Pairwise Geometric Histograms for Object Recognition: Developments and Analysis

by

Anthony P. Ashbrook

RERVM COGNOSCERE CAVSAS

Ph.D. Thesis

Department of Electronic & Electrical Engineering
The University of Sheffield

Submitted in January, 1998

Summary

One of the fundamental problems in the field of computer vision is the task of classifying objects, which are present in an image or sequence of images, based on their appearance. This task is commonly referred to as the *object recognition* problem. A system designed to perform this task must be able to learn visual cues such as shape, colour and texture from examples of objects presented to it. These cues are then later used to identify examples of the known objects in previously unseen scenes. The work presented in this thesis is based on a statistical representation of shape known as a pairwise geometric histogram which has been demonstrated by other researchers in 2-dimensional object recognition tasks. An analysis of the performance of recognition based on this representation has been conducted and a number of contributions to the original recognition algorithm have been made. An important property of an object recognition system is its scalability. This is the ability of the system to continue performing as the number of known objects is increased. The analysis of the recognition algorithm presented here considers this issue by relating the classification error to the number of stored model objects. An estimate is also made of the number of objects which can be represented uniquely using geometric histograms. One of the main criticisms of the original recognition algorithm based on geometric histograms was the inability to recognise objects at different scales. An algorithm is presented here that is able to recognise objects over a range of scale using the geometric histogram representation. Finally, a novel pairwise geometric histogram representation for arbitrary surfaces has been proposed. This inherits many of the advantages of the 2-dimensional shape descriptor but enables recognition of 3-dimensional object from arbitrary viewpoints.

Related Publications

"Pairwise Geometric Histograms: A Scalable Solution for the Recognition of 2D, Rigid Shape", *Proc. SCIA95, pp 271-278, Uppsala, Sweden, 1995.*

"Multiple Shape Recognition using Pairwise Geometric Histogram based Algorithms", *Proc. IEE IPA95, pp 90-94, Edinburgh, UK, 1995.*

"Robust Recognition of Scaled Shapes using Pairwise Geometric Histograms", *Proc. BMVC95, pp 503-512, Birmingham, UK, 1995.*

"Finding Surface Correspondence for Object Recognition and Registration using Pairwise Geometric Histograms", *Proc. ECCV98, pp 674-686, Freiburg, Germany, 1998.*

Acknowledgements

One of the skills needed to complete a PhD is the ability to stay focussed on the relatively narrow field of study that final forms your thesis. During the study you inevitably come across new exciting ideas and it is hard not to wander, tangentially, to your original path. With this in mind, my supervisor Peter Rockett, provided the advice and gentle pushing needed to steer me past these crossroads to the final goal. I owe him many thanks for carrying out this unenviable task.

The work covered in this thesis is a continuation of the work of Alun Evans who was supervised by John Mayhew and Neil Thacker in the Artificial Intelligence Vision Research Unit in Sheffield. Fortunately, Neil Thacker had recently established himself in the Electronic and Electrical Engineering Department when I began my study. It was not long before Neil had convinced me of the merits of continuing with Alun's original work and I am very grateful for the necessary technical support and inspiration he provided.

The period of study for my thesis was carried out as a member of the Electronic Systems Group which hosted a large number of PhD students, contract researchers and academic staff. This variety of people created a stimulating working environment and provided plenty of opportunity for socialising. I would like to thank everyone in the Electronic Systems Group who helped to keep my daily life interesting and enjoyable.

I have very early memories of being fascinated by science and technology and have often wondered whether this inclination is due to nature or nurture. Either way, my family and friends have provided the encouragement and opportunities for me to take this inclination and develop it into a career that I thoroughly enjoy. In particular I would like to thank my parents, Noreen and Arnold, who have given a great start in life to myself and my brother and sister.

I believe that good philosophy in life is to avoid taking things *too* seriously. That is not to say that some things are not important, but this philosophy helps counter a natural tendency in people to become too caught up on minor issues. I would like to think that I have led my life by this philosophy but the reality is that I have needed to be reminded of this plenty of times. I would like to thank my partner, Caroline, whose good humour, personality and love for life is a constant reminder of what is really important.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Techniques for the automatic interpretation of image data are providing solutions to a very diverse range of problems. Production lines can be monitored automatically in industrial environments allowing manufactured items to be inspected for faults and providing visual feedback for automatic assembly. Medical images used in the diagnosis of disease and injury can be analysed, providing a valuable tool for doctors and surgeons. Image data collected by satellite can be analysed to provide data on crop growth and measure the the extent of pollution. With the growth of multimedia there is a need to organise large image and video archives efficiently and provide mechanisms for searching them. Image interpretation techniques are beginning to be used to achieve this. These are just a few examples of the potentially endless number of applications and as the sophistication of the technology develops it will play a more significant role in all aspects of our lives.

One of the fundamental problems of image interpretation is the automatic classification of objects which are present in image data. This is a process commonly referred to as *object recognition*. An object recognition system must learn visual cues, such as shape, colour and texture, from examples of objects presented to it. These visual cues are later used to identify examples of these objects in previously unseen images. A general architecture which most object recognition systems conform to is presented in Figure 1.1. Prior to recognition there is a training period in which visual cues are extracted from the example objects the system is expected to learn. These are recorded as an object description or representation. Recognition is then achieved by comparing model representations and representations constructed from the image of a scene in an appropriate manner.

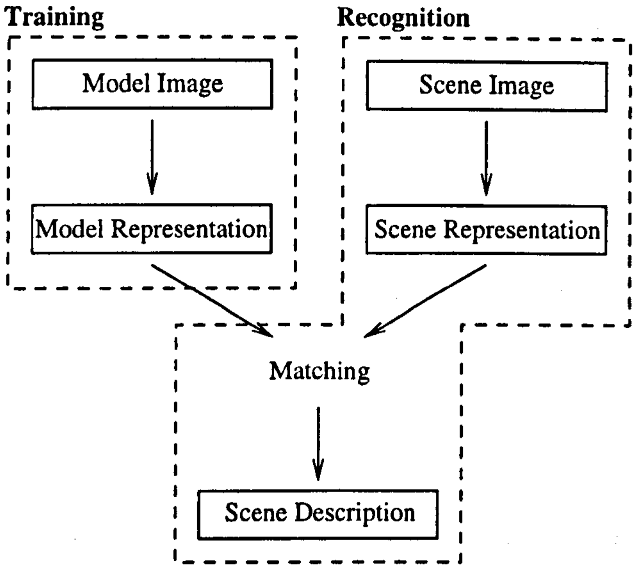


Figure 1.1: A typical architecture for object recognition systems

There are a variety of visual cues which can be used in the description of an object. It is likely that a general purpose vision engine will comprise specialised modules for tasks such as texture and shape based recognition with final classification of scene objects based on fusion of these different modalities. Although cues such as texture and colour are useful (if we want to distinguish between a red brick and a blue brick then colour is the only information needed) the work presented in this thesis concerns the recognition of objects using shape only. Shape is an attractive property of an object to use because it is highly descriptive and is easily quantified using the language of geometry. Central to the work presented in this thesis is the issue of system integration. An object recognition algorithm serves no purpose on its own and must be integrated into a larger system to provide any useful function. As just stated, it is also desirable to be able to integrate the data from different vision modules (and maybe other modes of sensory input) to take advantage of all of the information available about a scene in order to produce a system which is reliable and able to operate in a wide variety of environments. To enable this sort of integration the *quality* of the data produced by each vision module must be known so that the system can weight its significance in decision making.

Although the problem of object recognition is relatively simple in principle a straightforward solution is hindered because the appearance of an object can vary significantly from one image to the next. The key to developing a successful object recognition system is to develop object descriptions which are insensitive to this variation in appearance whilst still providing enough information to distinguish different objects reliably and efficiently. The appearance of an object in an image can vary for a variety of reasons and it is important to be aware of these when assessing the usefulness of any particular object representation. The most significant factors which affect appearance can be roughly grouped into four different categories.

- Lighting effects
- Change in the position of an object relative to the camera
- Image acquisition problems
- Object damage and deformation

The way an object appears in an image is directly related to the structure of the object, the way that the object's surfaces interact with incident light and the lighting conditions. If the relative position, intensity and colour of light sources is varied the appearance of the object can vary significantly. To minimise the effect this variation has on object descriptions, shape features such as edges and corners, which can be detected under a whole range of lighting conditions, are often used in the construction of representations.

As an object moves relative to the camera, or if the camera moves relative to the object, the image of the object changes. If the object is always viewed from the same direction then this change can be described as translations, rotations and scaling of the image data (ignoring the effects of perspective foreshortening). Object representations which are insensitive to these changes in the image data are said to possess translation, rotation and scale *invariance* respectively. If the direction in which an object is viewed changes, the change in the appearance of the object is more complex and depends upon the 3-dimensional structure of the object and its distance from the camera.

The process of acquiring and storing images digitally is subject to various imperfections which affect the final quality of the image data. Light reflected from the surface of an object is first focussed onto a sensor to produce an image using one or more lenses. Lenses can never be manufactured perfectly and as a result introduce some optical distortion. The function of the light sensor is to convert the level of incident light into electrical signals. As in all electronic systems, these signals are subject to a variety of sources of noise including thermal noise and cross-talk from other electrical signals and this will effect the final image data. Finally the electrical signals are sampled and quantised so that they can be represented digitally. The spatial sampling of the image data into an array cannot capture high spatial frequencies (a particular problem at shape edges) and can introduce artifacts because of aliasing. Further information is lost by the process of quantising the data to a finite number of intensity levels.

The actual shape of an object may change over time for a number of reasons. It may be composed of a number of moving parts which affect the shape of the object as they move or the object may be deformable and not have a well defined shape at all. Physical objects are also subject to wear and damage which affect their shape.

The task of recognition is complicated further because objects in real scenes rarely appear is isolation. Usually there will be other objects present and the background will contain lots of detail. This extraneous visual data is usually referred to as scene clutter. To be able to classify objects reliably in real scenes it is necessary to either isolate (or *segment*) each object from the scene, which is a substantial problem, or to adopt a matching strategy which is insensitive to this extra information. A more serious problem arises when objects obscure each other so that they are only partially visible (the problem of *occlusion*). If an object is described using some *global* property of its shape then recognition will be unreliable under conditions of occlusion.

1.2 Aims and Objectives

The motivation for the work presented in this thesis is this development and analysis of object recognition using *pairwise geometric histograms.* This novel representation, which was first proposed by Evans [Evans *et al* 93, Evans 94], enables efficient and reliable classification of 2-dimensional rigid shape data, solving many of the problems outlined earlier. In brief, this representation is formed by recording the geometrical relationship between pairs of shape primitives (Evans uses line segments) in the form of a frequency distribution, known as a *pairwise geometric histogram.* By careful selection of the geometric measures which are recorded, a concise shape descriptor with good invariance properties and insensitivity to noise can be obtained. A full description of this representation will be presented in Chapter 3.

This shape representation and its accompanying matching algorithms provides a strong basis for a general purpose recognition system based on shape. The research conducted here aims to provide some important improvements to the original algorithm, allowing it to be applied to a wider range of problems, and to provide an analysis of its performance to enable designers to integrate it more easily into a system. It is also demonstrated how the representation can be extended for 3-dimensional surface features to enable 3-dimensional surface based recognition.

One of the most important properties of any classification system is its reliability but all too often the designers of object recognition algorithms fail to provided a sufficient analysis of this, only demonstrating the technique on a few selected test images. On their own, object recognition algorithms serve no purpose and must be integrated into a larger system. To do this in a way which will provide robust performance, a quantitative measure of its reliability must be known. In fact a quantitative measure of reliability has many other uses which will be discussed later. This has motivated an analysis of the reliability of object recognition using pairwise geometric histograms.

Another important property of a classification algorithm, and again an issue which is frequently neglected, is *scalability* - the ability of a classification algorithm to perform successfully as the number of object classes is increased. Algorithms which become impractical to use for more than a few different object classes cannot provide a generic object recognition solution. When evaluating the scalability of a particular algorithm three factors must be considered.

- The relationship between reliability and the number of object classes
- The relationship between the computation needed to perform recognition and the number of object classes
- The number of different objects which can be represented uniquely (capacity)

To assess the scalability of the geometric histogram algorithms these issues will be addressed. The relationship between reliability and the number of object classes will become evident from the analysis of reliability, which has already been identified as an important issue in its own right. The relationship between the number of object classes and the computation needed for matching is linear, which is good for scalability, and the algorithm also has the advantage that it can be implemented on an array of simple, homogeneous processors to improve performance. This issue of computational complexity is relatively straightforward will not be considered any further. This leaves an interesting question regarding the number of objects which can be represented uniquely. To answer this a number of approaches for estimating the capacity of the pairwise geometric histogram representation have been investigated.

In order to accumulate evidence for the presence of a model in a scene and to determine the models position a generalised Hough transform was adopted as part of the original recognition algorithm. The Hough transform is noted for its robustness and performs well in this application but significant improvements can be gained by taking proper account of variability in the position of shape primitives. This has motivated the development of a probabilistic Hough transform which determines the most likely position of models in a scene and provides an estimate of the error on this estimate.

The geometric histogram representation used by Evans is invariant to translations and rotations of shape data but not to changes in shape scale. This lack of scale invariance prevents the recognition of objects at arbitrary distances from the camera (which then appear at arbitrary scale). To extend the range of possible applications of this algorithm this issue has been addressed.

Although the majority of the work presented in this thesis concerns the representation and classification of 2-dimensional shape, the pairwise geometric histogram representation can be extended to represent 3-dimensional features. Such an extension for the representation of 3-dimensional surfaces has been developed, enabling surface based recognition of 3-dimensional objects.

1.3 Organisation of the Thesis

In the next chapter a selection of the most important algorithms for 2-dimensional shape recognition are described and compared. It is intended that this selection will cover all of the important principles used to date and that other algorithms can been seen as extensions or variations on these. The chapter concludes with a summary of the important properties of each algorithm, highlighting the principles which are most useful for recognition.

The representation and recognition of shape using pairwise geometric histograms is reviewed at the beginning of Chapter 3 and results are presented to demonstrate the effectiveness of this shape descriptor for classifying shape primitives. This is followed by a formulation of the probabilistic Hough transform which is used to find arrangements of shape primitives in scene data which are consistent with model objects, providing a number of advantages over the original scheme. Results of using the probabilistic Hough transform are presented.

In Chapter 4 an analysis of the reliability of shape classification using geometric histograms is developed. Although techniques for error estimation in classification problems are common, the aim of this work is to make the relationship between reliability and the number of stored models explicit. To achieve this a novel estimation scheme is proposed and this is used to estimate the probability that shape primitives are misclassified for two different classes of shape data. The original recognition scheme is found to become unreliable as the number of models becomes very large so a modification to the algorithm is recommended which ensures reliable recognition for any number of stored models.

The issue of representation capacity is investigated in Chapter 5. To determine the number of unique classes that can be represented it is necessary to partition the continuous class domain into enumerable units and to determine the proportion of the domain which is likely to be occupied after training. Two different approaches are taken to solve these problems, one based on geometrical intuition and the other on a statistical model, and these are used to estimate the capacity for different classes of shape data. The geometric approach provides a bound on capacity and the statistical approach provides an estimate of the capacity itself.

The representation of shape data over ranges of scale is considered in Chapter 6. This addresses one of the main criticisms of the original pairwise geometric histogram approach. Experimental results are presented to demonstrate successful recognition of shape over ranges of scale and the algorithm is also used to track an object over a sequence of images as it approaches a fixed camera.

Chapter 7 presents a novel representation for 3-dimensional surface data which is based on a new pairwise geometric histogram descriptor. This 3-dimensional representation inherits many of the advantages of the 2-dimensional representation but enables full, 3-dimensional object recognition. Experimental results are presented to demonstrate the effectiveness of this novel scheme.

In chapter 8 the main contributions provided by the thesis are reviewed and the conclusions of each piece of work discussed. A number of suggestions for continued research are also presented.

Chapter 2

Representation and Classification of 2-Dimensional Objects

2.1 Introduction

Since the development of digital computers capable of storing and manipulating images, a large number of researchers from a surprisingly diverse range of fields of expertise have worked on the object recognition problem (the problem of identifying objects known *a priori* in previously unseen images). This has produced an equally diverse range of potential solutions with differing merits and ranges of applicability. The purpose of this chapter is to describe the most important solutions that have been published and ultimately to identify which of the underlying principles are of general importance, with a view to building on these.

Broadly speaking, the object recognition algorithms reported in the literature fall into either of two categories - those which require scene images to be divided into *regions of interest* prior to classification (a process commonly referred to as image segmentation) and those which focus on image features such as corners, edges, holes, arcs etc. In fact, the distinction between regions of interest and features is not that precise but typically a region of interest will be defined by some global characteristic and may contain many features which themselves are defined locally. Representations based on regions of interest may be further divided into those which use the region's shape or topology, those which use the shape of the region's contour and those which use information contained within the region itself. This categorisation of representations is depicted in Figure 2.1.

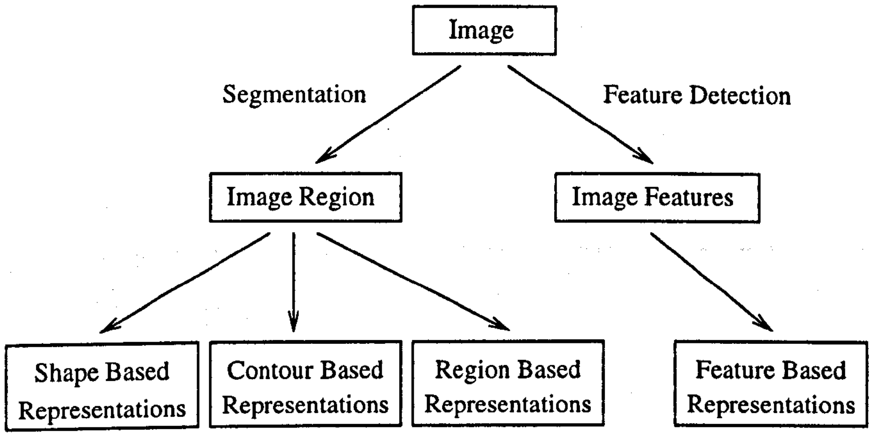


Image
Segmentation
Feature Detection
Image Region
Image Features
Shape Based Representations
Contour Based Representations
Region Based Representations
Feature Based Representations

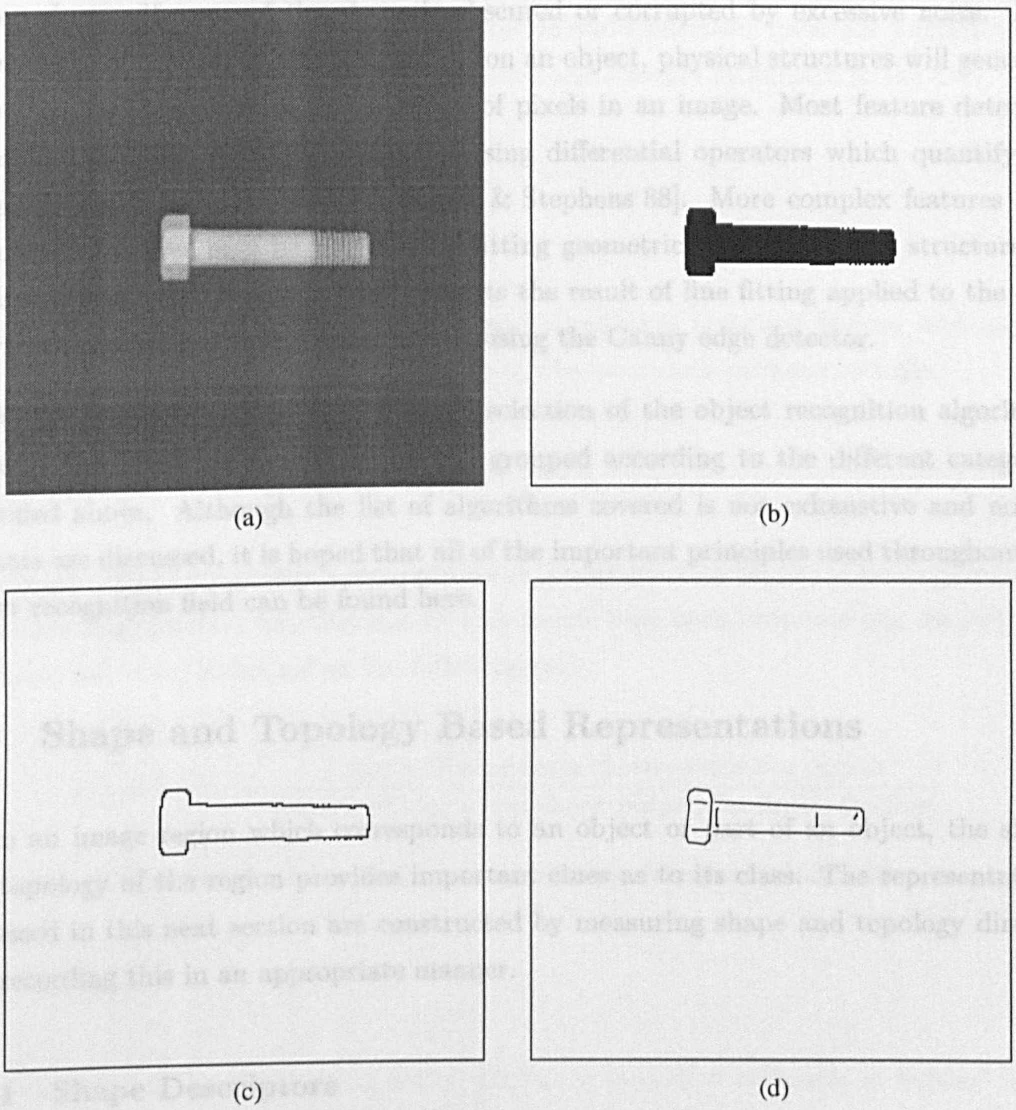
Figure 2.1: The different categories of object representations based on segmented image regions and detected image features.

2.1.1 Image Segmentation

One of the fundamental problems facing the designer of an object recognition algorithm is distinguishing between data which is relevant to the classification of an object (for example the pixels belonging to an object under investigation) and data which is not relevant (pixels belonging to the background or other objects). To some extent this problem can be avoided by designing classifiers which are insensitive to irrelevant scene data (scene clutter) but a more direct approach is to search for regions of a scene image which are believed to contain single objects or object parts and then to classify these regions in isolation.

Techniques for segmenting images into regions of interest are usually based on the premise that objects, or object parts, exhibit global characteristics which can be identified in images. A simple example might be the segmentation of machine parts appearing against a light background. The global characteristic common to these objects is that they are darker than the background and as such pixels can easily be classified as *object* or *background* by considering only their intensity. Figure 2.2 (a) contains a simple shape on a uniform background which is easily segmented by thresholding, as shown in Figure 2.2 (b). The region can also be segmented by detecting its bounding contour as shown in Figure 2.2 (c).

In more realistic scenes which may contain multiple objects possibly occluding each other, background clutter and other artifacts such as shadows, the segmentation process is much more demanding. Researchers have used additional information such as texture, colour, depth and motion to varying degrees of success but no generic solution has yet been found. Robust segmentation of objects from real scenes has proved to be somewhat paradoxical in that the only sure way to segment out a complete object is to have first recognised it.


(a)

(b)

(c)

(d)

Figure 2.2: Examples of region, boundary and line feature segmentation. (a) The original image. (b) The image is segmented by simple grey-level thresholding. (c) The image is segmented by a bounded contour. (d) Line segment features.

2.1.2 Feature Detection

Features such as edges and corners provide a valuable source of information about the content of an image as they correspond directly with physical structures in the real world. Features represent a local property of a shape's geometry and can usually be recovered even if some of the shape is obscured or corrupted by excessive noise. As a consequence of the way in which light falls on an object, physical structures will generally appear as discontinuities in the intensity of pixels in an image. Most feature detection techniques identify these discontinuities using differential operators which quantify image gradients [Sobel 70, Canny 86, Harris & Stephens 88]. More complex features such as holes and curves may be detected by fitting geometric models of these structures to the image gradients. Figure 2.2 (d) presents the result of line fitting applied to the edge information recovered from Figure 2.2 (a) using the Canny edge detector.

In the remainder of this chapter a broad selection of the object recognition algorithms found in the literature are described and grouped according to the different categories identified above. Although the list of algorithms covered is not exhaustive and not all variants are discussed, it is hoped that all of the important principles used throughout the object recognition field can be found here.

2.2 Shape and Topology Based Representations

Given an image region which corresponds to an object or part of an object, the shape and topology of the region provides important clues as to its class. The representations discussed in this next section are constructed by measuring shape and topology directly and recording this in an appropriate manner.

2.2.1 Shape Descriptors

The shape of arbitrary image regions can be described by a range of features derived from distance and area measurements taken directly from the image data. When grouped to form *feature vectors* they can be used to classify image regions using standard pattern recognition techniques. Although these features do not provide a *complete* description of the region shape (in the sense that the image region cannot be reconstructed from the description) an efficient and reliable classifier can be implemented for simple recognition tasks by careful selection of the appropriate features [Strachan *et al* 90]. A number of the most common features used in region classification are described here along with their invariance properties.

Distance

Simple distance measures, such as the lengths of the maximum and minimum chords across a shape or the shape's perimeter, provide rotation and translation invariant features. More usually, ratios of distances will be used as these also provide invariance to the scale of the shape.

Area

The area of a shape provides a rotation and translation invariant feature and may be normalised using some recoverable distance to also provide invariance to scale.

Eccentricity

The eccentricity of a region is a measure of the tendency of the region to be long and thin. A number of alternative formulations for this metric have been proposed but the simplest, and most intuitive, is defined as the following ratio:

$$eccentricity = \frac{\text{Length of maximum chord across the region}}{\text{Length of maximum chord perpendicular to the first}} \tag{2.1}$$

This ratio of distances provides invariance to rotation, translation and scale.

Rectangularity

The similarity of the shape of an image region to that of a rectangle is defined by its *rectangularity*. This is simply the ratio of the area of the image region to the area of the smallest rectangle which bounds the image region:

$$rectangularity = \frac{\text{Area of image region}}{\text{Area of smallest bounding rectangle}} \tag{2.2}$$

By taking the ratio of areas, the rectangularity is invariant to rotation, translation and scale.

Compactness

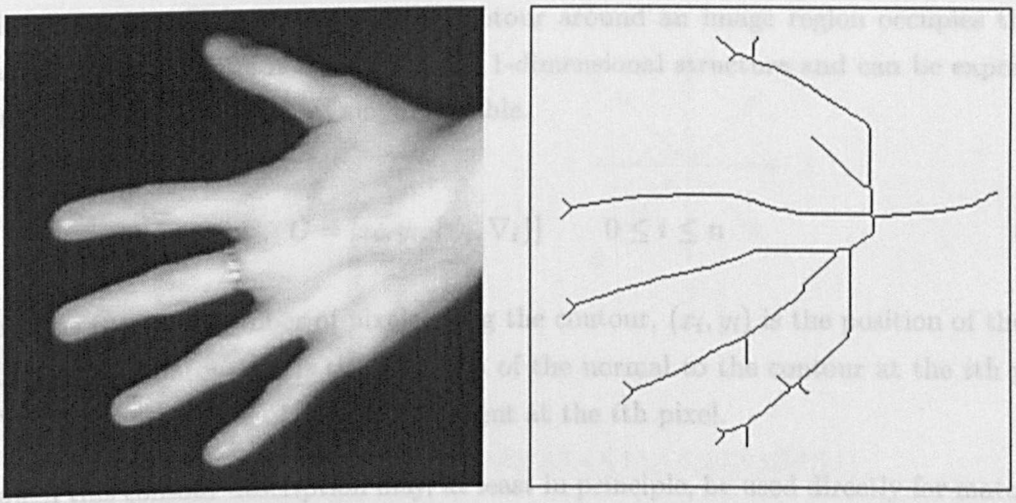
The tendency of an image region to be confined is measured by its compactness. Circular regions are the most compact and the least compact class of shapes are fractal structures. A common formulation of compactness is defined as:

$$compactness = \frac{(\text{Perimeter length of image region})^2}{\text{Area of image region}} \tag{2.3}$$

The ratio of a squared distance to an area provides scale invariance in addition to invariance to rotation and translation.

2.2.2 Shape Skeletons

The topology of a shape can provide important clues about its class. A shape's topology can be recovered by *thinning* until it becomes a unit pixel width network and then describing the resulting *skeleton* as a connected graph. A shape and its skeleton are presented in Figure 2.3. Recognition is achieved by comparing topological graphs constructed from models with graphs constructed from scenes using well established graph matching techniques.



Original Image

Extracted Skeleton

Figure 2.3: The topology of the hand can be recovered by first thresholding the image and then stripping away pixels until a thin skeleton remains.

Thinning strategies generally work on the principle of stripping away successive layers of shape boundary points on the condition that the removal of a point does not change the connectedness of the shape. This can be done using either erosion morphological operators or *fire-front* type algorithms (a good example is presented by Xia [Xia 89]).

Topological networks posses all of the invariant properties required of a general vision system although for most applications a purely topological description is too ambiguous and some structural constraints have to be added. These structural constraints may well compromise the invariance properties of the description.

The topology of a shape is largely unaffected by random noise although its presence can introduce short spurs into the shape skeleton. These are easily removed by pruning. By their nature, thinning algorithms are sensitive to occlusion (to a thinning algorithm an occluded shape looks like a different shape with different topology) and in general recognition schemes based on skeletons cannot cope with occluded schemes.

2.3 Contour Based Representations

In applications where objects, or object parts, can be successfully segmented from a scene image the *shape* of the contour around the object can be used for classification. When extracted from a region of interest, the raw contour C comprises a string of edge pixels (*edgels*) defined by their position in the image and possibly their orientation and/or the local edge gradient. Although the contour around an image region occupies the 2-dimensional image plane it is essentially a 1-dimensional structure and can be expressed parametrically as a function of single variable.

$$C = [x_i, y_i, \{\psi_i, \nabla_i\}] \qquad 0 \leq i \leq n \tag{2.4}$$

Where n is the total number of pixels along the contour, (x_i, y_i) is the position of the ith pixel along the contour, ψ_i is the direction of the normal to the contour at the ith pixel and ∇_i is the magnitude of the edge gradient at the ith pixel.

Although this contour description may, at least in principle, be used directly for matching unseen contours with known model contours it lacks the necessary properties to promote efficient and reliable recognition. A good contour description for efficient recognition is one which is both compact and invariant to the position, orientation and scale of the contour but the raw contour description has none of these properties. For reliable recognition the contour description must be repeatable and insensitive to the types of noise encountered in typical scene images but again this is not a property of C. To address these problems a number of alternative contour descriptors have been proposed.

2.3.1 Chain Codes

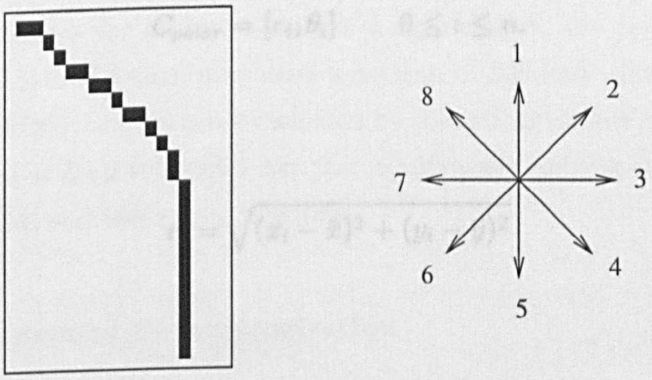
The chain code, C_{chain}, is a contour descriptor which has been developed to remove the dependency of the descriptor on the position of the contour within an image [Freeman 61]. Instead of defining a contour pixel by its position within the image it is defined by its position relative to the previous edge pixel in the contour. Because an edgel can be in any one of eight[1] possible pixel locations relative to its previous neighbour each edgel may be uniquely defined by an integer between 1 and 8, and the complete contour described by a string (or chain) of integers in this range.

$$C_{chain} = [c_i] \qquad c_i \in 1,2,3,4,5,6,7,8 \quad \text{and} \quad 1 \leq i \leq n-1 \tag{2.5}$$

Each chain element, c, is derived from the position of the ith and $(i-1)$th pixel.

$$c(i) = F\left(x_i - x_{i-1}, y_i - y_{i-1}\right) \tag{2.6}$$

Where $F(\Delta x, \Delta y)$ maps the position of the current pixel relative to its neighbour to an integer in the range 1 to 8. An example is provided in Figure 2.4 for a section of contour.



C_{chain} = [3,3,4,4,4,3,4,3,4,4,3,4,4,4,5,5,4,5,5,5,5,5,5,5,5]

Figure 2.4: A section of a shape contour and its associated chain code.

By describing contours in this manner boundaries found in unseen images can be classified by comparing them to model boundaries using string matching techniques. This approach has the advantage that broken or occluded contours can still be classified by treating their chain codes as sub-strings of those describing the stored model contours. This relative

[1] Assuming 8-connectivity is used when forming the contour

description of contour pixels is invariant to translations of the contour within the image. However, the descriptor is still dependent on the scale and orientation of the shape and is therefore of limited general applicability. A useful extension of this coding which removes the dependency on the orientation of the contour is to record the first derivative (modulo-8) of the chain code so that each chain element represents the *change* in direction of the contour rather than its absolute direction. A more serious problem with this type of descriptor is its sensitivity to even moderate amounts of scene noise. This arises because of the discontinuous mapping, $F(\Delta x, \Delta y)$, between pixel positions and chain codes with the consequence that small variations in pixel positions due to noise can produce abrupt changes in the descriptor.

2.3.2 The Polar Parameterisation

A useful shape descriptor is obtained by transforming the position of contour pixels from Cartesian coordinates to a polar coordinate system whose origin lies on the contours centroid $(\bar{x}, \bar{y})$. Each pixel along the contour C_{polar} is defined by its radial distance from the centroid, r, and its angular displacement around the contour from some arbitrary reference, θ.

$$C_{polar} = [r_i, \theta_i] \qquad 0 \leq i \leq n. \tag{2.7}$$

Where

$$r_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \tag{2.8}$$

and

$$\theta_i = \arctan\left(\frac{y_i - \bar{y}}{x_i - \bar{x}}\right) \tag{2.9}$$

By interpolating between adjacent boundary pixels the contour can be described as a continuous function of r in θ as depicted in Figure 2.5.

$$C_{polar} = r(\theta) \tag{2.10}$$

One advantage of this type of contour description is that it is independent of the position of the contour within an image and, importantly, rotations of the the contour produce a

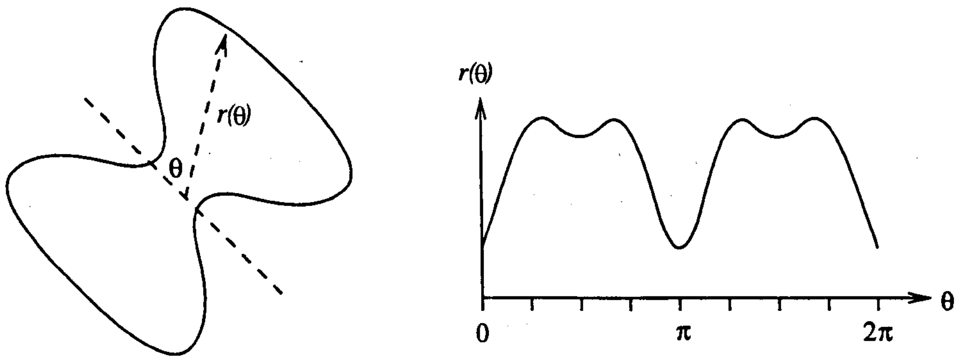


Figure 2.5: Describing a shape contour in polar coordinates.

shift in the descriptor along the θ-axis. This promotes efficient matching between scene and model contours using a 1-dimensional shift correlation. A further advantage of this representation is that small changes in the position of contour pixels due to scene noise only result in small changes in the descriptor enabling reliable recognition in moderately noisy scenes.

Like all representations which rely on centroid measurements the (r, θ) parameterisation becomes distorted if the position of the centroid changes. This seriously impairs shape recognition in scene images exhibiting occlusions or significant background clutter where repeatable recovery of a shape's centroid is not possible. A particular problem with the (r, θ) description is that for all but the most simple shapes, the profile becomes multi-valued (for some values of θ their may exist a number of different values of r). It has been suggested that multiple r values can be avoided by discarding all but either the smallest or the largest r values at each value of θ but this significantly reduces the descriptive power of the representation and leads to ambiguities.

2.3.3 The Tangential Parameterisation

The tangential representation is an alternative contour parameterisation which is position invariant without the need to recover the contour's centroid and which cannot become multi-valued so that arbitrary contours can be represented completely and without ambiguity. Starting at any arbitrary point along the contour the orientation of each boundary edgel, relative to the orientation of the first edgel, ψ, and the distance travelled along the contour, s are recorded. This is a measure of the degree of *bending* of the contour at a distance s from the start. An example is given in Figure 2.6 for the simple case of a circular contour.

$$C_{tan} = [\psi_i, s_i] \qquad 0 \leq i \leq n \tag{2.11}$$

Like the polar representation the contour can be described as a continuous function by interpolating between adjacent pixels, as shown in Figure 2.6.

$$C_{tan} = \psi(s) \tag{2.12}$$

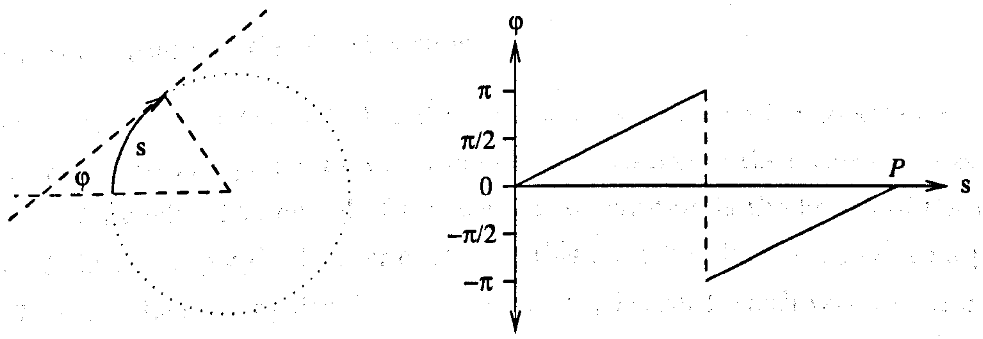


Figure 2.6: A two dimensional shape and its tangential representation. P is the perimeter of the boundary.

By measuring the orientation of contour pixels relative to the orientation of the first pixel, the dependency of the descriptor on the orientation of the contour is removed. However, variation in the selection of a starting point along the boundary produces a shift in the descriptor along the s-axis *and* along the ψ-axis. Matching can proceed as a 2-dimensional shift correlation but more usually, to improve efficiency, the mean value of ψ is removed from the descriptor and matching is performed as a 1-dimensional shift correlation along the s-axis. The drawback of this improvement is that the mean value of ψ is sensitive to occlusions. A significant problem with this contour representation is that estimates of the distance travelled along the contour tend to be over-estimated and variable depending upon the way the original image is quantised.

2.3.4 Fourier Descriptors

It is well understood that any periodic function can be expressed as Fourier series and Cosgriff [Cosgriff 60] recognised the merit in applying this to functions describing contours in images. Any contour parameterisation which is periodic (or can be made periodic) can be described by an infinite series of Fourier coefficients but usually only a small number of these coefficients are necessary for classifying the contour's shape, producing a concise contour description.

Although the polar parametrisation is a naturally periodic function, variants of the tangential function are more usually used as the basis for Fourier descriptors because of their ability to describe arbitrary contours without ambiguity. The tangential function itself is not periodic because the measure of bending, ψ, decreases[2] by 2π for each circuit of the closed contour, but this is easily rectified by defining a variant, $\psi^*(s)$.

$$\psi^*(s) = \psi(s) + \left(\frac{s}{L}\right) 2\pi \qquad (2.13)$$

Where L is the length of the closed contour.

It is interesting to note that $\psi^*(s) \equiv 0$ when the contour shape is perfectly circular so this variant can be interpreted as a measure of the deviation of the shape of the contour from that of a circle. The period of this contour description is the length of the closed contour, L, but to apply the Fourier expansion this must first be normalised to a period of 2π. This may be done by defining a new boundary length t which varies from 0 to 2π as s varies from 0 to L.

$$t = \left(\frac{s}{L}\right) 2\pi \qquad (2.14)$$

A periodic contour description with period 2π is obtained by substituting t into expression 2.13.

$$\psi^*(t) = \psi\left(\frac{Lt}{2\pi}\right) + t \qquad (2.15)$$

This periodic contour description can then be expressed as a Fourier series.

$$\psi^*(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k) \qquad (2.16)$$

The shape contour is now fully described by the Fourier coefficients A_k and α_k and the mean value μ_0. The details of how these coefficients are determined from expression 2.16 are not included here but can be found in any standard Fourier series text, although a good example specifically for shape contours is reported by Zahn *et al* [Zahn & Roskeis 72]. Like the tangential parameterisation discussed earlier the mean value μ_0 reflects the choice of a boundary starting point and is not useful as an invariant descriptor but the harmonic

[2] When the contour is followed in a clockwise direction

amplitudes, A_k, and some functions of the phase angles, α_k are invariant to translations, rotations and scalings of the shape contour.

An interesting property of Fourier descriptors is that the lower order Fourier coefficients describe the macroscopic behaviour of the contour's shape whilst higher order coefficients describe more detailed variations in the shape of the boundary. Truncating the coefficients used to describe a particular shape boundary not only results in a concise shape descriptor but results in a descriptor which is very insensitive to small variations in the boundary due to noise. Unfortunately more serious degradation in a contour's shape, perhaps as a result of object occlusions, dramatically changes the Fourier description both because the macroscopic behaviour of the contour changes significantly and because the contour length, L, used in determining the Fourier coefficients cannot be recovered.

2.4 Region Based Representations

The shape of the contour which bounds an image region is only one of the properties of the region which can be used for classification. Other characteristics such as texture and colour provide further information which may be used to construct more descriptive object representations. A number of representational schemes which make explicit use of the intensity of pixels within some image region, usually obtained by segmentation, are discussed in this section. Given an image region, R, this region may be described by the intensity and position of each of the pixels which it contains.

$$R = I(x,y) \qquad (x,y) \in R \tag{2.17}$$

Like the raw contour descriptor introduced earlier, R can be used for matching directly and this is the basis of the block matching scheme discussed next. However, it possesses none of the desired invariance properties and this has motivated the development of more sophisticated algorithms, some of which will be discussed subsequently. Although all of these algorithms can be used to represent raw intensity images, it is more usual to use edge-enhanced images as this provides a degree of insensitivity to lighting variations.

2.4.1 Template Matching

Given an image region describing a model object it is possible to detect instances of this model in a scene by placing it at every possible location and measuring the similarity between corresponding model and scene pixel intensities. Given a model region $M(i,j)$ and a scene image $S(x,y)$ this can be expressed formally as:

$$c(x,y) = \sum_{i=1}^{M}\sum_{j=1}^{N}[M(i,j) - S(x+i, y+j)]^2 \tag{2.18}$$

Where the model region is a block with a width of M pixels and a height of N pixels, and $c(x,y)$ is a measure of the similarity between the model and the scene when the model is placed at (x,y) ($c = 0$ implies a perfect match).

Using raw pixel intensities to represent objects allows arbitrary shape to be described and because pixels are compared individually matching can be performed reliably when an object might be partially obscured. The representation possesses no invariance properties so if objects are free to translate, rotate and scale an excessively large amount of computation is needed to detect them.

2.4.2 The Log-Polar Mapping

Using grey-level templates to represent image regions makes good use of the information provided but this type of descriptor does not promote efficient recognition because of its lack of invariance properties. Sensitivity to changes in lighting conditions can be reduced by using edge-enhanced images but this still leaves a representation which is sensitive to changes in the scale and orientation of image regions. The log-polar descriptor, $R_{logpolar}$, addresses this limitation by mapping pixels (edge-enhanced or otherwise) into a domain where changes in scale and orientation in the image space manifest themselves as translations in the log-polar space. This permits a relatively efficient 2-dimensional shift-correlation to be used for classification.

The log-polar representation is constructed by sub-pixel interpolation of the original image data using the mappings given by expression 2.20 and 2.21. Like the polar contour parameterisation, discussed earlier, the log-polar representation is only useful if the centroid (x', y') can be recovered.

$$R_{logpolar} = J(\rho, \theta) \quad (\rho, \theta) \in R_{logpolar} \tag{2.19}$$

Where ρ is the logarithm of the radial distance, r, of (x,y) from the region centroid, (x', y').

$$\rho = \log(r) \tag{2.20}$$

$$\theta = tan^{-1}\left(\frac{y-y'}{x-x'}\right) \tag{2.21}$$

Both Wechsler et al [Wechsler & Zimmerman 88] and Rak et al [Rak 91] take the Fourier Transform of the log-polar space and use its magnitude as an invariant measure. This works because the magnitude of the Fourier Transform is invariant to translation.

Wechsler et al identify the problem that small variations in located centroid result in dramatic variations in the resulting log-polar representation. Unfortunately, random noise and occlusion produce such variations.

The nature of the mapping is that many samples are taken at the centre of the image where the radial lines are closer together but the resolution falls off moving out from the centre. This has two consequences. Firstly the method is really only suitable for objects which are significantly smaller than the image size so that the resolution of the representation is sufficiently high. Secondly, because outlying objects have only a minor effect on the representation due to the low sampling, the object under analysis does not have to be segmented out from the image, providing that it can be centred correctly.

2.4.3 Moment Invariants

The use of moments as invariant binary shape representations was first proposed by Hu [Hu 62], who successfully used this technique to classify handwritten characters. The regular moment of a shape in an M by N binary image is defined as:

$$u_{pq} = \sum_{y}\sum_{x} x^p y^q I(x,y) \tag{2.22}$$

Where $I(x,y)$ is the intensity of the pixel at the coordinates (x,y) and $p+q$ is said to be the order of the moment.

To remove the dependency of high order moments on the position of a shape within an image, measurements are made in relation to the shapes centroid (x',y'). The coordinates of the centroid are determined using the first order moments.

$$x' = \frac{u_{10}}{u_{00}} \quad \text{and} \quad y' = \frac{u_{01}}{u_{00}} \tag{2.23}$$

Relative moments are then calculated using the equation for central moments which is defined as:

$$u_{pq} = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} (i - x')^p (j - y')^q I(i,j) \tag{2.24}$$

Individual moments values do not have the descriptive power to uniquely represent arbitrary shapes, nor do they posses the required invariance characteristics. However, sets of functions based on these moments can be determined which do have this power. Hu derived a set of seven rotational invariant moment functions which form a suitable shape representation (or feature vector).

$$M_1 = (u_{20} + u_{02}) \tag{2.25}$$

$$M_2 = (u_{20} - u_{02})^2 + 4u_{11}^2 \tag{2.26}$$

$$M_3 = (u_{30} - 3u_{12})^2 + (3u_{21} - u_{30})^2 \tag{2.27}$$

$$M_4 = (u_{30} + u_{12})^2 + (u_{21} + u_{03})^2 \tag{2.28}$$

$$\begin{aligned} M_5 &= (u_{30} - 3u_{12})(u_{30} + u_{12}) \left((u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2\right) \\ &+ (3u_{21} - u_{03})(u_{21} + u_{03}) \left(3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2\right) \end{aligned} \tag{2.29}$$

$$M_6 = (u_{20} - u_{02}) \left((u_{30} + u_{12})^2 - (u_{21} + u_{03})^2\right) + 4u_{11}(u_{30} + 3u_{12})(u_{21} + u_{03}) \tag{2.30}$$

$$\begin{aligned} M_7 &= (3u_{21} - u_{03})(u_{30} + u_{12}) \left((u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2\right) \\ &- (u_{30} - 3u_{12})(u_{21} + u_{03}) \left(3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2\right) \end{aligned} \tag{2.31}$$

Classification is achieved by matching a shape vector extracted from an image with previously encountered shape vectors from the training set. The shape representation can be improved to exhibit scale invariance by a process of normalisation.

Moment invariants do not inherently possess translational invariance and this variability is removed by centring the coordinate system on a shape's centroid. Unfortunately, moment calculations are sensitive to the position of a shape's centroid and attempts to determine this are marred by random noise, poor segmentation and occlusion. Hence, moment invariant schemes are not robust to these types of problem.

2.5 Shapes as Feature Groupings

The use of image features to represent objects promotes a shape descriptor which can be reliably recovered in noisy scenes, can represent arbitrary shape by carefully selection of appropriate features and provides a powerful, possibly complete, description of the shape. Object recognition strategies based on image features typically comprise two distinct stages. Initially features are recovered from the scene image data and then a search is conducted to find groups of features which are consistent with stored models. A number of techniques for finding consistent groupings of features have been investigated but the most important are discussed here.

2.5.1 Relational Graph Matching

Model features are well defined by their geometrical relationship to each other and the technique of relational graph matching exploits this to find consistent feature groupings. Given a set of model features, M, and a set of detected scene features, S, each possible pairing between a model feature and a scene feature is considered to be a possible match. These pairings can be represented as nodes in a, so far unconnected, graph as shown in Figure 2.7 (a). Each node of the graph then represents a mapping from the model to the scene.

To find consistent feature groupings each pair of graph nodes is considered in turn and the mappings they represent compared. If these mappings are the same, within some specified tolerance, an arc is formed between the pair of nodes. When this has been completed for all node pairs, consistency between groups of model and scene features manifest themselves as fully connected networks of graph nodes, referred to as cliques.

The invariance properties of this type of representation depend upon the geometrical relationships recorded between features but generally it is possible to attain translation, rotation and scale invariance. The computational complexity of this approach is a serious disadvantage and arises because of the large number of mappings which can exist between

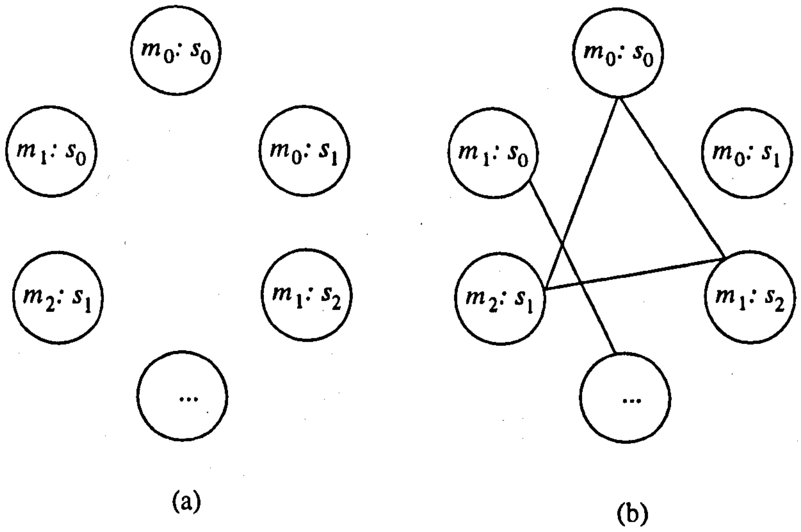


Figure 2.7: Relational graph matching. (a) A graph node is constructed for each model and scene feature pair. (b) Geometrically consistent pairs are labelled by a connecting arc. Consistent groups manifest themselves as fully connected groups or *cliques.*

scene and model features and the complexity of the *clique* detection techniques. This problem can be minimised by directing the search for features within some local area as in the Local Feature Focus method suggested by Bolles *et al* [Bolles & Cain 82].

2.5.2 The Interpretation Tree

Consistency between model and scene features can be found much more efficiently if the relational graph is reorganised as a search tree. The use of a search tree to search the relational graph, referred to as the *interpretation tree,* was first proposed in [Gaston & Lozano-Perez 84], although the sensor data used in this first example was tactile rather than visual. The approach has since been used for many visual recognition tasks using a range of different image features [Grimson & Lozano-Perez 87, Grimson 90].

The interpretation tree is constructed such that each possible pairing between scene and model features is represented by a unique path through the tree. A simple example is illustrated in Figure 2.8 for a set of model features $M = \{m_0, m_1, m_2\}$ and a set of scene features $S = \{s_0, s_1, s_2\}$. At each node of the tree a check is made to ensure that the model to scene pairings described by the path from the tree root to the node are self consistent. If not then the tree below the node in question is pruned away.

As the tree is descended, more constraints become available for consistency checking. At the first layer only a single model feature to scene feature pairing is specified so consistency

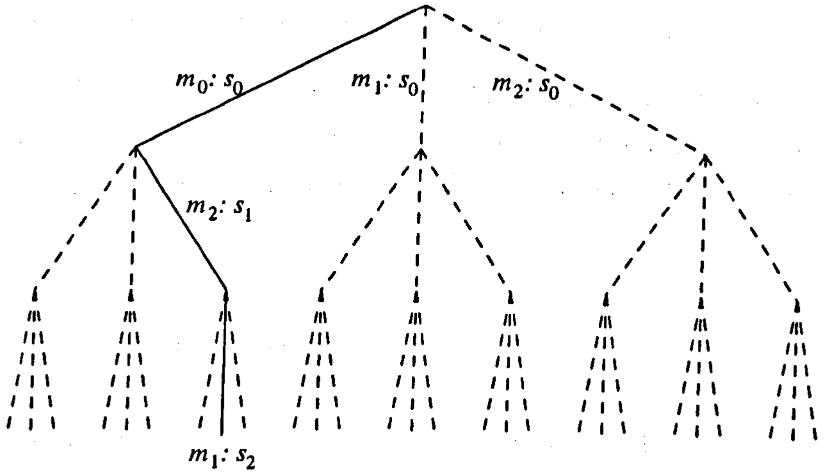


Figure 2.8: The interpretation tree provides an efficient scheme for finding consistent model-to-scene feature pairings.

can only be determined using properties of the features themselves, usually referred to as *unary* constraints. For example, when utilising line features the scene and model line lengths might be used to decide whether or not a particular pairing is consistent. At the second layer of the tree, however, relationships between pairs of model features can be used to provide stronger constraints for consistency checking. These are usually referred to as *binary* constraints. The final objective is to reach one of the leaf nodes of the interpretation tree, at which point a consistent set of feature pairings has been determined.

2.5.3 Geometric Hashing

Geometric Hashing was first proposed by Lamdan and Wolfson [Lamdan & Wolfson 88] as an alternative to the feature matching approaches already discussed. The approach differs in that no explicit matching between scene and model features is conducted, thus avoiding the computational explosion this produces. Instead, invariant measures are derived from scene features and these are used to index a precompiled *hash table* of models. Votes are then cast for the indexed models and recognition is achieved by finding models which have received a significant number.

The invariant measurements used to index the hash table are derived by transforming features into an invariant frame of reference, which itself is determined from a small number of the features. In the original work proposed by Lamdan and Wolfson, an arbitrary pair of point features are mapped to the coordinates $(0,0)$ and $(1,0)$. The same mapping is then applied to all of the other points and the resulting coordinates are used to index the hash table. This approach has also been applied to arbitrary views of 3-dimensional objects by using more features to define an affine invariant frame of reference.

Compilation of the hash table can be computationally expensive because of the number of combinations of features that can be selected to define the invariant frame. The compilation is performed off-line though and allows models to be indexed very quickly during recognition.

2.5.4 The Hough Transform

The Hough transform was first devised in 1962 as a means of detecting the paths of high energy particles [Hough 62] but has since evolved and been applied to many different image processing and computer vision tasks. In essence, the Hough transform maps complex patterns of pixels or features from the image domain into compact features in some selected parameter space. This greatly simplifies the task of searching for complex patterns in an image when working in the parameter space.

Probably the most straightforward application of the Hough transform is in the detection of straight lines in images (or strictly, the detection of collinear edge pixels). Although there are some earlier versions, the accepted formulation was first presented by Duda [Duda & Hart 72]. In this formulation, lines are described using the P-θ equation, as shown in Figure 2.9.

$$P = x\cos\theta + y\sin\theta \tag{2.32}$$

Where P is the length of the normal from the line being detected to the origin and θ is the angle between the normal and the positive x-axis.

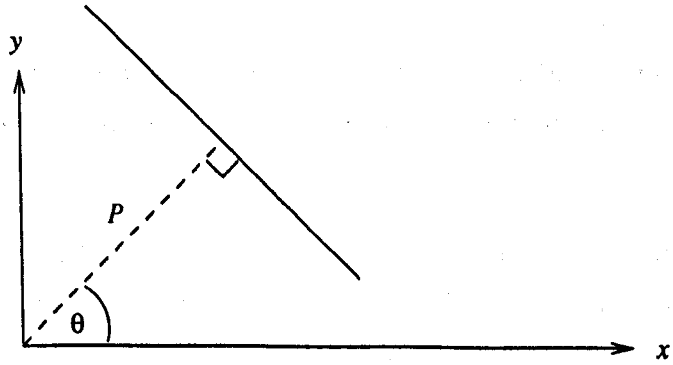


Figure 2.9: A line in image space defined by the two parameters P and θ.

Each edge pixel in an image can potentially lie along an infinite number of lines passing through it, and the parameter values associated with these lines describes a sinusoid in P-θ

space. If a sinusoid entry is accumulated into a quantised, 2-dimensional parameter space for each detected edge pixel, collinear pixels result in sinusoids which intersect at the same point producing a detectable peak. The position of the peak determines the parameters of the line the edge pixels lie along and the height of the peak gives the number of pixels which lie along that line.

A number of developments of this basic scheme have been devised for the recognition of parameterised shapes such as circles and ellipses but the most the significant improvement in the field of object recognition has been the development of the generalised Hough transform [Ballard 81]. This formulation of the Hough transform allows arbitrary objects to be recognised in scenes from their edge features and is robust to partial occlusion of the object data and the presence of scene clutter. Central to the technique is the parameterisation of an object's edge pixels in terms of some arbitrary reference point in the image plane. For each edge pixel, the vector $\mathbf{p}(r, \alpha)$ from this edge to the reference point is recorded in a table as a function of the orientation of the edge, ϕ. There may be a number of edge pixels with the same orientation so each row of the table, commonly referred to as the R-table, may contain multiple entries. Figure 2.10 shows the measurements made to construct the R-table.

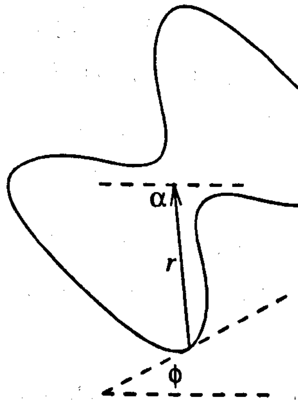


Figure 2.10: The generalised Hough transform. The shape boundary is parameterised in terms of the vector, $\mathbf{p}(r, \alpha)$, which defines an arbitrary reference point from each edge pixel.

Given a scene image, each detected edge pixel is used to index the R-table for each stored model to provide a hypothesis of the position of the model's reference point, and this hypothesis is accumulated in a 2-dimensional, quantised parameter space. Edge pixels consistent with a particular model generate consistent hypotheses producing a peak in the parameter space. The height of this peak relates to the number of edge pixels consistent with the model and the position of the peak provides an estimate of the position of the model in the image.

The generalised Hough transform possesses translational invariance but not scale or rotational invariance. To detect shapes at different orientations and scale, an explicit search must be made for each instance by applying a suitably transformed R-table. The table is transformed for scale variations by simply scaling the r values accordingly whilst it is transformed for orientation variations by shifting the r values along the ϕ-axis. The result is a four dimensional parameter space with two axes for image position, an axis for orientation and an axis for scale. Significant peaks in this four dimensional space then indicate the presence of an object along with its position, orientation and scale within a scene. The disadvantage of this approach is the amount of computation and storage needed to search and store the large parameter spaces although significant work has been conducted to avoid this problem. This is typically done by constructing coarse parameterisations and then focusing on dense areas of the parameter space [O'Rourke 81], or by decoupling the parameters and searching through the resulting lower dimensional spaces.

2.6 Affine and Projective Invariance

For the majority of 2-dimensional recognition problems it is reasonable to assume that objects are constrained to lie in a plane, but relaxing this constraint can open up additional applications. A good example is in an industrial environment in which objects are represented using their 2-dimensional appearance but may lie on top of one another so that they are not constrained to a fixed plane.

When the depth across an object is small compared to its distance from the camera the mapping from world coordinates to image coordinates can be approximated using an affine transformation. This is the assumption of *weak perspective*. A number of researchers have proposed affine invariant representations but when the assumption of weak perspective cannot be made a more general, projective transformation must be used. In the next section a projective invariant shape representation scheme is presented.

2.6.1 Projective Invariants

A shape representation for two dimensional planar objects which is invariant to projective transformations as well as changes in pose and scale is presented by Rothwell et al [Rothwell *et al* 92]. The representation relies upon the fact that points of tangency on a two dimensional planar object are preserved under different projections and also that the mapping of any four points from one plane to another is sufficient to determine the transformation matrix **T** which fully defines that transformation. Consequently, by mapping four points of tangency from a planar object to four fixed but arbitrary points in a second plane, this second plane will possess the required invariant properties, and by determining the transformation matrix **T** from the four mappings all points on the planar object can be mapped onto the invariant plane.

In this scheme, planar object concavities are used to determine four tangency points, referred to as distinguishing points, as shown in Figure 2.11. The first two distinguishing points (A and D) are located by the bitangent that marks the entrance to the concavity. The other two distinguishing points (B and C) are located by the tangents to the inner curve of the concavity which pass through each of the first two distinguishing points.

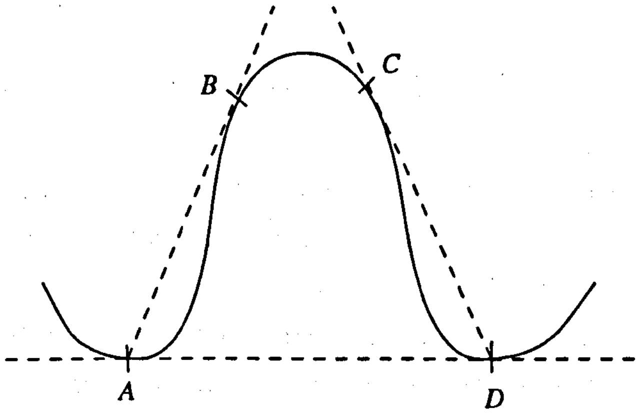


Figure 2.11: For this projective invariant, two distinguishing points (A and D) are located by the bitangent that marks the entrance to the concavity. Two other distinguishing points (B and C) are located by the tangents to the inner curve of the concavity which pass through each of the first two distinguishing points

These four points are then mapped to the corners of a unit square on the invariant plane, which is referred to as the canonical plane, and then the same transformation is used to map all other boundary points within the concavity onto this plane. The mapping of the concavity shown in Figure 2.11 to the canonical plane is shown in Figure 2.12 below.

Rothwell uses this mapping to construct an invariant object representation by projecting planar objects onto the invariant plane and then taking a number of area and moment measurements which then constitute invariant feature vectors. The use of local object concavities provides some robustness to partial object occlusions although the need for concavities limits the scope of objects that can be represented in this way. Although an extension of the approach to 3-dimensional objects would be desirable it would appear to be impossible as Burns [Burns *et al* 93] has proved that this type of projective invariant cannot exist for arbitrary 3-dimensional structures.

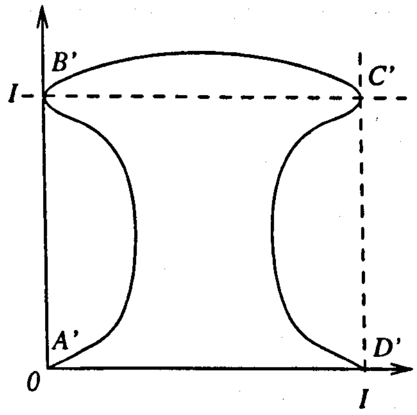


Figure 2.12: The mapping of the concavity in Figure 2.11 onto the canonical plane.

2.7 Representing Deformable Shape

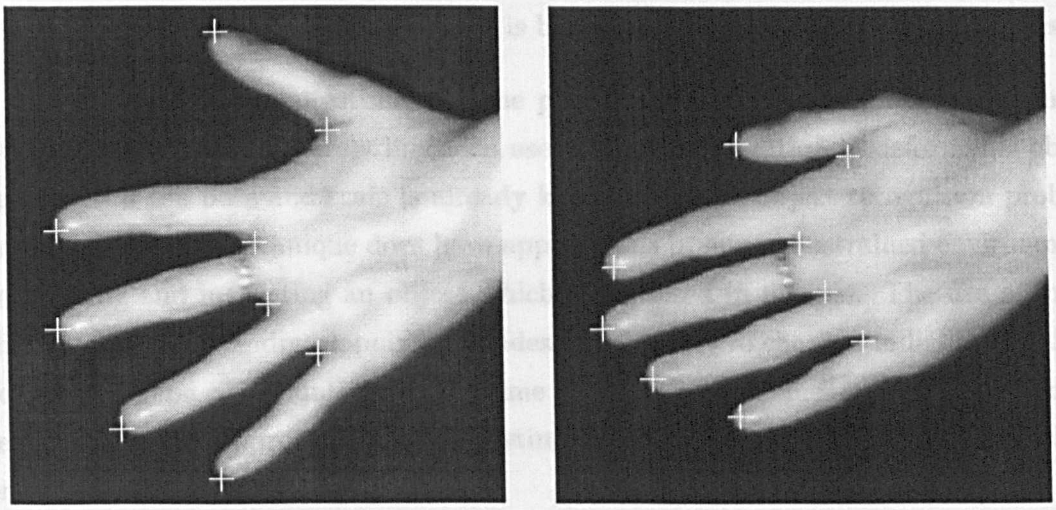
The object representations considered so far in this chapter have been limited to the description of rigid shape, and certainly this is an important class of problem. Recently though, interest in representations for deformable objects has grown significantly with key areas such as medical imaging and face recognition providing much of the motivation. Although the work presented in this thesis concerns the recognition of rigid shape, a summary of shape representations would be incomplete without a brief look at deformable shape.

2.7.1 Point Distribution Models

Cootes [Cootes & Taylor 92] has presented a deformable shape representation which models the way in which the position of landmark points located on an object vary as the object deforms. Given a set of examples of the object which exhibit the expected modes of deformation, landmark points are placed on each example (usually by hand) and the movement of these points, between examples, is measured. Figure 2.13 shows a pair of examples from a training set with labelled landmark points. To be able to find correspondences between landmarks in different examples, or later between the model and scene data, they must be placed on recoverable features such as edges or corners.

If the position of all of the landmark points (x_i, y_i) are concatenated into a vector $\mathbf{x}$, such that:

$$\mathbf{x} = [x_1, y_1, \ldots x_n, y_n]^T \tag{2.33}$$



Example 1

Example 2

Figure 2.13: Two examples from a training set with labelled landmark points, exhibiting some deformation.

Then the positions of landmark points across the range of examples can be expressed as:

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{P}\mathbf{b} \tag{2.34}$$

Where $\overline{\mathbf{x}}$ is the mean position of the points, $\mathbf{P}$ is the matrix of eigenvectors of the covariance matrix which describes how points vary from their mean and $\mathbf{b}$ is a vector of weights or *model parameters*. If the variation of landmark point positions exhibit any linear correlation, which is usually the case, then only a few model parameters are needed to describe the modes of deformation in the training set. This results in a relatively concise representation of deformable shape and a small space in which to search for model parameters during recognition.

Given an estimate of the pose and scale of an object within a scene, the point distribution model can be placed over the scene and the landmark points iteratively moved towards image gradients. The incremental change in the position of each point at each iteration can be resolved to give a global change in the pose and scale of the model and a change in the model parameters, constrained by the modes of variation seen in the training examples.

Strictly speaking, this is a model fitting algorithm rather than an object recognition algorithm although the *goodness of fit* of the model can be used to decide whether or not a particular object is present. The technique is more suited to discriminating between similar objects from different classes, a good example of which is the recognition of faces [Lanitis *et al* 94]. In this application the model is trained on many examples from each class and classification of test objects is based upon the fitted model parameters.

The need to have an initial estimate of the pose and scale of an object before using a point distribution model severely limits its use as a general solution to deformable object recognition - if the pose and scale is already known then the object recognition problem is largely solved. The technique does have applications in more constrained environments when locating and inspecting an object which is expected in a scene. The use of image gradients to position landmark points provides insensitivity to changes in lighting and the use of global shape constraints provides some robustness to occlusion, clutter and image noise. Overall, the approach is an interesting solution for the representation of shape deformation.

2.8 Conclusions

The motivation behind this chapter has been to identify both the successful and unsuccessful approaches for shape based object recognition by critically reviewing some of the important contributions in the field to date. As a means of comparing the algorithms discussed, each has been assessed in terms of four significant properties, namely: reliability in the presence of typical scene noise including occlusion and clutter; good invariance characteristics to promote efficient recognition; the scope to represent arbitrary shape and sufficient descriptive power to represent large numbers of shapes uniquely. An accurate assessment like this is difficult without quantitative results but these are rarely provided by algorithm designers - in fact, this problem has motivated an analysis of reliability and descriptive power in chapters 4 and 5 respectively. Instead, each algorithm has been judged as being *reliable*, *invariant*, etc. and although this maybe somewhat subjective it does highlight the relative advantages of the different approaches. Table 2.1 presents the results of this analysis for each of the considered algorithms.

Probably the most important property required of a generic object recognition algorithm, and yet the one which few of the algorithms discussed have achieved, is reliability in a realistic environment exhibiting pixel noise, clutter and occlusions. The problem with most of the representations which fail in this respect is that they are based on global measurements which change if any part of the shape data changes. In the case of shape descriptors and Fourier descriptors the global shape is measured directly but in other cases, such as the polar parameterisation, measures are dependent on the recovery of the object centroid which itself is a global measure. Although the chain code is constructed from local measurements its lack of reliability can be attributed to the fact that the representation changes abruptly even for small changes in the shape data. The shape skeleton is also

		Reliable	Invariant	Good Scope	Descriptive
Shape	Shape descriptors	-	•	•	-
	Shape skeletons	-	•	-	-
Contours	Chain codes	-	-	•	•
	Polar param.	-	•	•	-
	Tangential param.	•	•	•	•
	Fourier descriptors	-	•	•	•
Regions	Template matching	•	-	•	•
	Log-polar mapping	-	•	•	•
	Moment invariants	-	•	•	•
Features	Relational graphs	•	•	•	•
	Interpretation tree	•	•	•	•
	Geometric hashing	•	•	•	•
	Hough transforms	•	•	•	•

Table 2.1: A summary of some of the most important aspects of the reviewed algorithms. A • indicates that this algorithm has good properties of the respective type

constructed from local information but it only takes small amounts of scene clutter and occlusion to prevent the thinning procedure from recovering the correct topology.

The template matching scheme promotes reliable recognition whilst providing good descriptive power but its lack of invariance properties renders it computationally unattractive. Of the contour representations the tangential parameterisation is the most attractive, exhibiting all of the desired properties. The disadvantage of this approach is that, like all of the contour and region based representations, it depends upon good prior segmentation of the image.

The class of representations which outperforms all others in this analysis are those based on image features. Image features, by their nature, are local descriptions of shape and can be recovered in realistic scenes and by basing representations on relative measurements between features, good invariance properties can also be attained. By selecting appropriate features it is possible to represent all types of shape providing good scope and descriptive power. The disadvantage of feature based representations is that matching scenes to models can be computationally very expensive when considering all model-feature to scene-feature mappings. In the next chapter, however, a strategy for limiting the number of mappings between model and scene features is introduced providing a feature based object recognition strategy with all of the properties discussed here but which is also efficient for matching.

Chapter 3

Object Recognition using Pairwise Geometric Histograms

3.1 Introduction

Many of the object recognition techniques reviewed in the last chapter provide a good solution in sufficiently constrained environments, and this is reflected by their adoption in many practical systems. In general though, we would like to be able to relax many of these constraints, opening up many more areas of application and ultimately find a generic solution which is an equal to (or even superior to) our own vision systems. Although such a generic solution appears to be a very long way off, a great deal of useful experience has been gained during the development of the existing techniques. It is unlikely that any one approach will be appropriate for all types of scenes, under a range of different viewing conditions and a generic solution is likely to involve fusion of the results of specialised *vision modules.* Current research must focus on these specialised modules, and in particular on visual properties which carry a large amount of information such as shape.

It is quite clear that one of the primary requirements of a generic object recognition strategy is insensitivity to large changes in the image data due to occlusions and scene clutter. It was concluded in the last chapter that algorithms based on image features can help promote such insensitivity (because image features tend to be defined locally) but it was also seen that current feature based approaches suffer from the large number of mappings that can exist between scene and model features. In the next section the use of image features is discussed in a little more detail and it is shown how the number of mappings between scene and model features can be greatly reduced by employing an appropriate feature representation.

The use of *Pairwise Geometric Histograms* as a robust, statistically based descriptor of image features was first presented in the work of Evans [Evans 94]. These descriptors allow scene image features to be classified according to known model features, simplifying the task of identifying arrangements of scene features which are consistent with stored models. This promotes a complete object recognition strategy which is both reliable and efficient. In section 3.3 the construction of pairwise geometric histograms for representing line features is explained, and some of the important properties of this representation are discussed.

An important factor in the design of any classifier is the selection of a similarity metric which allows meaningful comparison between training data and unseen data. A generally accepted technique is to use the likelihood of observing some data given each of the classes that data may have been drawn from as the measure of similarity between the data and each of those classes. This has led to the use of statistical metrics such as the mean absolute distance (MAD), χ^2 and the Malhalanobis distance. This issue is discussed in more detail in section 3.4 and a metric appropriate for classifying scene image features based on the similarity of geometric histograms is derived. The performance of this classification scheme is then demonstrated in section 3.5 for a range of test images exhibiting realistic levels of occlusion and clutter.

In the original work by Evans [Evans 94], a generalised Hough transform is used to collate evidence from classified features about the presence of particular objects in a scene, whilst simultaneously identifying their likely positions. Although this technique performs adequately, a number of advantages are gained by replacing this with a probabilistic Hough transform - a maximum likelihood formulation of the Hough transform presented by Stephens [Stephens 90]. This allows variability in the relative pose of features to be modelled allowing the position of objects to be determined to a much greater accuracy and removing the dependency of the position estimate on the quantisation of the parameter space to be removed, increasing robustness. The probabilistic Hough transform not only identifies the most likely position of an object within a scene but explicitly provides information about the expected error on the position. Knowledge of this error is essential if the algorithm is to be robustly integrated into a larger system. In section 3.6 the use of the probabilistic Hough transform in this application is described and in section 3.7 experimental results are presented.

3.2 Shape Features for Object Recognition

Image features such as corners, line segments and curved sections provide a good basis for object recognition because they can be recovered over a wide range of viewing conditions and object transformations. These types of features also promote successful recognition under conditions of scene occlusion and clutter because they are defined locally. The main criticism of feature based techniques is the need to exhaustively consider many possible mappings between features recovered from a scene with stored model features. Consistency between scene and model features is usually identified by searching through many combinations of pairings between scene and models using graph matching or by accumulating evidence using Hough transform type methods. When using graph matching to explore model-to-scene consistency a combinatorial explosion of mappings occurs rendering the technique impractical for more than a few relatively simple models. Even when using Hough transform based approaches, which are computationally more efficient than graph matching [Davies 90], the amount of processing necessary can still be very large and, perhaps more significantly, the parameter space, which is searched to find agreement between models and scenes, quickly becomes *noisy* and reliability suffers.

Considerable improvements in efficiency and reliability can be gained by introducing constraints which limit possible mappings between scene and model features. For example, when considering corner features the local image curvature may be used as a distinguishing characteristic of the corner and only pairings between scene and model corners with similar local curvature need to be considered. By introducing further constraints the number of potential pairings is pruned even more with a further increase in performance. In the work that follows, this approach has been extended to the extent that each feature is represented by a (near) unique descriptor based upon local geometry. The result is that only a single mapping (or at least very few mappings) exist between scene and model features promoting efficient and reliable recognition.

This approach can be adopted, at least in principle, to represent and classify all types of image feature although the work here is limited to the representation of straight line segments as first investigated by Evans [Evans 94]. The choice of line segments as the basic shape primitive means that the class of shapes to be represented is not restricted, as would be the case with corners for example. Even shapes comprising of only curves can be approximated to a specified precision by straight lines and, as will be seen shortly, errors introduced by this approximation can be accounted for.

3.2.1 Approximating Image Data with Straight Line Segments

The straight line approximation algorithm which has been used is a development of the recursive-split algorithm described by Ballard and Brown [Ballard & Brown 82]. Image data of models during training and of scenes during recognition is first processed by a suitable edge detection scheme which produces strings of connected edge pixels - Canny [Canny 86] has been used here although any edge detection and linking scheme could be adopted. In Ballard and Brown's scheme, each edge string is then repeatedly approximated by increasingly shorter line segments until the maximum perpendicular distance from any line to the edge string is below some specified threshold, λ. This is achieved by repeatedly splitting line segments with a maximum perpendicular distance greater than the threshold at the point of maximum distance. The development introduced here is to place a threshold on the ratio of a line segment's length to its maximum perpendicular distance to the edge string rather than on the maximum perpendicular distance alone. The result of this modification is that inaccuracies in representing edge strings are proportional to the absolute size of the image structures so that fine details are represented more closely without the need to *over represent* coarser details. This approach was designed so that it would be invariant to the scale of the image data in the sense that the same image at different scales should be approximated by the same number of line segments. An example of the line segmentation process is shown in Figure 3.1.

Figure 3.1: Approximation of edge strings by straight line segments. (a) The original image data. (b) The polygonised data.

3.3 Statistical Representation of Shape

Although image features can be characterised to some extent by intrinsic attributes such as local image gradients and curvatures, the context of the surrounding shape geometry provides the basis for a much more powerful descriptor. By recording the geometrical relationships between a feature and *each* of the surrounding shape features or primitives the feature is fully defined in terms of its shape context. By carefully selecting appropriate measures and storing these measurements in the form of a frequency histogram, a concise shape descriptor which promotes efficient and robust feature classification can be produced. This frequency histogram is referred to as a *pairwise geometric histogram* because it records geometric measures made between pairs of image features.

The selection of geometrical measurements with which to form a particular type of pairwise geometric histogram is motivated by two, possibly opposing, requirements. On the one hand it is desirable to make measurements which together, fully define the relationship between a pair of features, producing a unique descriptor. On the other hand, it is important to select measures with good invariance properties to promote efficient classification and which are stable under expected noise conditions to promote robustness.

The geometric relationship between a pair of line segments is well defined by the relative angle between them and the range of perpendicular distance obtained when the endpoints of the second line are projected onto the first. These relationships are depicted in Figure 3.2. Although this does not fully constrain the relationship between the pair of lines (the second line is free to translate parallel to the first) it is invariant to rotations and translations of the line pair. Importantly, these measures also exhibit stability if any of the lines become fractured which frequently occurs in real images. Entries made in the histogram for these measurements are weighted by the product of the lengths of the two line segments. This assigns an equal amount of significance to each edge pixel of the shape and ensures that fragmented entries add up correctly. Figure 3.3 (a) depicts the histogram entry made for the line pair in Figure 3.2 and Figure 3.3 (b) depicts the multiple entries made if the second line becomes broken. Clearly these representations are very similar.

3.3.1 Accounting for Measurement Errors

Naturally, the measurement of image features is prone to measurement errors and further error is introduced by the approximation of edge strings by straight line segments. Unless some account is taken of these, the resulting pairwise geometric histogram will not be truly representative of the shape data it describes. Ideally, a *set* of training examples of

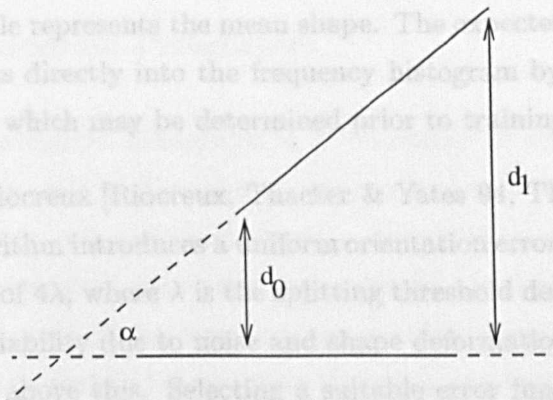
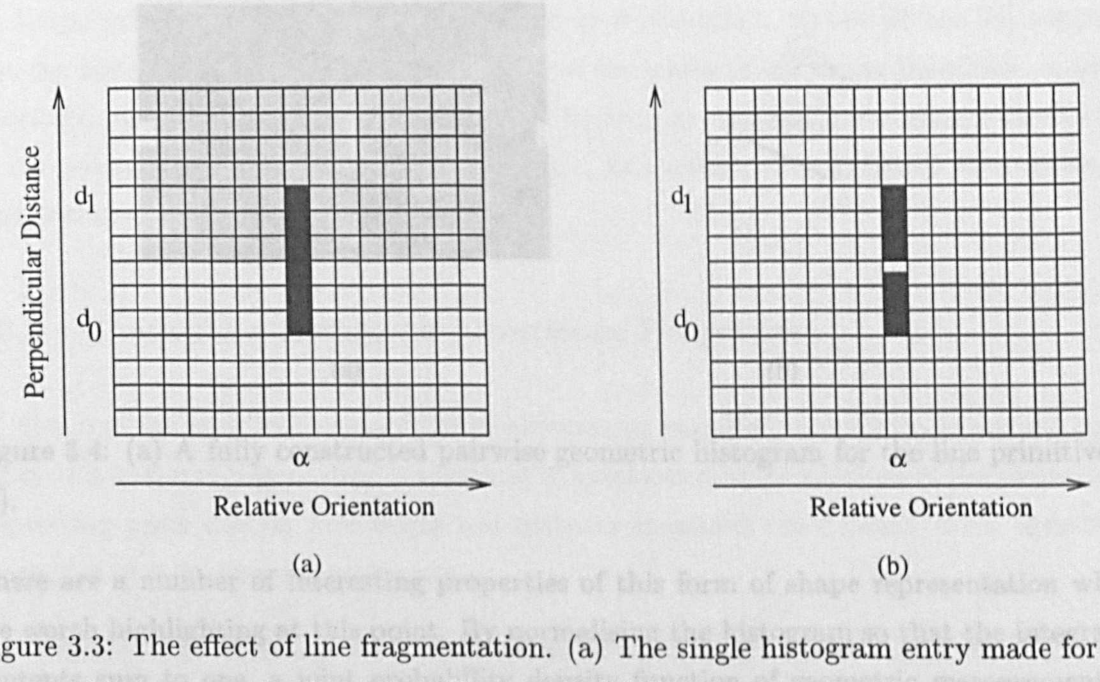


Figure 3.2: The relationship between a pair of line segments may be represented by their relative orientation, α, and the range of perpendicular distance from d_0 to d_q.



(a)

(b)

Figure 3.3: The effect of line fragmentation. (a) The single histogram entry made for the relationship between the line pair in Figure 3.2. (b) The multiple entries made for the same line pair when the second line is fragmented add to give almost the same representation.

each shape line segment should be used to construct each geometric histogram so that the distribution of the feature measurements is recorded. An alternative to this approach which gives approximately the same result but only requires single training examples is to assume that each example represents the mean shape. The expected error is then encoded on feature measurements directly into the frequency histogram by blurring (convolving) with the error function, which may be determined prior to training.

It has been shown by Riocreux [Riocreux, Thacker & Yates 94, Thacker*et al* 95] that the line approximation algorithm introduces a uniform orientation error for curved edge strings with a maximum width of 4λ, where λ is the splitting threshold described in section 3.2.1. To allow for further variability due to noise and shape deformation the magnitude of the error may be increased above this. Selecting a suitable error function for perpendicular distance measurements is less clear but a uniform distribution is a reasonable choice as this corrects for line fragmentation and allows for some lateral shift of lines. The scale of this error is chosen to be of the order of the bin width along the perpendicular distance axis of the histogram. An example of a fully constructed histogram with appropriate blurring which represents the line primitive highlighted in Figure 3.4 (b) is shown in Figure 3.4 (a).

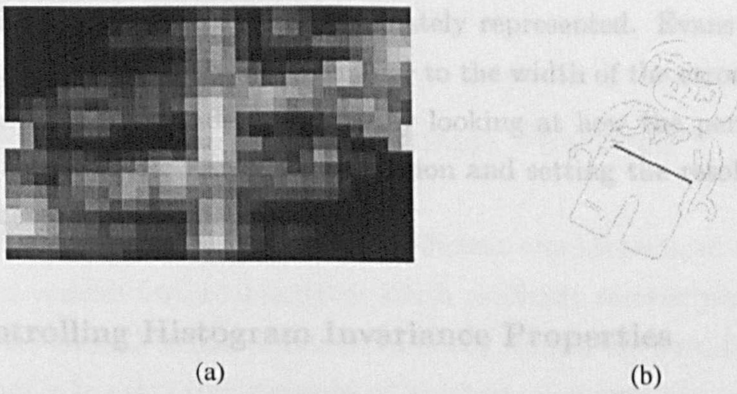


Figure 3.4: (a) A fully constructed pairwise geometric histogram for the line primitive in (b).

There are a number of interesting properties of this form of shape representation which are worth highlighting at this point. By normalising the histogram so that the integrated contents sum to one, a joint probability density function of geometric measurements is obtained. This statistical interpretation permits the use of statistically based classification techniques which are discussed in the next section. An important property of the geometric histogram used here is the proportion of the bins which are empty. This is described as sparseness. Typically, for the shape data used in this thesis, more than half of the bins are empty. This promotes robust classification in cluttered scenes as clutter data in scene histograms is *unlikely* to correlate with model data in model histograms.

3.3.2 Selecting Histogram Parameters

Prior to constructing geometric histograms it is necessary to decide on the histogram scale and resolution. The choice of maximum perpendicular distance, d_{max} is driven by two opposing requirements. On one hand the d_{max} should be small so that the pairwise geometric histogram represents local shape and is robust to missing data and occlusion. On the other hand d_{max} should be large enough so that enough shape information is present in each pairwise geometric histogram so that they are distinct from each other. In practice a good rule of thumb is to ensure that about half of a shape is encoded into each histogram.

Again, the selection of the histogram resolution is driven by opposing requirements. If the resolution of the pairwise geometric histogram is high then it will precisely describe the shape primitive features. However, this is at the expense of requiring a large amount of memory to store and large amount of computation to match. On the other hand if the histogram resolution is coarse, then storing and matching will be less expensive but the shape primitives will only be approximately represented. Evans [Evans 94] suggests that the histogram bin size should be similar to the width of the errors functions. A more principled choice can be made, however, by looking at how the performance is effected by the pairwise geometric histogram resolution and setting the resolution appropriately. This is considered later in chapter 4.

3.3.3 Controlling Histogram Invariance Properties

By virtue of the relative nature of the measurements recorded in pairwise geometric histograms this shape representation is invariant to rotations and translations of the shape data. Depending upon exactly how angle and distance measures are defined, other invariance properties may or may not also exist.

The simplest type of histogram is constructed by restricting angles to the range 0 to π and distances to the range 0 to d_{max}. This histogram is invariant to reflections of the shape data about the reference line and is described as *mirror symmetric*. Mirror reflection invariance is not always desirable and can be removed by using the handedness of angles (clockwise or anti-clockwise) to extend the range of angle measurements to $-\pi$ to π. This doubles the area of the histogram which in turn doubles the computation needed for matching but also increases the sparseness which improves robustness of matching in cluttered scenes. This type of geometric histogram is named *rotate symmetric*. The area of the histogram can be doubled again, further improving the descriptive power of the representation, by directing the reference line towards the point where the line pair intersect and using this to define a reference frame. Measures of distance can then be signed depending upon whether they are to the left or right of the directed reference line, extending the distance range to $-d_{max}$ to d_{max}. This type of geometric histogram is named *directed* and is the type used in the experiments presented in this thesis. These 3 histogram types are shown in Figure 3.5. A full description of further histogram types is described by Riocreux [Thacker*et al* 95].

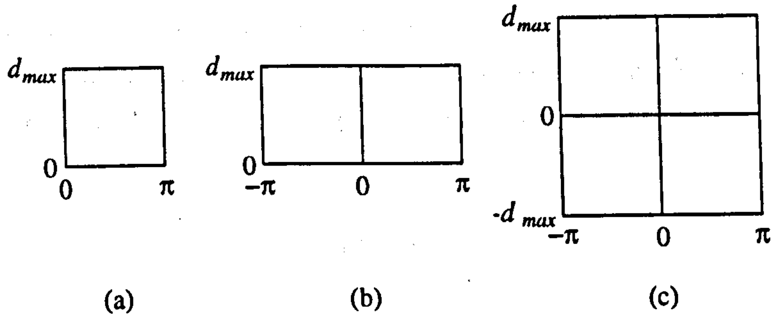


Figure 3.5: Some of the different geometric histogram types. (a) Rotation, translation and mirror reflection invariant. (b) Rotation and translation invariant. (c) Rotation and translation invariant.

3.4 Classification of Scene Image Features

Geometric histograms promote robust and efficient classification of scene image features by providing a concise feature descriptor which explicitly records statistical variability in the shape data. Central to the classification of any data is the need for a similarity metric which provides a quantitative measure of similarity between seen and unseen data. A good metric is one which degrades gracefully as the data degrades and should take proper account of the errors on the data being compared.

3.4.1 Defining Similarity

Because pairwise geometric histograms are essentially binned conditional probability density distributions it is appropriate to use conventional metrics for comparing probability distributions as a measure of histogram similarity. The standard technique for deciding whether two sample distributions are drawn from the same underlying distribution is the χ^2 test. Given two sample distributions $S = \{s_1, \ldots, s_N\}$ and $M = \{m_1, \ldots, m_N\}$, the χ^2

statistic is defined as follows.

$$\chi^2 = \sum_i \frac{(s_i - m_i)^2}{s_i + m_i} \tag{3.1}$$

The term $s_i + m_i$ is used as an estimate of the error on the measurement of $(s_i - m_i)^2$ so χ^2 is essentially a sum of square differences in which each component is normalised by the expected measurement error. In practice this estimate on the measurement error is only valid when the two distributions are very similar and a better metric should be adopted.

To clarify this situation it is worth going back to first principles. Given a random variable X, a statistical measure of the distance D between the endpoints $X = x$ and $X = x + \delta x$ of a short line is obtained by normalising by the standard deviation σ.

$$D = \frac{\delta x}{\sigma} \tag{3.2}$$

In general, the statistical distance between any two points $X = s$ and $X = m$ can be determined by the definite integral:

$$D = \int_s^m \frac{dx}{\sigma} \tag{3.3}$$

For N independent measurements the statistical distance is given by a sum of squared components:

$$D^2 = \sum_i (\int_{s_i}^{m_i} \frac{dx_i}{\sigma_i})^2 \tag{3.4}$$

It is well known that binned data conforms to a Poisson distribution and that the variance of a Poisson variable is equal to its mean. A statistical distance metric for binned data is then obtained by substitution of $\sigma_i = \sqrt{x_i}$.

$$\begin{aligned} D^2 &= \sum_i (\int_{s_i}^{m_i} \frac{dx_i}{\sqrt{x_i}})^2 && (3.5) \\ &= 4\sum_i (\sqrt{s_i} - \sqrt{m_i})^2 && (3.6) \end{aligned}$$

Removing the constant factor in this expression gives the statistical metric proposed by Matusita [Matusita 55] which is known as the Matusita distance.

$$D_{matusita} = \sum_i (\sqrt{s_i} - \sqrt{m_i})^2 \tag{3.7}$$

Expanding this expression gives:

$$D_{matusita} = \sum_i s_i + \sum_i m_i - \sum_i \sqrt{s_i}\sqrt{m_i} \tag{3.8}$$

If both m and s are normalised, or when using this metric to compare a single scene pattern with a set of normalised model patterns, this is simply:

$$D_{matusita} = const - \sum_i \sqrt{s_i}\sqrt{m_i} \tag{3.9}$$

Removing the constant results in the Bhattacharyya distance.

$$D_{bhattacharyya} = \sum_i \sqrt{s_i}\sqrt{m_i} \tag{3.10}$$

The Bhattacharyya metric, then, is both a statistically valid and computationally efficient metric for comparing geometric histograms. It is computationally efficient as it requires only one multiply and one square root if model histograms are square-rooted during training.

3.4.2 Nearest-Neighbour Classification

Pattern classification strategies tend to partition the pattern space into class regions and test patterns are classified according to the region in which they fall. Class boundaries can either be defined explicitly, as is the case with discriminant functions which describe partitioning hyper-surfaces, or can be implied by the presence of exemplars as is the case with nearest-neighbour classifiers.

The classification of scene features using geometric histograms is performed here using a nearest-neighbour classifier. During training the pattern space is populated by normalised model pairwise geometric histograms and scene data is later classified according to the closest model. It is well understood that the optimal classifier in terms of reliability is the Bayes classifier where boundaries describe regions of pattern space where the probability of data coming from two or more classes is equal. Providing that the training data lies close to the class means the use of the Bhattacharyya distance, which properly accounts for errors in the data, results in a nearest-neighbour classifier which is equivalent to a Bayes classifier.

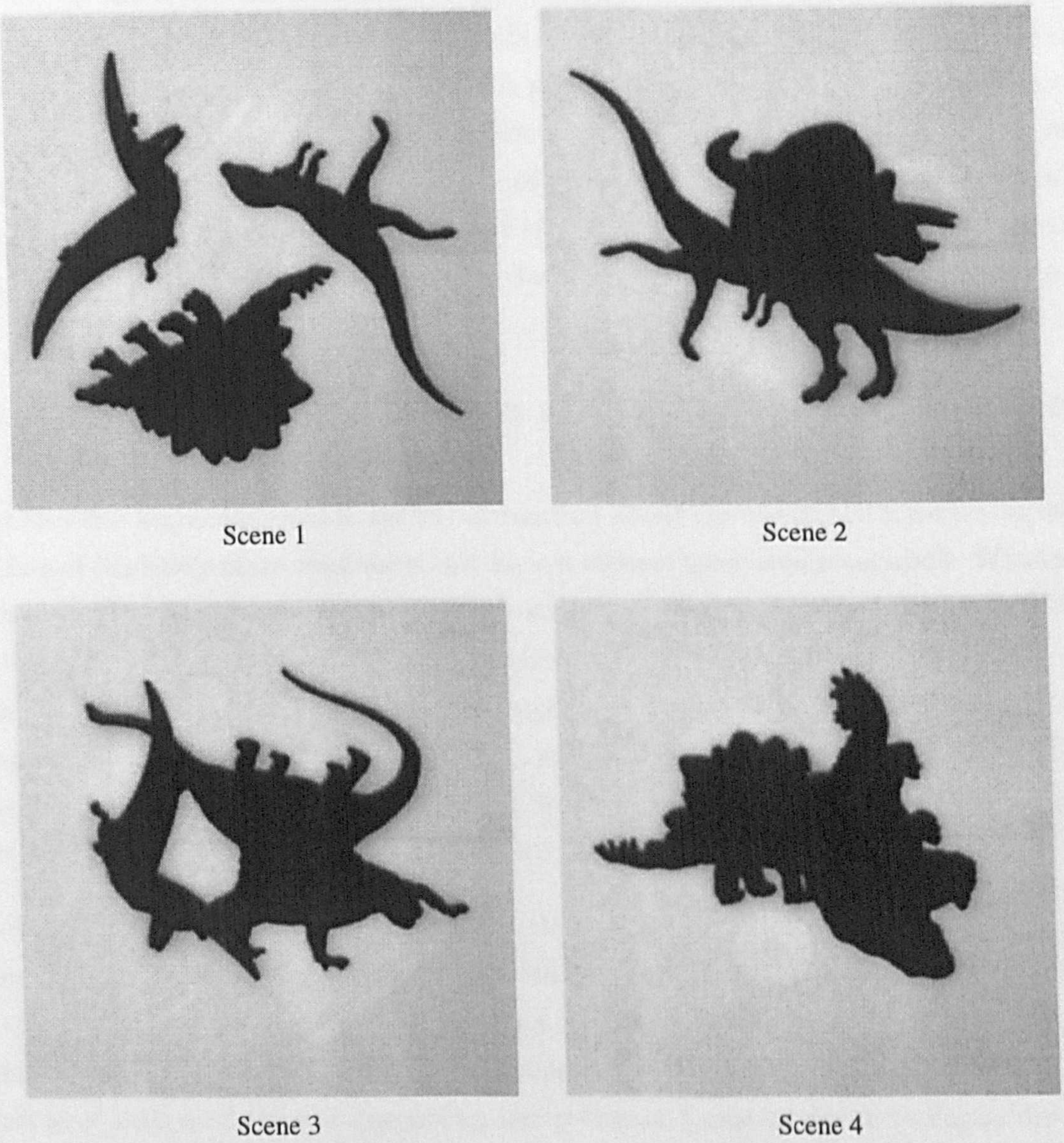
In practice, the excessive amounts of noise encountered in real images means that classification based solely upon the nearest-neighbour is found to be unreliable once large numbers of model shapes are being stored. A better strategy is to not only form a single classification based on the nearest-neighbour but to form multiple classifications based on a number of the nearest-neighbours. These additional classifications can be removed later when looking for a consistent set of primitives to form shape classifications. This issue is considered in detail later in chapter 4.

3.5 Experiments: Classification of Line Segments

The experiments presented here have been devised to demonstrate the effectiveness of using pairwise geometric histograms to classify scene line segments in real scenes which exhibit clutter and occlusion. The test images selected for these experiments contain several planar shape templates and a number of views of mechanical parts. The original images are presented in Figures 3.6 and 3.7 for the planar objects and mechanical parts respectively. The advantage of using some planar objects is that they can be arranged to overlap each other to produce good examples of occlusion. Objects are rarely planar in practice which is why mechanical parts have also been used to form more realistic scene data. Further examples can be found in Evans' thesis [Evans 94].

The dinosaur templates and mechanical parts shown in Appendix A Figures A.1 and A.3 were used as the training data for all of these experiments and they have been represented using the *directed* histogram type with a resolution of 40 distance bins and 64 angle bins. Figures 3.8 and 3.9 show the polygonised data derived from the test scenes after classification. The models present in each scene have been assigned a colour and line segments classified as belonging to that model have been drawn in this colour. Lines classified as belonging to a model not present in the scene have been drawn in black.

It can be seen from these examples that there are generally two cases where line segments are incorrectly classified. The first case is close to the point where occluding shapes meet such as the head of the Brontosaurus (green) and the uppermost wing of the pterodactyl (red) in Figure 3.8 scene 3. This is because in the vicinity of the occlusion the shape information in the scene is quite different to that in the training data and incorrect features may well give a better match. It is important to note, however, that away from the occluding regions the classifications are very reliable. The second problem is that the



Scene 1

Scene 2

Scene 3

Scene 4

Figure 3.6: These four test images contain planar shape templates which have been arranged to produce examples of scene clutter and occlusions.

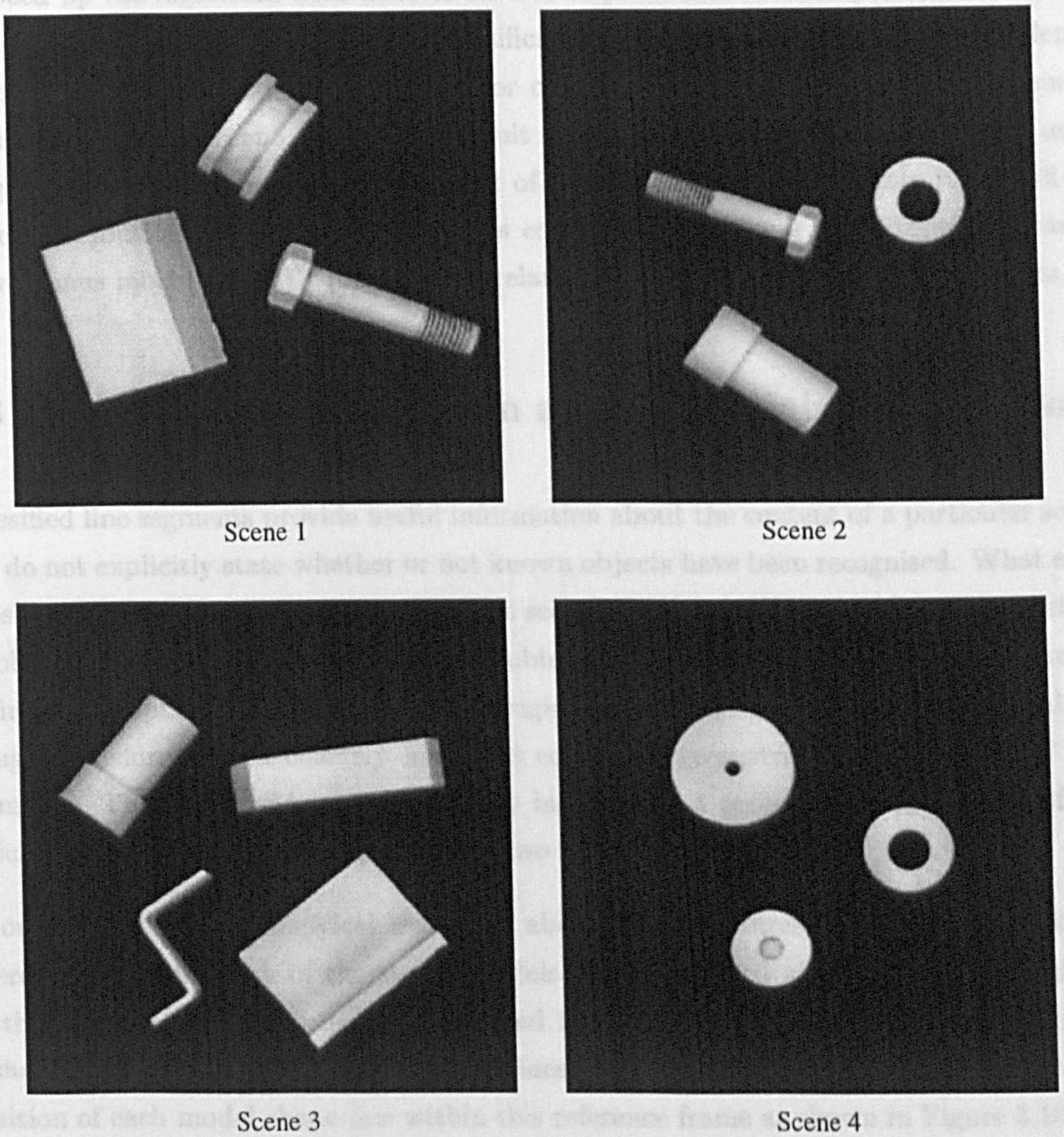


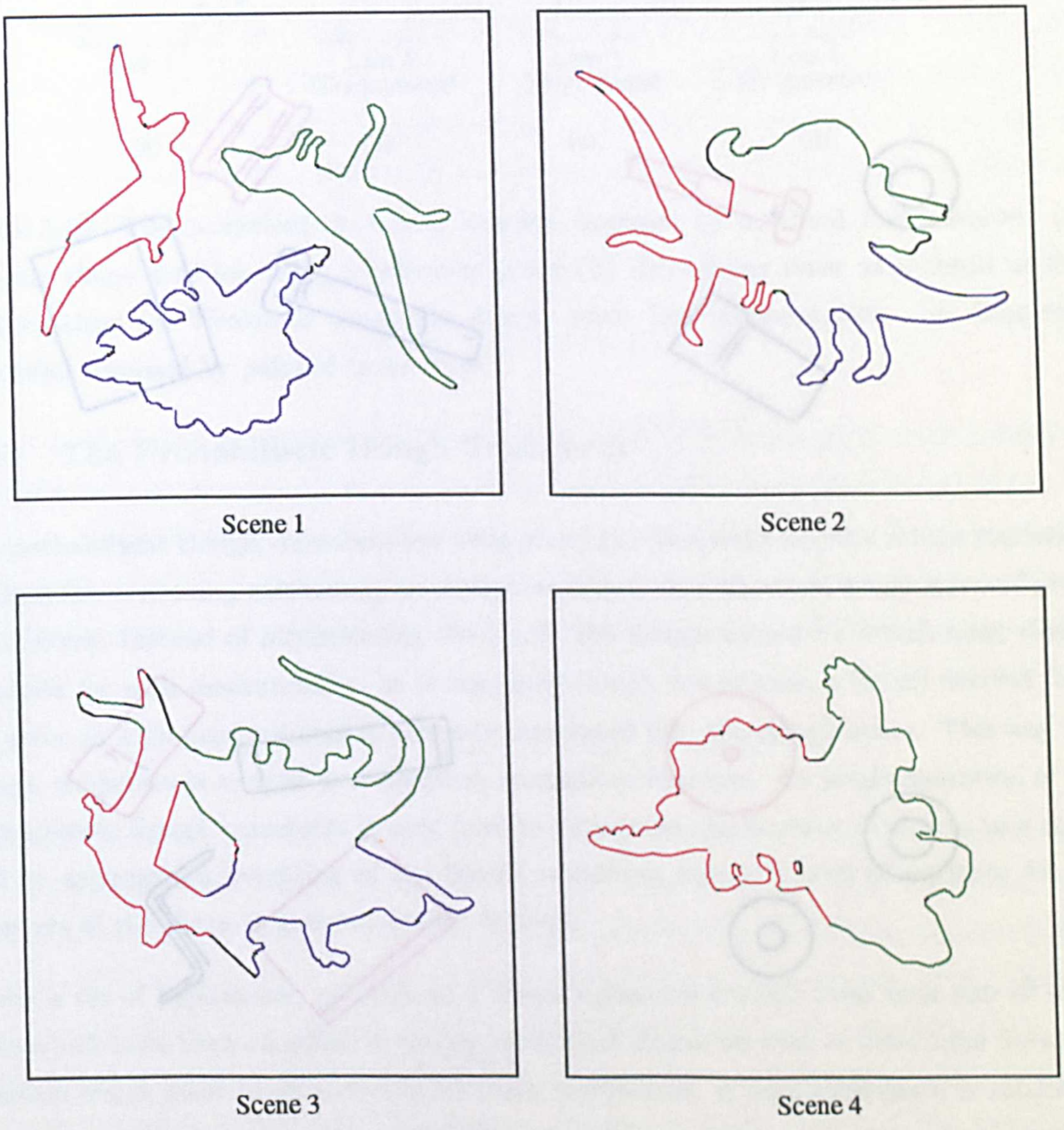
Figure 3.7: These four test images contain views of mechanical parts arranged to form relatively clutter scenes.

orientation and position of very short lines is much more variable than that of longer lines which leads to poor classification. Many incorrectly classified short line segments can be seen in all of the test images. This does not seriously impair performance as the significance of classified lines on the further stages of the algorithm is proportional to their lengths and, in practice, a threshold may be used to discard lines below a certain length to speed up the algorithm with little or no loss in performance. It is possible that scene clutter may influence some of the misclassifications although there is no direct evidence of this in any of the test scenes. The poor classification of the tail of the Stegosaurus (blue) in Figure 3.8 scene 1 may be a result of the shortness of the line segments used to approximate the fine detail on this part of the shape although certainly there will be some contribution to the scene histograms constructed for these lines from the nearby Antrodemus model (green) which may correlate well with incorrect training examples.

3.6 Hypothesis Combination and Determining Object Pose

Classified line segments provide useful information about the content of a particular scene but do not explicitly state whether or not known objects have been recognised. What each classified line provides is a hypothesis of the scene content and decisions as to the whether an object is present or not are made by combining hypotheses, in an appropriate manner, to find an acceptable level of agreement. Originally this has been done using a generalised Hough transform which coarsely identified consistent geometrical arrangements of line segments. This has now been improved by introducing a probabilistic Hough transform which is described and tested in the next two sections.

In order to do any geometrical reasoning about shape features it is useful to define a reference frame for each of the stored models shapes, centred at any arbitrary position, so that all of the features can be measured in relation to this frame. The position of a shape hypothesised by classified scene lines can then be estimated using the relative position of each model shape line within this reference frame as shown in Figure 3.10 (a). Unfortunately, the position of a shape hypothesised by each single classified scene line is unreliable because of the effect of scene line fragmentation. This problem is shown in Figure 3.10 (b). In fact, the actual shape position must lie along a line, shown as dashed in Figure 3.10 (c), which is parallel to the scene line and passes through the hypothesised position. A more constrained hypothesis for the position of a shape is obtained by finding the intersection, see Figure 3.10 (d), of the parallel line constraints from pairs of scene lines which have been classified according to the same shape.



Scene 1

Scene 2

Scene 3

Scene 4

Figure 3.8: The polygonised test data formed by approximating the edge strings detected in the original images by straight line segments. In each scene, each colour represents a different model from the training set and classified line segments are coloured accordingly.

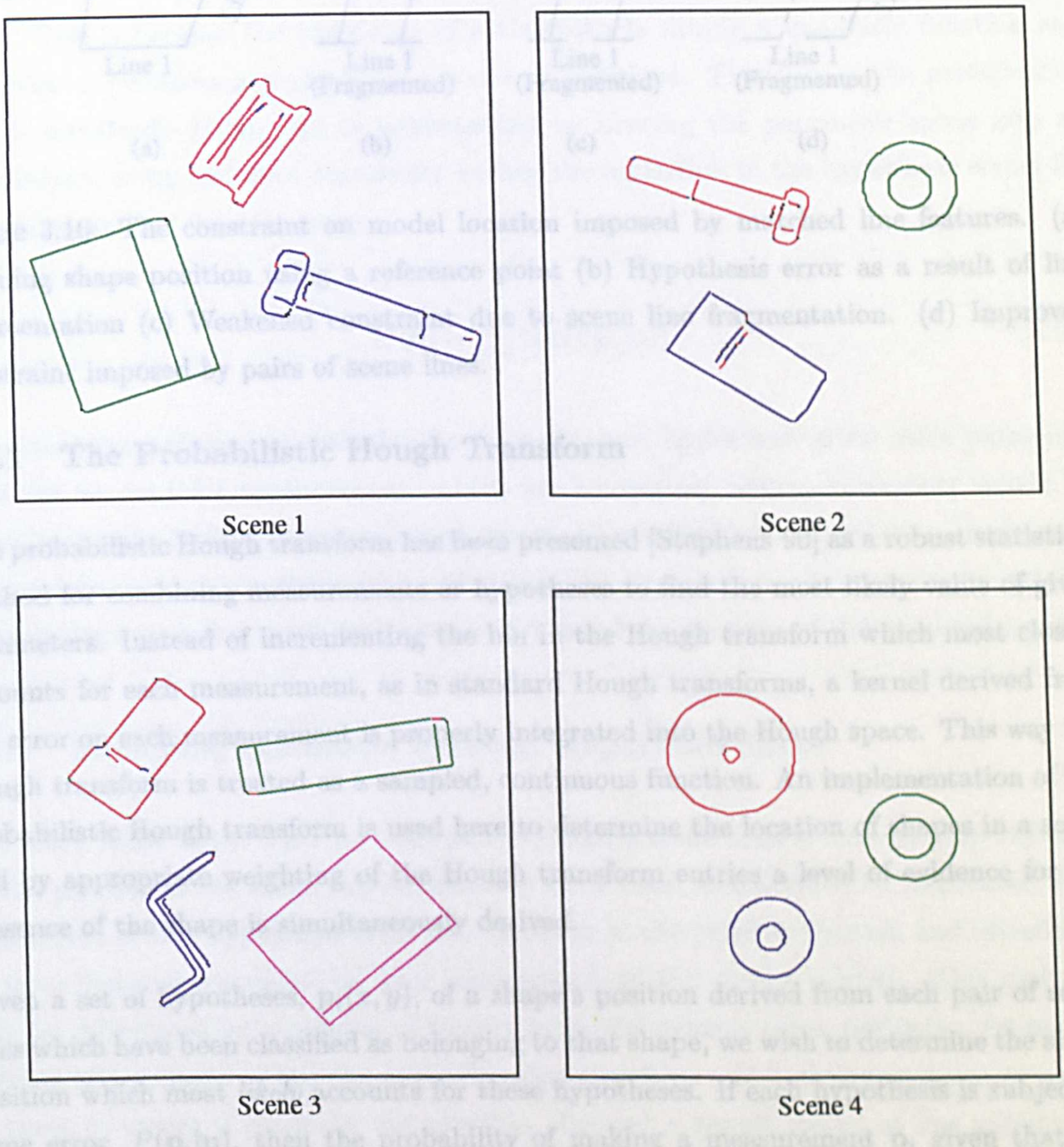


Figure 3.9: The polygonised test data formed by approximating the edge strings detected in the original images by straight line segments. In each scene, each colour represents a different model from the training set and classified line segments are coloured accordingly.

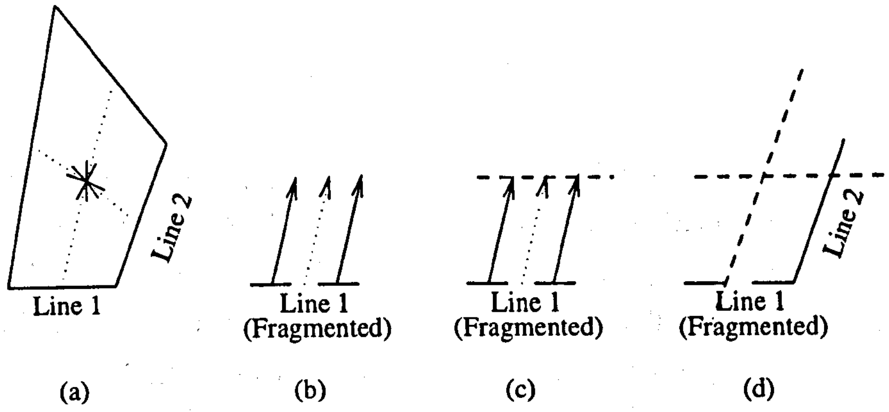


Figure 3.10: The constraint on model location imposed by matched line features. (a) Defining shape position using a reference point (b) Hypothesis error as a result of line fragmentation (c) Weakened constraint due to scene line fragmentation. (d) Improved constraint imposed by pairs of scene lines.

3.6.1 The Probabilistic Hough Transform

The probabilistic Hough transform has been presented [Stephens 90] as a robust statistical method for combining measurements or hypotheses to find the most likely value of given parameters. Instead of incrementing the bin in the Hough transform which most closely accounts for each measurement, as in standard Hough transforms, a kernel derived from the error on each measurement is properly integrated into the Hough space. This way the Hough transform is treated as a sampled, continuous function. An implementation of the probabilistic Hough transform is used here to determine the location of shapes in a scene and by appropriate weighting of the Hough transform entries a level of evidence for the presence of the shape is simultaneously derived.

Given a set of hypotheses, $\mathbf{p}_i(x,y)$, of a shape's position derived from each pair of scene lines which have been classified as belonging to that shape, we wish to determine the shape position which most *likely* accounts for these hypotheses. If each hypothesis is subject to some error, $P(\mathbf{p}_i|\mathbf{p})$, then the probability of making a measurement $\mathbf{p}_i$ given that the shape is actually at $\mathbf{p}$, and the error on each hypothesis is independent is simply:

$$\begin{aligned} P(\mathbf{p}_1 \dots \mathbf{p}_n|\mathbf{p}) &= P(\mathbf{p}_1|\mathbf{p})P(\mathbf{p}_2|\mathbf{p})\dots P(\mathbf{p}_n|\mathbf{p}) \\ &= \prod_i P(\mathbf{p}_i|\mathbf{p}) \qquad (3.11) \end{aligned}$$

This function of $\mathbf{p}$ is called the likelihood function and it is intuitive to base the estimate of the shape position $\hat{\mathbf{p}}$ on the value of $\mathbf{p}$ where this function is a maximum.

$$\hat{\mathbf{p}} = \max \prod_i P(\mathbf{p}_i|\mathbf{p}) \tag{3.12}$$

When the error function $P(\mathbf{p}_i|\mathbf{p})$ is represented by a Gaussian distribution it is easier to calculate the logarithm of the likelihood function rather than the likelihood function itself. This is because the logarithm of a Gaussian is simply a quadratic function and the series of products is replaced by a series of additions. This then is the probabilistic Hough transform, $H(\mathbf{p})$, and is implemented by binning the parameter space into an *accumulator array* and then repeatedly adding the logarithm of the hypothesis errors for each measurement into the array.

$$H(\mathbf{p}) = \sum_i \ln\left[P(\mathbf{p}_i|\mathbf{p})\right] \tag{3.13}$$

In general the pdf used to describe the error on some hypothesis given some parameter does not account for measurements which are completely wrong, commonly known as *flyers*, and these flyers can grossly distort the maximum likelihood calculation. A common approach to avoid this problem is to add tails to the error pdfs to allow a finite probability of measurements being made which are well beyond the confines of the expected error.

3.6.2 Modelling the Shape Position Hypothesis Error

The entry made in the PHT for each pair of labelled scene lines is derived from the error on the position of the shape hypothesised by those lines, as described by Expression 3.13. The source of this hypothesis error is a variation in the relative position and orientation of scene lines as compared to their counterparts in the stored models. This variation may be introduced by a number of factors including camera noise, occlusion, changes in lighting and artifacts of the line approximation algorithm and building an accurate model of these processes is generally not possible. Instead a simple model of line variation is adopted here which is found to significantly increase both the robustness and accuracy of the shape recognition and location process - the validity of this model is verified later in Section 3.6.3. The error model that has been used here assumes that the positions of scene line end points are subject to an isotropic, normally distributed variation which can be described by a covariance matrix $\mathbf{\Sigma}_{end}$.

$$\mathbf{\Sigma}_{end} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \tag{3.14}$$

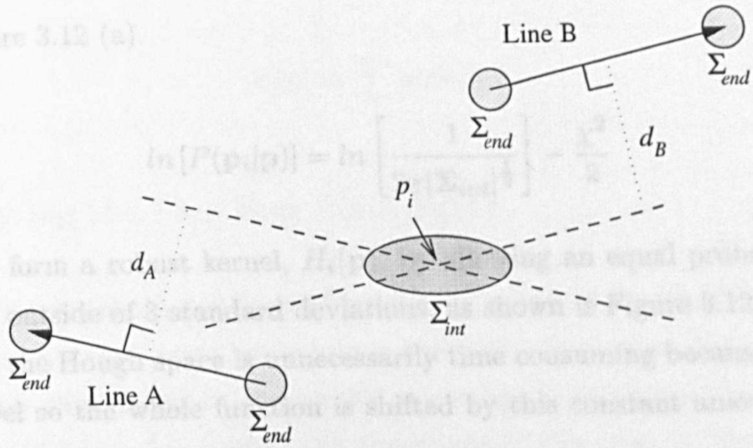


Figure 3.11: The variability in the pose of line segments can be approximately modelled by assuming normally distributed line endpoint errors. The resulting error on the position of the object reference point can then be determined using error propagation.

For each pair of lines this end point error manifests itself as an error in the point of intersection of the parallel constraints, as depicted is Figure 3.11. If the position of the line end points are $\mathbf{p}_{A1}$ and $\mathbf{p}_{A2}$ for "Line A" and $\mathbf{p}_{B1}$ and $\mathbf{p}_{B2}$ for "Line B" then the point of intersection, $\mathbf{p}_i$, is described by a function $\mathbf{p}_i(\mathbf{p}_{A1}, \mathbf{p}_{A2}, d_A, \mathbf{p}_{B1}, \mathbf{p}_{B2}, d_B)$. A full derivation of this can be found in Appendix B. If the error on the intersection point is described by the covariance matrix $\mathbf{\Sigma}_{int}$ and $\mathbf{p}_i()$ is assumed to be approximately linear locally then $\mathbf{\Sigma}_{int}$ can be expressed as:

$$\mathbf{\Sigma}_{int} = \nabla \mathbf{p}_i^T \mathbf{\Sigma}_{end} \nabla \mathbf{p}_i \tag{3.15}$$

Where $\nabla \mathbf{p}_i$ is the matrix of partial derivatives (the Jacobian matrix) of $\mathbf{p}_i$ with respect to the position of the line endpoints (again see Appendix B for a full derivation of this). For the assumption of normally distributed endpoint errors $\mathbf{\Sigma}_{int}$ describes an oriented bivariate normal distribution of the form:

$$P(\mathbf{p}_i|\mathbf{p}) = \frac{1}{2\pi|\mathbf{\Sigma}_{int}|^{\frac{1}{2}}} e^{-\chi^2/2} \tag{3.16}$$

Where χ^2 is the distance between $\mathbf{p}$ and $\mathbf{p}_i$ weighted by the magnitude of the errors. In fact, χ is the distance from $\mathbf{p}$ in standard deviations.

$$\chi^2 = [\mathbf{p}_i - \mathbf{p}]^T \mathbf{\Sigma}_{int}^{-1} [\mathbf{p}_i - \mathbf{p}] \tag{3.17}$$

Taking the logarithm of this gives an oriented quadratic surface, a cross section of which is shown is Figure 3.12 (a).

$$ln\left[P(\mathbf{p}_i|\mathbf{p})\right] = ln\left[\frac{1}{2\pi|\Sigma_{int}|^{\frac{1}{2}}}\right] - \frac{\chi^2}{2} \tag{3.18}$$

This is used to form a robust kernel, $H_i(\mathbf{p})$, by allowing an equal probability of finding a measurement outside of 3 standard deviations, as shown is Figure 3.12 (b). Integrating this kernel into the Hough space is unnecessarily time consuming because of the constant background level so the whole function is shifted by this constant amount to produce a localised function.

$$H_i(\mathbf{p}) = \begin{cases} 9.0 - \chi^2 & \text{if } \chi < 3 \\ 0.0 & \text{otherwise} \end{cases} \tag{3.19}$$

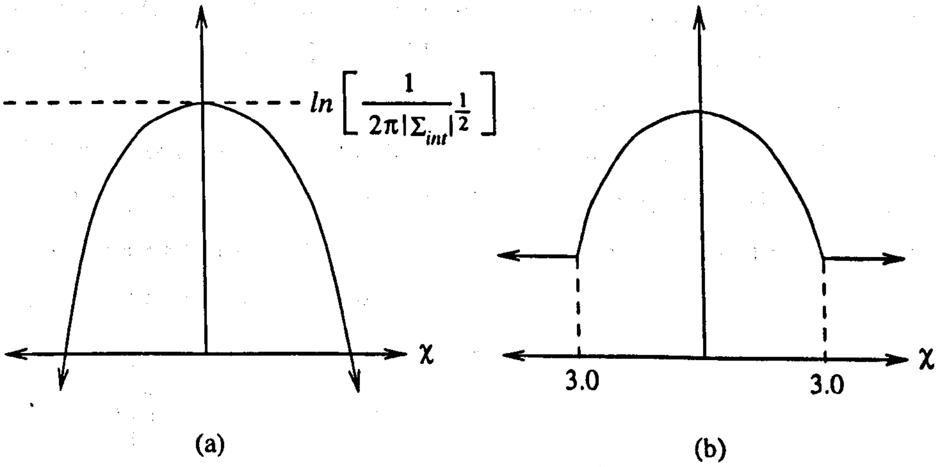


Figure 3.12: Formation of a robust kernel from the log likelihood function. (a) A cross section through the quadratic surface defined by $ln\left[P(\mathbf{p}_i|\mathbf{p})\right]$ (b) A cross section through the robust kernel derived from the surface in (a).

Although this shift is different for each hypothesis, the overall effect is simply to shift the height of the final Hough space surface, with no effect on the actual shape of the surface. Finally the magnitude of the kernel is weighted by the product of the length of the lines used to derive the hypothesis. This is necessary for entries to add together properly such that the entry for a pair of line segments is equivalent to the sum of the entries for pair of edge pixels which the lines approximate (except of course that pose error on individual edge pixels is greater resulting in a more spread entry). This weight also allows the height of the peak to be interpreted as a measure of the proportion of the shape present in the scene. The probabilistic Hough transform is then constructed by accumulating the weighted kernels.

$$H(\mathrm{p}) = \sum_i w_i H_i(\mathrm{p}) \tag{3.20}$$

3.6.3 Validating the Line End Point Error

The use of a normally distributed line end point error to model the variability in the relative pose of line segments leaves two important questions to be answered. First we would like to know whether this model is an acceptable one given the actual pose variation and secondly we would like to know what the magnitude of this error is. These questions can be answered by examining the distribution of the error on hypothesised shape positions, normalised by the error predicted by the model, for an arbitrary test scene. This is the distribution of χ for each of the hypotheses. Figure 3.13 shows the distribution of χ for the location of the Pterodactyl in Figure 3.8 scene 1 with an end point error of 1 pixel assumed. The approximately Gaussian shape of this distribution suggests that the endpoint error model is an appropriate one and the width of the distribution gives an estimate of the magnitude of the end point error. A point to consider when setting the magnitude of the end point error is that entries in the Hough transform must extend over several bins to be represented properly. This is effectively a sampling rate criterion. It is perfectly acceptable to choose errors which are higher than those measured so that the resolution of the Hough space can be controlled.

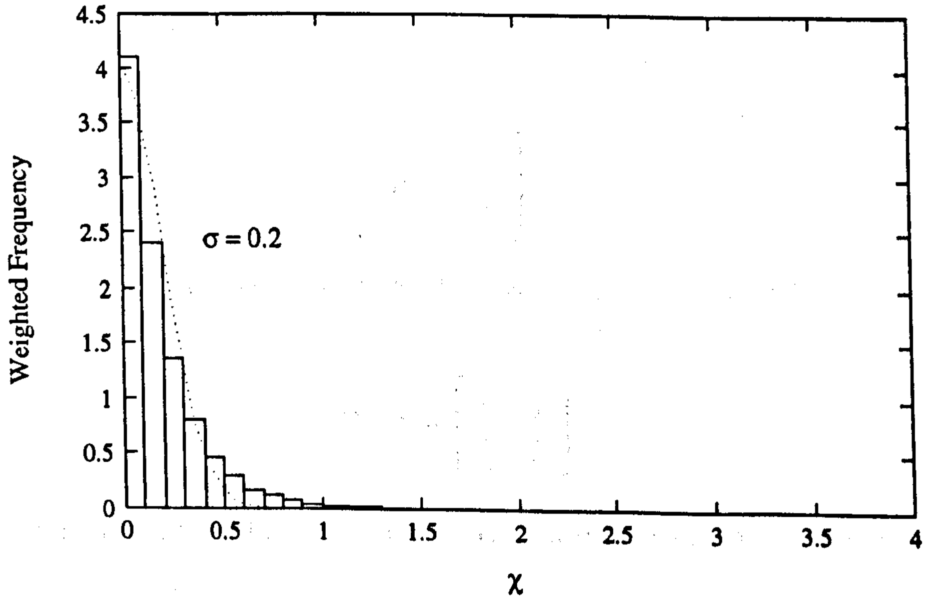


Figure 3.13: Validating the line end point error model and measuring the magnitude of the error by examining the distribution of χ for a set of hypotheses.

3.6.4 Estimating the Location Uncertainty

By using the probabilistic Hough transform the variability in the pose of scene lines can be accounted for and the most likely shape position determined as outlined above. Of course there will be an element of uncertainty associated with the determined shape position and a quantitative estimate of this uncertainty is needed. The position alone is really meaningless without some knowledge of the error.

The error on the estimated shape position $\hat{\mathbf{p}}$ is simply the width of the peak in the Hough transform and is quantified by finding the covariance matrix, $\mathbf{\Sigma_p}$ which describes the peak shape. As the peak is constructed by summing quadratic kernel entries, its shape is also quadratic and this can be measured by surface fitting.

$$S(x,y) = a + bx + cy + dx^2 + exy + fy^2 \tag{3.21}$$

As the surface describes a log-probability function, its height away from the peak can be expressed as a χ^2 and this is used to relate the covariance matrix $\mathbf{\Sigma_p}$ to the surface fit parameters.

$$[\mathrm{p} - \hat{\mathrm{p}}]^T \Sigma_p^{-1} [\mathrm{p} - \hat{\mathrm{p}}] = S(x,y) \tag{3.22}$$

Where.

$$\mathrm{p} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.23}$$

When this is expanded the coefficients of the inverse covariance matrix are found to be:

$$\Sigma_p^{-1} = \begin{bmatrix} d & \frac{e}{2} \\ \frac{e}{2} & f \end{bmatrix} \tag{3.24}$$

This is easily inverted to obtain the error on the estimated shape position Σ_p.

$$\Sigma_p = \frac{1}{df - \frac{e^2}{2}} \begin{bmatrix} f & -\frac{e}{2} \\ -\frac{e}{2} & d \end{bmatrix} \tag{3.25}$$

3.6.5 The orientation Hough transform

Significant peaks in the probabilistic Hough transform suggest the presence of known models in the scene data and the positions of these peaks indicate the most likely positions of those models, but no attempt has been made at this stage to determine their likely orientation. This is done separately for each identified model by isolating the lines which have contributed to the Hough peak and accumulating the difference in the orientation between these lines and the associated model lines. Isolating lines which have contributed to a particular peak is a matter of finding those lines whose parallel constraint passes within 3 standard deviations. A 1-parameter Hough transform is employed to accumulate the relative angle between the scene and model line data and the estimate of the model orientation is based upon the position of the peak in this space. The normally distributed line end point error implies that orientation estimates will also be subject to normal errors and these can be accounted for by making appropriate quadratic entries into the 1-parameter Hough transform. As with the location Hough transform this improves robustness, allows orientations to be determined to sub-bin accuracy by interpolation and provides a measure of confidence in the estimate.

3.7 Experiments: Hypothesis Combination and Determining Object Pose

The experiments presented here demonstrate the effectiveness of the probabilistic Hough transform for combining evidence from classified line segments allowing decisions to be made about the presence and pose of known objects. The first of these experiments combines evidence from the classified line segments for the test images presented in section 2.5. Further results are then provided which quantify the uncertainty in recovered shape positions and it is shown that this uncertainty can be estimated from the shape of the peak in the Hough transform.

3.7.1 Demonstration

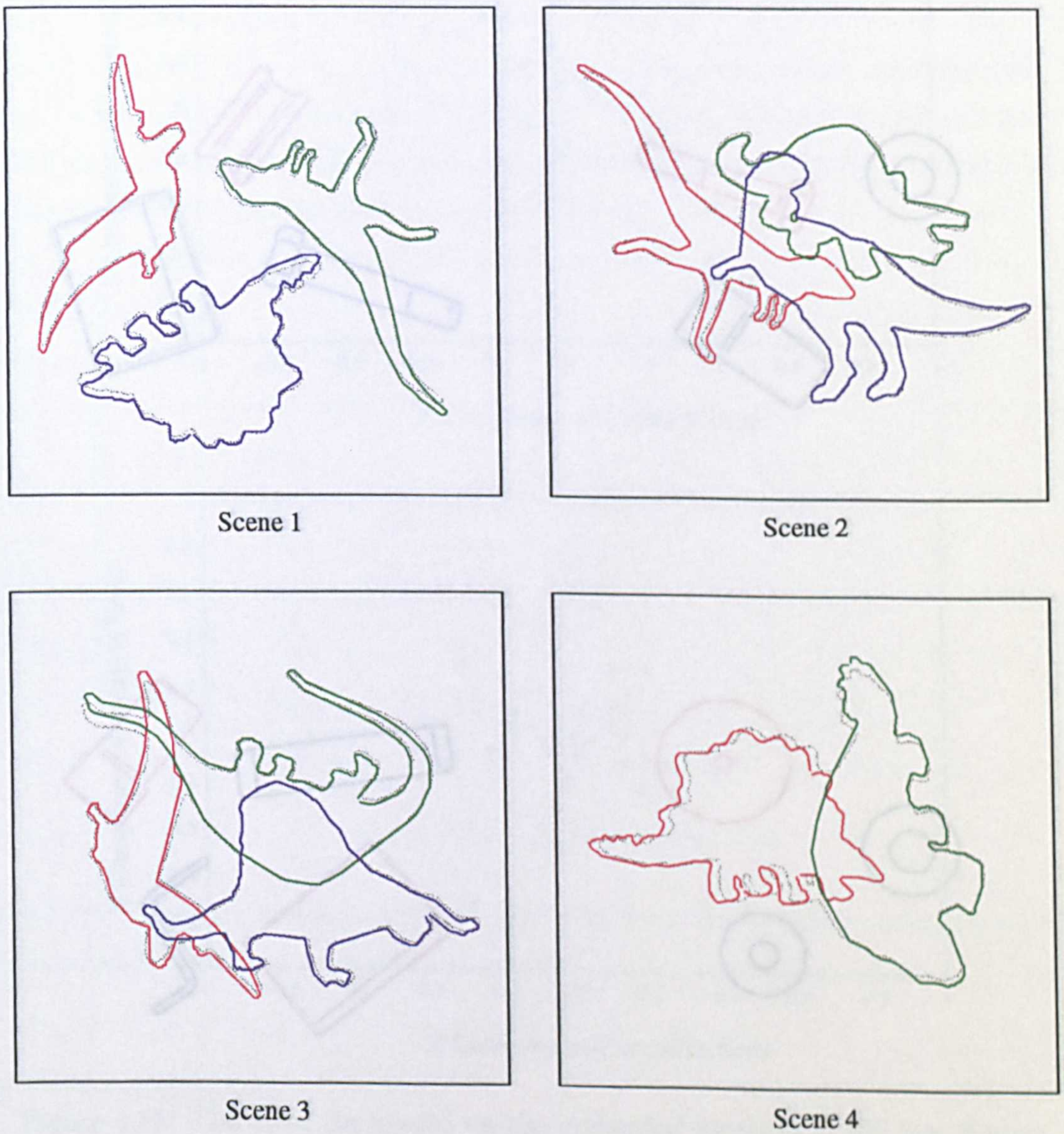
In Section 2.5 a number of test images of planar shape templates and views of mechanical parts were selected to demonstrate how pairwise geometric histograms can be used to classify scene line segments according to the line segments found in stored models. The results of this classification (see Figures 3.8 and 3.9) are used here to identify known shape models in the test scenes. A probabilistic Hough transform has been constructed for each of the known models and entries made for all appropriately labelled scene lines. Models for which enough evidence can be found in each of the scenes have been presented at their most likely pose in Figures 3.14 and 3.15.

In these examples the probabilistic Hough transform has successfully detected all of the models present in each of the scenes and determined what at least qualitatively appear to be good estimates of the shape poses. Most of the misalignment between the scene data and the overlaid models is explained by the variability in the scene data due to change of lighting, slight scale variation where templates are placed on top of each other and slight perspective effects for the mechanical parts. There are a number of specific problems which are worth highlighting however. In Figure 3.14 scene 3 the Brontosaurus' head (green) is poorly aligned with the scene data but by looking at the classified data in Figure 3.8 it can be seen that the head of the Brontosaurus has been poorly classified, providing little constraint on the position of the model. Another potential problem is the Stegosaurus model (red) in Figure 3.14 has been poorly localised along its length. In fact the model is poorly constrained in this direction because most of the vertical lines are very short and thus prone to a large error in position and orientation. This is a prime example of why it is important to be able to predict the uncertainty in the recovered shape positions.

3.7.2 Estimating location uncertainty

In the theoretical discussion above it was suggested that the shape of the peak in the Hough transform can be used to estimate the error on the recovered shape position. In the experiments presented here the magnitude of the positional error is quantified for a number of model shapes and the ability to predict this error is tested by comparing the predicted error with the variation of the recovered position of test shapes in known positions.

In order to measure the variability in the estimated shape position for a particular model, 100 test images were generated by applying a uniformly distributed, random transformation to the original image data and then adding some random, pixel noise. The range of transformations allowed was constrained so that the whole shape was present in each test image. The added pixel noise was generated from a Gaussian distribution with a variance of 5% of the pixel grey-level. The line segment data extracted from the generated test images exhibits significant variation over the original model, providing a good test whilst ensuring that the position of the model is still known so that this may be compared with the estimates for each test. Figure 3.16 displays the error in the position estimate for each of the 100 test examples for two different model shapes. Note that all points have been



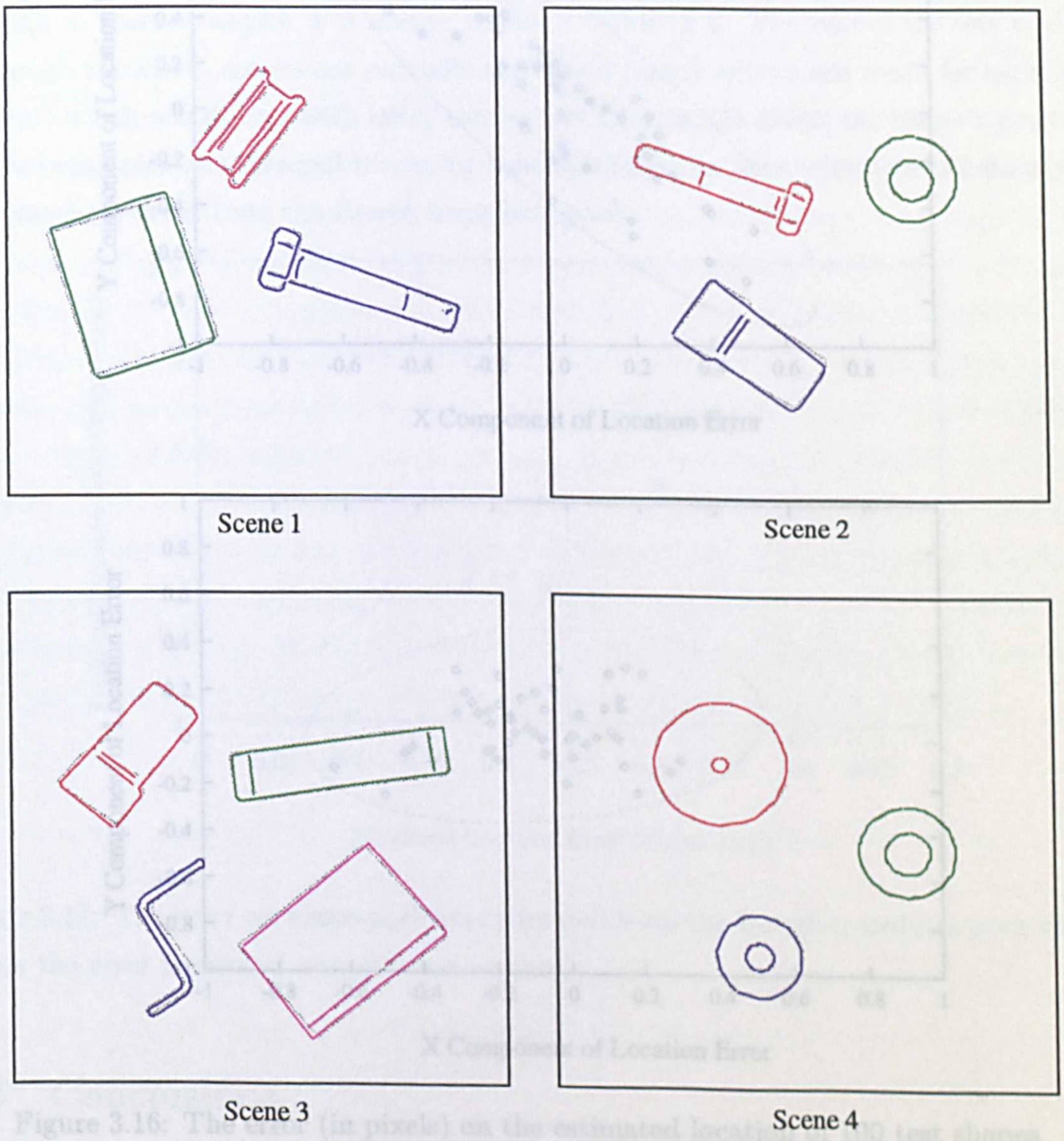
Scene 1

Scene 2

Scene 3

Scene 4

Figure 3.14: Recognition and localisation of silhouette shape data.



Scene 1

Scene 2

Scene 3

Scene 4

Figure 3.15: Recognition and localisation of mechanical components.

rotated into the model's local coordinate frame. The fitted ellipses represent the third standard deviation from the mean of each cluster. In both of these examples the error on the estimate, along the direction of greatest error, is of the order of 0.2 pixels.

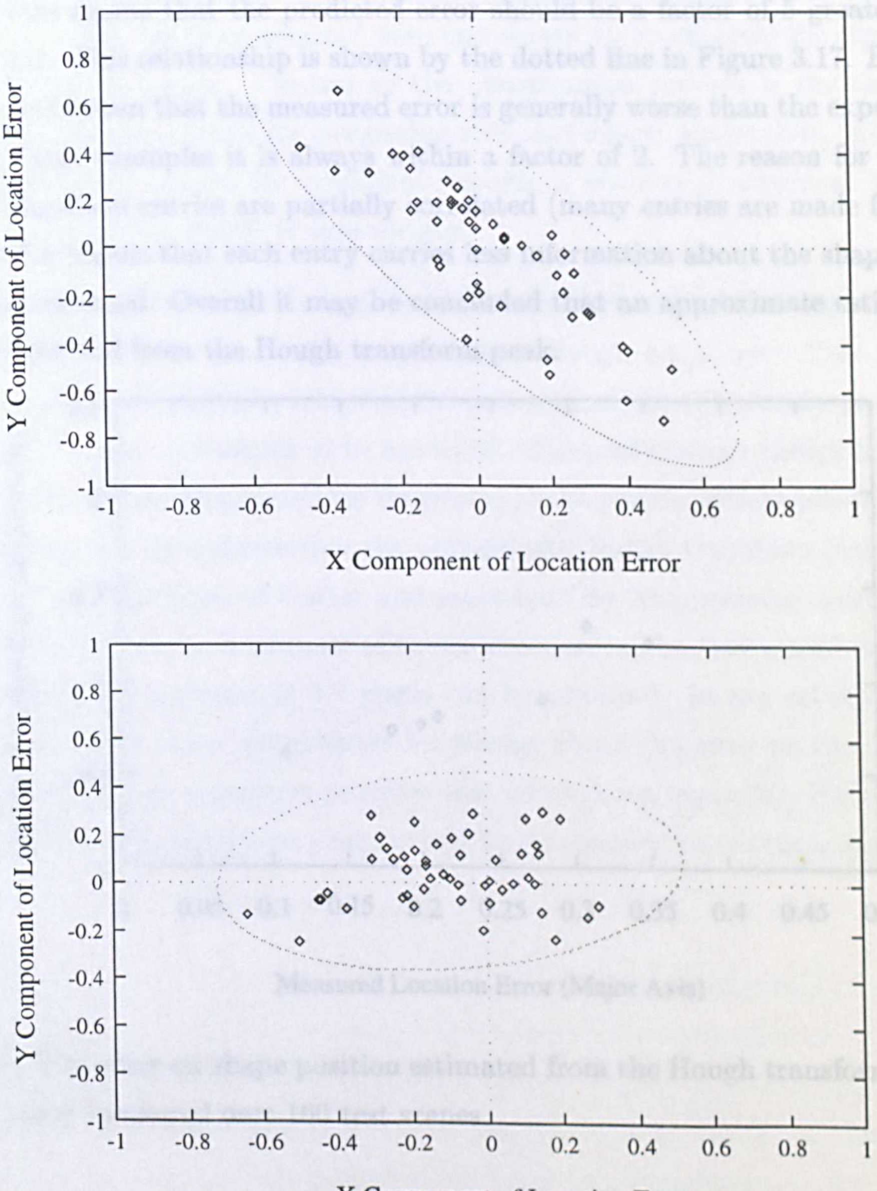


Figure 3.16: The error (in pixels) on the estimated location of 100 test shapes.

The ability to predict the error on the position estimate is tested here by both measuring and estimating this for a number of different models. The estimate is obtained by locating each model in its original image and measuring the variance along the major axis of the Hough transform peak. Each model is then located in 100 test scenes in which the position is known and the magnitude of the error along the major axis of the cluster is measured. Figure 3.17 shows the predicted error versus the measured error for each of the models used.

The first thing to notice about this data is that the predicted error is much worse than the measured error. This is because the magnitude of the line end point error used in the experiment is 1 pixel but the actual error, measured in Section 3.6.3 earlier was around 0.2 pixels. This means that the predicted error should be a factor of 5 greater than the measured error. This relationship is shown by the dotted line in Figure 3.17. Bearing this in mind, it can be seen that the measured error is generally worse than the expected value, although in these examples it is always within a factor of 2. The reason for this is that the Hough transform entries are partially correlated (many entries are made for each line segment) which means that each entry carries less information about the shape's position than is being assumed. Overall it may be concluded that an approximate estimate of the error can be derived from the Hough transform peak.

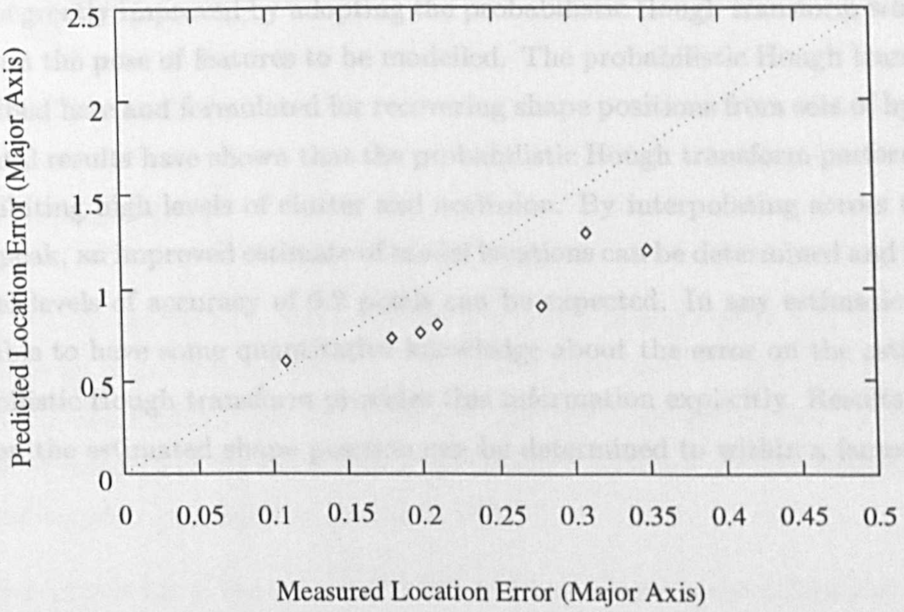


Figure 3.17: The error on shape position estimated from the Hough transform peak shape versus the error measured over 100 test scenes.

3.8 Conclusions

In this chapter it has been suggested that the key to developing a powerful object recognition strategy based on image features is to use a highly descriptive representation which permits individual features to be classified providing a limited number of hypotheses of a scene's content. This greatly improves efficiency by limiting the search needed to find feature groupings consistent with known model shapes. A shape representation which provides this level of description can be constructed by storing geometric measurements between pairs of features in the the form of a frequency distribution. The compactness of these *pairwise geometric histograms* promotes efficient feature matching and by explicitly coding feature measurement errors into the representation classifications it is reliable.

The representation of polygonised shape data using geometric histogram, as first presented by Evans [Evans 94], has been adopted here and shown to be effective for classifying line segments in difficult scenes. Central to the classification process is the choice of the Bhattacharyya distance metric for comparing seen and unseen shape data. Although this metric was originally chosen for empirical reasons, a statistical argument is presented which suggests why this metric is appropriate.

Final object classifications are formed by combining hypotheses provided by classified line segments, previously done using a generalised Hough transform. This stage of the algorithm is greatly improved by adopting the probabilistic Hough transform which allows variability on the pose of features to be modelled. The probabilistic Hough transform has been described here and formulated for recovering shape positions from sets of hypotheses. Experimental results have shown that the probabilistic Hough transform performs well on scenes exhibiting high levels of clutter and occlusion. By interpolating across the Hough transform peak, an improved estimate of model locations can be determined and it is shown that typical levels of accuracy of 0.2 pixels can be expected. In any estimation problem it is desirable to have some quantitative knowledge about the error on the estimate, and the probabilistic Hough transform provides this information explicitly. Results show that the error on the estimated shape position can be determined to within a factor 2.

Chapter 4

An Analysis of the Reliability of Recognition

4.1 Introduction

In general, the classification of measured data is prone to some error and the magnitude of this error is an important measure of a classifiers performance. Not only does the error rate reflect the classifiers success but when the classifier is integrated into a larger system, a quantitative measure of the error rate allows the performance of the complete system to be predicted by error propagation [Haralick 96].

Quantitative knowledge of the error rate has a number of uses in the design and application of a classifier. When designing a system which incorporates some element of classification the designer may wish to specify the maximum error rate which is acceptable. Knowledge of the error rate of different classification techniques not only allows the designer to compare these techniques with each other but allows the most appropriate one, which meets the systems requirements, to be selected.

Another use of the error rate is the selection of appropriate, or possibly even optimal algorithm parameters. Complex algorithms often have a large number of such parameters which control their behaviour and these need setting appropriately. Sometimes appropriate values for these parameters relate directly to quantities which can be either measured or derived from sample data or from other algorithm parameters. For example, when entries are made into a pairwise geometric histogram, the blurring applied to account for errors in the measurement of the angle between line primitives is derived from the algorithm which approximates the image data by line segments in the first place. When the meaning of algorithm parameters is less obvious selecting an appropriate value, and justifying this choice, is less straightforward. In these instances parameter values may be selected by observing their effect on the performance of the algorithm and selecting a set of parameters which result in acceptable, or even optimal, performance. To do this, a measure of performance must be defined and the error rate is a useful component of this.

A similar use of the error rate is to assess modifications made to the classification algorithm itself. This determines whether a change improves or degrades reliability and allows the changes in reliability to be compared with other important factors such as changes in the algorithm's complexity. This promotes a more methodical approach to algorithm development where the costs and benefits of specific modifications are well understood.

A fair criticism of most of the computer vision research conducted to date is the lack of analysis needed to quantify important performance characteristics such as the error rate [Haralick 92, Courtney Thacker & Clark 97]. This is both problematic for the researcher who wishes to build on existing work and to the engineer who wishes to identify solutions to well specified problems. For the research field to develop effectively it is necessary to be able to assess the relative merits of different algorithms and theories of vision. This allows research to be focussed on the techniques which look most promising. In order to be adopted by engineers, the performance of vision algorithms must be well understood so that they fit into the engineering design methodology. These issues have motivated the work presented in this chapter which investigates the reliability of scene feature classification when using pairwise geometric histograms.

4.1.1 Algorithm Scalability

Intuitively, the reliability of a classifier is likely to depend upon the number of different classes from which samples can be drawn. For many classification tasks this number is fixed and possibly quite small, and the classification error is frequently estimated from a number of test examples taken from each class. Conventional approaches to error estimation are briefly discussed in the next section. Even in some vision tasks the number of classes is sufficiently constrained to allow this conventional treatment. In general though, the number of classes in an object recognition application can be very large and possibly even unknown and conventional error estimation techniques are no longer practical. Even in relatively straightforward applications such as the inspection of industrial parts, the system may need to be flexible enough to allow additional shape classes to be added in the future and the impact of this change needs to be predictable.

Many of the object recognition algorithms which have been developed are demonstrated with a small number of model object classes and the error rate based on this small sample. It is not obvious how well these algorithm will perform as the number of models is increased however. This poses a more general question about the performance of object recognition algorithms. What is the impact on the performance of the algorithm as the number of model object classes is increased? This is a question of scalability. To begin to answer this question the four following issues must be considered:

1. How does the algorithm's computational requirement increase as the number of classes is increased?
2. How does the algorithm's memory requirement increase as the number of classes is increased?
3. How does the reliability of the classification process degrade as the number of classes is increased?
4. How many *unique* classes can be described using a given representation?

The designers of all types of algorithm are very familiar with the concepts of algorithm complexity which concerns the first two of these issues. The last two issues are specific to classification algorithms and in particular computer vision algorithms in which the number of classes can be very large. This more general issue of scalability is initially approached in this chapter by investigating the relationship between the number of model classes and the reliability of classification. To complement this, the fourth issue concerning the number of shape classes which can be uniquely represented is investigated in Chapter 5.

Most of the parameters used in the construction of geometric histograms have well defined values with the exception of the number of bins along the perpendicular distance axis. To provide a mechanism for selecting an appropriate bin size the effect of this parameter on the reliability of the algorithm is investigated in Section 4.8. The result of this analysis allows an appropriate bin size to be selected that will give a specific classification error.

4.2 Classification Error Estimation

The standard framework used for handling classification errors is probability theory. Given a number of classes $\{\omega_i : 1 \leq i \leq N\}$ and a sample $\mathbf{x}$, the error ε is the probability that the sample is assigned to the wrong class. If the conditional *a posteriori* probability of each class given the sample is known then it is usual to assign the sample to the class for which this probability is maximum. This is the Bayes decision rule.

$$\hat{\omega} = \max \arg P(\omega_i|\mathbf{x}) \tag{4.1}$$

where $\hat{\omega}$ is the class to which the sample is assigned and $P(\omega_i|\mathbf{x})$ is the a posteriori probability of the data belonging to class ω_i given the measurement. This rules results in the minimum possible error rate which is called the Bayes error. Unfortunately, even when the class density distributions are known, calculating the Bayes error is very difficult as it requires the integration of the regions of the class density functions which intersect each other. Usually the class density distributions are not known anyway and this has motivated other approaches to error estimation.

The common approach for estimating the error is to measure the rate of success of the classifier when applied to a set of test data, but selecting a representative test set which accounts for the natural variability in the data and effects of noise is difficult for vision problems. Also, in some circumstances the amount of data available for testing may be limited resulting in poor estimates of the error. An improved method for error estimation has been proposed by Haralick [Haralick 92] in which models are constructed to describe the data to be classified and to describe the variability of the data. Test examples can then be drawn at random from these models and providing these models are representative of the data to be classified, good estimates of the error rate can be determined. A criticism of this approach is that real data rarely conforms to the types of models which are adopted.

4.3 Classification Error for Many Classes

The methods already discussed for classification error estimation may be appropriate when the number of classes is small but they become impractical as the number of classes is increased. These methods also say nothing about how the reliability scales as the number of classes is increased. This has motivated the development of an alternative approach to error estimation which makes the relationship between the error rate and the number of model classes explicit.

Before proceeding there are some important observations which should be made about this particular classification problem as these have moulded the approach taken. First of all, the domain of all shape classes is not a discrete space but describes a continuous and infinite variety of possible shape. Discrete classes are imposed on this space by the set of training examples, but these are simply a sample taken from the distribution of all possible shape. It is useful to adopt the concept of a *prior* density distribution [Fukunaga & Flick 84] when considering a large number of classes under these circumstances. This prior density reflects the fact that some shape classes are more likely to be encountered than others and that some configurations are unlikely to be encountered at all. The prior density distribution $p_{prior}(\mathbf{x})$ describes the probability of drawing a random pairwise geometric histogram or pattern $X = \mathbf{x}$, within a small range $\Delta\mathbf{x}$.

$$P(X = \mathbf{x}) = p_{prior}(\mathbf{x})\Delta\mathbf{x} \tag{4.2}$$

The situation, then, is that we have some continuous domain of patterns and impose N classes $\{\omega_i : 1 \leq i \leq N\}$ on this domain by defining a set of exemplars $\{\overline{\mathbf{x}}_i : 1 \leq i \leq N\}$. Given a previously unseen pattern, $\mathbf{x}$, we would like to estimate the probability, ε, that the pattern is incorrectly classified and relate this to the number of classes N.

We begin by considering a pair of classes ω_i and ω_j represented by class exemplars $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{x}}_j$ and a sample $\mathbf{x}_i$ drawn from class ω_i. For a nearest-neighbour classifier an error occurs when $\mathbf{x}_i$ is closer to the wrong class exemplar $\overline{\mathbf{x}}_j$ than it is to its own class exemplar $\overline{\mathbf{x}}_i$. This is the pairwise error $\varepsilon_p(i,j)$.

$$\varepsilon_p(i,j) = P(|\mathbf{x}_i - \overline{\mathbf{x}}_i| > |\mathbf{x}_i - \overline{\mathbf{x}}_j|) \tag{4.3}$$

where the notation $|\cdot|$ is used to represent the distance between a pair of patterns defined by the appropriate distance metric. As already discussed in Chapter 3 the classification of pairwise geometric histograms is based on the Bhattacharyya distance.

$$|\mathbf{m} - \mathbf{s}| = \sum_i \sqrt{m_i}\sqrt{s_i} \tag{4.4}$$

where $\mathbf{m}$ represents a model pairwise geometric histogram and $\mathbf{s}$ represents the pairwise geometric histogram constructed for a scene line primitive. The mean pairwise error ε_p is determined by averaging over all of the classes.

$$\varepsilon_p = \frac{1}{N(N-1)} \sum_i^N \sum_{j,j\neq i}^N \varepsilon_p(i,j) \tag{4.5}$$

The mean pairwise error, ε_p, is the probability that a sample pattern will be closer to a randomly selected class exemplar than it is to its own class exemplar. Conversely, the probability that a sample pattern will be closer to its own exemplar than a randomly selected one is by definition $1 - \varepsilon_p$. If there are a total of N stored exemplars then the sample pattern will only be classified correctly if it is closer to its own exemplar than it is to any of the other $N-1$ class exemplars. Assuming that class exemplars are independently sampled from the distribution of patterns, then the probability that the test pattern is correctly classified, P, is simply:

$$P = (1-\varepsilon_p)^{N-1} \tag{4.6}$$

If P is the probability of the test pattern being classified correctly then the classification error, ε is simply $1-P$.

$$\varepsilon = 1-(1-\varepsilon_p)^{N-1} \tag{4.7}$$

This expression for ε, which is exponential with respect to N, suggests that even when the pairwise error is small the classification error can become large when the number of stored exemplars, N, is large. One way in which the classification error can be improved when the number of classes is large is to propose a number of likely classes for each test pattern based on the nearest n neighbours. This is only useful if the extra, incorrect classes can be discarded later, which for shape data can be done by finding consistency amongst the classes assigned to a group of shape primitives. In fact, this is done already in the recognition algorithm by the probabilistic Hough transform as only consistent primitive classifications contribute to the winning Hough space. This is in contrast to Fukunaga's suggestion that classes are grouped to reduce the classification error when the number of classes is very large [Fukunaga & Flick 84].

Typically the number of classes that should be proposed for each test pattern so that it is likely that the correct class is included is the expected number of pairwise misclassifications, $(N-1)\varepsilon_p$. More specifically, if the number of proposed classes is n, then the probability that none of those classes will be the correct class is described as a sum of binomial terms. This is derived in Appendix C.

$$\varepsilon = 1-\sum_{i=0}^{n-1}(1-\varepsilon_p)^{N-1-i}\varepsilon_p^i \begin{pmatrix} N-1 \\ i \end{pmatrix} \tag{4.8}$$

This allows the user of the algorithm to select n according to the data to be classified to obtain the required classification error, at the expense of having to find a single, correct class later.

4.3.1 Estimation of the Mean Pairwise Error

So far it has been suggested that the classification error ε can be determined from the mean pairwise error ε_p but nothing has been said about how this is determined. It is shown here that the mean pairwise error can be estimated from the distribution of distances between class exemplars and the distribution of distances between class exemplars and samples drawn from those classes.

The typical distance between patterns drawn at random from $p_{prior}(\mathbf{x})$ gives an indication of the difficulty of a particular classification problem. This can be characterised by the interclass density distribution $p_{inter}(d)$ which gives the probability that a pair of patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ drawn at random are at a distance d from each other, within some small range Δd.

$$P(|\mathbf{x}_i - \mathbf{x}_j| = d) = p_{inter}(d)\Delta d \tag{4.9}$$

This is shown in Figure 4.1 for clarity. The term interclass distance density distribution is used because class exemplars are treated as random samples from the prior density distribution.

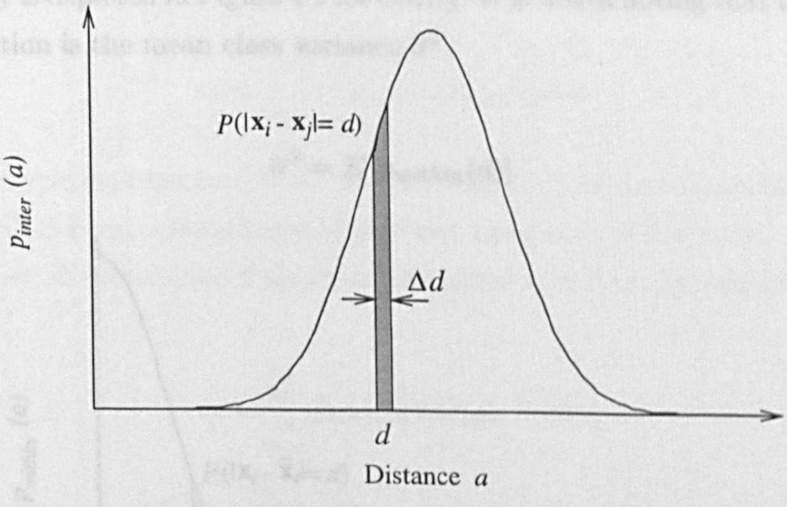


Figure 4.1: Calculating the probability that two random samples $\mathbf{x}_i$ and $\mathbf{x}_j$ will be separated by a distance d using the interclass density distribution $p_{inter}(a)$.

It is also convenient to define the cumulative, interclass density function, $P_{inter}(d)$, where:

$$P_{inter}(d) = \int_0^d p_{inter}(a)da \tag{4.10}$$

The probability that a pair of randomly selected patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ are less than a distance

d from each other is given directly by the cumulative distribution.

$$P(|\mathbf{x}_i - \mathbf{x}_j| < d) = P_{inter}(d) \tag{4.11}$$

Given a set of exemplar training patterns, $\{\overline{\mathbf{x}}_i : 1 \leq i \leq N\}$, an estimate of the cumulative interclass distance density function $\hat{P}_{inter}(d)$ can be determined by recording the distance between each pair of class exemplars as a frequency distribution and then normalising.

The typical distance between a class exemplar and patterns drawn from that class also gives an indication of the difficulty of the classification problem. For the time being gross effects such as scene clutter and missing data are ignored so that this distance reflects variability introduced by the different stages of the recognition system. The statistics of this distance can be characterised by the mean within-class density distribution $p_{within}(d)$ which gives the probability that pattern $\mathbf{x}_i$ drawn at random from a class ω_i also drawn at random is at a distance d from the class exemplar $\overline{\mathbf{x}}_i$.

$$P(|\overline{\mathbf{x}}_i - \mathbf{x}_i| = d) = p_{within}(d)\Delta d \tag{4.12}$$

This probability is depicted in Figure 4.2 for clarity. It is worth noting that the expectation of this distribution is the mean class variance $\overline{\sigma}^2$.

$$\overline{\sigma}^2 = E\left[p_{within}(d)\right] \tag{4.13}$$

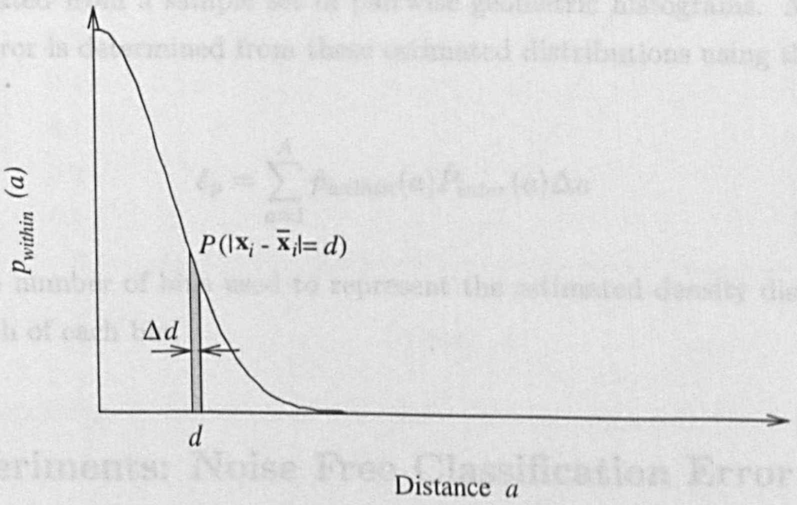


Figure 4.2: Calculating the probability that a pattern $\mathbf{x}_i$ drawn at random and the class exemplar $\overline{\mathbf{x}}_i$ will be separated by a distance d using the within-class distance distribution $p_{within}(a)$.

As with the interclass distance density distribution, an estimate of the mean within-class distance density $\hat{p}_{within}(d)$ can be determined from a number of samples.

It is now possible to derive an expression for the mean pairwise error ε_p. Recall from earlier that.

$$\varepsilon_p = P(|\mathbf{x}_i - \overline{\mathbf{x}}_j| < |\mathbf{x}_i - \overline{\mathbf{x}}_i|) \tag{4.14}$$

That is, a pairwise error occurs if the distance from a sample $\mathbf{x}_i$ to the wrong class exemplar $\overline{\mathbf{x}}_j$ is less than the distance to the correct exemplar $\overline{\mathbf{x}}_i$. This can be written as a joint probability integrated over a marginal variable a.

$$\begin{aligned} \varepsilon_p &= \int_0^\infty P(|\mathbf{x}_i - \overline{\mathbf{x}}_j| < a, |\mathbf{x}_i - \overline{\mathbf{x}}_i| = a)da & (4.15)\\ &= \int_0^\infty P(|\mathbf{x}_i - \overline{\mathbf{x}}_j| < a)P(|\mathbf{x}_i - \overline{\mathbf{x}}_i| = a)da & (4.16) \end{aligned}$$

Then, by substituting in Expressions 4.11 and 4.12, an expression for ε_p in terms of the interclass and within-class density distributions is obtained.

$$\varepsilon_p = \int_0^\infty p_{within}(a)P_{inter}(a)da \tag{4.17}$$

In practice, analytic expressions for the density distributions are unavailable and instead they are estimated from a sample set of pairwise geometric histograms. An estimate of the pairwise error is determined from these estimated distributions using the expression:

$$\hat{\varepsilon}_p = \sum_{a=1}^{A} \hat{p}_{within}(a)\hat{P}_{inter}(a)\Delta a \tag{4.18}$$

where A is the number of bins used to represent the estimated density distributions and Δa is the width of each bin.

4.4 Experiments: Noise Free Classification Error

The classification error for shape data represented by pairwise geometric histograms is investigated here using the analysis discussed above. This has been done for two sets of shape data, the first being shape outlines and the second being views of mechanical parts.

Both sets can be found in Appendix A, the outline data in Figures A.1 and A.2 and the mechanical data in Figure A.3. Each set of shape data was used to construct exemplar geometric histograms, each line primitive defining a different class, and these were then used to construct the respective cumulative interclass distance distributions. This data provides only a single example for each class but to construct the within-class distance distributions a number of example shape primitives from each class are needed. To obtain this additional data, 99 examples were generated for a few selected classes by applying random rotations and translations to the original image data for a number of models, and then approximating the new images by line segments. Line segments were clustered into groups of 100 based upon proximity of their geometric histograms which should group them according to class and the within-class distance distributions generated.

4.4.1 Shape Outlines

The outline shape data found in Appendix A, Figures A.1 and A.2, has been used to construct a database of 906 exemplar pairwise geometric histograms of type *directed* with a resolution of 40 distance bins and 64 angle bins. Figure 4.3 shows the normalised, cumulative interclass distance distribution constructed for this data. The distribution has been constructed with 100 bins over the range of distance from 0.0 to 1.0.

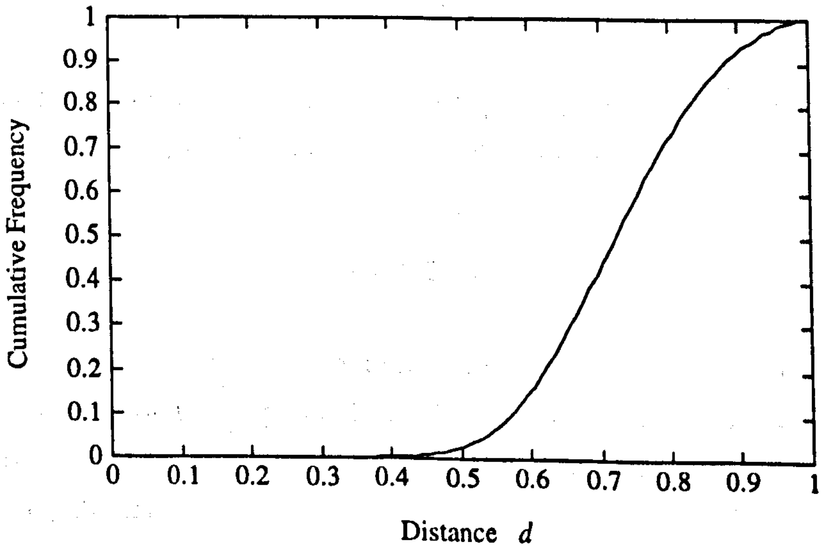


Figure 4.3: Cumulative distribution of interclass distance, $\hat{P}_{inter}(d)$, for the shape outline data.

Figure 4.4 shows the normalised, within-class distance distribution constructed for a selection of line primitives taken from the outline data set. Again, the distribution has been constructed with 100 bins over the range of distance from 0.0 to 1.0.

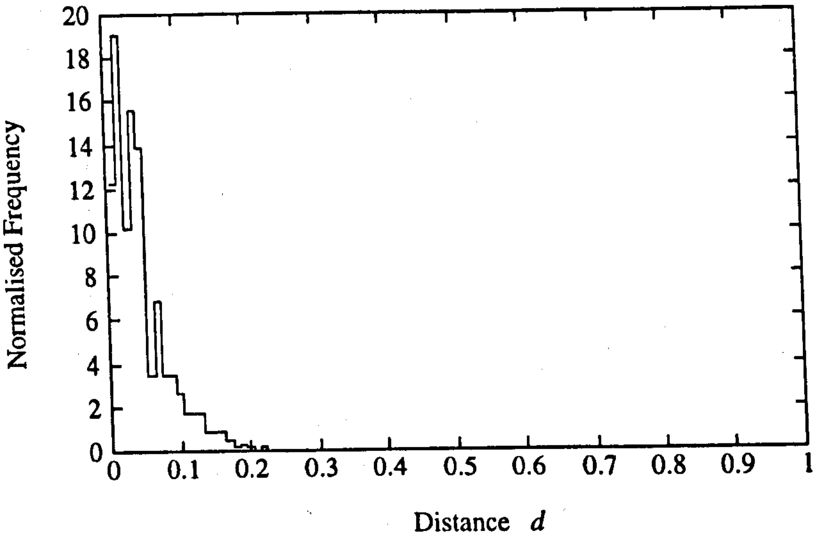


Figure 4.4: Distribution of within-class distance, $\hat{p}_{within}(d)$, for a selection of line primitives taken from the shape outline data.

Using Expression 4.18, these distributions have been used to estimate the probability, ε_p, that a scene geometric histogram will be closer to an exemplar geometric histogram selected at random from the training set than it is to its own class exemplar.

$$\hat{\varepsilon}_p = 3.029 \times 10^{-5} \tag{4.19}$$

For the database of 906 exemplar geometric histograms, and when classification is based on the nearest neighbour ($n = 1$), this gives a classification error of:

$$\varepsilon = 2.71\% \tag{4.20}$$

Figure 4.5 presents the error rate as a function of the number of stored exemplars for different values of n. For this type of shape data a classification error of less than 0.5% can be expected with tens times as many stored models when classification is based on the nearest 3 neighbours.

4.4.2 Mechanical Parts

The outline shape data found in Appendix A, Figure A.3 has been used to construct a database of 449 exemplar pairwise geometric histograms of type *directed* with a resolution of 40 distance bins and 64 angle bins. Figure 4.6 shows the normalised, cumulative inter-class distance distribution constructed for this data. As in the previous experiment, the

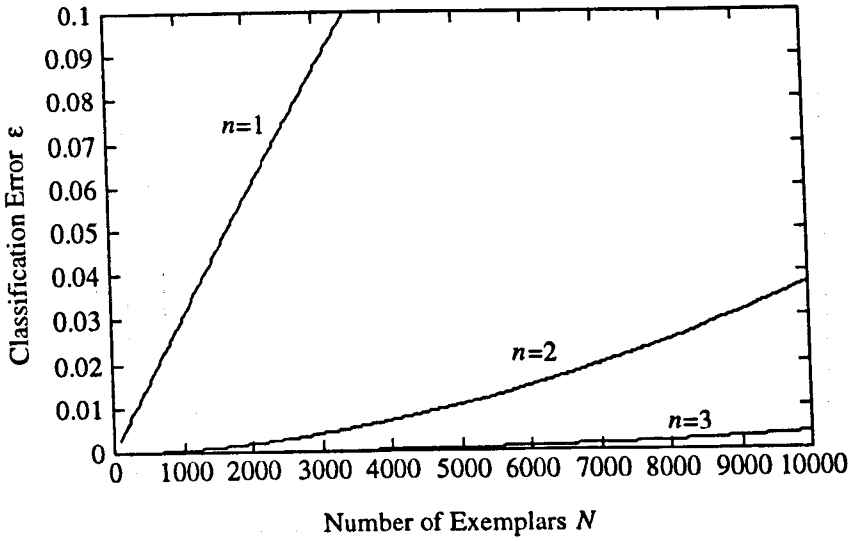


Figure 4.5: The classification error for the outline data as a function of the number of exemplar histograms for different values of n.

distribution has been constructed with 100 bins over the range of distance from 0.0 to 1.0.

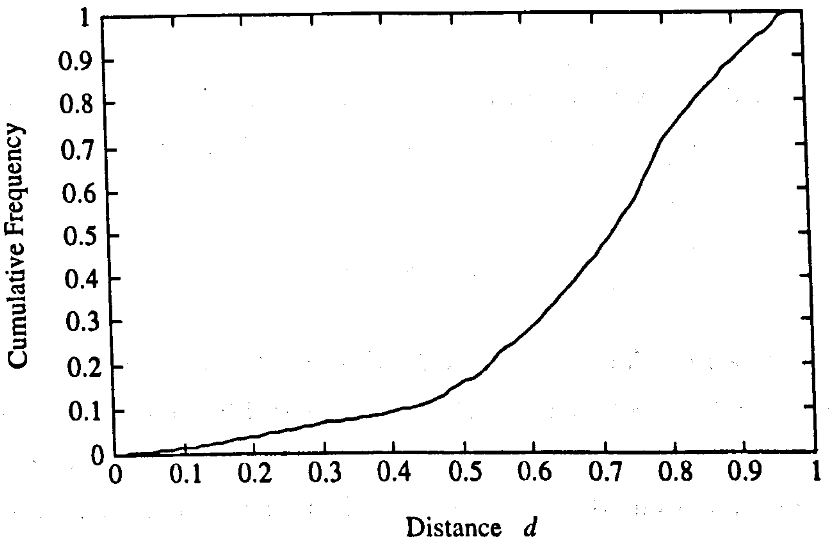


Figure 4.6: Cumulative distribution of interclass distance, $\hat{P}_{inter}(d)$, for the mechanical part data.

Figure 4.7 shows the normalised, within-class distance distribution constructed for a selection of line primitives taken from the mechanical part data set. Again, the distribution has been constructed with 100 bins over the range of distance from 0.0 to 1.0.

These distributions have been used to measure the probability, ε_p, that a scene geometric histogram will be closer to an exemplar geometric histogram selected at random from the training set than it is to its own class exemplar using Expression 4.18.

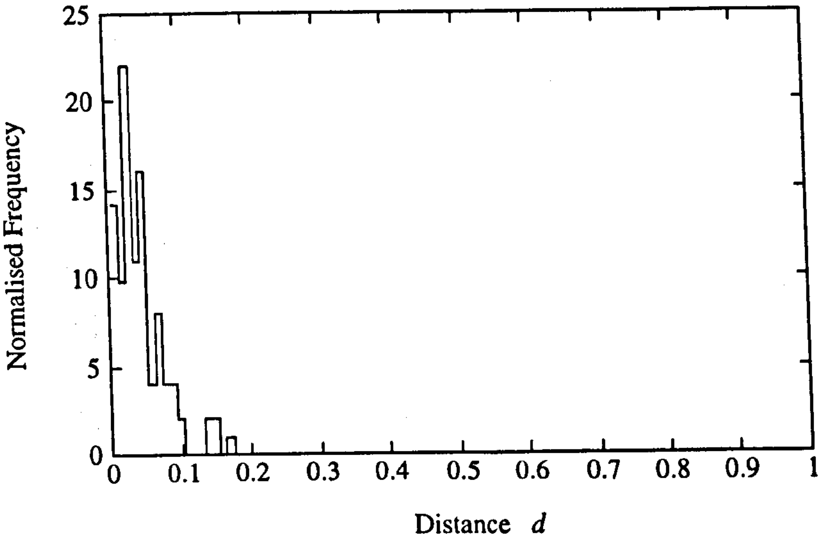


Figure 4.7: Distribution of within-class distance, $\hat{p}_{within}(d)$, for a selection of line primitives taken from the mechanical part data set.

$$\hat{\varepsilon}_p = 5.142 \times 10^{-3} \tag{4.21}$$

For the database of 449 exemplar geometric histograms, and when classification is based on the nearest neighbour ($n = 1$), this gives a classification error of:

$$\varepsilon = 90.1\% \tag{4.22}$$

Figure 4.8 presents the error rate as a function of the number of stored exemplars for different values of n. The reliability of classification is much worse for this data than it is for the shape outline data. To obtain a classification error of less than 0.5% for around 5000 stored exemplars more than 40 of the nearest neighbours must be used. The main reason for this apparently poor performance is that many of the features in the mechanical shape data are very similar, resulting in ambiguities. For example, all of the line segments describing a circular shape, of which there are several in the mechanical part database, are represented by identical geometric histograms because of shape symmetry. It should be noted that this large classification error does not necessarily result in poor object detection and localisation, as is apparent from the recognition results in the previous chapter. Good overall performance is still attained provided that ambiguous features impose the same constraint on the object pose. The issue of feature ambiguity is studied in more detail in Chapter 5.

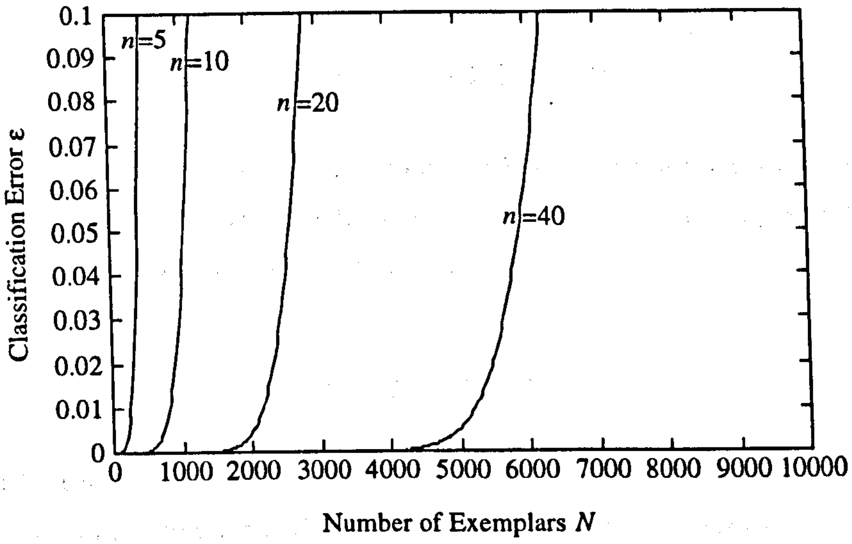


Figure 4.8: The classification error for the mechanical part data as a function of the number of exemplar histograms for different values of n.

4.5 Classification Error in Noise

So far the analysis of classification error has assumed that patterns drawn from some class will be centred around the class exemplar and vary from that exemplar according to some statistical distribution which reflects variability in the construction of the pattern. This variability was measured in the form of a within-class distance distribution and used to estimate the classification error. For well constrained problems this level of analysis is sufficient but for many problems, including the classification of shape in real scenes, patterns can move much further from the class exemplar than the within-class variability would suggest because of incomplete or contaminated data. Although data may become incomplete or contaminated for a whole host of different reasons the general term used to describe these factors is *noise.*

If the distance moved by patterns from their class exemplars in noisy data is significantly greater than suggested by the within-class variability then the calculation of classification error based on the earlier analysis will give an under estimate. This has motivated the development of an alternative way of estimating the classification error for noisy data.

Instead of quantifying the distance moved by patterns from their exemplars using a frequency distribution of distance, the typical (or mean) distance moved in noisy data is now used. If the mean distance moved by a pattern, $\mathbf{x}_i$, from its class exemplar, $\overline{\mathbf{x}}_i$, is ΔD then the probability that $\mathbf{x}_i$ is nearer to a randomly selected class exemplar, $\overline{\mathbf{x}}_j$ (this is the pairwise error, $\varepsilon_p(i,j)$) is simply:

$$\varepsilon_p(i,j) = P(|\mathbf{x}_i - \overline{\mathbf{x}}_j| < \Delta D) \tag{4.23}$$

The mean pairwise error ε_p over all of the classes is simply the area under the interclass distance distribution below ΔD, see Expression 4.9. This can be expressed in terms of the cumulative interclass distance distribution.

$$\varepsilon_p = P_{inter}(\Delta D) \tag{4.24}$$

In other words, the cumulative interclass distance distribution can be interpreted as the pairwise error as a function of the distance patterns move from their exemplars because of noise, see Figure 4.9. This new estimate of the pairwise error can then be used as before to estimate the classification error, ε, as in Expressions 4.7 and 4.8.

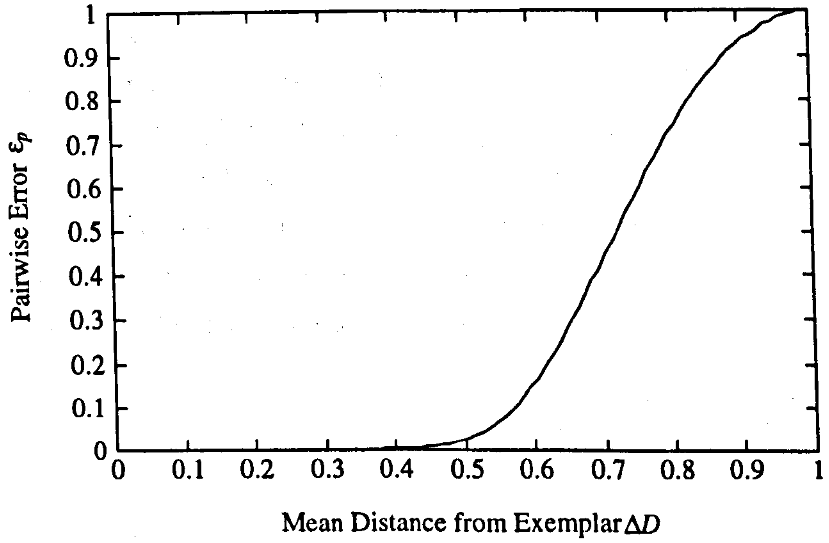


Figure 4.9: The cumulative interclass distance distribution can be interpreted as the pairwise error as a function of the distance moved by patterns from their class exemplars due to noise.

4.6 A Noise Model for Shape Data

Rather than measuring the typical distance that a geometric histogram moves from its class exemplar from a selection of noisy scenes a model of this movement is developed here for different sources of noise. This is advantageous because it allows the classification error to be predicted for arbitrary scenes given an estimate of the magnitude of the noise in the scene. To test the validity of this model, the distance that patterns move as predicted by the model has been compared to the distance moved in scenes containing known levels of simulated noise, and it has been found that the model provides an upper bound.

The two major sources of noise in the construction of geometric histograms investigated here are missing line data and scene clutter. Shape data can be missing for a variety of reasons. If sections of objects are poorly lit, in shadow, badly focussed or if the level of pixel noise is high then boundaries may not be detected. More seriously, whole sections of a shape can be missing if it is occluded by other objects. Except under constrained viewing conditions clutter is an integral part of any scene and unless this is removed by a segmentation strategy it will contaminate geometric histograms constructed for shape primitives in its vicinity.

4.6.1 The Effect of Missing Data

To predict the effect of missing line data on the classification process an expression relating the distance moved by patterns, ΔD, as a function of the proportion of missing data, m, is derived. A normalised, exemplar histogram, $\hat{H} = [\hat{h}_0, \hat{h}_1, ..., \hat{h}_{N-1}]$, constructed from some arbitrary shape data is defined. A second normalised histogram, $\hat{M} = [\hat{m}_0, \hat{m}_1, ..., \hat{m}_{N-1}]$, is then defined which is constructed from the same shape data as $\hat{H}$ except that a proportion, m, of each line of the shape has been removed. This distance between $\hat{H}$ and $\hat{M}$ is then evaluated using the distance metric:

$$\Delta D(m) = 1 - \sum_i \sqrt{\hat{h}_i}\sqrt{\hat{m}_i} \tag{4.25}$$

As the shape data represented by $\hat{M}$ is a subset of the shape data represented by $\hat{H}$, the entries in $\hat{M}$ must be a subset of the entries in $\hat{H}$ which have been rescaled to maintain normalisation. The distance metric only depends upon elements of $\hat{M}$ and $\hat{H}$ which are *both* non-zero and $\hat{M}$ is a subset of $\hat{H}$. The distance can therefore be expressed in terms of $\hat{M}$ and the scaling factor, s, alone.

$$\begin{aligned} \Delta D(m) &= 1 - \sum_i \sqrt{s\hat{m}_i}\sqrt{\hat{m}_i} && (4.26) \\ &= 1 - \sqrt{s} && (4.27) \end{aligned}$$

Where s is the difference in scale between $\hat{H}$ and $\hat{M}$.

$$s = \frac{\hat{h}_i}{\hat{m}_i} \qquad \text{for } \hat{m}_i \neq 0 \tag{4.28}$$

This scaling factor, s, can be determined from the difference in the height of a single geometric histogram entry made for a pair of lines before and after a proportion of the line data is removed. Figure 4.10 depicts a single histogram entry for a reference line of length l_0 compared to another line of length l_j and then the same entry when a proportion of the line data has been removed.

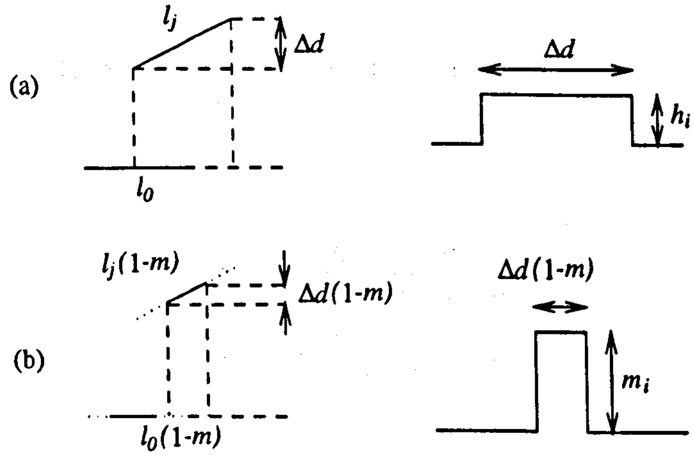


Figure 4.10: The effect of missing line data on a single histogram entry. (a) The entry made for the original line data. (b) The entry made for the line data after a proportion m of each line is removed.

For the original line data the entry is weighted by the product of the lengths of the two lines, $l_0 l_j$, and normalised by the total entries made into H which is given by the expression:

$$\sum_i H_i = \sum_j l_0 l_j \tag{4.29}$$

$$= l_0 l \tag{4.30}$$

Where l is the total length of lines in the shape. If the width of the entry is Δd then the height, $\hat{h}_i$, is simply (there should also be a term relating to the width of the entry along the angle axis but this falls out and so is ignored):

$$\hat{h}_i = \frac{l_0 l_j}{l_0 l \Delta d} \tag{4.31}$$

$$= \frac{l_j}{l \Delta d} \tag{4.32}$$

For the shortened line data, the entry is weighted by the product of the lengths of the two shortened lines, $l_0 l_j(1-m)^2$, and normalised by the total entries made into M which is given by the expression:

$$\begin{aligned} \sum_i H_i &= \sum_j l_0 l_j (1-m)^2 && (4.33) \\ &= l_0 l (1-m)^2 && (4.34) \end{aligned}$$

The width of the shortened entry is now $\Delta d(1-m)$ which means that the height of the entry, $\hat{m}_i$, is:

$$\begin{aligned} \hat{m}_i &= \frac{l_0 l_j (1-m)^2}{l_0 l (1-m)^2 \Delta d (1-m)} && (4.35) \\ &= \frac{l_j}{l \Delta d (1-m)} && (4.36) \end{aligned}$$

The scaling factor s can now be determined by substituting the expressions for $\hat{h}_i$ and $\hat{m}_i$ into Expression 4.28.

$$s = 1 - m \qquad (4.37)$$

The relationship between ΔD and m is then determined by substituting s in equation 4.27:

$$\Delta D(m) = 1 - \sqrt{1-m} \qquad (4.38)$$

To demonstrate this effect, 5 of the models taken from the data set in Appendix A have been matched with an increasing proportion of line data removed and the mean distance moved by all of the geometric histograms for each shape recorded. The experimental results are given in Figure 4.11 along with the theoretical expression shown by the solid line. The reason why the distance moved in practice is smaller than predicted is because quantisation of the entries into bins results in a greater overlap than the model would suggest. This means that the model can be used as an upper bound on ΔD. One of the model shapes used in this experiment is depicted in Figure 4.12 with 25% and then 75% of the line data removed.

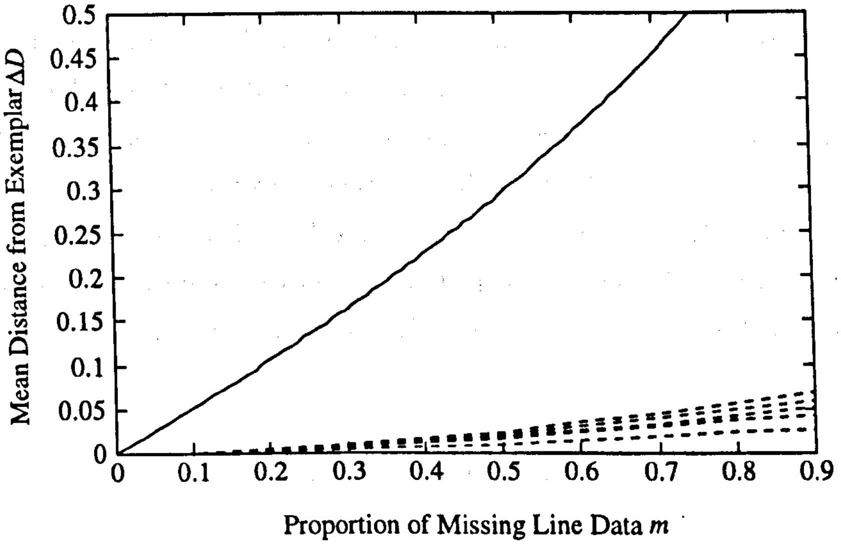


Figure 4.11: The distance moved by patterns from their class exemplars as a function of the proportion of missing data m. The solid line is the distance predicted by the model and the dotted lines are the mean distances moved for all of the geometric histograms for each of 5 shapes taken from the training set.

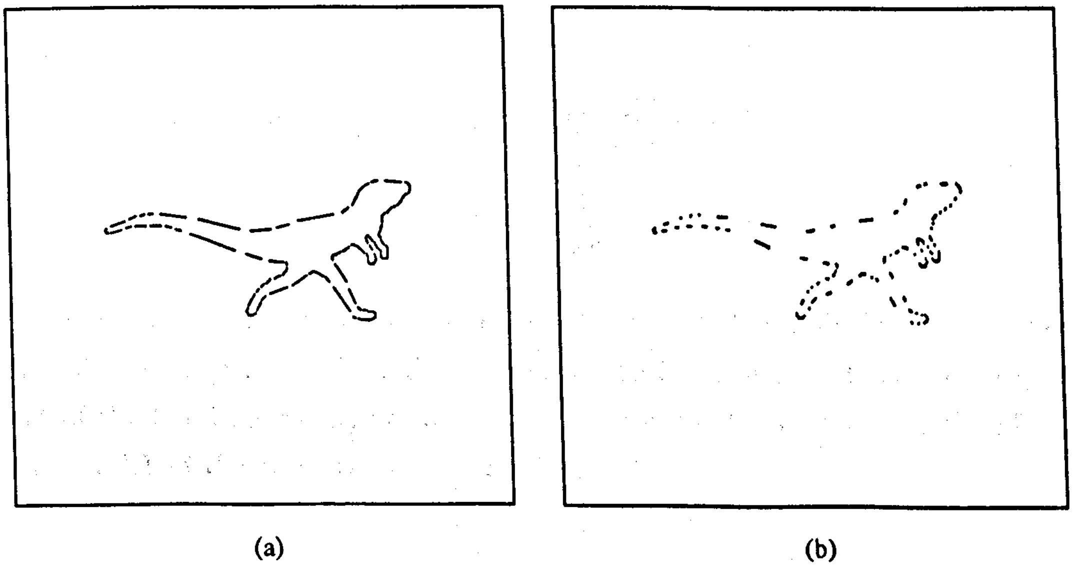


Figure 4.12: An example of model shape data with (a) 25% of each line primitive removed, and (b) 75% of each line primitive removed.

4.6.2 The Effect of Scene Clutter

To predict the effect of scene clutter on the classification process an expression relating the distance moved by patterns from their exemplars, ΔD, as a function of the proportion of scene clutter, c, is derived. A normalised exemplar histogram, $\hat{H} = [\hat{h}_0, \hat{h}_1, ..., \hat{h}_{N-1}]$, constructed from some arbitrary shape data is defined. A second normalised histogram, $\hat{C} = [\hat{c}_0, \hat{c}_1, ..., \hat{c}_{N-1}]$, is then defined which is constructed from the same shape data as $\hat{H}$ except that a proportion, c, of the length of lines in the model are randomly added as clutter. The distance between $\hat{H}$ and $\hat{C}$ is then evaluated using the distance metric:

$$\Delta D(c) = 1 - \sum_i \sqrt{\hat{h}_i}\sqrt{\hat{c}_i} \qquad (4.39)$$

If the entries added to $\hat{H}$ for the clutter data to produce $\hat{C}$ are assumed not to correlate with any of the original entries in $\hat{H}$ then correlated entries between $\hat{H}$ and $\hat{C}$ are related the scale factor needed to normalise $\hat{C}$.

$$s = \frac{\hat{h}_i}{\hat{c}_i} \qquad \text{for } \hat{h}_i \neq 0 \qquad (4.40)$$

$\hat{H}$ is a subset of the data in $\hat{C}$ so the distance between them can be determined in terms of $\hat{H}$ and the scale factor.

$$\begin{aligned} \Delta D(c) &= 1 - \sum_i \sqrt{\hat{h}_i}\sqrt{\frac{\hat{h}_i}{s}} & (4.41) \\ &= 1 - \frac{1}{\sqrt{s}} & (4.42) \end{aligned}$$

The scale factor can be determined, as before, by considering a single entry made for a reference line of length l_0 and another line of length l_j. The weight for the entry is the product of the line lengths $l_0 l_j$ normalised by the total entries made into the histogram $l_0 l$. If the width of the entry is Δd then the height $\hat{h}_i$ is:

$$\hat{h}_i = \frac{l_0 l_j}{l_0 l \Delta d} \qquad (4.43)$$

Similarly, for the cluttered data the entry is weighted by the product of the lengths of the two lines $l_0 l_j$ normalised by the total entries made into the histogram, $l_0 l(1 + c)$. The height of this entry, $\hat{c}_i$, still of width Δd is simply:

$$\hat{c}_i = \frac{l_0 l_j}{l_0 l(1+c)\Delta d} \tag{4.44}$$

The scale factor can now be determined by substituting the expressions for $\hat{h}_i$ and $\hat{c}_i$ into Equation 4.40.

$$s = 1 + c \tag{4.45}$$

The relationship between ΔD and c is then determined by substituting s in Equation 4.42:

$$\Delta D(c) = 1 - \frac{1}{\sqrt{1+c}} \tag{4.46}$$

To demonstrate this effect, 5 of the models taken from the data set in Appendix A have been matched with an increasing proportion of scene clutter added and the mean distance moved by all of the geometric histograms for each model recorded. The experimental results are given in Figure 4.13 as dotted lines along with the predicted distance shown by the solid line. The reason why the prediction is always greater than the measured distances is that some of the added clutter correlates with the original histogram entries increasing the similarity. This means that the model prediction can be used as an upper bound in the distance moved by patterns because of clutter. One of the model shapes used in this experiment is depicted in Figure 4.14 with as much clutter as there is model data and with twice as much clutter than there is model data.

This model of the distance moved by geometric histograms as a function of the level of missing data and scene clutter can be used, along with the cumulative interclass distance distribution, to calculate the classification error for noisy scenes. If the mean levels of missing data and scene clutter are m and c respectively then the mean distance moved by input patterns, ΔD, is described by the sum of the models.

$$\Delta D = 2 - \sqrt{1-m} - \frac{1}{\sqrt{1+c}} \tag{4.47}$$

The pairwise error, ε_p, can then be determined from the cumulative interclass distance distribution, $P_{inter}(\Delta D)$:

$$\varepsilon_p = P_{inter}\left(2 - \sqrt{1-m} - \frac{1}{\sqrt{1+c}}\right) \tag{4.48}$$

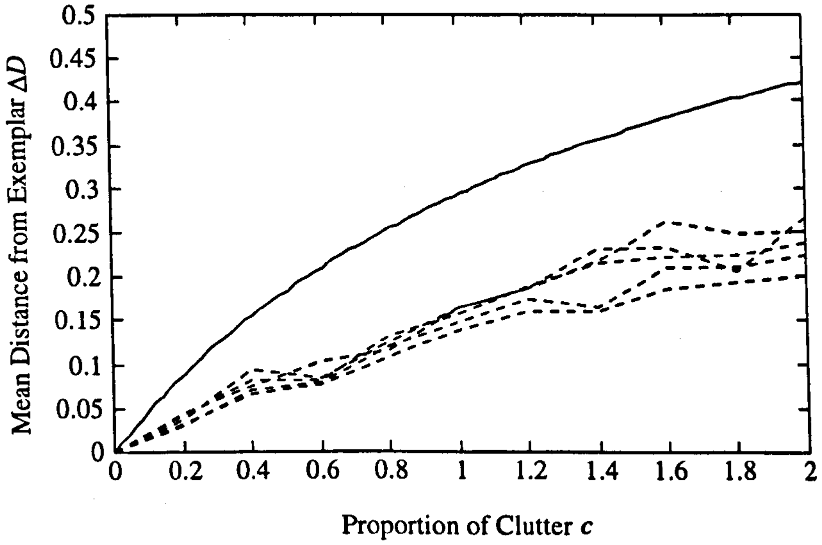


Figure 4.13: The distance moved by patterns from their class exemplars as a function of the proportion of added clutter data c. The solid line is the distance predicted by the model and the dotted lines are the mean distances moved for all of the geometric histograms for each of 5 shapes taken from the training set.

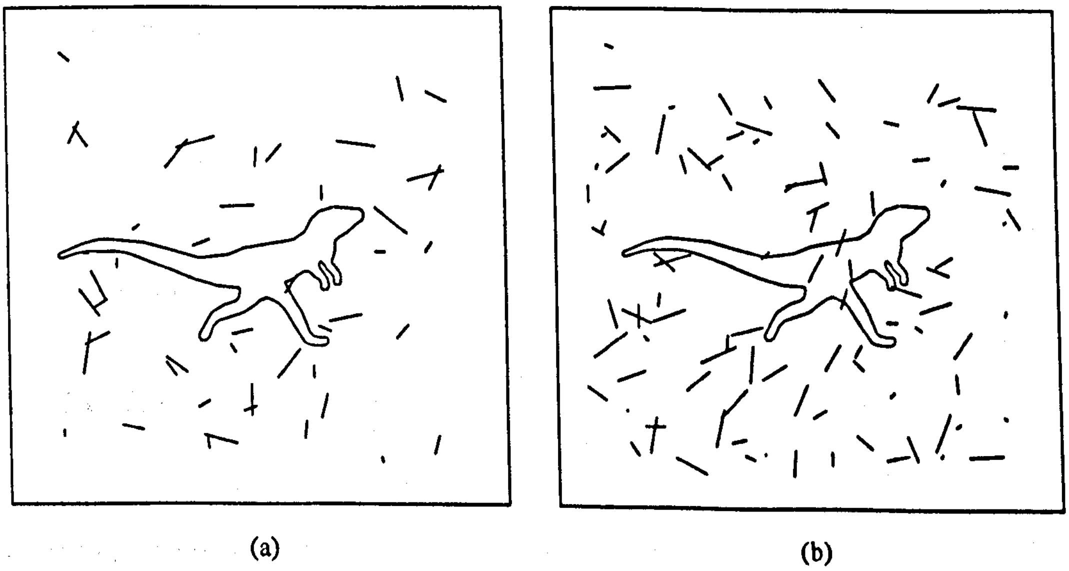


Figure 4.14: An example of model shape data with (a) the same amount of clutter as there is model, and (b) twice as much clutter as there is model.

4.7 Experiments: Classification in Noise

The classification error is estimated here for noisy scenes using the shape data used in the earlier, noise free, experiments. A reasonable mean distance between geometric histograms constructed for scene data and their correct class exemplars, ΔD, of 0.25 has been picked to represent realistic levels of scene noise. The noise model suggests that this would be the mean distance moved by patterns from their exemplars if about half of the shape data was missing or if there was as much scene clutter as there was shape data. Because the noise model gives an upper bound on the distance moved by patterns this value of ΔD probably represents much worse viewing conditions.

4.7.1 Shape Outlines

The cumulative interclass distance distribution constructed earlier for the shape outline data, see Figure 4.3, is interpreted as the pairwise error as a function of the mean distance moved by patterns from their exemplars. For a distance $\Delta D = 0.25$, the estimated pairwise error is:

$$\hat{\varepsilon}_p = 6.488 \times 10^{-4} \tag{4.49}$$

For the database of 906 exemplar geometric histograms, and when classification is based on the nearest neighbour ($n = 1$), this gives a classification error of:

$$\varepsilon = 44.4\% \tag{4.50}$$

Figure 4.15 presents the error rate as a function of the number of stored exemplars for different values of n.

4.7.2 Mechanical Parts

The cumulative interclass distance distribution constructed earlier for the mechanical part data, see Figure 4.6, is interpreted as the pairwise error as a function of the mean distance moved by patterns from their exemplars. For a distance $\Delta D = 0.25$, the estimated pairwise error is:

$$\hat{\varepsilon}_p = 5.046 \times 10^{-2} \tag{4.51}$$

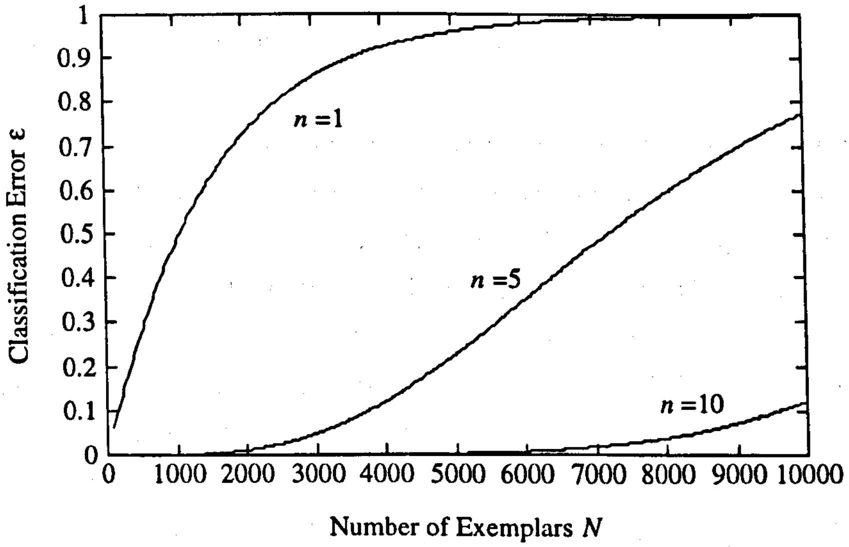


Figure 4.15: The classification error for the outline data as a function of the number of exemplar histograms for different values of n.

For the database of 449 exemplar geometric histograms, and when classification is based on the nearest neighbour ($n = 1$), this gives a classification error of:

$$\varepsilon = 99.9\% \tag{4.52}$$

Figure 4.16 presents the error rate as a function of the number of stored exemplars for different values of n.

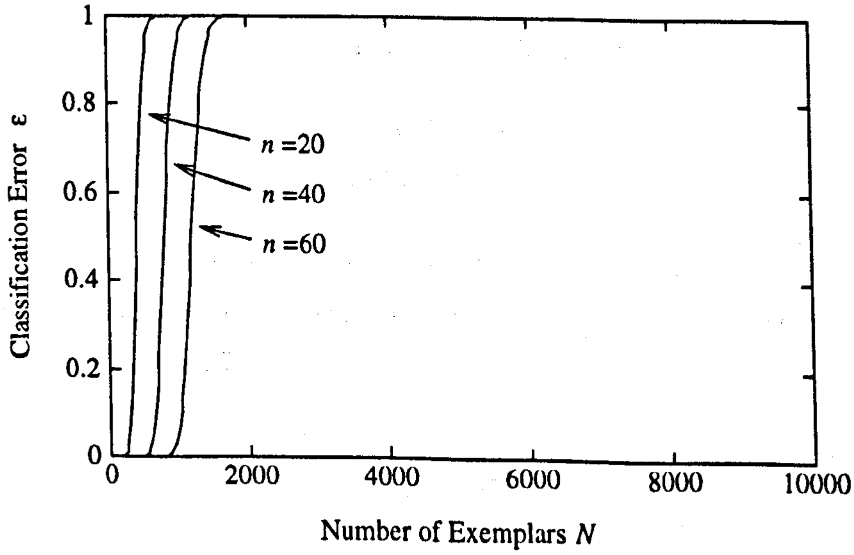


Figure 4.16: The classification error for the outline data as a function of the number of exemplar histograms for different values of n.

4.8 Optimum Selection of Parameters

It was mentioned in the introduction that one of the motivations for determining the classification error is to be able to select algorithm parameters in order to provide a specific level of performance. This is particularly useful when an appropriate value for a particular parameter cannot be determined in any other way. One such parameter used in the construction of pairwise geometric histograms is the histogram resolution along the perpendicular distance axis. To provide a more principled method for selecting an appropriate resolution, cumulative interclass distance distributions have been constructed for geometric histograms of increasing resolution. These can be seen in Figure 4.17 for geometric histograms with 10, 20, 40, 80, 160 and 320 bins along the perpendicular distance axis.

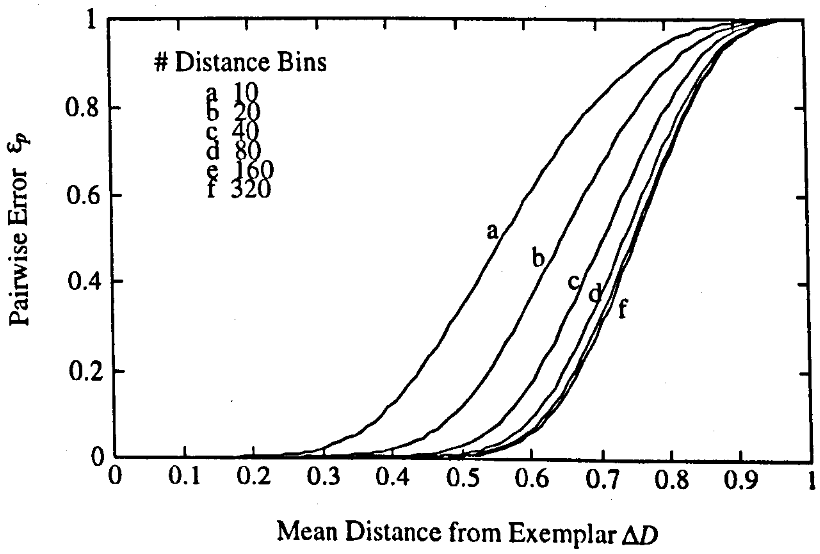


Figure 4.17: Cumulative interclass distance distributions constructed for geometric histograms with increasing perpendicular distance resolution.

Given a specified level of performance, in terms of the classification error ε and the number of stored exemplars N, the pairwise error which gives this performance can be determined be rearranging Expression 4.7.

$$\varepsilon_p = 1 - 10^{\frac{\log \varepsilon}{N}} \tag{4.53}$$

The histogram resolution which gives this performance for a specified level of scene noise can then be read directly from the curves in Figure 4.17. Intuitively the classification error cannot be continually improved by increasing the histogram resolution. This is confirmed by the data in Figure 4.17 where the curves converge to a minimum error. At this resolution all of the measurements entered into the geometric histogram are being accurately represented.

4.9 Conclusions

Error analysis of object recognition algorithms is essential if the research is to progress effectively and if the proposed solutions are to be successfully embedded into larger systems. A more general question, and one which is very relevant to object recognition in particular, concerns the relationship between reliability and the number of model classes that the system is expected to distinguish between. This argument motivated the analysis of the reliability of shape primitive classification when using pairwise geometric histograms. Conventional error estimation techniques do not make the relationship between the error and the number of (shape) classes explicit so an alternative approach has been proposed.

Central to this new approach is the idea, proposed by Fukunaga, that shape classes are effectively drawn at random from some *prior* density distribution. Although this distribution is never used directly it does mean that the distribution of distances between pairs of class exemplars has a characteristic shape and this *interclass* distribution can be used in estimating the error.

The main criticism of the approach is likely to be that the shape of the interclass density distribution depends to some extent on the shape data used. Compare Figures 4.3 and 4.6 for example which present the interclass density distribution for the two different datasets used in the experiments. There is clearly a difference in their shape. The intuitive counter argument would be that these distributions must have some uniformity in their shape, otherwise the performance of the algorithm would vary wildly from one shape to the next, but this is not found in practice. If a researcher demonstrates their algorithm working on some shape data then we accept that it is likely to also work on other similar shapes under the same conditions. In fact, simply observing distributions like the interclass density distribution which are central to the behaviour of the algorithm can provide significant insight into the performance of the algorithm even before any quantitative analysis is conducted. A more rigorous argument is provided in the next chapter where it is found that the shape of the interclass density distribution for small distances is related to the complexity of the shape data, or more specifically the intrinsic dimensionality of the data. It is also this part of the interclass density distribution which affects the estimated error.

One of the observations of the analysis was that the error increases exponentially with respect to the number of model shape classes. Even when the error rate for a small number of shape classes is very small this can change rapidly as the number of shape classes is increased. It was proposed that this problem can be avoided by assigning a number of the best classes to each scene primitive and then identifying the correct single class later by looking for consistency. Results presented here have shown that error rates better than 1% can be attained for large numbers of stored models when this strategy is adopted.

One of the uses of a quantitative measure of reliability is to enable algorithm parameters to be selected in order to give a specified level of performance. Most the parameters used in the construction of pairwise geometric histograms are well defined but an exception to this is the resolution along the perpendicular distance axis. To provide a principled means for selecting this parameter, its effect on reliability has been made explicit by plotting distance distributions for different resolutions. This allows the user to specify a required classification error and choose an appropriate histogram resolution accordingly.

Chapter 5

Estimating the Capacity of the Shape Representation

5.1 Introduction

In the previous chapter the reliability of the shape classification process and the way that this scales with the number of stored shape primitives were investigated. A second important issue regarding the scalability of the algorithm concerns the number of *distinguishable* shape primitives which can be described using pairwise geometric histograms without confusion. The term *capacity* is coined to describe this. Given a simple representation, for example the binary representation of integers, the number of different integers which can be described for a given number of bits can easily be determined by counting the number of binary combinations that can be formed. This is a straightforward problem because it is clear what is meant by a unique or distinguishable integer and also because we know that all combinations of binary digits represent allowable integers. Given that a geometric histogram can only be represented to some level of precision in practice, a naive approach for estimating the capacity might be to simply count the number of unique histograms that can be formed. This, of course, gives a very poor estimate which does not reflect any intuition about what is meant by distinguishable shape and also allows for instances of shape which are unlikely to be encountered. This example highlights two particular problems which must be addressed in order to obtain a meaningful estimate of capacity. First of all, how can the continuous domain in which geometric histograms exist be quantised in a sensible manner to represent unique patterns so that they can be counted? Secondly, what proportion of this domain represents the sorts of shapes that are likely to be encountered and can this be determined from a relatively small number of example shapes?

Intuitively there appears to be some relationship between capacity and reliability but at first the nature of this relationship may not be obvious. Certainly, if a representation with a small capacity is used to represent a large number of exemplars then the system will not be able to uniquely classify test data without error. The problem here is one of ambiguity and can only be avoided by proposing multiple hypotheses for the class of each test example. On a statistical level this ambiguity arises because class distributions overlap each other so that it is sometimes impossible to tell, with absolute certainty, from which class a particular test example may have been drawn. The capacity can be defined in this context as the number of classes that can be represented so that the uncertainty is below a specified, acceptable level.

The purpose of the work presented in this chapter is to estimate the number of shape primitives which may be described by the pairwise geometric representation without ambiguity and ultimately to demonstrate that this is not a limiting factor on the scalability of the algorithm. Two different approaches are taken, one based on geometric intuition which allows a lower bound on capacity to be estimated and the second based on a statistical model of the training process permitting a more precise estimate of the capacity itself to be determined.

The geometric approach is an improvement on the work of other researchers [Swain & Ballard 91, Stricker 94, Stricker & Swain 94] who have estimated the capacity of other representations. In this approach each stored exemplar is imagined to occupy some finite region of the pattern space and the storage capacity is defined to be the number of exemplars which, when tessellated, fully occupy the pattern space. On its own this is a grossly simplified view which inevitably results in a gross over-estimate of capacity because it is assumes that all regions of the pattern space represent shapes primitives that are likely to be encountered. The approach has been refined here by estimating the local dimensionality of the training data and calculating the number of unique patterns which can be stored along a manifold of this dimensionality embedded in the pattern space, which can be shown to give a lower bound on capacity.

The statistical approach treats the storage of exemplar patterns as a binning process where the domain of the representation is imagined to consist of a finite number of potential storage locations. If it is assumed that each location is as likely to be occupied as any other then the filling process can be described using Poisson statistics and an estimate of the capacity can be derived using this statistical model. The advantage of this approach is that no assumption is made about the distribution of the potential storage locations through the pattern space.

5.2 A Geometric Approach

The domain of all possible pairwise geometric histograms can be visualised as a high-dimensional and continuous space which, ignoring practical limitations on the precision to which numbers can be represented, describes an infinite variety of histogram patterns. As model geometric histograms are normalised they define unit vectors in this space which describe the surface of a hyper-sphere and it is this surface which defines the set of all possible histograms. In fact, the whole of the surface is not allowable because geometric histogram bins can only contain positive values as they represent frequencies. This limits the representational domain to lie on the positive quadrant of the hyper-sphere. Although, in principle, unique classes can be defined arbitrarily close on this surface, variability introduced by the various processes performed in the construction of a geometric histogram, such as image sampling and polygonisation, means that neighbouring classes may become indistinguishable. This variability was quantified in the last chapter for a given set of shapes by measuring the spread of the within-class distance distribution for a number of shape primitives from the same class. Figure 5.1 shows one of the within-class distribution previously constructed.

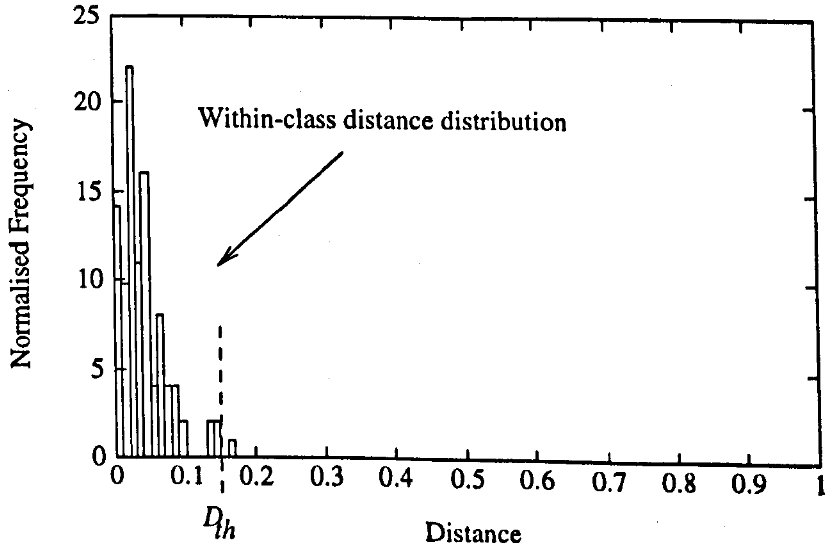


Figure 5.1: Within-class variability can be quantified by measuring the spread, D_{th}, of the within-class distance distribution.

The distance, D_{th}, which accounts for most of the spread of the data (about three standard deviations has been chosen here) is a measure of how far data from a given class can be expected to vary from the class mean. Provided that class centres are never less than twice this distance from each other then it is reasonable to say that those classes are distinct. In this way a class can be described as a patch on the surface of the hyper-sphere with a radius of D_{th} as depicted in Figure 5.2.

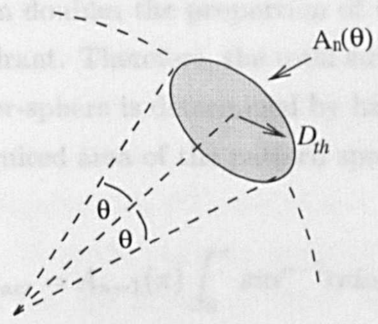


Figure 5.2: A surface patch describing a region of similar shape is defined by the radius D_{th} which is determine by measuring the typical variability of the data within a class.

A geometrically intuitive approach for estimating the capacity is to simply count the number of patches needed to cover the hyper-sphere's surface. This is similar to the approach used by Swain [Swain & Ballard 91] and by Stricker [Stricker 94, Stricker & Swain 94] to estimate the number of colour images which can be stored and individually indexed using colour histogram descriptors. If the patches defining unique classes are assumed to tessellate then the capacity, C, is simply the ratio of the total area[1] of the pattern space, A_{space}, to the area of a single patch, A_{class}.

It can be shown that the surface area, $A_n(\theta)$, of an n-dimensional hyper-spherical patch with unit radius, defined by a solid angle 2θ is given by the reduction formula (see Appendix C).

$$A_n(\theta) = A_{n-1}(\pi) \int_0^{\theta} sin^{n-2} \alpha d\alpha \qquad (5.1)$$

Where.

$$A_2(\theta) = 2\theta \qquad (5.2)$$

The total surface area of an n-dimensional hyper-sphere is determined by letting $\theta = \pi$.

$$A_n(\pi) = A_{n-1}(\pi) \int_0^{\pi} sin^{n-2} \alpha d\alpha \qquad (5.3)$$

[1] The term area is used because this is determined by integrating over a surface

This gives the total surface area but must be modified to give the required surface area of the positive quadrant. As the dimensionality of the space is increased an increasing proportion of the hyper-sphere's surface exists outside of the positive quadrant. Specifically, the addition of each dimension doubles the proportion of the hyper-surface's area which is not within the positive quadrant. Therefore, the total surface area of the positive quadrant of an n-dimensional hyper-sphere is determined by halving the area of the complete sphere n times, giving the required area of the pattern space, A_{space}.

$$A_{space} = A_{n-1}(\pi) \int_0^{\pi} sin^{n-2}\alpha d\alpha \ \frac{1}{2^n} \tag{5.4}$$

The surface patch of radius D_{th} is described by a solid angle of $\theta = acos(1 - D_{th})$, see Figure 5.2. The surface area of patch representing a distinct pattern class is then simply:

$$A_{class} = A_{n-1}(\pi) \int_0^{acos(1-D_{th})} sin^{n-2}\alpha d\alpha \tag{5.5}$$

The estimate of capacity is then given by the ratio:

$$C = \frac{\int_0^{\pi} sin^{n-2}\alpha d\alpha}{\int_0^{acos(1-D_{th})} sin^{n-2}\alpha d\alpha} \ \frac{1}{2^n} \tag{5.6}$$

5.2.1 Improved Estimate using Intrinsic Dimensionality

Although shape data is represented in a space whose dimensionality is spanned by the number of pairwise geometric histogram bins, the *intrinsic* or local dimensionality, n_i, of the data is significantly less than this. Consequently, the pattern space will never be fully populated. This reduction in dimensionality arises as a result of two different phenomena. Firstly, the histogram bins are not independent pieces of data but are highly correlated because of natural geometric correlations found in shapes which are a function of the complexity of the shape. Correlations also exist between bins because histogram entries are blurred by the error functions representing measurement variability. Secondly, not all areas of the pattern space correspond to shape data that is ever likely to be encountered so these areas will always be empty of exemplars.

The consequence of the local dimensionality of the histogram data being less than the actual dimensionality of the hyper-sphere described by histogram patterns is that the quadrant is not completely covered but the data describes trajectories across the surface. Simply estimating the storage by counting the number of patches which completely cover the quadrant's surface results in a gross over-estimate. Figure 5.3 depicts a hypothetical situation where, although the data is 2-dimensional, it is constrained to lie along a trajectory which locally is 1-dimensional. The capacity in this example is the number of distinct patterns which can lie along the trajectory. In practice we would expect the surface to be covered by many disjoint, low dimensional trajectories which are locally smooth. This is because, given an arbitrary shape, smooth deformations such as scalings and shearings are likely to produce different but equally viable shapes, all of which lie on a locally continuous manifold. Significantly different shapes are unlikely to lie on the same manifold however. In general, then, the capacity is driven by the local dimensionality of the geometric histogram data and the extent to which the trajectories described by the data tend to fill the higher-dimensional space they occupy.

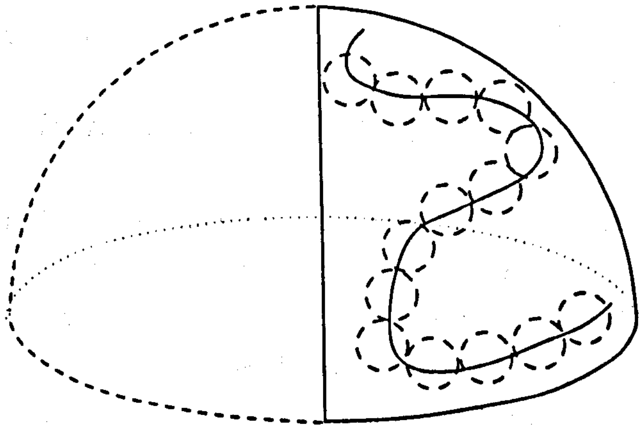


Figure 5.3: Geometric histogram data does not fully populate the surface of the hypersphere's positive quadrant but describes low-dimensional trajectories across the surface. Therefore, the capacity is the number of distinct patterns which can be placed along these trajectories.

The extent to which the histogram pattern data fills the higher dimensional space is difficult to calculate but it is possible to estimate the local dimensionality of the data and this can be used to estimate a lower bound on the capacity. The capacity along a *space-filling* trajectory will always be greater than the capacity along a trajectory of the same dimensionality which does not fill space so the number of patterns which fit along a straight trajectory through the pattern space is likely to be a lower bound on the capacity, see Figure 5.4. The area along a straight trajectory of dimensionality, n_i, can be determined using the same integral which is used to calculate the surface area of an n-dimensional hyper-sphere, where n is replaced by n_i. The capacity is simply calculated using the same expression as before (Equation 5.6) but where n is replaced by n_i.

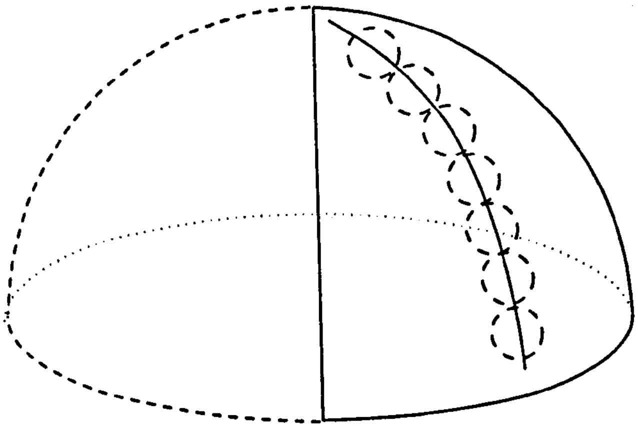
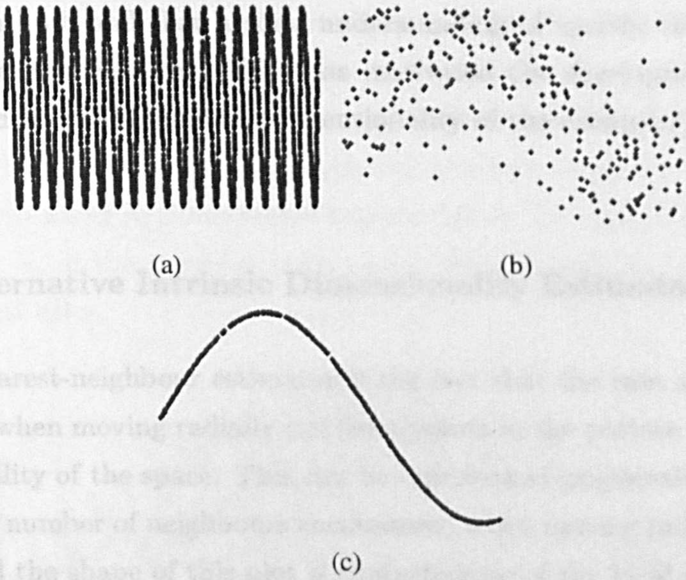


Figure 5.4: A lower bound on capacity can be estimated using the local dimensionality, n_i, of the pairwise geometric histogram data by counting the number of distinct patterns which lie along a straight, n_i-dimensional trajectory.

5.2.2 Estimating Intrinsic Dimensionality

The problem of determining the intrinsic dimensionality of a set of data is an interesting one that has many uses in the field of pattern recognition but, as yet, no general solution has been found. At first the problem does not appear to be a difficult one (a human observer can look at 1- or 2-dimensional data embedded in a 3-dimensional space and recognise the intrinsic dimensionality with little trouble) but there are complications. First of all the *apparent* dimensionality of a data set depends upon the scale at which the data is observed, so an appropriate scale must be determined before the intrinsic dimensionality can be calculated. Again the human observer has little trouble in selecting the appropriate scale but embodying this process into an algorithm has proved to be difficult. Secondly, real data tends to be influenced by noise which has the effect of thickening the hypersurfaces that the data lies on, resulting in an increase on the apparent dimensionality. A robust technique to decide whether data is thickened because of noise or really does have a larger intrinsic dimensionality has not been found. Examples of these problems are depicted in Figure 5.5.

A number of techniques for measuring intrinsic dimensionality have been published although they all tend to be variations on either the *local PCA* (Principal Component Analysis) approach or the *nearest-neighbour* approach. The local PCA approach was developed by Fukunaga [Fukunaga & Olsen 71] and, as the name suggests, is based on the principal component analysis technique which is traditionally used to determine the minimum number of orthogonal directions needed to describe a data set. In the local PCA approach an estimate of the intrinsic dimensionality is obtained by determining the typical number of orthogonal directions needed to describe local regions of the data. The local re-



(a) (b)

(c)

Figure 5.5: Estimation of the intrinsic dimensionality of the data in (c) is hindered by (a) viewing the data at an inappropriate scale, and (b) noise which increases the apparent dimensionality.

gions used in this analysis can be defined in a number of different ways. Trunk [Trunk 76] progressively increases the number of nearest-neighbours, k, in each locally defined region until the $(k+1)$ th neighbour fits into the coordinate frame defined by the first k data-points. Other approaches define local regions by attempting to cluster the data according to measures of topology or distance [Schwartzmann & Vidal 75]. The need to select an appropriate neighbourhood size in which to calculate principal components is a consequence of the scale problem discussed earlier.

The nearest-neighbour approach utilises the fact that the relative distance from points in the pattern space to the nearest-neighbouring exemplars is a function of the local dimensionality of the data whilst tending to be unaffected by the total number of stored exemplars and the actual distribution of the data [Fukunaga 90] [Pettis 79]. In fact, it can be shown that:

$$n_i = \frac{d_k}{(d_{k+1} - d_k)k} \tag{5.7}$$

Where d_k is the mean distance to the kth nearest neighbour from each pattern in the space. Again local regions are effectively defined, this time by selecting k, and good results depend upon selecting k appropriately.

A recent evaluation of these existing methods [Verveer & Duin 95] concludes that there are problems with both and that a good understanding of specific data sets is required to interpret the results they give. This has motivated the development of an alternative technique to measure the intrinsic dimensionality of the exemplar pairwise geometric histograms.

5.2.3 An Alternative Intrinsic Dimensionality Estimator

Central to the nearest-neighbour estimator is the fact that the rate at which neighbours are encountered, when moving radially out from points in the pattern space, is a function of the dimensionality of the space. This can be represented graphically by plotting a histogram of the the number of neighbours encountered when moving radially outwards from any exemplar and the shape of this plot is characteristic of the local dimensionality. The local density of the data introduces a scale factor which can be removed by constructing the histogram up to some specified radial distance, $D_{neighbour}$, and then normalising its area. This selection of a neighbourhood size is again a consequence of the problem of analysing the data at an appropriate scale. For convenience, we shall refer to the histogram constructed in this way as the *neighbourhood distance histogram* or simply *distance histogram.*

It is proposed here that the local dimensionality of a set of data is estimated by fitting distance histograms constructed for the data to distance histograms constructed for simulated data of known intrinsic dimensionality. Distance histograms constructed for a single data point characterise the local dimensionality at that data point but an average, local dimensionality can also be determined by using the sum of the distance histograms constructed at every data point. Interestingly, these mean distance histograms can be derived directly from the interclass distance distributions used to predict reliability in the previous chapter. The neighbourhood distance histogram is simply the section of the interclass distance distribution below $D_{neighbour}$ which is then normalised. An important point to note is that the shape of these distributions characterises both the reliability of classification *and* the local dimensionality of the data. This agrees with the observation made by Fukunaga that for classification problems involving large numbers of classes the error rate is driven by the intrinsic dimensionality of the data [Fukunaga & Flick 84].

To test this approach, distance histograms constructed for data uniformly distributed over the positive quadrant of a hyper-sphere have been compared to distance histograms constructed for data lying in low-dimensional manifolds on higher dimensional hyperspheres. To generate random data points on the surface of a hyper-sphere which are uniformly distributed, each component of the vector describing a point is selected randomly according to a normal distribution and then the vector is normalised to the sphere's radius (in this case unity). In order to constrain data points to the positive quadrant of the hyper-sphere, the absolute value of each vector component is used. Figure 5.6 shows a set of points generated this way which lie on the positive quadrant of a 3-dimensional sphere. To generate data points along n_i-dimensional trajectories on the surface of an n-dimensional hyper-sphere only n_i of the n vector components are selected randomly, the rest being assigned a constant value.

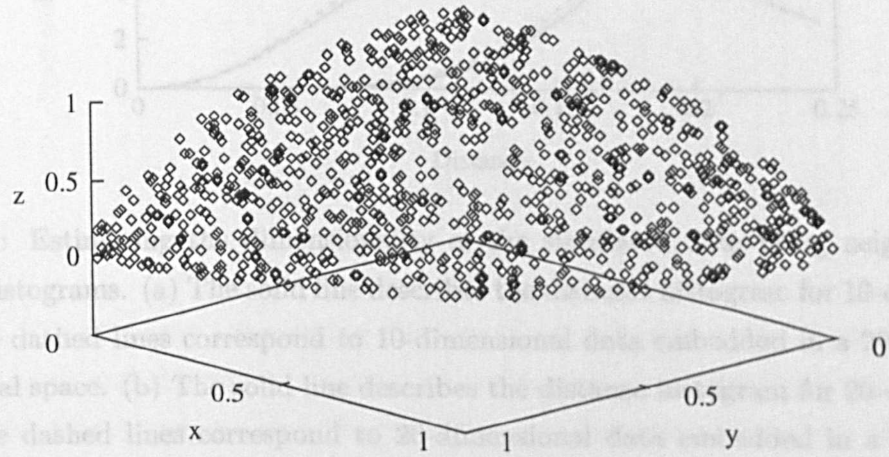


Figure 5.6: Simulated data uniformly distributed over the positive quadrant of a sphere.

Figure 5.7 shows distance histograms constructed for 10-dimensional data in 10, 20 and 40-dimensional spaces and 20-dimensional data in 20, 40 and 80-dimensional spaces. It is clear from this result that the shape of the distance distribution is characterised by the intrinsic dimensionality of the data.

To re-cap, the proposed method for estimating the capacity of the geometric histogram representation is to first estimate the mean, intrinsic dimensionality of the data, n_i, characterised by the shape of the interclass distance distribution for small distances. A lower bound on capacity can then be estimated by counting the number of distinct classes which will fit on trajectory of this dimensionality using the expression:

$$C = \frac{\int_0^{\pi} sin^{n-2} \alpha d\alpha}{\int_0^{acos(1-D_{th})} sin^{n-2} \alpha d\alpha} \frac{1}{2^{n_i}} \qquad (5.8)$$

5.3 Results: The Geometrical Approach

A lower bound on the capacity of the pairwise geometric histogram representation is estimated here for two different sets of shape data, the first comprising outlines of shape

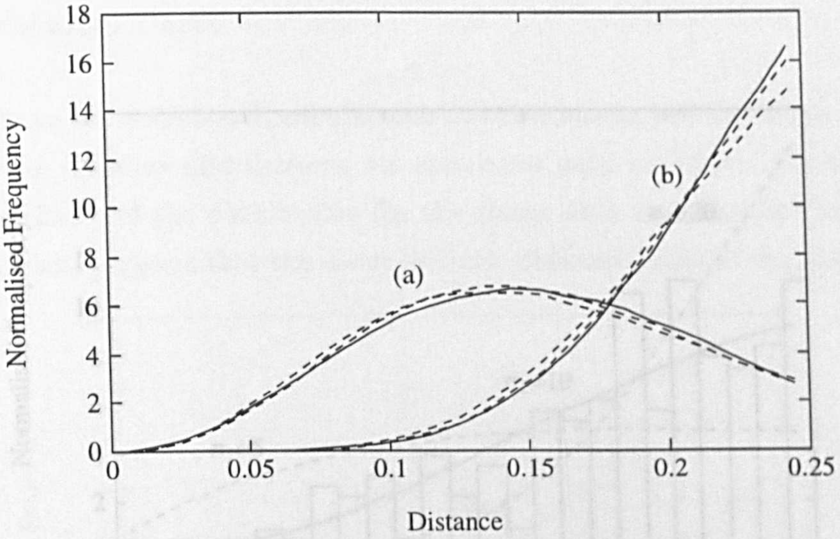


Figure 5.7: Estimating the dimensionality of the simulated data using neighbourhood distance histograms. (a) The solid line describes the distance histogram for 10-dimensional data. The dashed lines correspond to 10-dimensional data embedded in a 20- and a 40-dimensional space. (b) The solid line describes the distance histogram for 20-dimensional data. The dashed lines correspond to 20-dimensional data embedded in a 40- and an 80-dimensional space.

silhouettes (Appendix A, Figures A.1 and A.2) and the second comprising views of mechanical parts (Appendix A, Figure A.3). For each data set the mean intrinsic dimensionality has been estimated using neighbourhood distance histograms and then, with the within-class variability measured in the last chapter, the capacity is derived. Because measures of within-class variability and intrinsic dimensionality are subject to some error the capacity is plotted for a range of these parameters, providing an insight into the sensitivity of the capacity estimate to these values.

5.3.1 Shape Outlines

Figure 5.8 shows the neighbourhood distance distribution for the shape outline data as bars and the distance distributions for simulated data of known dimensionality as lines. The similarity of the distribution for the shape data and the distribution for the 10-dimensional data suggests that the mean intrinsic dimensionality of the shape data is 10.

The estimated capacity of the outline shape data, based on an estimated dimensionality of 10 and a within-class variability of 0.15, is plotted in Figure 5.9 with some margin. This result suggests that in excess of 1000 shape primitives of those typically present in the training data should be capable of being stored without ambiguity.

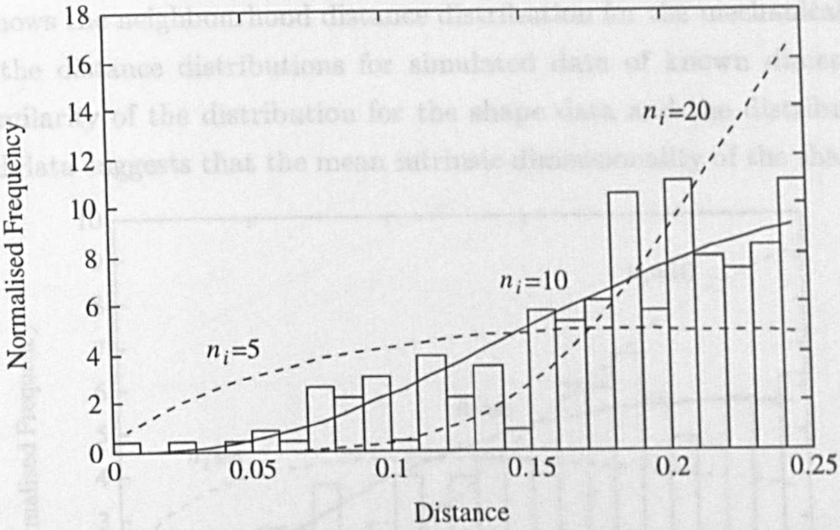


Figure 5.8: The similarity between the distance histogram constructed for the shape data (shown as bars) and the distance histogram constructed from simulated 10-dimensional data (the solid line) suggests that the mean intrinsic dimensionality of the shape data is about 10.

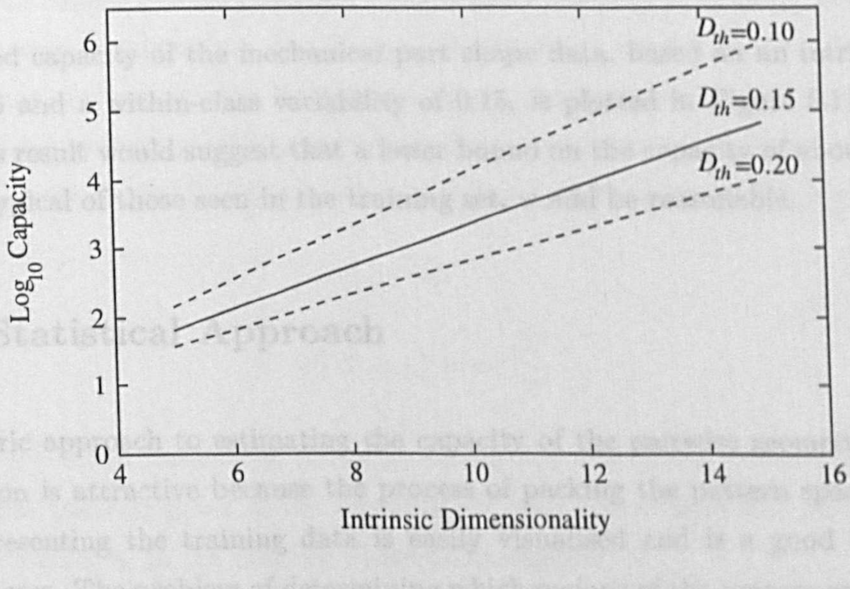


Figure 5.9: Lower bound on capacity as a function of intrinsic dimensionality and within-class variability around the estimated values.

5.3.2 Mechanical Parts

Figure 5.10 shows the neighbourhood distance distribution for the mechanical shape data as bars and the distance distributions for simulated data of known dimensionality as lines. The similarity of the distribution for the shape data and the distribution for the 6-dimensional data suggests that the mean intrinsic dimensionality of the shape data is 6.

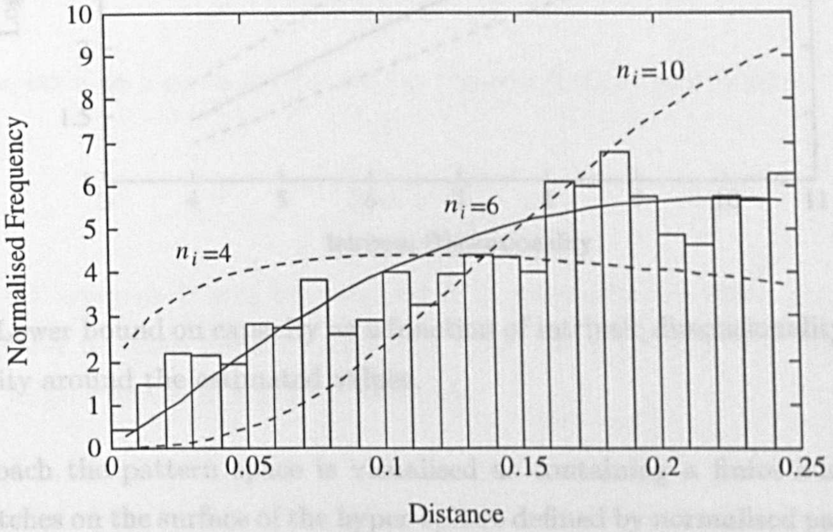


Figure 5.10: The similarity between the distance histogram constructed for the shape data (shown as bars) and the distance histogram constructed from simulated 6-dimensional data (the solid line) suggests that the mean intrinsic dimensionality of the shape data is about 6.

The estimated capacity of the mechanical part shape data, based on an intrinsic dimensionality of 6 and a within-class variability of 0.15, is plotted in Figure 5.11 with some margin. This result would suggest that a lower bound on the capacity of about 100 shape primitives, typical of those seen in the training set, would be reasonable.

5.4 A Statistical Approach

The geometric approach to estimating the capacity of the pairwise geometric histogram representation is attractive because the process of packing the pattern space with small regions representing the training data is easily visualised and is a good model of the training process. The problem of determining which regions of the pattern space represent likely shape configurations is a difficult one though, and has restricted the use of this technique to finding a lower bound. This limitation has motivated the development of an alternative approach based on a statistical model of the training process.

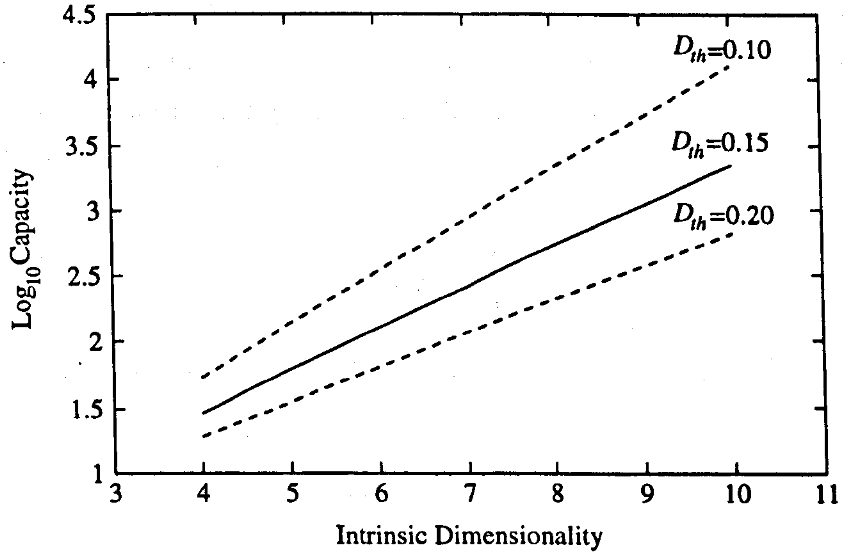


Figure 5.11: Lower bound on capacity as a function of intrinsic dimensionality and within-class variability around the estimated values.

In this approach the pattern space is visualised as containing a finite number of class locations (patches on the surface of the hyper-sphere defined by normalised patterns) which can become filled as exemplar patterns are stored. The importance of this alternative approach is that no assumption need be made about the distribution of these potential storage locations through the space but it is assumed that any training pattern has an equal probability of falling into any particular patch. This poses the problem as a simple binning process which can be modelled using Poisson statistics. Having trained with a particular number of exemplars, the Poisson distribution can be used to determine the probability that any one storage location will contain a given number of patterns. If the number of stored exemplars which have fallen into any one patch is given by the random variable X then the probability of a patch containing x patterns is described by the Poisson distribution.

$$P(X = x) = e^{-\lambda}\frac{\lambda^x}{x!} \tag{5.9}$$

Where λ is the mean number of patterns within any one patch. The total number of patches in the pattern space is the capacity, C, and this relates the mean number of patterns per patch, λ, to the number of stored exemplars, N.

$$\lambda = \frac{N}{C} \tag{5.10}$$

After training with a set of exemplars some storage locations are likely to be empty, some will contain a single exemplar and some may even contain several. Defining C_x to be the number of patches containing x patterns, such that $C = C_0 + C_1 + C_2 + ...$, then the probability of a patch containing x patterns can be estimated as:

$$P(X = x) \approx \frac{C_x}{C} \qquad (5.11)$$

This can be equated to Expression 5.9 for the Poisson distribution to give:

$$\frac{C_x}{C} = e^{-\lambda}\frac{\lambda^x}{x!} \qquad (5.12)$$

From this point onwards it will be assumed that relatively few class locations will be occupied by three or more patterns and that many more patches will contain a single pattern than those containing a pair of patterns.

$$C_1 \gg C_2 \qquad (5.13)$$

This is a fair assumption provided that the number of training examples is less than the number of storage locations and given the earlier assumption that exemplars are equally likely to fall into any patch. These assumptions also lead to another expression which will be used shortly.

$$C_1 \approx N \qquad (5.14)$$

Considering only those patches containing either one or two exemplars, Equation 5.12 yields the following two expressions.

$$P(X = 1) = e^{-\lambda}\lambda = \frac{C_1}{C} \qquad (5.15)$$

$$P(X = 2) = e^{-\lambda}\frac{\lambda^2}{2} = \frac{C_2}{C} \qquad (5.16)$$

Dividing Expression 5.15 by Expression 5.16 yields an expression relating the ratio of the number of patches containing one exemplar to the number of patches containing a pair of exemplars to λ.

$$\frac{C_1}{C_2} = \frac{2}{\lambda} \tag{5.17}$$

From which the expression for capacity C can be derived by substituting in the expression for λ from Equation 5.10 above.

$$C = \frac{N}{2}\frac{C_1}{C_2} \tag{5.18}$$

This can now be simplified using the approximation that $C_1 \approx N$.

$$C \approx \frac{N^2}{2C_2} \tag{5.19}$$

This expression defines the capacity in terms of the number of stored exemplars, N, which is known, and the number of patches containing a pair of exemplars, which can be counted. As in the geometric approach, a patch describing what we intuitively mean by a distinguishable shape is defined by a radius D_{th}, which is set according to the observed variability of the data. The number of patches containing pairs of exemplars, C_2, is simply assumed to be the number of pairs of exemplars less than a distance D_{th} from each other.

This information is provided explicitly by the distribution of interclass distances introduced in the last chapter. C_2 is simply the number of entries made in the distribution at a distance less than D_{th}. Rather than expressing the capacity C in terms of C_2 it is useful to express it in terms of the proportion, p, of the interclass distribution below D_{th}, depicted in Figure 5.12.

Recall that the interclass distribution is constructed by matching all pairs of exemplars drawn from the training set of N. The number of entries made in the distribution is therefore:

$$\text{Number of entries} = \frac{N^2 - N}{2} \tag{5.20}$$

The proportion of entries, p, made below D_{th} is the ratio of the number of entries, C_2 made below D_{th} to the total number of entries made into the distribution.

$$p = \frac{2C_2}{N^2 - N} \tag{5.21}$$

Provided that the number of exemplars, N, is large such that $N^2 \gg N$, then this can be simplified to:

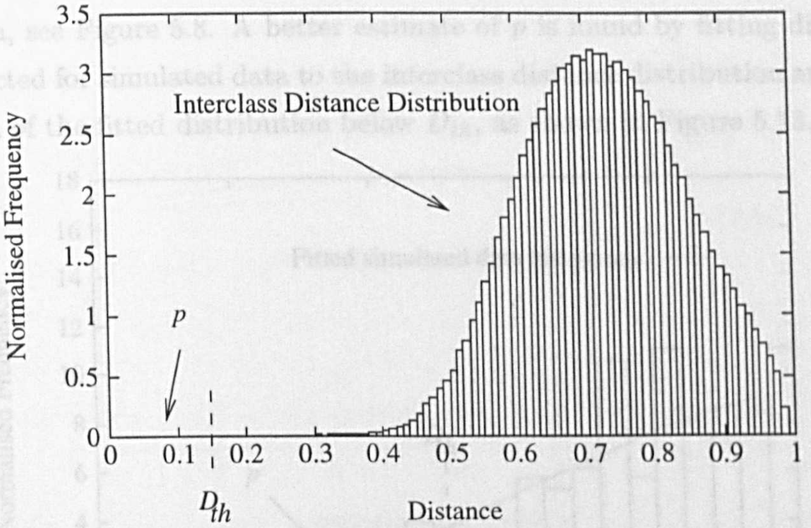


Figure 5.12: The capacity can be derived directly from the proportion, p, of the interclass distance distribution below D_{th}.

$$p = \frac{2C_2}{N^2} \tag{5.22}$$

Using this to replace C_2 in the expression for capacity given by Equation 5.19 gives:

$$C = \frac{1}{p} \tag{5.23}$$

This expression provides a good estimate of the capacity which makes no assumption about the distribution or dimensionality of the geometric histogram pattern data and allows the capacity to be determined using a simple measurement taken from the interclass distance distribution. In order to obtain an accurate estimate of the capacity it is necessary to have a good measure of p and this depends upon a good sample of data in the interclass distribution below D_{th}. Unfortunately there tends to be very little data in this part of the interclass distribution, see Figure 5.12 for example, and this leads to a large error in the estimate of the capacity.

5.4.1 Estimating p by curve fitting

To minimise errors on the estimate of capacity, a better measurement of p can be determined by fitting an appropriate model to the interclass distance distribution over a range where there is sufficient data. In fact, suitable model fitting was done in the geometric approach described earlier when measuring the intrinsic dimensionality of the geometric histogram data, see Figure 5.8. A better estimate of p is found by fitting distance histograms constructed for simulated data to the interclass distance distribution and measuring the proportion of the fitted distribution below D_{th}, as shown in Figure 5.13.

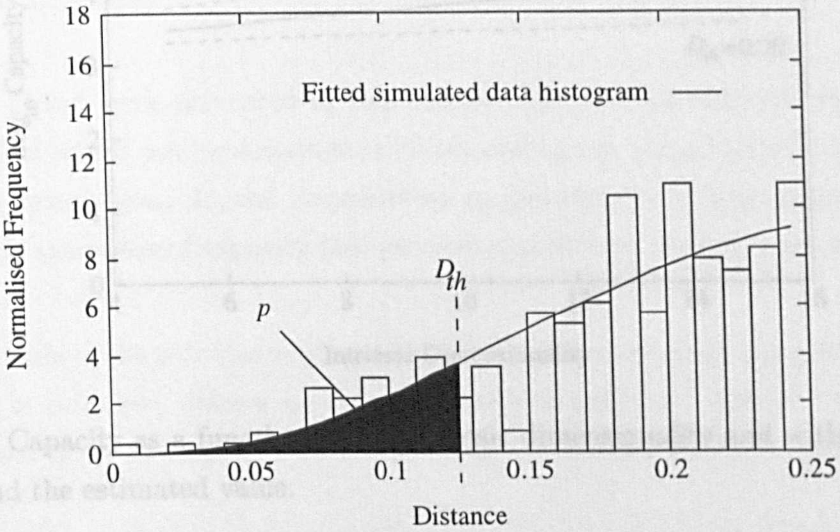


Figure 5.13: A better estimate of p can be determined by measuring the proportion of the simulated distance distribution, which best fits the data, below D_{th}.

5.5 Results: The Statistical Approach

The capacity of the pairwise geometric histogram representation is estimated here for two different sets of shape data, the first comprising outlines of shape silhouettes (Appendix A, Figures A.1 and A.2) and the second comprising views of mechanical parts (Appendix A, Figure A.3). For each data set the proportion, p, of the interclass distance distribution below D_{th} (D_{th} being determined from the spread of the within-class distance distribution) is measured and the capacity derived. To improve the accuracy of this calculation p is derived from the estimate of the intrinsic dimensionality of the data determined earlier. The capacity is then plotted as a function of the intrinsic dimensionality around the estimated value.

5.5.1 Shape Outlines

The estimated capacity for the outline shape data, based on the value of p predicted from the estimated intrinsic dimensionality and a within-class variability of 0.15, is plotted in Figure 5.14 with some margin. This result suggests that a capacity of up to 10000 seems reasonable for this type of shape data. It is interesting to note that the capacity estimate

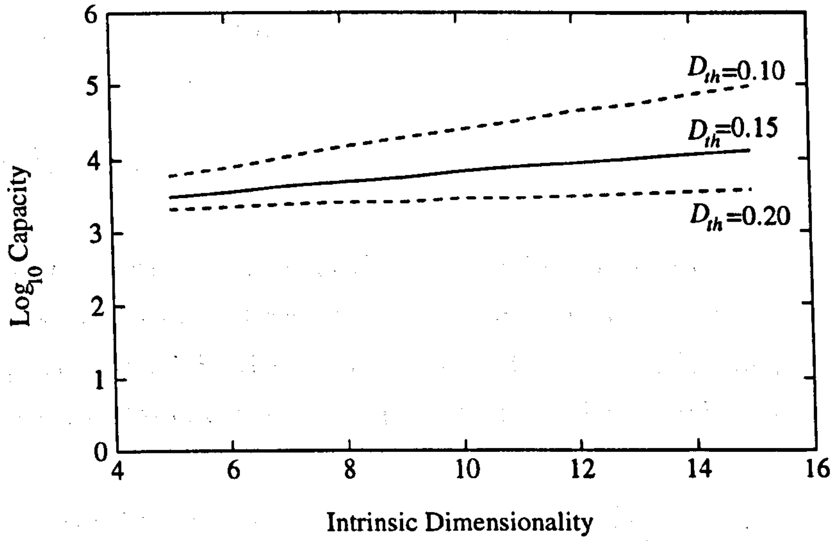


Figure 5.14: Capacity as a function of the intrinsic dimensionality and within-class variability around the estimated value.

agrees within an order of magnitude over a large range of intrinsic dimensionality and within-class variability.

5.5.2 Mechanical Parts

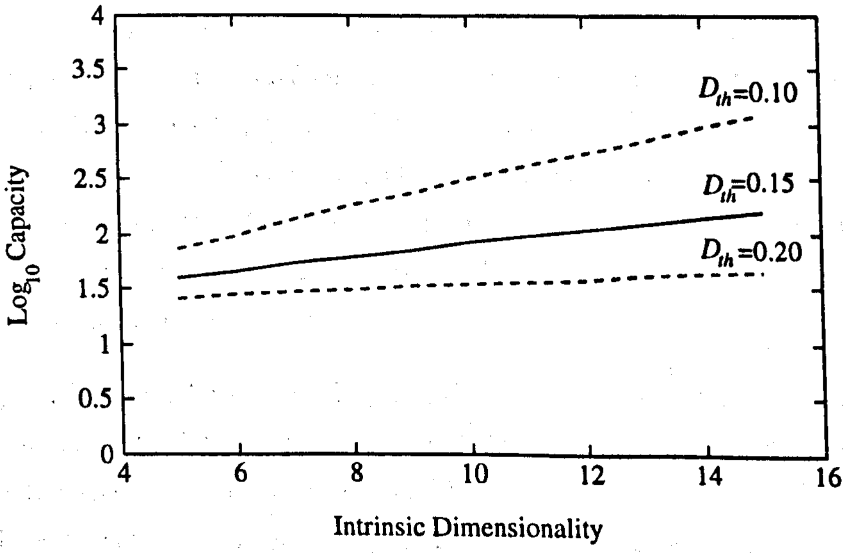


Figure 5.15: Capacity as a function of the intrinsic dimensionality and within-class variability around the estimated value.

The estimated capacity for the mechanical part shape data, based on a value of p predicted for the estimated intrinsic dimensionality and a within-class variability of 0.15, is plotted in Figure 5.15. It seems reasonable to say that a capacity of about 100 distinct shape primitives can be expected for this type of data.

5.6 Conclusions

The purpose of the work presented in this chapter has been to estimate the number of shape primitives which can be described, without ambiguity, using the pairwise geometric histogram representation. In the introduction to the chapter it was suggested that to obtain sensible estimates of capacity two particular problems needed to be solved.

1. The domain of the pairwise geometric histogram representation is a continuous space which, in principle, defines an infinite variety of patterns. How can this space be quantised in an appropriate manner so that the number of unique patterns can be counted?
2. Not all points in the geometric histogram domain represent configurations of shape which are ever likely to be encountered. Given only a relatively small number of training shapes, is it possible to derive an estimate of capacity which takes this into account?

The problem of quantising the space has been addressed by defining *uniqueness* using hyper-circular patches, on the surface of the hyper-sphere which describes all possible patterns. The radius of each patch is determined by observing the typical variability of geometric histograms within the same class. Although rather simplistic, this solution seems reasonable as it is founded on the distance metric, which is by definition the measure of similarity. This definition of uniqueness has been used as the basis of two different approaches for estimating the capacity.

The first approach derived capacity by counting the number of patches which can be packed onto the surface of the representational domain. The problem of predicting the proportion of the space which represents viable shape primitives was solved by estimating the local, or intrinsic, dimensionality of the training shape data and assuming that further shape data will be confined to trajectories of this dimensionality. Estimation of intrinsic dimensionality is an on-going area of research area and problems with current solutions motivated the use of an alternative technique. This technique bases the estimate of dimensionality on the shape of the interclass distance distribution for small distance. It was argued that this was likely to provide a lower bound on the capacity rather than the capacity itself.

The second approach modelled the storing of training data as a binning process which can be described using Poisson statistics. It was shown that the capacity can be defined in terms of the number of patches containing pairs of shape exemplars after training and this is easily measured using the interclass distance distributions. The advantage of this approach is that no assumption is made about the distribution of the classes through the space so that the proportion of the space representing viable shape is implicit in the calculation. In practice, using the interclass distributions directly leads to large errors on the estimated capacity because of the small number of patches containing pairs of patterns. This problem has been minimised by fitting model curves to the interclass distributions where there is sufficient data and extrapolating.

An important result, although maybe an obvious one, to come from this work is that the capacity of the representation depends upon the type of shape data being used. These experiments suggest that for data typical of the shape outlines used, many thousands of primitives can be represented uniquely, whereas, for the mechanical data set which exhibits many similar features such as right-angled corners, the capacity was estimated to be around 100. It is interesting that this difference between the shape data sets has been made explicit by measuring the data's intrinsic dimensionality. Another interesting observation which should be made is that the interclass distance distributions, used in the last chapter to estimate reliability, have been used here to estimate intrinsic dimensionality. This agrees with the intuition that there is a relationship between reliability and capacity and confirms the observation made by Fukunaga that for classifications tasks involving large numbers of classes, the reliability is driven by the intrinsic dimensionality of the data.

Chapter 6

Algorithm Extensions for the Recognition of Scaled Shapes

6.1 Introduction

One of the main criticisms of the pairwise geometric histogram representation is that it is not invariant to changes in the apparent size or scale of a shape. A geometric histogram constructed to represent shape data provides a description of that shape at a specific scale and is different to a geometric histogram representing the same shape data at a different scale. This lack of scale invariance is a consequence of using distance measurements in the description of shape and, until now, has limited recognition to shapes of a fixed size. A scale invariant representation can be obtained by recording only the relative angle between pairs of line segments in a 1-dimensional descriptor, effectively projecting the data in a 2-dimensional pairwise geometric histogram onto the relative angle axis. This was tried during the development of the current geometric histogram representation but lacked the sparseness which promotes robust recognition in cluttered scenes. An alternative geometric histogram representation has been developed by Kumar [Kumar & Rockett] in which triplets of points on an objects boundary are used to define a pair of angles and an entry is made in a 2-dimensional histogram accordingly. Although these angle measures are invariant to the scale of the shape data the errors on the measurements made from the image data are not. The correct level of blurring made to each histogram entry to account for measurement errors is a function of the scale of the shape data and so the representation is not entirely scale invariant. In fact, any scale invariant representation of shape which properly accounts for measurement errors, whether based on angles or ratios of distances, will suffer from this problem. It is interesting, though, that the problem can be minimised by centring an object on a sensor with a resolution which increases exponentially towards its centre, so that measurement errors become constant up to the maximum resolution of the sensor. This may be one of the reasons why the human retina has a structure similar to this.

The motivation for developing an algorithm which can recognise shapes at any scale really comes from two sources. The first originates from the concept of shape and its independence of scale. From our own experience we can look at a shape over a large range of scale and still perceive it as the same shape. It is desirable for a shape classification system to be able to reflect this. The second source of motivation is more practical and concerns recognising objects at different distances from the camera. Except in artificially constrained environments objects can be expected at arbitrary distances from the camera and consequently appear at different scales. Many more vision problems could be solved using geometric histograms if the constraint on scale is removed.

Although the pairwise geometric histogram representation is not invariant to the scale of shape data, an algorithm for recognising shape over a range of scale using this type of descriptor is developed and tested in this chapter. In the next section the relationship between the apparent size of an object in an image and its distance from the camera is investigated and the calibration needed to predict the distance to an object from its scale is explained. The effect of scale on the geometric histogram representation is studied in some detail and this leads to a technique for representing and recognising shapes over a specified range of scale. Qualitative and quantitative results are then presented to demonstrate the effectiveness of this extension to the original algorithm. Finally, results of an application where an object is tracked by recognising it in a series of images are presented.

6.2 Scale, Distance and Calibration

One of the main motivations for developing a shape recognition system which can recognise shapes at arbitrary scale is the fact that the apparent size of an object in an image is a function of its distance from the camera. This relationship is investigated here and the measurements which must be made to calibrate a shape model so that an object's distance from the camera can be determined are defined. The analysis begins by defining the scale, s, as the factor which relates the apparent size of an object in an image, x, to its actual size in the world, X.

$$s = \frac{x}{X} \tag{6.1}$$

The relationship between image and object sizes and the distance between the object and the camera can be determined, to first order, using the pinhole camera model, as presented in Figure 6.1.

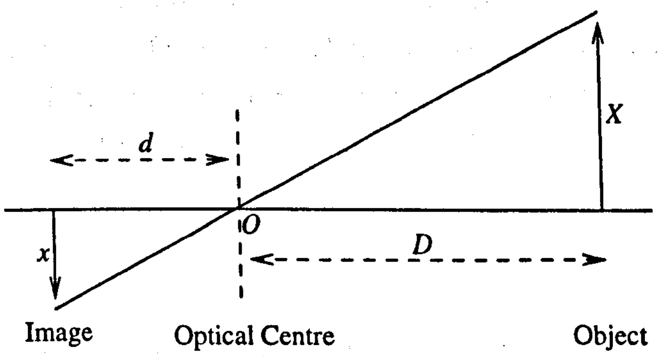


Figure 6.1: The simple pinhole camera model can be used to relate image and object sizes to image and object distances.

If the distance from the object to the optical centre, O, is D and the distance from the optical centre to the image plane is d then, by noticing that a ray from a point on the object to the image defines a pair of similar triangles, the following expression can be formed.

$$\frac{x}{X} = \frac{d}{D} \tag{6.2}$$

By rearranging this expression and substituting in the expression for scale an expression for D in terms of d and the scale of the image is obtained.

$$D = \frac{d}{s} \tag{6.3}$$

Typically the distance d from the image plane to the optical centre is not well known but this can be replaced by suitable calibration of each model. If an object is placed at a distance D_x from the camera and the scale of the image at this distance is s_x then, using Expression 6.3.

$$d = s_x D_x \tag{6.4}$$

Substitution of this back into Expression 6.3 gives an expression for D in terms of these calibration measurements and the scale of the image, s.

$$D = \frac{s_x D_x}{s} \tag{6.5}$$

Rather than expressing D in terms of the size of the image relative to the size of the object it is convenient to use the size of the image relative to the size of the model constructed for the training data, s_m. If the model is constructed for the object when it is at a distance D_x from the camera then the model scale is simply:

$$s_m = \frac{s}{s_x} \tag{6.6}$$

This can be used to obtain an expression for D in terms of the model scale by substitution into Expression 6.5.

$$D = \frac{D_x}{s_m} \tag{6.7}$$

In other words, the distance from an object to the camera can be determined from the scale of the model which fits the image data and the distance that the object was placed from the camera when the model was constructed.

6.3 The Effect of Scale on the Similarity Metric

As measures of distance are used in the construction of pairwise geometric histograms, the representation changes as the scale of the encoded shape data is changed. As a consequence, histograms constructed from the same shape data but at different scales exist at different locations in pattern space and this complicates the classification process. In order to develop a classification scheme which can work with shape data at various scales, it is necessary to understand the way that geometric histogram patterns move around the pattern space as the scale of the shape data is varied. This may be done by looking at the way the data in a geometric histogram changes as a function of scale and the consequence that this has on the distance metric.

The effect of scale on the representation can be seen by considering the effect of scale on a single entry, as demonstrated in Figure 6.2. For any particular scale, s, the perpendicular distances d_0 and d_1 are simply scaled to distances sd_0 and sd_1 respectively, such that the histogram entry becomes stretched (or compressed) along the perpendicular distance axis. This will be the case for each of the individual entries that make up a complete geometric histogram such that the overall effect is a stretching (or compressing) of the histogram data as a whole. Note that it has been assumed here that scaling the image data by some factor s simply results in the line data being scaled by the same factor. This is a good assumption over reasonable ranges of scale because of the scale invariant segmentation algorithm employed to perform the straight line approximation, but at very large or small scales other effects may become significant.

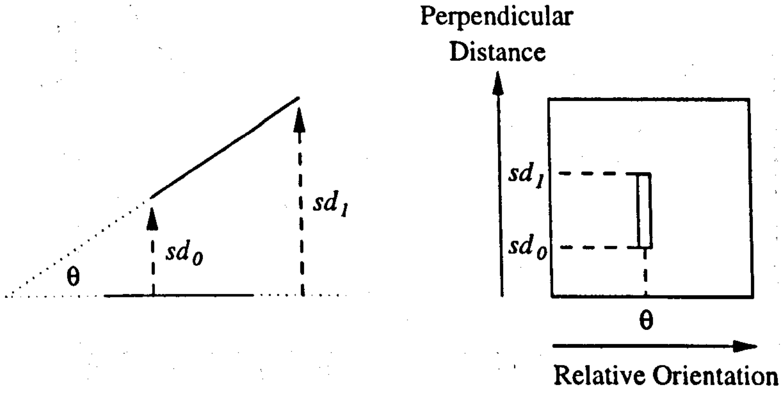


Figure 6.2: The effect of scale on the pairwise geometric histogram can be seen by considering a single entry. As the shape data is scaled the histogram entries become stretched or compressed.

Due to the fact that the pairwise geometric histogram data changes smoothly as a function of shape scale, the distance between a pair of histograms constructed from the same shape data must change smoothly as their relative scale is varied. This can be viewed as the histogram following a smooth trajectory through the pattern space as the scale of the shape data is varied. For clarity this will be called the ***shape trajectory***. The effect of scale on the distance metric is shown in Figure 6.3 which presents the distance between a pair of geometric histograms constructed from real shape data but at different scales. As geometric histograms remain similar over a reasonable range of scale, an individual histogram, although not scale invariant, can effectively represent a range of scale. It is this property which may be used to represent shape across any range of scale.

6.4 Representing Shape Over a Range of Scale

To enable shapes to be classified over some range of scale using pairwise geometric histograms it is necessary to represent the shape trajectories defined in the pattern space by these shapes across the scale range. Individual geometric histograms can effectively represent a small range of scale or section of the shape trajectory. Consequently, the complete trajectories may be stored in a piecewise fashion by storing a small number of exemplars. A hypothetical example is shown in Figure 6.4.

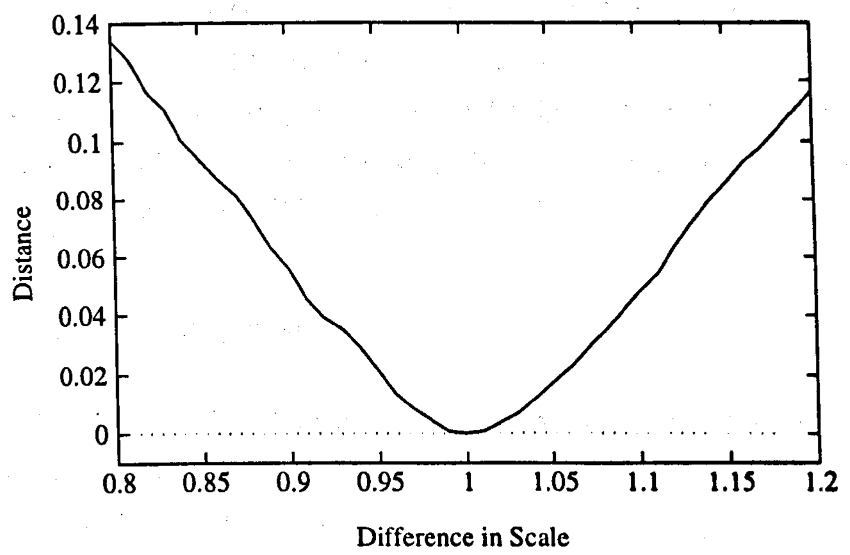


Figure 6.3: The distance between a pair of geometric histograms constructed for the same shape data at different scales.

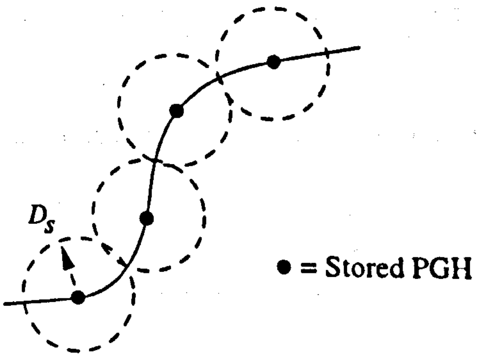


Figure 6.4: Shape primitives can be represented over a range of scale as a piecewise approximation to the trajectory described by the shape data.

To represent a shape primitive over some range of scale from s_{min} to s_{max} it is necessary to determine the number and positions of histograms to store along the shape trajectory. This depends upon how coarsely the trajectory is to be described and may be defined in terms of a distance threshold D_s, as shown in Figure 6.4. If this distance is chosen to be small then the trajectories will be well represented but at the expense of having to store a large number of geometric histograms. Alternatively, if this distance is large then few histograms will need to be stored but the trajectories will be poorly represented and the likelihood of misclassifications will increase.

6.4.1 The Training Algorithm

An algorithm has been designed to determine the values of scale at which histograms should be stored given values for D_s, s_{min} and s_{max}. The algorithm operates as follows, and can be visualised as shown in Figure 6.5.

1. Initially a *temporary* histogram is constructed at scale s_{min}. This is used to determine the scale s_0 which describes a geometric histogram at a distance D_s using a bisections search across the scale range.
2. A histogram is stored at scale s_0 and the temporary histogram is discarded.
3. The histogram stored at scale s_0 is used to find the next scale which describes a histogram a further distance D_s away. Another temporary histogram is constructed at this scale.
4. The new temporary histogram is used to determine the scale s_1 where the next histogram is stored.
5. This process is repeated until the complete range of scale is covered.

6.4.2 Classification of Shape Primitives

Ideally, an unknown shape primitive should be classified according to the shape trajectory which it is closest to in the pattern space and the scale of the primitive should be determined from the position of the closest point along the trajectory. This is possible, in principle, if the trajectory between the stored exemplars is assumed to be linear but in practice these trajectories are very non-linear and the computation needed to perform this calculation would seriously impair the recognition speed. Instead, unknown shape

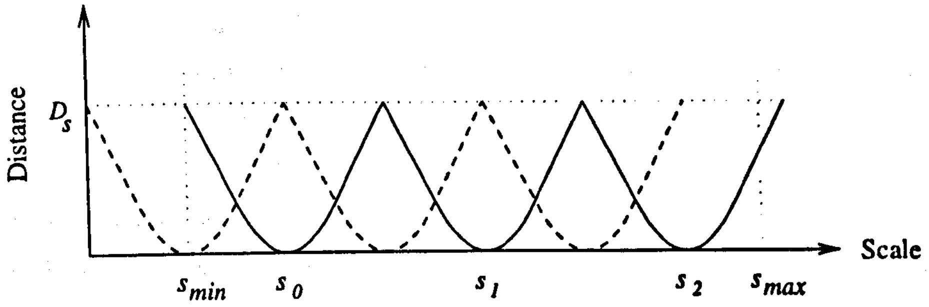


Figure 6.5: To represent a line segment over the range of scale from s_{min} to s_{max} to a level of precision defined by the distance threshold D_s three geometric histograms are stored for the shape data at scales of s_0, s_1 and s_2. The solid line represents the distance between the shape primitive and its nearest neighbour across the scale range.

primitives are classified, as before, according to the nearest stored exemplar and the scale estimate is simply based on the scale of that exemplar.

This introduces two problems. First of all, even if a shape primitive is a perfect example which lies on a shape trajectory it can lie up to a distance of D_s from its nearest neighbour, increasing the probability of misclassification. This is effectively a quantisation noise problem and can be included in the noise model introduced earlier to select an appropriate spacing at which to place exemplars to attain a given level of reliability. The second problem is that by basing the scale estimate on the scale of the nearest exemplar a uniform scale error is introduced. This complicates the determination of an object's position and scale in a scene but can be resolved by appropriate construction of the probabilistic Hough transform, which is the issue of the next section.

6.5 Determining Shape Location and Scale

When constructing a probabilistic Hough transform to determine the position of shapes identified within a scene for fixed scale data, the error on the position hypothesised by individual pairs of scene lines is largely due to variability in the line segmentation process. For variable scale data the position of a model hypothesised by labelled scene lines is also a function of the model scale and correct account of scale errors must be taken when constructing the Hough transform.

The effect of the uniform scale error on the hypothesised model position is demonstrated in Figure 6.6. The scales and scale errors associated with lines A and B constrain the position of the model to lie within the bands defined between the dotted lines. However, if both of these lines belong to the same shape then the model position must satisfy both constraints and lie in the shaded region. This does not provide a strong constraint on the position of the model and the precision of the recovered shape position suffers. However, if the scene lines belong to the same shape then they must also be at the same scale and this provides the tighter constraint represented by the dashed line. The section of the dashed line which intersects the shaded region satisfies all of the constraints so if both lines A and B are from the same model then its position must lie on this section.

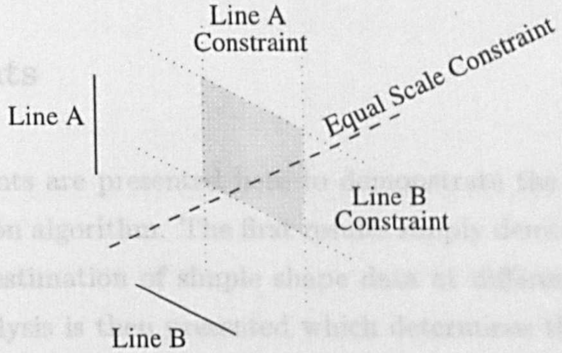


Figure 6.6: The effect of scale error on the hypothesised shape position.

The correct Hough transform entry for each pair of scene lines which accounts for both the scale error and the line endpoint error is obtained by convolving the respective error functions together. This is greatly simplified by approximating the scale error function by an ellipse which can be described using a covariance matrix. This has been done by orienting the major axis of the ellipse with the equal scale constraint and setting its length to 3 times the length of the section of equal scale constraint which defines the scale error. The minor axis of the scale error ellipse was set to one tenth of the length of the major axis. The covariance matrix which describes the combined scale and endpoint errors is then determined by simply adding their covariance matrices.

6.5.1 The Scale Hough Transform

Having determined the position of a model found in a scene, it is necessary to determine its scale. This may be done with a 1-parameter (Scale) Hough transform. The scale estimates associated with each matched scene line may be used to make entries into the scale Hough transform directly but the uniform scale errors on each estimate limit the precision of the final scale measurement. A better scale estimate can be determined for each line of the located model now that the position of the model has been determined. The improved model scale estimate, s_m, is simply calculated as:

$$s_m = \frac{d_{scene}}{d_{model}} \qquad (6.8)$$

Where d_{scene} is the perpendicular distance from the scene line to the determined shape location and d_{model} is the perpendicular distance from the corresponding model line to the corresponding model reference point.

These improved scale estimates provide good hypotheses for the scale of the shape and are used to form entries in the scale Hough transform to determine the actual model scale.

6.6 Experiments

A number of experiments are presented here to demonstrate the effectiveness of the extended shape recognition algorithm. The first results simply demonstrate the recognition, localisation and scale estimation of simple shape data at different scales. A more thorough, quantitative analysis is then presented which determines the accuracy of the scale estimate over a large number of example shapes. Finally the algorithm is used to track a moving object over a sequence of images by identifying the object in each image and estimating its distance from the camera by using the scale estimate.

6.6.1 Demonstration

In this experiment the recognition system has been trained with a single shape model over a range of scale from $s_m = 0.5$ to $s_m = 2.0$. With a distance threshold $D_s = 0.05$ an average of six geometric histograms were needed to represent each shape primitive over the scale range. Figure 6.7 shows three scenes containing the shape model at scales of 0.5, 1.0 and 2.0. The scene line data is shown in grey. These scenes were generated by scaling the original image data and then approximating the new image by line segments, rather than simply scaling the original line data. The identified models have been superimposed over each scene in black, at the determined location and scale. Close examination of the results reveals that the scene and model line data is generally quite different but the algorithm still performs well.

6.6.2 Quantitative Analysis

The experiment presented here has been devised to estimate the accuracy with which the scale of an object can be determined using the recognition algorithm. For two different shapes from Appendix A, one taken from the set of outlines and one from the mechanical parts, a large number of test images have been produced. For each shape, 10 examples at

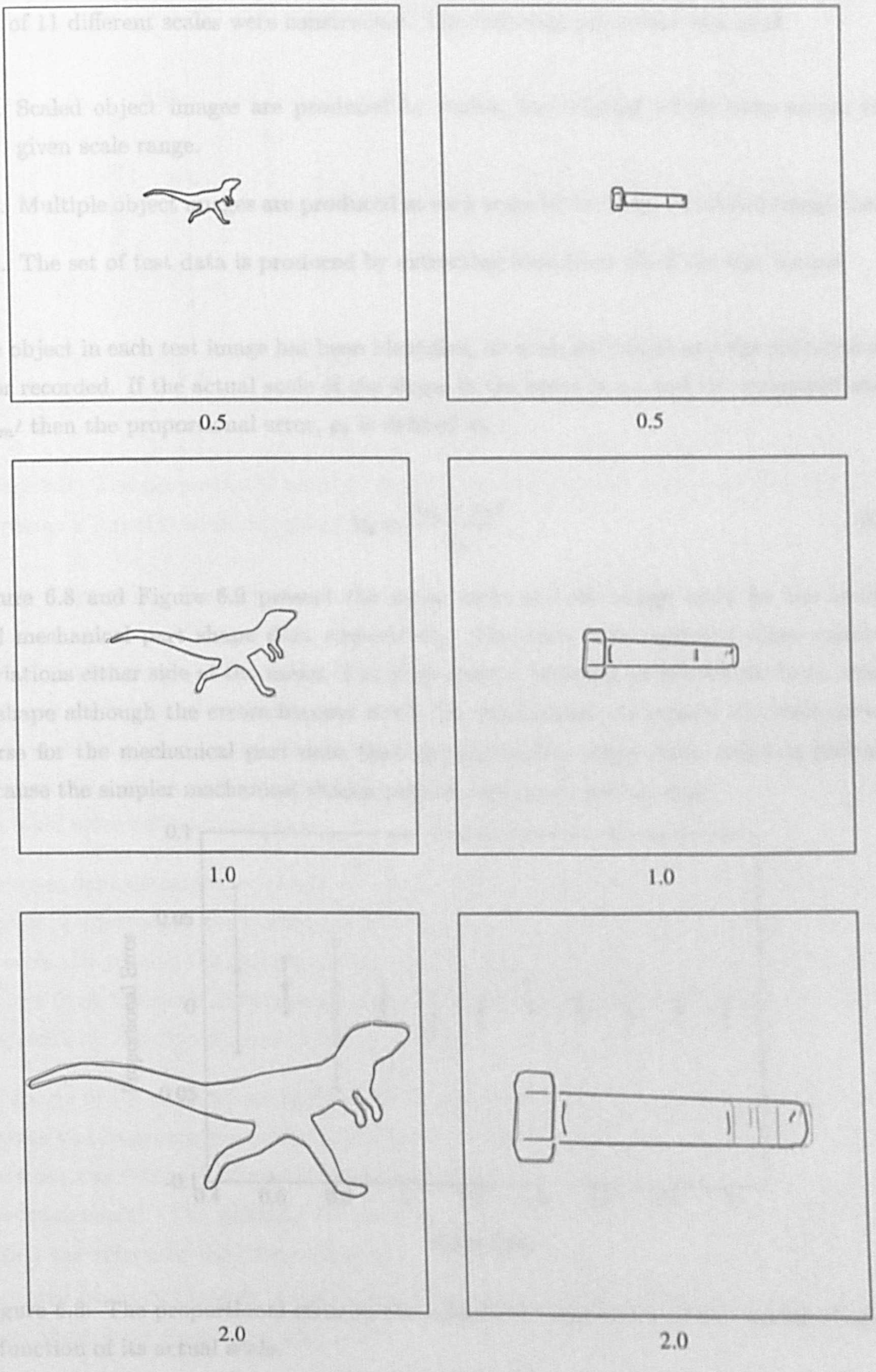


Figure 6.7: Scenes containing single models from the outline shape and mechanical part data sets at scales of 0.5, 1.0 and 2.0. The scene line data is shown in grey and the identified object is superimposed at the estimated location and scale in black.

each of 11 different scales were constructed. The following procedure was used.

1. Scaled object images are produced by scaling the original image data across the given scale range.
2. Multiple object images are produced at each scale by rotating the scaled image data.
3. The set of test data is produced by extracting lines from all of the test images.

The object in each test image has been identified, its scale estimated and the proportional error recorded. If the actual scale of the shape in the scene is s_m and the estimated scale is $s_{m'}$ then the proportional error, e_s is defined as:

$$e_s = \frac{s_m - s_{m'}}{s_m} \tag{6.9}$$

Figure 6.8 and Figure 6.9 present the mean error at each image scale for the outline and mechanical part shape data respectively. The error bars represent three standard deviations either side of the mean. The scale error is typically within 5% for both classes of shape although the errors become worse for small scales. In general the scale error is worse for the mechanical part data than for the outline shape data, which is probably because the simpler mechanical shapes provide less constraint on scale.

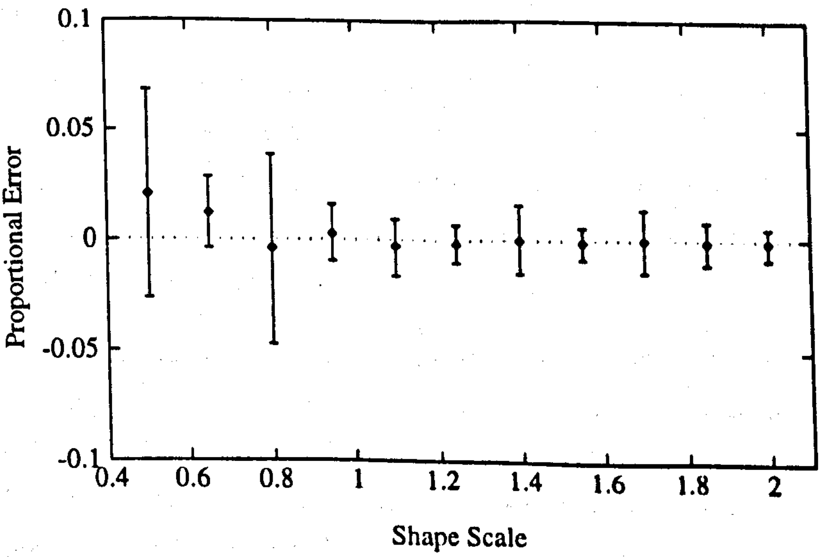


Figure 6.8: The proportional error in the estimated scale of one of the outline shapes as a function of its actual scale.

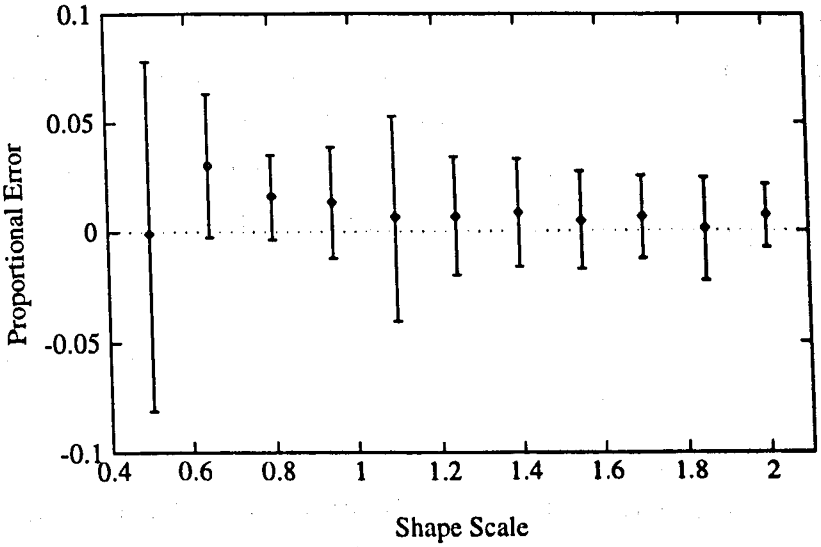


Figure 6.9: The proportional error in the estimated scale of one of the mechanical part shapes as a function of its actual scale.

6.6.3 Tracking Results

One of the main motivations for developing an algorithm which can recognise objects at arbitrary scale is to allow objects to be recognised at arbitrary distances from the camera. Once such an algorithm has been implemented it can be used to estimate the distance between the camera and the object from the object's scale in the image, providing there has been some calibration done.

A simple demonstration is presented here in which the distance to a model train is estimated as it approaches the camera by first recognising it in the scene, and then estimating its scale. By placing the camera along a straight section of railway track the train is always viewed from the same direction so, apart from slight perspective distortion, the problem is essentially a 2-dimensional recognition task.

An image of the train about half way along the track was first used to both calibrate the system and to generate a model of the train. A bounding box around the front section of the train was defined by hand and the image data within this region was used to construct the train model. The distance between the train and the camera was then recorded to define the reference used for calibration. A series of 25 images of the train were then captured at approximately 2 centimetre intervals as it approached the camera. Three of these images are shown in Figure 6.10.

The distance to the train in each image has then been estimated by first locating the train model in the scene and then estimating its scale as described earlier in this chapter.

The located model can be seen in the three example images in Figure 6.10, overlaid in black. Figure 6.11 presents the estimated distance from the camera to the train for each of the images. The solid line represents the actual distance to the train and the dotted lines provide a 5% margin either side of that distance. The distance to the train has been successfully estimated for all of the images within 5% of the actual value.

6.7 Conclusions

As distance measures are used in the construction of pairwise geometric histograms, the representation changes as the scale of the shape data it describes is changed, limiting the use of geometric histograms to recognising shapes of fixed scale. A shape primitive over a range of scale is not described by a single point in the pattern space but by a smooth trajectory through the space. An algorithm has been developed in this chapter which can recognise shape over a range of scale, using geometric histograms, by approximating these trajectories by a small number of example histograms. This has been possible because a single geometric histogram is able to represent a shape primitive over a relatively large range of scale because of the stability of the distance metric as a function of scale.

In this scheme, line segments are classified according to the nearest neighbour in the pattern space as before but now the classification also includes an estimate of the line segment scale. As the scale of the classified line segment is assumed to be the same as the scale of the nearest stored exemplar, this estimate is subject to a large, uniform error. To ensure robust recognition, this error has been accounted for in the construction of the probabilistic Hough transform.

The algorithm has been shown to work over a range of scale from 0.5 to 2.0 for simple shape data. A quantitative analysis of the scale estimate suggests that an error of between 5% and 10% is typical although this does depend to some extent on the shape data. By determining the scale of a known object in a scene it is possible to estimate its distance from the camera. This has been used to track a model train over a series of images.

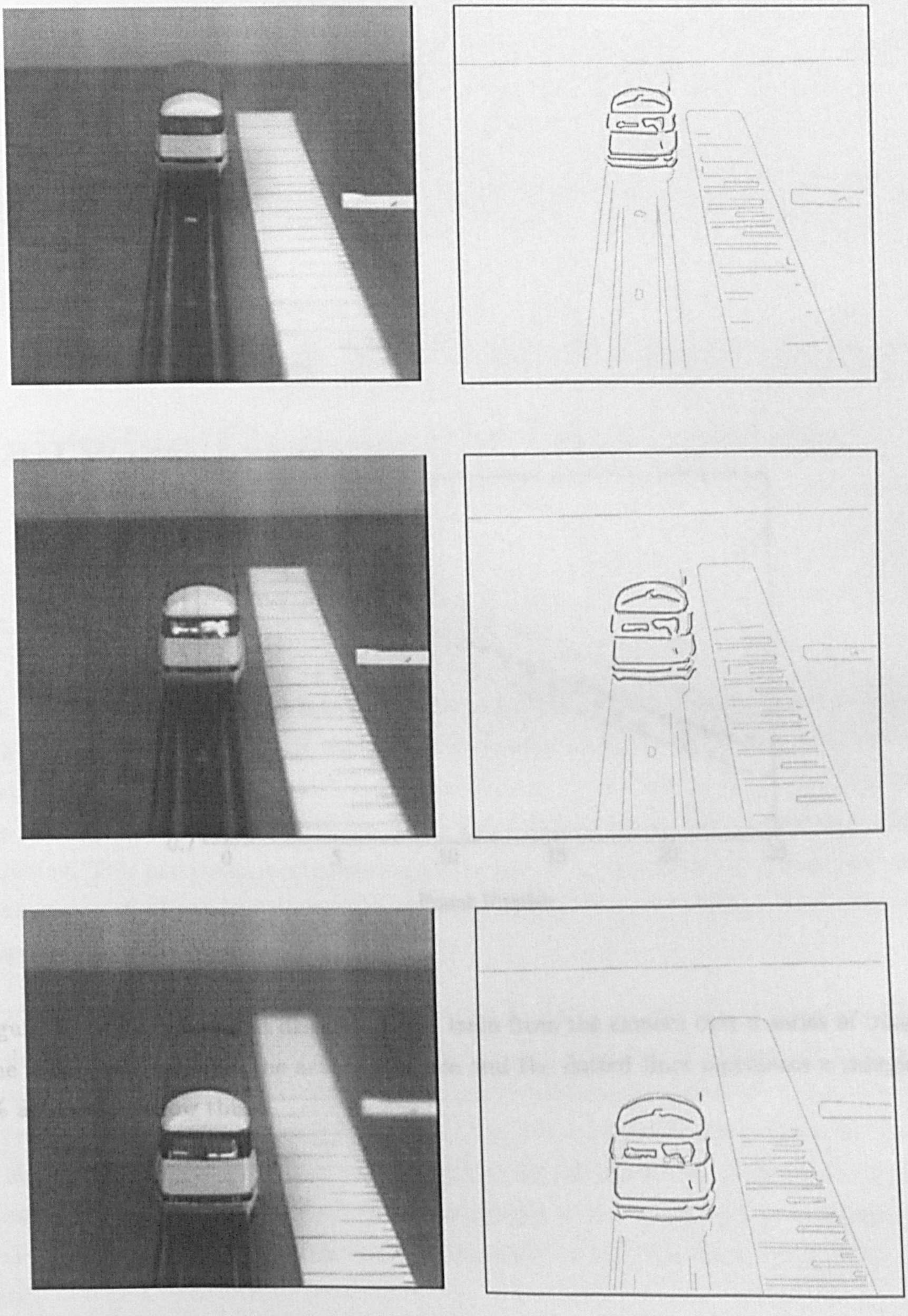


Figure 6.10: Estimating the distance to the train from it scale. The images on the left show the image of the train as it is captured approaching the camera. On the right hand side of each image are the line primitives extracted from the image data (in grey) and the located models of the train (superimposed in black).

Figure 6.11: The estimated distance to the train from the camera over a series of images. The solid line represents the actual distance and the dotted lines represents a margin of 5% above and below this.

Chapter 7

Representing Surface Shape using Pairwise Geometric Histograms

7.1 Introduction

The work presented so far in this thesis has concerned the development and analysis of 2-dimensional shape recognition using pairwise geometric histograms. It has been demonstrated that encoding the distance and angle between 2-dimensional shape line segments provides a powerful shape descriptor which can be used for reliable and efficient shape matching. This particular representation has proved to be very useful but other geometric histograms definitions based on other features and feature relationships have also been proposed [Kumar & Rockett, Evans 94].

Kumar has suggested that edge pixels around the boundary of an object can be represented using the pair of angles defined between the reference edge and all possible pairs of edges within some window. By only using angle measurements this representation is invariant to the scale of the object, although the error on the measurement which defines the amount of blurring to apply to each histogram entry does depend upon scale. Evans also suggested using 3-dimensional line segments recovered using a stereo vision system to produce a 3-dimensional shape representation which is invariant to rigid transformations of the shape data.

The use of line features for 2-dimensional recognition is a good choice because it focuses attention to information rich areas of the image and because all objects can be described in terms of these features. Even objects with smooth surfaces can be described by their bounding contour when always viewed from the same direction. Unfortunately, for 3-

dimensional applications the bounding contour of smooth objects is a function of the viewing direction so the boundary shape is of limited use for recognition. This general problem has motivated researchers to build representations based upon the surface shape of objects for general 3-dimensional recognition tasks.

In this chapter a novel pairwise geometric histogram representation for describing arbitrary surfaces is proposed, enabling the recognition of 3-dimensional objects with arbitrary surface shape. The process of constructing this representation and then matching model and scene surfaces can be summarised as follows:

1. Raw surface data is acquired using a range sensor. A number of data sets taken from different view-points are used when constructing models.
2. The surface is approximated by a triangular mesh. The details of this approximation and the algorithms employed for this are presented in Section 7.2.1.
3. Each triangular facet is represented by a pairwise geometric histogram which records the relationship between this facet and the surrounding facets. This representation is discussed in Section 7.2.2.
4. Correspondences between scene and model facets are found by matching their respective geometric histograms. These *local* correspondences provide hypotheses for the correspondence between the model and scene.
5. *Global* model to scene correspondence is found by finding the transformation that aligns most of the surface data. This is done using a variant of the RANSAC algorithm [Fischler & Bolles 81] which is discussed in section 7.5.

7.2 A Novel Surface Shape Representation

7.2.1 Surface Reconstruction and Approximation

Initially a given surface S, acquired using a range sensor, is described by a set of points samples $P = \{p_1, \ldots, p_N\}$. The points may represent a single view of the surface or a number of different views, for example from different viewpoints around an object. If a number of views are used then the data must be registered so that surfaces common to more than one view are aligned. The point set is then used to construct a triangular mesh approximation $\hat{S}$ to the original surface, where $\hat{S} = \{t_1, \ldots, t_M\}$ and t_i is a triangular facet of the mesh.

It is important to clarify at this stage that the only requirement of the mesh is that it is a good approximation of the surface shape. No assumptions are made about the actual distribution of facets over the surface as this is unlikely to be repeatable. To minimise the amount of memory and computation needed to solve the correspondence problem, the mesh should also contain the smallest number of facets needed to give a good approximation of the surface.

A number of algorithms have been proposed for reconstructing a triangular faceted mesh from a set of points. In the work presented here an initial, regular mesh was constructed from the sampled point data using a reconstruction algorithm by Hoppe *et al* [Hoppe *et al* 92]. The resulting regular mesh was then refined to minimise the number of facets whilst maintaining most of the surface shape using a surface simplification algorithm by Garland and Heckbert [Garland & Heckbert 97].

There are a number of advantages in using a triangular mesh to approximate the surface to be represented instead of more complex features such as quadric patches, the most obvious being efficiency. Constructing a mesh is also significantly more straightforward than segmenting a surface into more complex features. A second important issue is scope. Any surface can be approximated by a triangular mesh but selecting a fixed set of features can impose limitations on the types of surfaces that can be described. Another important issue is that of stability. If surface patches are assigned to different classes based on their shape then borderline cases can result in sudden changes in the representation because of slightly different viewing conditions or noise.

The disadvantage of using a triangular mesh is that it requires many facets to describe surfaces with high curvature to a high degree of accuracy. By statistically modelling the shape error introduced by the triangular shape approximation, it is still possible to obtain a good shape representation when only a relatively small number of facets are used.

7.2.2 Histogram Construction

A pairwise geometric histogram h_i is constructed for each triangular facet t_i in a given mesh which describes its pairwise relationship with each of the other surrounding facets within a predefined distance. This distance controls the degree to which the representation is a local description of shape. The histogram is defined such that it encodes the surrounding shape geometry in a manner which is invariant to rigid transformations of the surface data and which is stable in the presence of surface clutter and missing surface data.

Figure 7.1(a) shows the measurements used to characterise the relationship between facet

t_i and one of its neighbouring facets t_j. These measurements are the relative angle, α, between the facet normals and the range of perpendicular distances, d, from the plane in which facet t_i lies to all points on facet t_j. These measurements are accumulated in a 2-dimensional frequency histogram, weighted by the product of the areas of the two facets as shown in Figure 7.1(b). The weight of the entry is spread along the perpendicular distance axis in proportion to the area of the facet t_j at each distance. To compensate for the difference between the measurements taken from the mesh and the true measurements for the original surface, the entry is blurred into the histogram. For the work presented here a Gaussian blurring function has been used, but further analysis of the surface approximation error is needed to determine a more principled function. Certainly the scale of the blurring function relates to the coarseness of the mesh. The complete pairwise geometric histogram for facet t_i is constructed by accumulating these entries for each of the neighbouring facets.

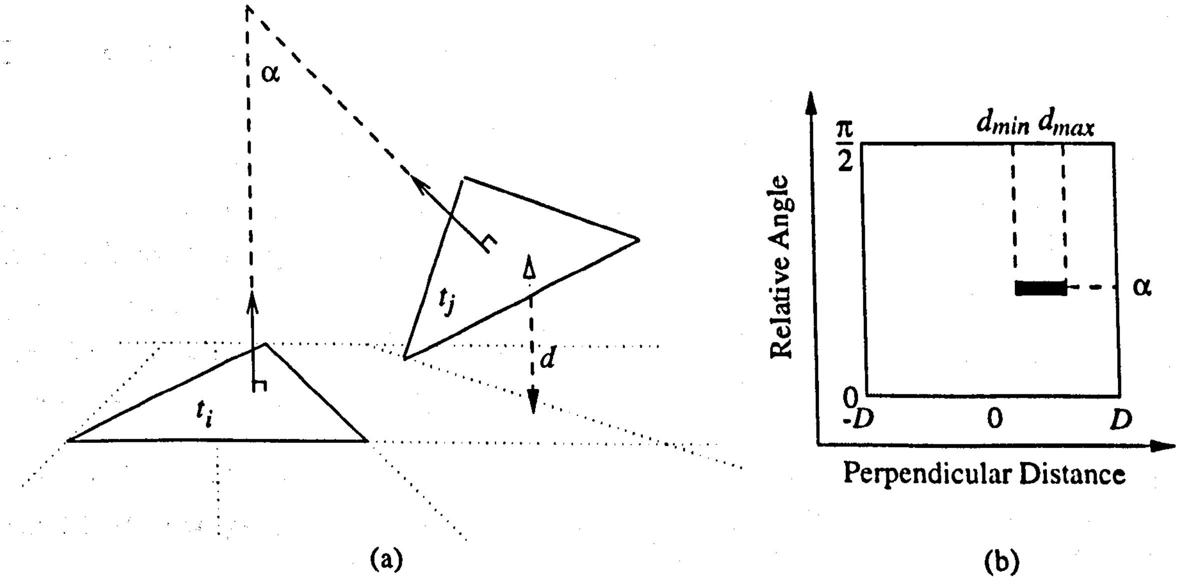
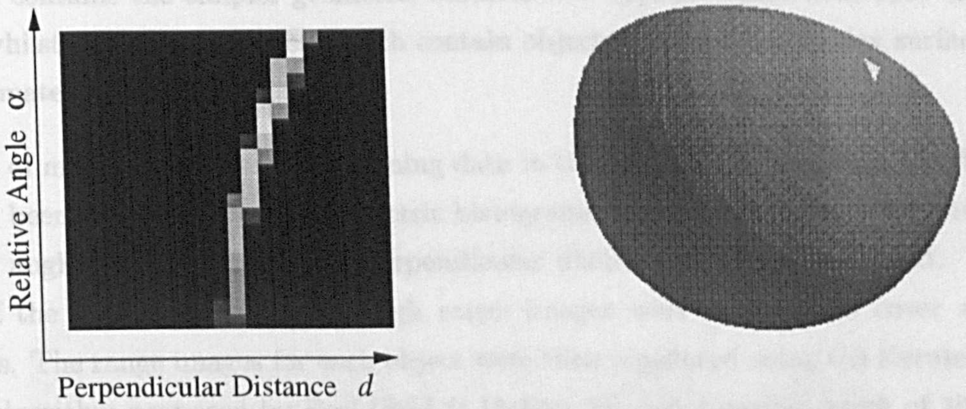


Figure 7.1: (a) The geometric measurements used to characterise the relationship between two facets t_i and t_j. (b) The entry made into the pairwise geometric histogram to represent this relationship.

For clarity, an example of a pairwise geometric histogram is presented in Figure 7.2(a). This has been constructed for the highlighted facet on the hemispherical mesh presented in Figure 7.2(b). Note that the representation only depends upon the surface shape and not on the distribution of facets over the surface. This independence on the distribution of the facets is important because recovering exactly the same mesh for the same surface under different viewing conditions is very unlikely, particularly if there is some surface occlusion.



(a) (b)

Figure 7.2: (a) The geometric histogram that characterises the relationship between highlighted facet and the other facets in the mesh in (b).

7.3 Classification of Scene Surface Features

Given two surface meshes, $\hat{S}^A$ and $\hat{S}^B$, the geometric histogram representation allows correspondences between all facets, t_i^A and t_j^B, from each of the meshes to be determined. A match for facet t_i^A is determined by finding the best match between its respective pairwise geometric histogram and all of the histograms representing the facets in surface $\hat{S}^B$. These *local* correspondences are treated as hypotheses for the correspondence between the two surfaces S^A and S^B.

The similarity, D_{ij}, between two pairwise geometric histograms h_i and h_j is defined using the Bhattacharyya metric as before. This is given by the expression:

$$D_{ij} = \sum_{\alpha,d} \sqrt{h_i(\alpha,d)}\sqrt{h_j(\alpha,d)} \tag{7.1}$$

7.4 Experiments: Classification of Scene Surface Features

The experiment presented here has been devised to demonstrate the effectiveness of using the proposed pairwise geometric histogram representation for classifying scene surface mesh facets. Figure 7.3 presents the four test scenes used in this experiment which contain a selection of objects, some with simple geometric surfaces and some with free-form surfaces. Each scene was generated by taking a single range image using a laser striper and then approximating the acquired surface points by a triangular faceted mesh. Scene

1 which contains the simpler geometric surfaces was approximated with 1000 triangular facets whilst the other 3 scenes, which contain objects with more complex surfaces, were approximated with 2000 facets.

The set of model objects used as training data in this experiment, presented in Appendix E, have been represented using geometric histograms with a resolution of 20 distance bins and 20 angle bins. A maximum perpendicular distance of 15mm was used. To build each of the first three models enough range images were acquired to cover all of the surfaces. The range images for each object were then registered using the Iterated Closest Point algorithm proposed by Besl [Besl & McKay 92] and a surface mesh of 1000 facets constructed. The remaining three models were each constructed from a pair of range images taken from different sides of the object and registered by hand. These surfaces were then approximated by 2000 facets each.

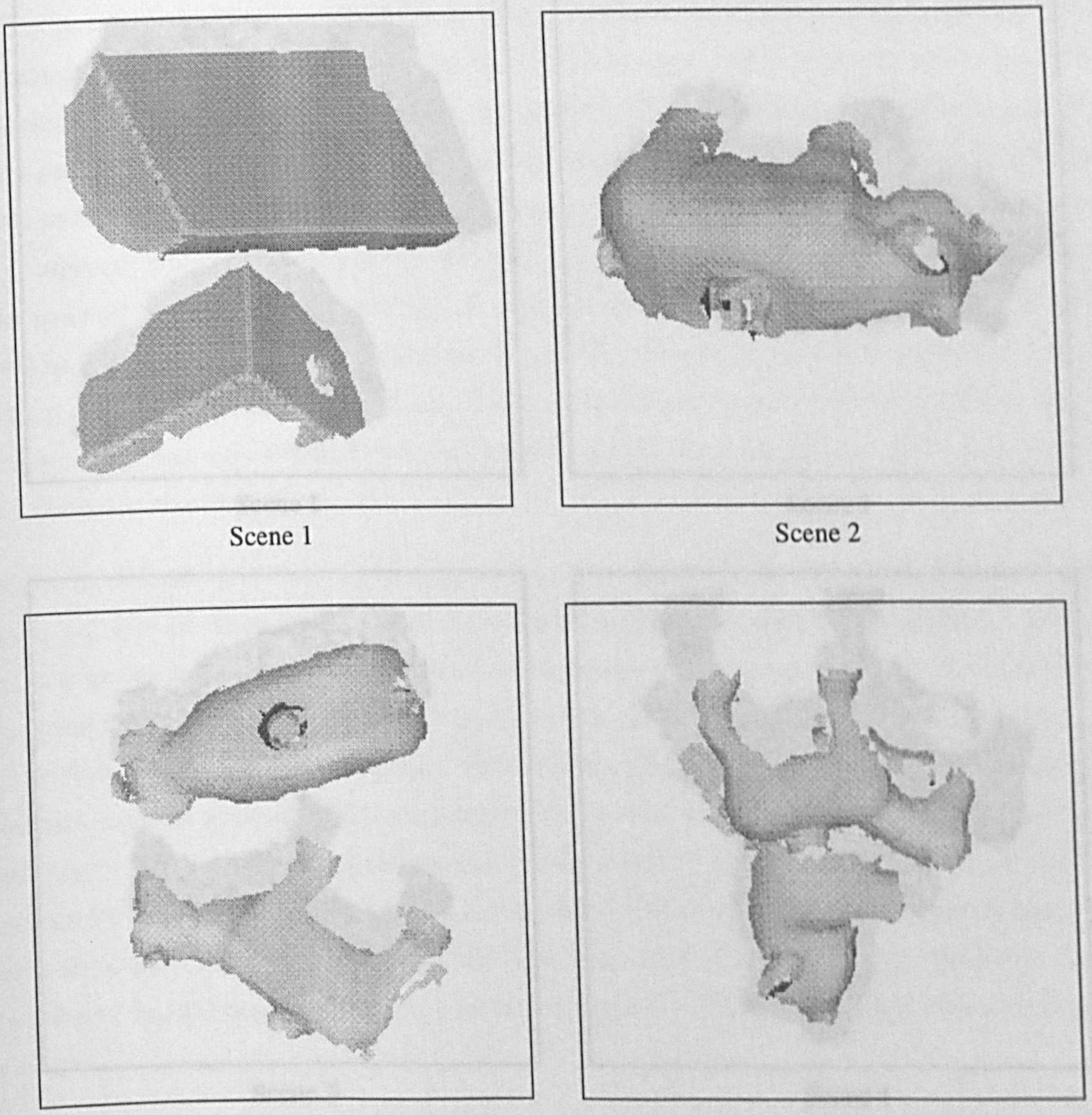
Figure 7.4 presents the classification results for all of the scene mesh facets when matched to the models. Each surface facet has been coloured according to the class of the model to which the best matching facet belongs. Surface facets which have no match to any of the models present in the scene have been coloured in black. Although this colour coding indicates which scene facets have matched to the correct model it does not necessarily mean that the scene facet has matched to an appropriate facet on that model. This is implied later, however, when the pose of the models in the scene is determined successfully.

In general the surface facet classification has performed relatively well in all of the scenes. As might be expected, better results are obtained on flatter surfaces where the estimation of the surface normal is more repeatable. Although histogram entries are blurred to account for variation in the surface normal direction the blurring function used was chosen for simplicity rather than correctness. Further analysis of this variability is needed to determine a more suitable blurring function and this should improve the classification of facets on more curved surfaces.

7.5 Hypothesis Combination and Determining Object Pose

Good matches between scene and model facets provide evidence for the presence of known models in the scene and provide constraints on the pose of those models. As with the recognition of 2-dimensional shape data considered earlier, the recognition process is completed by combining these local hypotheses into a *global* scene interpretation.

For 2-dimensional shape data a probabilistic Hough transform was employed as a robust



Scene 1

Scene 2

Scene 3

Scene 4

Figure 7.3: These four test scenes contain a number of objects with both simple geometric and free-form surfaces.

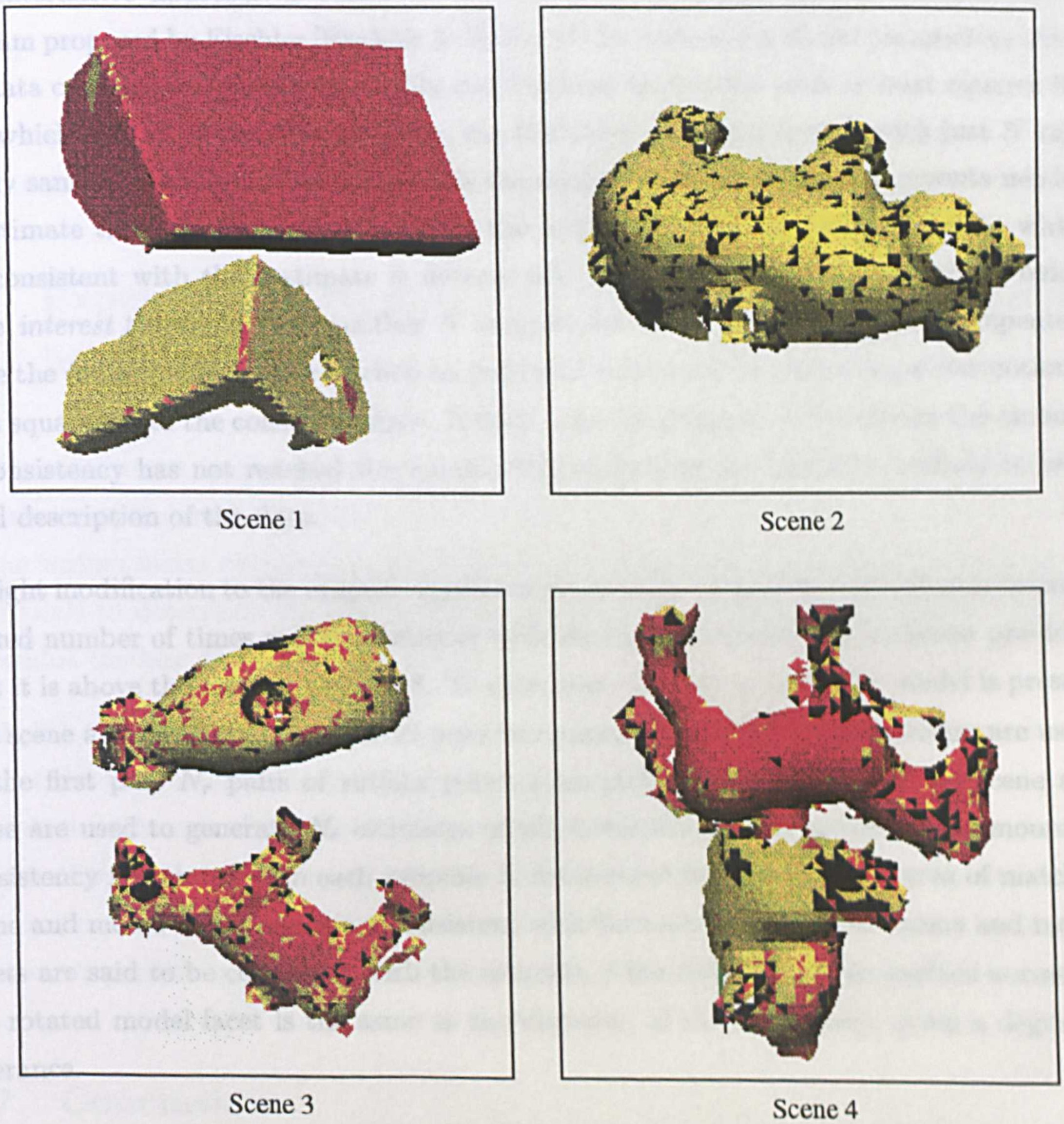


Figure 7.4: Classification of the scene surface facets. Each facet has been coloured according to the class of the model to which the best matching model facet belongs. Facets matching to model not present in the scene have been coloured in black.

estimator of pose and by weighting votes by line segment lengths a suitable measure of evidence was simultaneously obtained. Although this approach can be extended to estimate the pose of 3-dimensional surface meshes, the increase in computation is significant so an alternative approach has been taken.

The alternative approach is based on the RANSAC (Random Sample Consensus) algorithm proposed by Fischler [Fischler & Bolles 81] for estimating model parameters when the data contains many outliers. Unlike conventional approaches such as least squares fitting which uses all of the data available, the RANSAC algorithm begins with just N randomly sampled measurements where N is the minimum number of measurements needed to estimate the model parameters. Given the initial estimate, the amount of data which are consistent with that estimate is determined. If the amount of consistency is below some *interest* threshold then another N samples are taken and the process is repeated. Once the *interest* threshold is reached an improved estimate is formed using a conventional least squares fit to the consistent data. If after a specified number of iterations the amount of consistency has not reached the interest threshold then the model is unlikely to be a good description of the data.

A slight modification to the original algorithm is used here in that the algorithm is iterated a fixed number of times and the estimate with the highest consistency is chosen provided that it is above the interest threshold. To determine whether a particular model is present in a scene and to estimate the model pose two passes of the RANSAC algorithm are used. In the first pass N_r pairs of surface patches are picked at random from the scene and these are used to generate N_r estimates of the orientation of the model. The amount of consistency associated with each estimate is determined by summing the area of matched scene and model facets which are consistent with the estimate. Matched scene and model facets are said to be consistent with the estimate if the direction of the surface normal of the rotated model facet is the same as the direction of the scene facet, given a degree of tolerance.

In the second pass of the RANSAC algorithm N_t triplets of scene facets are picked at random from the set of scene facets which were consistent with the best estimate in the first pass of the algorithm. An estimate of the translation that aligns the model and scene is then determined for each triplet and the amount of consistency is determined as before. In this case, matched scene and model facets are said to be consistent if the perpendicular distance between the translated model facet and the scene facet is zero, within some tolerance. If the estimate with the maximum overall consistency is above the interest threshold then the model is said to be present in the scene and its pose estimate is improved using least squares fitting.

	Scene 1	Scene 2	Scene 3	Scene 4
Cylinder	1034.231	0	0	0
Block	52.127	12.408	0	0
Widget	781.701	0	0	0
Calf	0	30.283	506.636	0
Pig	23.4	929.196	724.893	256.541
Pony	0	20.696	27.231	590.585

Table 7.1: The area of each scene in millimetres^2 which was found to be consistent with each of the known models.

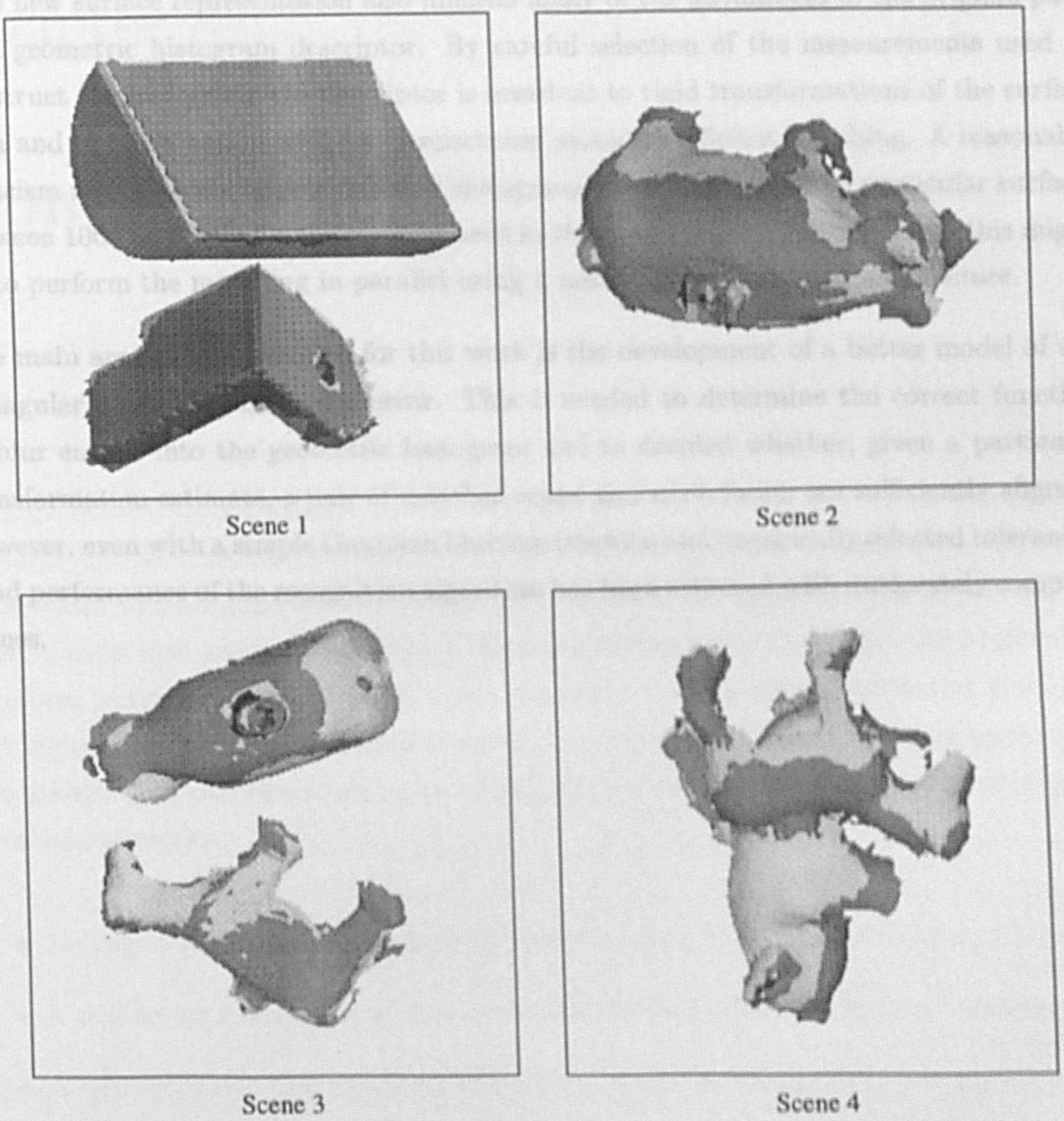
7.6 Experiments: Hypothesis Combination and Determining Object Pose

In this experiment, evidence to suggest the presence of the model objects in each of the four scenes in Figure 7.3 is accumulated and the pose of these models is estimated based on the surface facets classified earlier. For each scene the RANSAC algorithm was run for 5000 trials to determine the best orientation of each model and then for 10000 trials to determine the best translation of each model.

Table 7.1 and Figure 7.5 present the object recognition and pose estimation results for each of the four scenes. The table presents the area of each scene which was found to be consistent with each of the six models, providing evidence for the presence of the models in each of the scenes. The figure presents all of the detected models, in the lighter shade, superimposed over the scene data, in the darker shade, at the estimated poses. In all cases the models present have been detected successfully and the pose of each model determined.

7.7 Conclusions

In this chapter a novel approach for representing 3-dimensional surface data using pairwise geometric histograms has been proposed and the representation has been demonstrated in a surface based object recognition application. In the original work on pairwise geometric histograms, Evans [Evans 94] proposed that 3-dimensional line segments recovered using a stereo vision system can be represented using a histogram descriptor and he provides a 3-dimensional recognition demonstration. This approach is suitable for objects with distinct edges which can be recovered from a scene but is not suitable for objects with smooth surfaces. This general problem has motivated researchers to investigate the use



Scene 1

Scene 2

Scene 3

Scene 4

Figure 7.5: Recognition and pose estimation results for the test scene data. The original scene data is shown in the darker shade and the detected models are shown in their estimated poses in the lighter shade.

of surfaces for object classification but the common representations are limited to certain surfaces types such as planes or quadrics. In contrast to this, the representation presented here is suitable for arbitrary surface types and as such, provides a possible solution to an important problem in computer vision.

This new surface representation also inherits many of the advantages of the original pairwise geometric histogram descriptor. By careful selection of the measurements used to construct the histogram, the descriptor is invariant to rigid transformations of the surface data and in combination with its compactness promotes efficient matching. A reasonable criticism would be the large number of histograms needed to describe a particular surface, between 1000 and 2000 for the surfaces used in the experiments. The answer to this might be to perform the matching in parallel using a neural network type of architecture.

The main area of improvement for this work is the development of a better model of the triangular mesh approximation error. This is needed to determine the correct function to blur entries into the geometric histogram and to decided whether, given a particular transformation estimate, a pair of matched scene and mesh facets are sufficiently aligned. However, even with a simple Gaussian blurring function and empirically selected tolerances good performance of the recognition algorithm has been achieved with moderately complex scenes.

Chapter 8

Conclusions

8.1 Introduction

A large number of algorithms have been developed for solving the object recognition problem but it is debatable whether any of these are reliable enough or well enough understood to be integrated into a general purpose vision system. One of the central goals of computer vision research since the dawn of the field has been the development of an object recognition system which meets the needs of real scene interpretation problems and which provides the information needed for system integration. To be a useful part of a general purpose vision engine an object recognition scheme must have a number of essential properties.

- An object representation capable of representing a large class of objects.
- A concise representation with good invariance properties for efficient matching.
- A representation and matching scheme which are insensitive to object variation and scene clutter providing reliable recognition.

Although originally limited to the representation of 2-dimensional shape, the pairwise geometric histogram representation has been shown to solve many of the problems associated with object recognition and has been used here as the basis of further work. Geometric histograms provide a local descriptor of shape which is compact and stable to shape variability, permitting efficient and robust matching. The motivation for the work presented in this thesis has been to address some of the criticisms of the original algorithms and to quantify some of the representation's important properties. The next section provides a review of the main contributions made by this work and this is followed by a number of suggestions for future research.

8.2 Contribution

8.2.1 Critical Review of Recognition Algorithms

A large number of object recognition algorithms using a variety of different approaches can be found in the literature. Understanding these algorithms and appreciating their relative merits is a considerable task, particularly for newcomers to the field. In Chapter 2 a selection of the most important algorithms have been critically reviewed and a table of important properties compiled so that a coarse comparison can be made. Although the list of algorithms reviewed is not exhaustive it is intended that most of the main approaches to 2-dimensional shape recognition are included. Other algorithms are likely to be variants on these.

The most important conclusion to be drawn from this study is the importance of image features for robust recognition in real scenes exhibiting background clutter, lighting artifacts such as shadows and specularities and partial object occlusion. Techniques which depend upon good segmentation of image regions perform poorly under these conditions. The problem with using image features is the need to consider all viable pairings between model and scene features. Even when the number of model and scene features is relatively small the number of viable pairings can become very large. The application of pairwise geometric histograms provides a potential solution to this problem by providing a rich feature representation which can be used to find a small number of model to scene feature pairings.

8.2.2 Probabilistic Hough Transform Implementation

The pairwise geometric histogram representation allows shape features to be represented and classified in an efficient manner. However, to recognise complete objects, appropriately classified features must be arranged in a way which is consistent with stored models. The problem is formulated as a parameter estimation problem, in this case the parameters are the model pose, and recognition depends upon finding enough data which is consistent with a particular estimate.

This estimation problem has been implemented here using a probabilistic Hough transform in which votes are placed in a parameter space, representing the model pose, for pairs of matched model and scene line segments. The adoption of the probabilistic Hough transform is a considerable improvement over the generalised Hough transform which was used previously, allowing variability in the position of shape features to be accounted for correctly. This not only improves the robustness of the technique but also allows the position of models to be detected to a much greater accuracy and provides an estimate of the error on the estimated position. In fact, this approach is equivalent to using a robust least squares fit of the model line segments to the scene line segments. Experimental results suggest that typical errors of 0.2 pixels can be expected and that it is possible to predict this error with a factor of 2.

8.2.3 An Analysis of Reliability

There are a variety of reasons why a quantitative measure of the reliability of a classification algorithm is needed, the most significant concerns system integration. If a classification algorithm is to be integrated into a larger system then the performance of the complete system can be determined if the consequence of the classification error is propagated through every stage of decision making. The problem of estimating classification error has been considered by many researchers and different approaches have been suggested but these give no indication of how the reliability scales as the number of training models is increased. An alternative approach is adopted here in which the measurements used in forming classification decisions are identified and their statistical variation for different sets of shape data is observed. Provided that these statistics are representative of all shape data, the performance of the algorithm as a function of the number of stored models can be predicted. The classification error which is derived from these statistics has been used to estimate the error for two different sets of shape data.

One of the important observations made during this analysis was that scene feature classification based on only the best match becomes unreliable as the number of models is increased. To avoid this problem it has been suggested that scene features are given a number of potential class labels, based on a number of the best matches. These additional classes are resolved to obtain a single classification by finding global consistency using the probabilistic Hough transform.

A further use of a quantitative measure of reliability is to measure the effect of changes to the algorithm or the effect of varying algorithm parameters. This is demonstrated by plotting error curves for different geometric histogram resolutions, allowing an appropriate resolution to be selected to give a specified level of performance.

8.2.4 Estimating Capacity

One of the requirements of a general purpose object representation is the ability to describe a large number of different objects uniquely. This quantity is commonly referred to as *capacity.* Estimating the capacity of an object representation is essentially a counting exercise but is complicated by the need to define what is meant by a unique object and to ensure that only viable object descriptions are included. Two different approaches have been developed to estimate the capacity of the pairwise geometric histogram representation and applied to two different classes of shape data. It is not surprising to find that the capacity depends on the complexity of shape data being represented. What is interesting is that the complexity of the shape data can be made explicit by measuring its intrinsic dimensionality. Experimental results suggest that for reasonably complex shapes many thousands of different shapes can be described uniquely but for simple geometric objects the capacity is much lower.

8.2.5 Extensions for Scale

The pairwise geometric histogram representation utilises distances between shape features in its description of shape with the consequence that it is dependent upon shape scale. This has attracted a fair amount of criticism as it restricts recognition to objects of a fixed size, or more significantly, at a fixed distance from the camera.

Although shapes are represented at a specific scale, shape data at similar scales are described by similar geometric histograms. This observation has been used as the basis for representing shape data over reasonable ranges of scale by storing a few samples of each shape primitive at different scales. The construction of the probabilistic Hough transform has been modified to account for the variation in shape scale and to correctly handle errors in the estimation of the scale of each shape primitive. Results have been presented which demonstrate that the technique works and that, over a range of scale from 0.5 to 2.0, the scale of shapes can be typically estimated within 5%. One of the uses of a scale estimate is to determine the distance from a known object to the camera. A demonstration is presented in which a model train is tracked over a sequence of images as it approaches a fixed camera.

8.2.6 Representing Surface Shape

Although the original pairwise geometric histogram representation was used to describe 2-dimensional line features, other features can also be represented in a similar manner. Interest in 3-dimensional object recognition based on surface shape has motivated the development of a novel representation for 3-dimensional surface data. The representation has been successfully demonstrated in a surface based 3-dimensional object recognition application involving objects with a range of surface types.

In the same way that curves are approximated by short line segments in the original algorithm, surfaces acquired using a range finder are approximated by small triangular facets in this algorithm. Geometric histograms can then be constructed to represent these facets allowing the correspondence between model and scene surfaces to be determined. The representation is suitable for all surface types, including free-form surfaces, and unlike many surface based representations does not require any surface segmentation or classification which can be unreliable.

8.3 Future Work

During the course of this work a number of topics for continued development, beyond the scope of this thesis, have been identified. These topics are described in brief in this section.

8.3.1 Object Classification Error

The analysis in Chapter 4 provides a technique for estimating the reliability of the classification of shape primitives based on pairwise geometric histograms. The effect of this classification error on the recognition of complete objects was not considered however. To determine the probability that objects are misclassified, it is necessary to propagate the feature classification error through the probabilistic Hough transform formation process. This analysis will not only provide classification error figures but should also provide a better mechanism for selecting important Hough transform parameters such as the parameter space resolution and the peak threshold.

It was suggested in Chapter 4 that scene features are given a number of class labels, increasing linearly with the number of stored models, to ensure reliable classification. These additional labels result in more entries being made into the Hough transform parameter space and this will have some influence on the classification of complete objects. Further analysis is needed to determine whether this will have a detrimental effect on the recognition performance as the number of model objects is increased.

8.3.2 Surface Based Representation

A novel surface representation based on pairwise geometric histograms was presented in Chapter 7 and preliminary object recognition results using this descriptor were demonstrated. To ensure that the representation is a good descriptor of the underlying surface data, it is necessary to have a good model of the error introduced by approximating raw surface data using a triangular mesh. Further analysis of this process is needed and perhaps alternative triangulation algorithms with better statistical properties can be identified or developed.

A similar analysis to the one presented in Chapters 4 and 5 for the original pairwise geometric histogram representation should be carried out for the new surface shape representation. This will provide a better understanding of the representation and provide a more principled mechanism for selecting parameters such as the histogram resolution.

An important problem when reconstructing object models from multiple sets of range data is finding the transformation that aligns all of the sets into a common coordinate frame. This is the registration problem. Although some algorithms exist, for example the ICP algorithm was used to construct some of the models found in Appendix E, none of them provide a complete solution in all cases. Because the proposed geometric histogram representation can be used to find correspondences between surfaces it may prove to be a useful tool in solving the registration problem.

8.3.3 Non-Exhaustive Search Strategies

The matching of model and scene pairwise geometric histograms is very well suited to parallel implementation but in practice highly parallel machines are expensive and difficult to program. An alternative way to reduce the amount of time needed to classify scene primitives might be to develop an alternative matching strategy.

One possible approach might be to limit the amount of matching by directing the search. Rather than matching all scene primitives at once a single scene primitive is picked at random and classified. Having identified the class of this primitive, it should be possible to search the image for other primitives from the same shape. A second possibility is to first recognise shapes at a very coarse scale and use this result to limit search at an increased resolution. At coarse scales the shape data can be described by fewer shape primitives and a small number of geometric histograms. Which ever strategy is adopted, it is essential that the consequence of using a non-exhaustive search on the classification error is understood.

8.3.4 Appearance-based 3-dimensional Object Recognition

The work presented in Chapter 7 proposes a 3-dimensional shape representation in which the 3-dimensional structure is represented explicitly. Other researchers have suggested that 3-dimensional objects can be represented solely using their 2-dimensional appearance. Typically a number of characteristics views of the object are stored and intermediate views are generated by interpolation. In the same way that objects at different scales were represented by storing a number of examples in Chapter 6, it may also be possible to represent fully 3-dimensional objects in this way. The important characteristic which may permit this is the fact that a geometric histogram representing a particular feature varies smoothly as the view-point is changed.

8.4 Afterword

The original aim of the work presented in this thesis was to investigate the existing approach to object recognition using pairwise geometric histograms and to develop the method further. Initially it was anticipated that the investigation would identify areas for improvement and the main part of the work would involve developing these improvements. Certainly areas for improvement were identified and developed but the discussion of these only cover about half of the work in the thesis. The reason for this was that during the investigation of the original approach the importance of algorithm performance evaluation became clear. This issue covers the remaining part of the work.

One of the early criticisms for the representation was its lack of scale invariance. By providing a solution to the problem of recognising shape over ranges of scale the approach can now be viewed as a general solution to the recognition of arbitrary 2-dimensional shape in complex scenes.

The use of geometric histograms for representing shape is not limited to the specific implementation proposed originally but provides a more general method for solving shape classification problems. As an example of this, a geometric histogram representation for

3-dimensional surface shape has been proposed and initial experiments show the representation being successfully applied to recognition problems. This not only demonstrates the generality of the geometric histogram approach but provides a useful solution for real 3-dimensional problems.

Although the Hough transform has been around for some time it is still a valuable tool for the vision researcher, and in fact for anyone interested in robust parameter estimation problems. This is evident from the frequency that the Hough transform still appears in journals and conference papers. The insight that the Hough transform is closely related to maximum likelihood statistics is an important one and has allowed a more rigorous implementation of object pose estimation. Anyone interested in using the Hough transform should consider the benefits gained by taking this more rigorous approach.

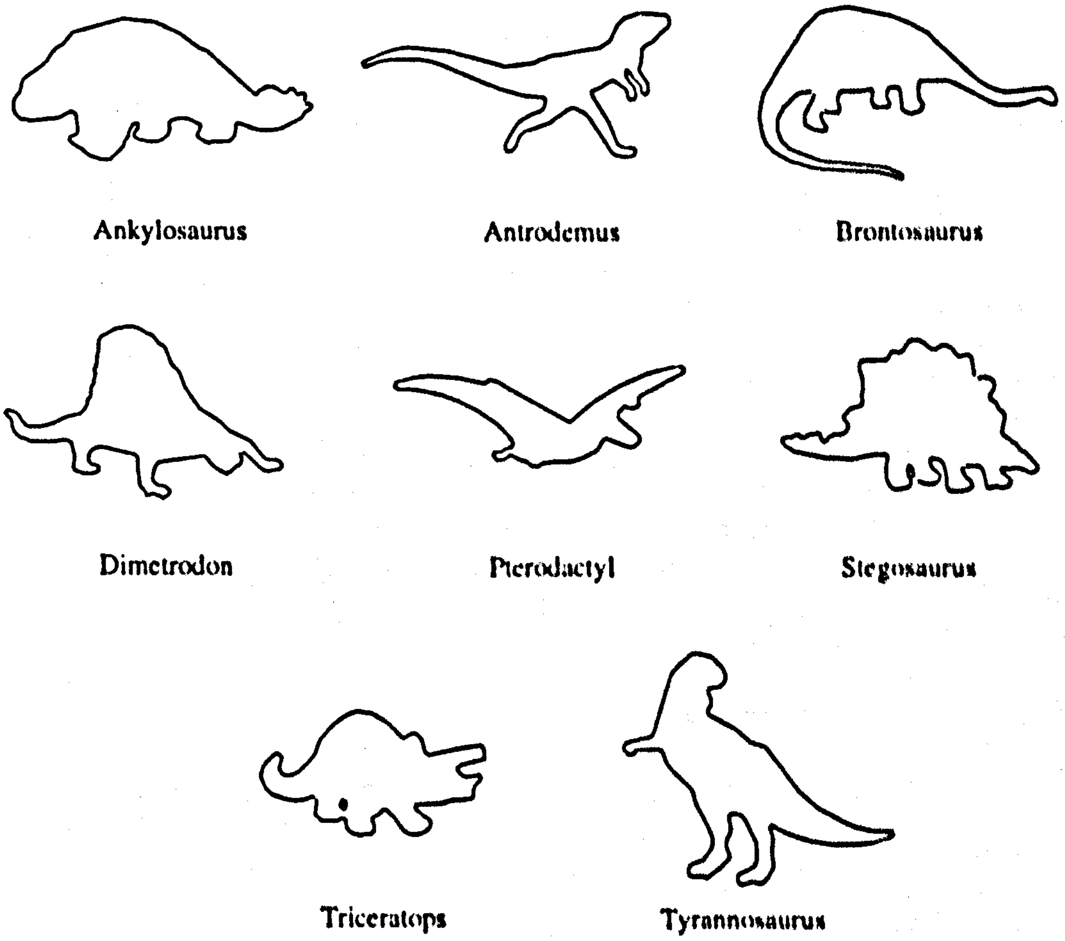
As is often the case with research, and one of the factors which makes research an interesting occupation, the direction that the work takes can be unpredictable. What began as a conventional approach to quantify the error rate of a classification system, in this case the classification of shape, led to a more general look at performance evaluation. In particular, it has been proposed that the issue of scalability, which has largely been neglected by the designers of vision algorithms, be adopted as an important indication of performance for object recognition systems.

The pairwise geometric histogram approach is now in a very strong position after this treatment. Gross predictions can now be made about its performance under varying conditions and the steps needed to be taken to ensure reliable performance can be determined. As a general statement, it is fair to predict that for shape data of similar complexity to that used in this work, the approach is suitable for recognition tasks involving many thousands of different objects.

Although the demonstration of the pairwise geometric histogram algorithm on relatively complex scenes and the analysis of the algorithms scalability are evidence that the approach is an important contribution to the field of object recognition, a more rigorous comparison with other approaches is still needed. This requires the type of evaluation suggested in this thesis to be performed on existing and future object recognition techniques. The author leaves this as a challenge to his colleagues.

Appendix A

Database of 2D Shape Models



Ankylosaurus

Antrodemus

Brontosaurus

Dimetrodon

Pterodactyl

Stegosaurus

Triceratops

Tyrannosaurus

Figure A.1: 2-Dimensional silhouette, shape data: Dinosaurs.

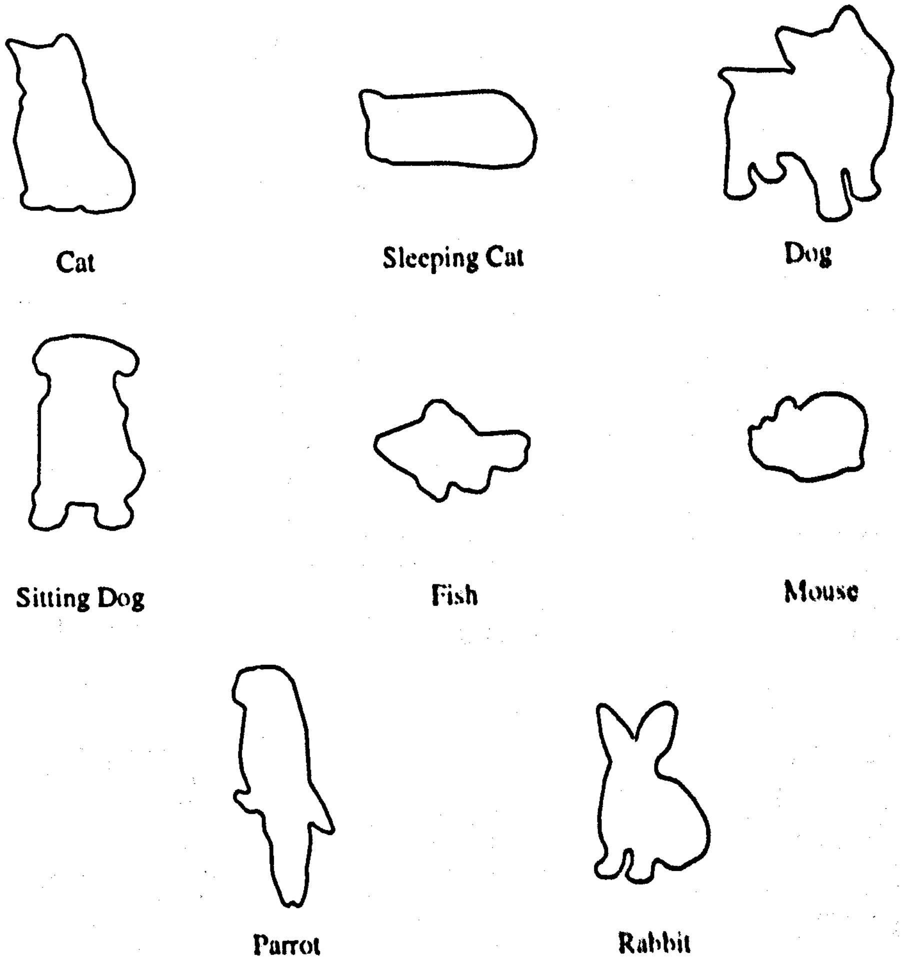


Figure A.2: 2-Dimensional silhouette, shape data: Pets.

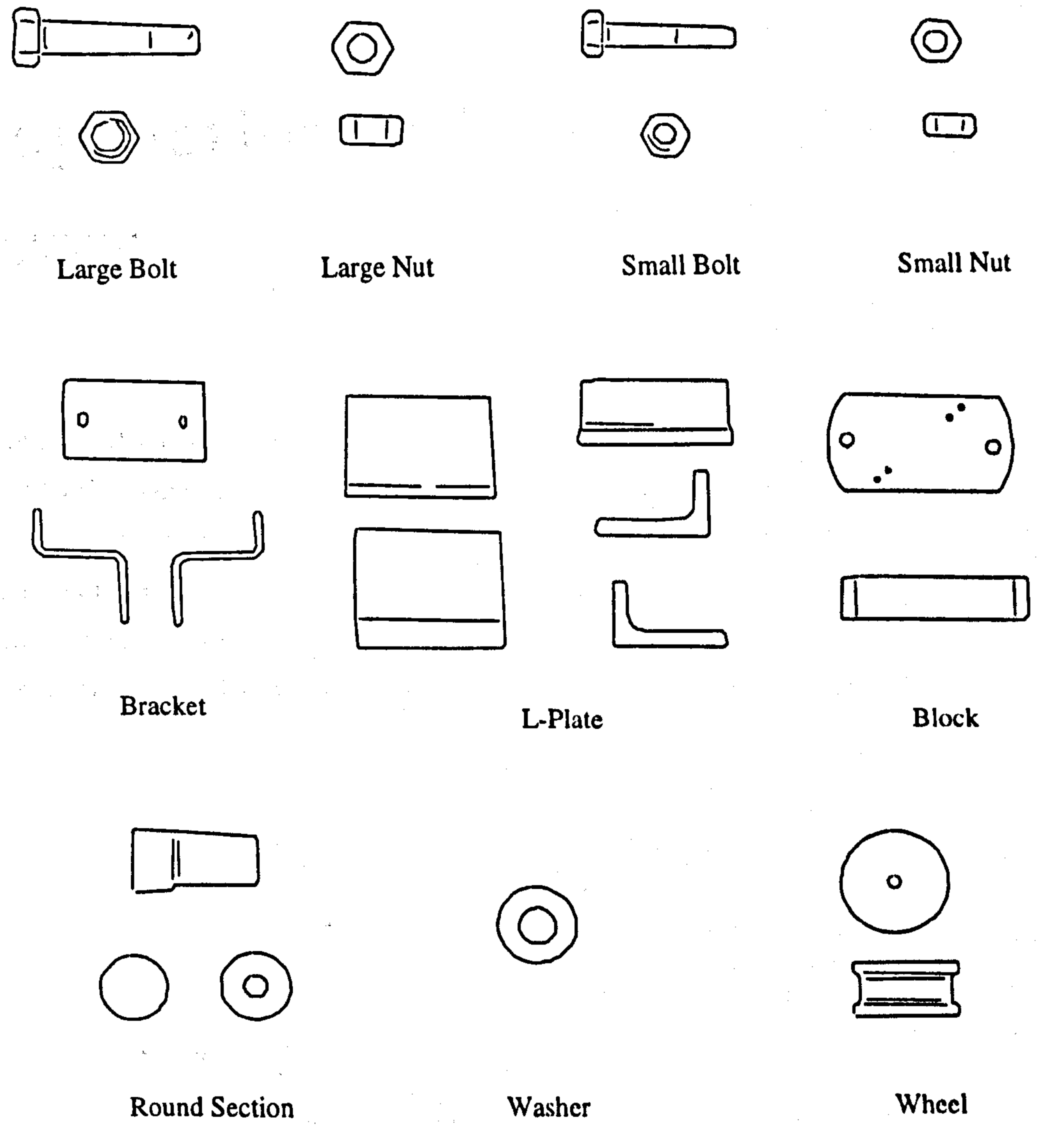


Figure A.3: 2-Dimensional views of real mechanical parts.

Appendix B

Propagation of Line End Point Errors

Each classified scene line segment constrains the position of the object to a line which is parallel to the scene line at an appropriate distance. Any pair of scene lines which belong to the same object and are not parallel define constraints which intersect at the position of the object, as shown in the figure below. If the endpoints of the scene lines are subject to some error, represented here by a covariance matrix Σ_{end}, we would like to calculate the error on the point of intersection, described by the covariance matrix Σ_{int}.

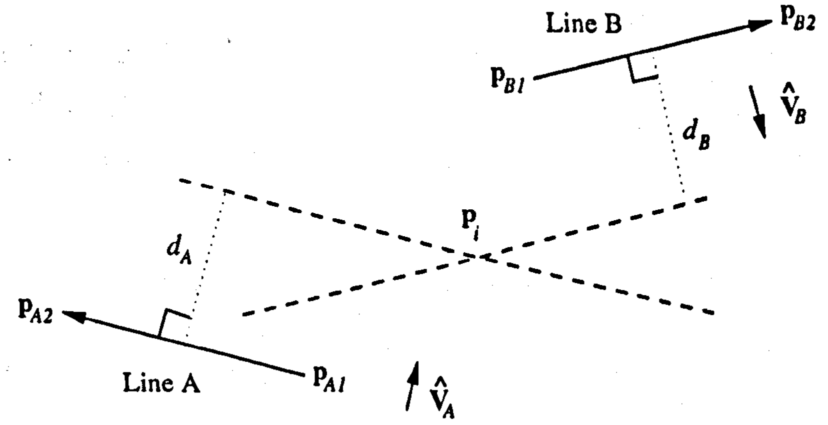


Figure B.1: The constraints imposed by a pair of objects lines intersect at the estimated position of the object.

First of all, the point of intersection is determined from the line endpoints and the perpendicular distances d_A and d_B. The line parallel to "Line A" at a perpendicular distance d_A can be described using the expression:

$$(\mathbf{p}_i - \mathbf{p}_{A1}).\hat{\mathbf{v}}_A = d_A \tag{B.1}$$

Expanding this and rearranging gives:

$$\mathbf{p}_i.\hat{\mathbf{v}}_A = d_A + \mathbf{p}_{A1}\hat{\mathbf{v}}_A \tag{B.2}$$

Similarly for "Line B".

$$(\mathbf{p}_i - \mathbf{p}_{B1}).\hat{\mathbf{v}}_B = d_B \tag{B.3}$$

Which again can be expanded and rearranged to give:

$$\mathbf{p}_i.\hat{\mathbf{v}}_B = d_B + \mathbf{p}_{B1}\hat{\mathbf{v}}_B \tag{B.4}$$

Expressions B.2 and B.4 define a pair of simultaneous equations which can be expressed in matrix form and then solved.

$$\begin{bmatrix} \hat{v}_{A_x} & \hat{v}_{A_y} \\ \hat{v}_{B_x} & \hat{v}_{B_y} \end{bmatrix} \begin{bmatrix} p_{i_x} \\ p_{i_x} \end{bmatrix} = \begin{bmatrix} d_A + p_{A1_x}\hat{v}_{A_x} + p_{A1_y}\hat{v}_{A_y} \\ d_B + p_{B1_x}\hat{v}_{B_x} + p_{B1_y}\hat{v}_{B_y} \end{bmatrix} \tag{B.5}$$

The change in the point of intersection, given changes in the position of the line endpoints, can be predicted using a Taylor expansion of $\mathbf{p}_i$. For small endpoint errors this function will be approximately linear and the intersection error can be expressed as:

$$\Sigma_{int} = \nabla \mathbf{p}_i^T \Sigma_{end} \nabla \mathbf{p}_i \tag{B.6}$$

Where $\nabla \mathbf{p}_i$ is the Jacobian matrix.

$$\nabla \mathbf{p}_i = \begin{bmatrix} \frac{\partial p_{i_x}}{\partial p_{A1x}} & \frac{\partial p_{i_x}}{\partial p_{A1y}} & \frac{\partial p_{i_x}}{\partial p_{A2x}} & \frac{\partial p_{i_x}}{\partial p_{A2y}} & \frac{\partial p_{i_x}}{\partial p_{B1x}} & \frac{\partial p_{i_x}}{\partial p_{B1y}} & \frac{\partial p_{i_x}}{\partial p_{B2x}} & \frac{\partial p_{i_x}}{\partial p_{B2y}} \\ \frac{\partial p_{i_y}}{\partial p_{A1x}} & \frac{\partial p_{i_y}}{\partial p_{A1y}} & \frac{\partial p_{i_y}}{\partial p_{A2x}} & \frac{\partial p_{i_y}}{\partial p_{A2y}} & \frac{\partial p_{i_y}}{\partial p_{B1x}} & \frac{\partial p_{i_y}}{\partial p_{B1y}} & \frac{\partial p_{i_y}}{\partial p_{B2x}} & \frac{\partial p_{i_y}}{\partial p_{B2y}} \end{bmatrix} \tag{B.7}$$

Appendix C

Classification Error for Multiple Hypotheses

Given a set of class exemplars, $\{\bar{\mathbf{x}}_i : 1 \leq i \leq N\}$, and a test pattern, $\mathbf{x}_i$, drawn at random from one of those classes, the pairwise error, ε_p, is defined as the probability that $\mathbf{x}_i$ is closer to a randomly selected class exemplar than its own class exemplar. We wish to calculate the probability, ε, that the correct class exemplar is not within the nearest n neighbours. This is the probability of misclassification when test patterns are associated with multiple classes based on the nearest n neighbouring exemplars.

The probability that the test pattern will be closer to its own class exemplar than a randomly selected one is $1-\varepsilon_p$ and, if each class exemplar is assumed to be an independent sample from the distribution of potential patterns, the probability, $P(0)$, that no incorrect class exemplars will be nearer over $N-1$ trials is:

$$P(0) = (1 - \varepsilon_p)^{N-1} \tag{C.1}$$

Similarly, the probability, $P(1)$, that only one incorrect exemplar will be nearer to the test pattern than it is to its own class exemplar is:

$$P(1) = (1 - \varepsilon_p)^{N-2}\varepsilon_p \begin{pmatrix} N-1 \\ 1 \end{pmatrix} \tag{C.2}$$

The first two factors of this expression imply an ordered sequence of trials in which the first $N-2$ trials are successful followed by a single failure. The combinatorial factor accounts for the different ordering in which these events can occur.

In general, the probability, $P(i)$, that the test pattern is closer to i randomly selected class exemplars than it is to exemplar of the class from which it was drawn is:

$$P(i) = (1 - \varepsilon_p)^{N-1-i} \varepsilon_p^i \begin{pmatrix} N-1 \\ i \end{pmatrix} \qquad \text{(C.3)}$$

The probability, $P(< n)$, that the test pattern is closer to less than n randomly selected exemplars is then:

$$P(< n) = P(n-1) + P(n-2) + ... + P(0) \qquad \text{(C.4)}$$

An expression for $P(< n)$ can be derived by substituting Expression C.3 into Expression C.4.

$$P(< n) = \sum_{i=0}^{n-1} (1 - \varepsilon_p)^{N-1-i} \varepsilon_p^i \begin{pmatrix} N-1 \\ i \end{pmatrix} \qquad \text{(C.5)}$$

This is the probability that the correct exemplar is within the nearest n neighbours. The classification error, ε, is then simply:

$$\varepsilon = 1 - \sum_{i=0}^{n-1} (1 - \varepsilon_p)^{N-1-i} \varepsilon_p^i \begin{pmatrix} N-1 \\ i \end{pmatrix} \qquad \text{(C.6)}$$

Appendix D

The Surface Area of a Hyper-spherical Patch

Given a hyper-sphere with radius r in an n-dimensional space we would like to determine the *area*, $A_n(\theta, r)$, of a surface patch defined by an angle θ. Figure D.1 shows a slice through an n-dimensional hyper-sphere with the surface patch defined by the angle θ.

We begin by observing that a slice through the n-dimensional hyper-spherical patch is a complete n-1-dimensional hyper-sphere. If we define the position of the slice by the angle α, as shown in Figure D.1, then the radius of this hyper-sphere is $r \sin \alpha$. A small change, $\delta\alpha$, in the angle α defines a *ring* on the surface patch of width $r\delta\alpha$. The area of this ring, $\delta A_n(\theta, r)$, is then given by:

$$\lim_{\delta\alpha \to 0} \delta A_n(\theta, r) = A_{n-1}(\pi, r \sin \alpha) r \delta\alpha \tag{D.1}$$

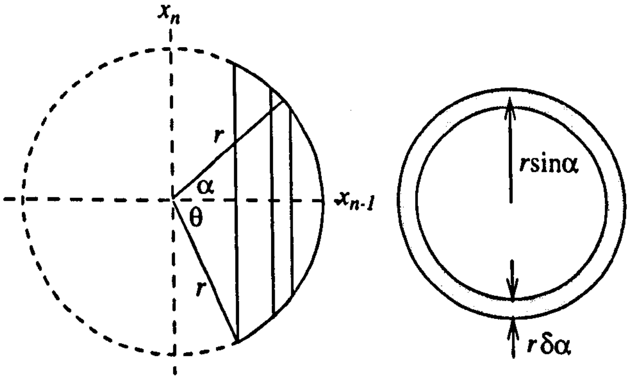


Figure D.1: Slicing through an n-dimensional hyper-spherical patch produces a complete $n-1$-dimensional hyper-sphere.

The area of the complete surface patch can then be determined by integration.

$$A_n(\theta, r) = \int_0^\theta A_{n-1}(\pi, r \sin \alpha) r d\alpha \tag{D.2}$$

It is both easily shown and intuitive that a change in the radius of a hyper-sphere by a factor a has the following effect on the surface area of the hyper-sphere:

$$A_n(\pi, ar) = a^{n-1} A_n(\pi, r) \tag{D.3}$$

For example, doubling the radius of a circle (a 2-dimensional hyper-sphere) doubles its circumference whilst doubling the radius of a sphere (a 3-dimensional hyper-sphere) quadruples its surface area. Using this relationship, Expression D.2 can be re-expressed as:

$$A_n(\theta, r) = \int_0^\theta \sin^{n-2} \alpha A_{n-1}(\pi, r) r d\alpha \tag{D.4}$$

And because $A_{n-1}(\pi, r)$ and r are independent of the angle α this can be simplified to:

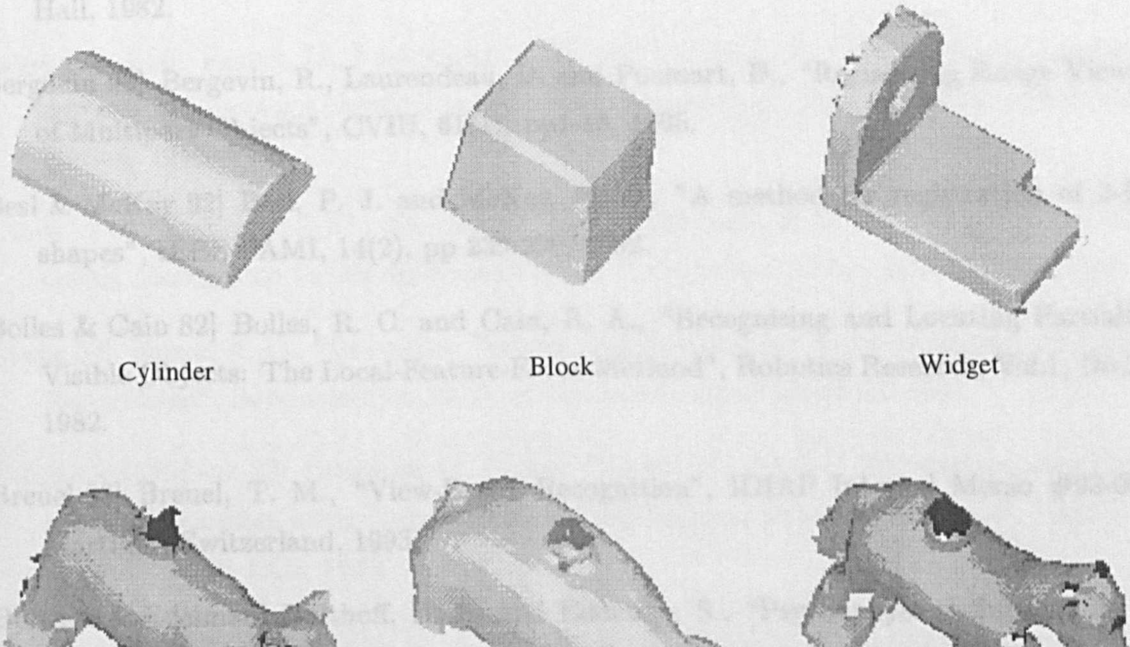
$$A_n(\theta, r) = A_{n-1}(\pi, r) r \int_0^\theta \sin^{n-2} \alpha d\alpha \tag{D.5}$$

This provides a recursive expression for the area of a hyper-sphere which is terminated by the 2-dimensional case where:

$$A_2(\pi, r) = 2\pi r \tag{D.6}$$

Appendix E

Database of 3D Shape Models



Cylinder Block Widget

Calf Pig Pony

Figure E.1: 3-Dimensional surface models.

Bibliography

[Ballard 81] Ballard, D. H., "Generalizing the Hough Transform to Detect Arbitary Shapes", Pattern Recognition, Vol. 13 No. 2, pp. 111-122, 1981.

[Ballard & Brown 82] Ballard, D. H. and Brown, C. M., "Computer Vision", Prentice Hall, 1982.

[Bergevin 95] Bergevin, R., Laurendeau, D. and Poussart, D., "Registering Range Views of Multipart Objects", CVIU, 61(1), pp1-16, 1995.

[Besl & McKay 92] Besl, P. J. and McKay, N. D., "A method for registration of 3-D shapes", IEEE PAMI, 14(2), pp 239-256, 1992.

[Bolles & Cain 82] Bolles, R. C. and Cain, R. A., "Recognising and Locating Partially Visible Objects: The Local-Feature-Focus Methiod", Robotics Research, Vol.1, No.3, 1982.

[Breuel 93] Breuel, T. M., "View-Based Recognition", IDIAP Internal Memo #93-09, Martigny, Switzerland, 1993.

[Bulthoff & Edelman] Bulthoff, H. H. and Edelman, S., "Psychophysical Support for a Two-dimensional View Interpolation Theory of Object Recognition", Proc. Natl. Acad. Sci. USA. Vol.89, pp 60-64, 1992.

[Burns *et al* 93] Burns, J. B., Weiss, R. S. and Riseman, E. M., "View Variation of Point-Set and Line-Segment Features", IEEE trans. Pattern Analysis and Machine Intelligence, Vol. 15, No. 1, pp. 51-68, 1993.

[Canny 86] Canny, "Computational Approach to Edge Detection", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-8 No. 6, 1986.

[Cootes & Taylor 92] Cootes, T. F. and Taylor, C. J., "Active Shape Models - 'Smart Snakes'", Department of Medical Biophysics, University of Manchester, 1992.

[Cosgriff 60] Cosgriff, R. L., "Identification of Shape", Ohio State University Research Foundation, Columbus, Rep. 820-11, ASTIA AD 254 792, 1960.

[Courtney Thacker & Clark 97] Courtney, P., Thacker, N. A. and Clark, A. F., "Algorithmic Modelling for Performance Evaluation", Machine Vision and Applications, Vol. 9, No. 5, pp. 219-228, 1997.

[Davies 90] Davies, E. R., "Machine Vision - Theory, Algorithms, Practicalities", Academic Press, 1990.

[Duda & Hart 72] Duda, R. O. and Hart, P. E., "Use of the Hough Transformation to Detect Lines and Curves in Pictures", Comm. ACM, Vol. 15, pp. 11-15, 1972.

[Dudani *et al* 77] Dudani, S. A., Breeding, K. J. and McGhee, R. B., "Aircraft Identification by Moment Invariants", IEEE trans. Computing, Vol. 26, No. 1, pp. 39-45.

[Eggert *et al* 96] Eggert, D., Fitzgibbon, A. W. and Fisher, R. B., "Simultaneous registration of multiple range views for use in reverse engineering", Proc. ICPR96, pp243-247, Vienna, 1996.

[Evans *et al* 93] Evans, A. C., Thacker, N. A. and Mayhew, J. E. W., "The Use of Geometric Histograms for Model-Based Object Recognition", Proc. BMVC 93, pp429.

[Evans 94] Evans, A. C., "Geometric Feature Distributions for Shape Representation and Recognition", Submitted for Ph.D. at the University of Sheffield, 1994.

[Faugeras & Hebert 83] Faugeras, O. D. and Hebert, M., "A 3-D Recognition and Positioning Algorithm using Geometric Matching between Primitive Surfaces", Proc. 8th IJCAI, pp-996-1002, 1983.

[Fischler & Bolles 81] Fischler, M. A. and Bolles, R. C., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", Communications of the ACM, Vol. 24, pp381, 1981.

[Fisher 89] Fisher, R. B., "From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis", John Wiley & Sons, 1989.

[Freeman 61] Freeman, H., "On the Encoding of arbitrary geometric configuration", IRE Trans. on Electronic Computers, Vol. 10, No. 2 ,pp 260-268, 1961.

[Fukunaga & Olsen 71] Fukunaga, K. and Olsen, D. R., "An Algorithm for Finding Intrinsic Dimensionality of Data", IEEE trans. on Comp., Vol. C-20, pp. 176-183, 1971.

[Fukunaga & Flick 84] Fukunaga, K. and Flick, T. E., "Classification Error for a Very Large Number of Classes", IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 6, pp. 779-788, 1984.

[Fukunaga 90] Fukunaga, K. "Introduction to Statistical Pattern Recognition", Academic Press, 2nd Edition, 1990.

[Garland & Heckbert 97] Garland, M. and Heckbert, P. S., "Surface Simplification using Quadric Error Metrics", SIGGRAPH97, pp209-216, 1997.

[Gaston & Lozano-Perez 84] Gaston, P. C. and Lozano-Perez, T., "Tactile Recognition and Localization Using Object Models: The Case of Polyhedra on a Plane", IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 3, pp 257-266, 1984.

[Grimson & Lozano-Perez 87] Grimson, W. E. L. and Lozano-Perez, T., "Localizing Overlapping Parts by Searching the Interpretation Tree", IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 4, pp. 469-482, 1987.

[Grimson 90] Grimson, W. E. L., "Object Recognition by Computer: The Role of Geometric Constraints"

[Haralick 92] Haralick, R. M., "Performance Characterization in Computer Vision", Proc. BMVC92, pp1-8, Leeds, 1992.

[Haralick 96] Haralick, R. M., "Propagating Covariance in Computer Vision", Series in Machine Perception and Artificial Intelligence Vol. 25: Studies in Pattern Recognition, Ed. H. Freeman, pp. 171-182, World Scientific, 1996.

[Harris & Stephens 88] Harris, C. and Stephens, M., "A Combined Corner and Edge Detector", Proc. Fourth Alvey Vision Conference, pp147-151, 1988.

[Hoppe *et al* 92] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., "Surface Reconstruction from Unorganised Points", Computer Graphics, 26(2), pp71-78, 1992.

[Hough 62] Hough, P. V. C., "Method and Means for Recognizing Complex Patterns", US Patent 3069654.

[Hu 62] Hu, M. K., "Visual Pattern Recognition by Moment Invariants", IRE trans. Information Theory, Vol. 8, pp. 179-187, 1962.

[Jiang & Bunke 91] Jiang, X. Y. and Bunke, H., "Simple and Fast Computation of Moments", Pattern Recognition, Vol. 24, No. 8, pp. 801-806, 1991.

[Johnson & Hebert 97] Johnson, A. E. and Hebert, M., "Recognizing Objects by Matching Oriented Points", Proc. CVPR97, pp684-689, 1997.

[Kumar & Rockett] Kumar, R. and Rockett, P.I., "Triplet Geometric Representation: A Novel Scale, Translation and Rotation Invariant Feature Representation based on Geometric Constraints for Recognition of 2D Object Feature", Image and Vision Computing, 15(3), pp235-24, 1997.

[Lamdan & Wolfson 88] Lamdan, Y. and Wolfson, H., "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme", Proc. IEEE ICCV88, Tampa, pp. 238-249, 1988.

[Lanitis *et al* 94] Lanitis, A., Taylor, C. J. and Cootes, T. F., "An Automatic Face Identification System using Flexible Appearance Models", Proc. BMVC94, York, pp65-74, 1994.

[Matusita 55] Matusita, K., "Decision Rules Based on Distance for Problems of Fit, Two Samples and Estimation", Ann. Mathematical Statistics, Vol. 26, pp. 631-641, 1955.

[O'Rourke 81] O'Rourke, J., "Dynamically Quantised Spaces for Focusing the Hough Transform", 7th IJCAI, Vancouver, pp737-739, 1981.

[Pettis 79] Pettis, K. W., Bailey, T. A. and Jain, A. K., "An Intrinsic Dimensionality Estimator from Near-Neighbor Information", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-1 No. 1, pp. 25-36, 1979.

[Rak 91] Rak, S. J. and Kolodzy, P. J., "Performance of a Neural Network Based 3-D Object Recognition System", SPIE Automated Object Recognition, Vol. 1471, pp. 177-184.

[Riocreux, Thacker & Yates 94] Riocreux, P. A., Thacker, N. A. and Yates, R. B., "An Analysis of Pairwise Geometric Histograms for View-Based Object Recognition", Proc. BMVC, York, pp.75-84, 1994.

[Rothwell *et al* 92] Rothwell, C. A., Zisserman, A., Forsyth, D. A. and Mundy, J. L., "Canonical Frames for Planar Object Recognition", Proc. ECCV, Santa Margherita, Italy, pp. 757-772, 1992.

[Schwartzmann & Vidal 75] Schwartzmann, D. H. and Vidal, J. J., "An Algorithm for Determining the Topological Dimensionality of Point Clusters", IEEE Trans. on Comp., Col. C-24, No. 12, pp. 1175-1182, 1975.

[Sobel 70] Sobel, I., "Camera Models and Machine Perception", Stanford Artificial Intelligence Project, Meme AIM-121, Computer Science Dept., Stanford University, 1970.

[Sonka *et al* 93] Sonka, M., Hlavac, V. and Boyle, Roger, "Image Processing, Analysis and Machine Vision (1st Edition)", Chapman and Hall Computing Series, 1993.

[Stephens 90] Stephens, R.S., "A Probabilistic Approach to the Hough Transform", Proc. BMVC, 1990.

[Strachan *et al* 90] Strachan, N. J. C., Nesvadba, P. and Allen, A. R., "Fish Species Recognition by Shape Analysis of Images", Pattern Recognition, Vol. 23, No. 5, pp. 539-544, 1990.

[Stricker 94] Stricker, M. A., "Bounds for the Discrimination Power of Color Indexing Techniques", Proc. SPIE 94, San Jose, 1994.

[Stricker & Swain 94] Stricker, M. A. and Swain, M., "The Capacity of Color Histogram Indexing", Proc. CVPR 94, 1994.

[Swain & Ballard 91] Swain, M. J. and Ballard, D. H., "Colour Indexing", Intern. Journal of Computer Vision, Vol. 7, No. 1, pp. 11-32, 1991.

[Thacker *et al*] Thacker, N. A., Abraham, I. and Courtney, P., "Supervised Learning Extensions to the CLAM Network", sumbitted to Neural Networks.

[Thacker & Mayhew 90] Thacker, N. A. and Mayhew, J. E. W., "Designing a Layered Network for Context Sensitive Pattern Classification", Neural Networks, Vol.3, pp291-299, 1990.

[Thacker*et al* 95] Thacker, N. A., Riocreux, P. A. and Yates, R. B., "Assessing the Completness Properties of Pairwise Geometric Histograms", Image and Vision Computing, Vol.13, No.5, pp423-429, 1995.

[Thacker *et al* 97] Thacker, N. A., Aherne, F. J. and Rockett, P. I., "The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data", STIPR97, 1st International Workshop on Statistical Techniques in Pattern Recognition, Prague, Czech Republic, 1997.

[Thirion 96] Thirion, J., "New Feature Points based on Geometric Invariants for 3D Image Registration", IJCV, 18(2), pp121-137, 1996.

[Trunk 76] Trunk, G. V., "Statistical Estimation of the Intrinsic Dimensionality of a Noisy Signal Collection", IEEE trans. on Comp., Vol C-25, No. 2, pp. 165-171, 1976.

[Verveer & Duin 95] Verveer, P. J. and Duin, P. W., "An Evaluation of Intrinsic Dimensionality Estimators", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-17 No. 1, pp. 81-86, 1995.

[Wechsler & Zimmerman 88] Wechsler, H. and Zimmerman, G. L., "2D Invariant Object Recognition Using Distributed Associative Memory", IEEE trans. Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, pp. 811-821, 1988.

[Xia 89] Xia, Y., "Skeletonization Via the Realization of the Fire Front's Propagation and Extinction in Digital Binary Shapes", IEEE trans. Pattern Analysis and Machine Intelligence, Vol. 11 No. 10, pp. 1077-1086, 1989.

[Zahn & Roskeis 72] Zahn, C. T. and Roskies, R.Z., "Fourier Descriptors for Plane Closed Curves", IEEE trans. Computing, Vol. 21, No. 3, pp. 269-281, 1972.