 Levels of Representation

The previous section has described the method for recording evidence for an individual geometric relationship. However, the proposed representational scheme is based upon recording the distribution of *multiple* relationships within a shape. Exactly which set of geometric relationships should be considered is determined by the type of object recognition that is to be supported. If the goal is to establish correspondences between individual line segments then local geometric feature distributions are the appropriate level of representation. Alternatively, if the aim is to match whole shapes then global geometric feature distributions should be used. Both levels of representation are now presented.

2.5.1 Local Geometric Feature Distributions

The goal in correspondence recognition is to establish matches between the set of image primitives $\mathbf{I} \equiv \{\vec{i_1}, \vec{i_2}, ...\vec{i_m}\}$ and the set of model primitives $\mathbf{M} \equiv \{\vec{m_1}, \vec{m_2}, ...\vec{m_n}\}$. This can be achieved by comparing the distribution of values in the two sets $\mathbf{G}_p$, and $\mathbf{G}_q$:

$$\mathbf{G}_p \equiv \{g_{pj} : g_{pj} = g(\vec{i_p}, \vec{i_j}) \quad \forall \vec{i_j} \in \mathbf{I}\}$$

$$\mathbf{G}_q \equiv \{g_{qj} : g_{qj} = g(\vec{m_q}, \vec{m_j}) \quad \forall \vec{m_j} \in \mathbf{M}\}$$

where g is some geometric feature. If the distribution of feature values in $\mathbf{G}_p$ matches that in $\mathbf{G}_q$ then there is a high probability that image primitive i_p matches model primitive m_q.

Computing the values in $\mathbf{G}_p$ involves defining a local coordinate frame in which the *x-axis* is aligned with the direction vector of the line ℓ_p, (termed the *base* line), and the positive *y-axis* is placed 90° anti-clockwise from the *x-axis*, figure 2–14(a). Entries for the geometric relationship between the base line and all other lines in the shape, measured in this coordinate frame, are made in a a histogram, H_p, associated with ℓ_p, figure 2–14(b). The total value of entries in H_p is therefore equal to $|\ell_p|.L$, where L is the total length of lines in the shape.

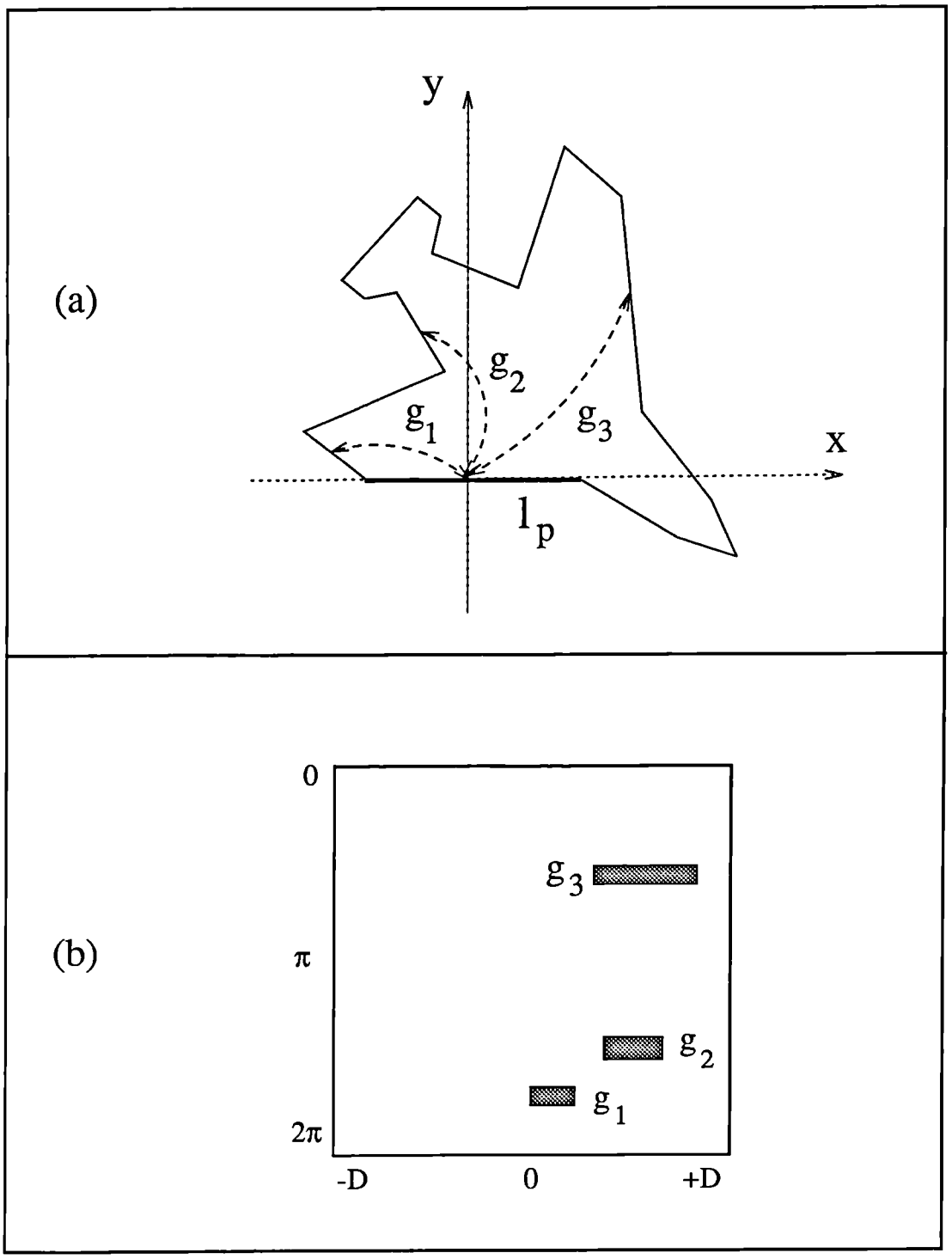

Figure 2–14: (a) the local coordinate frame defined for the base line and (b) the associated histogram.

Figure 2–15 shows the histograms recording the local geometric feature distributions for two particular lines within a shape. Each histogram can be thought of as a template for the line. Given the way in which individual entries are made in the histogram, the value of each bin records the relative frequency with which edgels at a particular geometric relationship to the base line occurred within the shape. If the histogram is normalised then bin valies can be taken to indicate the probability of such an event, for the particular line and shape being represented. This is important, since it means that the representation is well suited for use within a statistical pattern classification approach to matching.

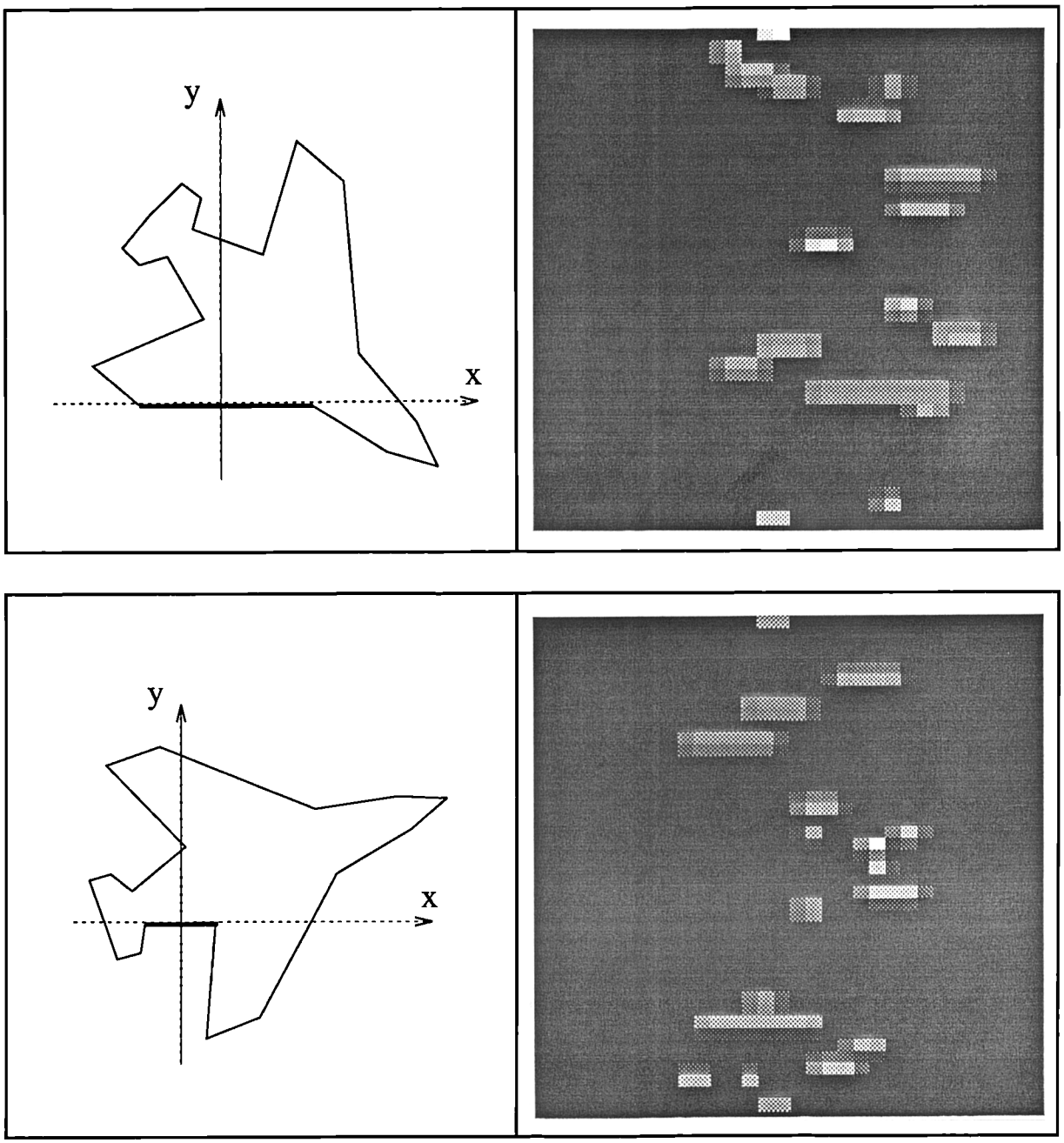

Figure 2–15: The histograms for two lines within a shape.

Figure 2–16 shows the histogram recording the geometric feature distribution for a line within a circle. It can be seen that the smoothness of the circle's shape is captured in the representation. It will be appreciated that the local geometric feature distributions of lines within completely symmetric shapes, such as the circle, will be identical.

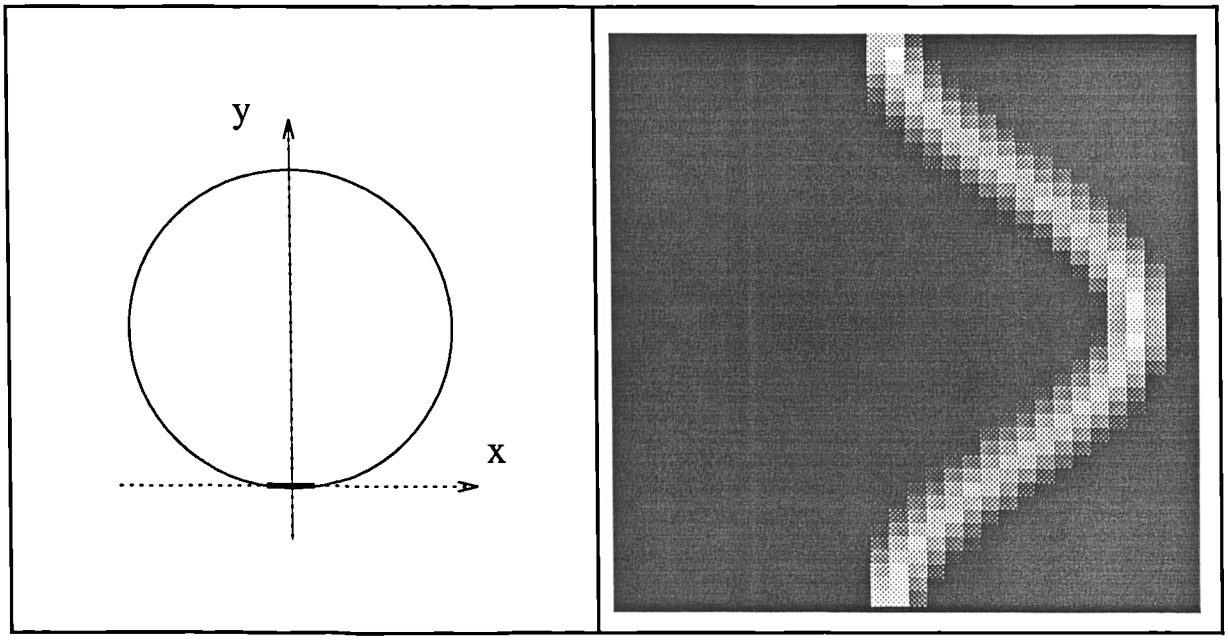

Figure 2–16: The histogram for a line within a circle.

The full shape representation is constructed by recording the local geometric feature distribution for each line within a shape. The full shape representation is therefore composed of n histograms, where there are n lines in the shape, figure 2–17.

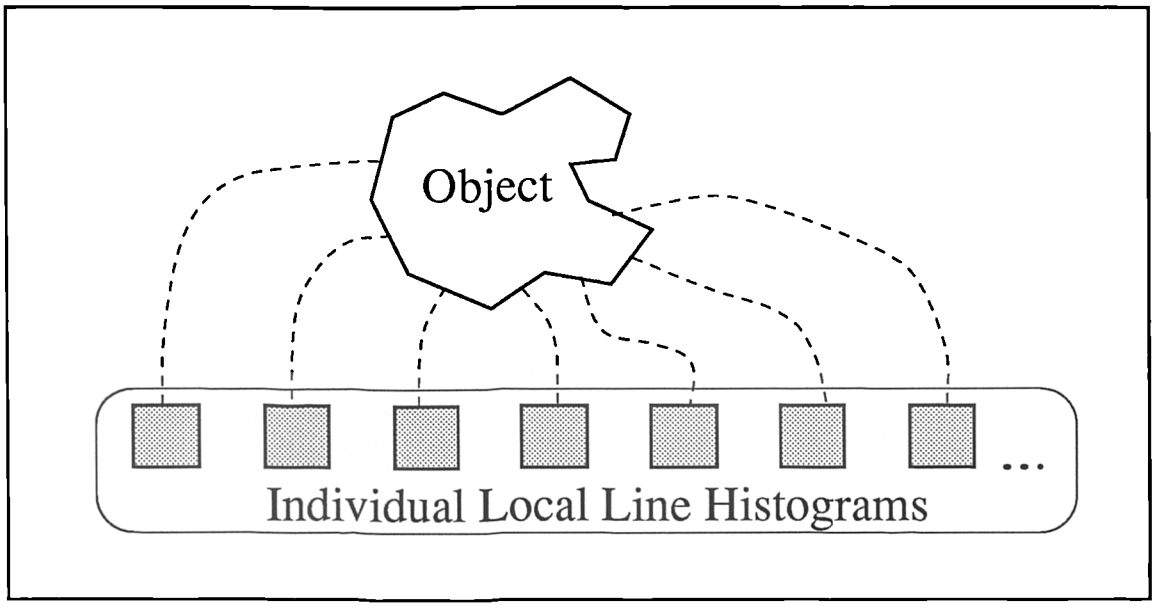

Figure 2–17: Complete shape representation

2.5.2 Global Shape Matching

Non-correspondence recognition is based upon matching global representations of shape. Such representations can be obtained in the present scheme by recording the distribution of geometric features within the set **G**, where

$$\mathbf{G} \equiv \{g_{pq} : g_{pq} = g(\vec{m}_p, \vec{m}_q) \quad \forall \vec{m}_p, \vec{m}_q \in \mathbf{M}\}$$

The histogram, H, recording the distribution of features values within **G** can be treated as a representation of the complete shape. In practical terms, this global form of shape representation is obtained by summing the individual histograms recording local geometric feature distributions, figure 2–18. The total value of entries in the histogram, once all relationships have been recorded, is equal to L^2, where L denotes the sum of the lengths of all lines within the shape. The value in each bin of the histogram records the relative frequency with which two edgels at a particular geometric relationship occurred within the shape. If H is normalised then bin values can be taken to indicate the probability of such an event, for the particular shape being represented.

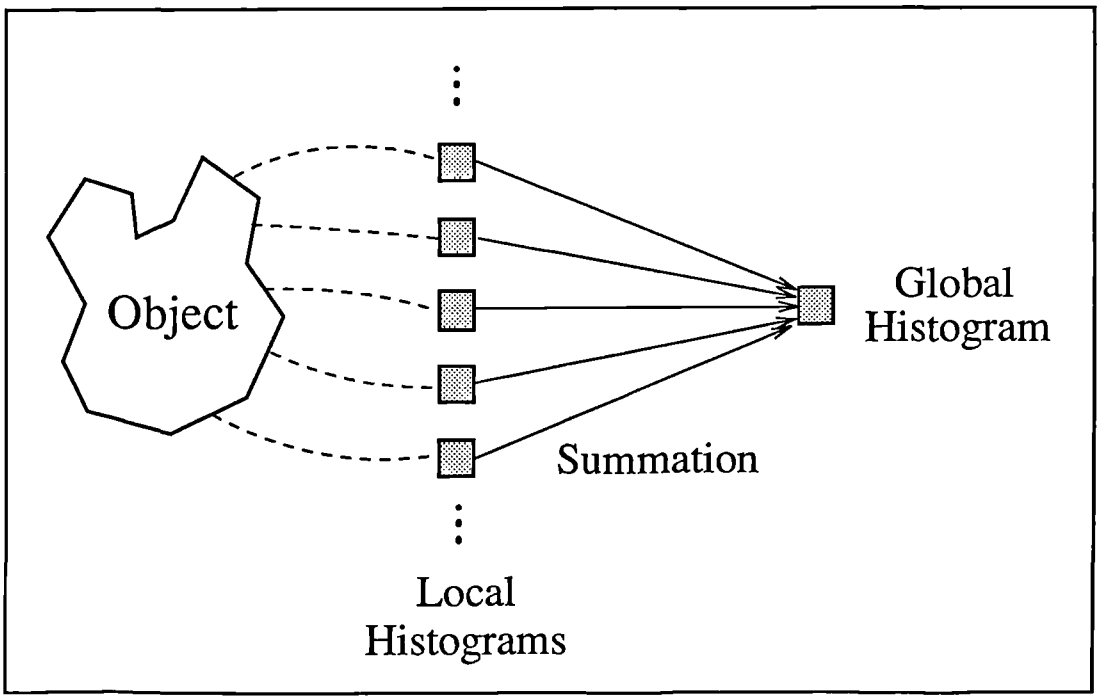

Figure 2–18: The construction of global geometric feature distributions.

The histogram recording the global geometric feature distribution for a particular shape is shown in figure 2–19. No blurring was used. It can be seen that the histogram is symmetric about π on the angle axis, due to the symmetry within the shape.

This form of representation is capable of supporting the matching of whole shapes. However, there is a fundamental difference between this form of global representation and that provided by such measures as Fourier coefficients or moment invariants. Whereas the latter are based on a single, global shape characteristic, and are therefore

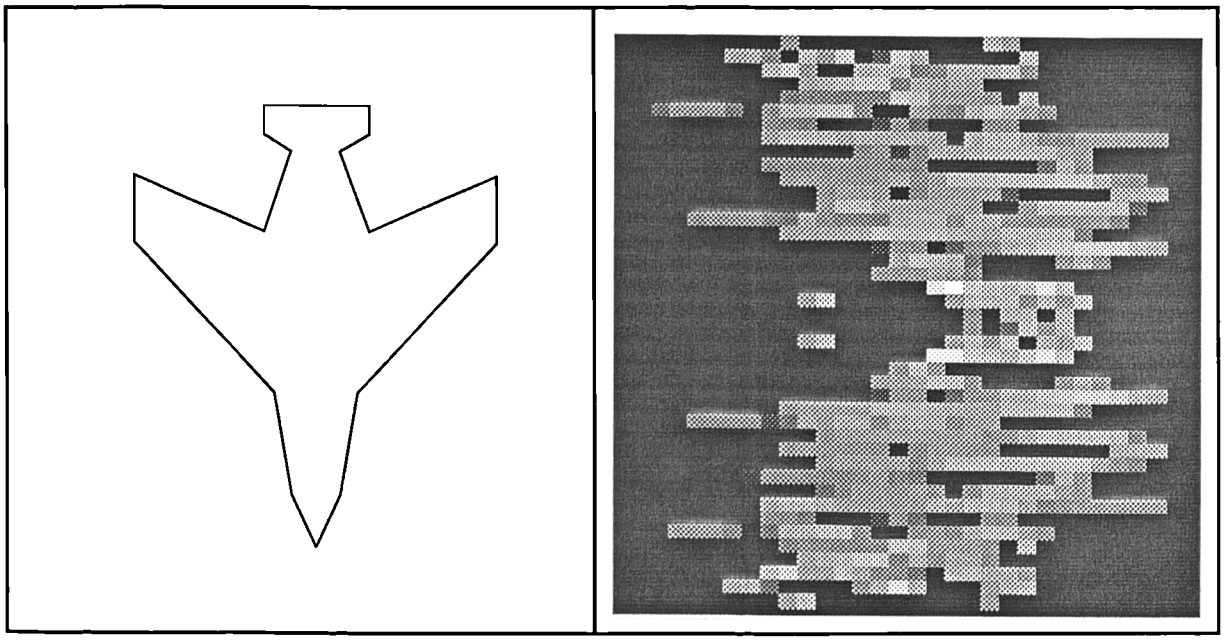

Figure 2–19: The global geometric feature distribution for a shape.

sensitive to data loss, global geometric feature distributions are constructed from multiple local measurements, and should therefore degrade gracefully as shape primitives are lost through fragmentation noise.

2.5.3 Extensions

This basic representational scheme can obviously be extended in a number of ways. Two strategies are considered; one based on exploiting a measure of *saliency* and the other based on restricting the range over which geometric relationships are defined.

Local Shape Representation

The above scheme is based upon recording the the distribution of geometric features computed between a base line and *all* other lines in the shape. This means that if a shape description is corrupted, either through fragmentation noise or occlusion, or if there are lines due to spurious elements in the scene, then the representation will be affected. While the proposed representational scheme is quite robust to such changes, as demonstrated in Chapter 3, it is obviously preferable if the likelihood of the representation being affected can be reduced. This can be achieved by restricting the range over which the geometric relationships included in the representation are measured. This involves using a function f to define a local region around the base line. The definition of the set $\mathbf{G}_q$ then becomes:

$$\mathbf{G}_q \equiv \{g_{qj} : g_{qj} = g(\vec{m}_q, \vec{m}_j) \quad \forall \vec{m}_j \in \mathbf{M} \;\; AND \;\; f(m_q, m_j) = TRUE\}$$

The function f can obviously take a number of forms. One of the simplest is to define a circle, of radius D, centred on the mid-point of the base line. Measurements are

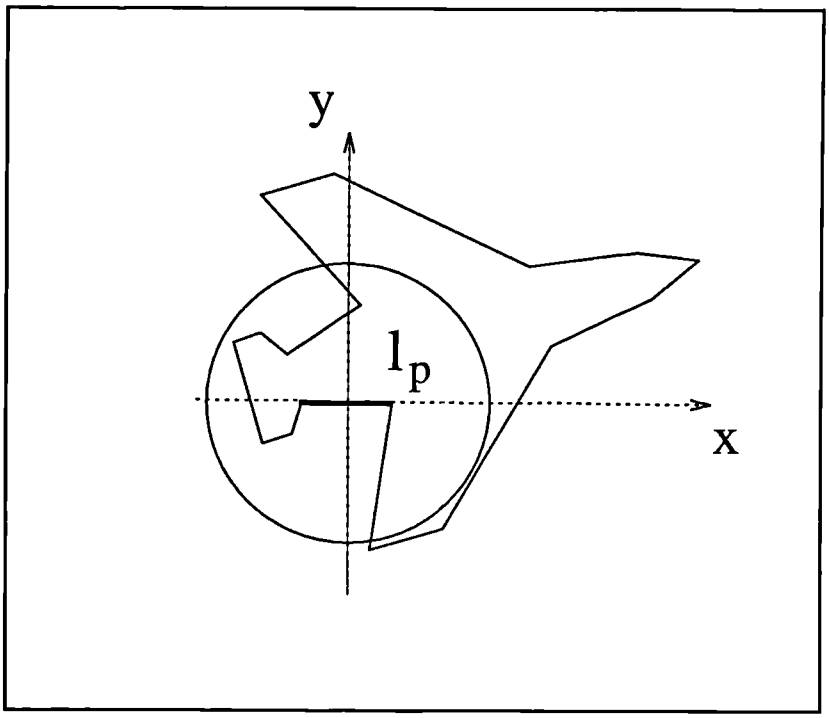

Figure 2–20: A circular local region defined around the base line.

made with all other shape lines whose mid-points lie within this circle, figure 2–20. While this effectively restricts the range over which measurements are made it does, however, impose a *hard* cut-off. As a consequence, the representation of smoothly deforming shapes may suffer. Also, its reliance on the mid-points of lines, which are not robust characteristics, means that it is liable to be badly affected by image noise. Another method would be to use a Gaussian weighting function, again centred on the base line, which accepted all other lines but which weighted their entry with the value of a Gaussian computed at their mid-point. Alternatively, a more sophisticated definition could be used, based on some notion of an object "part" [46]. For example, Kalvin et al. [52] have proposed splitting shapes at concavities and treating the sections between them separately. However, despite the potential problems listed above, it was found that the simple circular function performed adequately.

Conflicting factors must be taken into account when determining the optimal size of the local region. While reducing the size of the region lessens the likelihood of the representation being affected by shape variation, it does so at the cost of a decrease in its strength. The optimal region size is therefore a trade-off between the strength and robustness of the representation. As such it should be determined by the difficulty of the recognition task and the likelihood of scene clutter. The size of the region should also be related to the scale of the shapes to be represented. Assuming that this scale is fixed, and that objects are of a similar size, the size of the region can be expressed as some fraction of the maximum distance between any two lines within a shape, (cf. [13]).

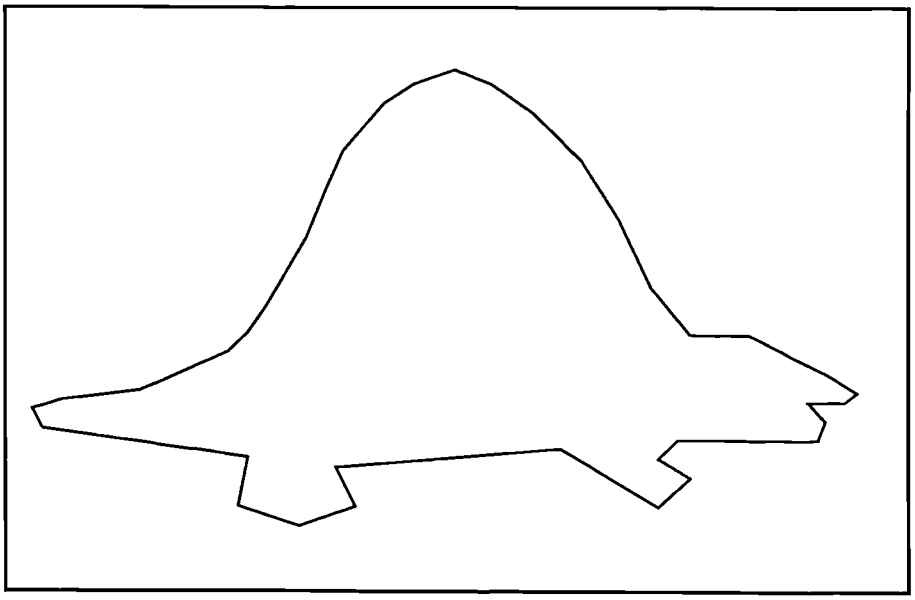

Figure 2–21:

Saliency

Despite the use of high-level primitives, complex shapes may still require a large number of histograms to be stored, placing strains both on the amount of memory needed to store object models and on the speed of recognition. One solution to this problem would be to rank primitives on the basis of some measure of *saliency*. Consideration could then be restricted to a subset of the most salient primitives, (cf. [13]). This would serve to reduce both the number of histograms that need be stored and the amount of computation involved in performing recognition.

The exact meaning of saliency is not clear, and obviously depends on the nature of the primitives being used. Most definitions of saliency attempt to capture the importance or distinctiveness of a primitive within a shape. However, the overriding constraint is that it should provide a reliable ordering of primitives, otherwise the subset of primitives considered may vary between images, leading to instability in recognition.

In the case of line segments, a possible measure of saliency that has been proposed is their length, [3]. This solution is discounted on two counts. Firstly, the ordering of lines based on their length is liable to break down in noisy conditions where lines become fragmented. Secondly, the spatial distribution of long lines within a shape may be uneven. For example, in the line description of the shape shown in figure 2–21 long lines are restricted to a particular region of the shape, since the description of the curved section requires many short lines. The obvious danger in this situation is that occlusion of this area of the shape will cause a break down in recognition.

2.6 Uniqueness

This section discusses the factors affecting the uniqueness of representations of shape in the form of geometric feature distributions.

A representation can be said to be *unique* if the set of possible shapes that could have produced it differ only by the allowable object transformations, as determined by the invariance characteristics of the representational scheme. Within the statistical pattern classification approach to recognition, the uniqueness of a representational scheme ensures that the recognition system will be able to distinguish between all dissimilar shapes. The factors affecting the uniqueness of the proposed representational scheme are now discussed.

Feature Set Completeness

The uniqueness of the present representational scheme is determined primarily by the *completeness* of the set of geometric features used to define the relationship between two primitives. As stated above, the completeness of a feature set describes the degree to which its values combine to characterise the geometric relationship between a pair of shape primitives. The values of a complete feature set are therefore sufficient, given details of one primitive, to unambiguously recreate the second. It seems clear that the distribution of the values from a complete feature set provides a unique representation of shape, (although non-complete feature sets may also generate unique representations in certain circumstances). Since a minimum of 5 geometric feature values are needed to uniquely define a geometric relationship, [16], in practice it is often the case that the chosen feature set will be incomplete. The proposed feature set, since it returns only 3 values, represents just such a set. In this situation there is the possibility that representations may not be unique. It is therefore important to gain an understanding of the range of shapes which generate a common set of feature values.

If the direction vectors of line segments are available then, given the values of the relative angle and perpendicular distance features, ϕ and d_{min}, d_{max} respectively, together with the position of the first line, possible positions of the second line are as shown in figure 2–22. It can be seen that all lines which differ only by a translation in a direction parallel to the base line produce the same set of feature values. Also, all collinear lines produce the same set of values. This latter feature actually proves very useful, since it ensures that all fragments of a line have the same representation as the original line, modulo a scaling factor determined by the proportion of lost data. This provides considerable advantages when attempting to match line fragments, (see Section 3.3.1).

Moreover, the use of the local region to restrict the range over which relationships are measured means that the representation of collinear lines that are far apart, eg. because they belong to different objects, are not identical. This is demonstrated by the two lines, ℓ_1 and ℓ_2, shown in figure 2–23.

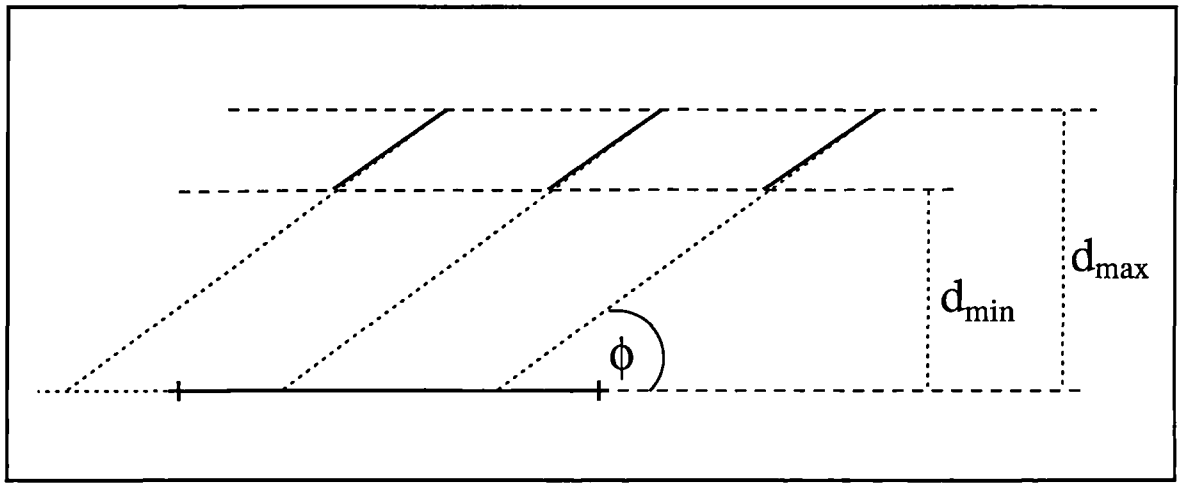

Figure 2–22: Possible interpretations of the geometric feature values.

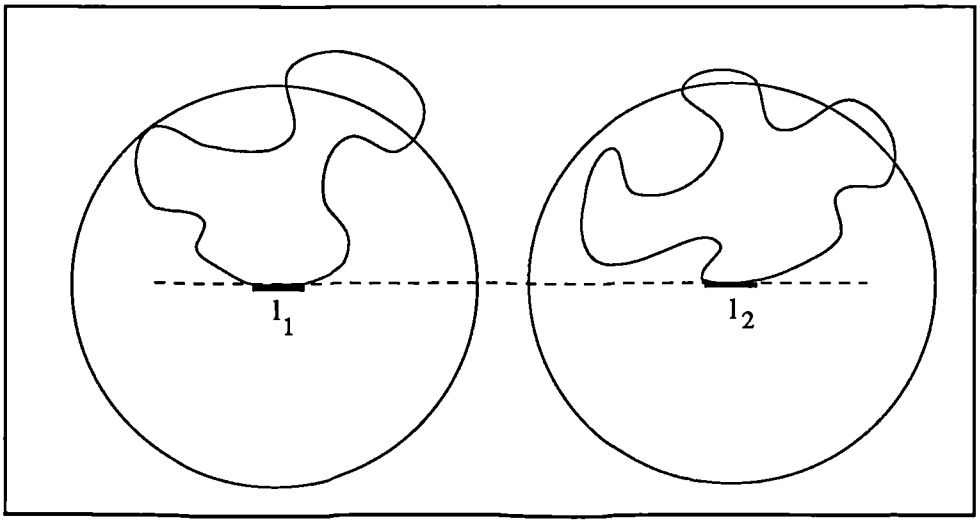

Figure 2–23: Collinear lines with differing representations.

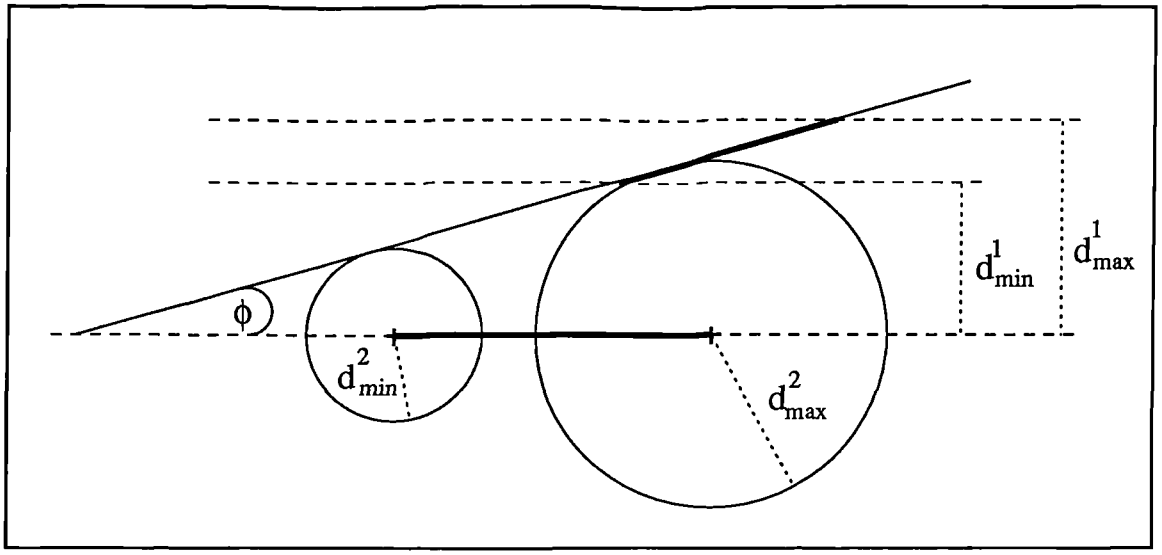

Figure 2–24: A complete feature set.

It is possible to apply the perpendicular distance feature in both directions, to provide 5 feature values. These are sufficient to uniquely describe the geometric relationship between the line segments, as demonstrated in figure 2–24. Recording the distribution of these feature values would require the use of a histogram with 3 axes, which can be expected to require a large amount of memory.

Binning & Blurring

The process of binning and blurring the values of the geometric features obviously results in the loss of exact geometric information. Therefore, in practice, even in the case where a complete geometric feature set is used, representations are not unique. However, the set of shapes generating a common representation differ only by a predefined limit based on the acceptable variation between shapes. Indeed, these factors can be used to determine the scale at which shapes are represented in the histogram. For example, the use of a small blur, with a correspondingly large number of bins, produces representations that are able to support fine scale discrimination between shapes, while the use of a large blur and a small number of bins provides stable representations which capture the large scale similarities between shapes. Thus, varying both the width of blur and, correspondingly, the number of bins used in the histogram provides a flexible way in which the acceptable difference between two shapes can be controlled.

2.7 Discussion and Summary

This chapter has described a scheme for representing shape which is based upon recording the distribution of geometric features between its primitive elements.

It was argued that there are significant advantages in basing representation on a line-based description of shape. Firstly, line segments can be obtained straightforwardly from a linear approximation of the edgel strings extracted by a Canny edge detector, with favourable consequences for the accessibility of the representational scheme. Secondly, the fact that any shape can be described to an arbitrary degree of accuracy using a sufficient number of straight line segments means that the scheme is versatile in its application. Finally, since line segments can be either 2D or 3D, their use places no restrictions on the dimensionality of the shape that can be represented.

The set of geometric features used to measure the relationship between pairs of line segments was introduced. These features provide an acceptable balance between the conflicting aims of providing a strong measurement of local shape while ensuring that representations based upon their values are not critically affected by line fragmentation. The chosen features also possess the required invariance to transformations in the 2D position and orientation of an object in the scene.

The structure of the histogram used to record the distribution of geometric feature values was presented, along with the method of recording individual geometric rela-

tionships. The flexibility of the scheme was highlighted by examining the different levels at which shape could be represented. It was shown that by simply varying the set of pairwise geometric relationships recorded within a single histogram it is possible to produce either local or global representations of shape. These can be used to match individual shape primitives or whole shapes respectively.

The strategy of blurring entries in the histogram was shown to provide a flexible method for encoding in the representation the allowable differences between shapes. This can be used to provide a certain degree of robustness to small variations in the position and orientation of line segments within a shape arising from image noise or object transformation. It was argued that by varying both the resolution and width of blur used in the histogram it is possible to alter, in a flexible way, the scale at which shape is represented.

One of the most important properties of the proposed representational scheme is its robustness to changes in the shape description extracted from an image caused by fragmentation noise or occlusion. In fact, the representation is robust in a number of ways. Firstly, the fact that recording geometric feature distributions involves making multiple local measurements of shape suggests that the representation should degrade gracefully as the shape is degraded. Particular care was taken to ensure that representations constructed from a line-based shape description retained this property. This is important since it means that the practical advantages of basing representation on line segments are combined with the robustness that comes from considering low level edgels. Secondly, the robustness of the scheme was improved by restricting the range over which geometric relationships are measured, thereby reducing the likelihood of the representation being affected by shape variation. Finally, the fact that local shape representations are composed of multiple histograms, one for each shape primitive, means that recognition based on the representation will be robust to the loss of data through fragmentation noise or occlusion.

Chapter 3

2D OBJECT RECOGNITION

3.1 Introduction

This chapter presents the application of the representational scheme based on geometric feature distributions, (GFD's) to the the problem of 2D object recognition. The chapter is organised into the following sections:

1. **Matching geometric feature distributions**
   The method of matching geometric feature distributions is presented. This includes a discussion on the appropriate form of similarity metric to be used within a matching scheme based on nearest-neighbour classification. The ability of this scheme to establish valid matches between local shape primitives is briefly demonstrated and the factors affecting the uniqueness of the representation are assessed.

2. **Dealing with variable line description**
   The performance of the proposed scheme is examined under measurable conditions of shape variation, including fragmentation noise, scene clutter and sensor error. The scheme is shown, both theoretically and empirically, to possess considerable robustness to these forms of variation.

3. **Determining object pose**
   The use of the generalised Hough transform to determine object pose, based on matches established using the GFD scheme, is presented.

4. **Shape matching**
   The ability of the GFD scheme to support the recognition of complete shapes through the matching of global shape representations is demonstrated.

3.2 Matching Geometric Feature Distributions.

The basic element in performing 2D object recognition is the ability to match the description of shape extracted from an image to that of the set of object models. One of the primary motivations behind the study of geometric feature distributions was to develop a strong, robust and invariant form of shape representation with properties such that this matching could be performed using techniques from statistical pattern classification. This section describes the appropriate form of similarity metric to be used within a matching scheme based on nearest-neighbour classification.

3.2.1 Computing A Similarity Metric

Essential to the pattern classification approach is the availability of some form of similarity metric, D, that can be used to provide a quantitative measure of the degree of similarity between two shape representations. In the present scheme this involves treating values in the histogram recording the distribution of geometric features within a shape as the components of a feature vector. The chosen metric should ideally meet the following requirements:

- It should provide a measure of the similarity between two shapes, based on the information available in their representations.
- It should be robust to changes in the representations caused by variations in shape description, eg. through fragmentation noise, sensor error or scene clutter.

The physical interpretation of this metric depends on the level of representation adopted; if histograms are a record of global feature distributions then D indicates the degree of similarity between *whole* shapes, while for local feature distributions, D provides a measure of similarity between individual shape primitives. One of the advantages of the proposed recognition scheme is that both levels of representation can be matched using a common metric.

We require a metric, D, that can be applied to histograms representing the two shape primitives q_i and m_j, where q_i is drawn from the set of image primitives, $\mathbf{I}$ and m_j is drawn from the set of model primitives, M. The chosen similarity measure is the *Bhattacharrya distance*, (for a detailed discussion of the advantages of this metric see Mardia et al. [64], page 378). It is interesting to note that this metric can be related to the standard χ^2 statistic used in determining the "goodness-of-fit" between two frequency distributions, (see *Appendix*). For the two histograms, H_{q_i} and H_{m_j}, representing q_i amd m_j respectively, the value of D is given by

$$D(q_i, m_j) = \sum_{x}^{n_\theta} \sum_{y}^{n_d} \sqrt{H_{q_i}(x,y).H_{m_j}(x,y)}$$

Where n_θ and n_d represent the number of bins used along the relative angle and perpendicular distance axes respectively. In order to guarantee that this metric is independent of the "length" of the histograms, which is related to the length of m_j and the total length of lines within the shape, it is necessary to perform some form of normalisation. The chosen form of normalisation is to ensure that

$$\sum_{x}^{n_\theta} \sum_{y}^{n_d} H'_{m_i}(x,y)^2 = 1$$

where

$$H'_{m_j}(x,y) = \sqrt{H_{m_j}(x,y)}$$

Thus, the value, H''_{m_j}, used in computing the metric D, is given by

$$H''_{m_j}(x,y) = \frac{H'_{m_j}(x,y)}{L}$$

where

$$L = \sqrt{\sum_{x}^{n_\theta} \sum_{y}^{n_d} H_{m_j}(x,y)}$$

If H_{q_i} is similarly normalised then D is given by,

$$D(q_i, m_j) = \sum_{x}^{n_\theta} \sum_{y}^{n_d} H''_{q_i}(x,y).H''_{m_j}(x,y)$$

A consequence of the normalisation step is that the histograms representing each primitive are constrained to lie on the unit hypersphere in $m \times n$-dimensional space. The value returned by D therefore has a straightforward geometric interpretation, in that it gives the cosine of the angle, β, between the feature vectors describing each primitive, figure 3–1.

Thus, in the case where the model and image primitives match exactly, D equals unity. The proposed metric therefore meets the first of the conditions listed above, in that it returns a principled measure of the similarity between the two representations.

However, in situations where the shape description extracted from the image is depleted by fragmentation noise or augmented by spurious elements, the behaviour of the metric

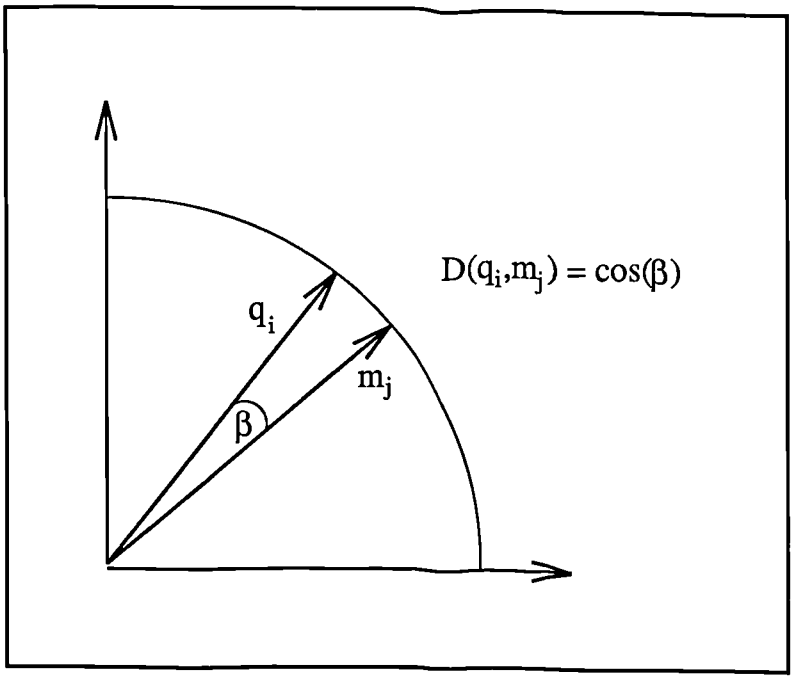

Figure 3–1: Geometric interpretation of D

becomes more intuitive if the histogram representing q_i is not normalised. Obviously, this also has benefits in terms of the amount of computation required in computing D. If H'_{q_i} is not normalised then D is given by,

$$D(q_i, m_j) = \sum_{x}^{n_\theta} \sum_{y}^{n_d} H'_{q_i}(x,y) H''_{m_j}(x,y)$$

In the case where the two primitives match, ie. $H_{q_i}(x,y) = H_{m_j}(x,y)$, we have that

$$D(q_i, m_j) = \sqrt{\sum_{x}^{n_\theta} \sum_{y}^{n_d} H_{q_i}(x,y)}$$

From Chapter 2 we have that

$$\sum_{x}^{n_\theta} \sum_{y}^{n_d} H_{q_i}(x,y) = |q_i| . |\mathbf{Q}|$$

Therefore, in the case where two primitives match exactly, D returns a value related to the product of the length of q_i and the total length of lines within the shape. In general, D returns a value related to the proportion of lines within each shape description which match, based on the distribution of geometric features computed between these lines. The suitability of this revised metric for use within a nearest-neighbour classification system is established in the following section, while its robustness in conditions where the image shape description is affected by various forms of image noise is assessed in Section 3.3.

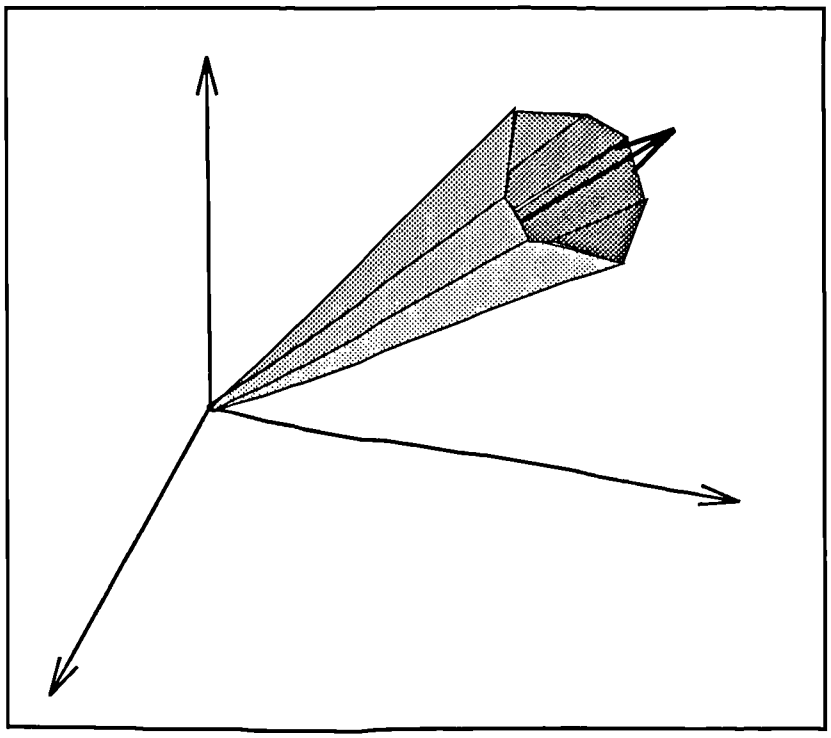

Figure 3–2: A *Voronoi cone* in 3D space.

3.2.2 Nearest-Neighbour Classification

If $\mathbf{M} \equiv \{m_1, m_2, ...m_n\}$ is the set of model primitives and $m^* \in \mathbf{M}$ is the model primitive whose histogram representation is nearest to that of a particular image primitive q_i, then the nearest-neighbour classification rule dictates that q_i should be matched to m^*. Determining the identity of m^* involves finding the maximum in the set $\mathbf{C}$, where

$$\mathbf{C} \equiv \{D_{ij} : D_{ij} = D(q_i, m_j) \quad m_j \in \mathbf{M}\}$$

This explanation will be aided by considering a geometric interpretation of the classification process. A consequence of the normalisation is that the feature vectors representing the set of model primitives each lie on the unit hypersphere in $m \times n$-dimensional space. The value of D computed between the feature vector representing a model primitive m_j and that representing an image primitive q_i is therefore a measure of the projection of q_i in the direction of m_j. A result of the nearest-neighbour classification based on the value of D is that the feature vectors representing the set of model primitives tessellate the feature space into a series of *Voronoi cones*. The *Voronoi cone* associated with the model primitive, m_j, defines a volume of the feature space in which the projection of q_i in the direction of m_j is greater than for any other $m \in \mathbf{M}$. The feature vector representing m_j will lie somewhere in the cone, while the sides of the cone represent decision planes with nearby model primitives. Therefore, an image primitive whose feature vector lies within the *Voronoi cone* associated with a particular model primitive is matched to that primitive. A *Voronoi cone* in 3D space is shown in figure 3–2, while multiple cones in 2D space are shown in figure 3–3. The intersection

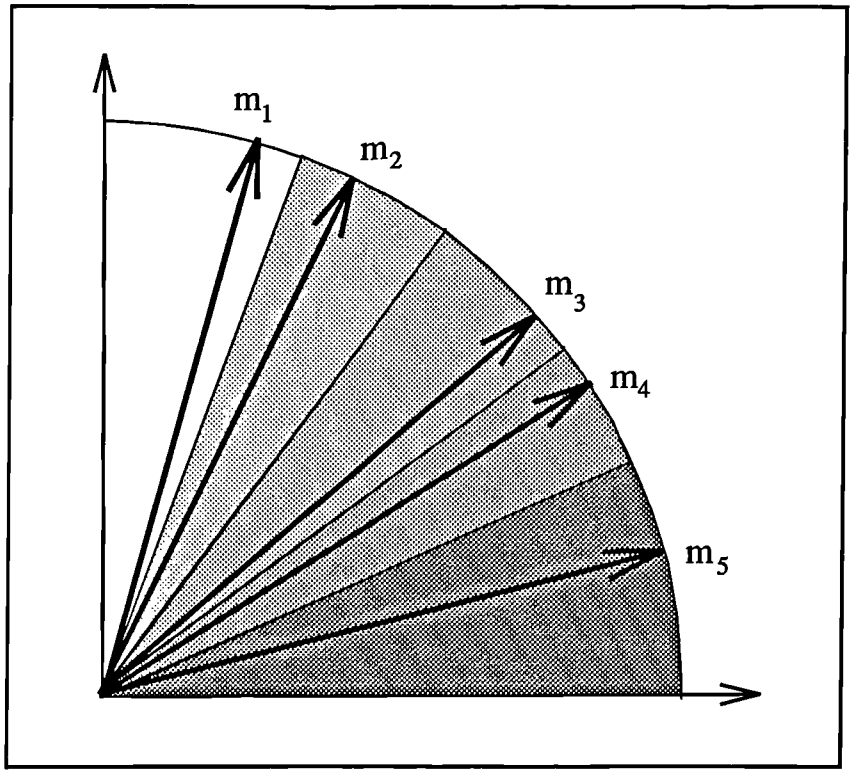

Figure 3–3: A series of *Voronoi Cones* in 2D space.

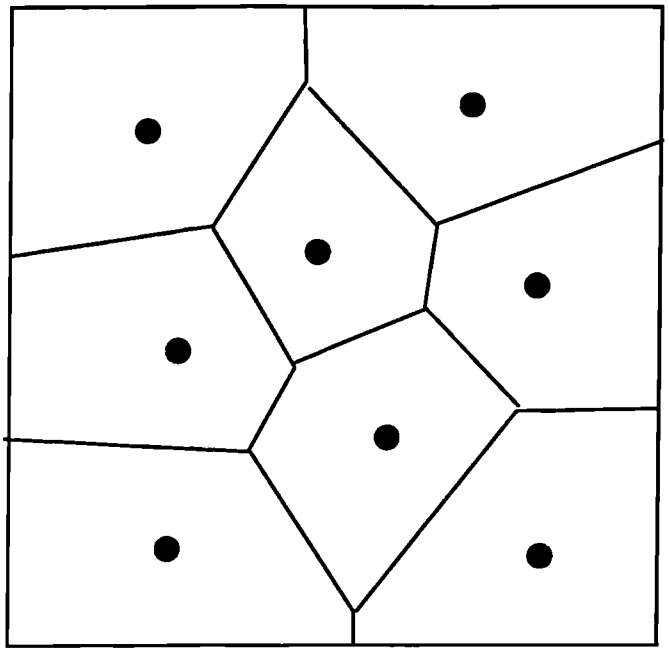

Figure 3–4: A *Voronoi* tessellation.

of these cones with any centred hypersphere in the feature space produces a Voronoi tessellation of its surface, figure 3–4.

In terms of matching local shape primitives, nearest-neighbour classification can be seen as implementing the constraint that an image primitive may only match a single model primitive. This is entirely appropriate, since it is only in a very small number of

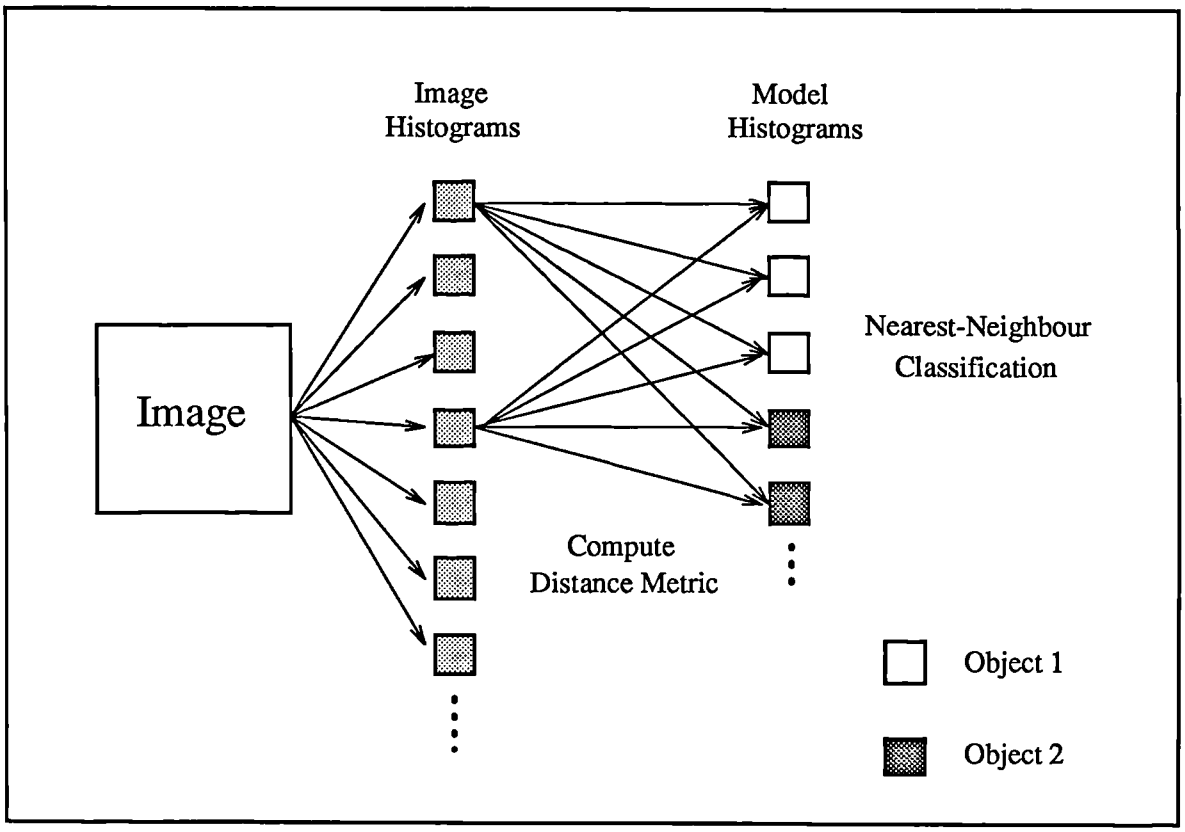

Figure 3–5: A practical scheme for performing recognition.

situations that an image primitive may legitimately match multiple model primitives.[1] Another property of the proposed scheme is that a model line may be matched to more than one image line. This is important in handling the recognition of multiple instances of an object in a scene and in the matching of multiple fragments of a line segment. Furthermore, not all model lines must be matched. This is important if the system is to be robust to missing data caused by shape fragmentation or occlusion, and is a problem that has to be explicitly catered for in certain recognition systems, eg. tree-search [40].

The classification process is performed for all $q \in \mathbf{I}$, such that each primitive is matched to a single model primitive. The structure of a practical scheme for performing this matching is shown in figure 3–5. Assuming no saliency measure is used to pre-select shape primitives, the number of computations involved in this scheme is given by,

$$\text{Number of Computations} = I \times M \times n_\theta \times n_d$$

where I and M denote the number of image and model line segments respectively. This obviously represents a lot of computation, and on conventional serial processors recog-

[1] A possible instance is where the model is described by multiple noisy line fragments or where there is a decrease in linear approximation accuracy between the acquisition of model and image shape descriptions.

nition will be relatively slow. However, processing is uniform, local and involves only simple array multiplication. This suggests that it should lend itself straightforwardly to implementation in parallel hardware. Indeed, further research has recently begun on developing methods for achieving this, [101].

3.2.3 Demonstration of Matching

The ability of the proposed scheme to correctly match line segments is demonstrated using the shape shown in figure 3–6. The elements of the set **C** for each image primitive can be conveniently expressed as a *correspondence array*, (cf. [15]), in which rows represent model primitives and columns represent image primitives. Element i, j of this array represents the value of D computed between model primitive m_i and image primitive q_j.

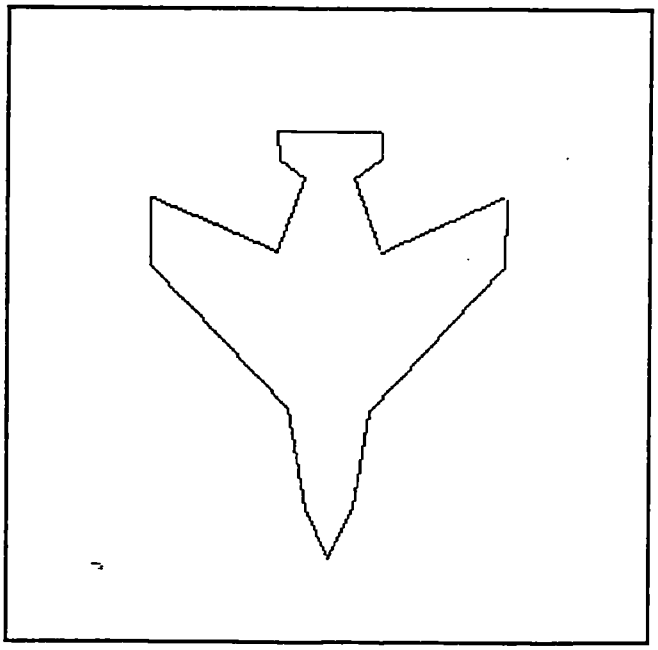

Figure 3–6: The shape, **A0**, used to demonstrate matching.

The correspondence array for the line segments describing the above shape matched against themselves is shown in table 3–1. The parameters of the histogram used were $n_\theta = 40$, $n_d = 30$, $\sigma_\theta = \sigma_d = 1.0$. No local region was used. In order to make the outcome of recognition clearer, the values of D within a column are normalised relative to D^*, the value of D for the nearest model primitive m^*. It can be seen that peak values do indeed lie along the diagonal, indicating that each line segment has been correctly matched with itself. Moreover, it can be seen that there is a significant separation between the values of D for correct and incorrect associations.

This can be further appreciated by examining a *correspondence image*, in which the intensity of a pixel is proportional to its value in the correspondence array, figure 3–7.

1.00	0.20	0.13	0.11	0.14	0.09	0.29	0.22	0.11	0.15	0.04	0.07	0.15	0.12	0.06	0.18	0.19
0.20	**1.00**	0.11	0.16	0.24	0.36	0.18	0.29	0.30	0.13	0.09	0.21	0.28	0.17	0.08	0.20	0.14
0.13	0.11	**1.00**	0.17	0.16	0.11	0.36	0.09	0.10	0.14	0.09	0.14	0.19	0.20	0.11	0.09	0.28
0.11	0.16	0.17	**1.00**	0.38	0.17	0.23	0.13	0.08	0.16	0.12	0.13	0.11	0.12	0.08	0.21	0.12
0.14	0.24	0.16	0.38	**1.00**	0.16	0.15	0.11	0.13	0.22	0.08	0.09	0.10	0.15	0.14	0.17	0.08
0.09	0.36	0.11	0.17	0.16	**1.00**	0.11	0.14	0.27	0.09	0.12	0.26	0.18	0.14	0.09	0.12	0.11
0.29	0.18	0.36	0.23	0.15	0.11	**1.00**	0.18	0.15	0.19	0.09	0.17	0.29	0.21	0.11	0.14	0.30
0.22	0.29	0.09	0.13	0.11	0.14	0.18	**1.00**	0.19	0.18	0.08	0.13	0.16	0.06	0.06	0.16	0.11
0.11	0.30	0.10	0.08	0.13	0.27	0.15	0.19	**1.00**	0.14	0.07	0.18	0.14	0.20	0.13	0.16	0.26
0.15	0.13	0.14	0.16	0.22	0.09	0.19	0.18	0.14	**1.00**	0.11	0.06	0.16	0.06	0.14	0.42	0.16
0.04	0.09	0.09	0.12	0.08	0.12	0.09	0.08	0.07	0.11	**1.00**	0.06	0.10	0.08	0.24	0.14	0.14
0.07	0.21	0.14	0.13	0.09	0.26	0.17	0.13	0.18	0.06	0.06	**1.00**	0.21	0.17	0.07	0.06	0.20
0.15	0.28	0.19	0.11	0.10	0.18	0.29	0.16	0.14	0.16	0.10	0.21	**1.00**	0.18	0.10	0.15	0.15
0.12	0.17	0.20	0.12	0.15	0.14	0.21	0.06	0.20	0.06	0.08	0.17	0.18	**1.00**	0.05	0.06	0.19
0.06	0.08	0.11	0.08	0.14	0.09	0.11	0.06	0.13	0.14	0.24	0.07	0.10	0.05	**1.00**	0.11	0.08
0.18	0.20	0.09	0.21	0.17	0.12	0.14	0.16	0.16	0.42	0.14	0.06	0.15	0.06	0.11	**1.00**	0.13
0.19	0.14	0.28	0.12	0.08	0.11	0.30	0.11	0.26	0.16	0.14	0.20	0.15	0.19	0.08	0.13	**1.00**

Table 3–1: The correspondence array for **A0**.

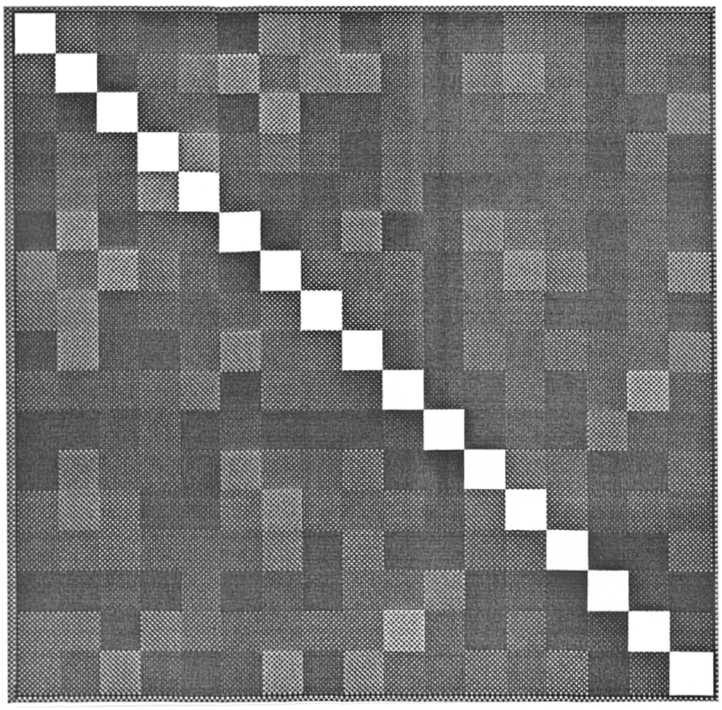

Figure 3–7: The correspondence image for **A0**

The correspondence array and correspondence image will prove useful in assessing the performance of the recognition scheme as the image shape description is varied. As a preliminary example, the invariance of geometric feature distributions to 2D rotation is demonstrated. Figure 3–8 shows the lines extracted from of an image of **A0** after

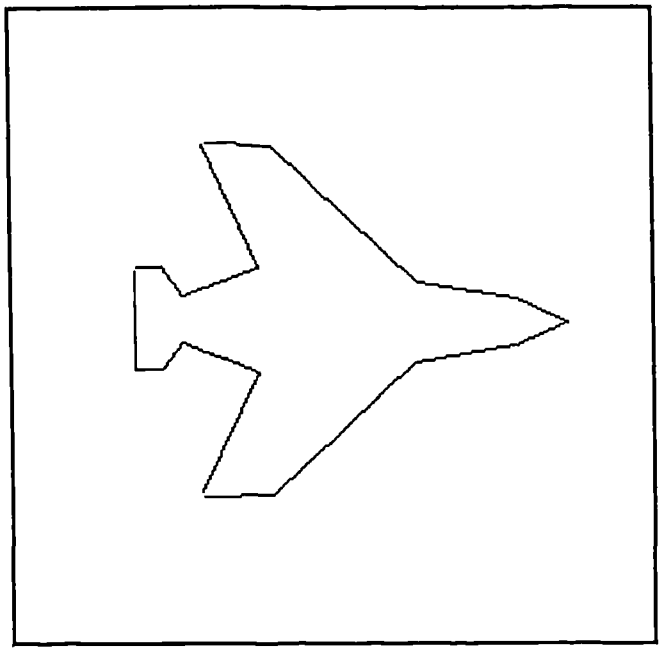

Figure 3–8: The shape, **A0**, rotated through 90°.

it has been rotated through 90°. Table 3–2 and figure 3–9 show, respectively, the correspondence array and correspondence image computed for the lines in this shape.

0.11	0.17	0.05	0.07	0.15	0.12	0.08	0.18	0.20	**1.00**	0.20	0.14	0.11	0.14	0.09	0.29	0.22
0.30	0.15	0.09	0.23	0.28	0.17	0.08	0.20	0.15	0.19	**1.00**	0.11	0.16	0.25	0.37	0.18	0.30
0.10	0.14	0.10	0.13	0.19	0.19	0.13	0.09	0.29	0.13	0.12	**1.00**	0.18	0.19	0.11	0.35	0.09
0.08	0.16	0.12	0.12	0.10	0.12	0.11	0.21	0.12	0.11	0.15	0.16	**1.00**	0.41	0.18	0.23	0.14
0.13	0.25	0.07	0.08	0.12	0.15	0.13	0.16	0.08	0.15	0.24	0.16	0.40	**1.00**	0.17	0.15	0.10
0.28	0.09	0.12	0.31	0.17	0.14	0.10	0.12	0.11	0.09	0.38	0.11	0.17	0.17	**1.00**	0.11	0.14
0.15	0.22	0.09	0.17	0.29	0.20	0.10	0.15	0.31	0.29	0.18	0.38	0.25	0.16	0.11	**1.00**	0.21
0.19	0.18	0.08	0.13	0.16	0.07	0.05	0.16	0.11	0.22	0.30	0.09	0.15	0.12	0.14	0.18	**1.00**
1.00	0.14	0.07	0.19	0.14	0.21	0.16	0.17	0.26	0.11	0.31	0.10	0.08	0.12	0.29	0.15	0.21
0.14	**1.00**	0.10	0.06	0.17	0.06	0.15	0.45	0.17	0.16	0.13	0.14	0.18	0.22	0.09	0.19	0.19
0.07	0.12	**1.00**	0.07	0.10	0.09	0.21	0.15	0.14	0.04	0.10	0.09	0.13	0.09	0.13	0.09	0.07
0.19	0.06	0.07	**1.00**	0.21	0.17	0.09	0.06	0.22	0.07	0.21	0.13	0.16	0.11	0.21	0.18	0.13
0.14	0.16	0.11	0.22	**1.00**	0.18	0.12	0.15	0.16	0.16	0.29	0.19	0.12	0.11	0.16	0.30	0.17
0.20	0.07	0.08	0.16	0.19	**1.00**	0.07	0.06	0.20	0.12	0.18	0.22	0.09	0.15	0.13	0.21	0.07
0.13	0.16	0.26	0.07	0.11	0.05	**1.00**	0.11	0.08	0.06	0.08	0.11	0.07	0.15	0.09	0.11	0.06
0.17	0.45	0.14	0.06	0.14	0.06	0.13	**1.00**	0.13	0.18	0.21	0.08	0.26	0.17	0.13	0.14	0.18
0.28	0.18	0.14	0.20	0.15	0.20	0.08	0.13	**1.00**	0.19	0.15	0.28	0.13	0.08	0.10	0.30	0.12

Table 3–2: The correspondence array for **A0** rotated through 90°.

The displacement of the peak is due to the fact that the ordering of model and image

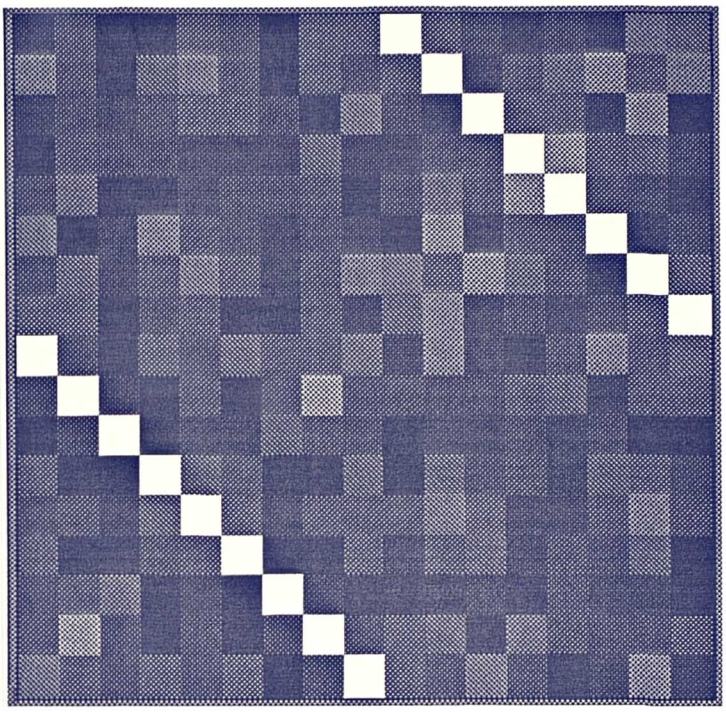

Figure 3–9: The correspondence image for **A0** rotated through 90°.

line segments is no longer preserved. Also, the slight variations in the position and orientations of line segments extracted from the image means that there is a small decrease in the separation between correct and incorrect matches.

A graphical illustration of the correctness of the established correspondences between line segments can be obtained by displaying matched model and image line segments in the same colour. In figure 3–10(a), each model line has been been assigned a unique colour. In figure 3–10(b) each image line is shaded with the colour of the model line to which it is matched.

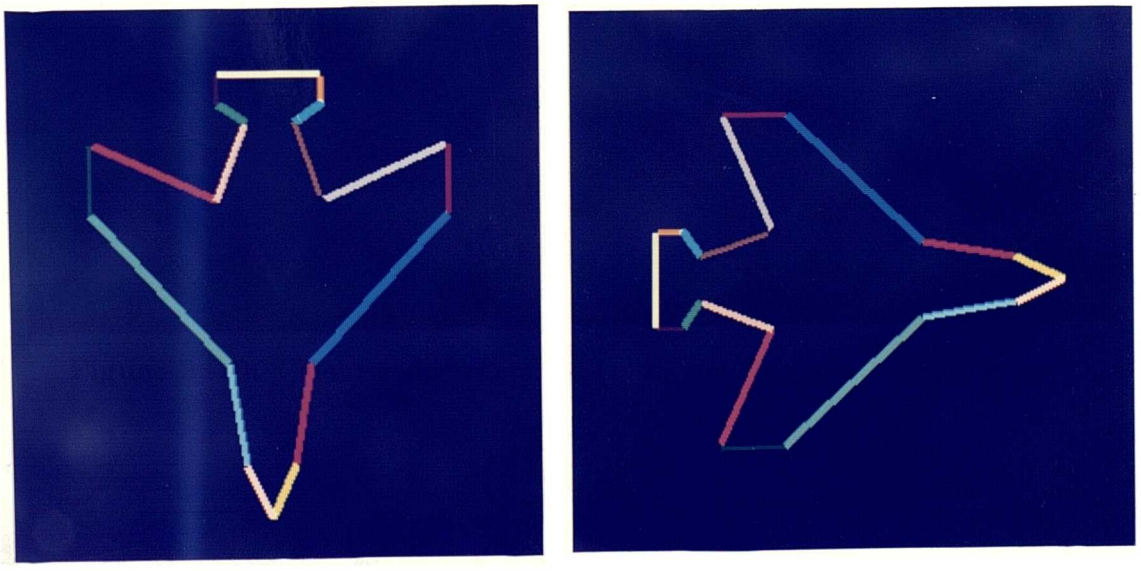

Figure 3–10: Colour-coded matches. (a) model lines and (b) image lines.

3.2.4 Uniqueness

Having described the method by which geometric feature distributions are matched, we are now in a position to be able to assess the uniqueness of the representational scheme and the factors affecting it. It was argued in Section 2.6 that the primary factor in determining the uniqueness of geometric feature distributions was the completeness of the proposed geometric features. However, given a fixed set of geometric features, the uniqueness of the representation is affected by the number of bins and the amount of blurring used in the histogram. The purpose of this section is to examine this relationship by recording the change in the relative values of D computed between the set of model primitives as both the number of bins and the width of blur are varied. This requires a measure of the degree of separation between the values within the correspondence array. The chosen measure is given by

$$S = \frac{\sum_i^M \sum_j^I (1 - D'_{ij})}{I(M-1)} \quad i \neq j$$

where D'_{ij} is the normalised value of D computed between m_i and q_j. While this measure is somewhat arbitrary, it does provide a means by which the affect of histogram resolution and blurring on the uniqueness of representations can be gauged.

Number of Bins

In order to assess the effect of the resolution of the histogram on the uniqueness of the representation, a graph of S against n, the number of bins used on each axis, was plotted. No blurring was used. Figure 3–11 shows the graph of S against n. The higher the value of S, ie. the greater the mean separation between the value of D for the model primitives, the more unique the representation can be said to be.

A geometric interpretation of this result can be attempted. The behaviour of the proposed distance metric is such that the uniqueness of a representational scheme is related to the degree of orthogonality between the feature vectors it produces. In the present scheme, increasing the number of bins used in the histogram increases the dimensionality of the feature space. This will tend to increase the degree of orthogonality between the feature vectors produced.

Width of Blur

In order to examine the effect of the width of blur used in the histogram on the uniqueness of the representation, a graph of S against σ was plotted, where σ describes the width of blur used on each axis. The number of bins used on each axis was fixed at $n_\theta = n_d = 40$. Figure 3–12 shows the graph of S against σ. It can be seen that, as expected, there is a steady decrease in the uniqueness of the representations as the amount of blurring is increased.

Again, a geometric interpretation of this result can be attempted. The effect of blurring entries in the histogram is to move all feature vectors towards the vector in feature space

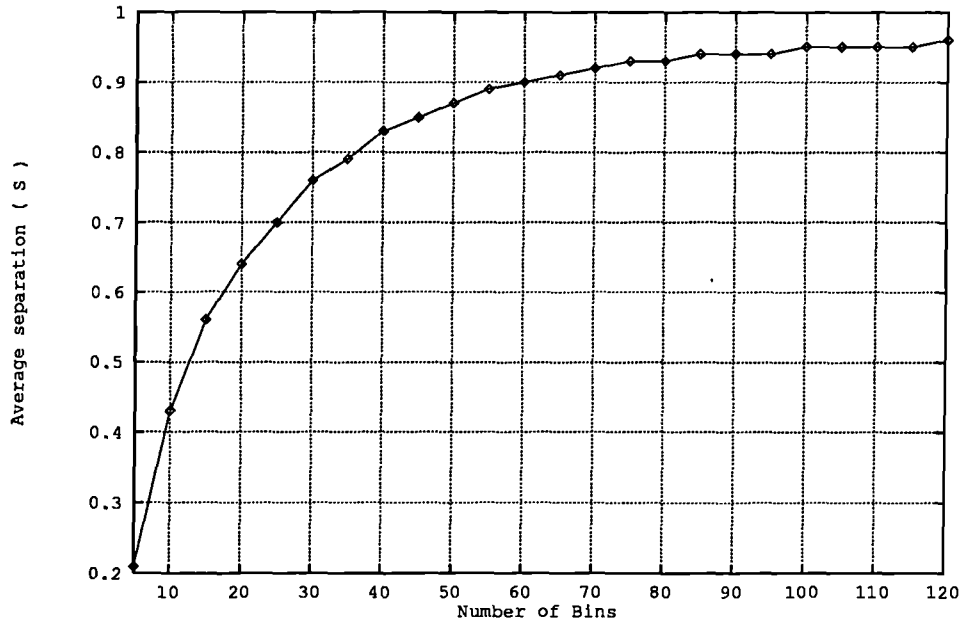

Figure 3–11: A graph showing the relationship between S and the resolution of the histogram.

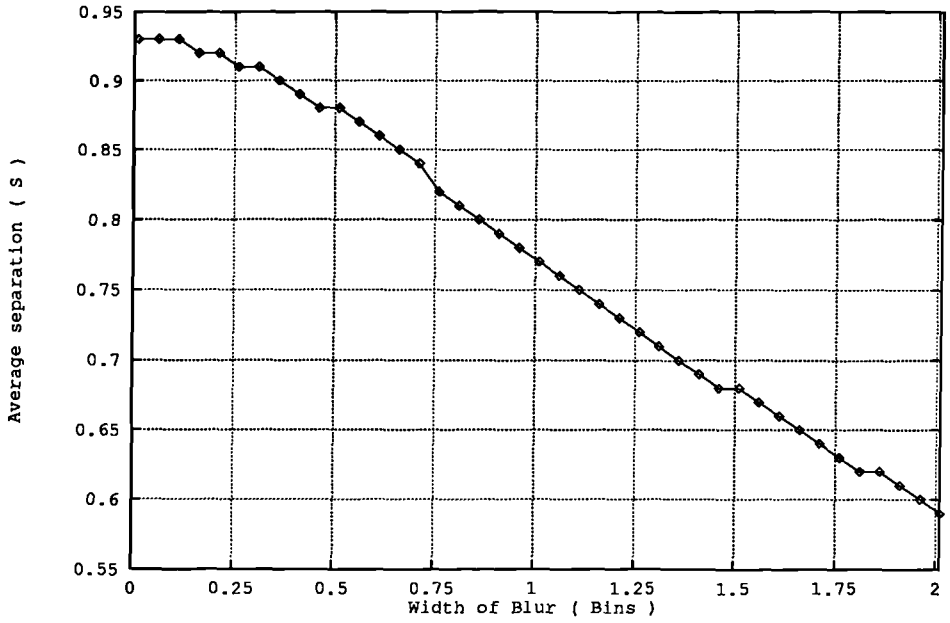

Figure 3–12: A graph showing the relationship between S and the width of blur used in the histogram.

in which all components are equal. The has the effect of compressing feature vectors into a smaller volume of feature space. Thus, the effect of blurring is to reduce the relative *distance* between feature vectors, effectively lessening the uniqueness of the representation. However, in the noise free case, where shape descriptions are constant, the increased blurring has no effect on classification.

This section has described the use of techniques from statistical pattern classification in the matching of local shape primitives represented as geometric feature distributions. The next section examines the extent to which such matching is robust to variations in the line description caused by fragmentation noise, sensor error and scene clutter.

3.3 Dealing With Variable Line Description

Central to the practicality of the proposed recognition system is its ability to perform matching across the full range of potential imaging situations. This requires that matching be preserved in situations where the description of an object's shape extracted from an image differs quite drastically from that obtained during model acquisition. In the vast majority of cases, variations in image shape description will be caused by one or more of the following factors:

- **Shape Fragmentation**
- **Scene Clutter**
- **Sensor Error**

It is important to examine the performance of the recognition scheme under such conditions. One approach to this issue would be to simply demonstrate the performance of the system on a particular subset of example scenes in which such problems occur. However, such an exercise would provide no general, quantifiable information upon which predictions about expected performance in different situations could be made. Consequently, the motivation behind the studies reported in this section is a desire to provide a characterisation of the performance of the proposed algorithm under *measurable* conditions of shape variation. As such it can be seen as an attempt to meet the challenge proposed by Haralick [44], namely that designers of computer vision algorithms should attempt to answer the question,

> "...what is the performance of the algorithm under various kinds of random degradations of the input data?" *Haralick*, [44].

In order to answer this question it is necessary to generate a set of samples which fully cover the expected "input image population", [44]. If the performance of the system is to be quantifiable then ideally each member of this set should have associated with it a parameter describing the amount of shape variation. This involves proposing a model of the effects of each particular form of shape variation. Providing plausible models is not easy in all cases. Also, it is often necessary to propose a simplified model so that the effects of the particular form of shape variation can be isolated. Consequently, the proposed models may be criticised on the grounds that they do not fully describe

the changes that occur in "real" image data. However, provided the results of studies based on the models provide an increased understanding of the likely performance of the system under such conditions then their use is justified.

Each of the following sections considers a separate form of shape variation. Each is structured in the following way: firstly a model of the shape variation is presented, secondly the effect of this variation on geometric feature distributions is assessed, thirdly the effect of the changes in shape representation on the similarity metric D is considered, finally the likely effect of changes in D on the nearest-neighbour classification is presented. Each of these sections includes a theoretical discussion, the validity of which is then assessed experimentally. This ensures that the likely performance of the recognition scheme under each form of shape variation is fully assessed.

3.3.1 Shape Fragmentation

Changes in the lighting of a scene may cause sections of the projected contour of an object to go undetected, as evidenced by a break in the edgel strings returned by the Canny operator. These will obviously have an affect on the line-based shape description. Providing a general account of such changes is difficult, since they depend on a number of factors, including the characteristics of the approximation algorithm, the accuracy with which it is applied, the position at which the fragmentation occurs and whether or not objects are polyhedral. However, if the performance of the proposed recognition scheme in conditions of fragmentation noise are to be assessed then some form of model must be proposed. The following model is based on that detailed in Bray [16].

A Model of Fragmentation Noise

In describing the effects of fragmentation noise it will prove useful to represent a line segment, $\mathbf{s}$, by a single endpoint, $\mathbf{e}$, a unit direction vector, $\mathbf{d}$ and a length ℓ, figure 3–13(a).

$$\mathbf{s} = (\mathbf{e}, (\mathbf{e} + \ell.\mathbf{d})$$

The fragmentation of the line segment $\mathbf{s}$ to produce a single line fragment $\mathbf{s}'$ is then described by the two values, f_1 and f_2, figure 3–13(b), such that

$$\mathbf{s}' = ((\mathbf{e} + f_1.\mathbf{d}), (\mathbf{e} + f_2.\mathbf{d}))$$

The degree of fragmentation, n_f, is given by

$$n_f = \frac{f_2 - f_1}{\ell}$$

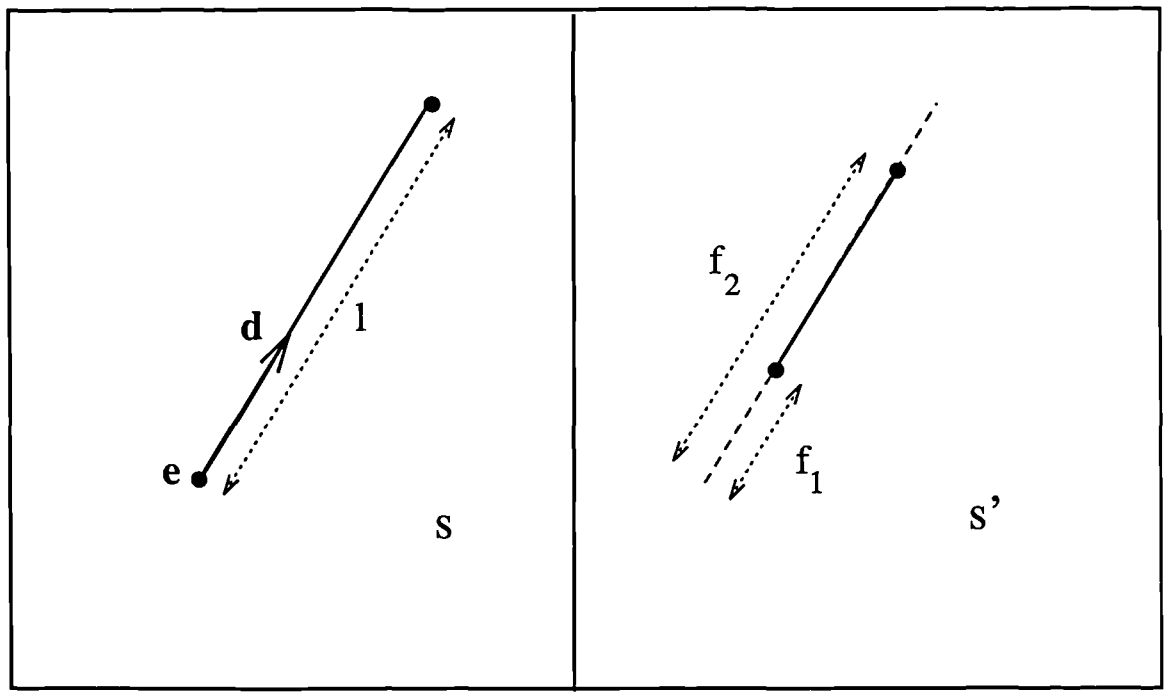

Figure 3–13: (a) line description and (b) a model of line fragmentation

This model makes the assumption that the direction vector of a line is preserved through fragmentation. In the case where the fragmented edgel strings are due to the projection of a polyhedral object this assumption should be valid. However, in cases where objects are non-polyhedral its validity is much less certain, and will depend both on the accuracy of linear approximation and on the position of the fragmentation. Given that the goal in this section is to isolate the effects of data loss, this assumption is justified, (the effects of varying the direction vectors of line segments is addressed in Section 3.3.3).

The effect on geometric feature distributions

Consider the pair of line segments ℓ_p and ℓ_q shown in figure 3–14(a). Also shown is the entry made in the histogram, H_{ℓ_p}, associated with ℓ_p. Both lines are now fragmented to give the two line segments ℓ'_p and ℓ'_q shown in figure 3–14(b). Again, the entries made in $H'_{\ell'_p}$, the histogram associated with ℓ'_p, are shown. It can be seen that, as a consequence of the fact that the entry made in the histogram for a pair of line segments approximates the net effect of considering individual edgels, the set of non-zero entries in $H'_{\ell'_p}$ is is a subset those in H_{ℓ_p}. This is further demonstrated in figure 3–15, which shows the effect of fragmentation on the histogram associated with a particular line in a shape.

We now analyse the effect of fragmentation on the *value* of the entries in the histogram. From Chapter 2 we have that V, the value of the entries in H_{ℓ_p} is equal to

$$V = \ell_p.(\ell_p + \ell_q)$$

The value of the entries in $H'_{\ell'_p}$, denoted by V', is equal to

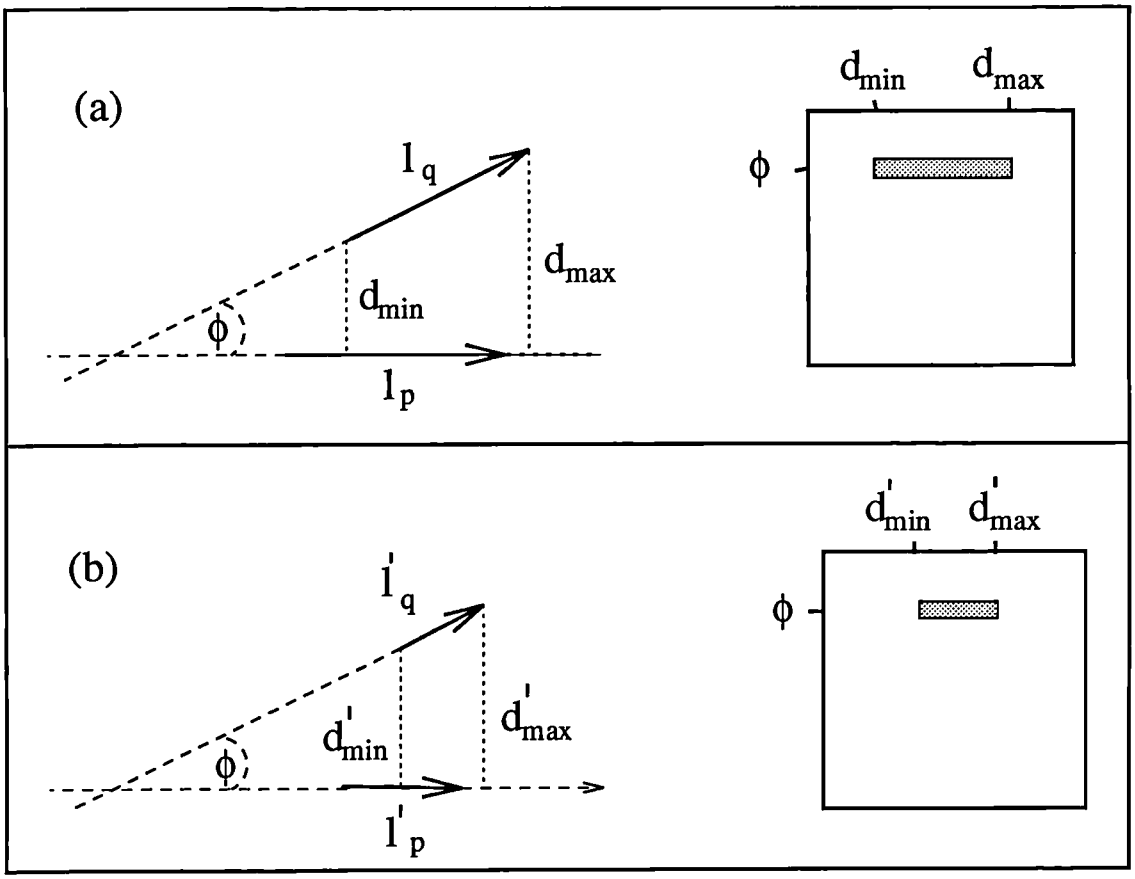

Figure 3–14: (a) a pair of line segments and (b) their fragmented counterparts, along with the associated histograms.

$$V' = \ell'_p.(\ell'_p + \ell'_q)$$

If we make the simplifying assumption that the fragmentation factor, n_f, is the same for both lines, then

$$\ell'_p = n_f.\ell_p \qquad and \qquad \ell'_q = n_f.\ell_q$$

thus

$$V' = n_f^2.\ell_p.(\ell_p + \ell_q)$$

$$V' = n_f^2 V \tag{3.1}$$

This analysis shows that the sum of the values in the histogram representing the fragment of a line segment is scaled by the square of the fraction of data remaining.

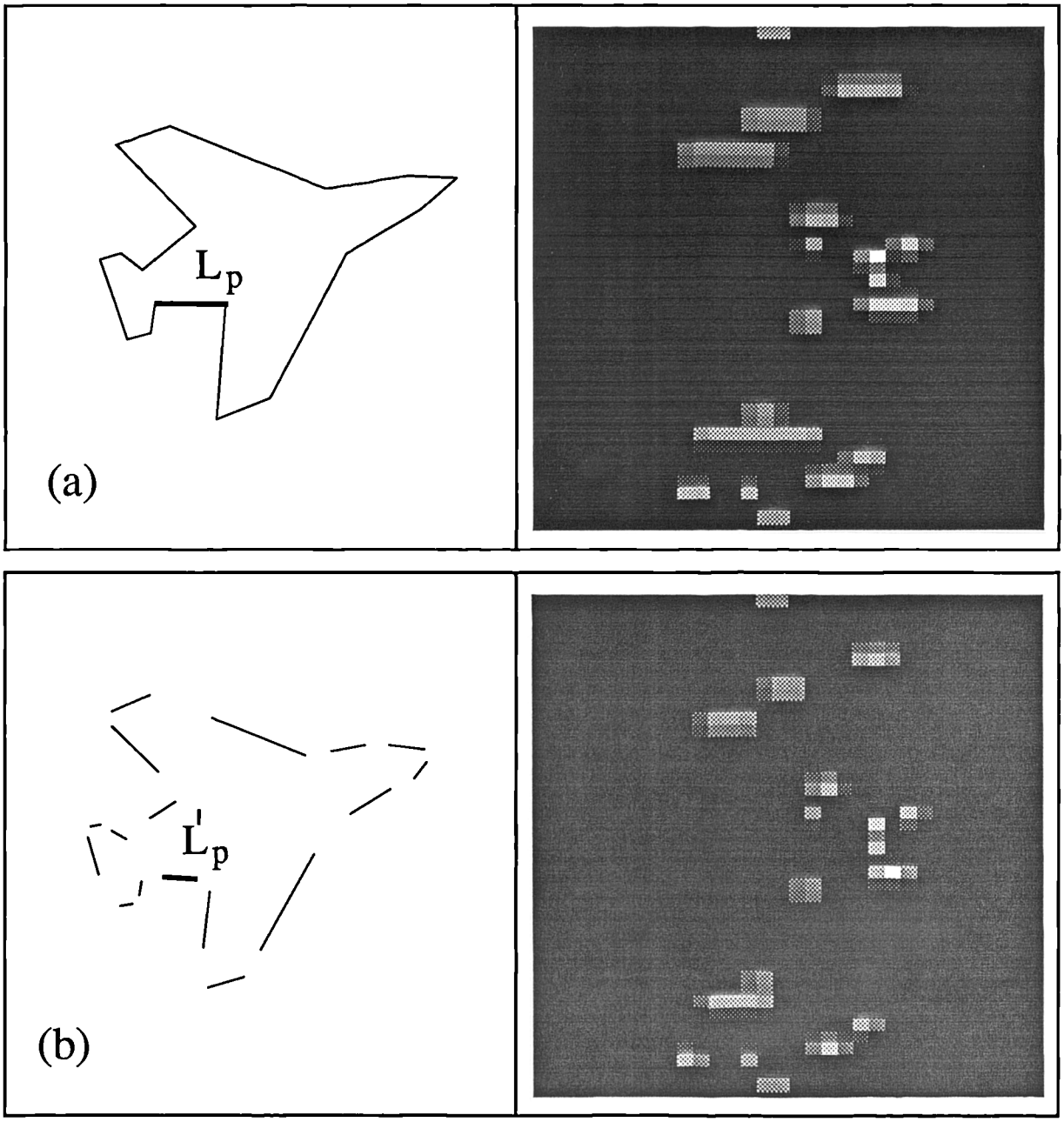

Figure 3–15: (a) the line ℓ_p and histogram H_{ℓ_p} (b) the line ℓ'_p and histogram H'_{ℓ_p}.

The effect on D

We require an expression relating the change in the value of D computed between the histogram representing a line m_i and that of its image counterpart, q_i, to the amount of fragmentation in the image shape description is increased. From Section 3.2.1 we have that

$$D(q_i, m_j) = \sqrt{\sum_{x}^{n_\theta} \sum_{y}^{n_d} H_{q_i}(x,y)} \quad = \sqrt{V}$$

in the case where q_i and m_j match exactly. Therefore, in the case where the image shape description is fragmented to produce the line q'_i the value of D is equal to

$$D(q_i', m_j) = \sqrt{V'} \tag{3.2}$$

Substituting 3.1 in 3.2 we have

$$D(q_i', m_j) = \sqrt{n_f^2 V}$$

which gives

$$D(q_i', m_j) = n_f D(q_i, m_j)$$

Thus, as the shape in an image is fragmented, the value of the similarity metric D computed between a model line, m_j, and the corresponding image line, q_i', falls as the fraction of shape remaining. The validity of this analysis can be demonstrated by examining the graph of D against n_f for a particular line in the shape **A0** as the amount of fragmentation is increased, figure 3–16.

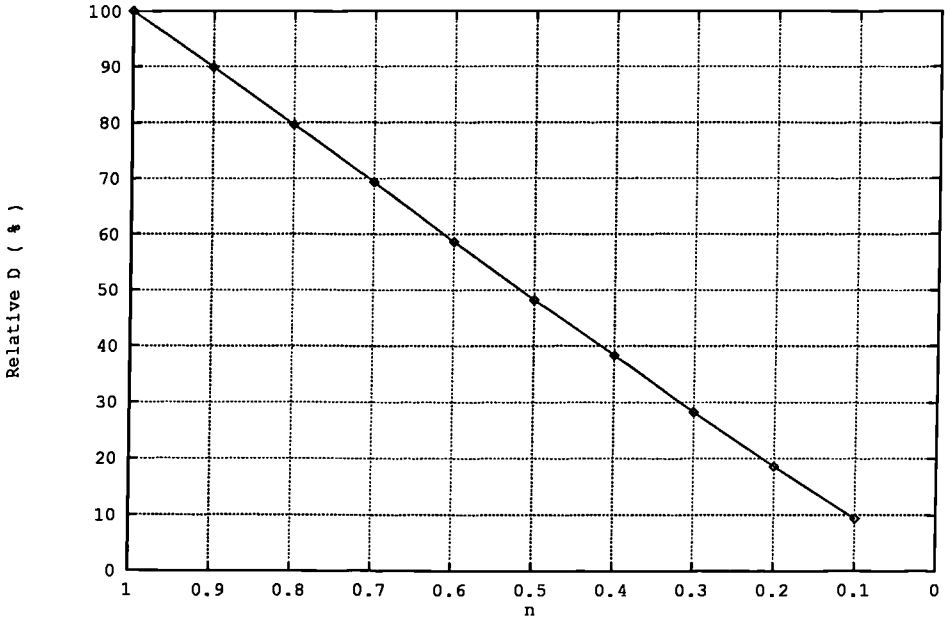

Figure 3–16: A graph of D against n_f.

The effect on nearest-neighbour classification

The above analysis suggests that shape fragmentation should have little effect on the outcome of classification. That this is the case can be confirmed by considering a graph showing the fall in the relative value of D computed between an image line, q_i, and the set of model lines, **M**, as the image shape description is increasingly fragmented, figure 3–17. It can be seen that while the values of D for correct and incorrect model lines converge, correct matching is preserved, theoretically at least, up to very high levels of fragmentation.

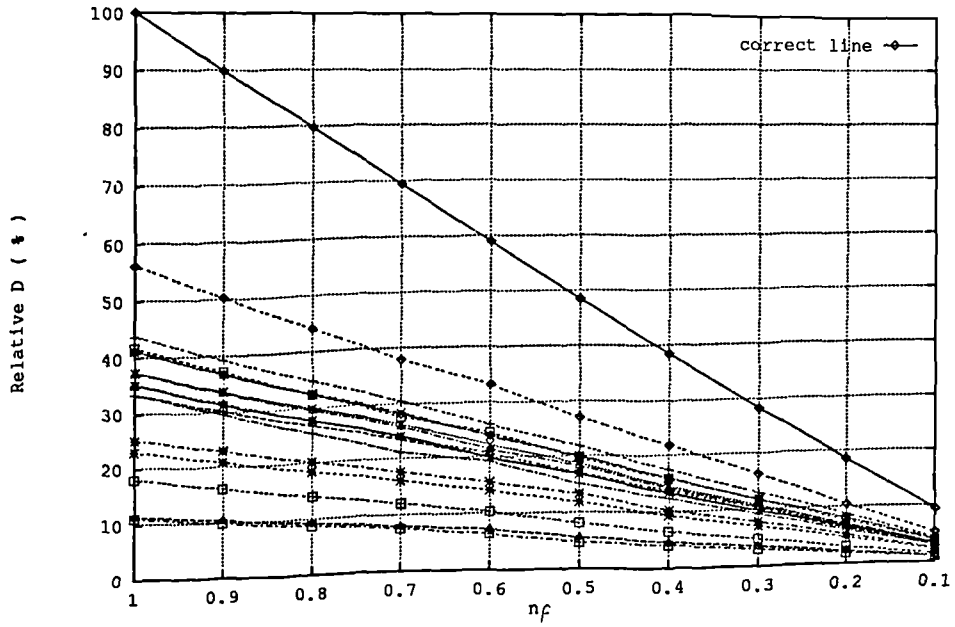

Figure 3–17: A graph of D against n_f for all model lines.

This analysis demonstrates that correct matching is preserved for a single image line, q_i. In order to show that matching is preserved across the whole shape it is necessary to examine the correspondence array at increasing levels of noise. The correspondence array for the shape shown in figure 3–18, for which $n_f = 0.5$, is shown in table 3–3. The correspondence image and colour-coded matches for this shape are shown in figure 3–19 and figure 3–20 respectively.

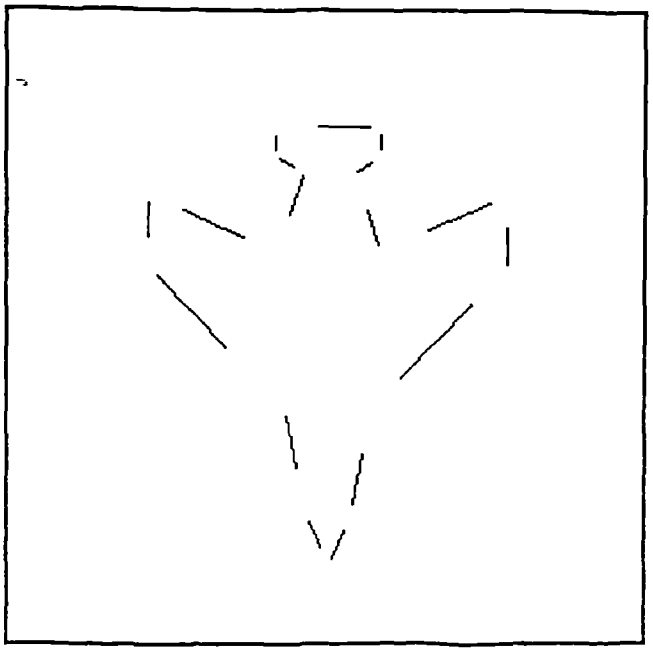

Figure 3–18: A fragmented version of **A0**, at $n_f = 0.5$.

1.00	0.21	0.11	0.09	0.13	0.09	0.30	0.24	0.12	0.18	0.05	0.08	0.13	0.08	0.06	0.20	0.19
0.23	**1.00**	0.12	0.17	0.28	0.39	0.18	0.21	0.31	0.14	0.10	0.21	0.29	0.15	0.10	0.23	0.16
0.14	0.12	**1.00**	0.15	0.17	0.11	0.33	0.07	0.10	0.15	0.09	0.16	0.21	0.22	0.07	0.09	0.30
0.13	0.17	0.17	**1.00**	0.39	0.17	0.27	0.11	0.08	0.17	0.12	0.14	0.11	0.12	0.08	0.18	0.12
0.10	0.21	0.16	0.38	**1.00**	0.17	0.17	0.09	0.13	0.25	0.08	0.10	0.12	0.15	0.15	0.17	0.07
0.10	0.35	0.11	0.17	0.13	**1.00**	0.11	0.13	0.22	0.09	0.13	0.26	0.18	0.13	0.08	0.13	0.11
0.31	0.21	0.33	0.20	0.16	0.12	**1.00**	0.17	0.17	0.22	0.09	0.20	0.27	0.18	0.11	0.15	0.31
0.24	0.30	0.10	0.12	0.13	0.13	0.19	**1.00**	0.16	0.18	0.08	0.12	0.16	0.07	0.07	0.12	0.12
0.10	0.29	0.11	0.08	0.15	0.31	0.16	0.14	**1.00**	0.12	0.07	0.18	0.15	0.19	0.16	0.20	0.27
0.12	0.13	0.16	0.19	0.16	0.09	0.21	0.18	0.15	**1.00**	0.11	0.07	0.17	0.05	0.16	0.38	0.18
0.05	0.09	0.09	0.11	0.07	0.07	0.09	0.09	0.07	0.13	**1.00**	0.05	0.10	0.08	0.22	0.15	0.15
0.09	0.19	0.14	0.11	0.10	0.26	0.15	0.08	0.20	0.06	0.07	**1.00**	0.19	0.19	0.07	0.07	0.22
0.15	0.29	0.19	0.11	0.09	0.20	0.25	0.12	0.15	0.15	0.13	0.22	**1.00**	0.15	0.08	0.13	0.14
0.11	0.18	0.19	0.11	0.17	0.16	0.21	0.07	0.19	0.05	0.07	0.19	0.21	**1.00**	0.04	0.06	0.17
0.06	0.08	0.12	0.09	0.14	0.08	0.11	0.07	0.12	0.13	0.23	0.07	0.12	0.06	**1.00**	0.09	0.08
0.17	0.21	0.09	0.26	0.17	0.13	0.16	0.17	0.15	0.49	0.14	0.05	0.16	0.05	0.11	**1.00**	0.13
0.21	0.16	0.26	0.12	0.07	0.11	0.31	0.13	0.28	0.14	0.14	0.21	0.16	0.20	0.09	0.14	**1.00**

Table 3–3: The correspondence array for **A0** at $n_f = 0.5$.

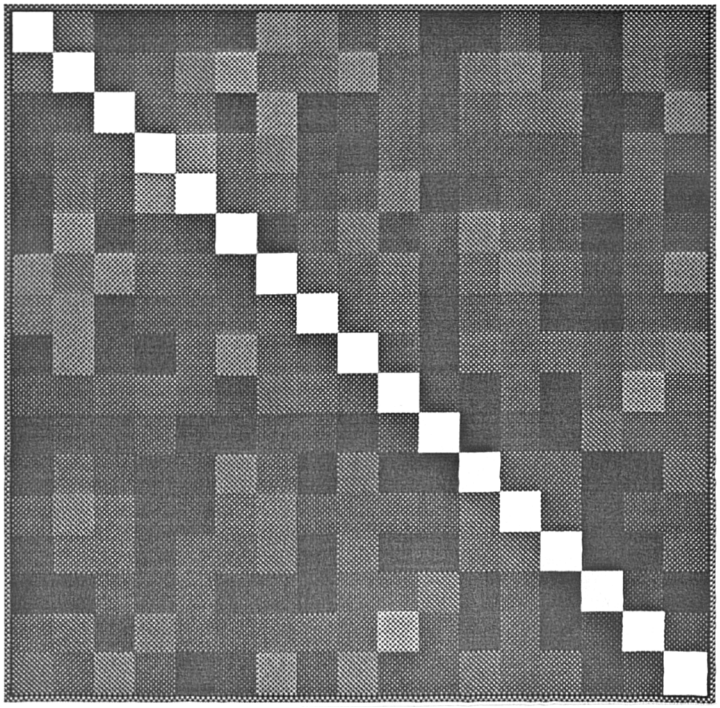

Figure 3–19: The correspondence image for **A0** at $n_f = 0.5$.

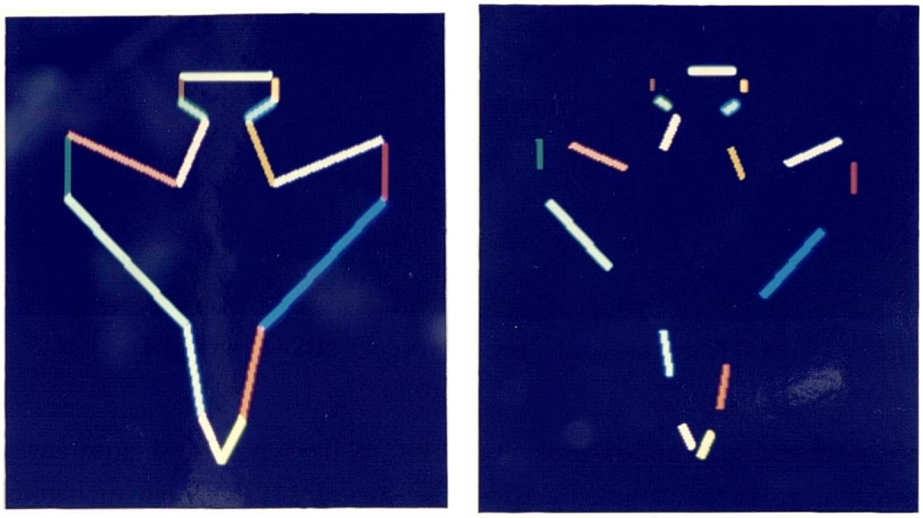

Figure 3–20: Colour-coded matches for **A0** at $n_f = 0.5$.

This was repeated for $n_f = 0.25$, and the colour-coded matches for the resulting shape are shown in figure 3–21. It can be see that correct matching is preserved across all lines, as expected from the above analysis. The histograms used to obtain this result had parameters $n_\theta = 40$, $n_d = 30$, $\sigma_\theta = \sigma_d = 1.0$. No local region was used.

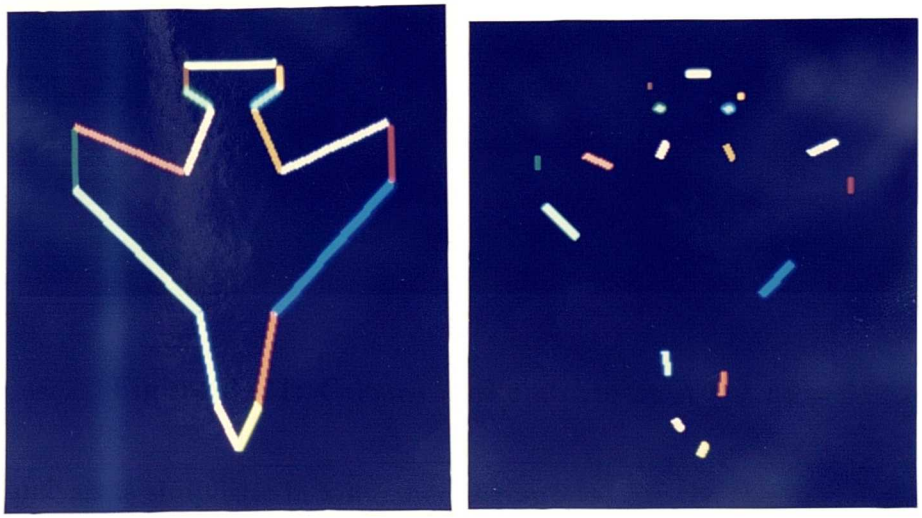

Figure 3–21: Colour-coded matches for **A0** at $n_f = 0.25$.

This analysis is simplistic in a number of respects. Firstly, the model of the effects of fragmentation on a line accounts only for cases where the line is fragmented at both ends. In reality it is quite possible that a line will be fragmented at multiple points along its length, resulting in the creation of a series of line fragments. However, while the analysis of this case is more difficult, similar results are obtained. This can be seen by examining matches for the multiply fragmented shape shown in figure 3–22, for which $n_f = 0.5$. The correspondence image for this shape is shown in figure 3–

23. It can be seen that peaks now run horizontally, as successive image fragments are matched to the same model line.

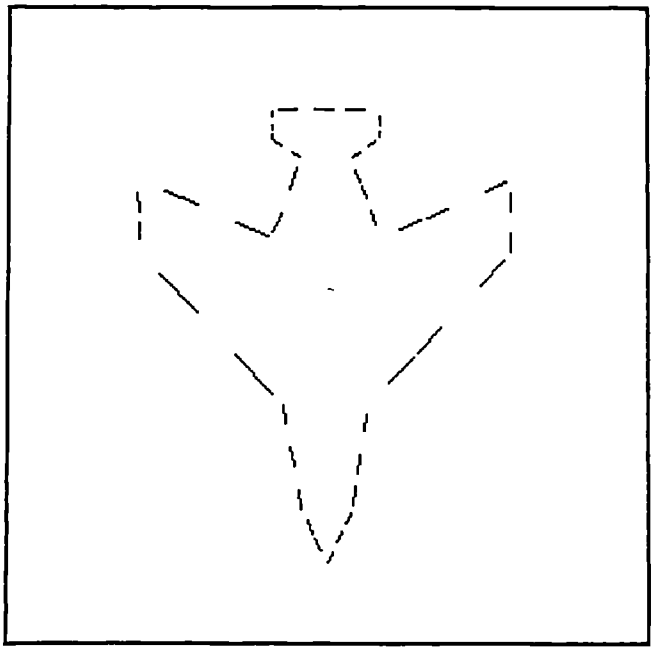

Figure 3–22: A multiply fragmented version of **A0**, at $n_f = 0.5$.

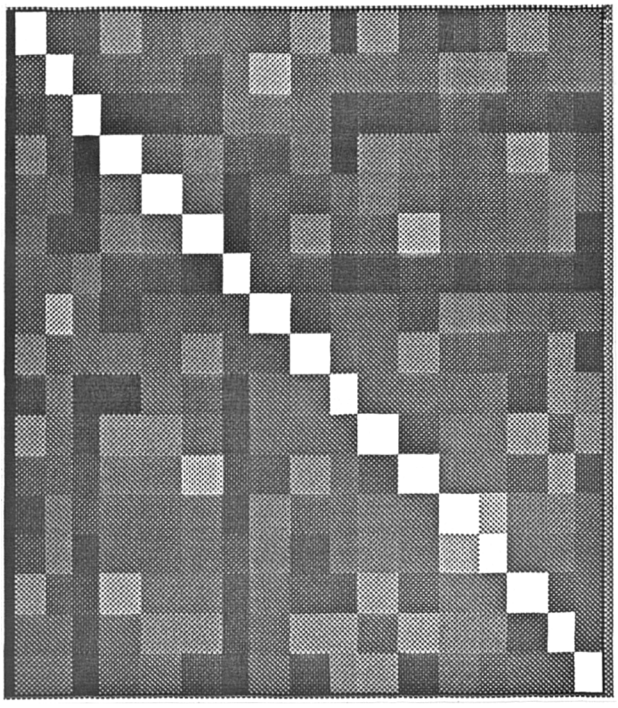

Figure 3–23: The correspondence image for the above shape.

This can be seen clearly by examining the colour-coded matches, shown in figure 3–24.

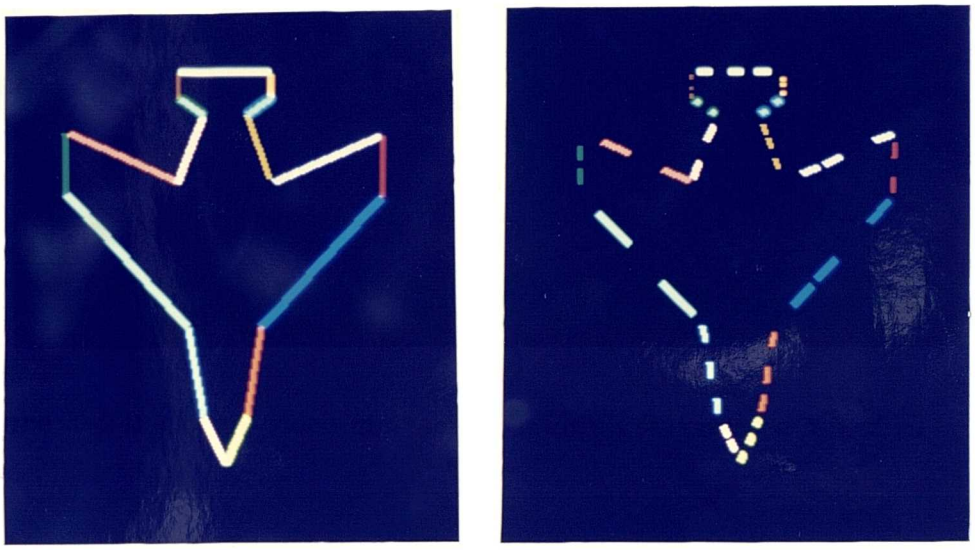

Figure 3–24: Colour-coded matches for multiply fragmented **A0** at $n_f = 0.5$.

Again this was repeated for $n_f = 0.25$. The colour-coded matches for the resulting shape are shown in figure 3–25.

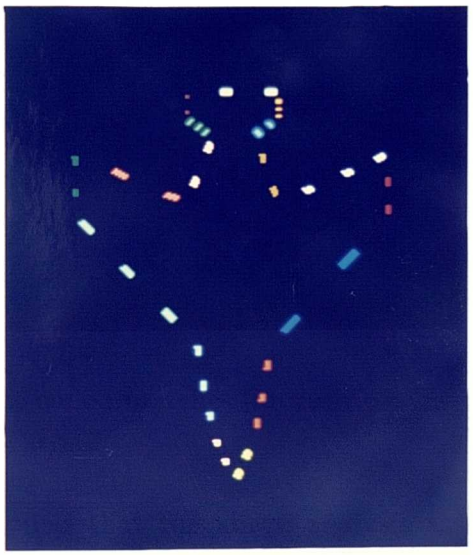

Figure 3–25: Colour-coded matches for multiple fragmented **A0** at $n_f = 0.25$.

The further simplification is that the above analysis assumes that the effect of the fragmentation is evenly distributed across the shape. In more realistic situations it is possible that the effect will be restricted to a particular region of the shape. In such circumstances it is possible that the change in the histogram representation will not be a simple scaling, and so matching may break down at lower fragmentation levels.

3.3.2 Scene Clutter

The second form of shape variation addressed is the presence in the image of spurious line segments. A spurious line segment is defined as any image line which does not belong to the object of interest. This could be due to another object in the scene or to some artifact of the lighting, eg. a shadow. Again, in order to assess the performance of the matching scheme under measurable conditions it is necessary to propose a generative model of the likely effects of detecting spurious lines in an image.

A model of scene clutter

The proposed model involves adding randomly oriented and positioned line segments to the set of lines describing a shape. The amount of added noise in the shape, n_a is quantified by

$$n_a = \frac{L'}{L}$$

where L' is the total line length in the noisy shape and L is the same measure for the original, noise-free, shape. An example of the effect of this form of noise on the shape **A0** is shown in figure 3–26, for which $n_a = 5$.

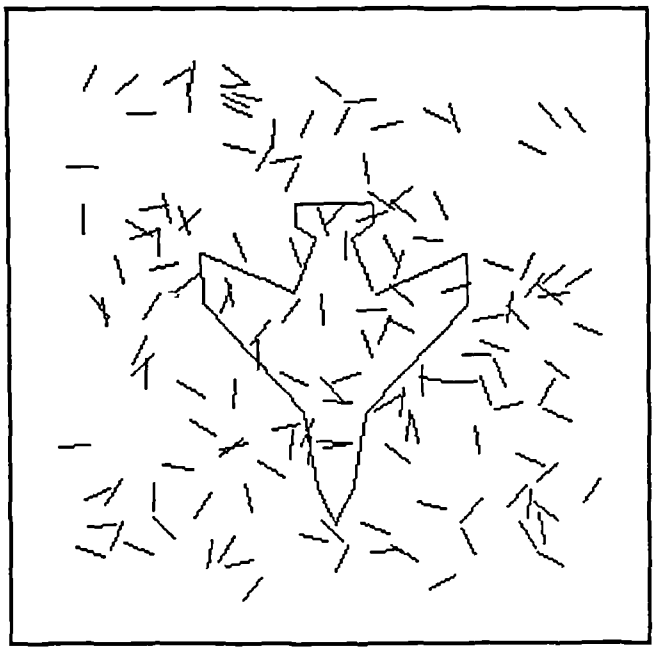

Figure 3–26: The effect of added noise on **A0**, at $n_a = 5$.

It could be argued that this model does not describe the full range of problems encountered in real scenes. In particular, the fact that they are randomly positioned with no reference to the original shape ignores the problems caused by spurious lines that are in some way *correlated*, eg. those arising from shadows. However, such conditions can be thought of as a special case; the aim of this section is to provide a general analysis of the effects of scene clutter on the matching of geometric feature distributions.

The effect on geometric feature distributions

The lines describing the shape extracted from an image of a cluttered scene can be split into two sets, the set **S**, which represents lines belonging to the object of interest and the set **N**, which represents the remaining, spurious lines. We are concerned here with describing the effect that members of **N** have on the matching of an image line, $q_i \in \mathbf{S}$. Recording the geometric feature distribution for q_i involves computing its geometric relationship not only with other shape lines but also with spurious lines. The effect of such entries can be seen in figure 3–27.

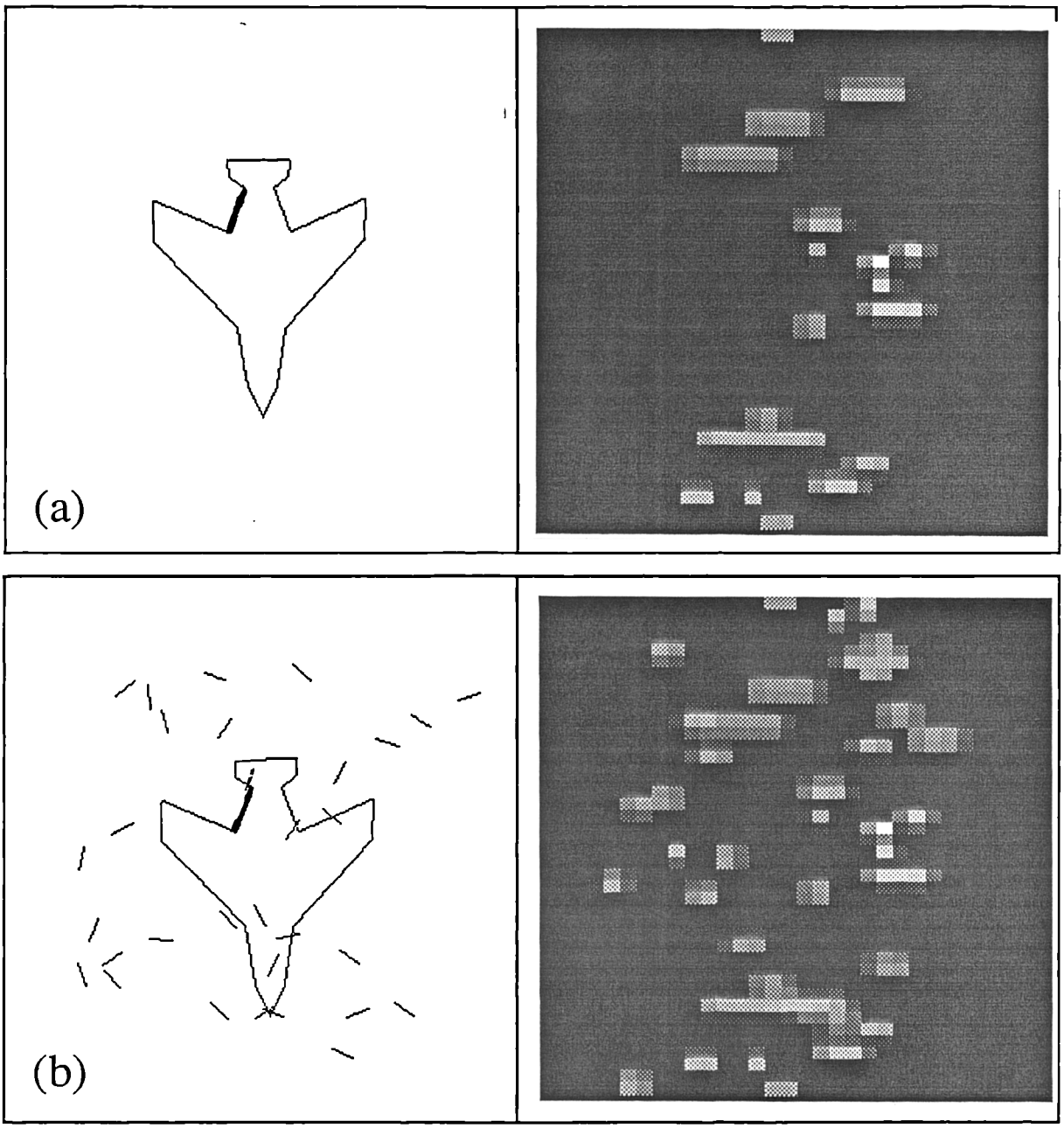

Figure 3–27: (a) the histogram for a line and (b) for its counterpart in a cluttered scene.

It will prove useful to define the set **U**, representing non-zero bins in H_{q_i} that receive entries recording the geometric relationship between q_i and $q_s \in \mathbf{S}$,

$$\mathbf{U} \equiv \{(x,y)|(x,y) \in \mathbf{\Psi},\ H_{q_i}(x,y) > 0\ \ AND\ \ G(q_i,q_s) \mapsto x,y,\quad q_s \in \mathbf{S}\}$$

where $\mathbf{\Psi}$ is the set of positions in the histogram and $G(q_i,q_s) \mapsto (x,y)$ indicates that recording the geometric relationship between q_i and q_s involves making an entry in the bin whose position is given by x,y. Similarly, for spurious lines we can define the set **T**,

$$\mathbf{T} \equiv \{(x,y)|(x,y) \in \mathbf{\Psi},\ H_{q_i}(x,y) > 0\quad AND\quad G(q_i,q_n) \mapsto x,y,\quad q_n \in \mathbf{N}\}$$

The relationship between **U** and **T** depends to a large extent on the uniqueness of the representation, as determined by the completeness of the geometric feature set and the resolution and width of blur used in the histogram. In the theoretical case where representations are unique, then

$$\mathbf{U} \cap \mathbf{T} \equiv \{\}$$

In more realistic case where representations are not unique then the intersection between these two sets will not be empty. The next section examines the effect this has on the distance metric D.

The effect on D

The behaviour of the proposed distance metric is such that its value increases with the the size of the set $\mathbf{U} \cap \mathbf{T}$. This is due to a "lengthening" of the projection of the feature vector representing image line q_i in the direction of the vector representing the corresponding model line m_i. The rate at which the value of D rises therefore provides a good indication of the uniqueness of the representation. The validity of this argument can be assessed by examining the change in the value of D computed between the histograms representing m_i and q_i as the number of added lines in the shape is increased. In order to average out the random nature of the noise the results were averaged over many trials. The graph of relative D against n_a is shown in figure 3–28. It can be seen that the value of D does, on average, increase with the number of added lines in the shape. The rate of increase is quite slow, indicating that the proposed representational scheme is quite discriminatory. The histogram used in this and subsequent experiments on scene clutter had parameters $n_\theta = 40$, $n_d = 30$, $\sigma_\theta = \sigma_d = 1.0$.

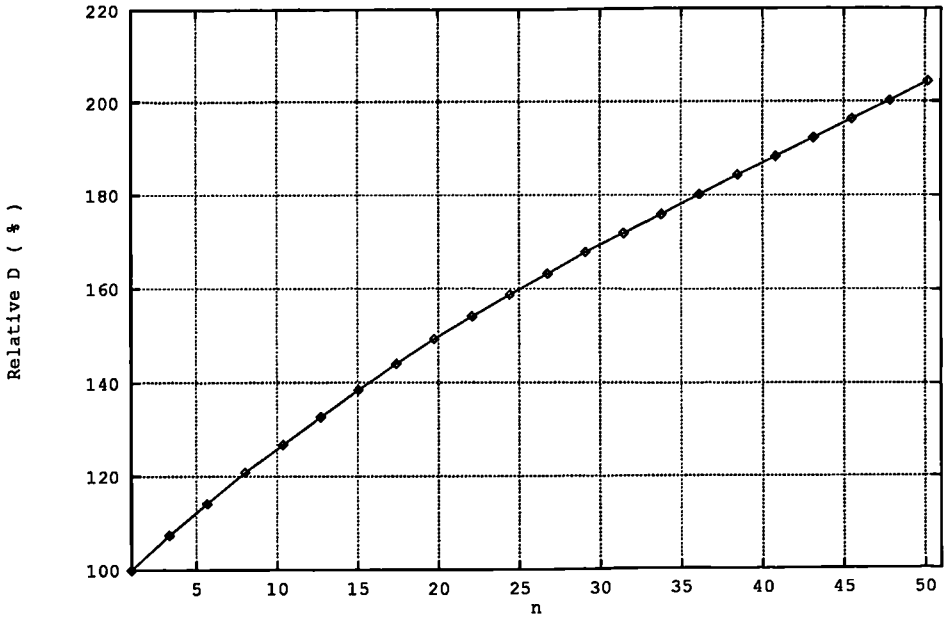

Figure 3–28: A graph of D against n_a.

The effect on nearest-neighbour classification

The above argument can be generalised to deal with the change in the value of D between q_i and model lines other than m_i, ie. the value of D should increase uniformly across all model lines. That this is the case can be confirmed by considering a graph showing the rise in the relative value of D computed between an image line, q_i, and the set of model lines, M, as the amount of added lines in the image shape is increased, figure 3–29.

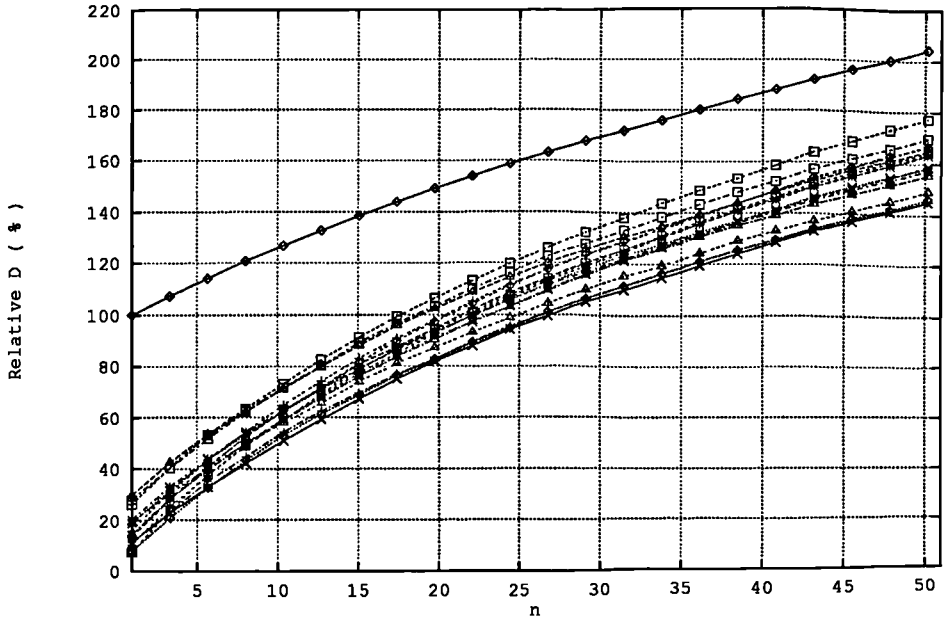

Figure 3–29: A graph of D against n_a for all model lines.

It can be seen that the separation in the value of D between correct and incorrect model lines remains roughly constant up to very high levels of added noise. This result can be appreciated by examining the effect of added noise on the shape **A0** at $n_a = 50$, figure 3–30.

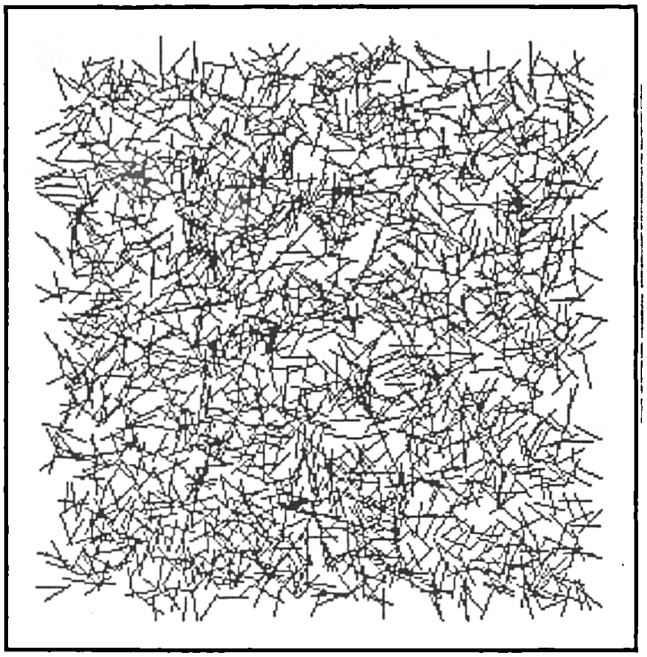

Figure 3–30: The effect of added noise on **A0**, at $n_a = 50$.

The reason for the decision not to normalise histograms representing image lines can be appreciated by examining the corresponding graph in the case where such normalisation is performed, figure 3–31.

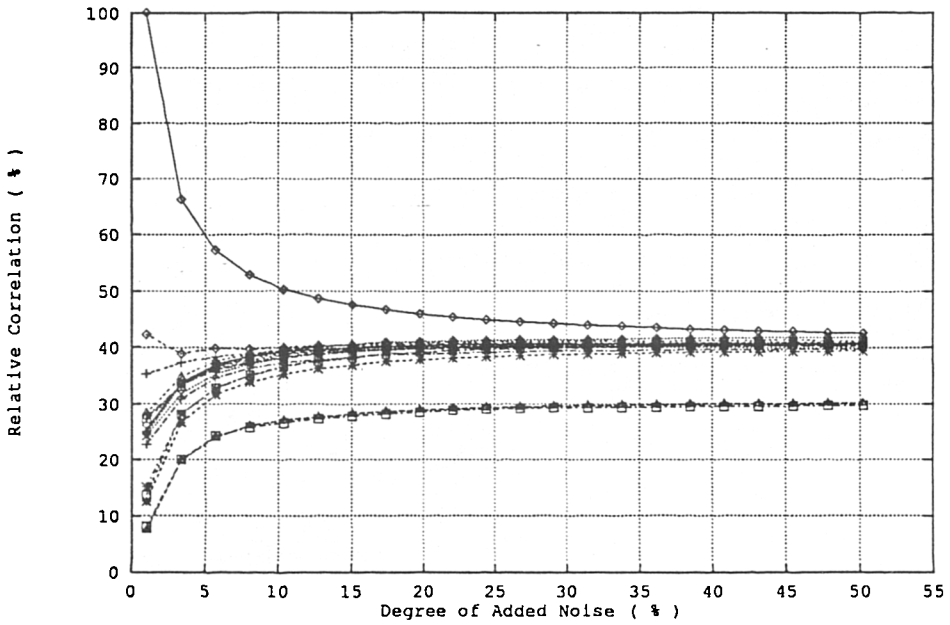

Figure 3–31: A graph of D against n_a for all model lines, with normalisation.

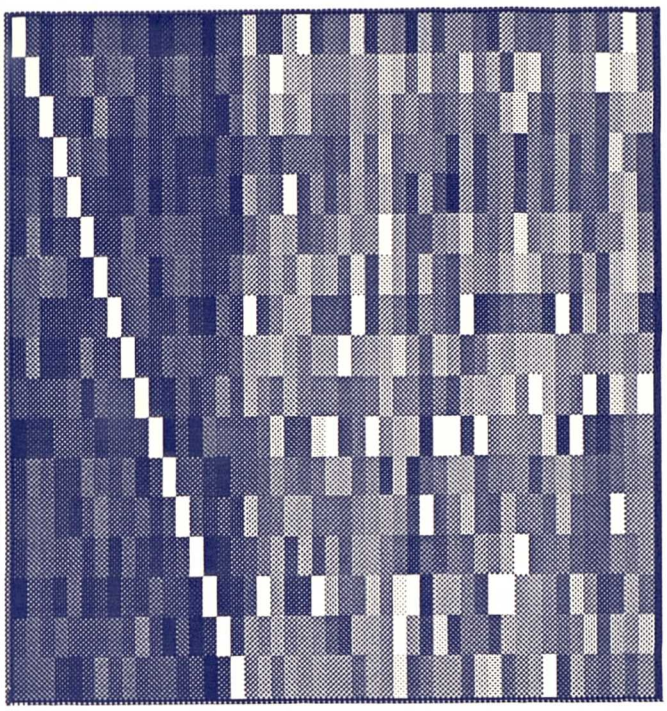

Figure 3–32: The correspondence image for **A0** at $n_a = 1.8$.

This study demonstrates that correct matching is preserved for a single image line q_i. In order to show that matching is preserved across the whole shape it is necessary to examine the correspondence array at increasing levels of noise. The correspondence image for a noisy shape, in which $n_a = 1.8$, is shown in figure 3–32. It can be see that correct matching is preserved across all shape lines, as expected from the above analysis. This can be further appreciated by examining the colour-coded matches for the noisy shape shown in figure 3–26, for which $n_a = 5$, figure 3–33.

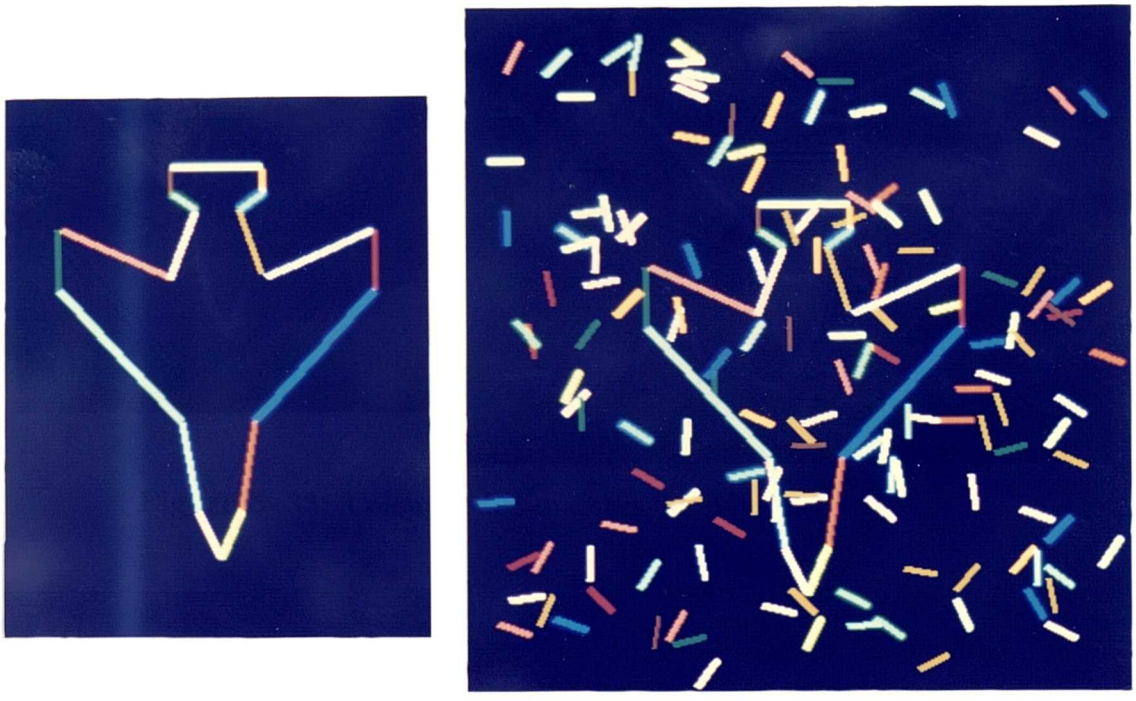

Figure 3–33: Colour-coded matches for **A0** at $n_a = 5$.

As a consequence of the matching scheme, each spurious image line is matched to a particular model line. Strategies for ensuring that such matches do not adversely affect recognition are presented in Section 3.4.

Of course, this result does not guarantee that the proposed recognition scheme will be robust to similar levels of noise arising from *real* imaging situations. Indeed, it is difficult to imagine a situation in which so many spurious lines are detected in an image while the description of the original shape remains intact. There is also the fact that the proposed model does not model correlated noise. However, it does illustrate that the matching of geometric feature distributions is theoretically robust to the presence of arbitrary spurious lines in the image shape description.

3.3.3 Sensor Error

This section assesses the effect of variations in shape description involving changes in the position and orientation of line segments. Variations in the detected position of edgels caused by true sensor error are typically very small, ($\approx$ 0.5 pixels). Of much greater importance are the problems caused by shape fragmentation and variable linear approximation. Fragmentation of an edgel string describing a curved shape is likely to result in a change in both the number and the pose of the line segments used to approximate it. Changes in the accuracy with which an approximation algorithm is applied between model acquisition and recognition can cause very similar changes. For convenience, variations due to these factors are combined under the heading of sensor error, since their effect is generally similar.

In order to assess the performance of the recognition scheme it is again necessary to provide a parameterised model of the effects of sensor error on line description.

A Model of Sensor Error

There are many possible models of the effects of sensor error on a particular line segment. One possibility is to define circular regions of uncertainty around each endpoint of a line, as in [16]. This model has the advantage of being quite realistic, since it accounts for the fact that the degree of variation in the orientation of a line segment is directly related to its length; longer line segments are more likely to be stable. However, this means that the lines within a shape are rotated through a range of angles, making it difficult to isolate the effects of a specific change in orientation. Also, the model allows the length of lines to vary. This is not desirable since line length has a significant effect on the representation, which is difficult to separate out. Therefore, the present study is based upon the model proposed in Grimson, [40]. This involves rotating each line segment around its mid-point by a fixed angle n_α, irrespective of its length, figure 3–34. This overcomes the difficulties mentioned above. An example of the effect of sensor error is shown in figure 3–35.

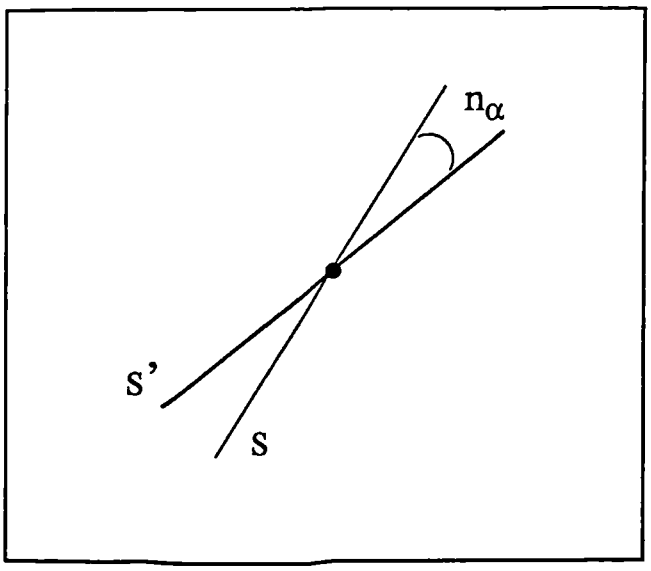

Figure 3–34: A model of sensor error.

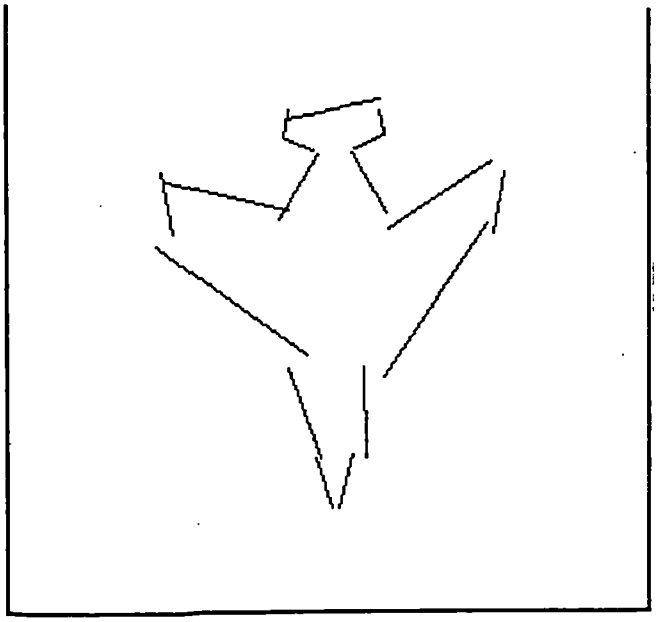

Figure 3–35: The effect of sensor error on **A0** at $n_\alpha = 10°$.

The concentration on the effects of line rotation is justified since, as will be shown, the displacement of a line segment has a second order effect on the values of the geometric features computed between pairs of lines.

Changes in the orientation of line segments that are sufficiently large will obviously result in a break down of matching based on geometric feature distributions. Indeed, this must be true of any approach based on measuring the geometric relationships between line segments. The purpose of this section section is to determine the factors affecting the degree of variation that can be tolerated.

The effect on geometric feature distributions

In order to understand the effects of sensor error on the proposed representational scheme it is necessary to analyse its effect on the values of the proposed geometric features. We therefore require expressions relating the change in the values of g_θ and g_d, the relative angle and perpendicular distance features respectively, to n_α, the angle through which each line is rotated.

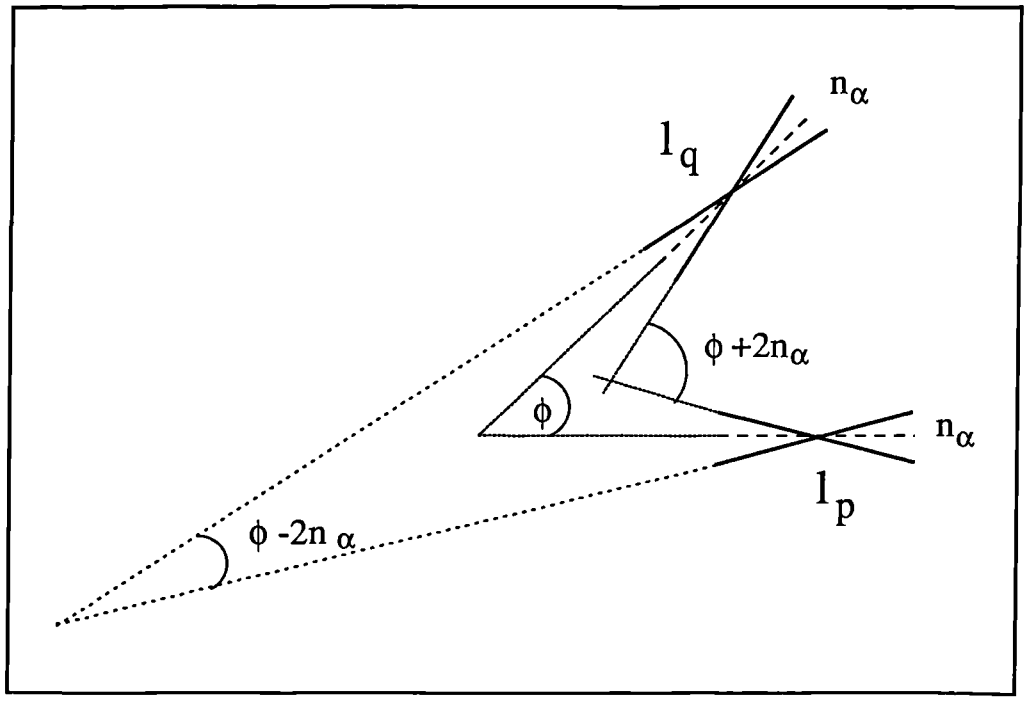

Figure 3–36: Worst case relative angle variation

Consider two line segments, ℓ_p and ℓ_q. The variation in g_θ, termed V_{g_θ}, is independent of the relative position of ℓ_p and ℓ_q. The worse case variation, shown in figure 3–36, is given by

$$V_{g_\theta} = \pm 2n_\alpha$$

Conversely, the variation in g_d, termed V_{g_d}, is strongly dependent upon the lateral displacement of the line segments, figure 3–37. The general expression for the new value of g_d for a particular endpoint **p** is given by

$$g'_d = g_d \cos n_\alpha + S \sin n_\alpha + d_q$$

Where g_d is the original value of the perpendicular distance feature, S is the distance from the mid-point of the first line to the perpendicular dropped from **p** and d_q is the change in g_d caused by the rotation of ℓ_q. The latter term will typically be small compared with the change caused by the rotation of the base line ℓ_p. A change in the orientation of the base line therefore has a dominant effect on the change in the value

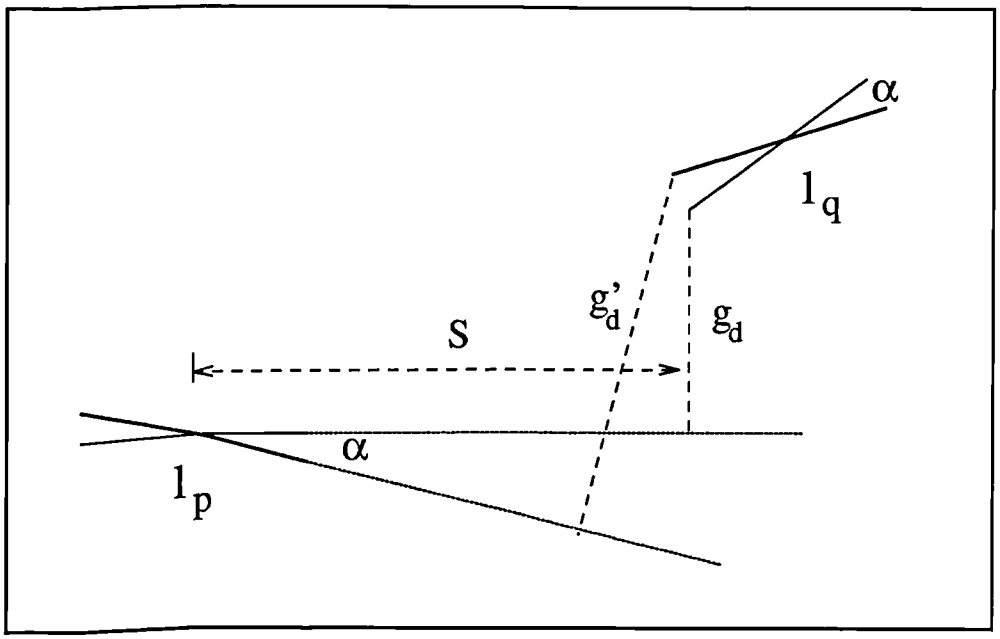

Figure 3–37: Variation in the perpendicular distance feature.

of the perpendicular distance feature. The worst case variation occurs when two lines have the maximum possible lateral displacement, ie. $S = R$.

These changes in the values of the geometric features computed between two rotated lines will obviously cause a displacement in the position in the histogram at which their geometric relationship is recorded. The next section examines the likely effects of such a change on the similarity metric D.

The effect on D

The change in the distribution of geometric features representing an image line q_i' caused by sensor error will result in a displacement of its feature vector. This suggests that the projection of q_i' in the direction of m_i, and therefore the value D computed between m_i and q_i', will fall as the amount of distortion in the image shape description is increased. The rate at which D falls is determined by the displacement, in terms of the number of bins, of the entry recording the geometric relationship between two line segments as the lines are rotated by a fixed amount. This depends on two factors, the lateral displacement of the lines, which is specific to each pair of line segments, and the resolution and width of blur used in the histogram, which is the same for all line pairs. These factors are now assessed.

Relative Line Position

The above analysis suggests that the position of lines within a shape will affect the rate at which their geometric feature distributions are affected by sensor error. Lines that are at the extreme of a shape and parallel with its major axis should be affected more than those that are perpendicular to the axis. This was investigated by plotting the change in D against n_α for three lines within the shape **A0**, figure 3–38.

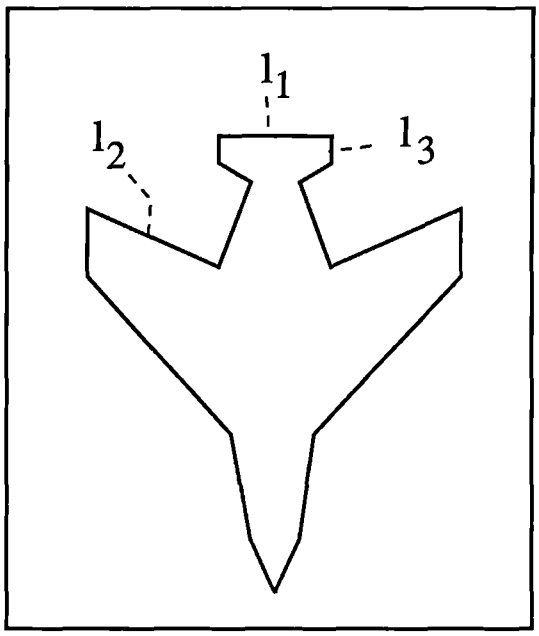

Figure 3–38: The three lines ℓ_1, ℓ_2 and ℓ_3.

The two lines, ℓ_1 and ℓ_3 provide examples of these extreme case, while ℓ_2 provides an intermediate case. From the graph shown in figure 3–39 it can be see that D does indeed fall with n_α, and the rate at which D falls is related to line position in the expected manner.

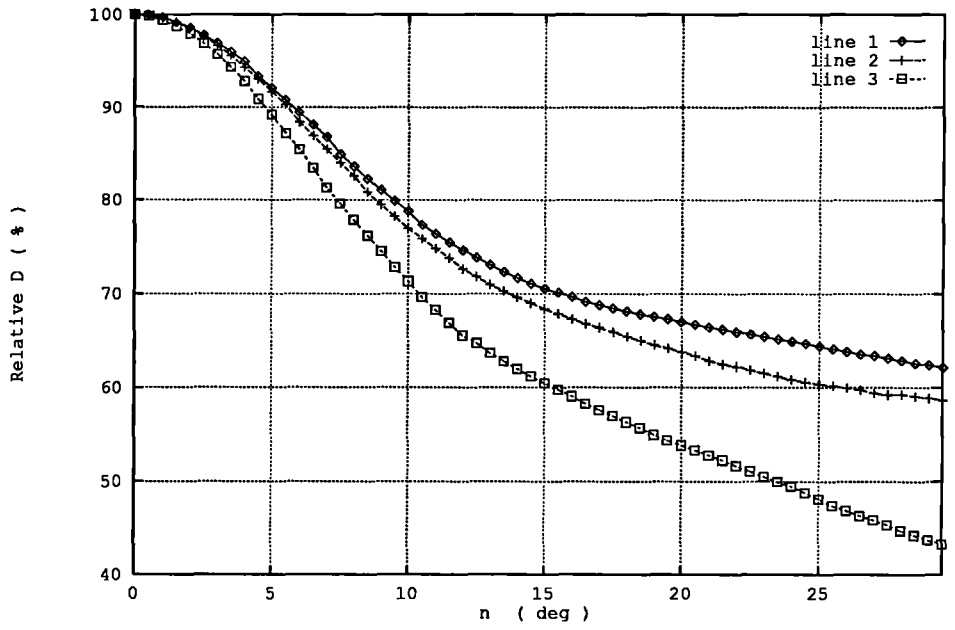

Figure 3–39: A graph relating D to n_α for the three lines ℓ_1, ℓ_2 and ℓ_3.

Number of Bins

The rate at which D falls for a particular line segment depends on the number of bins, (n_θ, n_d), used in the histogram. In particular, D will fall at a quicker rate the higher the resolution of the histogram. This is demonstrated by the graph shown in figure 3–40.

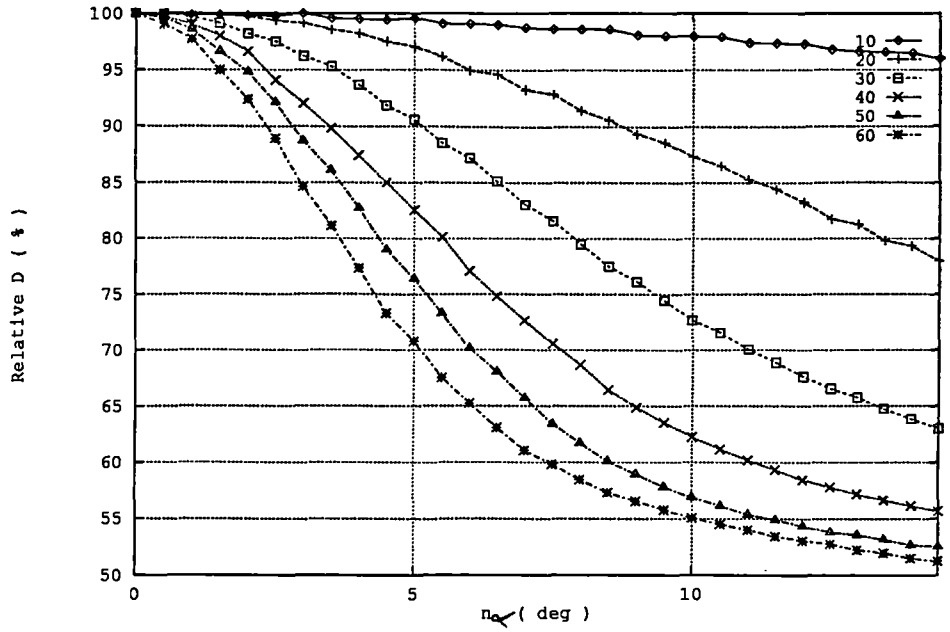

Figure 3–40: A graph relating D to n_α for different histogram resolutions.

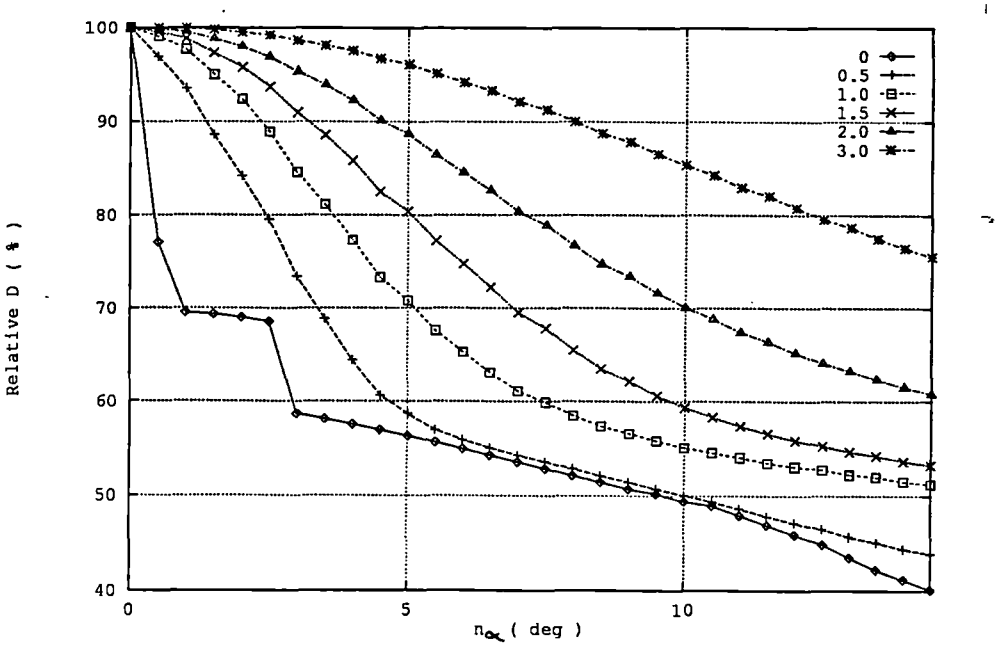

Figure 3–41: A graph relating D to n_α for different widths of blur.

Width of Blur

Similarly, the width of blur, $(\sigma_\theta, \sigma_d)$, used in the histogram can be used to vary the effect of sensor error. If no blurring is used then the smooth change in the values of the geometric features resulting from line variation is not translated to the representation,

and the fall in D is quick and irregular. As the width of blur is increased so the rate of fall in D is slowed, figure 3–41.

Ideally it would be possible to automatically determine the optimal setting of (n_θ, n_d) and $(\sigma_\theta, \sigma_d)$ for a particular value of n_α. This is an area for further study.

The effect on nearest-neighbour classification

Providing a general argument for the effects of sensor error on classification is more difficult that in the cases of fragmentation noise and scene clutter. However, it is clear that matching will be preserved up to the point at which the value of D computed between m_i and q_i' falls below that for another model line. In terms of the geometric interpretation provided above, this occurs when the feature vector representing q_i' moves out of the *Voronoi cone* associated with m_i.

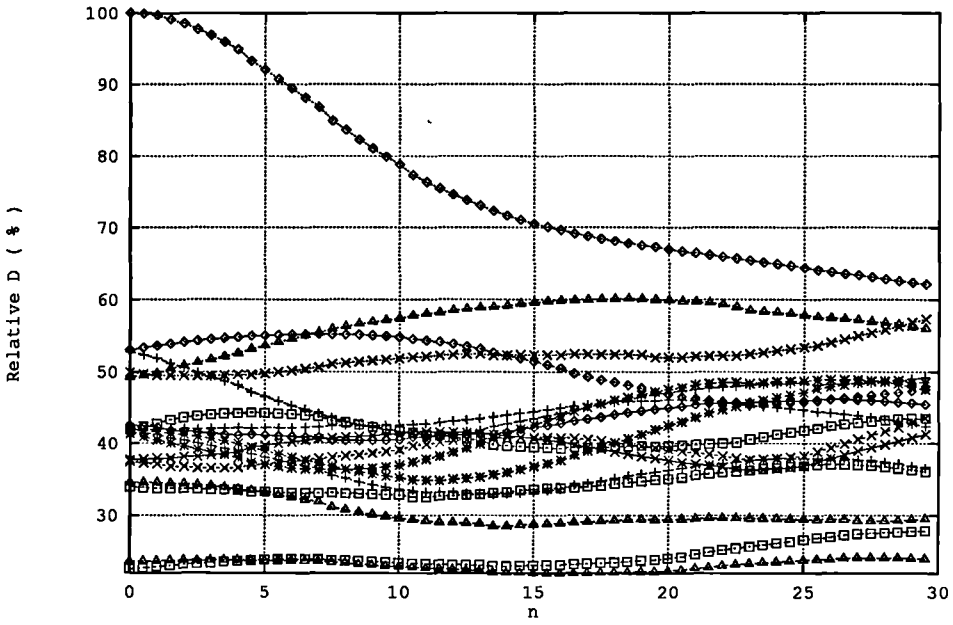

Figure 3–42: A graph showing the fall in D between ℓ_1 and all model lines.

Clearly, the amount of sensor error that can be tolerated depends on the rate at which D falls for a particular line, as determined by the factors described above. This can be seen by examining the graph in figure 3–42 which shows the fall in the relative value of D computed between the image line ℓ_1 and the set of model lines, **M**, as the amount of sensor error in the image shape description is increased. It can be seen that correct matching is preserved up to $n_\alpha = 30°$. This is to be contrasted with the corresponding graph for the line ℓ_3, figure 3–43, which shows that matching breaks down at $n_\alpha = 15°$. This study demonstrates that the position of a line within a shape critically affects the amount of sensor error that can be tolerated before matching breaks down. Of course, for a fixed line, the amount of noise that can be tolerated is also affected by the resolution and width of blur used in the histogram.

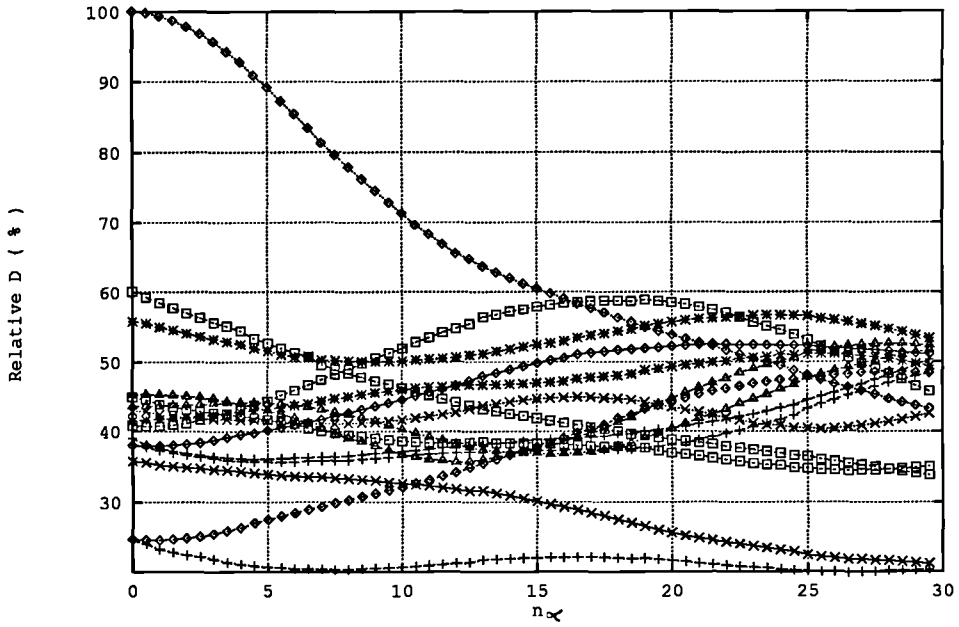

Figure 3–43: A graph showing the fall in D between ℓ_3 and all model lines.

The above studies demonstrate the effect of sensor error on the matching of individual lines. In order to examine the effect across the whole shape it is necessary to examine the correspondence array at increasing levels of noise. The correspondence array for the shape shown in figure 3–35, for which $n_\alpha = 10°$, is shown in table 3–4.

1.00	0.39	0.32	0.30	0.50	0.35	0.81	0.56	0.24	0.46	0.24	0.33	0.65	0.45	0.26	0.62	0.19
0.40	1.00	0.47	0.44	0.57	0.55	0.53	0.77	0.42	0.57	0.43	0.67	0.53	0.57	0.35	0.67	0.45
0.29	0.42	1.00	1.00	0.35	0.28	0.59	0.66	0.36	0.34	0.44	0.40	0.35	0.45	0.23	0.45	0.66
0.35	0.58	0.86	0.80	0.42	0.34	0.59	0.84	0.28	0.37	0.37	0.53	0.44	0.38	0.26	0.40	0.52
0.55	0.56	0.38	0.35	1.00	0.80	0.72	0.45	0.54	0.34	0.34	0.40	0.53	0.48	0.34	0.47	0.31
0.38	0.62	0.30	0.28	0.94	1.00	0.50	0.33	0.64	0.46	0.40	0.49	0.43	0.43	0.34	0.51	0.36
0.77	0.48	0.58	0.57	0.61	0.41	1.00	0.54	0.44	0.52	0.43	0.45	0.54	0.57	0.36	0.63	0.48
0.45	0.68	0.39	0.36	0.27	0.26	0.43	1.00	0.14	0.52	0.26	0.48	0.61	0.46	0.19	0.59	0.30
0.29	0.53	0.33	0.35	0.52	0.65	0.41	0.19	1.00	0.51	0.63	0.43	0.27	0.66	0.64	0.58	0.55
0.42	0.59	0.36	0.37	0.37	0.47	0.52	0.56	0.48	1.00	0.61	0.69	0.49	0.63	0.50	0.89	0.56
0.26	0.33	0.35	0.35	0.33	0.27	0.40	0.31	0.52	0.52	1.00	0.43	0.39	0.67	0.77	0.66	0.58
0.33	0.73	0.37	0.35	0.42	0.50	0.48	0.40	0.47	0.61	0.55	1.00	0.41	0.48	0.41	0.62	0.63
0.57	0.54	0.33	0.31	0.41	0.32	0.60	0.64	0.26	0.50	0.40	0.47	1.00	0.57	0.34	0.62	0.27
0.46	0.47	0.46	0.48	0.40	0.37	0.56	0.40	0.60	0.39	0.47	0.39	0.46	1.00	0.44	0.52	0.57
0.26	0.41	0.27	0.26	0.35	0.37	0.36	0.27	0.62	0.59	0.91	0.46	0.38	0.61	1.00	0.68	0.51
0.51	0.49	0.43	0.47	0.37	0.36	0.55	0.49	0.51	0.69	0.60	0.40	0.47	0.98	0.50	1.00	0.62
0.16	0.49	0.54	0.58	0.30	0.35	0.38	0.39	0.57	0.51	0.79	0.59	0.28	0.55	0.47	0.65	1.00

Table 3–4: The correspondence image for **A0** at $n_\alpha = 10°$.

The correspondence image and colour-coded matches for this shape are shown in figure 3–44 and figure 3–45 respectively. It can be seen that while the degree of separation in the value of D between lines is decreased, correct matching is preserved for all but one line. To achieve this result it was necessary to use a coarser histogram than in the previous sections, with parameters $n_\theta = 30$, $n_d = 20$, $\sigma_\theta = \sigma_d = 1.5$.

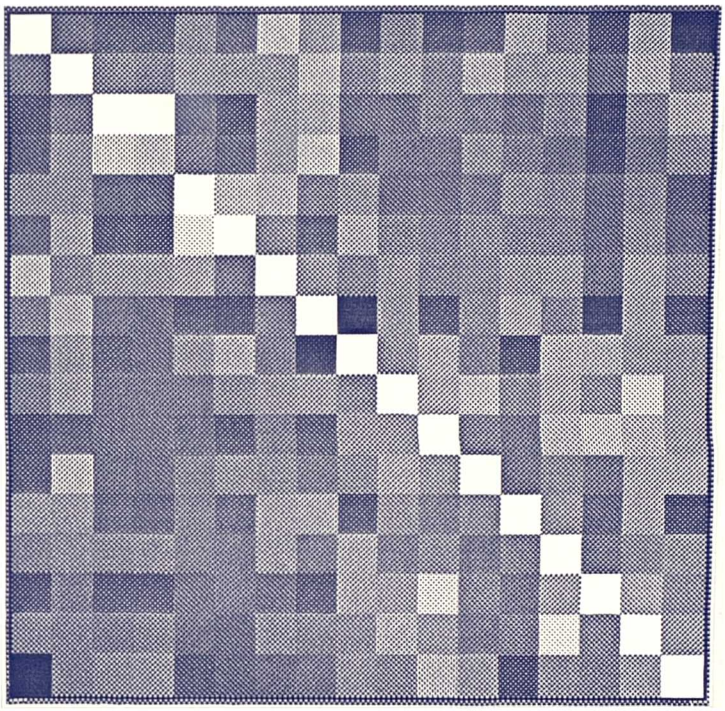

Figure 3–44: Correspondence image for **A0** at $n_\alpha = 10°$.

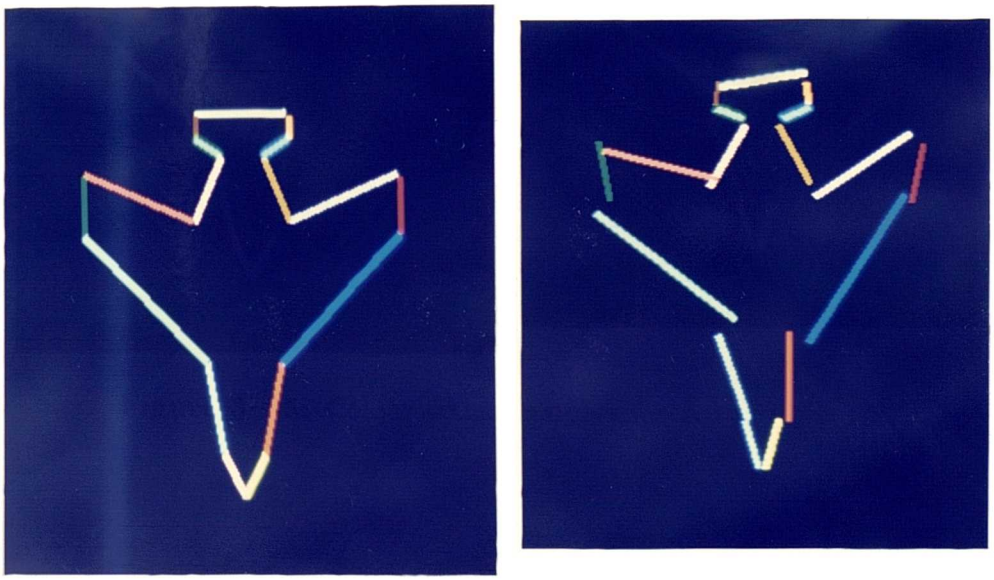

Figure 3–45: Colour coded matches for **A0** at $n_\alpha = 10°$.

This was repeated for $n_\alpha = 20°$. The colour-coded matches for the resulting shape are shown in figure 3–46.

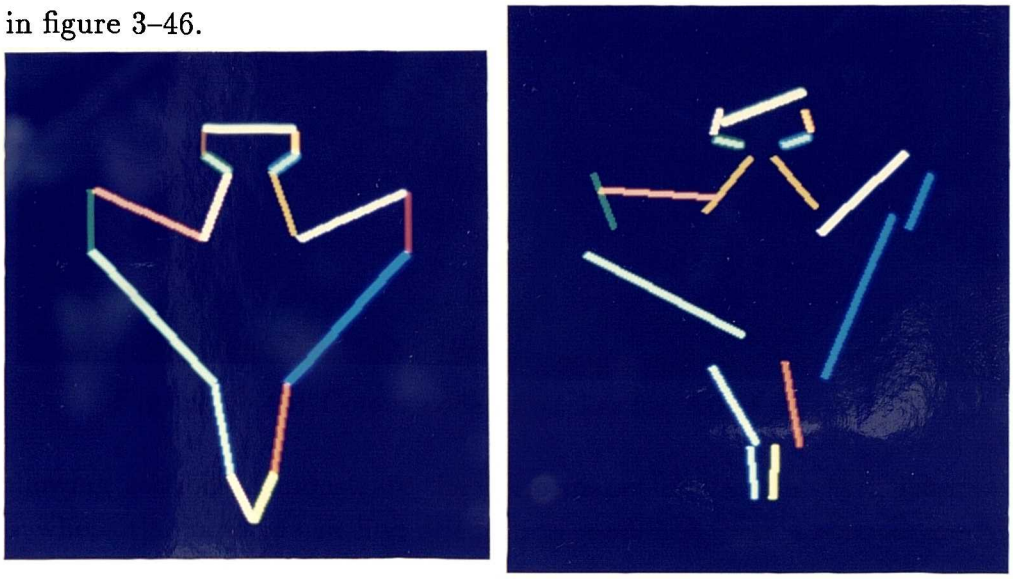

Figure 3–46: Colour coded matches for **A0** at $n_\alpha = 20°$.

The following sections demonstrate the performance of the matching scheme in situations where the variation in line orientation modelled in this section are likely to occur.

Variable straight line approximation

Variation in the accuracy with which linear approximation process is applied between model acquisition and recognition can produce effects very similar to those modelled above. Providing a general account of the changes is very difficult, since it depends on the characteristics of the approximation algorithm, the degree of curvature of the shapes and accuracy range in which the changes occur. However, a possible situation for a curve of constant radius where the approximation accuracy is increased is shown in figure 3–47. It can be seen that both the position and orientation of the two image lines differ from that of the original model line.

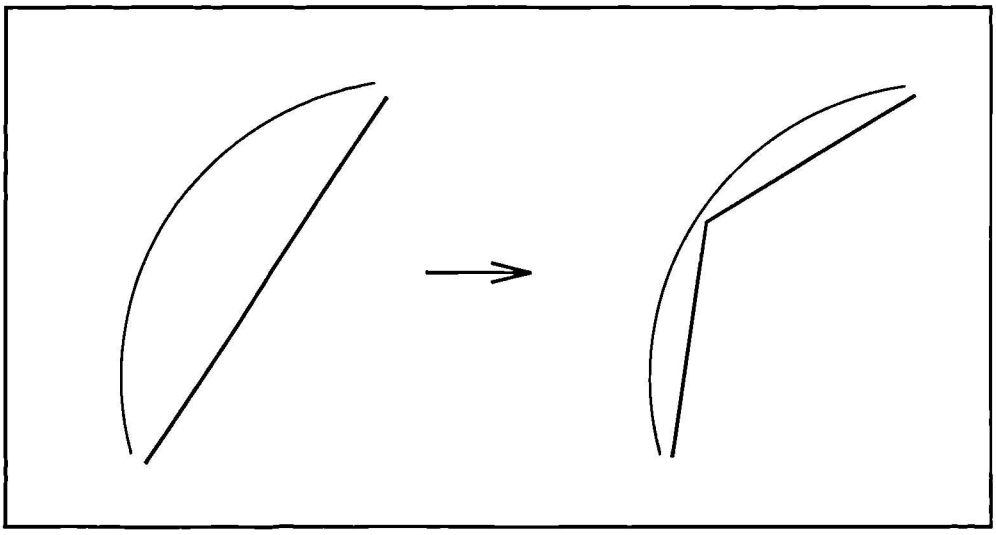

Figure 3–47: The effect of increasing linear approximation accuracy.

The performance of the matching scheme under such conditions was tested in the following way. The curved shape shown in figure 3–48 was approximated at a relatively coarse level of accuracy to give a model line description composed of 21 lines. A more accurate approximation was then performed to give a scene line description containing 35 lines. The correspondence image and colour-coded matches for these shapes are shown in figure 3–49 and figure 3–50 respectively. It can be seen that despite the change in orientation of the lines they are, in general, correctly matched. The histograms used in this experiment had parameters $n_\theta = 40$, $n_d = 30$, $\sigma_\theta = \sigma_d = 1.0$.

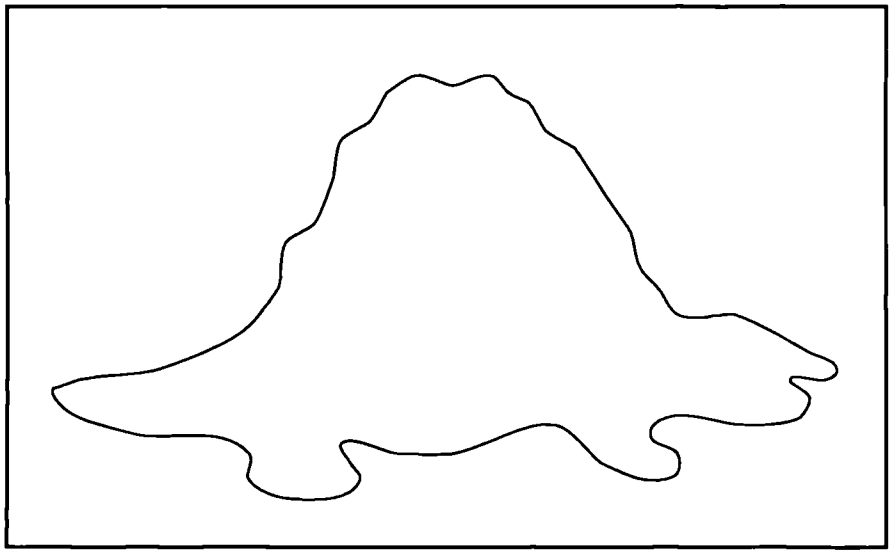

Figure 3–48: A curved shape.

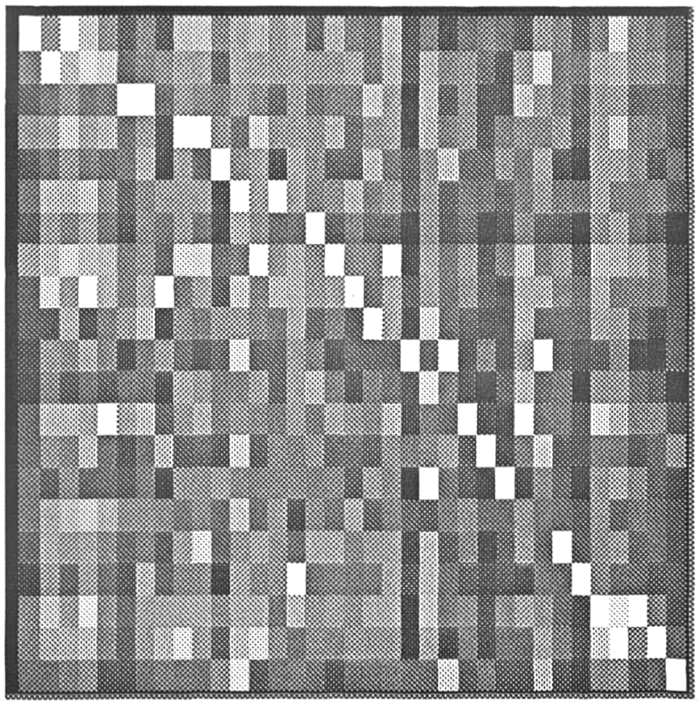

Figure 3–49: The correspondence image.

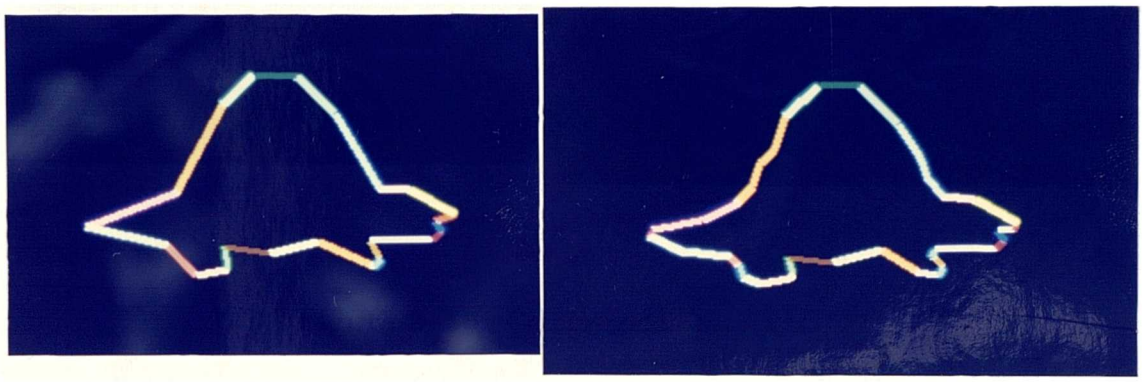

Figure 3–50: Colour-coded matches.

The proposed matching scheme should therefore be fairly robust to small changes in the straight line approximation of shape. If robustness to larger changes is required then an obvious solution would be to store multiple representations, each based on a line approximation obtained at a different level of accuracy, (cf. [96]), although this obviously increases both the memory and computation requirements of the system.

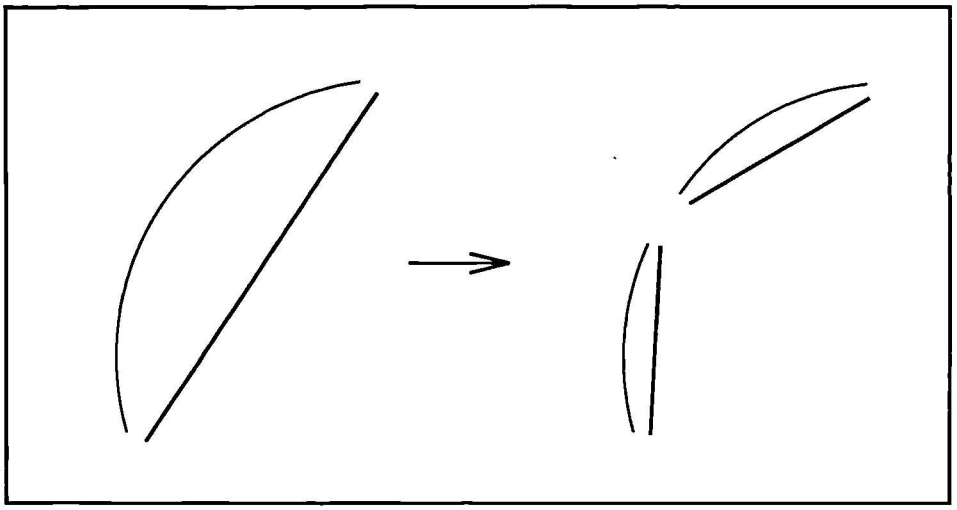

Figure 3–51: The effect of fragmentation on a curve of constant radius.

Fragmentation of curved shapes

Section 3.3.1 established that the loss of shape information will, in general, have little or no effect on classification. This was based on a model of fragmentation in which the orientation of lines was preserved. However, if fragmented edgel strings are describing the projection of a non-polyhedral object then this is unlikely to be the case. An example of the kinds of problems that can occur is shown in figure 3–51. It can be seen that the effect of fragmentation is very similar to that arising from variable straight line approximation, although there is the added problem of data loss.

The performance of the matching scheme under such conditions was tested by fragmenting the edgel strings describing the shape in figure 3–48 by removing randomly spaced continuous sections, such that the total loss of edgels amounted to 50% of those in the original shape. The remaining strings were then approximated at the same level of accuracy. The correspondence image and colour-coded matches for the resulting shape are shown in figure 3–52 and figure 3–53 respectively. It can be seen that despite both the fragmentation and the change in orientation of the lines they are, in general, correctly matched. The histograms used in this experiment had parameters $n_\theta = 40$, $n_d = 30$, $\sigma_\theta = \sigma_d = 1.0$.

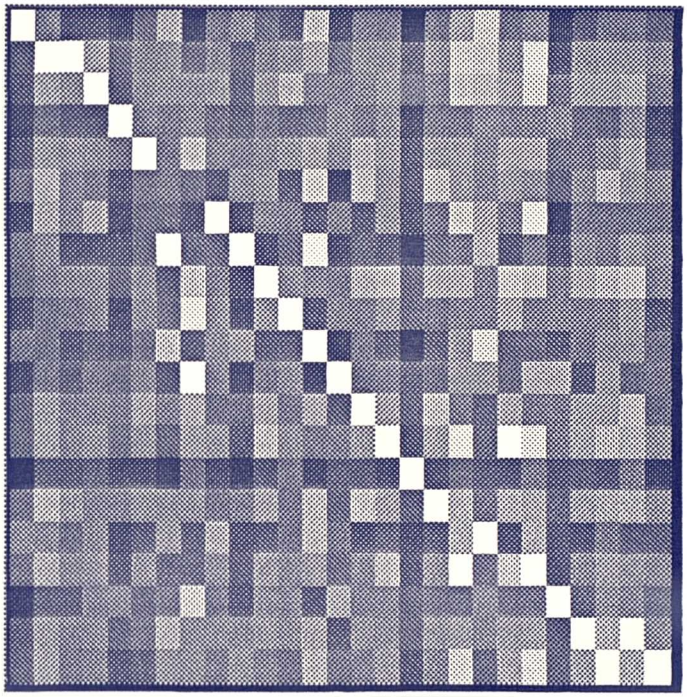

Figure 3–52: The correspondence image.

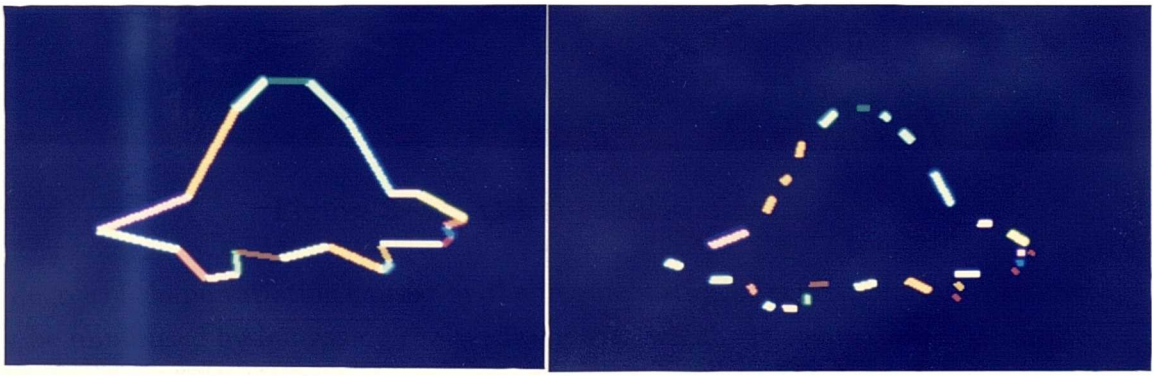

Figure 3–53: Colour-coded matches.

The degree of shape variation caused by the fragmentation of curved contours can obviously be minimised by increasing the accuracy with which approximation is performed at the model acquisition stage.

3.4 Determining Object Pose

The preceding sections have demonstrated that a scheme based upon statistical classification of local geometric feature distributions is able to establish valid line correspondences in a wide range of changed viewed conditions. The matches established in this way represent a first order segmentation of the lines detected in the image. However, many applications require that in addition to this, the pose of each object in the scene should be determined. If all matches provided by histogram classification were guaranteed to be correct then this could be achieved straightforwardly by performing a least squares fit of the transformation parameters computed from these matches.

However, the local nature of the proposed matching scheme means that there can be no guarantee as to the validity of the set of matches produced. Mismatches may occur through the classification of spurious lines or through confusions between valid object lines caused by severe noise. In general, such mismatches are not even *approximately* correct, and their inclusion in the computation of the transformation parameters will result in large errors.

Two solutions to this problem are considered; the first involves applying *local* constraints in an attempt to remove all mismatches prior to computing the transformation parameters using direct methods, while the second approach exploits the powerful global constraint that the set of line matches must be consistent with a uniform transformation of the object in the scene.

3.4.1 Local Methods

The primary objective of any local constraint is to remove from consideration as many mismatches as possible, while not ruling out any valid matches. We now consider a number of possible local constraints.

The Unary Geometric Constraint

The notion of a geometric feature was introduced in Chapter 2. While the proposed representational scheme is based on binary geometric features, computed between pairs of line segments, it is also possible to make use of unary geometric features as a direct constraint on matching. In the case of line segments the only available unary geometric feature is line length, [16,40].

This can be used as a constraint on matching by requiring that the length of an image line is not greater, by some fraction e, than that of the model line to which it is matched.

$$\textit{length constraint } (\ell_q, \ell_m) \equiv TRUE \;\; iff \;\; |\ell_q| \leq |\ell_m| + e.|\ell_m|$$

The use of the inequality means that valid image lines that are shorter than the model line, through fragmentation or occlusion, are not ruled out. The advantage of this constraint is that it is very efficient to apply and can be used to rule out possible line pairings *prior* to the computation of D, thereby speeding up recognition. However, its use does place certain constraints on the relationship between model and image line descriptions. Firstly, the model line description should be free of fragmentation noise. Secondly the accuracy of linear approximation used to derive the image line description should be equal to, or greater than, that used in obtaining the model line description. These conditions should ensure that valid matches are not ruled out as a result of this constraint.

Thresholding on the value of D

While the nature of the matching scheme is such that all spurious lines must be matched to a particular model line, it would seem reasonable to expect that the maximum value of D computed for spurious lines would be lower than that for valid shape lines. Since a spurious line does not, by definition, belong to any known object, it is unlikely that the feature vector representing it will fall in the region of feature space occupied by the set of model lines. Its projection in the direction of the feature vector representing any model line is therefore unlikely to be high. This suggests that matches with spurious lines can be ruled out by placing a threshold on the maximum value of D.

Unfortunately, the value of D depends on the length of the feature vector representing each image line, which in turn depends on the total length of lines in the shape. There is, therefore, no way of distinguishing between low values of D resulting from spurious lines and those arising from valid lines in a fragmented shape. A possible solution to this might be to normalise the value of D with the sum of the length of lines in the shape. The normalised value of D for lines in fragmented shapes would then be high. However, this solution fails in cases of scene clutter, since the values of D for valid lines are forced downwards by the added lines. This approach is therefore discounted.

3.4.2 Generalised Hough Transform

The previous section has demonstrated that attempts to remove mismatches by relying on locally computed constraints do not provide a complete solution. This section presents a method for determining object pose which exploits the powerful *global* constraint that all valid matches must correspond to a uniform transformation of the object in the scene. The particular method used is the generalised Hough transform, [4], which relies on clustering to implement the global constraint. Alternative methods which rely on establishing multiple pairwise geometric consistency between matches could have been used, eg. tree-search, [40] or maximal clique analysis [13].

The generalised Hough transform has been proposed as a method of recognising arbitrary shapes in cluttered scenes, [4]. In its original form this involves considering each

image line as potential match for each model line. Each hypothesised match generates a set of transformation parameters which are used to cast a vote in a quantised representation of transformation space. This scheme relies on the fact that valid associations will tend to vote for a consistent transform, while votes due to mismatches will be evenly distributed throughout the space. By detecting peaks in the histogram representing transformation space, likely object transformations can be obtained.

The GHT is often criticised on the grounds that making votes for all I^M possible matches constitutes an excessive amount of computation. In the present scheme, where each image feature has already been matched, the number of votes that have to be made is reduced to I. This also has the effect of reducing the clutter in the transformation space, since the majority of matches will be correct.

The use of the GHT to determine object pose, based on matches established through the classification of geometric feature distributions, is now described.

Voting

Objects are restricted to transformations in 2D position and orientation. These can be described using 3 parameters, 2 translation and 1 rotation, which means that a 3-dimensional Hough space must be used.

In order to make a vote in Hough space it is necessary to compute values for these 3 transformation parameters from a pair of matched model and image lines. While the rotation parameter can be computed straightforwardly, determining the translation parameters is more difficult. In order to uniquely determine these values from a pair of matched line segments it is necessary to define a fixed point on each line; the centroid is an obvious choice. However, as we have seen, such characteristics are not robust to line fragmentation caused by image noise or occlusion. This suggests that each pair of matched line segments can only be used to constrain the translation parameters to lie along a line in transformation space. This can be seen from figure 3–54. The position of the centre of the object, relative to the matched image line q_i, may lie at any point along a line parallel to q_i and at a fixed distance from it, (for non-directed line segments two such lines must be considered.) The length of this line is equal to the length of q_i plus some factor, e, to account for noise.

A direct solution to this problem would be to vote for each point along this line. However, this is both computationally expensive and increases the amount of clutter in the Hough space. The chosen solution is to exploit the fact that pairs of non-parallel matched lines can be used to uniquely determine the translation, by computing the intersection of the possible lines in transformation space. This suggests that $\frac{n.(n-1)}{2}$ entries must be made in each hough space, where n denotes the number of matched image lines for each object. However, two tests can be used to rule out a large proportion of the possible pairings. The first involves checking that the two matches belong to a consistent transformation of the object in the scene. This is determined by whether or not the lines associated with each match intersect in transformation space. The

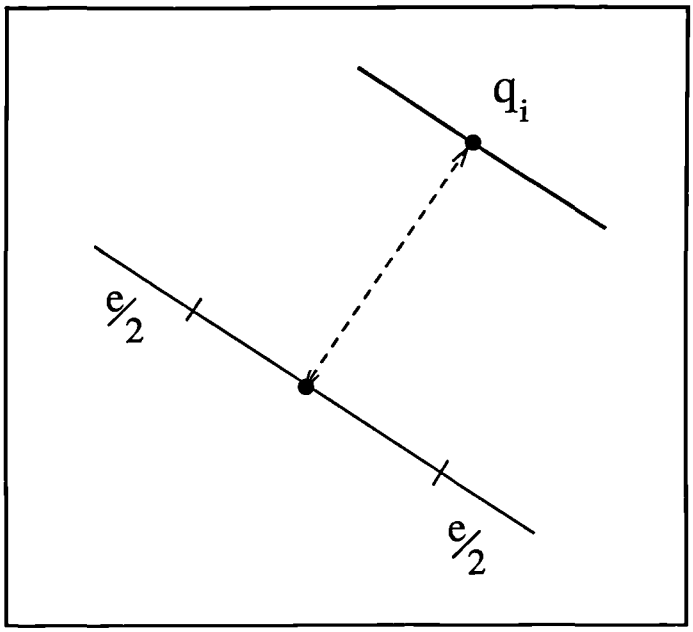

Figure 3–54: Ambiguity in the translation parameters from a single matched line.

second test is a check for geometric consistency between the two matches. These two tests combine to dramatically reduce the number of spurious votes in Hough space, especially in cluttered scenes. Votes are weighted with the product of the line lengths and blurred with a Gaussian to encode possible variations in the pose of matched image lines.

Pose Validation

Once votes for all consistent pairs of matches have been made then possible object transformations are obtained by detecting peaks in the Hough space. It is possible to attempt to validate hypotheses by projecting the model into the image and computing the amount of local support it receives from image line segments. A threshold is then placed on the fraction of model lines that must receive local support for an hypothesised transformation to be regarded as valid. However, this method is not able to deal robustly with situations in which a large proportion of the lines from an object are missing due to occlusion. Consequently, the test used in this system is the relative height of the peak. That is, the process of peak detection is repeated until the height of the peak, relative to the highest peak, falls below a particular threshold value. This method was found to perform robustly in scenes containing high degrees of occlusion. While it is possible to use information gained from image lines that support a projected model line to update the hypothesised transformation, eg. [3], this was not done in the present system.

The ability of the proposed scheme to perform recognition is presented in Chapter 4.

3.5 Global Shape Matching

One of advantages of the proposed scheme is that the matching of local and global shape representations can be achieved using essentially the same mechanism. In the case of global geometric feature distributions the value of D indicates the degree of similarity between complete shapes. The ability of the GFD scheme to perform non-correspondence recognition was investigated using the set of animal shapes shown in figure 3–55. These were chosen as they provide a set of arbitrary curved shapes containing a range of complexity, and so represent a reasonable test of the representation and matching schemes.

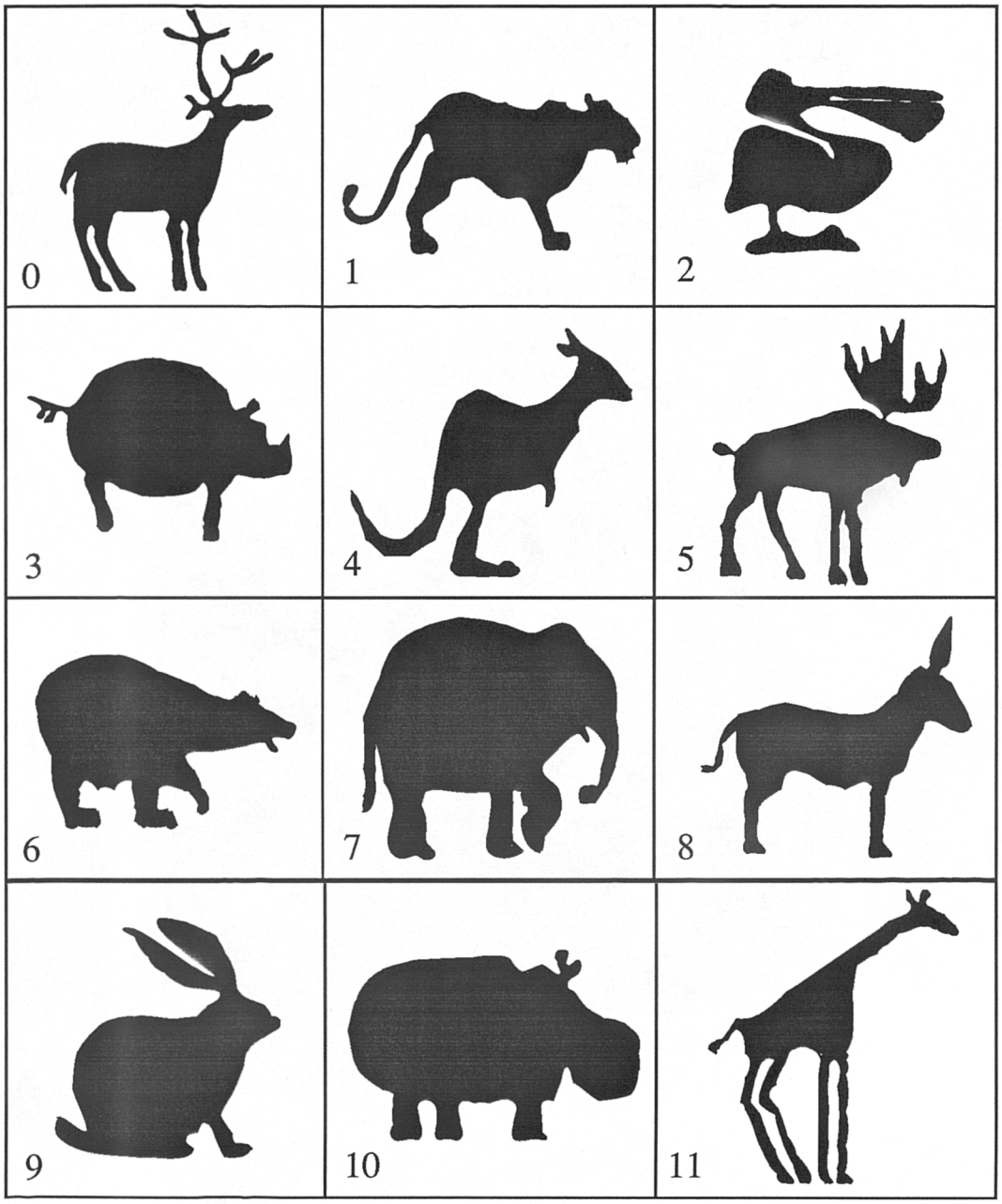

Figure 3–55: A series of animal shapes.

An image of each animal was captured and used to construct object models. The histogram used to record the global geometric feature distributions had parameters $n_\theta = 50$, $n_d = 40$, $\sigma_\theta = \sigma_d = 1.5$. An image of each animal at a different position and orientation within the scene was then captured and matched to the model shapes. The correspondence array showing the value of D computed between model and image shapes is shown in table 3–5, while the correspondence image is shown in figure 3–56. It can be seen that, while the degree of separation in the value of D computed between global geometric feature distributions is smaller than in the case of local representations, the fact that peak values still lie along the diagonal indicates that each shape is correctly matched.

1.0000	0.9725	0.9496	0.9409	0.9688	0.9820	0.9278	0.9449	0.9674	0.9523	0.9118
0.9718	**1.0000**	0.9507	0.9595	0.9627	0.9649	0.9479	0.9480	0.9636	0.9611	0.9408
0.9496	0.9496	**1.0000**	0.9229	0.9247	0.9594	0.9273	0.9230	0.9478	0.9395	0.9111
0.9349	0.9529	0.9172	**1.0000**	0.9315	0.9310	0.9554	0.9564	0.9468	0.9609	0.9652
0.9686	0.9642	0.9236	0.9364	**1.0000**	0.9530	0.9266	0.9362	0.9487	0.9421	0.9164
0.9847	0.9718	0.9644	0.9439	0.9603	**1.0000**	0.9398	0.9491	0.9753	0.9591	0.9233
0.9303	0.9511	0.9286	0.9576	0.9291	0.9372	**1.0000**	0.9333	0.9447	0.9433	0.9599
0.9411	0.9458	0.9171	0.9579	0.9344	0.9410	0.9290	**1.0000**	0.9508	0.9651	0.9502
0.9693	0.9680	0.9521	0.9536	0.9528	0.9727	0.9451	0.9564	**1.0000**	0.9623	0.9343
0.9534	0.9639	0.9405	0.9669	0.9457	0.9549	0.9456	0.9716	0.9616	**1.0000**	0.9457
0.9120	0.9338	0.9052	0.9625	0.9162	0.9148	0.9534	0.9475	0.9266	0.9383	**1.0000**

Table 3–5: The correspondence array for the animal shapes.

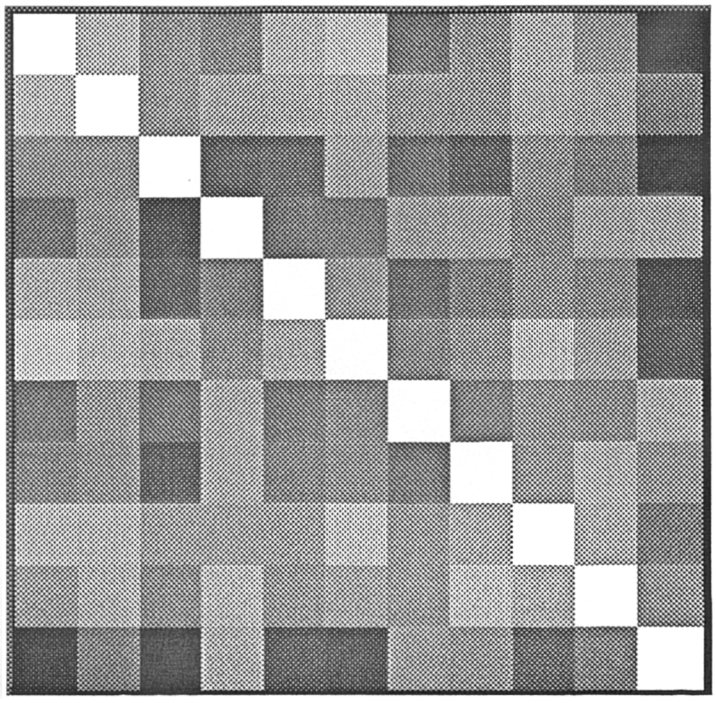

Figure 3–56: The correspondence image for the animal shapes.

In the case of the animal shapes it is difficult to interpret whether the value of D computed between the representation of two shapes correctly captures their similarity,

since the relationships between shapes are unknown. If a set of shapes were available in which the similarity relationships between shapes were known, then the ability of the combined representational and matching schemes to capture this relationship could be tested. Such a set was created by applying a "morphing" process to transform animal 1, the cheetah, into animal 8, the donkey. This involved describing each shape using an equal number of control points. Intermediate shapes were then generated by moving each point along a line linking it with the corresponding control point on the other shape. This process was applied in 10 evenly spaced steps to give the 10 shapes shown in figure 3–57.

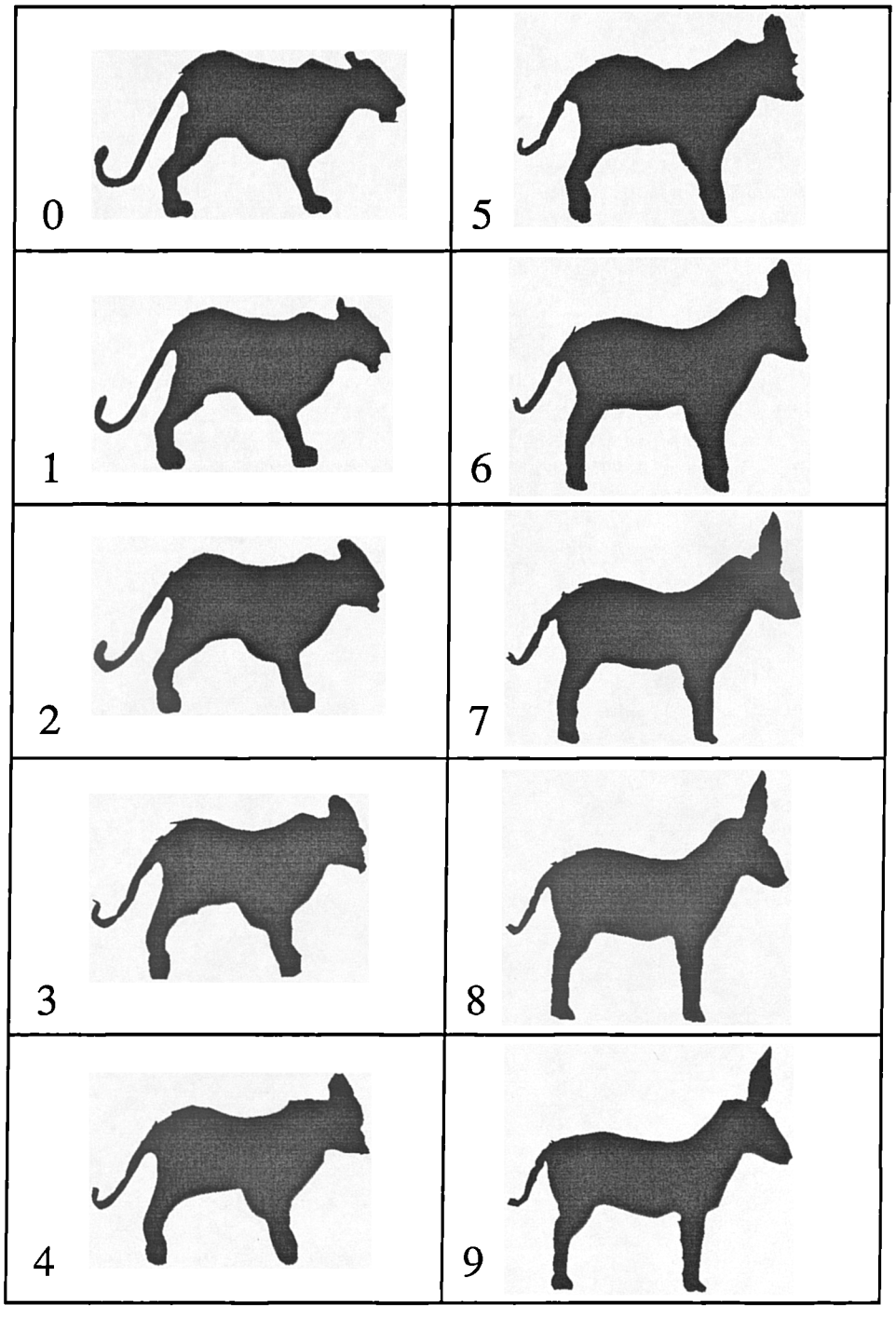

Figure 3–57: A series of "morphed" shapes.

The above procedure was repeated for these "morphed" shapes to give the correspondence image shown in figure 3–58. It can be seen that the smoothly changing nature of the shapes is successfully captured, both by the representational scheme and by the similarity metric computed between representations.

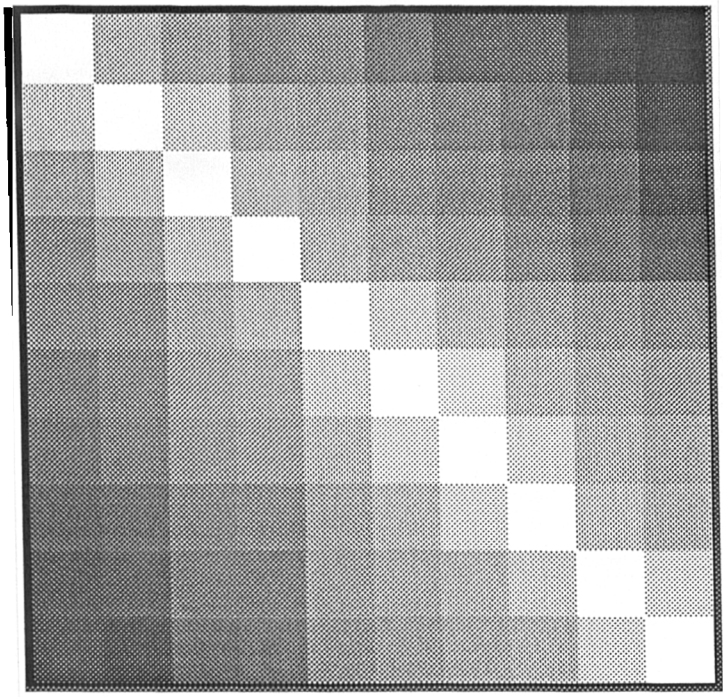

Figure 3–58: The correspondence image for the "morphed" shapes.

A full analysis of the performance of global geometric feature distributions under conditions of variable shape description is not attempted. However, it is possible to consider the extent to which the properties established in Section 3.3 for local representation generalise to the global form of representation. The analysis of the effect of shape fragmentation can be extended straightforwardly to the case of global shape representations; the effect is again to scale the projection of the feature vector representing an image shape in the direction of the corresponding model. Non-correspondence recognition should therefore be robust to the loss of data caused by fragmentation noise. This constitutes an advance over previously proposed global shape representations, such as Fourier coefficients or moment invariants. However, as with these schemes, global geometric feature distributions are adversely affected by the presence of spurious shape primitives arising from scene clutter. In such conditions matching can be expected to break down quickly.

3.6 Discussion and Summary

This chapter has presented a recognition system capable of identifying and locating both 2D and 3D objects from a fixed viewpoint. The system is based on the classification of shape representations in the form of geometric feature distributions. Particular attention was paid to the matching of local shape primitives, a form of recognition suitable in cluttered scenes. This involved a discussion on the appropriate form of similarity metric to be used within a matching scheme based on nearest-neighbour classification. The ability of this scheme to correctly match lines within a shape was demonstrated.

There are a number of advantages to the nature of the processing involved in the proposed scheme, as compared with previous approaches to recognition. Firstly, matches can be established using simple array multiplication. Secondly, elements of the scene can be matched to all object models in parallel. This is in contrast to many other recognition schemes in which each model has to be matched to the scene individually. Together with the strength of the representational scheme, this property means that the proposed system has the potential to provide recognition based on large numbers of objects.

The performance of the combined representation and matching schemes was analysed in conditions where the line description extracted from an image is degraded by various forms of shape variation. In particular, the effects of fragmentation noise, scene clutter and sensor error were investigated. This involved proposing a generative model of each type of shape variation and using these models to analyse, both theoretically and empirically, their effect on both the representational scheme and on the outcome of classification.

It was found that pure shape fragmentation has little effect on the matching of geometric feature distributions. It was also shown that, under the proposed model, the classification of local geometric feature distributions is robust to very high levels of added noise. The factors determining the robustness of matching in conditions where the line description is affected by sensor error were analysed. It was found that the relative position of a line within a shape is crucial in determining the levels of sensor error that can be tolerated before matching based on local geometric feature distributions breaks down. The robustness of matching within a curved shape to changes in line description caused by variable linear approximation and fragmentation was demonstrated. This analysis suggests that the proposed recognition system should be able to successfully operate in conditions where the image shape description is considerably degraded by fragmentation noise, occlusion and scene clutter.

The use of the generalised Hough transform to determine object pose, based on the

line matches provided by the classification of local geometric feature distributions, was presented.

In conditions where objects are encountered in isolation the use of a recognition system based on the matching of global shape representations is entirely appropriate. The ability of global geometric feature distributions to support the matching of whole shapes was illustrated using a set of 12 animal shapes. The ability of the combined representation and matching schemes to capture the similarity relationship between a set of 10 smoothly deforming shapes was also demonstrated.

Chapter 4

SYSTEM DEMONSTRATION

4.1 Introduction

This chapter presents a demonstration of the ability of the proposed recognition system to identify and locate both 2D and 3D objects from their 2D projected shape. The objects are constrained to lie flat on a table and are viewed from directly above, ie. the optical axis of the camera is aligned with the normal to the table. This viewpoint is maintained over both model acquisition and recognition, such that variations in projected shape arising from changes in view direction need not be considered. The object-camera distance is also fixed such that objects appear at a constant scale. This is a restriction which is common to many approaches to 2D object recognition, eg. [52, 3], and leaves the objects free to occur at any 2D position and orientation within the scene. It will be appreciated that these viewing conditions coincide with the particular invariance properties of the representational scheme described in Chapter 2. The application of the GFD scheme to the recognition of objects viewed under more general conditions is addressed in Chapter 5.

The performance of the system is demonstrated using three sets of objects; cut-out dinosaur shapes, industrial parts and a 3D widget. Each of these is intended to highlight a different facet of the recognition system. Specifically, the dinosaur shapes were chosen in order to highlight the ability of the system to recognise arbitrary curved shapes containing a range of complexities. The recognition of industrial parts is included in order to demonstrate that the system is able to deal with more conventional objects containing features such as straight lines and circles. These examples also show that the system is able to deal with large numbers of objects in a scene. Finally, the recognition of a 3D widget from its 2D projection is shown in order to demonstrate that the system is able to deal with problems that occur in "real" image data, such as imperfect edge detection and shadows. The results presented in this chapter provide a practical demonstration of the ability of the system to perform recognition in conditions

of severe fragmentation noise and clutter, and follow directly from the properties of the combined representation and matching schemes established in the previous chapter.

4.2 Dinosaur Recognition

The 5 dinosaur shapes used in this experiment are shown in figure 4–1. These particular shapes were chosen as they demonstrate the ability of the system to deal with arbitrary curved shapes. They also provide wide a range of shape complexity, eg. the "spikes" on the back of **D1** and the smooth "wings" of **D3**. The fact that the objects are planar and black, and are viewed against a light background, means that the description of shape extracted from an image is relatively stable to fragmentation noise. Thus, the main purpose of the examples presented in this section is to demonstrate that the recognition system is able to deal with high degrees of scene clutter and occlusion, although performance under shape fragmentation is also investigated using a simulation of its effect.

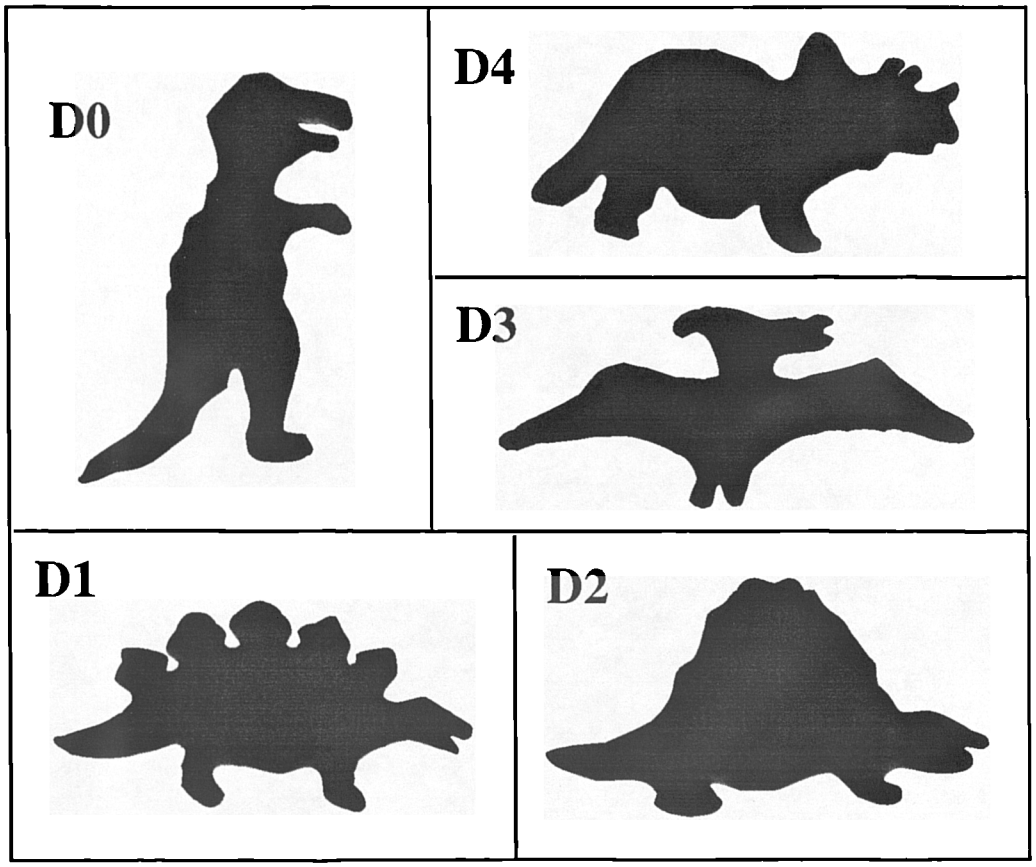

Figure 4–1: The five dinosaur shapes used in recognition.

4.2.1 Procedure

A 512×512 image of each object was captured and processed using a Canny operator to extract a set of edgel strings describing the shape of the object. A linear approximation process was then applied to the edgel strings to give a line-based shape description. This required, on average, about 60 lines per object. These were then used to construct models of each object in the form of local geometric feature distributions, which involved storing a histogram for each line within each shape. The histograms used in these experiments had parameters $n_\theta = 40$, $n_d = 30$, $\sigma_\theta = \sigma_d = 1.0$, while the radius of the circular local region was was set to 50 pixels. The relative size of this region as compared with object **D4** is shown in figure 4–2.

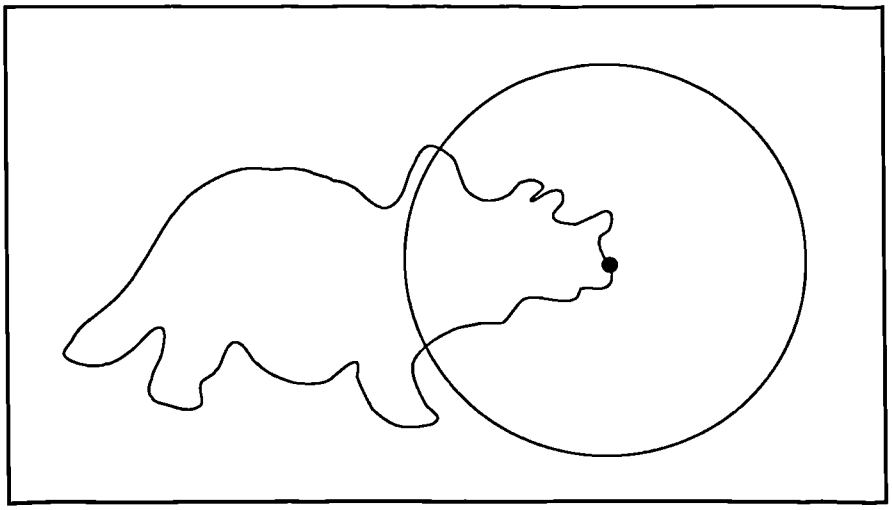

Figure 4–2: The circular region for a line in **D4**.

4.2.2 Demonstrating Performance

The performance of the recognition system is demonstrated in two ways.

Graphical Illustration.
A graphical illustration is provided of the ability of the system to correctly match line segments and to locate, on the basis of these matches, the objects present in the scene. This is involves showing results at three levels of processing:

i) an image of the scene containing the objects.

ii) the lines describing the shape in the image coloured according to the object to which they are matched. The colour coding scheme is as follows, **D0** - green , **D1** - yellow, **D2** - blue, **D3** - red, **D4** - orange. This is useful in demonstrating that the matching scheme is able to correctly segment the scene lines belonging to each object.

iii) the located objects projected into the image, using the same colour coding scheme. The pose of each object was determined using the generalised Hough transform, as described in Section 3.4.

Quantitative Assessment.
A set of quantitative measures of the performance of the system are provided. Providing statistics on the accuracy of matching or on the precision of localisation is not possible, since the information needed to compute them is not available from real image data. This problem was overcome to some extent by analysing the number and distribution of entries in the Hough space associated with each object.

According to the scheme described in Chapter 3, the number of votes in Hough space indicate the number of pairs of matched scene lines that are pairwise geometric consistent and which correspond a uniform transformation of the object. This is quite a stringent test, and so high numbers of votes indicate that a large proportion of the line matches are correct. Also of interest is the relative values of correct and incorrect peaks in the Hough space, as this determines the ease with which objects can be robustly located. Thus, the absolute and normalised peak values in the Hough space associated with each objects are shown. This serves to demonstrate whether the peak values for correct localisations are significantly higher than those resulting from incorrect matches. Indeed, the normalised peak value is used as a terminating condition, the process of peak detection being stopped once its value falls below a particular level. It was found that a value of 0.1 was, in the vast majority of cases, able to distinguish between correct and incorrect peaks. The proportion of projected model lines that receive support for each hypothesis is also shown. This provides a good measure of the accuracy of the localisation, although care must be taken in interpreting the values arising from scenes containing occluded objects.

4.2.3 Examples of Recognition

The performance of the recognition system was demonstrated for different combinations of objects viewed under various conditions. Each example was generated by placing objects at different positions and orientations within the scene and is intended to show a particular characteristic of the recognition system. Five examples are shown:

Example 1.1 - Multiple objects.

Example 1.2 - Multiple objects plus occlusion.

Example 1.3 - Multiple objects plus severe occlusion.

Example 1.4 - Multiple instances of a single object plus occlusion.

Example 1.5 - Multiple objects with unknown objects occluding.

These examples are now presented.

Example 1.1 - Multiple objects

This example demonstrates the ability of the system to perform basic recognition on a scene containing a single instance of each object. The image of the scene is shown in figure 4–3. Table 4–1 shows the results of localisation based on the generalised Hough transform. It can be seen that a significant number of votes are made in the Hough space associated with each object, indicating that a large proportion of the matches produced by the system are correct. This is confirmed by an examination of the colour-coded segmentation shown in figure 4–4.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
D0	710	1	2426	1.0	76
		2	1	0.0	-
D1	755	1	1256	1.0	60
		2	1	0.0	-
D2	478	1	1659	1.0	80
		2	0	0.0	-
D3	661	1	2648	1.0	79
		2	0	0.0	-
D4	558	1	1471	1.0	71
		2	0	0.0	-

Table 4–1: The results of localisation for example 1.1

As described in Chapter 3, the process of locating objects in Hough space continues until the normalised peak value falls below some threshold. It can be seen from the values in Table 4–1 that the Hough space of each object contains only a single significant peak, corresponding to the correct localisation of the object in the scene. This provides further evidence of the quality of the matches produced by classifying local geometric feature distributions, although the test applied prior to voting obviously rules out a significant proportion of any incorrect matches. The accuracy of the localisations can be gauged by examining the projection of the objects onto the image of the scene, figure 4–5.

Ex. 1.1

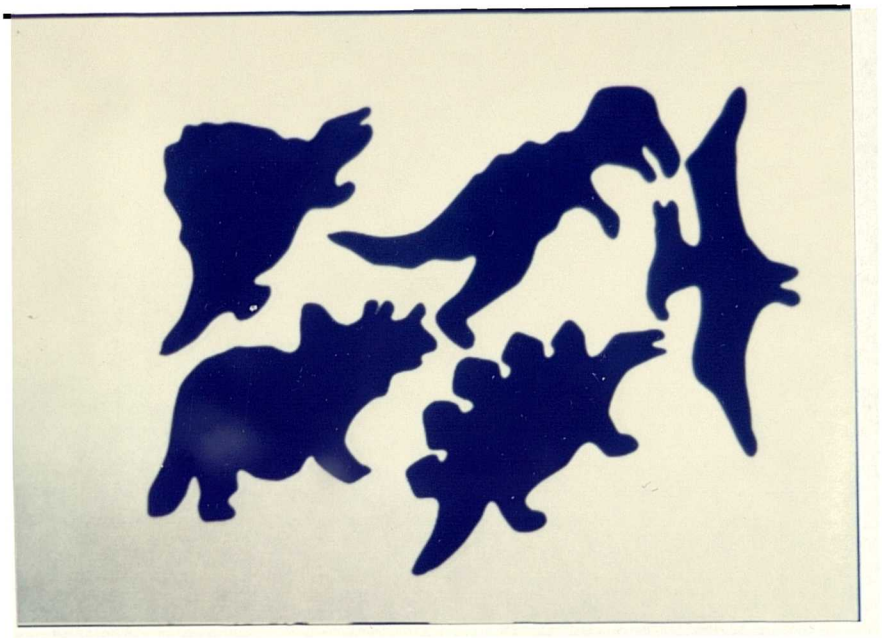

Figure 4–3: Example 1.1 - An image of the scene.

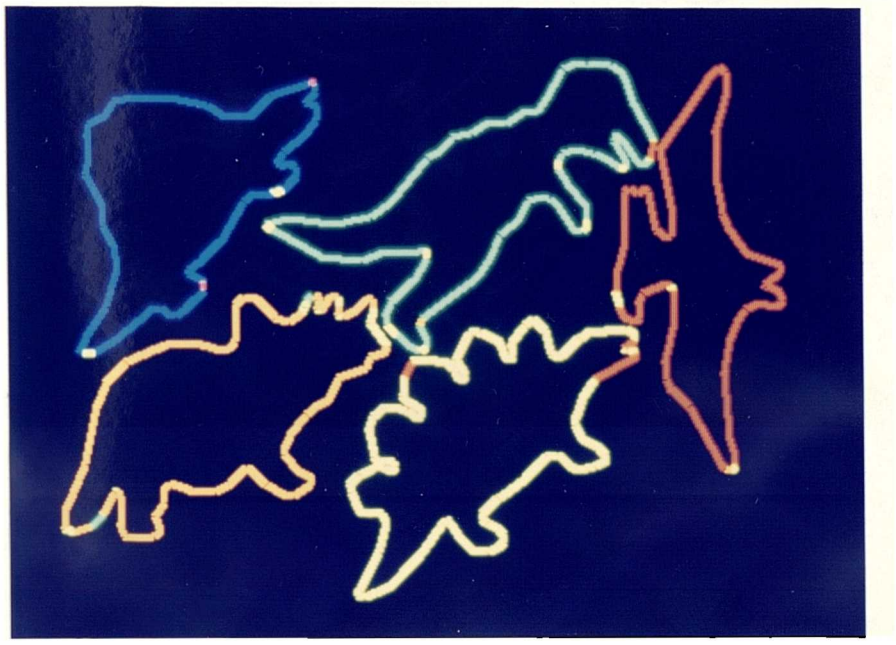

Figure 4–4: Example 1.1 - A colour-coded segmentation of the scene lines.

Figure 4–5: Example 1.1 - The located object(s) projected into the image.

Example 1.2 - Multiple objects plus occlusion

This example demonstrates the ability of the system to recognise objects in a jumbled scene where objects occlude one another. The scene contains a single instance of each object. An image of the scene is shown in figure 4–6. The difficulty of performing recognition in this particular scene is considerable, since certain objects have as much as half of their shape missing through occlusion. Table 4–2 presents the results of localisation. It can be seen that there is a drop in the number of votes made in each Hough space, as compared with the previous example. This is explained by the reduction in the number of lines in the scene caused by occlusion. However, the quality of the matches is not seriously affected by the occlusion, as can be seen from the colour-coded segmentation shown in figure 4–7. This is due in part to the use of the circular local region to restrict the range over which geometric relationships are measured, since it ensures that the effect of occlusion on the representation of lines in the visible parts of the objects is minimised.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
D0	128	1	522	1.0	28
		2	1	0.0	-
D1	176	1	442	1.0	34
		2	1	0.0	-
D2	200	1	621	1.0	58
		2	0	0.0	-
D3	17	1	140	1.0	21
		2	0	0.0	-
D4	99	1	222	1.0	36
		2	0	0.0	-

Table 4–2: The results of localisation for example 1.2

It can also be seen from Table 4–2 that the Hough space of each object again contains only a single significant peak, corresponding to the correct localisation of the object. The accuracy of this localisation can be seen in figure 4–8, which shows the projection of the located objects into the image. It can also be seen that the proportion of projected model lines receiving local support is reduced from example 1.1. Again this is due to the occlusion. This example illustrates the difficulty of basing terminating conditions on the amount of support received by an hypothesised set of transformation parameters.

Ex. 1.2

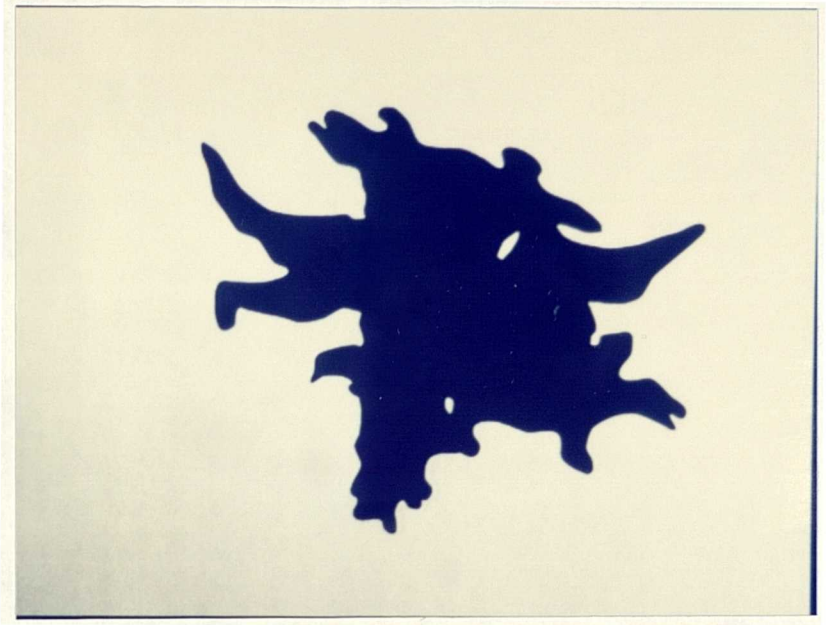

Figure 4–6: Example 1.2 - An image of the scene.

Figure 4–7: Example 1.2 - A colour-coded segmentation of the scene lines.

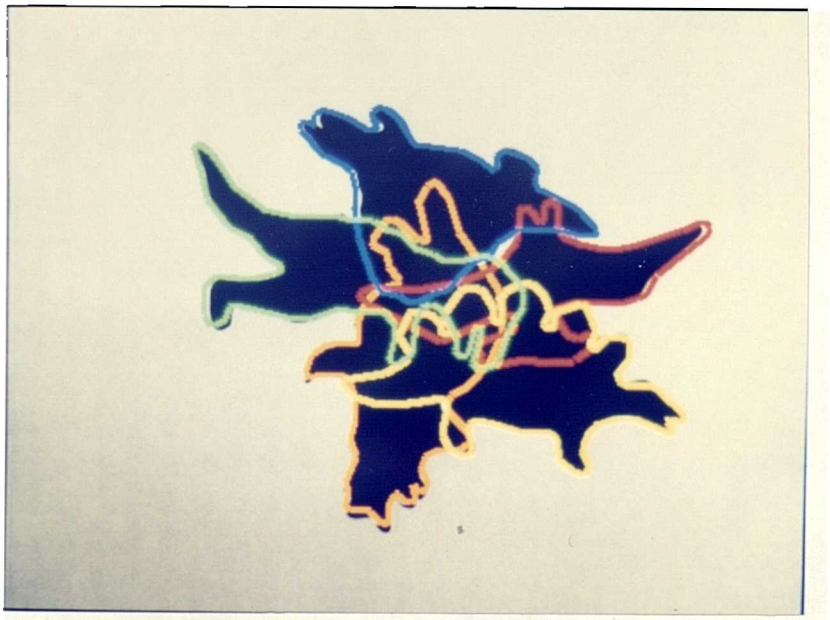

Figure 4–8: Example 1.2 - The located object(s) projected into the image.

Example 1.3 - Multiple objects plus severe occlusion

This example presents a more severely jumbled scene containing two instances of each object. The image of the scene is shown in figure 4–9. That correct line matching is preserved in this more severe case of occlusion can be seen from the colour-coded segmentation of the scene lines shown in figure 4–10.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
D0	301	1	562	1.0	42
		2	327	0.58	33
		3	1	0.0	-
D1	190	1	313	1.0	34
		2	121	0.39	26
		3	0	0.0	-
D2	241	1	815	1.0	52
		2	84	0.10	34
		3	0	0.0	-
D3	570	1	1430	1.0	63
		2	556	0.39	55
		3	0	0.0	-
D4	305	1	654	1.0	44
		2	250	0.38	47
		3	0	0.0	-

Table 4–3: The results of localisation for example 1.3

Table 4–3 presents the results of localisation. It can be seen that the Hough space associated with each object contains two significant peaks, corresponding to the two instances of each object in the scene. The particularly low value of the second peak for object **D2** is explained by the fact that only a small region of the rear of the object is visible, (the object appears in the lower right hand portion of the image). The accuracy of the localisations can be seen from the projection of the models into the image, as shown in figure 4–11.

Ex. 1.3

Figure 4–9: Example 1.3 - An image of the scene.

Figure 4–10: Example 1.3 - A colour-coded segmentation of the scene lines.

Figure 4–11: Example 1.3 - The located object(s) projected into the image.

Example 1.4 - Multiple instances of a single object plus occlusion

This example demonstrates the ability of the system to recognise multiple instances of a single object in a scene. An image of the scene, which contains 10 instances of object **D3**, is shown in figure 4–12. The fact that the objects are placed at random positions and orientations in the scene means that the problem of occlusion must also be overcome. The recognition of multiple instances of a single object is handled naturally in the proposed classification scheme, since multiple scene lines may be matched, in parallel, with the same model line. This can be seen from the colour-coded segmentation shown in figure 4–13. Certain other recognition schemes, eg. tree-search, have to address this problem by matching the model to the scene for each instance of the object.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
D0	0	1	0	-	-
D1	1	1	1	-	-
D2	0	1	0	-	-
D3	1863	1	1310	1.0	55
		2	1209	0.92	62
		3	991	0.76	56
		4	880	0.67	58
		5	864	0.66	51
		6	568	0.43	44
		7	483	0.37	37
		8	449	0.34	44
		9	258	0.20	48
		10	183	0.14	43
		11	1	0.0	-
D4	0	1	0	-	-

Table 4–4: The results of localisation for example 1.4

Another reason for testing the system on a scene containing multiple instances of a single object was to test the ability of the system to recognise the fact that certain known objects were not present. That this is the case can be seen from the results of localisation, shown in Table 4–4. It can be seen that there are no votes for objects **D0**,**D2** and **D4**, and only a single vote for **D1**, while there are a large number of votes for **D3**. While the test applied before making each vote does play a part in ensuring that any invalid matches are not reflected in Hough space, this result, together with the colour-coded segmentation, indicates that the vast majority of the matches are correct. It can also be seen from Table 4–4 that there are 10 significant peaks in the Hough space of object **D3**, corresponding to the ten instances of the object in the scene. The projection of the located objects into the image is shown in figure 4–14.

Ex. 1.4

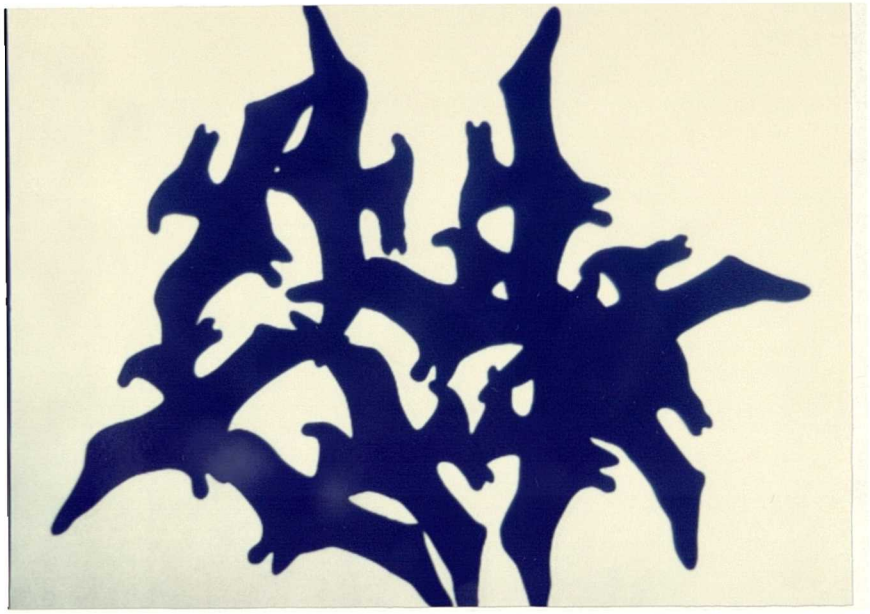

Figure 4–12: Example 1.4 - An image of the scene.

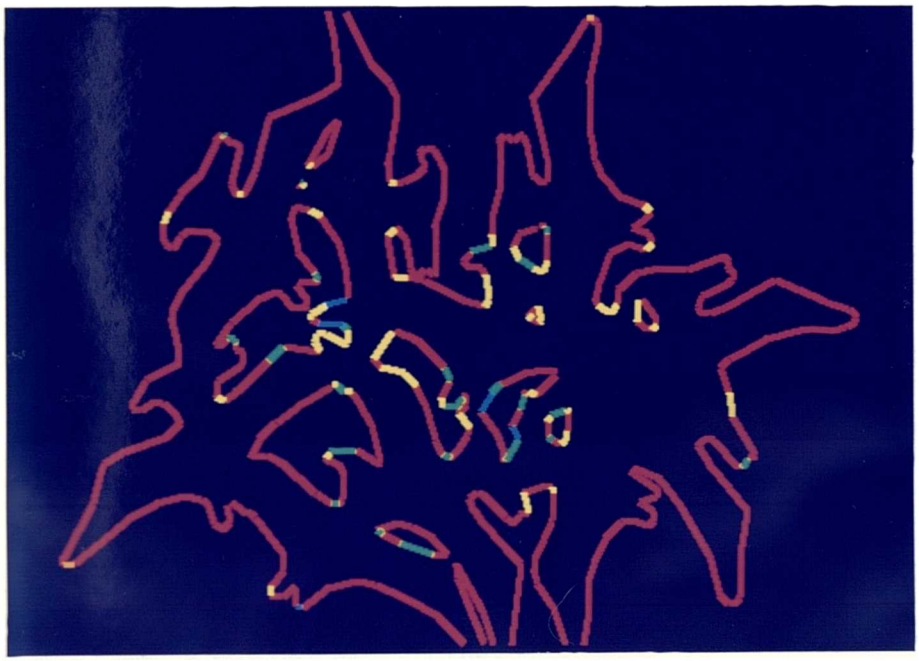

Figure 4–13: Example 1.4 - A colour-coded segmentation of the scene lines.

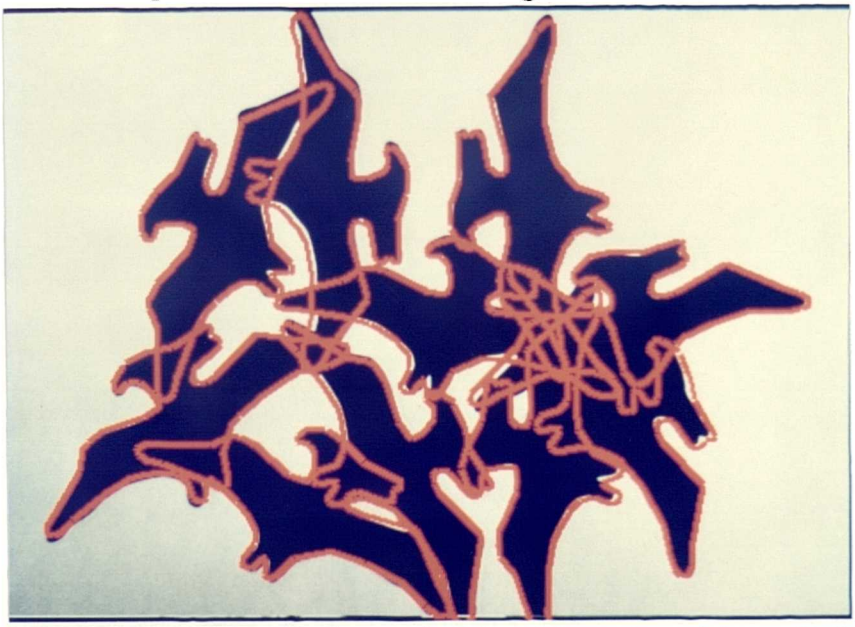

Figure 4–14: Example 1.4 - The located object(s) projected into the image.

Example 1.5 - Multiple objects with unknown objects occluding

In all four previous examples the lines extracted from the scene have been due to a known object. The purpose of this example is to demonstrate that the proposed system is able to deal with the presence of unknown, possibly occluding, objects in the scene, where an object is unknown if the system possesses no model for it. An image of a scene containing an instance of each dinosaur shape, partially occluded by a series of human shapes, (these served as the unknown objects), is shown in figure 4–15. It was established in Chapter 3 that the matching of local geometric feature distributions is very robust to the presence of spurious lines in an image. This suggests that matching in this scene should be preserved. That this is the case can be confirmed by examining the colour-coded segmentation of the scene lines, shown in figure 4–16. It can be seen that the visible contours of each object are, on the whole, correctly matched, despite the loss of shape information arising from occlusion and the presence of spurious lines due to the unknown objects.

Object	*Results*				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
D0	321	1	1060	1.0	70
		2	23	0.02	-
D1	404	1	732	1.0	63
		2	6	0.01	-
D2	88	1	366	1.0	68
		2	0	0.0	-
D3	325	1	1299	1.0	67
		2	1	0.0	-
D4	176	1	481	1.0	65
		2	0	0.0	-

Table 4–5: The results of localisation for example 1.5

Of course, the local nature of the matching scheme means that each spurious image line is matched to a particular model line. One of the reasons for using the generalised Hough transform to locate objects was to provide a robust method of dealing with these spurious matches by applying, via pose clustering, a global constraint to the set of matches. The results of localisation for this scene are shown in Table 4–5. The fact that the Hough space associated with each object contains only a single significant peak suggests that the test for pairwise geometric consistency applied prior to voting is very effective in ruling out spurious matches. This calls into question whether the clustering aspect of the Hough transform is actually needed, and whether some other, less expensive method could be used. This is an area for further study. The projection of the located objects into the image of the scene is shown in figure 4–17.

Ex. 1.5

Figure 4–15: Example 1.5 - An image of the scene.

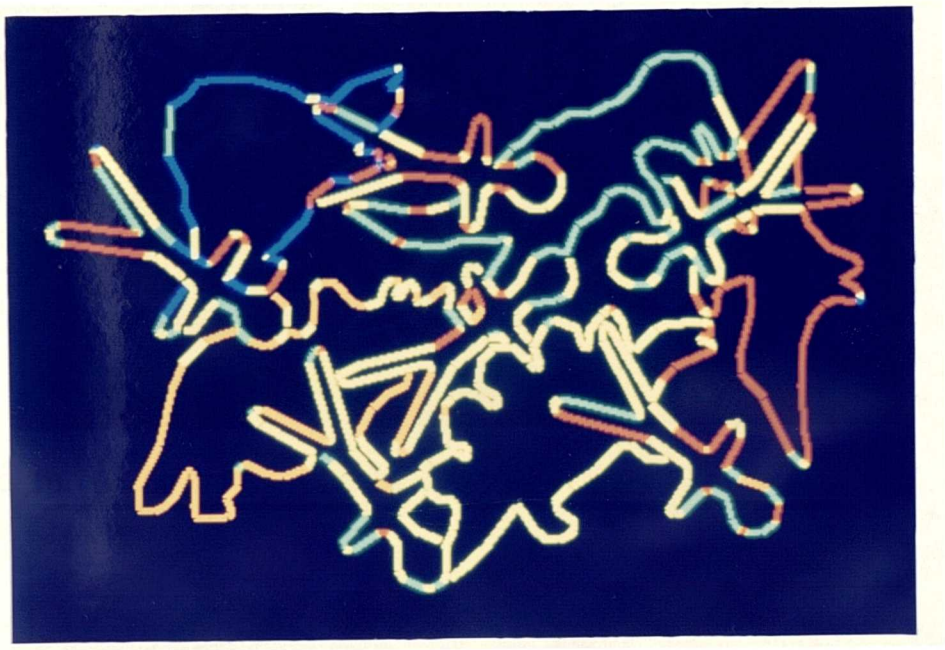

Figure 4–16: Example 1.5 - A colour-coded segmentation of the scene lines.

Figure 4–17: Example 1.5 - The located object(s) projected into the image.

4.2.4 Simulating the Effects of Fragmentation Noise

As stated above, the fact that the dinosaur shapes are planar and black, and are viewed against a light background, means that edge detection is very robust. The ability of the system to deal with shape fragmentation has not therefore been tested. This was overcome to some extent by simulating the effects of fragmentation noise. This involved fragmenting the edgel strings extracted for the scene by removing continuous sections at random intervals, such that the 50% of the original edgels were remaining. A linear approximation of the fragmented edgel strings was then performed. This model could be criticised on the grounds that its effects are uniformly distributed across the visible sections of an object. Also, the orientation of line segments was not explicitly changed. However, it is likely that the combination of the shape fragmentation and the re-application of the linear approximation algorithm will result in a small change in line orientation.

Recognition based on fragmented shape descriptions is demonstrated in examples 1.6,1.7 and 1.8. These were obtained by applying the above process to the shape descriptions extracted in examples 1.3, 1.4 and 1.5 respectively. In each example results are shown at two levels of processing:

i) A colour-coded segmentation of the fragmented line description, using the same colour scheme as above.

ii) The located objects projected onto the the fragmented line description.

Example 1.6 - Fragmented version of example 1.3

The effect of applying the model of fragmentation noise to the scene lines extracted in example 1.3, (which showed a jumbled scene containing two instances of each object), is shown in figure 4–18. This also represents the colour-coded segmentation of the fragmented lines. From this it can be seen that correct matching is, on the whole, preserved, despite the loss of shape information arising from fragmentation. This result follows directly from the analysis presented in Chapter 3, which showed that the matching of local geometric feature distributions is theoretically robust to shape fragmentation.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
D0	107	1	168	1.0	30
		2	77	0.46	17
		3	0	0.0	-
D1	24	1	45	1.0	19
		2	15	0.32	16
		3	1	0.03	-
D2	51	1	118	1.0	28
		2	6	0.05	18
		3	0	0.0	-
D3	137	1	304	1.0	36
		2	79	0.26	37
		3	0	0.0	-
D4	46	1	65	1.0	28
		2	24	0.37	22
		3	0	0.0	-

Table 4–6: The results of localisation for example 1.6

The results of localisation for the fragmented scene lines is shown in Table 4–6. It can be seen that while the number of votes made in the Hough space associated with each object is reduced, due to the reduction in the number of scene lines caused by fragmentation, the presence of dual peaks corresponding to each instance of the object is preserved. A slight problem occurs in the low significance of the second peak for object **D2**. This is due to the fact that only a very small number of fragments of the object remain. However, it can be seen from the projection of the object at the position and orientation corresponding to this peak, shown along with the set of valid localisations in figure 4–19, that it is correct, despite its low relative size. Finding a reliable method of distinguishing between peaks of low significance that are due to a small number of valid shape lines and those due to background noise is a fundamental problem in any proposed recognition system, and is not due to any particular weakness of the current system.

Example 1.6

Figure 4–18: Example 1.6 - A colour-coded segmentation of the fragmented lines.

Figure 4–19: Example 1.6 - The located object(s) projected onto the fragmented lines.

Example 1.7 - Fragmented version of example 1.4

The fragmented scene description of example 1.4, (which showed 10 instances of object **D3**), is shown in figure 4–20. This also shows the colour-coded segmentation of the fragmented lines. Again it can be seen that the shape fragmentation has little or no effect on the ability of the system to correctly match line segments. The results of localisation based on these matches is shown in Table 4–7. It can be seen that there are again 10 significant peaks in the Hough space associated with object **D3**, corresponding to the ten instances of the object, while the Hough spaces associated with all other objects receive no votes. This further demonstrates the ability of the system to deal robustly with shape fragmentation.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
D0	0	1	0.0	-	-
D1	0	1	0.0	-	-
D0	0	1	0.0	-	-
D3	592	1	266	1.0	45
		2	243	0.91	28
		3	220	0.83	22
		4	191	0.72	43
		5	161	0.60	40
		6	109	0.41	24
		7	93	0.35	36
		8	45	0.17	20
		9	44	0.17	20
		10	36	0.14	37
		11	0	0.0	-
D4	0	1	0.0	-	-

Table 4–7: The results of localisation for example 1.7

Example 1.7

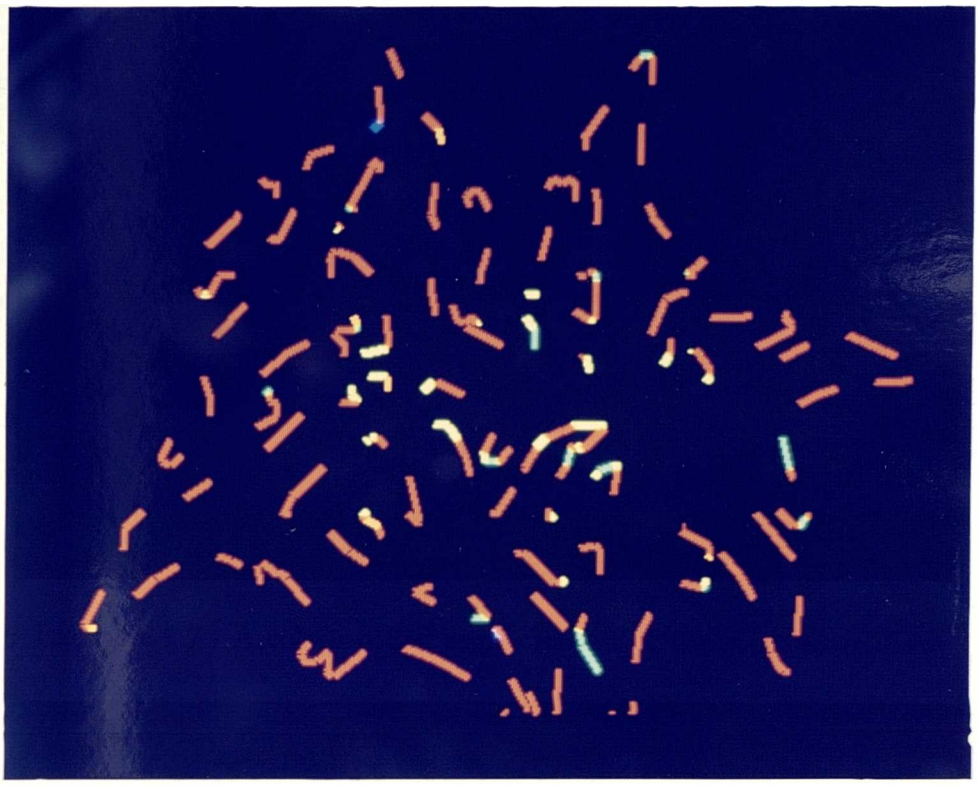

Figure 4–20: Example 1.7 - A colour-coded segmentation of the fragmented lines.

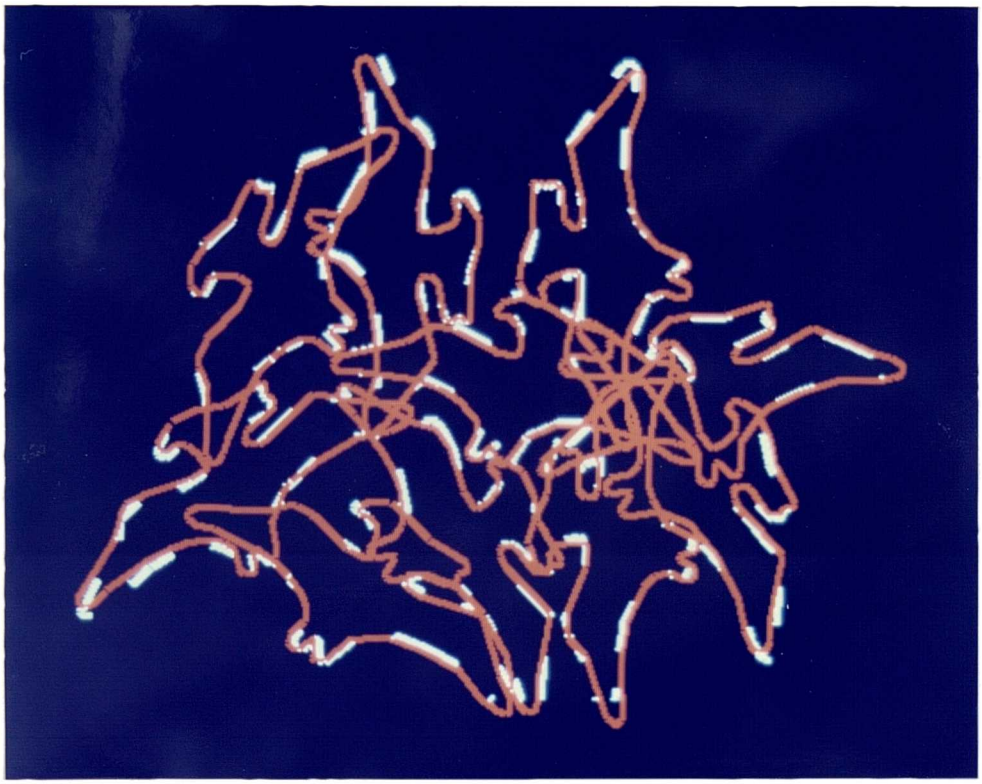

Figure 4–21: Example 1.7 - The located object(s) projected onto the fragmented lines.

Example 1.8 - Fragmented version of example 1.5

The effect of applying the model of fragmentation noise to the scene description of example 1.5, (which showed an instance of each object with unknown objects occluding), is shown in figure 4–22. This also represents the colour-coded segmentation of the fragmented lines. This example represents a significant challenge to any recognition system, since it contains examples of occlusion, fragmentation noise and scene clutter, ie. lines due to unknown objects. However, it can be seen that a large proportion of the matches are maintained. The results of localisation are shown in figure 4–8. This shows that while the number of votes made in the Hough space associated with each object is often quite small, eg. 10 in the case of **D4**, each space contains only a single significant peak, corresponding to the single instance of each object in the scene. The projection of the localised objects onto the fragmented line description is shown in figure 4–23.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
D0	33	1	158	1.0	44
		2	1	0.01	-
D1	42	1	122	1.0	30
		2	3	0.03	-
D2	17	1	62	1.0	36
		2	0	0.00	-
D3	63	1	268	1.0	41
		2	11	0.04	-
D4	10	1	46	1.0	37
		2	0	0.00	-

Table 4–8: The results of localisation for example 1.8

Example 1.8

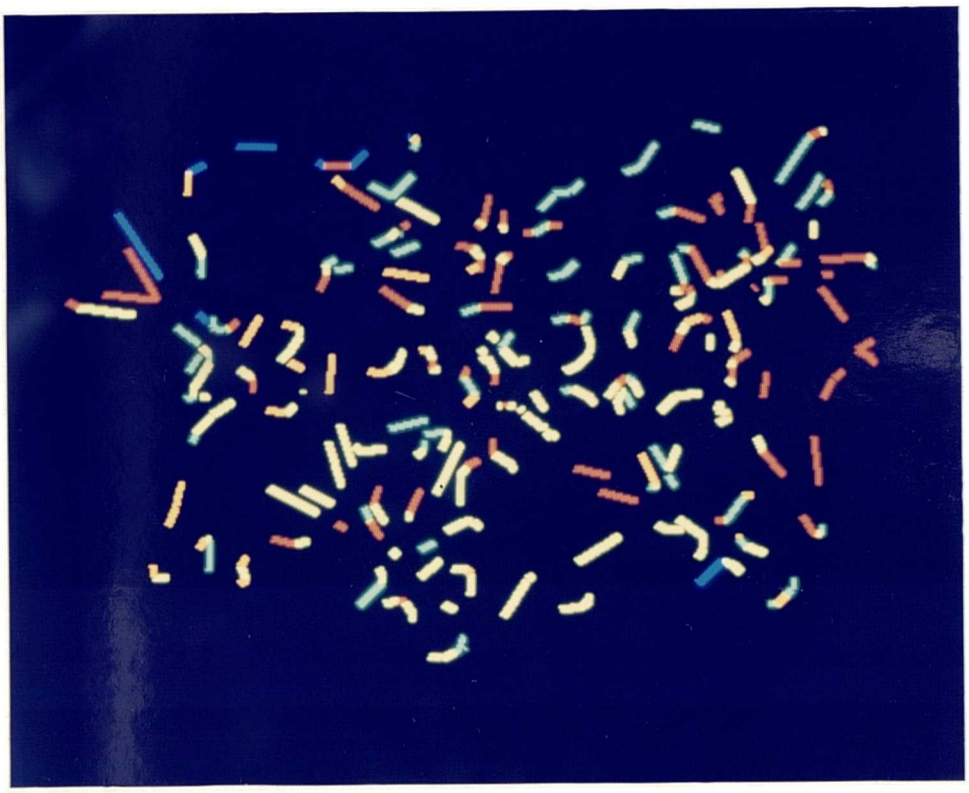

Figure 4–22: Example 1.8 - A colour-coded segmentation of the fragmented lines.

Figure 4–23: Example 1.8 - The located object(s) projected onto the fragmented lines.

4.3 Industrial Part Recognition

This section demonstrates the recognition of a set of 4 parts from the mechanism of a typewriter, figure 4–24. These are based upon objects shown in Grimson [40]. The purpose of showing recognition on these shapes is to demonstrate that the system is able to deal with conventional objects, containing features such as long lines and circular arcs. Furthermore, the reduced size of the objects means that many more instances of each object can be included in a scene, further demonstrating the ability of the system to operate in conditions of severe scene clutter and occlusion. However, the fact that the objects are planar and black means that edge detection should again be stable.

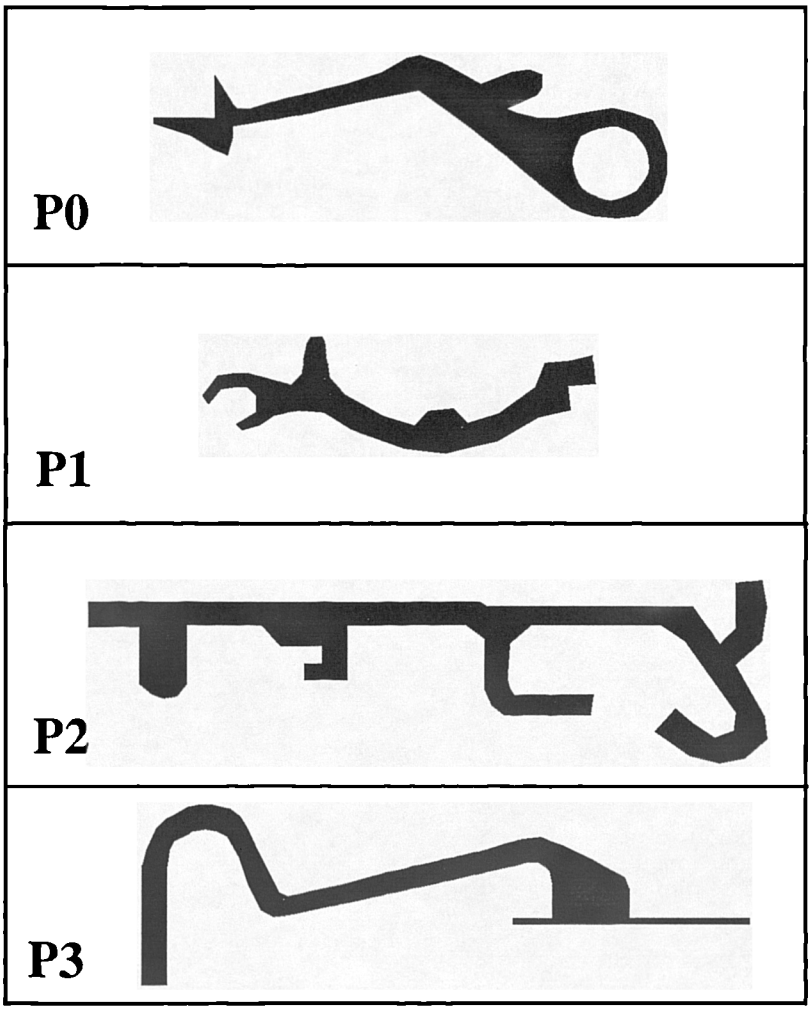

Figure 4–24: The four industrial parts used in recognition.

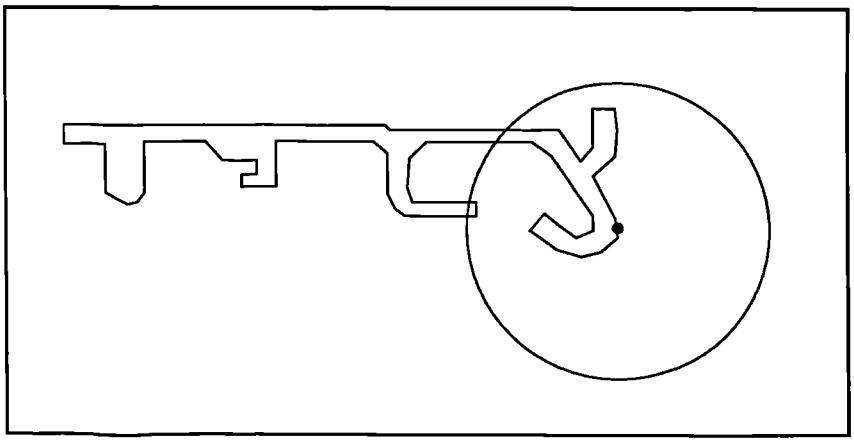

Figure 4–25: The circular region for a line in **P2**.

The performance of the recognition system on these shapes was tested using the same procedure as used in the previous section, using histograms with the same parameters. In order to gain satisfactory performance it was found necessary to reduce the radius of the local region to 30 pixels. The size of this region relative to object **P2** is shown in figure 4–25.

Recognition was demonstrated for two cases.

Example 2.1 - Multiple objects plus severe occlusion.
Example 2.2 - Multiple instances of a single object plus occlusion.

These examples are now presented.

Example 2.1 - Multiple objects plus severe clutter

This example demonstrates the ability of the system to recognise objects in a severely cluttered scene. An image of the scene is shown in figure 4–26. The scene contains 5 instances of **P0**, 6 of **P1**, 4 of **P2** and 4 of **P3**. It was envisaged that these particular objects might cause certain problems due to the fact that they contain elongated sections described by long, straight lines that provide relatively little shape information. However, it can be seen from the colour-coded segmentation of the scene lines, shown in figure 4–27, that the lines belonging to each object are, on the whole, correctly matched. The results of localisation for this scene are shown in Table 4–9.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
P0	565	1	1457	1.0	55
		2	1426	0.98	52
		3	952	0.65	69
		4	388	0.27	57
		5	168	0.12	57
		6	9	0.01	-
P1	665	1	1200	1.0	63
		2	1106	0.92	66
		3	962	0.80	68
		4	299	0.25	58
		5	256	0.21	55
		6	229	0.19	52
		7	7	0.01	-
P2	625	1	3417	1.0	60
		2	1636	0.48	61
		3	1368	0.40	53
		4	487	0.14	51
		5	190	0.06	-
P3	234	1	4610	1.0	65
		2	1993	0.43	58
		3	528	0.11	49
		4	496	0.11	49
		5	77	0.02	-

Table 4–9: The results of localisation for example 2.1

It can be seen that the Hough space associated with each object contains the appropriate number of significant peaks, given the number of instances of each object in the scene, ie. **P0**(5), **P1**(6), **P2**(4), **P3**(4). Although a number of the lower ranked peaks are only just above significance, eg. for object **P3**, this again is due to the small number of actual scene lines describing the instance of the object, (cf. example 1.8). However, the accuracy of the localisation can be appreciated by examining the projection of the located objects into the image of the scene, shown in figure 4–28.

Ex. 2.1

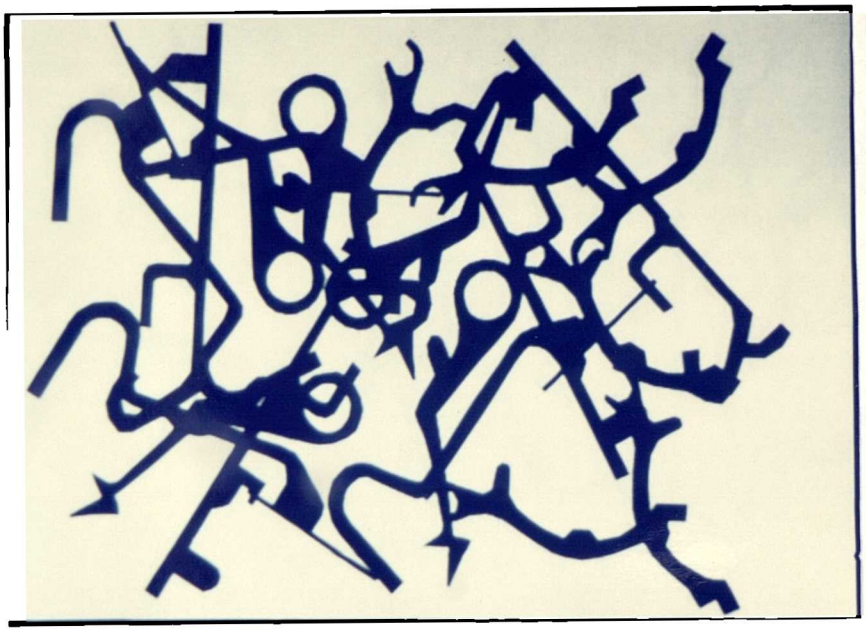

Figure 4–26: Example 2.1 - An image of the scene.

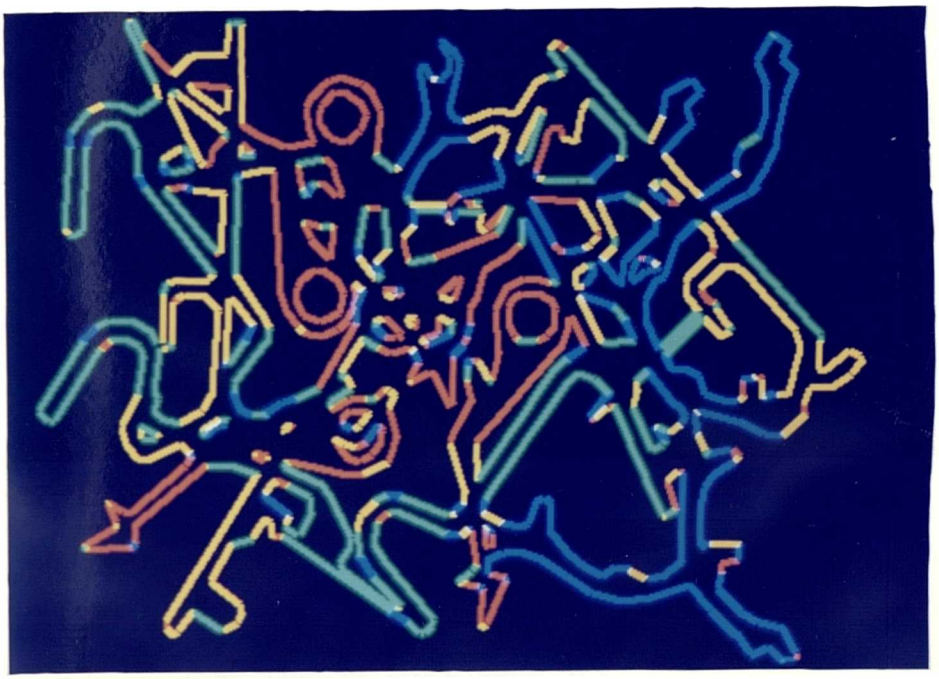

Figure 4–27: Example 2.1 - A colour-coded segmentation of the scene lines.

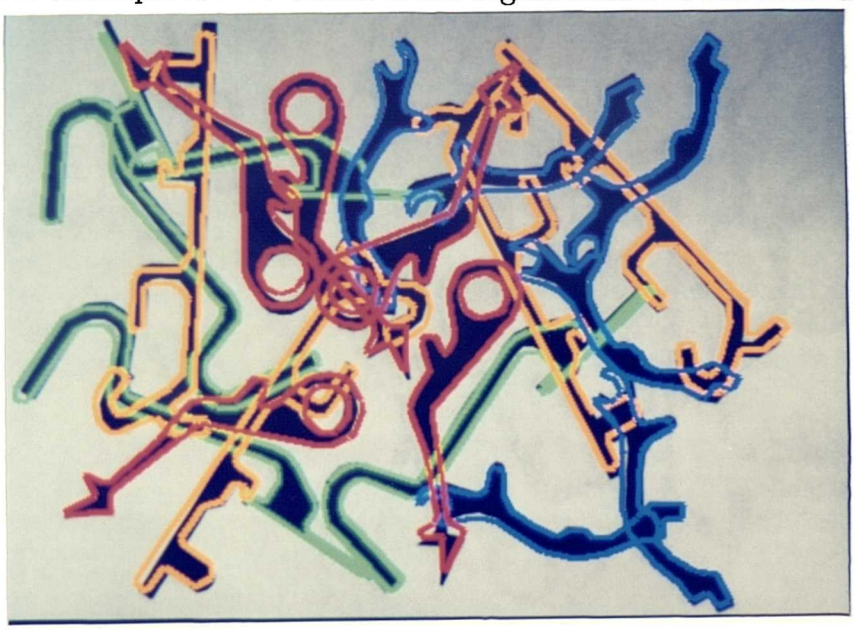

Figure 4–28: Example 2.1 - The located object(s) projected into the image.

Example 2.2 - Multiple instances of a single object plus occlusion

This example demonstrates the ability of the system to recognise multiple overlapping instances of the same object. As mentioned above, the reduced size of the industrial parts, as compared with dinosaurs, means that it is possible to include more objects in a scene. An image of a scene containing 15 instances of object **P3** is shown in figure 4–29. It can be seen that there is a significant level of occlusion in this scene. However, the colour-coded segmentation of lines in the scene, shown in figure 4–30, illustrates that matches are, on the whole, correct.

Object	Results				
	Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
P0	0	1	0	-	-
P1	0	1	0	-	-
P2	6	1	23	1.0	
P3	1233	1	6108	1.0	73
		2	4476	0.73	43
		3	3251	0.53	59
		4	2962	0.48	51
		5	2752	0.45	67
		6	2533	0.41	43
		7	2326	0.38	64
		8	2305	0.38	54
		9	1548	0.25	62
		10	1491	0.24	62
		11	1247	0.20	64
		12	1160	0.19	62
		13	1042	0.17	40
		14	907	0.15	46
		15	848	0.14	51
		16	405	0.07	-

Table 4–10: The results of localisation for example 2.2

The results of localisation for this scene are shown in Table 4–10. It can be seen that the Hough spaces associated with objects **P0**, **P1** and **P2**, which are not in the scene, receive very few votes, while that of object **P3** receives a large number. Although the Hough space of object **P2** receives 6 votes, the small size of the primary peak means that no hypothesis verification is attempted. It can also be seen that the Hough space of object **P3** contains 15 significant peaks, corresponding to the 15 instances of the object in the scene. The accuracy of these localisations associated with each of these peaks can be appreciated by examining the projection of the objects into the image of the scene, shown in figure 4–31.

Ex. 2.2

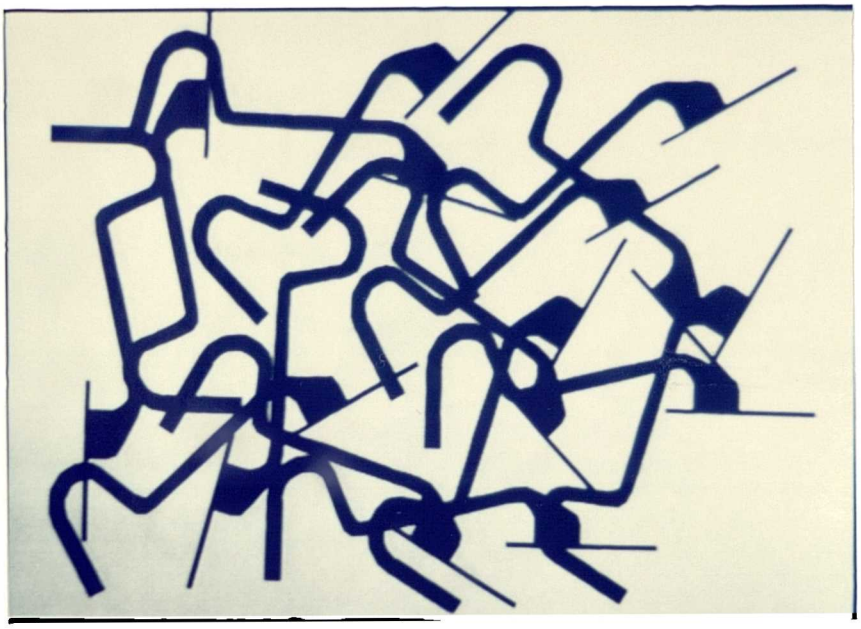

Figure 4–29: Example 2.2 - An image of the scene.

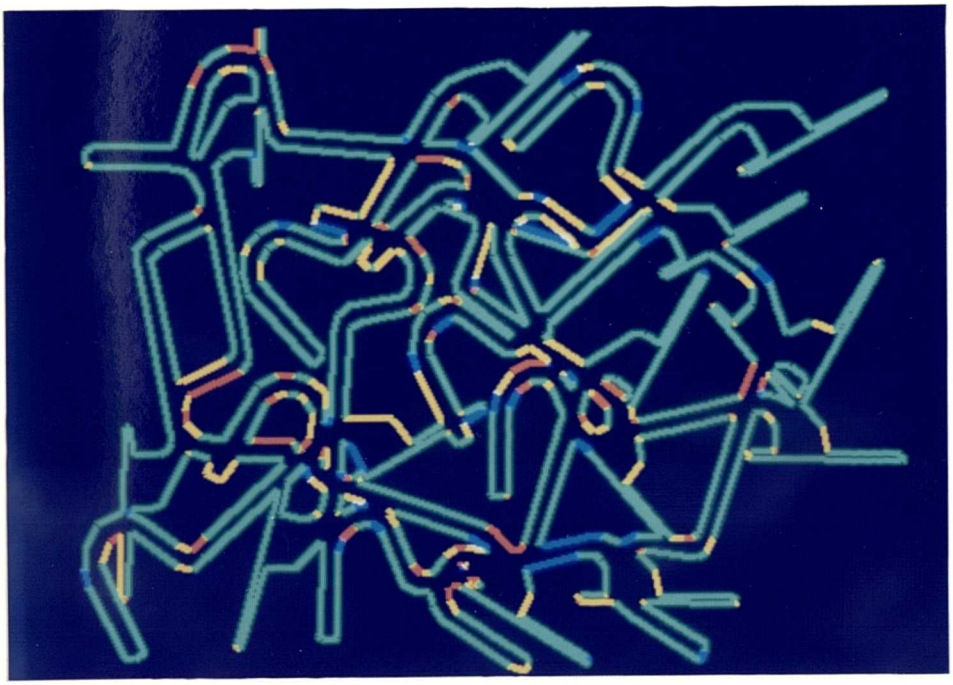

Figure 4–30: Example 2.2 - A colour-coded segmentation of the scene lines.

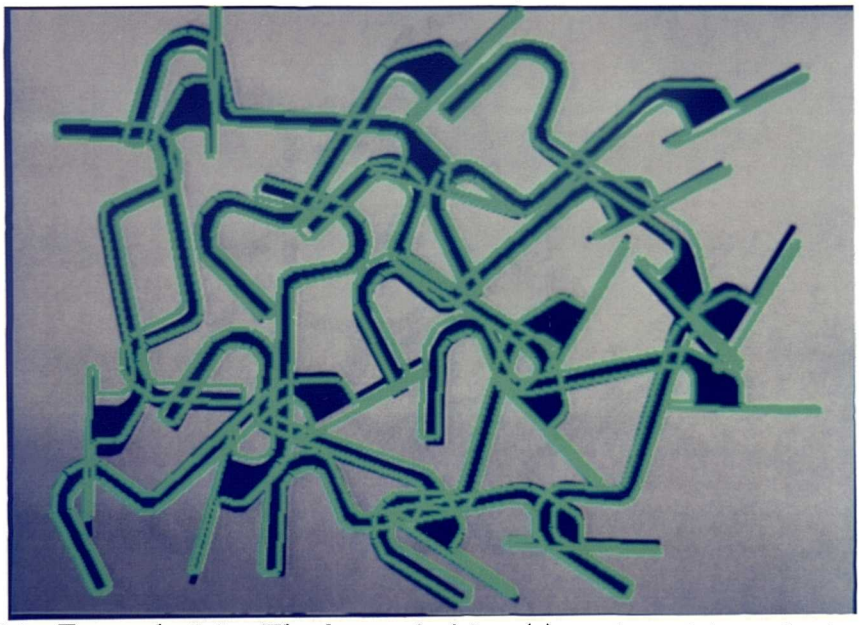

Figure 4–31: Example 2.2 - The located object(s) projected into the image.

4.4 Projection of a 3D Object

It could be argued that the previous examples do not provide a very severe test of the recognition system, since the shape description extracted from the image of black, 2D planar objects viewed against a white background is very stable. The shape description extracted from the image of a 3D object on the other hand will typically be affected by problems such as shadows and shape fragmentation due to changes in the amount of incident light falling on adjacent 3D surfaces. The purpose of this section is to demonstrate that the proposed recognition system is able to deal with such problems.

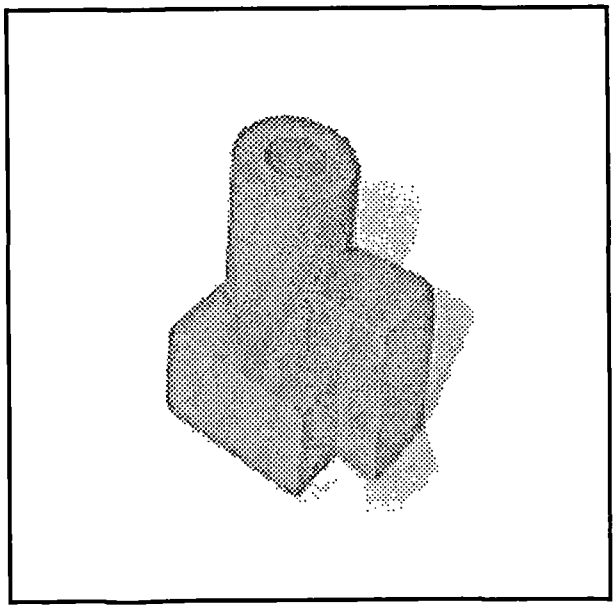

Figure 4–32: The 3D object used in the demonstration.

The 3D object shown in figure 4–32 was viewed from directly above in favourable lighting conditions and the resulting image processed to give a line-based description of its shape. This was used as a model of the object. The histograms used in this demonstration had parameters $n_\theta = 30$, $n_d = 20$, $\sigma_\theta = \sigma_d = 1.0$. This is a little coarser than in the previous examples and was found to be necessary in order to overcome the variation in shape description. The radius of the circular local region was set to 40 pixels, as shown in figure 4–33.

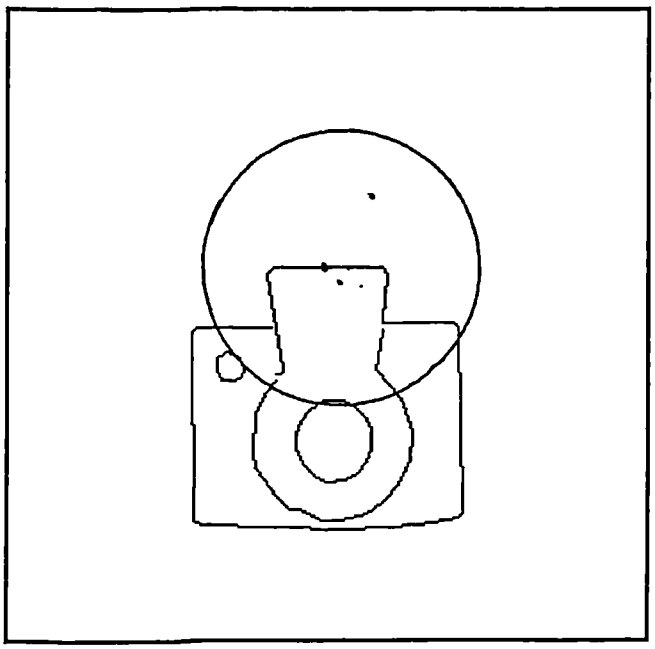

Figure 4–33: The circular region for a line in the object.

The performance of the recognition system was demonstrated for a number of example scenes:

Example 3.1 - A single object.

Example 3.2 - A pair of objects.

Example 3.3 - A pair of objects in a cluttered scene.

In each case results are shown at three levels of processing:

i) an image of the scene containing the object(s).

ii) the lines describing the shape in the image.

iii) the located object(s) projected into the image.

These examples are now shown.

Example 3.1 - A single object

This example provides a demonstration of the ability of the recognition system to perform basic recognition on a scene containing a single instance of the object. An image of the scene is shown in figure 4–35. The line description extracted from this image, shown in figure 4–36, is affected by the presence of spurious lines resulting from shadows.

Results				
Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
93	1	234	1.0	49
	2	156	0.67	32
	3	111	0.47	29
	4	8	0.04	-

Table 4–11: The results of localisation for example 3.1

Table 4–11 shows the results of localisation for this scene. It can be seen that there are three significant peaks in the Hough space associated with the object. Initially this might seem to indicate a breakdown in the performance of the system, since the scene only contains a single object. However, by analysing the localisation associated with each peak it is possible to provide a satisfactory explanation for the presence of these additional peaks. The projection of the object into the image at the position and orientation indicated by these three peaks is shown in figure 4–34. It can be seen that each corresponds to positions in which the cylindrical section of the object is correctly located, but in two cases the orientation of the object is incorrect. This behaviour can be explained by the fact that the system has mismatched lines describing the inner or outer rings of the cylinder. Localisations based on these mismatches will be correct in position but not in orientation. These localisations then receive a significant amount of support, since the cylindrical sections are correctly matched. However, the localisation corresponding to the strongest peak is correct, as shown in figure 4–37.

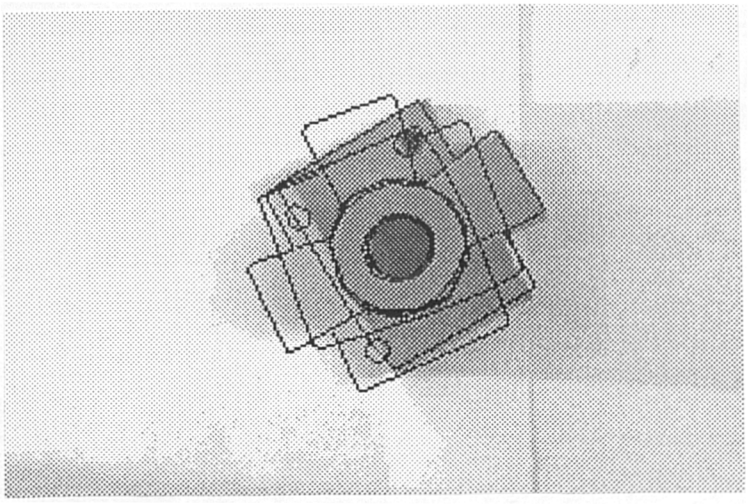

Figure 4–34: Object localisations associated with the three peaks.

Ex. 3.1

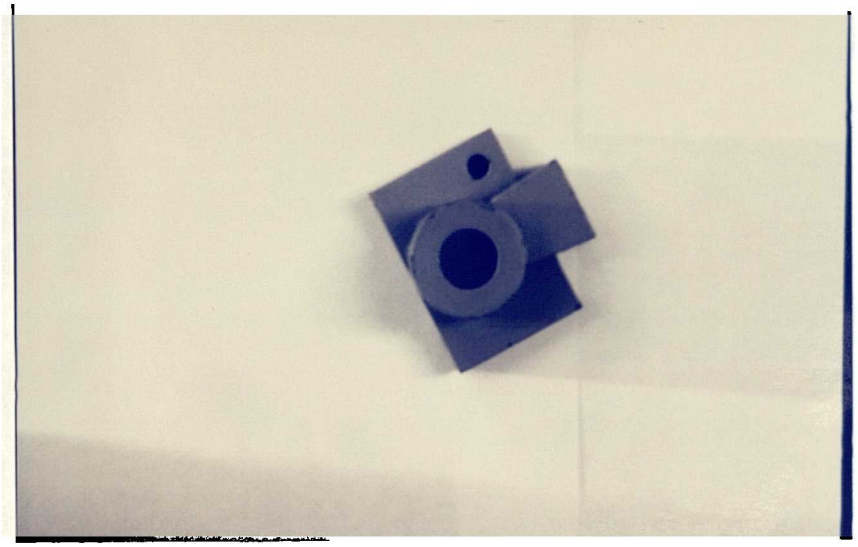

Figure 4–35: Example 3.1 - An image of the scene.

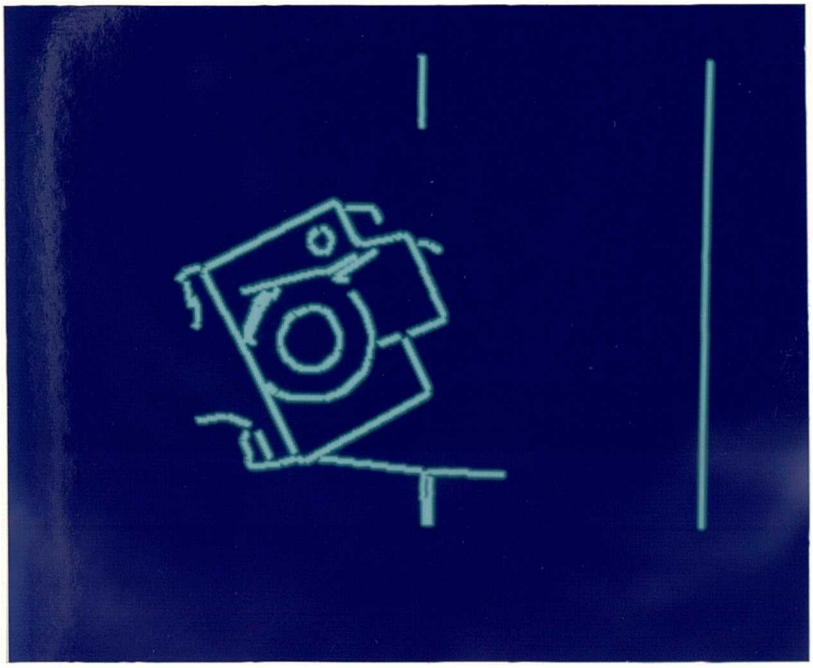

Figure 4–36: Example 3.1 - A colour-coded segmentation of the scene lines.

Figure 4–37: Example 3.1 - The located object(s) projected into the image.

Example 3.2 - A pair of objects

This example demonstrates that the system is able to deal with multiple instances of an object in a scene. An image of the scene is shown in figure 4–39.

Results				
Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
339	1	600	1.0	54
	2	552	0.92	64
	3	136	0.23	38
	4	83	0.14	45
	5	73	0.12	47
	6	56	0.09	-

Table 4–12: The results of localisation for example 3.2

Table 4–12 shows the results of localisation for this scene. It can be seen that there are five significant peaks in the Hough space associated with the object. This is due to the same factor as in the previous example, namely the mismatching of lines describing the circular sections, resulting in incorrect estimates of object orientation. The localisations corresponding to the five significant peaks are shown in figure 4–38. It can be seen that two of the localisations are correct. Indeed, these correspond to the two most significant peaks, whose relative values are much higher than the incorrect peaks. These are shown separately in figure 4–41.

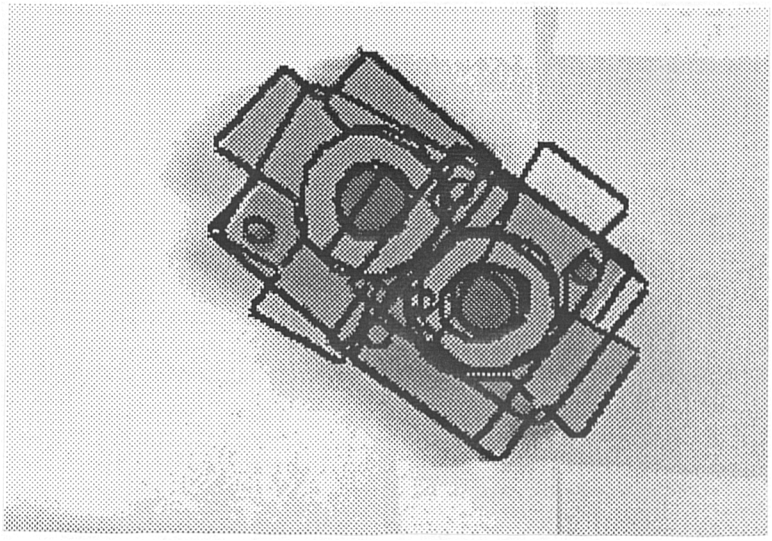

Figure 4–38: Object localisations associated with the five peaks.

This problem should be particular to objects containing circular sections. It could be overcome quite easily by noting that 3D objects cannot occupy the same position in space. This fact could be used to rule out correctly positioned, but incorrectly oriented, localisations arising from spuriously significant peaks in Hough space.

Ex. 3.2

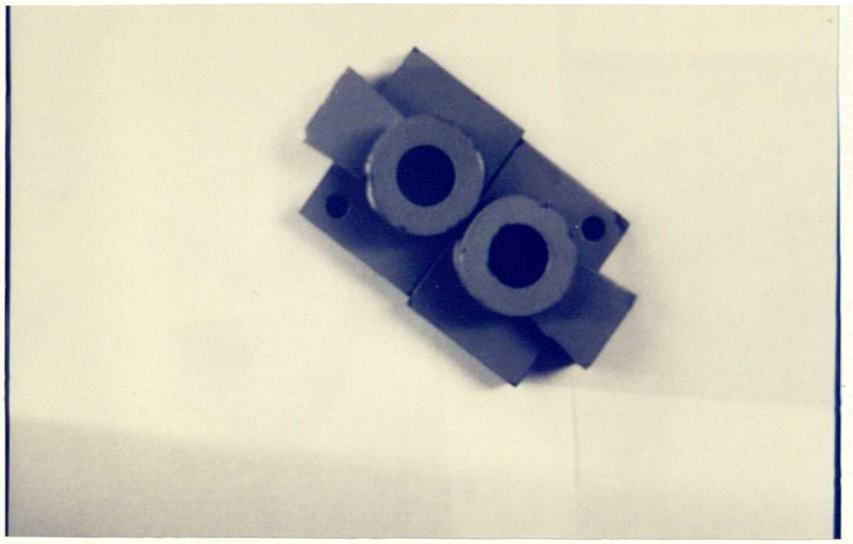

Figure 4–39: Example 3.2 - An image of the scene.

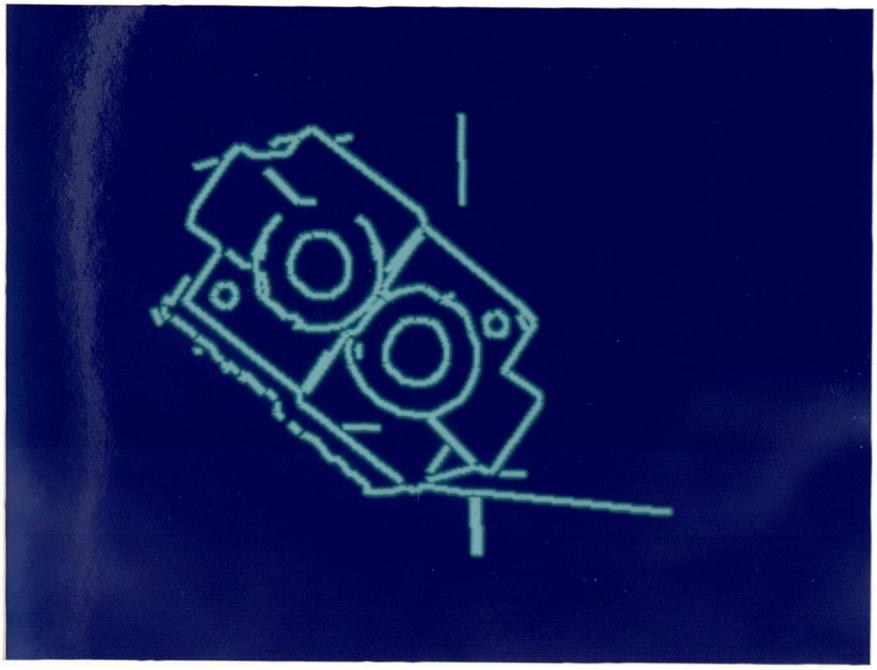

Figure 4–40: Example 3.2 - A colour-coded segmentation of the scene lines.

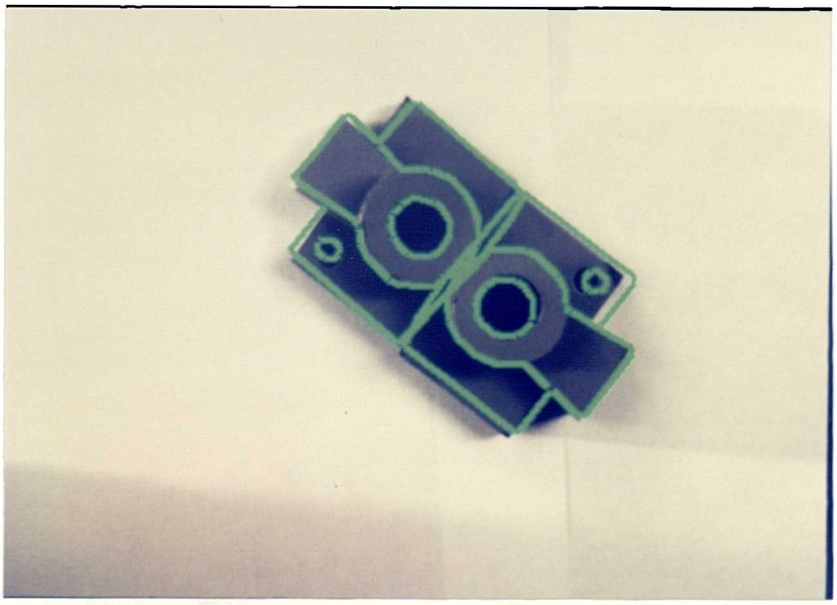

Figure 4–41: Example 3.2 - The located object(s) projected into the image.

Example 3.3 - A pair of objects in a cluttered scene

This example demonstrates that the system is able to deal with the presence of unknown objects in the scene. An image of the scene, which contains two instances of the known object, is shown in figure 4–43. It can be seen that there are also three unknown objects present; a large block, (in the top right hand corner), a small polyhedral object, (in the lower left hand corner), and an object similar to the object used in these experiments but with a differently proportioned base, (in the top left hand corner). The line description of this scene is shown in figure 4–44.

Results				
Number of Votes	Peak Number	Peak Value	Normalised Peak Value	Amount of Support(%)
367	1	546	1.0	59
	2	396	0.73	42
	3	371	0.68	47
	4	291	0.53	49
	5	114	0.21	41
	6	106	0.20	45
	7	47	0.09	-

Table 4–13: The results of localisation for example 3.3

Table 4–13 shows the results of localisation for this example. It can be seen that there are six significant peaks in the Hough space associated with the object. The localisations associated with these six peaks are shown projected into the image in figure 4–42.

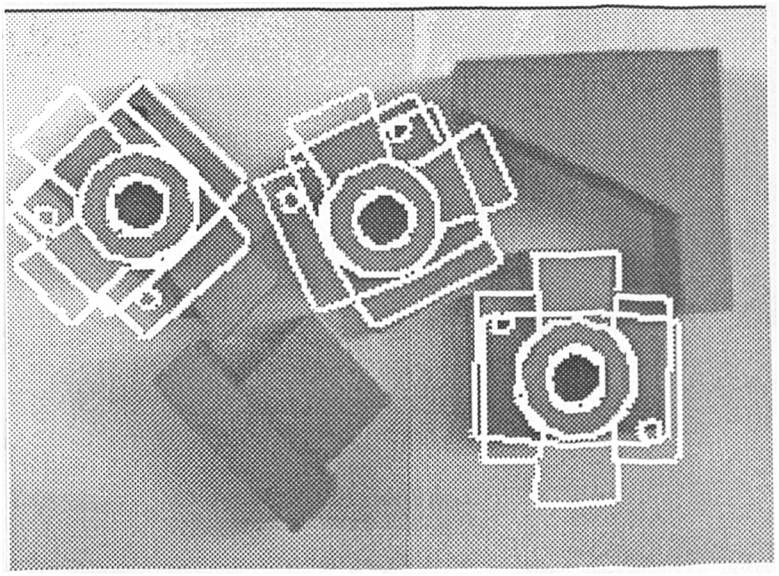

Figure 4–42: Object localisations associated with the six peaks.

Ex. 3.3

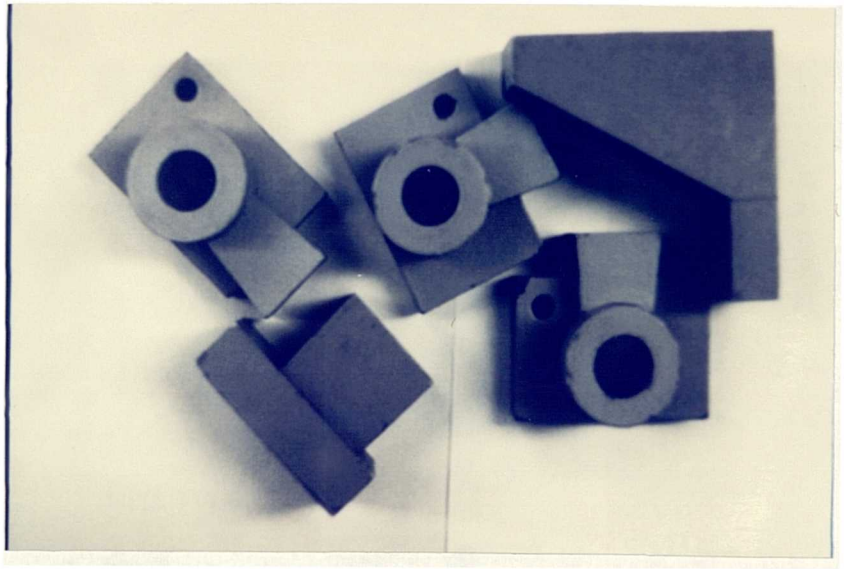

Figure 4–43: Example 3.3 - An image of the scene.

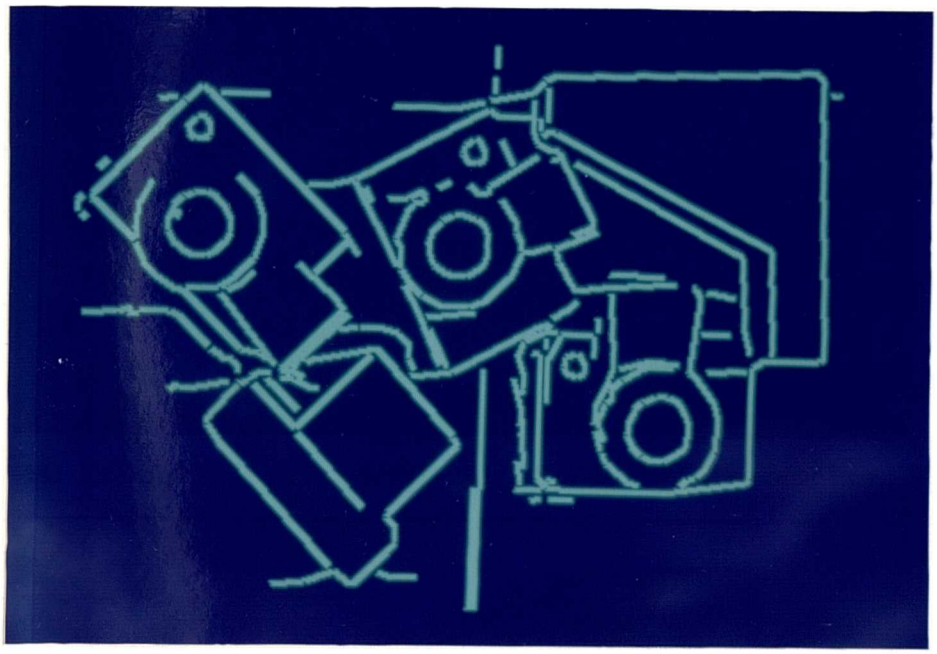

Figure 4–44: Example 3.3 - A colour-coded segmentation of the scene lines.

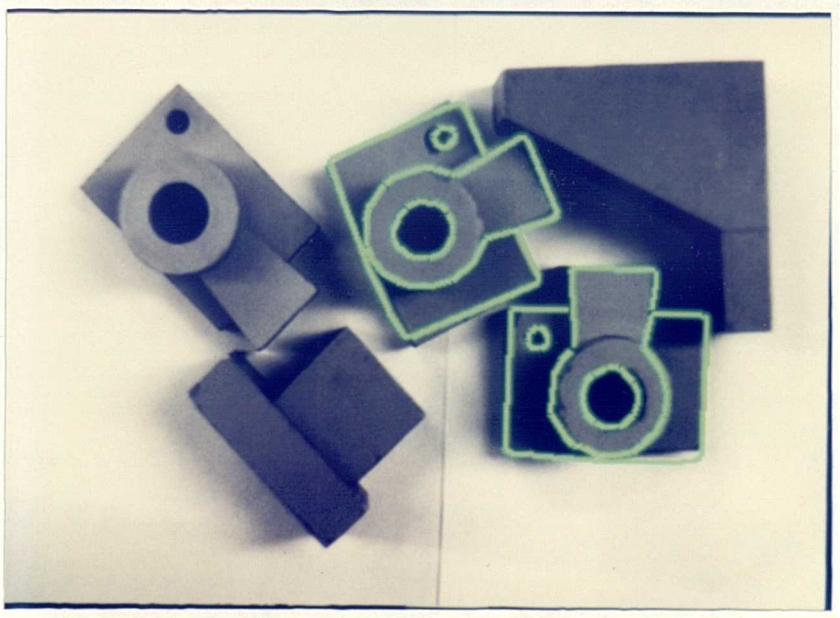

Figure 4–45: Example 3.3 - The located object(s) projected into the image.

The characteristic confusion in the orientation of the objects is again present, although the two most significant peaks still correspond to the correct localisations, as shown in figure 4–45. However, in this example there is the additional problem of the unknown object containing a circular component. Peaks 5 and 6 in table 4–13, which have lower relative values that the first 4 peaks, correspond to localisations of the object in which the circular sections of this object are matched. Whether this is seen as a good or a bad thing depends on the requirements of the system. It could be considered useful that the system has generalised its recognition to include objects that have similar features. However, if the goal is to distinguish between such objects then the system, as it stands, is obviously not suitable.

4.5 Discussion and Summary

This chapter has demonstrated the ability of the recognition system based on the matching of local geometric feature distributions to identify and locate a range of 2D and 3D objects in conditions of considerable fragmentation noise, occlusion and clutter. The results presented in this chapter can be seen as a practical demonstration of the properties of local geometric feature distributions that were established through the theoretical analysis of Chapter 3.

The performance of the system was shown using three sets of objects. A set of five "cut-out" dinosaurs was used to demonstrate that the system is able to recognise arbitrary curved shapes in conditions of severe occlusion, and with unknown objects present in the scene. The system was also shown to be capable of dealing with the simulated effects of fragmentation noise. The recognition of a set of "industrial parts" in severely jumbled scenes was included in order to illustrate that the system is able to deal with more conventional objects containing features such as long lines and circular arcs. Finally, the recognition of a 3D object from a fixed viewpoint was presented to show that the system could deal with the problems arising from "real" image data, eg. fragmentation, sensor error and the presence of spurious image features resulting from shadows.

The results of recognition were demonstrated using both graphical illustrations of the accuracy of matching and localisation and quantitative measures of the results of the localisation process. It was found that the quality of the matching providing by the system, together with the stringest test applied prior to making a vote in Hough space, meant that the Hough space associated with each object often contained only as many significant peaks as there were instances of the object in the scene. This brings into question whether the relatively expensive clustering aspect of the Hough transform is needed, and whether some other, simpler method could not be used. This was identified as an area for further study.

The type of problem addressed in this chapter, ie. the recognition of objects from a fixed viewpoint directly above the objects, suggests that the proposed system could be successfully applied to the classic bin-picking problem, in which objects in jumbled scenes must be identified and located in order to guide some aspect of the manufacturing process. In a suitably calibrated system the information provided by the recognition system could be used to guide a robotic arm in some form of pick-and-place task, eg. the sorting of objects coming along a conveyor belt.

Chapter 5

3D OBJECT RECOGNITION

5.1 Introduction

This chapter presents the application of the representational scheme based on geometric feature distributions, (GFD's) to the the problem of 3D object recognition. It is shown that the GFD scheme is able to support two forms of 3D object recognition; the *3D approach*, which involves extending the scheme to handle the representation of 3D shape, and the *multiple view-based approach*, which involves the use of 3D models composed of a relatively small number of example 2D views of an object, represented as geometric feature distributions. The chapter is organised into the following sections.

1. **Problem Description**
   The difficulties involved in performing 3D object recognition are discussed and the two approaches that are to be investigated are described.

2. **The 3D Approach**
   The extension of the GFD scheme to the representation and matching of 3D shape descriptions is presented. It is shown that the representation of 3D shape can be achieved simply by proposing a set of geometric features defined between 3D line segments. The matching of 3D scene descriptions, extracted using an existing stereo algorithm, to 3D wire-frame models can then be performed by a simple extension of the system previously proposed for 2D recognition. The ability of this system to identify and locate 3D objects in a scene is demonstrated.

3. **The Multiple View-Based Approach**
   This section presents the application of the GFD scheme within a 2D, multiple view-based approach to recognition. This includes a general description of the process by which the representation of the set of all 2D views of an object generates a hypersurface in feature space. Through a discussion of the likely properties

of these hypersurfaces it is argued that there are significant advantages to providing probabilistic recognition information. The structure of a self-organising neural network is presented which is able to construct 2D, appearance-based object representations by clustering, on the basis of their similarity in feature space, views of an object represented as geometric feature distributions. The system is shown to be capable of performing accurate classification of a set of aeroplanes while storing a relatively small number of views of each object.

5.2 Problem Specification

This chapter addresses the problem of recognising 3D objects from an arbitrary viewpoint. As such it can be thought of as a relaxation of the set of viewing conditions assumed in Chapter 3. The fact that the spatial relationship between an object and the viewing camera is no longer fixed means that we must now address the problem caused by the variation in the 2D, projected shape of an object as it is viewed from different directions and distances. It will be noted that such variations are over and above those addressed in Chapter 3, ie. fragmentation noise, occlusion and scene clutter. The task in 3D object recognition is to associate each of the possible views of an object with a unique object classification.

This chapter presents two alternative methods for performing 3D object recognition using shape representations in the form of geometric feature distributions. The first, termed the *3D Approach*, involves matching 3D scene descriptions, obtained by establishing stereo correspondence between pairs of images, to 3D, wire-frame object models. This requires that the representational scheme presented in Chapter 2 be extended to deal with the the problem of representing 3D shape. The 3D object recognition problem then becomes a relatively straightforward extension of the 2D matching problem, and can be addressed using essentially the same system as that presented in Chapter 3.

The practical limitations of the 3D approach, together with the desire to provide a more physiologically plausible account of recognition, provides the motivation for studying a multiple view-based approach to 3D object recognition. This involves basing recognition on the 2D projected shape information present in a single image, while using object models composed of only a relatively small number of example views of each object. It will be appreciated that the variations in 2D shape arising in 3D object recognition go beyond the particular invariance properties of the 2D geometric feature distributions presented in Chapter 2. A mechanism must therefore be proposed for generalising recognition from the small set of example 2D views to all possible views of an object. The proposed solution involves using a self-organising artificial neural network to cluster views on the basis of similarity and to use the generalisation properties of network classification mechanism to provide recognition.

Both of these approaches are now addressed in detail.

5.3 The 3D Approach

This section describes the application of the geometric histogramming scheme to the problem of representing and matching 3D shape descriptions.

The major difficulty in performing 3D object recognition based on the projected shape information present in a single 2D image is that such descriptions are view-*dependent.* One way of overcoming this difficulty is to attempt to derive a description of the objects in a scene which is view-*independent*, or *object-centred.* This involves exploiting some form of depth cue present in an image, or images, in order to produce a description of the scene as a collection of 3D surfaces or contours. Since such descriptions are invariant to changes in viewpoint, modulo visibility constraints resulting from self-occlusion, they can be matched directly to similarly described 3D object models. This approach can therefore be seen as a relatively straightforward extension of the $2D \leftrightarrow 2D$ problem. Various strategies have been proposed for performing the matching of 3D shape primitives, including interpretation tree-search, [41,34,70], graph analysis, [14,77], relaxation labelling, [9], the generalised Hough transform, [6], and hashing techniques, [97].

To date, very few 3D object recognition systems have employed statistical pattern classification techniques. This has been primarily due to the fact that previously proposed shape representational schemes, being intrinsically global in nature, are not robust to the loss of object features resulting from self-occlusion. Given the results presented in Chapter 3 demonstrating the robustness of 2D GFD's to loss of shape information it seems reasonable to expect that will extend successfully to the representation and matching of 3D shape representations.

5.3.1 Obtaining 3D Shape Descriptions

The 3D approach requires that descriptions of the 3D shape of objects in the scene be matched to similarly described object models. In the present study, 3D scene descriptions are obtained using the PMF stereo algorithm, [76], as implemented in the TINA vision system, [79,78]. This is based on establishing correspondences between shape primitives detected in a pair of images of the scene. These correspondences can be used, together with information regarding the parameters of the viewing cameras, to produce a set of 3D shape primitives, in this case line segments, which describe the shape of objects in the scene. An example of the line segments obtained from the pair of images in figure 5–1 is shown in figure 5–2.

Objects to be recognised are described using 3D line segments. These combine to form a "wire-frame" model of the object's contours . While a number of researchers have

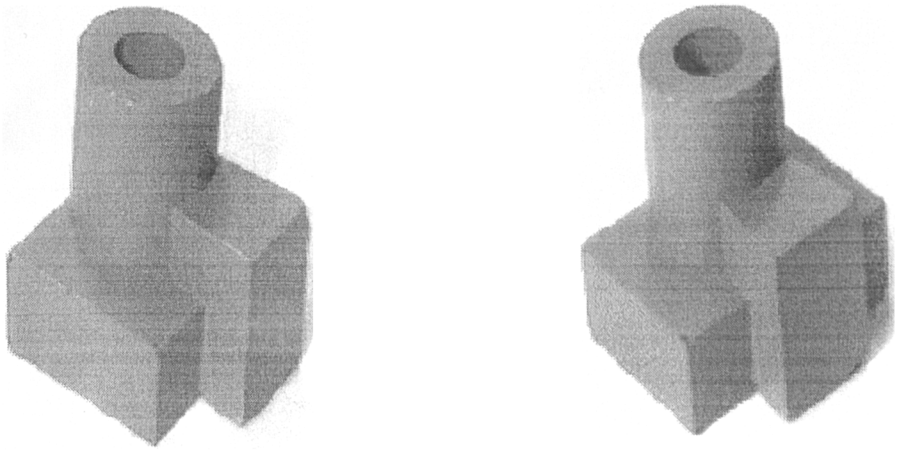

Figure 5–1: Left and right images of a scene.

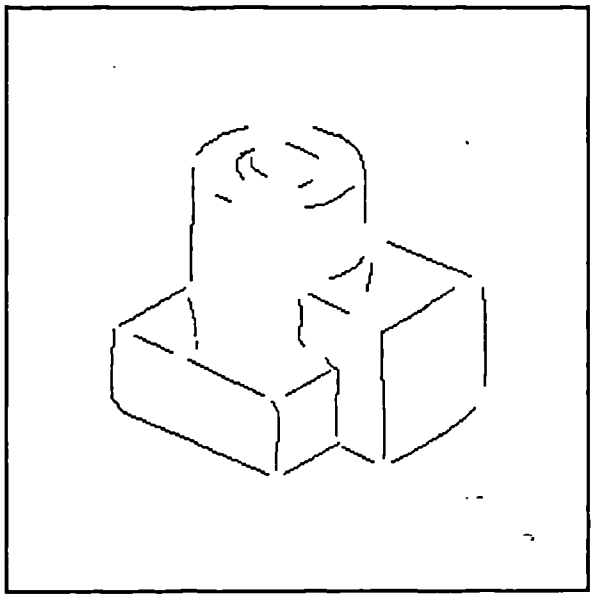

Figure 5–2: The 3D scene lines produced by the PMF algorithm.

attempted to devise methods for constructing such models automatically from multiple views of the object, the success of this approach has been limited. Consequently, the models used in this study were obtained through hand coding. The "wire-frame" models of the two objects in figure 5–3 are shown in figure 5–4. It can be seen that the model of the object shown in figure 5–4(b) is not complete, since the non-polyhedral sections of its contour are not well described using 3D line segments.

5.3.2 Representing 3D Shape

One of the major advantages of the GFD representational scheme is the ease with which it can be adapted to the representation of either 2D or 3D shape. The scheme presented in Chapter 2 for representing 2D shape can be extended to deal with the

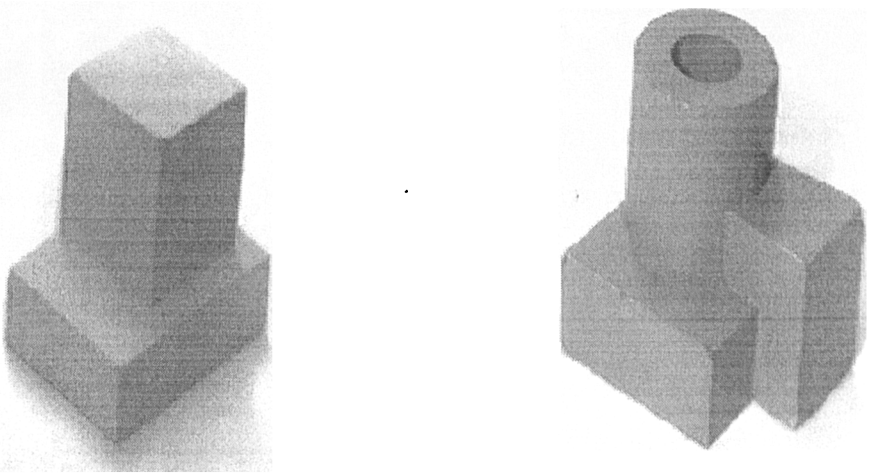

Figure 5–3: The objects used to demonstrate the system, (a) **B0** and (b) **B1**.

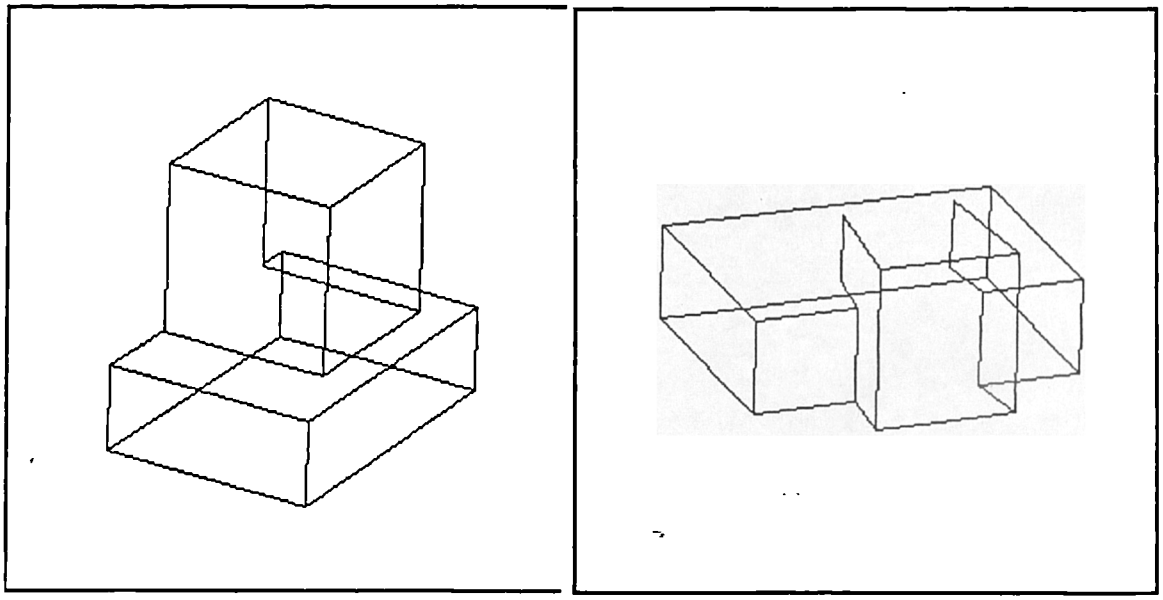

Figure 5–4: Wire-frame models of (a) **B0** and (b) **B1**.

representation of 3D shape by simply defining a set of geometric features that can be used to measure the relationship between pairs of 3D line segments. The distribution of the values of these features can then recorded in exactly the same way to provide a robust, discriminant representation of 3D shape.

A 3D Geometric Feature Set

The geometric features presented in Chapter 2 for measuring the relationship between pairs of 2D line segments can be generalised without too much difficulty to the case of 3D line segments. However, the nature of 3D object projection is such that the directionality of the 3D lines segments extracted from a scene may vary with lighting. Changes in the proportion of incident light falling on adjacent object surfaces may result in a change in the direction of the intensity gradient upon which the detection of line segments is based. Consequently, 3D line segments must be treated as being

non-directional. This fact must be taken into account when defining 3D geometric features.

Relative Angle

The relative angle feature can obviously be used to measure the geometric relationship between pairs of 3D line segments. However, the non-directionality of the line segments means that the range of the relative angle feature is reduced from $[0 \rightarrow 2\pi]$ to $[0 \rightarrow \pi]$. The value of g_θ is therefore given by

$$g_\theta = \begin{cases} \theta - \pi & \text{if } \theta > \pi \\ \theta & \text{otherwise} \end{cases}$$

where θ is the measured angle between the two lines segments.

The Perpendicular Distance Feature

The perpendicular distance feature defined between two 3D line segments is given by

$$g_d = \|p_j + \alpha \mathbf{d_j} - p_i - \langle (p_j + \alpha d_j - p_i), \mathbf{d_i} \rangle \mathbf{d_i}\| \quad \alpha \in [0, \ell_j]$$

where p_i and p_j are endpoints on lines ℓ_i and ℓ_j and $\vec{\mathbf{d}}_i$, $\vec{\mathbf{d}}_j$ are the unit direction vectors of ℓ_i and ℓ_j respectively. The values used in recording the relationship between two line segments are those that occur at the extrema of this expression, where $\alpha_j = 0, |\ell_j|$. Distances are now measured in physical, rather than pixel-based, units.

The distribution of the values of these geometric features measured between elements of a 3D shape description can be recorded in exactly the same way as those measured between 2D line segments. Consequently, both local and global levels of representation are possible. A slight difference is that the local region defined around a line segment now becomes a spherical volume in 3D space, rather than a circular region in the image plane. Also, the non-directionality of 3D line segments, and the resulting weakening of the geometric features, means that representations of 3D shapes are less discriminatory than those of 2D shape.

5.3.3 Global 3D Shape Matching

If objects are encountered in isolation then the GFD scheme could also be used in 3D, non-correspondence recognition, based on the matching of global representations of 3D shape. However, such recognition provides no information on the pose of objects in the scene. Given the difficulty in extracting a 3D scene descriptions it is likely that such information will be a requirement. However, global shape matching can be thought of as a method for indexing likely object models. The matching of local shape primitives within the indexed models could then be used to determine pose.

5.3.4 Demonstration of Local 3D Shape Matching

Correspondences between 3D model and scene line segments are established through the matching of local geometric feature distributions using essentially the same mechanism as that proposed for 2D object recognition. The ability of this 3D system to establish correct matches between 3D scene and model lines is demonstrated by the colour-coded matches shown in figure 5–5. It can be seen that the loss of shape information in the scene description arising from self-occlusion does not affect matching. This is to be expected given the robust properties of the GFD scheme established in Chapter 3.

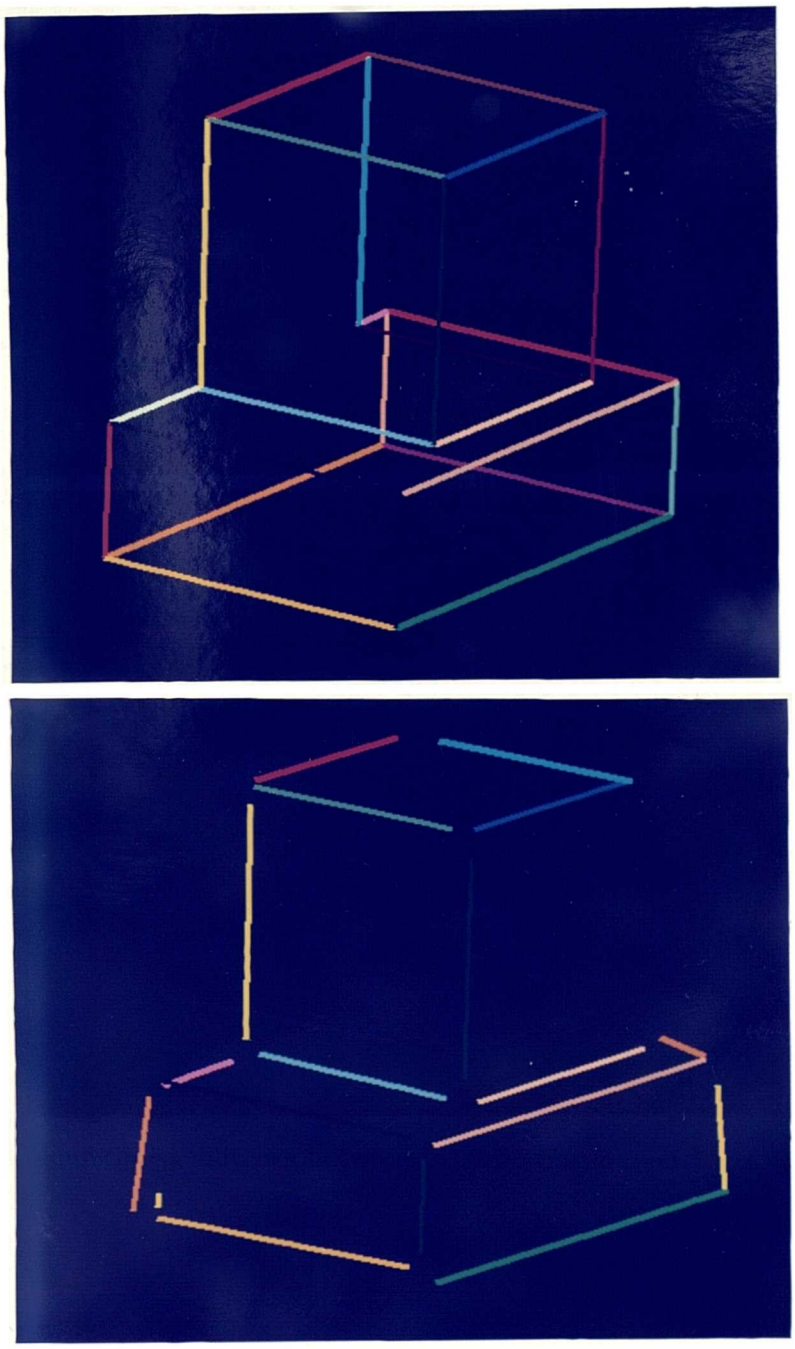

Figure 5–5: Colour-coded matches between (a) wire-frame model and (b) scene description.

5.3.5 Determining Object Pose

The line matches delivered by the classification scheme can be used to determine the 3D position and orientation of objects in the scene. However, in practice certain difficulties were encountered in performing localisation. The scene description shown in figure 5–5 was obtained by placing the object in a favourable pose, ie. one in which a large number of object contours are visible to both viewing cameras. However, in general the description of shape obtained from the scene will not be as complete. Also, the fact that the chosen objects are highly symmetric, together with the decrease in the strength of the geometric features caused by the non-directionality of 3D line segments, means that the the representation of a particular model line is unlikely to be unique. This introduces the possibility that a particular scene line may match identically to several model lines, only one of which is correct. This difficulty was overcome by making a number of modifications to the matching and recognition schemes previously described for the 2D case. In particular, the nearest-neighbour classification rule was relaxed such that the top k responding model lines are carried forward as potential matches for a particular scene line. It was found that $k = 3$ was sufficient.

Using the matches delivered by the GFD scheme to determine the position and orientation of objects is obviously more difficult than in the 2D case. Pairs of matched, non-parallel, lines can be used to determine the 6 parameters needed to describe the pose of an object in the scene, Grimson [40]. However, the high degree of symmetry within the chosen objects means that many pairs of lines are parallel, and so cannot be used in computation. The second, and more important, problem is due to the non-directionality of the line segments. If each each line is considered in both directions then each pair of matched lines gives rise to four sets of transformation parameters.

The use of the generalised Hough transform to determine object pose was investigated but found to be unsuitable, due to the number of spurious entries made in the Hough space. If one considers that there are kI potential matches, and therefore $\frac{kI(kI-1)}{2}$ possible pairings, and each pairing gives rise to 4 sets of transformation parameters, then given the high degree of symmetry in the objects it becomes clear why the GHT faces problems. In practice it was found that the Hough spaces were quite cluttered, giving rise to many, equally ranked, hypotheses. While these poses were often partially correct, eg. they placed the object being upside down or on its side, the solution was discounted.

In the present study object localisation was achieved using an alignment approach, [50]. While this does not overcome the problems highlighted above, it can be expected to return acceptable solutions, where they are available, in reasonable time. Each set of transformation parameters computed from a possible pairing was evaluated using a model test, ie. the model was projected into the scene at the hypothesised position and orientation and the amount of local support it received was determined. If an hypothesis received sufficient support, ie. a certain percentage of the contours receive

support, then the search was terminated. In the case where the scene contained a single occurrence of an object at a favourable pose it was found that search terminated very quickly. For scenes containing less favourable poses more model tests were necessary before an acceptable match could be found. In the latter case, where the number of scene contours is typically small, it was found that finding an acceptable threshold that distinguished between correct and incorrect localisations was quite difficult.

In the case where a scene contains multiple occurrences of an object this strategy is obviously inadequate, since search terminates after the first object is found. An obvious solution to this problem is to remove from consideration those scene lines that are matched by a validated localisation. The search could then continue until the number of lines in the scene falls below a certain level, (although this will obviously fail in scenes containing unknown objects). In the present system, in order to demonstrate that matching can be performed in cluttered scenes, the system was adapted to return a fixed number of hypothesis, eg. 5, which were then ranked on the basis of the amount of support they received. The localisations shown in the following examples were, in each case, the highest ranked hypotheses. Automating the localisation of objects in cluttered scenes is an area for further study.

5.3.6 System Demonstration

The ability of the system to perform 3D object recognition and localisation is now demonstrated using the objects **B0** and **B1** shown above. Left and right images of the scene containing the object(s) were captured and used as input to the PMF stereo algorithm. This provided a 3D, line-based description of the contours in the scene which was then matched to the wire-frame model. The matches produced in this way were used to determine object pose, using the method described above. The histograms used in these examples had parameters $n_\theta = 30$, $n_d = 20$, $\sigma_\theta = \sigma_d = 1.0$. The radius of the local region was set to 60mm, so as to just include the whole object.

The performance of the system is demonstrated in a number of examples:

Example 1.
Object **B0** is placed in a relatively favourable pose, such that a significant number of the lines in the wire-frame model are matched. In this case the object was located after only 1 model test.

Example 2.
Object **B0** is placed in a less favourable pose, upside down, such that the 3D scene description contains fewer object contours. In this case, 5 model tests had to be performed before the object was correctly located.

Example 3.
This example presents a scene containing two instances of object **B0** that partially occlude one another. The correct localisations correspond to the top two ranked hy-

potheses returned by the system. It is interesting to note that the occlusion caused by the presence of multiple objects is effectively no different from that caused by self-occlusion, which, as the previous examples show, the system is robust to.

Example 4.
Here object B1 is shown on its side. The object was located after 6 model tests. The fact that the wire-frame model of **B1** is incomplete means that elements of the scene description corresponding to the cylindrical section can be regarded as scene clutter. The fact that the system is able to deal with this noise follows from the property established in Chapter 3.

Example 5.
Two instances of object **B1** are shown. As in example 3, the correct localisations correspond to the top two ranked hypotheses.

In each example results are shown at three levels of processing:

i) the left image of the scene.
ii) the 3D lines extracted by the PMF algorithm.
iii) the located object(s) projected into the left image.

Ex. 1

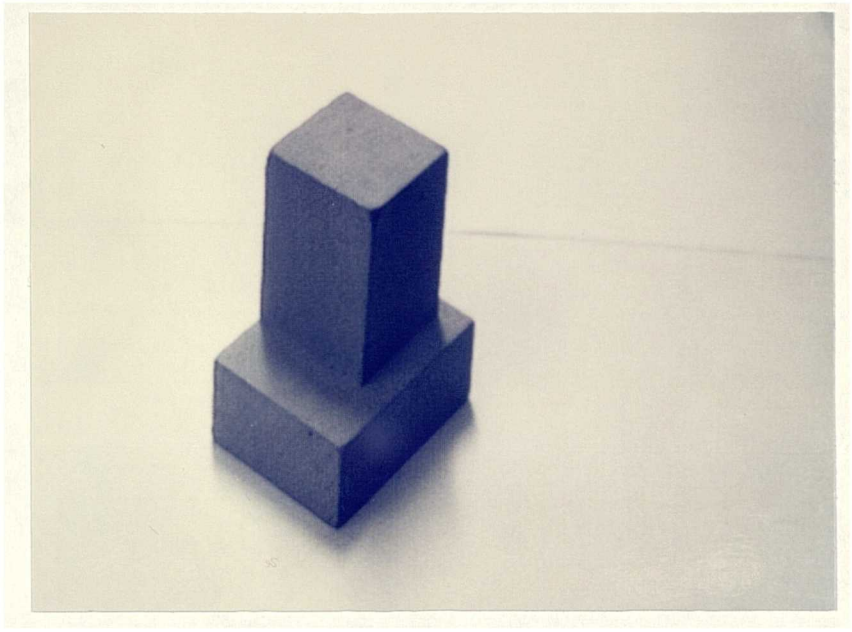

Figure 5–6: Example 1 - An image of the scene.

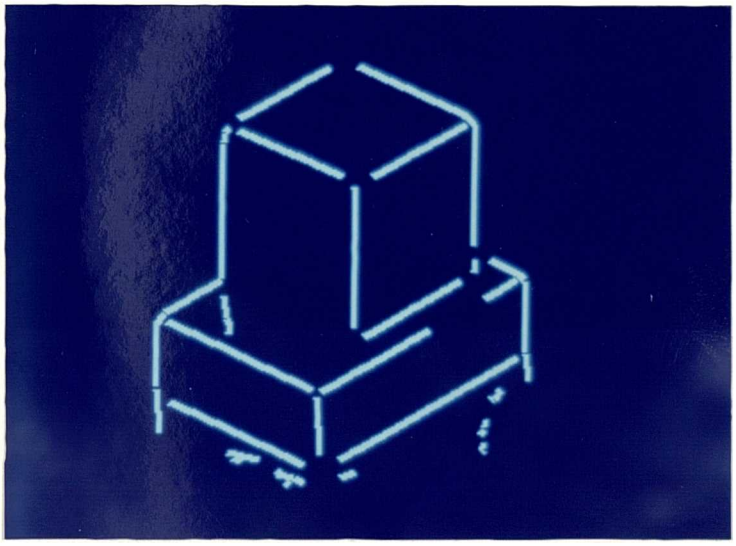

Figure 5–7: Example 1 - The 3D lines extracted from the scene.

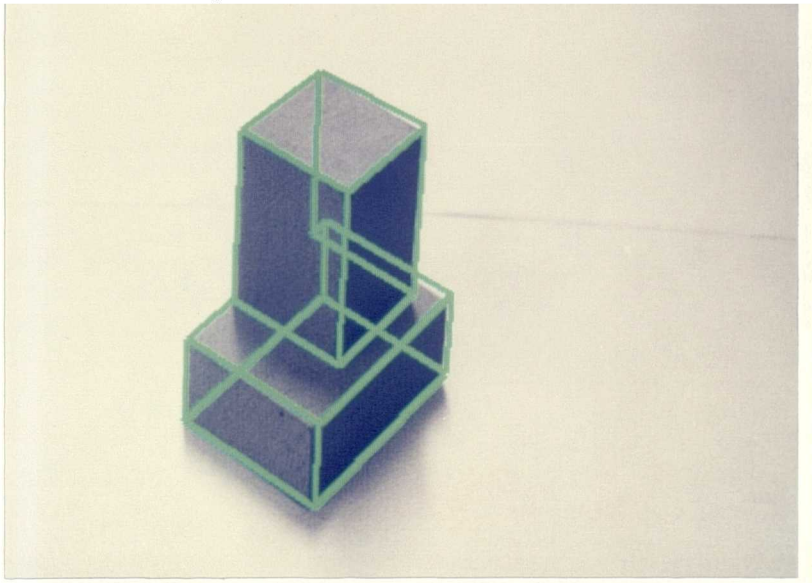

Figure 5–8: Example 1 - The located object(s) projected into the image.

Ex. 2

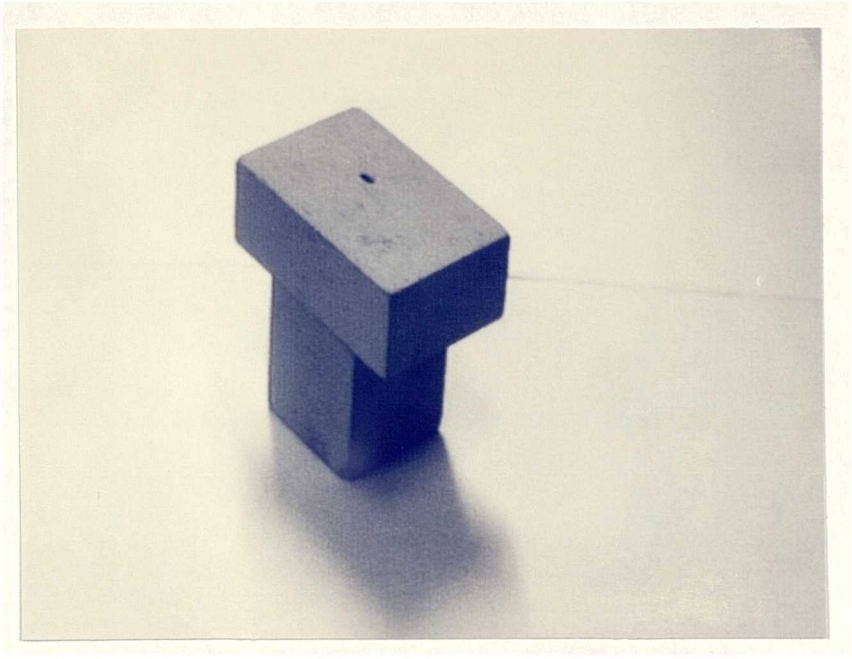

Figure 5–9: Example 2 - An image of the scene.

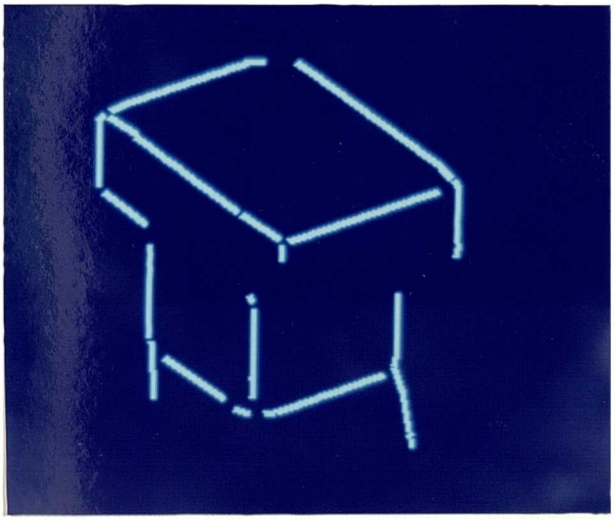

Figure 5–10: Example 2 - The 3D lines extracted from the scene.

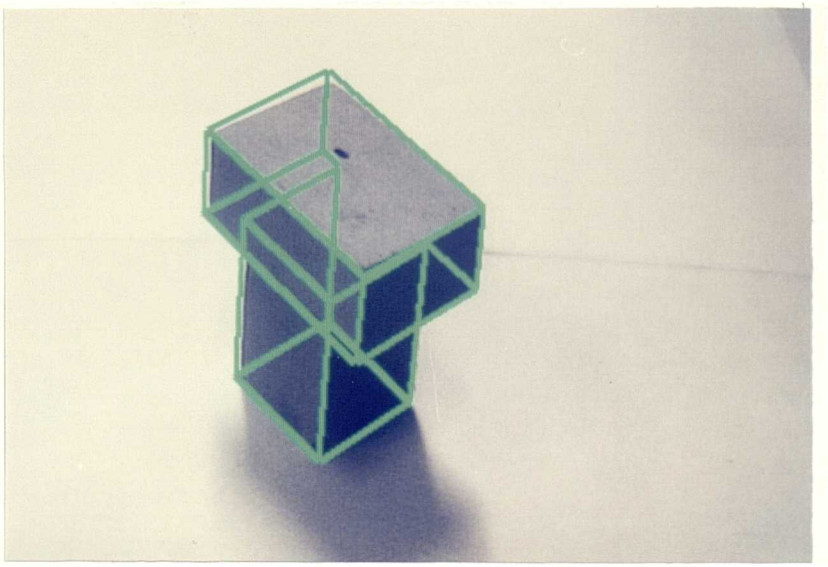

Figure 5–11: Example 2 - The located object(s) projected into the image.

Ex. 3

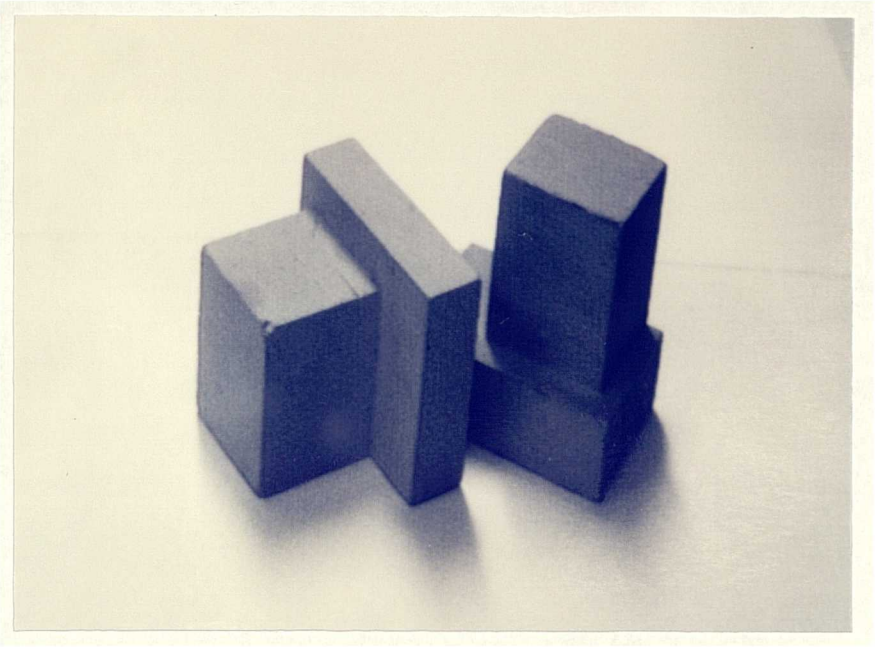

Figure 5–12: Example 3 - An image of the scene.

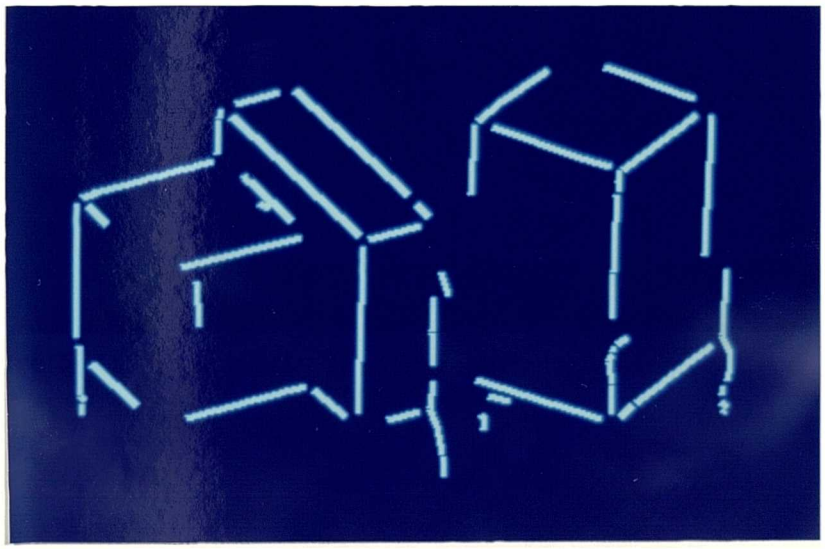

Figure 5–13: Example 3 - The 3D lines extracted from the scene.

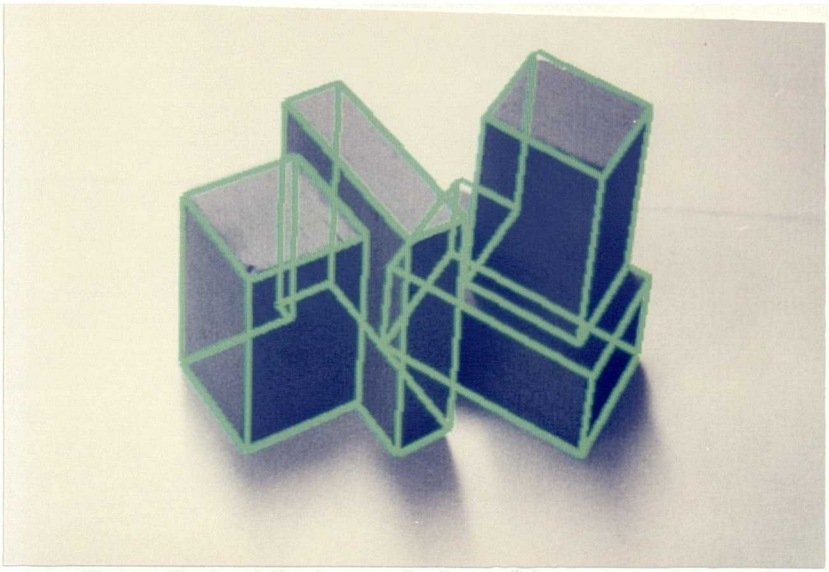

Figure 5–14: Example 3 - The located object(s) projected into the image.

Ex. 4

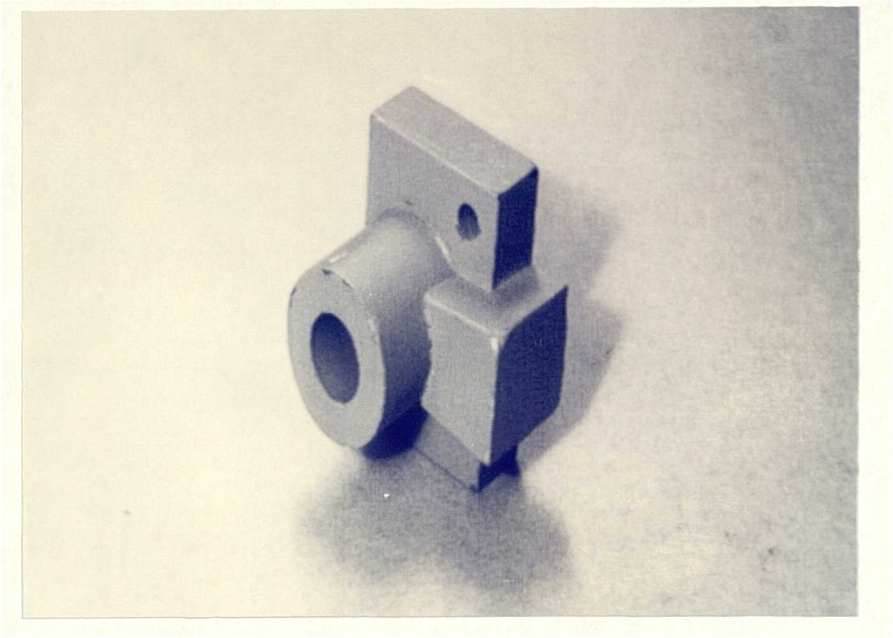

Figure 5–15: Example 4 - An image of the scene.

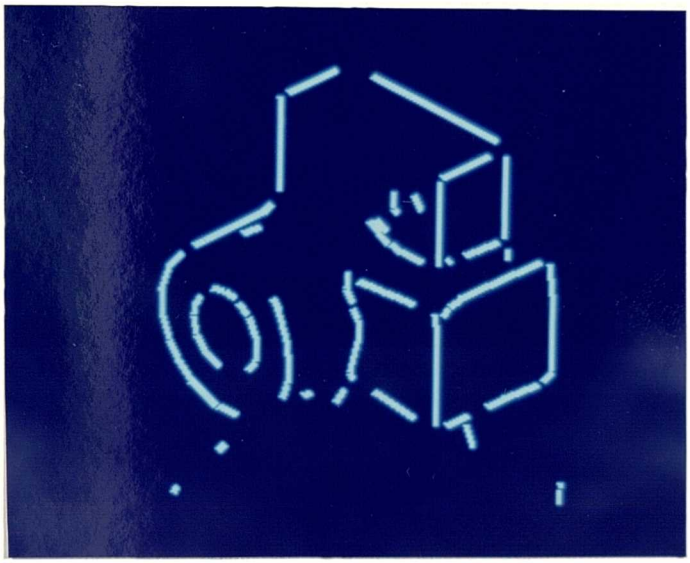

Figure 5–16: Example 4 - The 3D lines extracted from the scene.

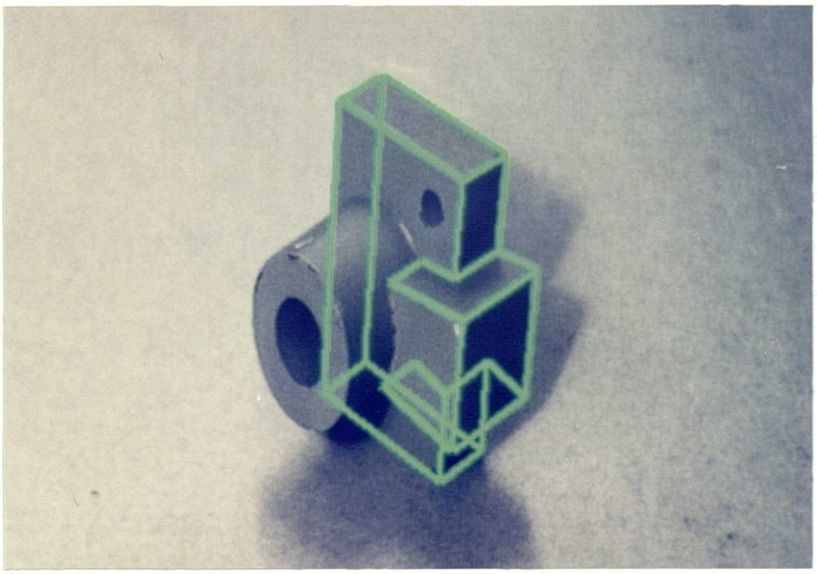

Figure 5–17: Example 4 - The located object(s) projected into the image.

Ex. 5

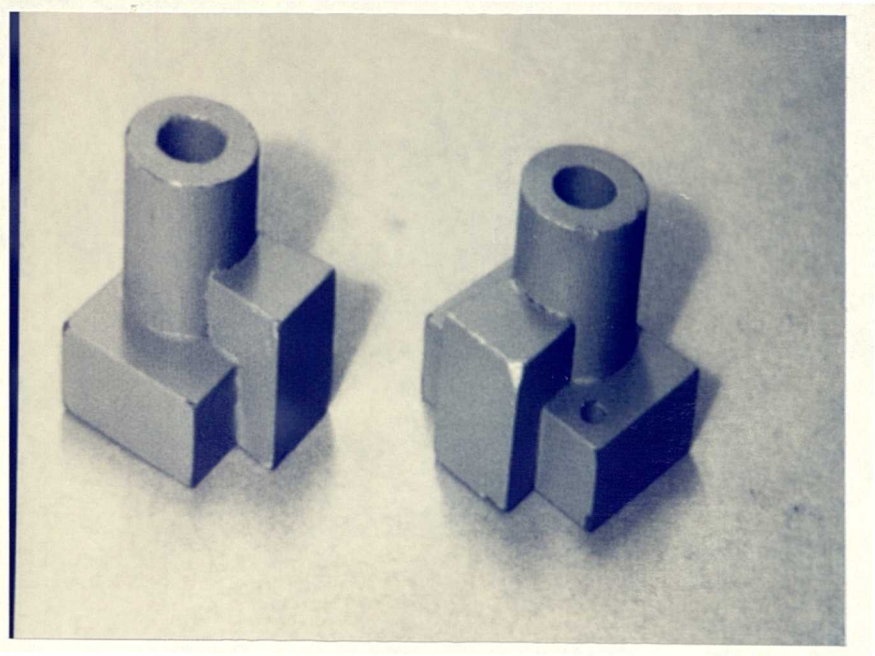

Figure 5–18: Example 5 - An image of the scene.

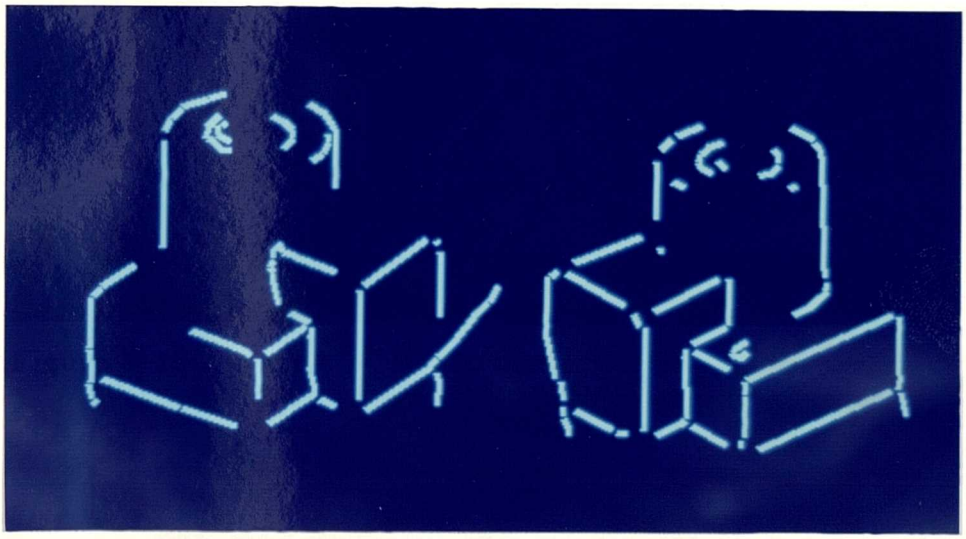

Figure 5–19: Example 5 - The 3D lines extracted from the scene.

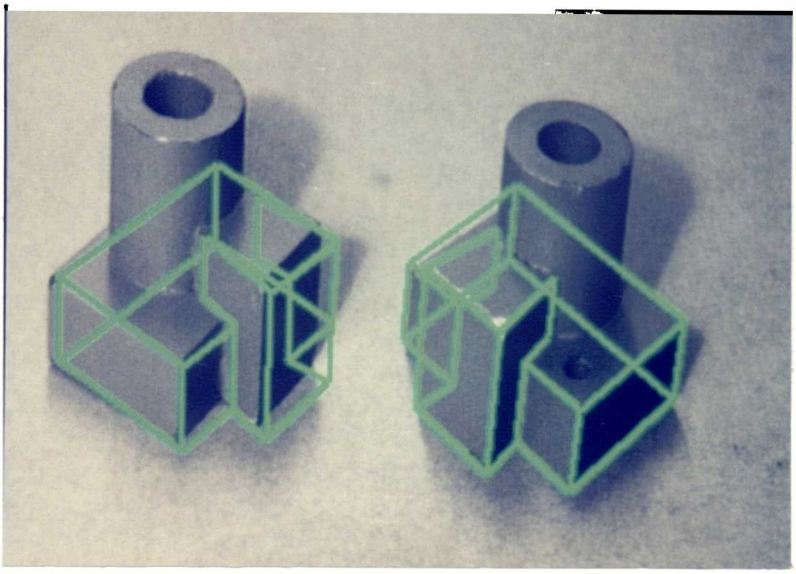

Figure 5–20: Example 5 - The located object(s) projected into the image.

5.3.7 Discussion

This section has demonstrated that a scheme based on the classification of local geometric feature distributions representing 3D line segments is able to both recognise and locate 3D objects in a scene. However, the performance of system could be improved in a number of ways. Firstly, the current system does not take visibility conditions into account when constructing object representations. That is, when recording geometric feature distributions, no account is taken of whether pairs of model lines could possibly be visible from the same viewpoint. The incorporation of such constraints into the representational scheme could be expected to have significant advantages in terms of the ability of the system to deal with unfavourable poses. Secondly, although the system was shown to be able to locate objects, the method of achieving this is not totally satisfactory. The development of strategies for automatically and robustly determining accurate transformation parameters from the matches delivered by the GFD scheme is an area for further study.

The remainder of this chapter examines an alternative form of 3D object recognition which does not rely on the matching of 3D shape. The proposed system is based upon the use of a self-organising neural network to cluster 2D views of an object, represented as geometric feature distributions, in order to provide probabilistic recognition information, and is based on work presented in Evans et al. [31].

5.4 A Multiple View-Based Approach

The practical limitations of the 3D approach, both in the difficulty of obtaining 3D scene descriptions and in model acquisition, together with the desire to explore a more physiologically plausible solution, provides the motivation for investigating a multiple view-based approach to 3D object recognition. Such approaches do not rely on the availability of 3D, object centred scene descriptions, but rather propose that recognition be based on the 2D, projected shaped information present in an image of the scene. Furthermore, models of the objects to be recognised do not explicitly represent the 3D structure of the object but instead are appearance-based, in that they are composed only of a relatively small number of example views of the objects. Such models are obviously much easier to acquire than those in which 3D structure is explicitly represented. Moreover, they place no restrictions on the types of object that can be modelled. The view-based approach therefore holds great promise for delivering a truly adaptive artificial recognition system in which object models are learned through experience.

5.4.1 Describing Shape Variation

The obvious difficulty in attempting to base 3D object recognition on 2D image data and 2D appearance-based models is in generalising from the small set of example views to all possible views of an object. If recognition is treated as a classification problem, where 2D projected shape is represented by a feature vector, then each object is represented not as a point, but as a hypersurface in feature space. Various approaches have been proposed for basing recognition on such hypersurfaces. These include extensions of the nearest-neighbour classification approach, [25,83,85,108], learning vector quantisation, [91,26,82], linear combinations, [104,7] and view interpolation, [75,27]. These approaches differ both in the mechanism they propose for generalising recognition from familiar to novel views, and in the processes by which models are acquired, and each makes certain assumptions about the nature of the hypersurfaces, eg. smoothness, linearity.

This section attempts to formalise the source and nature of the shape variation in 3D object recognition by describing the process by which the set of all views of an object generates a hypersurface in the space of possible shape representations. The form of this hypersurface determines to a large extent the ease with which multiple view-based recognition can be performed. Therefore, by considering possible characteristics of these hypersurfaces, together with the factors affecting them, a greater understanding of the difficulties involved in performing multiple view-based recognition will be gained.

Projection

The $2D$ shape found in an image is the result of the projection of a set of object features onto the image plane of the camera. This process can be approximated by the general projective transformation p, taken here to be a perspective projection. For a given object o the shape s produced by p depends on two factors, the viewing parameters $\boldsymbol{\lambda}$ and a noise term η.

$$p : (o, \boldsymbol{\lambda}, \eta) \rightarrow s \tag{5.1}$$

The vector $\boldsymbol{\lambda}$ describes the position and orientation of o relative to the viewing camera. If we restrict the camera to be foveated on the object, then this relationship can be fully described by 4 parameters; 3 orientation and 1 distance. The noise term η describes variations in s caused by changes in lighting and/or occlusion by other objects; a number of such models were presented in Chapter 3. It will be useful at this stage to distinguish between the set of shapes S_λ, obtained by applying projection p to o for all possible values of $\boldsymbol{\lambda}$, and S_η, the set of shapes produced by applying the noise model η to S_λ.

Shape Representation

The next stage is to produce an encoding of the 2D projected shape of an object in the form of a feature vector. The construction of such an encoding can be described in general terms by the representational function r, which takes the set of $2D$ primitives s and returns a vector $\mathbf{d}$, the components of which are related to some characteristic of s.

$$r : s \rightarrow \mathbf{d}$$

It will be noted that this description of the representation process places no restriction on the level of representation that is adopted. If representation is global then each view of an object is represented by a single feature vector, while representation of local shape elements produces k features per view, where there are k primitives in s.

The GFD scheme is a specific example of a representational function r. Other forms of shape encoding that have been employed in multiple view-based approaches include Fourier coefficients [85,108], moment invariants [25,83], log-polar maps [91,82], chain-codes, [107], coarse coding, [28], and ordered correspondence vectors, [104,75].

Hypersurface Generation

We now define a composite function $f = r \circ p$, which defines a mapping between Λ^4, the space of possible viewing parameters, and D^n, the space of possible shape representations. If we consider a fixed object o, viewed under conditions of zero noise, ie. $\eta = 0$, then f maps each point in Λ^4 to a point in D^n, Eq. 5.2. The application of Eq. 5.2 to all points in Λ^4 generates, via the intermediate set of shapes S^λ, a set of points H in D^n that can be said to lie on a hypersurface, Eq. 5.3.

$$f : (o, \boldsymbol{\lambda}) \rightarrow \mathbf{d} \tag{5.2}$$

$$H = f(o, \Lambda^4) \tag{5.3}$$

Hypersurface Properties

The nature of the hypersurfaces generated by a particular shape representation for a set of objects determines to a large extent the ease with which recognition can be performed. This section discusses likely hypersurface characteristics and the way in which particular features of both the objects and the representational scheme can be expected to affect them.

i) Smoothness

The ease with which generalisation from familiar to novel views can be performed is

determined largely by the smoothness of H. This depends crucially on the behaviour of the representational function r: if r is well-behaved, in the sense that a small change in s produces a small displacement of $\mathbf{d}$, then the hypersurface should be locally smooth.

ii) Dimensionality

Although H exists in the high-dimensional representation space D^n, it need not itself be n-dimensional. Provided that r is well-behaved, H can be considered as a low-dimensional manifold embedded in the high-dimensional space of possible shape representations [104]. The use of a function r which is invariant to changes in λ along some dimension, eg. rotation about the camera axis, will locally reduce the dimensionality of the hypersurface. Thus, in general, the local dimensionality of H is determined by the number of degrees of freedom in the viewing parameters λ, minus any invariances that r may have. An extreme example of this observation is provided in Rothwell et al. [87]. This shows that for planar objects there exists a function r which is invariant to changes in all 4 viewing parameters: the use of such a function produces a "hypersurface" consisting of a zero dimensional point.

iii) Symmetry

The possibility of symmetries within an object implies that Eq. 5.1 may represent a many-to-one mapping. For instance, in the projection of a cube all possible 2D shapes can be obtained from a sub-volume in Λ^4 corresponding to views from a single octant of the space surrounding the cube. Additionally, emergent symmetries may be created by the behaviour of r. For example, if r is invariant to rotations about the camera axis then certain views of a cube from *within* an octant are treated as being symmetric and mapped to a common point in D^n. The effect of these real and emergent symmetries is to collapse H and to cause it to become self-intersecting.

iv) Self-Occlusion

If objects are constrained to be transparent, [75], then all object features are visible for all values of $\boldsymbol{\lambda}$. Provided r is well-behaved the hypersurfaces of such objects will be continuous. The use of opaque objects on the other hand means that r must deal with rapid changes in the set of visible object features caused by self-occlusion. If r is sensitive to the presence of individual shape primitives in s then H may well be disjoint and grouped into regions corresponding to the notion of an aspect, Koenderink & van Doorn [55]. Within these regions, where the set of visible object features is quantitatively unchanging, the hypersurface should be smooth.

v) Noise

The description of hypersurface generation has thus far been limited to the set S_λ. The effect of noise on this process is now addressed by considering the mapping, by r, of S_η to D^n. Let $s \in S_\lambda$ be mapped to a point $\mathbf{c}$ in D^n. Let $A \subset S_\eta$ be the set of shapes produced by applying η to s. What is the mapping of A to D^n? If r is well-behaved under noise, in the sense that a small corruption of s produces a small displacement of $\mathbf{c}$, then A is mapped to a "cloud" of points C, centred on $\mathbf{c}$. The variance of C is

obviously related to the degree to which r is well-behaved under noise, ideally $C \equiv \mathbf{c}$. The expansion of each point $c \in H$ into a cloud of points will therefore cause the "thickening" of H.

The conclusion from this section is that, depending on the characteristics of the sets of objects and the representational function r, each hypersurface will be a locally smooth, low-dimensional manifold which may be self-intersecting, disjoint and thickened by noise.

5.4.2 Probabilistic Recognition

This section examines the nature of recognition based on these hypersurfaces. The application of Eq. 5.3 to the set of known objects $O : \{o_1 ... o_m\}$ generates a set of hypersurfaces $\mathbf{H} : \{H_1 ... H_m\}$ in D^n. We now define a function q, the inverse of f, which maps each vector $\mathbf{d} \in H_i$ to the identity o_i of an object, Eq. 5.4. The set $\mathbf{H}$ therefore defines the domain of q.

$$q : \boldsymbol{d} \in H_i \rightarrow o_i \tag{5.4}$$

An important factor in the ability of q to perform recognition is the degree to which individual hypersurfaces in $\mathbf{H}$ are seperated. If two hypersurfaces intersect at a point $\mathbf{k} \in D^n$, ie. $f(o_i, \lambda) \equiv f(o_j, \lambda') \equiv \mathbf{k}$, then given an approximation to q, recognition based on shapes whose representations fall in the region of $\mathbf{k}$ must necessarily be ambiguous. This implies that q should be revised to provide probabilistic, rather than absolute, classifications. If we consider recognition as a 1 from M problem then the distribution of each H_i provides an indicator of the probability density function of o_i in D^n. We therefore require, for each object o_i, the Bayesian *a posteriori* probability $P(o_i|\mathbf{d})$,

$$q_p : \mathbf{d} \rightarrow \begin{pmatrix} P(o_0|\mathbf{d}) \\ \cdot \\ \cdot \\ P(o_m|\mathbf{d}) \end{pmatrix} \tag{5.5}$$

The advantage of this probabilistic approach is that it provides principled behaviour in the presence of object symmetries, ambiguities between objects and the thickening of hypersurfaces caused by image noise. The next section presents the structure of an artificial neural network capable of approximating the function q_p to provide probabilistic, view-based recognition.

5.4.3 Neural Network Architecture

This section describes the use of an artificial neural network to approximate the probabilistic function q_p. This provides the basis for a flexible learning system that is able to generate the appropriate mapping through exposure to example views of the set of objects. The particular network architecture used has been previously shown to be capable of approximating Bayesian probabilities, [113]. This ensures that the system will provide optimal classification given the uncertainty caused by projective similarities between objects and by image noise. It has the further advantage that it places no restrictions on the distribution or structure of the data. This is important since the exact nature of the distribution of the hypersurfaces in feature space is not known.

The self-organising nature of the processing involved in the proposed network is common to many previously proposed systems, including Kohonen nets [56], Counter-propagation [45], Adaptive Resonance Theory [43], Competitive Learning [89], and CLAM, [102], and is related to the *k-means clustering* technique of standard pattern classification, (see Lippmann [61] for a review). The advantage of this approach over supervised learning, eg. achieved using a Multi-Layer-Perceptron, (MLP), [88], is in the ease with which the network can be trained.

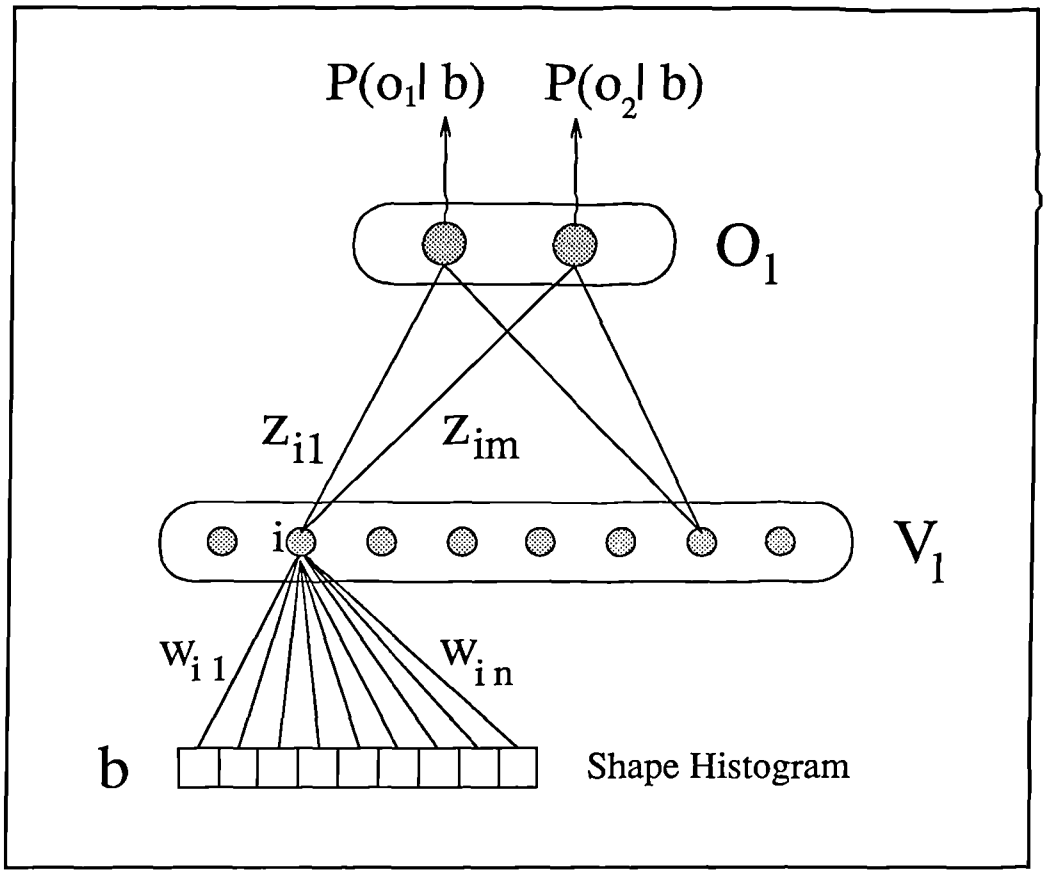

Figure 5–21: Network Architecture.

The network used to perform probabilistic classification has two layers, a *shape representation layer* (V_l) and an *object recognition layer* (O_l), figure 5–21. The purpose of

V_l is to distribute units throughout the space of possible shape representations D^n so as to cover the domain of the function q_p. The object layer O_l then encodes the value of q_p in the region of D^n covered by each unit in V_l. The function of each layer is now discussed in more detail.

Shape Representation Layer

The input to V_l is a set of unlabelled feature vectors I, each representing a particular view of an object, taken randomly, but with equal probability, from the set of hypersurfaces $\mathbf{H}$. Units in V_l are trained in an unsupervised manner using the standard competitive learning rule, [89]. If the current input vector is $\mathbf{b} \in I$ then the activation a_i of unit u_i in V_l is given by

$$a_i = \sum_{j=1}^{n} \sqrt{\mathbf{w}_{ij}.\mathbf{b}_j}$$

where $\mathbf{w}_i$ is a "weight" vector representing the position of unit u_i in D^n. The unit with maximum a_i, ie. the nearest unit, is taken to be the winner and is updated according to

$$\Delta \mathbf{w}_{ij} = \alpha(\mathbf{b}_j - \mathbf{w}_{ij})$$

where α is a variable that determines the learning rate of the system. This is gradually reduced throughout presentation of the training data such that the position of the units in D^n is forced to stabilise. Weights are re-normalised after being updated to ensure that their length remains constant. The conscience mechanism described in [45] is employed to guarantee that all units are recruited during training. This ensures that units in V_l distribute themselves throughout D^n so as to cover the statistical variation in the hypersurfaces $\mathbf{H}$, as sampled by the examples in I.

Once trained, units in V_i tessellate the feature space d^n into a series of *Voronoi cones*, (see Section 3.2.2), the size of each cone being inversely proportional to the number of units used in V_l.

Object Recognition Layer

The purpose of units in O_l is to approximate the value of the probabilistic function q_p in the region of D^n defined by the *Voronoi cone* associated with each unit in V_l. The distribution of hypersurfaces in D^n may be such that input points falling within the cone of a unit in V_l may represent views of more than one object. The value of $P(o_j|u_i)$ can be approximated by estimating the relative proportion of hypersurface H_m that falls within the *Voronoi cone* associated with u_i. This can be achieved by creating a connection $\mathbf{z}_{ij}$ between shape unit u_i and object unit x_j. These weights are trained by presenting a second, possibly different, set of example shapes to V_l. In order for the response of units in V_l to be calibrated these shapes must now be labelled with the identity of the viewed object. This is achieved by the supervisory vector $\mathbf{t}$, where $\mathbf{t}_j = 1$ for object o_j being viewed and $t_j = 0$ for all other objects. On each trial the output weights of the winning unit in V_l are updated according to

$$\mathbf{z}_{ij} = \mathbf{z}_{ij} + \mathbf{t}_j$$

Once training is complete the output vector of a shape unit u_i can be thought of as a "frequency of wins" histogram for each object. From this the estimated probability $P(o_j|\mathbf{b})$ can be encoded in connection $\mathbf{z}_{ij}$ simply by computing the ratio of the number of times u_i responded to a view of object o_j, to the total number of times that u_i has won,

$$\mathbf{z}_{ij} = \frac{\mathbf{z}_{ij}}{\sum_{j=1}^{m} \mathbf{z}_{ij}}$$

These layers combine to produce a network that is capable of learning the probabilistic function, q_p, mapping projected 2D shape to object identity.

5.4.4 Recognition Experiment

This section investigates the ability of the proposed system to perform view-based recognition. Ideally, this would be demonstrated using images captured from real objects. However, the problem of capturing the 100's of images of each object from different viewpoints in order to train the network was beyond the scope of the present study. Consequently, views are generated automatically from 3D, wire-frame models of objects, in this case aeroplanes. Models of four aeroplanes, an F-16, a BAe Hawk, a Jumbo Jet and a Sopwith Camel are shown in figure 5–22. These models were obtained by measuring the position of control points from scale models of the planes.

Generating a Data Set

In order to produce a data set that can be used to train the network it is necessary to compute the appearance of each object from a particular viewpoint, using the information contained in the wire-frame model. Unfortunately, the computations of true visibility conditions requires surface information, which was not included in the above models for reasons of simplicity. However, given the projection of a wire-frame model it *is* possible to compute the bounding contour, or silhouette, of the object. Various techniques can be used to achieve this, including carrying out a search for extremal lines, eg. Sykes [98]. This method has the advantage that the silhouettes can be computed directly from the projected position of 3D lines in the wire-frame models. This method was implemented but found to be overly sensitive to the projected position of model lines whose endpoints meet at a vertex.

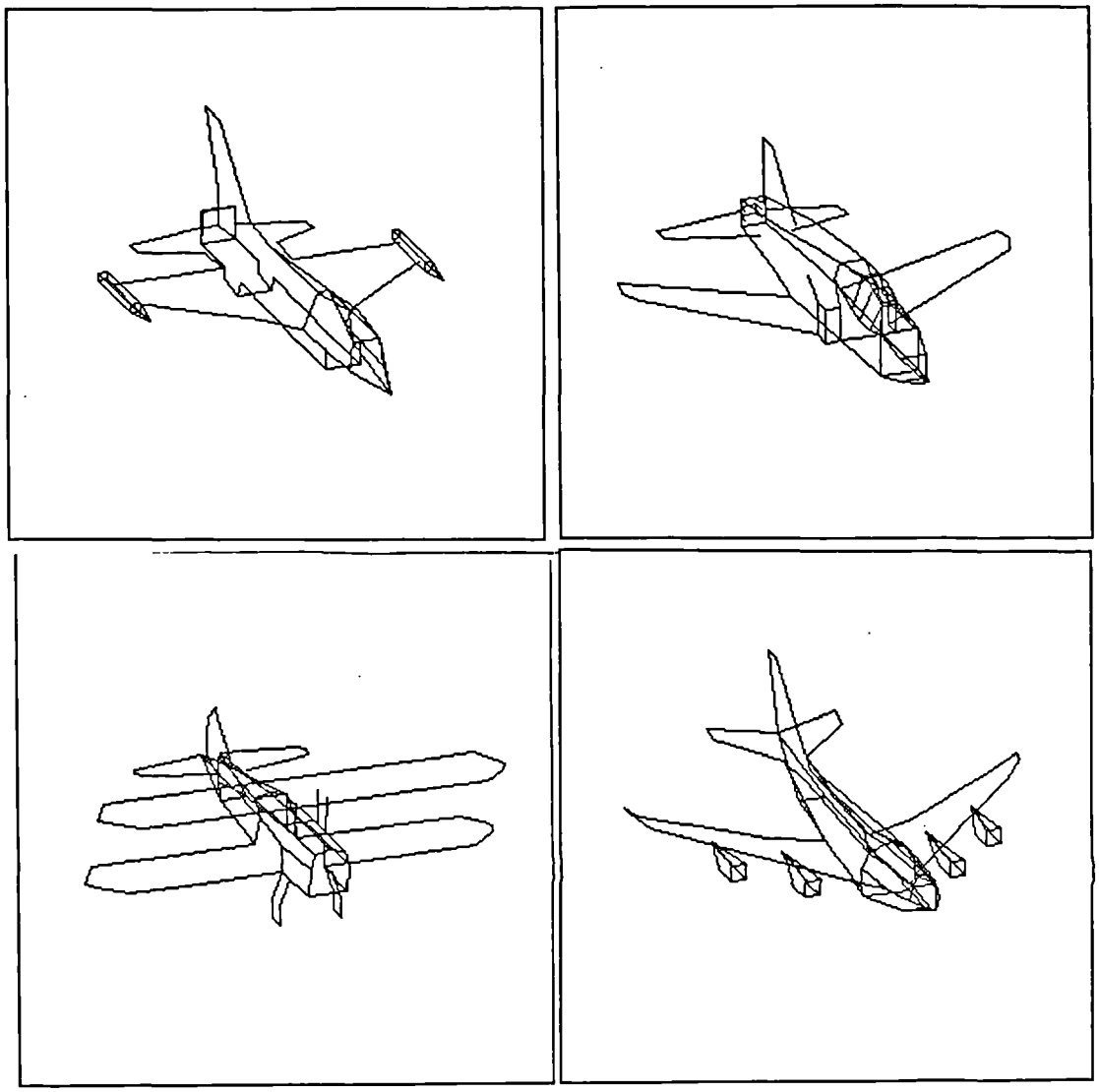

Figure 5–22: The planes used in the recognition experiment. Clockwise from top left, F-16, BAe Hawk, Jumbo Jet, Sopwith Camel.

The solution that was adopted involves first drawing each projected model line into an image, figure 5–23(a). A recursive flood-fill algorithm is them performed which sets all elements in the image that are outside the shape to 1 while setting those inside to 1, figure 5–23(b). Of course, at certain points in the rotation of the object this simple technique breaks down and produces a "false" silhouette, eg. figure 5–24. Such changes occur rapidly and pose an additional problem that must be handled by the recognition system. A set of lines describing the silhouette can be obtained by applying a Canny operator, (or any other edge detecting algorithm), to this image and performing a linear approximation of the resulting edgel strings. While this obviously involves more

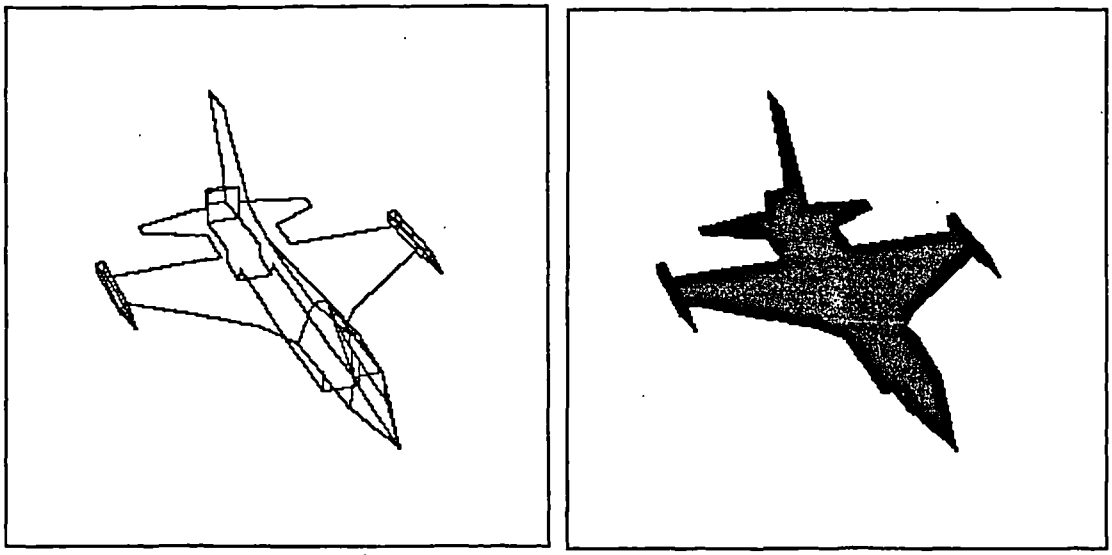

Figure 5–23: (a) the projected lines and (b) the result of applying the flood-fill algorithm.

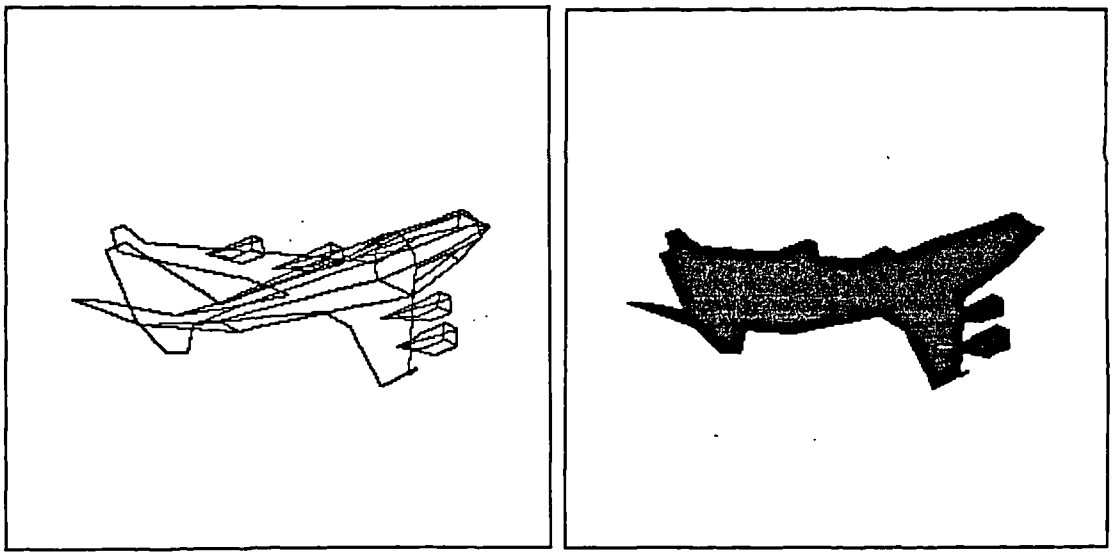

Figure 5–24: An example projection which produces a "false" silhouette.

processing than computing the silhouette directly from the projected wire-frame model, it was the only solution that was found to be robust across the whole view sphere.

A data set was obtained by computing the appearance of the objects at fixed intervals from a single quadrant of the view sphere. In particular, views were obtained at 5° intervals from [0° → 180°] azimuth, (ψ) and [0° → 90°] elevation, (ϕ), giving 703 views of each object. This was performed for each object, giving a total of 2812 views in the data set. The object-camera distance was fixed, such that changes in the apparent size of the objects were not considered. Each shape was processed and represented by a

single histogram recording the global geometric feature distribution, for which $n_\theta = 30$, $n_d = 20$, $\sigma_\theta = \sigma_d = 1$.

Training

The view representation layer was trained by presenting each view from the data set in a random order. This ensures that units are not "dragged around", following the latest set of inputs. Once the position of units in V_ℓ is fixed, the object representation layer is trained by presenting the training set once more.

Recognition

On presentation of an unknown shape the network produces a set of outputs indicating the probability of the unknown shape corresponding to a view of each object. This form of output is useful in that it effectively signals that when a shape is ambiguous, either through projective similarity between objects, image noise or because of the weakness of the representational scheme. It also provides a form of output that can be integrated with information from other systems, which similarly provide probabilistic information, to provide improved recognition. However, in order to evaluate the performance of the network, a forced decision regime was implemented in which the unknown shape is identified with the object having the highest probability.

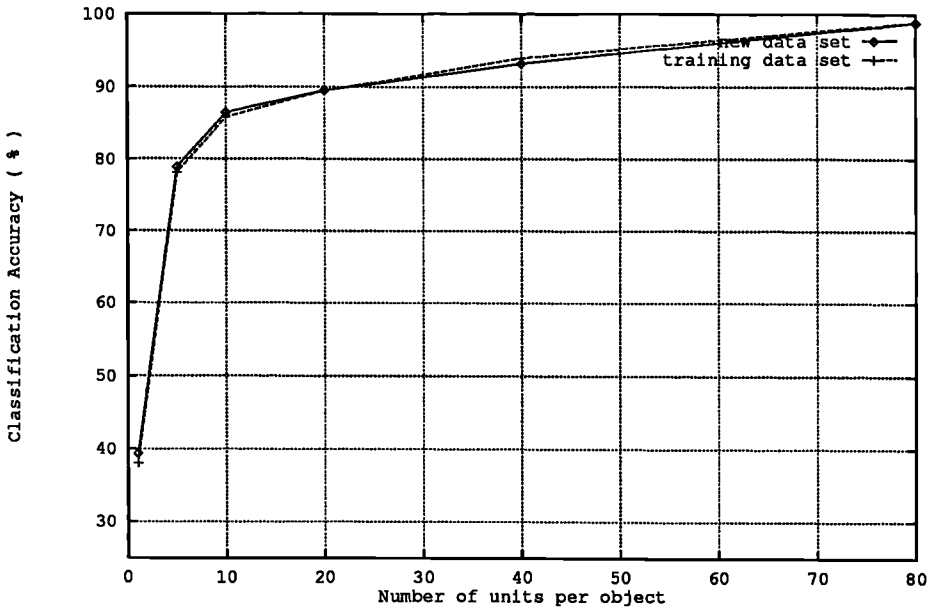

Figure 5–25: A graph showing classification accuracy for networks with different numbers of units.

The graph in figure 5–25 shows the classification error rate for networks with different numbers of units. Plotted on this graph are both the apparent error rates, obtained by testing the performance of the network on the training data set, and estimates of the true error rate obtained by testing the network on a different data set comprised of 500 views of each object taken from random values of ψ and ϕ. It can be seen from the graph that very good performance, (98.7% classification accuracy), can be

obtained storing only 80 views per object. This compares favourably with previous approaches based on nearest-neighbour classification, and using Fourier coefficients or moment invariants, which involve storing many more views of each object, (typically 500-1000), [25,83,107,85]. A confusion matrix describing the response of a network containing 40 units per object for this second data set is shown in figure 5–26.

	Hawk	Camel	Jumbo	F-16
Hawk	462	10	8	24
Camel	0	481	12	0
Jumbo	17	5	473	27
F-16	21	4	7	449

Figure 5–26: The confusion matrix for a network with 40 units per object.

Interestingly, it was found that extending the system by basing recognition on the interpolated output of the top k, (eg. $k = 3$), responding units drastically reduced performance. The likely reason for this is that units that are near to one another in feature space are more likely to represent views of different objects from the same viewpoint, rather than nearby views of the same object. Incorporating the response of the top k responding units therefore decreases the performance of the system.

The forced choice regime is a poor test of network performance, since units which represent views of more that one object, and which are therefore ambiguous, will, by definition, be wrong for a certain proportion of inputs. The performance of the network can be improved considerably by introducing a certainty threshold; a unit is then forced to make a choice only if its maximum probability is above a threshold value, C_T. If it is below this value then no classification is made. Figure 5–27 presents the performance of a network with 40 units per object at different levels of C_T. It can be seen that as the value of C_T is raised the performance of the system increases significantly, until at $C_T = 75\%$ the classification error rate has been raised to 98.6%. Of course, this result would not be of use if the proportion of inputs for which no decision could be made became too large. However, it can be seen that even at the $C_T = 75\%$ level, almost 89% of inputs are still being classified.

Cases where the ambiguity of the network's response is too high to support classification could be handled by initiating some form of action, in an attempt to resolve the ambiguity. A primary candidate for such an action would be to make a change in viewpoint, hopefully providing a less ambiguous view of the object. Moreover, the transitions between views can be used to provide a disambiguating source of infor-

C_T	Classification Accuracy (%)	Views Classified (%)
25	93.25	100.0
35	93.62	99.6
45	94.14	98.2
55	95.90	95.2
65	98.06	90.1
75	98.60	88,8
85	98.97	87.4
95	99.64	84.8

Figure 5–27: A table showing system performance for different values of C_T.

mation, eg. [91]. The construction of an aspect graph encoding the allowable links between views of an object, (possibly containing probabilistic information), is an area for further research.

Local Shape Matching

While the work presented in this section has been restricted to recognition based on global geometric feature distributions, and so is limited to the recognition of unoccluded objects, the scheme holds the potential to deliver view-based recognition based on local shape elements, achieved through the matching of local geometric feature distributions. This would enable view-based recognition to be achieved in cluttered scenes, and would inherit the robust properties established in Chapter 3. This issue is discussed further in Chapter 6.

5.5 Discussion and Summary

This chapter has presented the application of the GFD representational scheme to the problem of 3D object recognition. Specifically, two different approaches were investigated.

The first, termed the *3D approach*, involved generalising the GFD scheme to handle the problem of 3D shape representation and matching. Extending the scheme to deal with 3D shape representation involved simply defining a set of geometric features between 3D line segments. Local geometric feature distributions representing 3D scene and model lines can then be matched using essentially the same scheme as that proposed

for performing 2D object recognition, despite the difficulties caused by self-occlusion. A strategy was proposed for using these matches to determine the position and orientation of objects. Although certain difficulties were encountered, due both to the symmetry of the objects and to the non-directionality of 3D line segments, the system was shown to be capable of accurately locating objects in a scene.

Despite this success, there are a number of difficulties involved in basing recognition on the matching of 3D scene and model descriptions. Firstly there is the computational load involved in obtaining 3D scene descriptions. Secondly, the need to model the explicit 3D structure of the objects to be recognised means that the range of objects to which the approach can be applied is limited. Finally, even in cases where such modelling is appropriate, the problem of constructing such models remains, and often has to be achieved through hand coding. These difficulties combine to motivate the investigation into an alternative approach to 3D object recognition which does not rely on the representation of 3D shape.

The second half of this chapter described the use of the GFD scheme within a *multiple view-based approach* to 3D object recognition. This assumes that performing recognition involves matching the 2D, projected shape information found in a single image to 2D, appearance-based models, composed of a relatively small number of example views of each object. Describing the application of the GFD scheme within this approach involved first analysing the source of the variation in 2D, projected shape. This was then used as the basis for a description of the process by which the representation of the set of all possible views of an object generates a hypersurface in feature space. The likely behaviour of such hypersurfaces was related to properties of objects and to the characteristics of the scheme used to represent views. It was argued that there are numerous reasons why a view-based system should be able to provide probabilistic recognition information, eg. to signal the presence of ambiguity due to projective similarity, weakness in the representational scheme or image noise. The structure of a self-organising neural network capable of constructing view-based representations by clustering views on the basis of similarity was presented and shown to be capable of providing accurate classification of a set of aeroplanes.

Chapter 6

SUMMARY

This chapter summarises the contribution of the work presented in this thesis and highlights directions for further study.

6.1 Contribution

This thesis has presented a novel form of shape representation that is able to support the recognition and localisation of both 2D and 3D objects under conditions of severe fragmentation noise, occlusion and clutter.

The main features of this work are:

1. **The development of a novel scheme for representing shape.** This is based on recording the distribution of geometric features computed between pairs of primitive elements within a shape.

   The main features of this form of shape representation are:

   - Invariance. The properties of the geometric features upon which the scheme is based mean that representations are invariant to 2D transformations in position and orientation, in the case of 2D shape, and full object transformation in the case of 3D shape.
   - Robustness. The fact that geometric feature distributions are based upon combining multiple local measurements of shape mean that they degrade gracefully as shape information is lost through image noise or occlusion. Particular attention was paid to ensuring that representations constructed from line-based descriptions of shape retained this property.
   - Strength. The factors affecting the strength, or uniqueness, of geometric feature distributions were assessed. It was shown that the proposed scheme possesses sufficient strength to allow discrimination between most distinct

shape primitives, while the particular weaknesses of the representation provide considerable advantages in the representation of fragmented elements of a line segment.

- Redundancy. The fact that the construction of full, local representations of a shape involves recording multiple geometric feature distributions, one for each primitive, means that recognition based on the representation should be robust to the loss of lines through image noise or occlusion.
- A method of encoding allowable shape variation. The quantising effect of the binning process, together with the blurring of entries across many bins, provides a principled method for encoding the range of geometric feature values that may arise within a shape.
- A scalable representation. The scale at which shape is represented can be varied simply by changing the parameters of the histogram, ie. resolution and width of blur, used to record geometric feature distributions.
- Flexibility in the level of shape representation. By varying the set of geometric relationships recorded within a histogram it is possible to represent shape either locally or globally. The scheme is therefore able to support two forms of recognition, correspondence recognition, in which local shape elements are matched, and non-correspondence recognition, in which whole shapes are matched.
- Flexibility in the dimensionality of shape represented. The scheme can be adapted for the representation of 2D or 3D shape by simply defining the set of geometric features between either 2D or 3D shape primitives.
- A local form of representation. The range over which geometric relationships are measured can be limited to provide a representation of local shape. This effectively reduces the likelihood of the representation of a particular shape primitive being affected by fragmentation noise, occlusion or scene clutter.
- Accessibility. Representations can be constructed straightforwardly from descriptions of shape in terms of a set of primitive elements.
- Versatility. The only restriction placed on the type of shape that can be represented is that they can be described to a sufficient degree of accuracy using straight line segments.

2. **A pattern classification approach to recognition.** One of the primary aims of the work presented in this thesis was, as stated in Chapter 1, to develop a representational scheme that provided shape encodings which could be matched using techniques from statistical pattern classification. The fact that histograms recording the distribution of geometric feature values can be regarded as recording the probability of co-occurrence of two lines at a particular geometric relationship within a shape means that this has been achieved. This has a number of

advantages. Firstly, given a suitable metric of similarity defined between two shape encodings, in this case the *Bhattacharrya distance*, optimal classification accuracy can be achieved using the nearest-neighbour classification rule, modulo the strength of the representational scheme. Secondly, the processing involved in computing the similarity metric and in implementing the classification rule is both simple and uniform, and so lends itself directly to implementation in parallel hardware.

3. **An analysis of the effects of shape variation.** Crucial to the practicality of a recognition system is its ability to support matching across the full range of potential imaging situations. In the present system this requires that the properties of the representational scheme, similarity metric and classification rule combine to provide robustness to variations in the image shape description. It was deemed important that the performance of these elements be examined under *measurable* conditions of shape variation, which involved proposing generative models of each source of shape variation. Particular attention was paid to the effect of changes caused by fragmentation noise, scene clutter and sensor error. It was found that, under the particular models of shape variation used, the properties of the representational and matching schemes combined to provide theoretical robustness to very high levels of fragmentation noise and scene clutter. Providing a definite conclusion for the performance of the system under sensor error was more difficult, but the factors determining the effect of such changes on the system were fully analysed.

4. **A parallel solution.** The amount of computation needed in recording and matching geometric feature distributions is obviously quite large, and recognition in complex scenes can be expected to be relatively slow on conventional, serial machines. However, the fact that the processing involved in all stages of recognition is both local, simple and uniform, means that the scheme is well suited to implementation in parallel hardware. This can be expected to produce a significant increase in the speed of recognition.

5. **A memory/computation trade-off.** The need to store histograms for each primitive within a shape places heavy depends on memory. However, it seems fair to say that there is an inherent trade-off in recognition between the amount of memory needed to store object representations and in the speed and simplicity of recognition. In the present scheme much effort has been put into producing a shape representation with the desired properties of invariance, strength and robustness. This has produced a form of shape representation that requires a large amount of memory. This is justified on two grounds. Firstly, it enables matching to be performed through simple, local processing that can be implemented in hardware. Secondly, memory is a relatively cheap component of a system, and so the heavy demands of the proposed scheme do not effect its practical application.

6. **2D Object Recognition.** A scheme based on the classification of geometric feature distributions, together with the generalised Hough transform, was shown to be capable of recognising a range of objects their 2D projection, the objects being viewed from a fixed viewpoint. As expected, given the results of the noise analysis, the system based on the matching of local geometric feature distributions was able to recognise objects despite the problems caused by severe fragmentation noise and occlusion. This constitutes an advance over many previous pattern classification approaches that have been restricted by their reliance on global shape representation to the recognition of un-occluded objects.

7. **3D Shape Matching.** It was shown that the representational scheme was capable of supporting the matching of 3D scene and model shape descriptions. The scheme was extended to deal with 3D shape representation by simply proposing a set of geometric features defined between 3D line segments. The local geometric feature distributions representing 3D scene and model lines were then matched using essentially the same mechanism as for 2D object recognition. This matching was shown to be robust to the loss of shape information caused by self-occlusion. The line matches delivered by this scheme were used, together with an alignment approach, to determine the 3D position and orientation of objects in a scene.

8. **A Multiple View-Based Solution.** The GFD scheme was shown to be capable of supporting an alternative approach to 3D object recognition based on the matching of 2D image data to 2D, appearance-based object models. An understanding of the problems involved in performing multiple view-based recognition was aided by an analysis of the process by which the set of all possible views of an object forms a hypersurface in feature space. The effects of object properties and characteristics of the representational scheme on the behaviour of these hypersurfaces was discussed. The structure of a self-organising artificial neural network was presented which was able to cluster, on the basis of similarity, 2D views of an object, represented as global geometric feature distributions, in order to construct appearance-based object models.

   One of the main advantages of the proposed view-based system is its ability to provide probabilistic recognition information. This has a number of advantages. Firstly, it means that the output of the system can be combined with that from other recognition systems, which exploit alternative sources of information, in order to provide improved recognition. Secondly, the system is able to signal the ambiguity of a particular view, due either to projective similarity between objects, image noise or weakness of the shape encoding. This is important as it can be used to trigger actions that might resolve this ambiguity, eg. a change in viewpoint.

6.2 Further Work

Improvements could be made in the following areas:

6.2.1 Representation

Both the theoretical analysis and the empirical results presented in this thesis have hopefully combined to demonstrate the suitability of geometric feature distributions as a scheme for representing shape. However, various improvements could be made:

- Alternative geometric features could be proposed which are invariant to scale changes.

- In cases where the 2D shape found in an image is the result of the projection of a 3D object, the line segments describing this shape cannot be directed by exploiting the sign of the image intensity gradient. This results in a considerable loss of strength in the representation. Alternative strategies for directing line segments could be investigated. A possible solution is to direct pairs of line segments away from their point of intersection, although care must obviously be taken to ensure that this does not affect the robustness of the representation.

- The form of the blurring function used on each axis of the histogram could be better adapted to describe the true distribution of geometric feature values arising from image noise.

- The method for determining the optimal parameters of the representation for a particular task application are somewhat *ad hoc*. Basically, the parameters have to be varied until the performance of the system is deemed acceptable. Ideally, these parameters would be set automatically, based on an analysis of the statistics of the input data for a particular application. For example, given a set of objects that are to be viewed under a specified range of lighting conditions, it should be possible to determine optimum values for the resolution and width of blur used in the histogram by relating them to some measure of the mean variation in geometric feature values across a sample of typical images. Similarly, the optimal size of the local region could be determined by finding an acceptable balance between the strength of representations, as gauged by the number of misclassifications, and the likelihood of occlusion.

6.2.2 Improving Efficiency

The main aim of the work presented in this thesis has been to demonstrate the feasibility of representing shape by recording the distribution of geometric feature values measured between its primitive elements. Relatively little effort has been put into developing strategies for improving the efficiency with which recognition is performed. There are obviously many ways in which this could be achieved. One of the simplest methods would be to develop a sequential implementation of the recognition system. The set of complete matches produced by the proposed 2D recognition scheme, while useful for providing a first order segmentation of the image shape description, is largely redundant for the purposes of computing the position and orientation of objects in the scene; for this the matching of two non-parallel model lines to the correct image lines is sufficient. This fact could be exploited to provide an alternative recognition system, in which local geometric feature distributions representing individual line segments were recorded and matched sequentially. Once sufficient evidence had been accumulated for the presence in the scene of a particular object at a certain pose, an attempt would be made to validate the hypothesis. All matched image lines would then be removed and the hypothesis/validation cycle repeated, (cf. [3]). This "sequential" recognition scheme could be expected to be much quicker than the parallel scheme, although its robustness in conditions of high scene clutter is not so certain.

6.2.3 Hardware Implementation

One of the most important considerations in assessing the potential of an algorithm for practical application is the ease with which it can be implemented in hardware. As mentioned above, the processing involved in both the recording and matching of geometric feature distributions is simple, local and uniform, attributes which should make such implementation relatively straightforward. Indeed, research has recently begun on the feasibility of designing dedicated processors for performing specific tasks within the recognition system, [101]. The two main areas of interest are:

- The design of a processor dedicated to the task of computing and recording the distribution of geometric feature values between line segments.

- The use of an existing processor to perform the array multiplication involved in matching geometric feature distributions.

The successful incorporation of these elements into a recognition system implemented in dedicated hardware can be expected to bring the the time involved in matching complex scenes to large numbers of objects down to practical levels.

6.2.4 Extending the Multiple View-Based Approach

One of the most exciting areas of further research is in extending the multiple view-based approach. Of particular interest is the extension of the proposed system to deal with the matching of local shape elements. The view-based approach was demonstrated using global geometric feature distributions. While these are robust to the loss of data caused by fragmentation noise they cannot deal with cluttered scenes. This requires that recognition be based on the matching of local elements of shape, which necessitates the use of local geometric geometric feature distributions. The immediate difficulty with extending the view-based system to deal with local shape matching is in the amount of memory and computation involved in storing and matching histograms for each line segments within each stored view of each object. It would be interesting to investigate the use of the measures described above to overcome these problems in order to make recognition tractable. If successful then the resulting system would provide an adaptive, flexible 3D object recognition system that could operate under conditions of considerable image noise and scene clutter.

Another interesting area in which the view-based approach could be extended is the use of temporal information in resolving ambiguities. This would involve adapting the neural network architecture so as to enable it to incorporate information regarding the characteristic transitions between stored views for each object. If the shape extracted from an image was signalled by the system as being ambiguous, due to projective similarity between objects or to the weakness of the representational scheme, then a change in viewpoint could be initiated. Information stored in the object models regarding the temporal adjacency of views could then be exploited to resolve the ambiguity. The system proposed by Seibert & Waxman [91] would provide a starting point for this work.

6.2.5 Modelling Higher Level Recognition Processes

The recognition system proposed in this thesis operates in order to answer to the question "*Which* known objects are in the scene and *where* are they?". The recognition problem could obviously be phrased differently. For example, one can imagine that an intelligent agent, interested in achieving some task, might be more interested in answering the question "Is there an X in the scene?", where X refers to some known object. While this question can eventually be answered in the proposed system, one might expect that a truly intelligent agent would possess certain strategies, schemas or routines for answering specific queries about the presence of entities in the scene. Investigating strategies by which high-level, knowledge-based processes may guide or direct low-level visual processes in the pursuit of specific information about the world provides an exciting area for further research. Of particular interest is the incorporation of such processes into the sequential form of recognition described in Section 6.2.2.

Appendix

It is shown that the maximum of the *Bhattacharrya* similarity metric $D_{ij} = \sum_i^n \sqrt{o_i}\sqrt{m_i}$ is the minimum of a χ^2 variable comparing two frequency distributions o_i and m_i.

The maximum likelihood statistic χ^2 for comparing two distributions o_i and m_i is defined as

$$\chi^2 = \sum_i^n (o_i - m_i)^2/m_i \qquad (1)$$

for small $(o_i - m_i)$, the first order Taylor expansion of f at m_i can be written as

$$f(o_i) \approx f(m_i) + (o_i - m_i)\frac{\partial f(m_i)}{\partial m_i} \qquad (2)$$

which gives

$$(o_i - m_i) \approx \frac{f(o_i) - f(m_i)}{\frac{\partial f(m_i)}{\partial m_i}} \qquad (3)$$

Substituting (3) in (1) gives

$$\chi^2 = \sum_i^n \frac{(f(o_i) - f(m_i))^2}{(\frac{\partial f(m_i)}{\partial m})^2 m_i} \qquad (4)$$

In the special case of $f(x) = \sqrt{x}$ we have

$$\chi^2 = 4\sum_i^n (\sqrt{o_i} - \sqrt{m_i})^2$$

which expanded gives

$$\chi^2 = 4\sum_i^n o_i + 4\sum_i^n m_i - 8\sum_i^n \sqrt{o_i}\sqrt{m_i}$$

which for normalised m gives

$$\chi^2 = const - 8\sum_i^n \sqrt{o_i}\sqrt{m_i}$$

Thus, under these assumptions, taking the maximum of the *Bhattacharrya distance*, D_{ij} is the same as taking the minimum of the χ^2 statistic.

References

[1] A.P. Ambler, H.G. Barrow, C.M. Brown, R.M. Burstall, and R.J. Popplestone. A versatile computer controlled assembly system. *Artificial Intelligence*, 6(2):129–156, 1975.

[2] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61:183–193, 1954.

[3] N. Ayache and O.D. Faugeras. Hyper: A new approach for the recognition and positioning of 2d objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):44–54, 1986.

[4] D.H. Ballard. Generalizing the hough transform to detect arbitrary patterns. *Pattern Recognition*, 13(2):111–122, 1981.

[5] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice Hall, 1982.

[6] D.H. Ballard and D. Sabbah. On shapes. In *Proc. 7th Int. Conf. Artificial Intelligence, IJCAI*, pages 607–612, 1981.

[7] R. Basri and S. Ullman. Linear operator for object recognition. In M.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Network Information Processing Systems 4*, pages 452–459. Morgan Kauffman, San Mateo, CA., 1992.

[8] I. Beiderman. Recognition by components: A theory of human image understanding. *Psychological Review*, 92(4):115–147, 1987.

[9] B. Bhanu. Representation and shape matching of three-dimensional objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(3):340–351, 1984.

[10] B. Bhanu and O.D. Faugeras. Shape matching of two-dimensional objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(2):137–155, 1984.

[11] H.H. Bűlthoff and S. Edelman. Psychophysical support for a 2d view interpolation theory of object recognition. *Proc. Nat. Acad. Sci.*, 89:60–64, 1992.

[12] R.C. Bolles. Robust feature matching through maximal cliques. In *Proc. SPIE Tech. Symp. Imaging Appl. Automated Industrial Inspect. Assembly*, 1979.

[13] R.C. Bolles and R.A. Cain. Recognizing and locating partially visible objects: The local-feature-focus method. *Int. J. Robotics Res.*, 1(3):57–82, 1982.

[14] R.C. Bolles and P. Horaud. 3dpo: A three-dimensional part orientation system. *Int. J. Robotics Res.*, 5(3):3–26, 1986.

[15] A.J. Bray. Object recognition using local geometric constraints: A robust alternative to tree search. In *Proc. The First European Conf. on Computer Vision ECCV*, pages 499–515, 1990.

[16] A.J. Bray. Properties of local geometric constraints. In *Proc. British Machine Vision Conf. BMVC91*, pages 95–103, 1991.

[17] N. Burgess and M.N. Granieri. A growing network classifier for 3-d objects using multiple views. In *Proc. 11th Int. Conf. on Pattern Recognition*, volume 2, pages 512–515, 1992.

[18] N. Burgess, M.N. Granieri, and S. Paternello. 3-d object classification: Application of a constructive algorithm. *Int. J. Neural Systems*, 2(4):275–282, 1992.

[19] J.B. Burns, R.S. Weiss, and E.M. Riseman. The non-existence of general case view invariants. In J.L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 120–131. MIT Press, Cambridge, Mass., 1992.

[20] J.B. Burns, R.S. Weiss, and E.M. Riseman. View variation of point-set and line segment features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(1):51–68, 1993.

[21] A. Califano and R. Mohan. Multidimensional indexing for visual recognizing shapes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 28–35, 1991.

[22] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[23] I. Chakravarty and H. Freeman. Characteristic views as a basis for 3d object recognition. *Proc. SPIE Robot Vision*, 336:37–45, 1982.

[24] L. Davis. Shape matching using relaxation techniques. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1(1):60–72, 1979.

[25] S.A. Dudani, K.J. Breeding, and R.B. McGhee. Aircraft identification by moment invariants. *IEEE Trans. on Computers*, 26(1):39–45, 1977.

[26] S. Edelman. On learning to recognize 3d objects from examples. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(8):833–837, 1993.

[27] S. Edelman and T. Poggio. Bringing the grandmother back into the picture: a memory-based view of object recognition. In J. Skrzypek and W. Karplus, editors, *Neural Networks in Vision and Pattern Recognition*, pages 37–61. World Scientific, Singapore, 1992.

[28] S. Edelman and D. Weinshall. A self-organizing multiple-view representation of 3d objects. *Biological Cybernetics*, 64:209–219, 1991.

[29] A.C. Evans. The use of geometric histograms for 2d object recognition. *Image Processing Magazine*, 5(3):18–22, 1993.

[30] A.C. Evans, N.A. Thacker, and J.E.W. Mayhew. Pairwise representations of shape. In *Proc. 11th Int. Conf. on Pattern Recognition*, volume 1, pages 133–136, 1992.

[31] A.C. Evans, N.A. Thacker, and J.E.W. Mayhew. A practical view-based 3d object recognition system. In *Proc. 3rd Int. conf. on Artificial Neural Networks*, pages 6–10, 1993.

[32] A.C. Evans, N.A. Thacker, and J.E.W. Mayhew. The use of geometric histograms for model-based object recognition. In *Proc. British Machine Vision Conf. BMVC93*, volume 2, pages 429–438, 1993.

[33] G. Fekete and L. Davis. Property spheres: a new representation for 3d object recognition. In *Proc. IEEE Workshop on Computer Vision: Representation and Control*, pages 192–201, 1984.

[34] R.B. Fisher. Using surfaces and object models to recognise partially obscured objects. In *Proc. 8th Int. Conf. Artificial Intelligence, IJCAI*, pages 989–985, 1983.

[35] H. Freeman and I. Chakravarty. The use of characteristic views in the recognition of three-dimensional objets. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice*, pages 277–288. North-Holland, 1980.

[36] C. Goad. Special purpose automatic programming for 3d model-based vision. In *Proc. Image Understanding Workshop, Virginia, USA*, pages 94–104, 1983.

[37] C. Goad. Fast 3d model-based vision. In A.P. Pentland, editor, *From Pixels to Predicates*, pages 317–391. Norwood, NJ, 1985.

[38] J.W. Gorman, O.R. Mitchell, and F.P. Kuhl. Partial shape recognition using dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(2):257–266, 1988.

[39] G.H. Granlund. Fourier preprocessing for hand print character recognition. *IEEE Trans. on Computers*, 21(2):195–201, 1972.

[40] W.E.L. Grimson. *Object Recognition by Computer - The Role of Geometric Constraints.* MIT Press, Cambridge, MA., 1990.

[41] W.E.L. Grimson and T. Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *Int. Journal Robotics Research*, 3(3):3–35, 1984.

[42] W.E.L. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.

[43] S. Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11:23–63, 1987.

[44] R.M. Haralick. Performance characterization in computer vision. In *Proc. British Machine Vision Conf. BMVC92*, pages 1–8, 1992.

[45] R. Hecht-Neilsen. *Neurocomputing.* Addison-Wesley Publishing, 1990.

[46] D.D. Hoffman and W.A. Richards. Parts of recognition. In A.P. Pentland, editor, *From Pixels to Predicates*, pages 268–293. Norwood, NJ, 1985.

[47] M.K. Hu. Visual pattern recognition by moment invariants. *IEEE Trans. on Information Theory*, 8:179–187, 1962.

[48] G.K. Humphrey and S.C. Khan. Recognizing novel views of 3d objects. *Canadian Journal of Psychology*, 46(2):170–190, 1992.

[49] G.W. Humphrey and P.T. Quinlan. Normal and pathalogical processes in visual object constancy. In G.W. Humphrey and M.J. Riddoch, editors, *Visual Object Processing: A Cognitive Neuropsychological Approach*, pages 43–105. Lawrence Erlbaum, 1987.

[50] D.P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proc. First Int. Conf. Computer Vision*, pages 102–111, 1987.

[51] N. Intrator, J.I. Gold, H.H. Bűlthoff, and S. Edelman. 3d object recognition using unsupervised feature extraction. In D.S. Touretzky, editor, *Advances in Neural*

Network Information Processing Systems 4, pages 460–467. Morgan Kauffman, San Mateo, CA., 1992.

[52] A. Kalvin, E. Schonberg, J.T. Schwartz, and M. Sharir. Two-dimensional model-based, boundary matching using footprints. *Int. J. Robotics Research*, 5(4):38–55, 1986.

[53] J.R. Kender and D.G. Freudenstein. What is a degenerate view? In *Proc. 10th Int. Joint Conf. on Artificial Intelligence, IJCAI*, pages 801–804, 1987.

[54] J.R. Kender and D.G. 1987. Freudenstein. What is a degenerative view? In *Proc. of DARPA Image Understanding Workshop*, pages 589–598, 1987.

[55] J.J. Koenderink and A.J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–217, 1979.

[56] T. Kohonen. *Self-Organisation and Associative Memory.* Springer-Verlag, 1989.

[57] Y. Lambdan, J.T. Schwartz, and H.J. Wolfson. Affine-invariant model-based object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(5):578–589, 1990.

[58] Y. Lambdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition system. In *Proc. 2nd Int. Conf. on Computer Vision*, pages 238–249, 1988.

[59] V.F. Leavers. *Shape Detection in Computer Vision Using the Hough Transform.* Springer-Verlag, 1992.

[60] C.C. Lin and R. Chellappa. Classification of partial 2d shapes using fourier descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:102–105, 1986.

[61] R.P. Lippmann. An introduction to computing with neural networks. *IEEE ASSP Magazine*, pages 36–54, April 1987.

[62] D.G. Lowe. *Perceptual organization and visual recogntiion.* Kluwer, Boston, 1985.

[63] D.G. Lowe. Three-dimensional object recognition from two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[64] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis.* Academic Press, 1979.

[65] D. Marr. *Vision.* W.H. Freeman, 1982.

[66] S. Marshall. Review of shape coding techniques. *Image and Vision Computing*, 7(4):281–294, 1989.

[67] P. McAndrew and A.M. Wallace. Interpretation of 2d scenes using a general relational model. In *Proc. 3rd Alvey Vision Conference*, pages 107–115, 1987.

[68] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry.* The MIT Press, Cambridge, MA., 1969.

[69] D.J.H. Moore and D.J. Parker. Analysis of global pattern features. *Pattern Recognition*, 6:149–164, 1974.

[70] D.W. Murray. Model-based recognition using 3d shape alone. *Computer Vision, Graphics and Image Processing*, 40(2):250–266, 1987.

[71] D. Noll, M. Schwarzinger, and W. Seelen. Contextual feature similarities for model-based object recognition. In *Proc. Int. Conf. Computer Vision*, pages 286–290, 1993.

[72] D.I. Perret, A.J. Mistin, and A.J. Chitty. Visual neurons responsive to faces. *Trends in Neuroscience*, 10(9):358–363, 1987.

[73] D.I. Perret, P.A.J. Smith, A.J. Mistlin, A.S. Head, A.D. Milner, and M.A. Jeeves. Visual cells in the temporal cortex sensitive to face view and gaze direction. *Royal Society London, Series B*, 223:293–317, 1985.

[74] E. Persoon and K.S. Fu. Shape discrimination using fourier descriptors. *IEEE Trans. on Systems, Man and Cybernetics*, 7(3):170–179, 1977.

[75] T. Poggio and S. Edelman. A network that learns to recognise 3d objects. *Nature*, 343:263–266, 1990.

[76] S.B. Pollard, J.E.W. Mayhew, and J.P. Frisby. Pmf: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[77] S.B. Pollard, J. Porrill, J.E.W. Mayhew, and J.P. Frisby. Matching geometrical descriptions in three-space. *Image and Vision Computing*, 5(2):73–78, 1987.

[78] J. Porrill, S.B. Pollard, T.P. Pridmore, J.B. Bowen, J.E.W. Mayhew, and J.P. Frisby. Tina: A 3d vision system for pick and place. In *Proc. 3rd Alvey Vision Conf.*, 1987.

[79] J. Porrill, S.B. Pollard, T.P. Pridmore, J.B. Bowen, J.E.W. Mayhew, and J.P. Frisby. Tina: The sheffield vision system. In *Proc. 9th Int. Conf. on Artificial Intelligence IJCAI*, 1989.

[80] T.P. Pridmore, J. Porrill, and J.E.W. Mayhew. Segmentation and description of binocularly viewed contours. Aivru memo no. 16, University of Sheffield, Sheffield, England., 1989.

[81] R.J. Prokop and A.P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54(5):438–460, 1992.

[82] S.J. Rak and P.J. Kolodzy. Performance of a neural network based 3-d object recognition system. *SPIE Automatic Object Recognition*, 1471:177–184, 1991.

[83] A.P. Reeves, R.J. Prokop, S.E. Andrews, and F.P. Kuhl. Three-dimensional shape analysis using moments and fourier descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(6):937–943, 1988.

[84] H.J. Reitboeck and J. Altmann. A model for size and rotation invariant pattern processing in the visual system. *Biological Cybernetics*, 51:113–121, 1984.

[85] C.W. Richard and H. Hemani. Identification of 3d objects using fourier descriptors of the boundary curve. *IEEE Trans. on Systems, Man and Cybernetics*, 4(4):371–378, 1974.

[86] I. Rock and J. DiVita. A case of viewer-centred object perception. *Cognitive Psychology*, 19:280–293, 1987.

[87] C.A. Rothwell, A. Zisserman, D.A. Forsyth, and J.L. Mundy. Fast recognition using algebraic invariants. In J.L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 398–407. MIT Press, Cambridge, Mass., 1992.

[88] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, pages 318–362. MIT Press, Cambridge, MA, 1986.

[89] D.E. Rumelhart and D. Zipser. Feature discovery by competitive learning. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, pages 151–193. MIT Press, Cambridge, MA, 1986.

[90] M. Seibert and A.M. Waxman. Learning aspect graph representations from view sequences. In D.S. Touretzky, editor, *Advances in Neural Network Information Processing Systems 2*, pages 258–265. Morgan Kauffman, San Mateo, CA., 1990.

[91] M. Seibert and A.M. Waxman. Adaptive 3d object recognition from multiple views. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):107–213, 1992.

[92] M.C. Seibert. *Neural Networks for Machine Vision: Learning 3D Object Representations.* PhD thesis, Boston University Graduate School, 1991.

[93] A. Silberberg, S. Hardy, L. Davis, and D. Harwood. An iterative hough procedure for three-dimensional object recognition. *Pattern Recognition*, 17(6):621–629, 1984.

[94] A. Silberberg, S. Hardy, L. Davis, and D. Harwood. Object recognition using oriented model points. *Computer Vision, Graphics and Image Processing*, 33:45–71, 1986.

[95] S.P. Smith and A.K. Jain. Chord distributions for shapo matching. *Computer Graphics and Image Processing*, 20:259–271, 1982.

[96] F. Stein and G. Medioni. Structural indexing: Efficient 2d object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(12):1198–1204, 1992.

[97] F. Stein and G. Medioni. Structural indexing: Efficient 3d object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.

[98] F.P. Sykes. Hypothesis and verification in 3-d model matching. Master's thesis, University of Sheffield, 1989.

[99] M. Tarr and S. Pinker. Mental rotation and orientation dependence in shape recognition. *Cognitive Psychology*, 21:233–282, 1991.

[100] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991.

[101] N.A. Thacker. An analysis of pairwise geometric histograms and their suitability for hardware implementation. Ssg report 93/26, University of Sheffield, Sheffield, England., 1993.

[102] N.A. Thacker and J.E.W. Mayhew. Designing a layered network for context sensitive pattern classification. *Neural Networks*, 3(3):291–300, 1989.

[103] S. Ullman. Low-level aspects of segmentation and recognition. In H.B. Barlow, J.P. Frisby, A. Horridge, and M.A. Jeeves, editors, *Natural and Artificial Low-Level Seeing Systems*, pages 119–126. Oxford Science Publications, 1993.

[104] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(10):992–1005, 1991.

[105] S.A. Underwood and C.L. Coates. Visual learning from multiple views. *IEEE Trans. Computing*, 24(6):651–661, 1975.

[106] N.S. Walker. *Biomedical Image Interpretation.* PhD thesis, Queen Mary and Westfield College, 1993.

[107] T.P. Wallace, O.R. Mitchell, and K. Fukunaga. Three-dimensional shape analysis using local shape descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 3(3):310–322, 1981.

[108] T.P. Wallace and P. Wintz. An efficient, three-dimensional aircraft recognition algorithm using normalized fourier descriptors. *Computer Graphics and Image Processing*, 3:99–126, 1980.

[109] H. Wechsler and G.L. Zimmerman. 2-d invariant object recognition using distributed associative memory. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(6):811–821, 1988.

[110] H.J. Wolfson and Y. Lambdan. Transformation invariant indexing. In J.L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 120–131. MIT Press, Cambridge, Mass., 1992.

[111] C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. on Computers*, 21(3):269–281, 1972.

[112] A. Zisserman, D.A. Forsyth, J.L. Mundy, and C.A. Rothwell. Recognising general curved objects efficiently. In J.L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 228–251. MIT Press, Cambridge, Mass., 1992.

[113] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation*, 3(4):461–483, 1991.